INAUGURAL – DISSERTATION

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich–Mathematischen Gesamtfakultät

der

Ruprecht–Karls–Universität

Heidelberg

vorgelegt von

Dipl.-Math. Andreas Schmidt

aus Nürtingen

Tag der mündlichen Prüfung:

. . . .

Direct Methods for PDE-Constrained Optimization Using Derivative-Extended POD Reduced-Order Models

Gutachter:
Prof. Dr. Dres. h.c. Hans Georg Bock

. . . .

Zusammenfassung

In der vorliegenden Arbeit untersuchen und entwickeln wir Methoden zur Lösung von Optimierungsproblemen mit partiellen Differentialgleichungsbeschränkungen (PDEs) basierend auf Strategien der Modell-Ordungs-Reduktion (MOR). Die Methoden kombinieren einen direkten Lösungsansatz mit der Modellreduktion durch die Proper orthogonal decomposition (POD) und die Discrete empirical interpolation method (DEIM). Ordungsreduzierte Modelle (ROMs) werden zur Approximation von hochdimensionalen dynamischen Systemen verwendet, welche aus einer Ortsdiskretisierung der partiellen Differentialgleichung resultieren. Bei der Verwendung dieser Modelle in einem Optimierungsalgorithmus taucht häufig das Problem auf, dass der Gradient nicht adäquat approximiert werden kann. Wir stellen Methoden zur Verbesserung des ROMs für den Einsatz in der Optimierung vor, die auf der Verwendung von Ableitungsinformationen im POD und DEIM Unterraum basieren.

In Abhängigkeit der Optimierungsvariable unterscheiden wir zwischen zwei Arten von Fehlern in Größen die einmal mit Hilfe des hochdimensionalen Modells berechnet und einmal mit Hilfe des ROM approximiert werden: Dem Rekonstruktionsfehler, welcher mit demselben $\bar{q}$ berechnet wird, welches auch zur Konstruktion des ROMs genutzt wird und dem Vorhersagefehler, der Aussagen über Approximationen macht, die an einer Stelle q ausgewertet werden, wobei das ROM an einer Stelle $\bar{q} \neq q$ aufgestellt wird. Die neuen Ergebnisse für den Rekonstruktionsfehler umfassen Abschätzungen für die Lösungen der adjungierten Gleichung und der Sensitivitätengleichungen sowie für den Gradienten der Zielfunktion. Mit Hilfe dieser Abschätzungen erläutern wir, wie die POD und DEIM Basen mit entweder adjungierter order Sensitivitätsinformation erweitert werden sollten. Die verbesserten ROMs erlauben ein Steuern des Rekonstruktionsfehlers für die Zielfunktion als auch des Gradienten bis zur Maschinengenauigkeit. Des Weiteren schlagen wir eine neue Abschätzung zur POD Vorhersage der Zielfunktion des Optimierungsproblems vor, welche Aussagen in einer Umgebung von q macht mit welchem das ROM konstruiert wurde. Im Fall von POD und DEIM Basen die mit Sensitivitäten erweitert wurden erhalten wir ein analoges Resultat für Lösungen der Zustandsgleichung. Die durch Ableitungen erweiterten ROMs werden dann verwendet um adaptive Algorithmen zur Lösung von Optimalsteuerungs- und Parameterschätzproblemen zu entwickeln. Dies resultiert in enormen Laufzeitverbesserungen für die Optimierung, wobei gleichzeitig eine hohe Approximationsgüte der Lösung des ursprünglichen Problems gesichert wird. Für Parameterschätzprobleme stellen wir einen neuen a posteriori Fehlerschätzer vor, welcher Aussagen über die Qualität von suboptimalen Lösungen macht, die mit Hilfe des ROMs berechnet werden.

Ein weiterer wesentlicher Beitrag ist eine Diskussion zum Verhältnis zwischen Diskretisieren-Dann-Optimieren (DTO) und Optimieren-Dann-Diskretisieren (OTD) im Kontext von auf MOR basierender Optimierung. Wir untersuchen Vorteile und Nachteile dieser beiden Ansätze und diskutieren, inwiefern unsere Methoden Eigenschaften von beiden aufweisen. Wir stellen zudem Beispiele repräsentativer Optimierungsprobleme vor, in welchen gewöhnliche POD/DEIM ROMs ein inakzeptables Verhalten zeigen, die jedoch mit ableitungserweiterten ROMs erfolgreich gelöst werden können. Des Weiteren wurden die entwickelten Methoden implementiert, wobei auf eine effiziente Realisierung Wert gelegt wurde, welche wichtig für die Untersuchung des Potentials von MOR ist. Wir demonstrieren das praktische Verhalten der vorgeschlagenen Algorithmen und die Überlegenheit von ableitungserweiterten ROMs im Vergleich zu konventionellen ROMs an Beispielen von zwei akademischen und einer industrierelevanten Anwendung, welche verschiedene Herausforderungen an den Modellreduktionsansatz in der Optimierung stellen.

Abstract

In this thesis we analyze and develop methods based on model order reduction (MOR) for the solution of optimization problems constrained by time-dependent partial differential equations (PDEs). The methods combine a direct solution approach with model reduction via proper orthogonal decomposition (POD) and the discrete empirical interpolation method (DEIM). The reduced-order models (ROMs) are used to approximate the high-dimensional dynamic systems originating from a spatial discretization of a PDE. However, when used in an optimization algorithm, conventional POD/DEIM ROMs often lack the ability to give adequate approximations of the gradient. We propose methods for a suitable enhancement of the ROMs for the optimization purpose which are based on the inclusion of derivative information in the POD and DEIM subspaces.

We distinguish two types of error between quantities evaluated with the high-dimensional model and its ROM approximation in dependency on the optimization variable q: The reconstruction error which is evaluated with the same $\bar{q}$ which is used constructing the ROM and the prediction error which assesses approximations at q with a ROM constructed at $\bar{q} \neq q$. The novel reconstruction results we present include estimates for solutions of the adjoint equation and the sensitivity equations as well as for the gradient of the objective function. Based on the estimates we explain how the POD and DEIM bases should be extended with either adjoint or sensitivity information. The enhanced ROMs allow control of the reconstruction error for the objective and its gradient up to machine precision. Moreover, we propose a POD prediction estimate for the objective of the optimization problem in a neighborhood of q where the ROM is constructed. In case of sensitivity-extended POD and DEIM bases we give an analogous result for solutions of the states. The derivative-extended ROMs are then used to develop adaptive algorithms for the solution of optimal control and parameter estimation problems which results in great runtime improvements for the optimization while ensuring high approximation quality of the solution of the original problem. For the parameter estimation case a novel a posteriori error estimate is proposed which assesses the quality of suboptimal solutions obtained with the ROM.

A further fundamental contribution is a discussion of discretize-then-optimize (DTO) vs. optimize-then-discretize (OTD) approaches in the context of MOR for optimization. We analyze advantages and disadvantages of both approaches and discuss to which extent our methods exhibit properties of either strategy. We also give examples of representative optimization problems in which standard POD/DEIM ROMs show an inacceptable behavior and can be successfully solved by derivative-extended ROMs. We have further implemented the developed methods emphasizing an efficient realization which is important for the investigation of the MOR potential. We showcase the practical performance of the proposed algorithms and the superiority of derivative-extended over conventional ROMs on two academic and one industry-relevant application which exhibit a variety of challenges for the model reduction approach in optimization.

Contents

Introduction

The combination of mathematical modeling, simulation, and optimization has become an important procedure to tackle a variety of problems that arise in industry and academia. Many effects and processes that we face, e.g., in fields such as chemical engineering, geophysics, and computational fluid dynamics, can be described and modeled with the help of time-dependent partial differential equations (PDEs). After the modeling step, we are confronted with the task of a numerical simulation of the effect or the process, meaning that the governing PDEs need to be solved. This is often already a challenging task in itself and reasonable discretizations of the equations may involve up to 10^9 variables. Given the model and a simulation tool at hand, a variety of optimization tasks can be considered, e.g., parameter estimation problems with the aim of the calibration of the model or the optimal control of the underlying process. These are even more challenging problems as their solution usually requires multiple simulations of the model and additional evaluations of derivatives with respect to the optimization variables.

In this thesis we develop algorithms that significantly speed up the optimization procedure and that are based on model order reduction (MOR) via proper orthogonal decomposition (POD). The general idea is to approximate the large spatially discretized PDE models by smaller ones that we can use in the optimization algorithm. Our contribution is to enhance these so-called reduced-order models (ROMs) with derivative information for the optimization purpose. The theoretical foundations are provided for, however not restricted to, optimal control and parameter estimation problems that are constrained by semilinear time-dependent PDEs. We have implemented the developed algorithms and apply the novel methods to scientific and practical optimization problems.

POD model reduction is the subject of increasing attention in the PDE-constrained optimization community and is particularly popular when nonlinearities are involved in the model (see the introduction to POD for optimization by Sachs and Volkwein [99]). The "hidden beauty of the proper orthogonal decomposition", as Aubry refers to it in [9], has attracted researchers of many areas. In image processing POD has been extensively used to extract the essential information from large amounts of data [44, 98]. Historically, important contributions to the development of POD were made in the study of turbulence and coherent structures in the context of fluid dynamics [18, 84, 107]. POD with the aim of model reduction was used for the Burgers' equation [74] and the Navier–Stokes equation [29]. Applications to a diversity of models in practice are also found, such as saturated groundwater flows [122], lithium-ion batteries [80], airfoil design [27], and particulate processes which play an important role in the chemical and pharmaceutical industries [85]. Fundamental contributions to the use of POD for the optimization purpose were made in recent years. Among these we mention the articles by Arian et al. [6], Hinze and Volkwein [62], Tröltzsch and Volkwein [112], and Kunisch and Volkwein [77]. In this thesis we follow the conceptual POD model reduction approach in the latter works. We suggest extensions of this approach by derivative-enhancement of the reduced-order models and discuss the importance of this for optimization which we believe has not received sufficient attention in literature so far.

When speaking of optimization the mathematician has in mind the goal of reducing a certain objective to a minimum possible value, such that a set of given constraints is satisfied. In optimal control problems, on the one hand, the optimization variables are states that must satisfy a dynamic model equation and, on the other hand, control variables from an infinite-dimensional function space that need to satisfy certain restrictions. To solve the problem, at some point a discretization must be carried out. Typically for both types of optimization variables many discrete variables are introduced. However, it is foremost the states and their corresponding constraints that are of large dimension. A common technique is the method-of-lines approach, where the spatial discretization is carried out first, e.g., using finite element methods (FEMs) and then an integration in time is performed, e.g., with an implicit Euler method. In FEM methods the state solution is approximated via basis functions with local support employing a Galerkin discretization to obtain a typically large system of ordinary differential equations (ODEs). To this ODE we refer as the high-fidelity (HiFi) problem or HiFi model as we assume that we can trust the high-dimensional discretization to be adequately accurate for the infinite-dimensional problem. The idea of POD is to construct tailored global basis functions that contain information about the dynamic system at hand. Practical experience shows that this information can already be captured by only a few basis functions ($5-30$) in many applications. The model reduction procedure we employ can be summarized by two major steps. First, the POD basis is constructed from so called snapshots [107] that are obtained by a solution of the high-fidelity system. In a second step a projection of the large-scale system onto the POD subspace is performed. This yields a small problem that we refer to as ROM or also *surrogate model*.

While the dimension of the surrogate model is small, the evaluation of the nonlinear model components is still of the complexity of the HiFi discretization. This computational drawback was overcome by the introduction of the discrete empirical interpolation method (DEIM) [31]. With this enhancement the surrogate model becomes entirely independent of the dimensionality of the full spatial discretization and its evaluation costs can be further significantly reduced.

Regarding the optimization task itself, with the presence of dynamic constraints, several theoretical and practical obstacles need to be overcome – in particular in the presence of PDEs [61, 112]. We distinguish two main ways of tackling the problem, namely the discretize-then-optimize (DTO) and optimize-then-discretize (OTD) strategies, which both have advantages and disadvantages. With DTO we first discretize all infinite-dimensional components and subsequently solve the finite-dimensional problem. In contrast, with OTD the two steps are performed in reverse order. Often the latter is referred to as *indirect method* while DTO approaches are called *direct methods*. 'Direct' refers to the fact that during the optimization one explicitly iterates on the control variables while 'indirect' reflects that the control variables are eliminated from the optimization problem in an OTD approach. However, there are blurred lines between the respective approaches and whether the dichotomies OTD/DTO and direct/indirect should be used as synonyms is ambiguous in the optimization community. Unambiguously, in this thesis we follow a direct approach converting the infinite-dimensional problem into a nonlinear programming problem (NLP). The established direct concepts are then extended by the idea of model reduction via POD and we discuss POD regarding its use in DTO and OTD approaches. In particular we show that our methods exhibit properties from both approaches, thus, making the classification of our method a matter of perspective. We favor the direct point of view as relatively little knowledge is necessary to solve optimization problems with PDE constraints which are of complicated nature theoretically as well as numerically. We believe that user friendly interfaces are essential to make sophisticated numerical techniques widely accessible to practitioners.

We work with a reduced approach to optimal control, i.e., we eliminate the state variables from the optimal control problem by introduction of a solution operator. Our strategy is to approximate the discretized solution operator via the surrogate model, which is typically significantly cheaper to evaluate. To the reduced problem we apply derivative-based optimization methods. Thus, to solve the optimization problem, not only the computation of the objective function is necessary (which involves a simulation of the dynamic equations), but also we need to compute derivatives of the objective with respect to the control variables.

Here two main concepts can again be pointed out: The adjoint approach and the sensitivity approach. The two differ mainly in their complexity of computing the full Jacobian J of a function $f : \mathbb{R}^n \to \mathbb{R}^m$, which is given by $\mathcal{O}(n)$ in the sensitivity case and by $\mathcal{O}(m)$ in the adjoint case. In this thesis we show how for both cases the reduced-order model must be constructed to successfully use it in optimization. Current state-of-the-art POD reduced-order models lack the ability to appropriately approximate the required derivatives. We present examples where this leads to a failure of the model reduction approach. We explain, how by a suitable inclusion of derivative information in the POD and DEIM models, this drawback can be overcome.

Contributions of this thesis

The aim of this thesis is the investigation and the development of efficient methods which significantly speed up the numerical solution of optimal control and parameter estimation problems constrained by parabolic semilinear PDEs. The key aspects are the use of derivative-extended POD reduced-order models in combination with direct optimization methods. We provide numerical algorithms and the necessary theoretical foundations. The importance of the derivative enhancement for POD/DEIM surrogate models, which serve as replacements for the high-fidelity model in the optimization, is demonstrated by means of particular examples. We have efficiently implemented the proposed methods and apply them to academic and industry-relevant applications. Numerical results show that our POD model-reduction approach yields savings in computation time of up to a factor of ten for moderately sized dynamic problems while accurate solutions of the original problem are obtained. We demonstrate also numerically that the algorithms based on derivative-extended surrogate models yield great improvements compared to standard POD optimization algorithms.

In applications of POD and DEIM for optimization, the surrogate models are usually constructed at a reference optimization variable configuration such that they well-approximate solutions of the high-fidelity discretized state problem (or quantities depending on the solutions) at this configuration. To this approximation error we refer as the *reconstruction error*. However, POD ROMs often loose their ability to give good approximations if the optimization variable differs from the reference configuration. To this approximation error we refer as the *prediction error*. While the problem with POD prediction has been studied and tackled before, e.g., in [6, 65, 77], a further problem of POD ROMs is their lack of the ability to give good approximations of the derivatives already in the reconstruction sense. We believe that this drawback has not received sufficient attention and propose methods to overcome it.

On the algorithmic side the main achievement is a sophisticated construction of the surrogate model for the optimization purpose. To this end, we propose an inclusion of either adjoint or sensitivity information in the projection spaces which we call derivative-extended proper orthogonal decomposition (DEPOD) and derivative-extended discrete empirical interpolation method (DEDEIM). The use of sensitivity information in POD subspaces was suggested by Noor [88] and Peterson [91]. They show numerically that derivative inclusion may improve the POD prediction problem. The inclusion of sensitivity information in a POD ROM for the sake of derivative approximations was suggested by Zimmermann [125] and by the author of this thesis in [104]. When using adjoint information in POD ROMs, improved numerical performance was observed by Diwoky and Volkwein [39] as well as by Hinze and Volkwein [62]. An enhancement of the DEIM subspace with derivatives was not reported.

In this thesis we provide a rigorous analysis of the effects of an inclusion of derivative information in the POD and DEIM subspaces for the optimization. These results consist of novel reconstruction error estimates for the the adjoint and the sensitivity equations as well as for the gradient of the objective function of the optimization problem. The estimates are given for the time-continuous and the time-discrete case. The reconstruction results are an extension of the results for the state equations in [32] to the derivative equations in a more general problem formulation. The estimates state that the reconstruction errors are bounded by a constant multiplied by the sum over neglected eigenvalues belonging to truncated POD basis functions. Based on these estimates the construction of the DEPOD and DEDEIM subspaces is explained. Using DEPOD and DEDEIM we are able to numerically compute approximations with a reconstruction error close to machine precision. In addition we make sure that the derivative-extended surrogate models are enhanced such that they allow accurate reconstruction results in the direct approach, where we base the derivative computation on automatic differentiation (AD). Particular care must be taken in the DEIM projection step where the situation is more subtle.

Moreover, we present a novel POD prediction estimate for the objective of a nonlinear reduced optimal control problem, extending the results in [30] which are based on the Taylor expansion. Given the estimate we can ensure an alike local behavior of the objective when evaluated with the high-fidelity or the DEPOD/DEDEIM surrogate model around a control configuration, where the surrogate is constructed. Due to the fact that adjoint and sensitivity-extended POD bases have the same effect on the objective regarding POD prediction, the adjoint case can be interpreted as goal oriented, as here less information needs to be added to the POD subspace. In the sensitivity case we give an analogous result for local POD prediction of the states which is important for parameter estimation problems.

In the context of parameter estimation we present an a posteriori error estimate for the distance between a solution obtained with the high-fidelity model and with the surrogate model. The estimate is based on the local contraction theorem of Bock [21]. To solutions based on surrogate models we refer as suboptimal, as in general it is unclear how far they deviate from the exact solution due to the lack of reliable a priori error estimation for POD. The estimates are similar to the a posteriori error estimates for optimal control problems of Tröltzsch and Volkwein in [112] and are here carried over to parameter estimation problems.

We then exploit our theoretical findings to construct algorithms that efficiently solve optimal control and parameter estimation problems. These are based on adaptive algorithms as in [1, 61] that were proposed to overcome the problem of suboptimal solutions. With the use of DEPOD and DEDEIM bases these algorithms can be improved as the POD prediction error for which we give a local bound is closely related to the problem of suboptimal solutions. By controlling the reconstruction errors we can guarantee that with the suboptimal solution we have also found a solution to the high-fidelity optimization problem.

A further fundamental contribution is a comprehensive study of POD regarding the dichotomy of DTO and OTD. Similar to [8] we distinguish the two conceptual approaches approximate-then-optimize (ATO) and optimize-then-approximate (OTA) and discuss their relation to DTO and OTD. A classification of the existing algorithms into one of the two categories is possible. As for DTO and OTD, the two approaches have both advantages and disadvantages. We point out that many of the beneficial properties of the two strategies can be carried over to our proposed algorithms. Essentially this is achieved by combining a Galerkin discretization in space, the principle of internal numerical differentiation (IND) [19] for time integration and the use of DEPOD/DEDEIM subspaces for the model reduction step.

In the numerical results we show reconstruction error examples to give the reader an idea of the practical approximation properties we can expect from derivative-extended reduced-order models. Note that the efficiency of the overall approach still relies on a fast decay of the eigenvalues in the POD decomposition. Moreover, we present optimal control examples where a standard application of common POD techniques leads to a breakdown of the method which can, however, be successfully solved with our methods. We apply the developed optimization algorithms to two academic problems that contain nonlinear and space- or time-dependent controls as well as a problem from chemical industry. With the results we show the great potential of the approach and also point out limitations and propose further ideas for how these could be overcome.

The results for a sensitivity-extended POD basis and the POD a posteriori error estimate in parameter estimation are published by the present author in Schmidt et al. [104]. These are included for completeness and complemented, in particular, by the reconstruction estimates and the extension to the DEIM projection.

We sum up the main contributions of this thesis:

- A POD reconstruction error estimate for adjoint and sensitivity equations and the gradient of the reduced objective function
- A POD prediction estimate for the reduced objective (adjoint and sensitivity case) and the states (sensitivity case)
- A suitable extension of the POD and DEIM subspaces with either adjoint or sensitivity information
- Adaptive algorithms for the solution of optimal control and parameter estimation problems using derivative-extended reduced-order models
- An a posteriori error estimate for POD suboptimal solutions of parameter estimation problems
- A Discussion of POD model reduction approaches regarding the question of discretize-then-optimize vs. optimize-then-discretize and the relation to our methods
- An efficient implementation of the proposed methods
- Application of our methods to problems with nonlinear and space- or time-dependent controls and a industrial model problem

Thesis overview

The thesis is structured in three major parts. In Part 1 we formulate the optimization problems together with the dynamic constraints and introduce the direct approach strategies we employ for their solution. Part 2 contains the basic concepts for reduced-order modeling based on POD and DEIM and the main theoretical and algorithmic results. In Part 3 we present practical results where the developed methods that have been implemented are applied to two academic and one industry-relevant applications.

In Chapter 1 we introduce the optimal control and the parameter estimation problem together with the class of semilinear parabolic PDEs that occur as constraints. We establish the functional analytic framework in which we solve these problems, derive the reduced problems, and state necessary optimality conditions. These conditions involve first-order derivatives of the reduced objective with respect to the control variables. We present two possibilities for their evaluation, namely the adjoint and sensitivity approach.

In Chapter 2 we dedicate ourselves to the numerical strategies that we use to discretize and solve the infinite-dimensional problems. First, we explain the main differences between the DTO and OTD strategies and briefly address the issue of naming conventions. The conversion of the infinite-dimensional problem into a finite-dimensional one is done via the method of lines. Thus, we follow a hierarchical procedure in the discretization process, first employing a Galerkin discretization in space and then discuss the time discretization using the implicit Euler method. We recall aspects of consistency and stability in the context of

PDEs which become relevant for the practical derivative computation. Then we revisit the optimization problem from the semi-discrete perspective and discuss the relation between infinite-dimensional equations for the derivatives and their semi-discrete counterparts. This is important as large parts of the model reduction discussion are carried out on the semi-discrete level. The fully discretized NLPs are solved with established Newton type methods, namely sequential quadratic programming (SQP) for optimal control and the Gauss–Newton method for parameter estimation problems. In addition Bock's local contraction theorem is recalled, which we need for the POD a posteriori estimate. The chapter is concluded displaying the methods for derivative computation based on AD and IND and peculiarities of IND in the adjoint case are discussed.

We start the model reduction part with a characterization of the POD basis and POD optimality in Chapter 3. POD is discussed in the time-continuous and time-discrete setting and we explain how the reduced-order model is obtained via either Galerkin discretization or projection, both being equivalent in our context. Then we introduce the basic concepts of DEIM, which is used to reduce the computational costs of the nonlinearity in the POD surrogate model. In this chapter we define what we understand by POD reconstruction and POD prediction and sum up existing POD reconstruction results. More specifically, we recall the estimates of Kunisch and Volkwein [75, 76] in function space as the most general case of POD reconstruction estimates. We then extend the estimates of Chaturantabut and Sorensen [32] to a slightly more general scenario. These estimates are in the semi-discrete setting and involve POD and DEIM projection. We present the estimate for time-continuous POD which makes the results independent of particular time discretizations. For the case of the implicit Euler scheme it is shown that the semi-discrete estimates carry over to the fully discrete case. In addition we illustrate the practical behavior of POD on a convection-diffusion-reaction PDE.

Chapter 4 is about the enhancement of the POD/DEIM reduced-order model with derivative information. For the adjoint and the sensitivity case, we shed light on the relation between models of the different hierarchical discretization and approximation levels. In each case we consider the infinite-dimensional model in weak form, the semi-discrete model, the POD projected model and the DEIM projected model. We then carry over the reconstruction results of Chapter 3 to the derivative equations, i.e., reconstruction results in the time-continuous and the fully discrete POD setting. Based on the estimates we describe which derivative information should be included in the POD and DEIM subspaces. Subsequently, we give a reconstruction estimate for the gradient of the reduced optimal control problem which states that the error can be controlled by the size of the POD and DEIM basis. At the end of this chapter we present numerical results that confirm the theoretical findings and provide the reader with an idea of the practical performance of DEPOD/DEDEIM reduced-order models.

We then make use of the DEPOD/DEDEIM reduced-order models in the optimization context in Chapter 5. First, we give the reader an overview on existing methods to combine POD model reduction with optimization. Then POD prediction estimates for the reduced objective of optimal control problems are presented and analogous results are given for POD prediction of the state solutions in the sensitivity case. We present algorithms to solve optimal control and parameter estimation problems and discuss their properties and their relation to existing algorithms. For parameter estimation problems we present an a posteriori error estimate based on the local contraction theorem by Bock [21]. The application of POD model reduction regarding the different strategies DTO and OTD are discussed subsequently. Therefore, we introduce the dichotomy ATO and OTA. We discuss their meaning for optimization with POD and explain to which extent our methods combine advantages of both strategies. We illustrate by means of two numerical examples where a standard application of either one of the strategies can cause the POD approach to give unsatisfactory results.

In the last part of this thesis we apply the developed algorithms to two problems from academia and an industrial example. At first, we give an overview of the strategies employed to implement the proposed algorithms and point out the importance of some design decisions to our results. In either application we solve an optimal control and a parameter estimation problem. We start with a model for 2D heat transport. Here, the optimal control problem is particularly challenging due to the spatially distributed control and the large control variable space we use. The second application is an extension of the Lotka-Volterra predator-prey dynamics to a 2D PDE system for which we briefly discuss aspects of model reduction in case of PDE systems. In the last application we work with a model from petrochemical industry that describes a reactive flow in a tubular reactor. The challenges are the large system size, complicated nonlinearities, and the increased effort for the time integration.

Part I.

Optimization with PDE-Constraints

1. Foundations

In the first chapter we introduce the general form of optimization problems. We describe the optimal control problem and the parameter estimation problem, which we assume to be constrained by a time-dependent PDE. The dynamic model is presented in an abstract form and we comment on particular types of equation considered in this thesis. Moreover, we lay out conceptual solution approaches for a derivative-based optimization and briefly recall necessary optimality conditions in the infinite-dimensional setting. Most of the topics are discussed in the context of optimal control problems. The parameter estimation problem can formally be embedded into the optimal control setting.

	Infinite-dim.	Semi-discrete	Reduced-order model
State variable	y	x	$\widehat{x}$
Adjoint variable	p	z	$\widehat{z}$
Sensitivity variable	$\tilde{y}$	w	$\widehat{w}$
Control/parameter variable	u	q	-

Table 1.1.: Notation for state, adjoint, sensitivity, and control/parameter variables on the different levels of model problem considerations. We distinguish the infinite-dimensional and the semi-discrete case as well as the approximation of the semi-discrete case with reduced-order models.

On notation

In this thesis we combine concepts of several communities, i.e., optimal control for PDEs and ODEs, parameter estimation, model order reduction and numerical analysis. This makes a consistent notation tricky. In Table 1.1 we summarize the most important variables that we use to describe dynamic problems on different hierarchical levels of discretization and approximation. We distinguish the infinite-dimensional and the semi-discrete case obtained via spatial discretization. Via the model reduction approach, for each semi-discrete problem we consider a surrogate model counterpart. Therefore, in case of, e.g., states we use $\widehat{x}$ instead of x. Note the notation for the continuous adjoint p which must not be confused with the notation q for variables in parameter estimation problems where the notation p is common.

The dynamic problems are stated on $I \times \Omega$, with time domain $I := [0, T]$, final time $0 < T < \infty$, the spatial domain $\Omega \subset \mathbb{R}^n$, $n \in \{1, 2, 3\}$, and boundary $\Gamma := \partial\Omega$. The spatial variable we denote by $r \in \Omega$ and the time variable is represented by $t \in I$. We omit the space and time arguments where it is possible to ease legibility. For differentiation we use the common abbreviations, e.g., the derivative of the spatially continuous states with respect to time is denoted by $y_t := \frac{\mathrm{d}}{\mathrm{d}t} y$ while in the ODE context we use $\dot{x} := \frac{\mathrm{d}}{\mathrm{d}t} x$.

The common functional analytic notation is used. We consider the Lebesgue spaces $L^p(D)$, where either $D \subseteq I$ or $D \subseteq \Omega$, and the Sobolev spaces $H^k(D) = W^{k,2}(D)$ where $k \in \mathbb{N}$. The generalization to Lebesgue spaces of mappings from D to a Banach space Z we denote by $L^p(D, Z)$ which is consistent with the previous notation by choosing $Z = \mathbb{R}$. Moreover, for a Hilbert space H we denote its scalar product by $\langle \cdot, \cdot \rangle_H$ and the corresponding norm by $\|\cdot\|_H$. For the natural choice of $H = L^2(\Omega)$ we use $\langle\cdot,\cdot\rangle_\Omega$ and $\|\cdot\|_\Omega$. Analogously the Euclidean inner product is $\langle\cdot,\cdot\rangle$ with the norm $\|\cdot\|$.

1.1. Dynamic model structure

We now present the dynamic model problem and introduce the weak form which is of particular interest for the theoretical investigations and is the starting point for the discretization of the problems. Conceptually, the model reduction techniques proposed in this thesis are directly applicable to any time-dependent problem that is stated in the weak form and can be solved by a method-of-lines approach using a Galerkin discretization in space.

We consider the common setting for optimal control problems with PDE constraints of parabolic type as, e.g., described in [60, 111]. Let V, H be real separable Hilbert spaces with V being dense in H and continuously embedded. We identify H with its dual H^* and consider the Gelfand triple

$$V \hookrightarrow H \cong H^* \hookrightarrow V^*$$

with the continuous injection $\hookrightarrow$ and V^* being the dual of V. Further we denote by $\langle\cdot,\cdot\rangle_{V^*\times V}$ the duality pairing between V^* and V. When dealing with discretization and solutions of the dynamic problems in a weak sense, we apply the inner product on H to approximate duality pairings on V. We briefly comment on this topic in the following remark (for more details see [46, 123]).

Remark 1.1. Let $i : V \hookrightarrow H$ be the continuous injection of V into H. By definition its dual $i^* : H^* \hookrightarrow V^*$ is also continuous and injective. Thus, we have the identity

$$\langle i^*(h), v\rangle_{V^*\times V} = \langle h, i(v)\rangle_H\,, \quad v \in V,\ h \in H \cong H^*.$$

Due to the dense embedding of H^* in V^* and the injectivity of i^*, for each $v^* \in V^*$ there is an element $h \in H$ such that $\langle v^*, v\rangle_{V^*\times V}$ can be uniformly approximated by $\langle h, i(v)\rangle_H$ for all $v \in V$. The identification of v^* and h is typically given from the formulation of v^* by virtue of the Riesz representation theorem.

The abstract evolution problem we consider can be written as

$$y_t(t) = \mathcal{A}(y(t), u(t)), \quad y(0) = y_s \tag{1.1}$$

for almost all $t \in I$ and an initial condition $y_s \in H$. The nonlinear differential operator $\mathcal{A} : V \times R \to V^*$ is supposed to be elliptic in the states. The optimization variables are in general vector valued functions $u \in \mathcal{Q}$, which are either controls or parameters depending on the context. As most general case we consider the Hilbert space

$$\mathcal{Q} \subseteq L^2(I, R), \quad R \subseteq L^2(\Omega, \mathbb{R}^{n_u}).$$

We are also interested in the case of $u \in \mathbb{R}^{n_u}$ which fits into the definition of the control space via the assumption that $u \in \mathcal{Q}$ is constant in $I \times \Omega$. The space $\mathcal{Q}$ is equipped with the norm $\|\cdot\|_\mathcal{Q}$ and the inner product $\langle\cdot,\cdot\rangle_\mathcal{Q}$. Note that the initial condition for the states might also depend on u, however, the extension to this case is straightforward and not explicitly dealt with in the following.

We are interested in solutions of the general evolution problem (1.1) in a weak sense. Conditions for existence and uniqueness as well as for the continuous dependence of y on the data u, y_s depends on the particular form of $\mathcal{A}(y(t), u(t))$ (see Dautray [36], Wloka

[123]). Throughout, we make the assumption that for every control $u \in \mathcal{Q}$ there exists a unique solution y of (1.1) in the space $W(0,T)$, defined as

$$W(0,T) := \left\{ y : y \in L^2(I,V),\ y_t \in L^2(I,V^*) \right\}.$$

The space $W(0,T)$ is continuously embedded in $C(I,H)$, the space of continuous functions from I to H ([36, Ch. XVIII, Theorem 1]). This gives meaning to the trace of $y \in W(0,T)$ in H, i.e., we can work with $y(0), y(T) \in H$. This is of particular interest for transformations in the weak sense, as well as for the error estimation results later in this thesis. In the following we denote the solution space by $Y := W(0,T)$. With the norm

$$\|y\|_Y = \left(\int_I (\|y(t)\|_H^2 + \|y_t(t)\|_{H^*}^2) dt \right)^{\frac{1}{2}}$$

and corresponding inner product, Y becomes a Hilbert space.

With the preliminaries the variational formulation of problem (1.1) sounds as follows: For each $u \in \mathcal{Q}$ and $y_s \in H$ we seek a solution $y \in Y$ such that the weak form

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), v \rangle_H = \langle \mathcal{A}(y(t), u(t)), v \rangle_H, \quad \langle y(0) - y_s, v \rangle_H = 0 \quad \forall v \in V, \tag{1.2}$$

is satisfied. The inner product $\langle y(t), v \rangle_H$ must be understood in the sense of Remark 1.1 and the initial condition $y(0) = y_s$ becomes meaningful due to the continuous embedding of Y in $C(I,H)$. In addition, the embedding allows us to write

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), v \rangle_H \, dt = \langle y_t(t), v \rangle_H \, dt.$$

The temporal dimension is not explicitly considered in a weak sense since we follow a method-of-lines approach and deal with the semi-discrete problems from the ODE perspective.

We are concerned with the following showcase of semilinear problems. The operator $\mathcal{A}(y(t), u(t))$ is throughout assumed to be given in weak form. Therefore, we consider the bilinear form

$$a : V \times V \to \mathbb{R}, \tag{1.3}$$

which contains the spatial differential operators. Thus, the problem formulation is given as

$$\begin{aligned} \langle \mathcal{A}(y(t), u(t)), v \rangle_{V^* \times V} &= a(y(t), v) && (1.4) \\ &+ \left\langle \Theta(y(t), u(t)) + Bu(t) + f^{\mathrm{dat}}(t), v \right\rangle_{V^* \times V} \quad \forall v \in V. && (1.5) \end{aligned}$$

where $\Theta : V \times R \to V^*$ is a sufficiently smooth nonlinear operator, $B : R \to V^*$ is a linear control operator, and an $f^{\mathrm{data}} \in L^2(I, V^*)$. For controls affected by the linear operator B we consider the general case of spatially and/or temporally distributed controls. For control functions that enter nonlinearly in the equation we allow time-dependent or constant control functions, i.e., $u \in L^2(I, \mathbb{R}^{n_u})$ or $u \in \mathbb{R}^{n_u}$. We exclude the case of spatially distibuted control functions in the nonlinearity as with this additional challenges arise when the discrete empirical interpolation method (DEIM) projection step is later applied to the nonlinear part. We comment on the difficulties in §3.2.1. While we exclude this case it can be handled by the methods presented in this thesis if a proper orthogonal decomposition (POD) projection only (without DEIM) is used to obtain the surrogate model.

Moreover, in the application part of this thesis we consider problems that contain systems of PDEs. These are not explicitly dealt with in the theoretical discussions and peculiarities are discussed in the particular application.

Remark 1.2. We impose boundary conditions in a weak sense, hence, they are integrated in the bilinear form and the data part $f^{\mathrm{dat}}(t)$. Thus, the problems are Cauchy problems. The formulation as Cauchy problem allows a simple relation of the weak form in a space V and some subspace $V_0 \subset V$ obtained via projection, which is of particular interest for model order reduction.

Example 1.1. We consider the linear heat equation with Robin boundary conditions and distributed control on the unit square $\Omega = (0,1)^2$

$$\begin{aligned} y_t - \Delta y &= u && \text{in } I \times \Omega, \\ \partial_\nu y + \beta_1 y &= \beta_2 && \text{on } I \times \partial\Omega, \\ y(0) &= y_s && \text{on } \Omega. \end{aligned} \tag{1.6}$$

Here $\partial_\nu y$ denotes the outer normal vector and $\beta_1, \beta_2 \in \mathbb{R}$. Embedding the problem in our notation we assume $V = H^1(\Omega)$, $H = L^2(\Omega)$, $\mathcal{Q} = R = L^2(\Omega)$, and the control operator B is the identity. Taking the integral over the spatial domain, multiplying by a test function $v \in V$, and applying an integration by parts we obtain the weak form (1.2). This yields a linear differential operator given in a weak sense as

$$\langle \mathcal{A}(y(t), u(t)), v \rangle_\Omega = a(y(t), v) + \int_\Omega u(t) v \; dr + \int_{\partial\Omega} f^{\mathrm{dat}}(t) v \; dr,$$

with the bilinear form

$$a(y(t), v) = -\int_\Omega \nabla y(t) \nabla v \; dr - \int_{\partial\Omega} \beta_1 y(t) v \; dr.$$

and data $f^{\mathrm{dat}}(t) = \beta_2$.

1.2. Optimal control problems

We now introduce the guiding optimization problem which serves as basis for the discussions throughout this thesis. It is formulated as optimal control problem in an abstract form

$$\begin{aligned} \min_{y,u} \quad & J(y,u) \\ \text{s.t.} \quad & \frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), v \rangle_H = \langle \mathcal{A}(y(t), u(t)), v \rangle_H \,, \\ & \langle y(0) - y_s, v \rangle_H = 0 \quad \forall v \in V, \qquad u \in \mathcal{Q}_{ad}, \end{aligned} \tag{OCP}$$

with an objective $J : Y \times \mathcal{Q} \to \mathbb{R}$ and control constraints $\mathcal{Q}_{ad} \subset \mathcal{Q}$ such that $\mathcal{Q}_{ad}$ is convex, closed, bounded and non-empty. The PDE constraint is given as defined in (1.1). We do not include constraints on the states y since they would introduce additional difficulties, which are not in the focus of this thesis.

The only formal requirement on the objective regarding the algorithmical aspects in this thesis is that it is sufficiently smooth with respect to its arguments. However, for the sake of simplicity we assume the objective to depend on states $y(T)$ at the final time. This avoids an additional term in the adjoint equation, which improves readability throughout the theoretical discussion. Nevertheless, the extension to a more general case is straightforward. Some cases require to add an additional regularization term of Tychonoff type which smoothens the solution of the problem. Thus, we arrive at an objective of the form

$$J(y,u) = \quad \tilde{J}(y(T)) \; + \; \frac{\gamma}{2} \left\| u - \hat{u} \right\|_{\mathcal{Q}}^2 \;,$$

with a sufficiently smooth $\tilde{J} : V \to \mathbb{R}$ and a regularization factor $\gamma \geq 0$.

Existence and uniqueness of solutions to the abstract problem (OCP) is highly problem dependent and, hence, discussed in dependence on the character of the objective, the dynamic constraint, and particular control constraints. For a more detailed survey on optimal control problems with PDE constraints we refer the reader to Fursikov [45] or Tröltzsch [111]. In literature there are two different conceptual approaches to tackle the optimization problem which are of importance to this thesis:

1. The *reduced approach* where the dynamic state in the objective is expressed in terms of the control and, thus, is eliminated from the optimization problem

2. The *non-reduced approach* where state and control are handled explicitly in the optimization problem

In this thesis we follow the reduced approach which is used in many optimization problems where the dynamic problem is well-posed.

The formulation of the reduced optimal control problem is based on the existence of a unique solution of the dynamic problem (1.1). Under the assumptions of unique solvability of the dynamic problem, for all $u \in \mathcal{Q}_{ad}$, we can define a solution operator given as the mapping $u \mapsto y(u)$. The mapping satisfies the weak formulation (1.2), i.e.,

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle y(t;u), v \rangle_H = \langle \mathcal{A}(y(t;u), u(t)), v \rangle_H , \quad \langle y(0;u) - y_s, v \rangle_H = 0 \quad \forall v \in V,$$

where $y(t;u) \in V$ denotes the evaluation in the codomain of the solution operator at time $t \in I$.

Replacing y by $y(u)$ in the objective of (OCP) we obtain the *reduced optimal control problem*

$$\min_u \quad j(u) := J(y(u), u), \quad \text{s.t.} \quad u \in \mathcal{Q}_{ad}. \tag{ROCP}$$

Doing so, we have eliminated the state from the optimization problem. In the sequel we make the pragmatic assumption that solutions of both problems (OCP) and (ROCP) exist. The same assumption we make for the corresponding problems later in this thesis which we obtain from (OCP) and (ROCP) via discretization.

Example 1.2. In the recent literature on POD and optimal control, e.g. [61, 62, 112], the theoretical investigations are often carried out for problems of linear-quadratic type, e.g., a problem of the form

$$\begin{aligned} \min_{y,u} \quad & \frac{1}{2} \|y(T) - y_\Omega\|_H^2 \; + \; \frac{\gamma}{2} \|u\|_{\mathcal{Q}}^2 \\ \text{s.t.} \quad & \frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), v \rangle_H - a(y(t), v) = \langle Bu(t), v \rangle_H , \\ & \langle y(0) - y_s, v \rangle_H = 0 \quad \forall v \in V, \quad u \in \mathcal{Q}_{ad}. \end{aligned} \tag{1.7}$$

with a bilinear form $a : V \times V \to \mathbb{R}$ which is assumed to be V-elliptic, continuous and symmetric. Further, we have a linear control operator $B : R \to V^*$, $y_\Omega \in H$, and a regularization factor $\gamma \geq 0$. It is well known that for this problem there exists a solution $y \in Y$ which satisfies

$$\|y\|_Y \leq C \left(\|u\|_{\mathcal{Q}} + \|y_s\|_H \right)$$

for some constant C and given $u \in \mathcal{Q}$, $y_s \in H$. Thus, we have a solution operator that is continuous, linear, and bounded. Defining the solution operator via the mapping $u \mapsto y(t;u)$, the reduced problem has then the simple structure

$$\min_u \quad \frac{1}{2} \|y(T;u) - y_\Omega\|_\Omega^2 \; + \; \frac{\gamma}{2} \|u\|_{\mathcal{Q}}^2 \quad \text{s.t.} \quad u \in \mathcal{Q}_{ad},$$

with a convex objective. Using standard arguments it can be shown that (1.7) has a unique global solution $(y^\star, u^\star)$.

1.3. Necessary optimality conditions

In this section we recall the necessary conditions for optimality and present approaches two compute the required derivatives. The presentation is aligned with the textbook of Hinze et al. [60]. Throughout we are interested in *local solutions*, which are to be distinguished from a *global solution* that is more difficult to find. We speak of a local minimum $u^\star$ of (ROCP) when there is a neighborhood $\mathcal{Q}_0 \subset \mathcal{Q}$ of $u^\star$ such that

$$j(u^\star) \leq j(u) \quad \forall u \in \mathcal{Q}_0.$$

Further, we only give first order necessary conditions, as second order sufficient conditions are not the focus of this thesis and we refer to [111] for details.

Definition 1.1 (**Fréchet differentiability**)**.** Let U, Z be Banach spaces. An operator $A : U \to Z$ is called *Fréchet differentiable* on $U_0 \subset U$ if for every $\phi \in U_0$ there is a bounded linear operator $A'(\phi) : U \to Z$ such that

$$\lim_{\|h\|_U \to 0} \frac{\|A(\phi + h) - A(\phi) - A'(\phi)h\|_Z}{\|h\|_U} = 0, \quad h \in U$$

The linear operator $A'(\phi)$ is called the *Fréchet derivative* of A at ϕ.

Typically, we will be interested in derivatives in a given direction $\tilde{\phi} \in \mathcal{Q}$, i.e., we consider directional derivatives $A'(\phi)\tilde{\phi}$. Throughout we use the notation A_ϕ for the derivative with respect to its argument ϕ. Whether we mean the Fréchet derivative or the derivative in $\mathbb{R}^n$ will be clear from the context. Having in mind the general optimization problem (OCP) we sum up the assumptions that we require to state necessary optimality conditions.

Assumption 1.1.

1. $\mathcal{Q}_{ad}$ is non-empty, convex, and closed.
2. The objective $J(y, u)$ and the operator $\mathcal{A}(y(t), u(t))$ are continuously Fréchet differentiable.
3. The state equation (1.1) has a unique solution $y = y(u) \in Y$, for each $u \in \mathcal{Q}_\delta$, $\mathcal{Q}_{ad} \subset \mathcal{Q}_\delta \subset \mathcal{Q}$.
4. The linear operator $\mathcal{A}_y(y(t; u), u(t))$ has a bounded inverse $\forall u \in \mathcal{Q}_\delta$.

Proposition 1.2. *Under Assumption 1.1, the solution operator $y(u)$ corresponding to* (1.2) *is continuously differentiable with respect to u.*

The statement is directly obtained applying the implicit function theorem and noting that the above assumptions are naturally satisfied for the operator $\frac{\mathrm{d}}{\mathrm{d}t}$. Let us now state the conditions to hold in a solution of problem (ROCP).

Theorem 1.3 (**Necessary optimality condition**)**.** *Let Assumption 1.1 hold. If $u^\star$ is a local solution to the reduced optimal control problem* (ROCP) *it follows that $u^\star \in \mathcal{Q}_{ad}$ satisfies the variational inequality*

$$\langle j_u(u^\star), u - u^\star \rangle_{\mathcal{Q}^* \times \mathcal{Q}} \geq 0 \quad \forall u \in \mathcal{Q}_{ad}. \tag{1.8}$$

Proof. Due to the reduced approach the assertion is a result of variational calculus. A rigorous proof can be found, e.g., in [35]. q.e.d.

Inequality (1.8) is a first-order necessary condition for the reduced problem (ROCP). If the objective is convex, then (1.8) is even a sufficient condition and $u^\star$ is a global minimum (see, e.g., Example 1.2). Otherwise, for a general nonlinear objective, the point $u^\star$ is only a local minimum.

Later in the optimization we need to provide the gradient $\nabla j(u)$ of the reduced objective during the optimization algorithm. After discretization, in particular we need to compute derivatives in a set of directions $\tilde{u}_1, \dots, \tilde{u}_{n_{dir}}$, where n_{dir} may be large and $\{\tilde{u}_i\}_{i=1}^{n_{dir}}$ form a basis of some finite dimensional subspace of $\mathcal{Q}$.

In the following we write $\langle j_u(u^\star), u - u^\star\rangle_{\mathcal{Q}}$ instead of the duality pairing, assuming that $j_u(u^\star)$ is given as the Riesz representant. Depending on how the representant is computed, more specific optimality conditions can be stated. Two approaches are distinguished, namely the adjoint and the sensitivity approach. This distinction of approaches is made several times throughout this thesis and we start the discussions with the adjoint approach, as it is the standard in PDE-constrained optimal control problems. The sensitivity approach is of particular interest for parameter estimation problems.

1.3.1. Adjoint approach

We now discuss different ways of representing the derivative of the reduced objective. The discussion is of importance for the investigation of optimality of a solution as well as for the numerical algorithms that are presented in Chapter 2.

An operator based derivation

We introduce the abstract operator $e : Y \times \mathcal{Q} \to Y^*$, defined as

$$e(y, u) := \frac{\mathrm{d}}{\mathrm{d}t} y - \mathcal{A}(y(t), u(t)). \tag{1.9}$$

With this operator the state equation (1.1) can be written as

$$e(y, u) = 0,$$

which allows a compact introduction of the adjoint and the sensitivity approach as well as a clear comparison. With the formulation (1.9) and under Assumption 1.1 the derivative of the solution operator $y_u(u) \in \mathcal{Q}^*$ is given by the identity

$$e_y(y(u), u) y_u(u) + e_u(y(u), u) = 0, \tag{1.10}$$

recalling that $e_y(y(u), u)$ is continuously invertible. In contrast to [60] we derive the adjoint and the sensitivity equations in a Hilbert space setting. Thus, we use the corresponding inner product instead of the duality pairing.

Exploiting the Hilbert space structure of Y, we consider the derivative of $j(u)$ at some point $u \in \mathcal{Q}$ in a direction $\tilde{u} \in \mathcal{Q}$

$$\begin{aligned} \langle j_u(u), \tilde{u}\rangle_{\mathcal{Q}} &= \langle J_y(y(u), u), y_u(u)\tilde{u}\rangle_Y + \langle J_u(y(u), u), \tilde{u}\rangle_{\mathcal{Q}} \\ &= \langle y_u^*(u) J_y(y(u), u), \tilde{u}\rangle_{\mathcal{Q}} + \langle J_u(y(u), u), \tilde{u}\rangle_{\mathcal{Q}} \end{aligned}$$

where $y_u^*(u)$ is the adjoint of the linear operator $y_u(u)$. We observe that the derivative $j_u(u)$ is given by

$$j_u(u) = y_u^*(u) J_y(y(u), u) + J_u(y(u), u).$$

In the adjoint approach we exploit that we can efficiently compute the term $y_u^*(u)J_y(y(u),u)$ in the derivative representation. Solving equation (1.10) for $y_u(u)$ and considering its adjoint operator yields the equality

$$y_u^*(u)J_y(y(u),u) = -e_u^*(y(u),u)\,\underbrace{(e_y^*(y(u),u))^{-1}J_y(y(u),u)}_{=:p}\,. \tag{1.11}$$

From the last equation we can define the *adjoint equation* as

$$e_y^*(y(u),u)p = -J_y(y(u),u) \tag{1.12}$$

with the *adjoint variable* $p \in Y^* = Y$. Analogously to the state we can define a solution operator $p(u)$ which satisfies the adjoint equation. A new representation for (1.11) in terms of the adjoint solution $p(u)$ is then given as

$$y_u^*(u)J_y(y(u),u) = e_u^*(y(u),u)p(u).$$

Using the adjoint variable we find an expression for the derivative of the reduced objective as

$$j_u(u) = e_u^*(y(u),u)p(u) + J_u(y(u),u). \tag{1.13}$$

We end up with the following steps to compute directional derivatives $j_u(u)\tilde{u}$:

1. Solve the state equation (1.1)

2. Solve the adjoint equation (1.12)

3. Evaluate $j_u(u)\tilde{u}$ using equation (1.13)

Note that for each direction $\tilde{u}$ to be computed, we only have to evaluate once the solution operator $y(u)$ and once the solution operator $p(u)$, which makes the approach fairly attractive for optimal control problems.

A Lagrangian based derivation

The adjoint equation is so far given in an abstract form and it is in general not clear how the adjoint operators look like and which particular spaces to choose in the derivation process. Hence, in practice one often takes a formal approach to derive a particular form of the adjoint equation, using a Lagrangian based view and the standard inner product $\int_I \langle\cdot,\cdot\rangle_H$ in the derivation. Based on the weak form (1.2), we introduce the Lagrangian $\mathcal{L} : \mathcal{Q} \times Y \times Y \to \mathbb{R}$ as

$$\mathcal{L}(u,y,p) := J(y,u) - \int_I \langle y_t - \mathcal{A}(y(t),u(t)), p\rangle_H \,dt - \langle y(0) - y_s, p(0)\rangle_H\,, \tag{1.14}$$

where p is considered as Lagrange multiplier. Following the concepts of constrained optimization, one would have to use a separate Lagrange multiplier p_0 for the initial value 'constraint'. However, standard calculus gives $p(0) = p_0$.

Now the adjoint equation can be obtained by differentiation of the Lagrangian with respect to the states, i.e., we determine p such that for all $\tilde{y} \in Y$

$$\begin{aligned} 0 &= \mathcal{L}_y(u,y,p)\tilde{y} \\ &= J_y(y,u)\tilde{y} - \int_I \langle \tilde{y}_t - \mathcal{A}_y(y(t),u(t))\tilde{y}, p\rangle_H \,dt - \langle \tilde{y}(0), p(0)\rangle_H \\ &= J_y(y,u)\tilde{y} - \int_I \left\langle \tilde{y}, -p_t - \mathcal{A}_y^*(y(t),u(t))p\right\rangle_H dt - \langle \tilde{y}(T), p(T)\rangle_H\,. \end{aligned}$$

In analogy to the states for almost all $t \in I$ we consider the adjoint equation in weak form as

$$\left\langle -p_t(t) - \mathcal{A}_y^*(y(t), u(t))p(t), v \right\rangle_H = 0, \quad \langle p(T), v\rangle_H = \langle J_y(y,u), v\rangle_H \quad \forall v \in V, \tag{1.15}$$

noting that we assume the objective to depend only on states at final time T. With the solution operator $y(u)$ inserted in the Lagrangian (1.14) we find the identity

$$j(u) = \mathcal{L}(u, y(u), p),$$

which holds for every $p \in Y$. Hence, derivatives of the reduced objective in a direction $\tilde{u} \in \mathcal{Q}$ are given as

$$j_u(u)\tilde{u} = \mathcal{L}_y(u, y(u), p)y_u(u)\tilde{u} + \mathcal{L}_u(u, y(u), p)\tilde{u}.$$

With a $p(u)$ that solves the adjoint equation (1.15) we have

$$j_u(u)\tilde{u} = \mathcal{L}_u(u, y(u), p(u))\tilde{u} = J_u(y,u)\tilde{u} - \int_I \langle \mathcal{A}_u(y(t), u(t))\tilde{u}, p(u)\rangle_H \, dt.$$

We obtain the following result for the necessary optimality conditions in the adjoint approach expressed via the Lagrangian.

Theorem 1.4. *Let Assumption 1.1 hold. If $(y^\star, u^\star)$ is a solution of* (OCP) *then there exists a Lagrange multiplier $p^\star \in Y$ such that $u^\star \in \mathcal{Q}_{ad}$ and the following optimality system is satisfied*

$$\begin{aligned} \mathcal{L}_p(u^\star, y^\star, p^*)\phi &= 0 \quad \forall \phi \in Y && (\textit{state equation}),\\ \mathcal{L}_y(u^\star, y^\star, p^*)\phi &= 0 \quad \forall \phi \in Y && (\textit{adjoint equation}),\\ \mathcal{L}_u(u^\star, y^\star, p^*)(u - u^\star) &\geq 0 \quad \forall u \in \mathcal{Q}_{ad} && (\textit{gradient equation}). \end{aligned} \tag{1.16}$$

Proof. The result is direct consequence of Theorem 1.3. q.e.d.

The Lagrangian perspective allows for a compact formulation of the optimality conditions which are often the starting point for a numerical solution of the underlying problem. A simultaneous solution of the structured system (1.16) is one way to solve (1.1), which one would classify as non-reduced approach.

Example 1.3. The abstract adjoint operators can be illustrated by means of the following example. We consider the optimal control problem (1.7) of Example 1.2. With our practical approach via the Lagrangian we get

$$\begin{aligned} \mathcal{L}(u,y,p) = {} & \frac{1}{2}\,\|y(T) - y_\Omega\|_H^2 + \frac{\gamma}{2}\,\|u\|_{\mathcal{Q}}^2 \\ & - \int_I \big(\langle y_t, p\rangle_H - a(y(t), p(t)) - \langle Bu(t), p\rangle_H \big) dt - \langle y(0) - y_s, p(0)\rangle_H \,. \end{aligned}$$

Choosing $H = L^2(\Omega)$, $V = H^1(\Omega)$ and assuming $a(\cdot,\cdot)$ to represent the diffusion operator according to Example 1.1, we get

$$\begin{aligned} \mathcal{L}_y(u,y,p)\phi = {} & \langle y(T) - y_\Omega, \phi\rangle_\Omega \\ & - \int_I \big(\langle \phi_t, p\rangle_\Omega + \langle \nabla\phi, \nabla p\rangle_\Omega + \langle \beta_1 \phi, p\rangle_{\partial\Omega} \big) dt - \langle \phi(0), p(0)\rangle_\Omega \,. \end{aligned}$$

With $\phi \in Y$ we can carry out an integration by parts in time to arrive at the adjoint problem in weak form

$$\langle -p_t, v\rangle_\Omega = \langle \nabla p, \nabla v\rangle_\Omega + \langle \beta_1 p, \phi\rangle_{\partial\Omega} \quad \forall v \in V, \quad p(T) = y(T) - y_\Omega.$$

Due to the symmetry of the diffusion operator we find that $a(y(t),v) = a(p(t),v)^*$, however, $\mathcal{A}(y(t),v) \neq \mathcal{A}^*(p(t),v)$ due to the different boundary conditions of state and adjoint problem. Finally, the variational inequality sounds

$$\langle \gamma u^\star + B^* p, u - u^\star \rangle_{\mathcal{Q}} \geq 0 \quad \forall u \in \mathcal{Q}_{ad}, \tag{1.17}$$

where, $B^* : V^* \to \mathcal{Q}^* \cong \mathcal{Q}$ is the adjoint of the control operator, which, e.g., in case of B being the identity mapping is also the identity.

1.3.2. Sensitivity approach

A second way to represent the directional derivative $j_u(u)\tilde{u}$ is by the sensitivity approach, which can be considered as a rather straightforward way to compute derivatives. As in the operator based derivation of the adjoint we look at the directional derivative

$$j_u(u)\tilde{u} = J_y(y(u),u)y_u(u)\tilde{u} + J_u(y(u),u)\tilde{u}$$

with $u, \tilde{u} \in \mathcal{Q}$. Now, instead of introducing an adjoint variable one can also directly compute the occurring sensitivity $y_u(u)\tilde{u} \in Y$ for a given direction $\tilde{u}$. Differentiation of the abstract state equation $e(y,u) = 0$ in (1.9) with respect to u in a direction $\tilde{u}$ yields the *sensitivity equation*

$$e_y(y(u),u)\tilde{y} + e_u(y(u),u)\tilde{u} = 0, \tag{1.18}$$

in the variable $\tilde{y} = y_u(u)\tilde{u}$. With a solution $\tilde{y}(u) \in Y$ of (1.18) we get

$$j_u(u)\tilde{u} = J_y(y,u)\tilde{y}(T) + J_u(y,u)\tilde{u}. \tag{1.19}$$

We arrive at the following procedure to compute $j_u(u)\tilde{u}$ via the sensitivity approach:

1. Solve the state equation (1.1)
2. Solve the sensitivity equation (1.18) for each direction $\tilde{u} \in \mathcal{Q}$
3. Evaluate $j_u(u)\tilde{u}$ using equation (1.19)

Note that in contrast to the adjoint approach, the sensitivity equation has to be solved for each direction $\tilde{u}$ that is required. On the other hand, with $y_u(u)\tilde{u}$ given, the derivative of any objective of interest J can be cheaply evaluated. This makes the sensitivity approach appealing, when the number of variables n_u is rather small and there are multiple quantities of interest in the optimization. The issue will be revisited in §2.5.1 when discussing automatic differentiation (AD).

We owe to give an explicit equation for the sensitivity variable $\tilde{y}$ related to our reference state problem in weak form (1.2). Differentiation with respect to $\tilde{u}$ yields the problem

$$\begin{gathered} \frac{\mathrm{d}}{\mathrm{d}t} \langle \tilde{y}(t), v \rangle_H = -\langle \mathcal{A}_y(y(t),u(t))\tilde{y}(t), v \rangle_H - \langle \mathcal{A}_u(y(t),u(t))\tilde{u}(t), v \rangle_H, \\ \langle \tilde{y}(0), v \rangle = 0 \quad \forall v \in V, \end{gathered} \tag{1.20}$$

in the variable $\tilde{y}$. Problem (1.20) is often also called the tangent equation. As in the adjoint case the derivation of the weak form must be understood as formal. A precise justification of the differentiation procedure can be found in, e.g., [15]. Note that the sensitivity equation and the adjoint equation are linear in the sensitivity and adjoint variable respectively, which we exploit in the numerical solution process in §2.5.2.

For completeness, in analogy to Theorem 1.4 we state the infinite-dimensional optimality conditions resulting from the sensitivity approach, which follow immediately from formula

(1.19). Under Assumption 1.1, if $(y^\star, u^\star)$ is a local solution of (OCP), then $u^\star \in \mathcal{Q}_{ad}$, $y^\star$ solves the state equation (1.2), and

$$\langle J_y(y^\star, u^\star), \tilde{y}(T)\rangle_H + \langle J_u(y^\star, u^\star), u - u^\star\rangle_{\mathcal{Q}} \geq 0 \quad \forall u \in \mathcal{Q}_{ad}, \tag{1.21}$$

where $\tilde{y} = y_u(u - u^\star)$ solves (1.20) for each direction $(u - u^\star) \in \mathcal{Q}$.

1.4. Parameter estimation

In this section we introduce the parameter estimation problem and show in which way it can be considered as a special case of optimal control problems. This embedding allows for a simultaneous treatment of the two problems throughout most of this thesis. However, a different optimization method is applied to solve the parameter estimation problem where a distinction of the two problems is required. For parameter estimation problems we apply a Gauss–Newton method which has beneficial numerical and statistical properties. From an applications point of view the motivation of the two problems is entirely different.

Describing the parameter estimation problem in words, we aim to solve the following problem: Given a dynamic model, we try to find the model parameters, such that a considered real world process that is supposed to be described by the model, is approximated sufficiently well according to a criterion of choice. The data to compare to is typically obtained from experiments, thus, contains measurement errors with a certain statistical variance.

More precisely, we seek to identify a vector of variables $u \in \mathbb{R}^{n_u}$, which in this context we refer to as parameters, by solving the least-squares problem

$$\begin{aligned} \min_{y,u} \quad & \frac{1}{2}\sum_{i=1}^{n_{\text{meas}}} \left(\frac{\eta_i - h(y(\tilde{t}_i))}{\varsigma_i}\right)^2 \\ \text{s.t.} \quad & \frac{\mathrm{d}}{\mathrm{d}t}\langle y(t), v\rangle_H = \langle \mathcal{A}(y(t), u(t)), v\rangle_H, \quad \langle y(0) - y_s, v\rangle_H = 0 \quad \forall v \in V. \end{aligned} \tag{1.22}$$

We introduce a measurement data vector $\eta \in \mathbb{R}^{n_{\text{meas}}}$ where each measurement η_i is taken at some specified time instances $\tilde{t}_i$. Moreover, the objective comprises the standard deviation ς_i of a measurement error ϵ_i and the measurement function $h : H \to \mathbb{R}$, which we assume to be Fréchet differentiable. For simplicity we assume h to be a scalar function, however, the results can easily be extended to a vector valued h. We further require $n_{\text{meas}} \geq n_u$ which is necessary for the parameters to be identifiable.

For the sake of simplicity we consider the unconstrained problem and refer to [21] for constrained parameter estimation problems. Let u^{true} be the 'true' parameter vector determined by the laws of nature, then we assume the relations

$$\begin{aligned} \eta_i \quad & = h(y(\tilde{t}_i; u^{\text{true}})) + \epsilon_i, \\ \epsilon_i \quad & \sim \mathcal{N}(0, \varsigma_i^2), \quad i = 1, \ldots, n_{\text{meas}}, \end{aligned}$$

i.e., we assume the measurement errors to be normally distributed around 0 with standard deviation ς_i. As for the optimal control case we use again a solution operator which is given as the mapping $u \mapsto y(u)$ and denote with $y(t; u) \in V$ the evaluation of the codomain of the solution operator at time $t \in I$. This yields the *reduced parameter estimation problem*

$$\min_{u \in \mathbb{R}^{n_u}} \quad \frac{1}{2}\sum_{i=1}^{n_{\text{meas}}} \left(\frac{\eta_i - h(y(\tilde{t}_i; u))}{\varsigma_i}\right)^2 \quad =: \frac{1}{2}\left\|\mathcal{F}(u)\right\|^2, \tag{1.23}$$

where we introduced the function $\mathcal{F} : \mathbb{R}^{n_u} \to \mathbb{R}^{n_{\text{meas}}}$, defined as

$$\mathcal{F}_i := \frac{\eta_i - h(y(\tilde{t}_i; u))}{\varsigma_i}, \quad i = 1, \ldots, n_{\text{meas}}.$$

With this reformulation, problem (1.23) formally becomes a standard nonlinear least-squares problem. Considering the measurement data as a given data vector and assuming the control function space $\mathcal{Q}$ to be the space of constant functions from I to $\mathbb{R}^{n_u}$, we can set

$$j(u) := \frac{1}{2} \sum_{i=1}^{n_{\text{meas}}} \left(\frac{\eta_i - h(y(\tilde{t}_i; u))}{\varsigma_i} \right)^2 .$$

The least-squares problem is, thus, embedded in the optimal control context.

2. Numerical Strategies

The aim of the present chapter is to provide the necessary numerical methods to solve the optimal control and the parameter estimation problem with a high-dimensional problem discretization. In Part II we will complement these methods by applying model reduction to the underlying dynamic systems and, thus, obtain a significant speed up of the solution process.

Conceptually, we transform the considered optimization problems into a nonlinear programming problem (NLP) and use derivative-based algorithms for its solution. We pursue a method-of-lines approach, meaning that we first employ the spatial discretization to transform the PDE into an ODE and then deal with the time discretization. In analogy to §1.3 we eliminate the discretized states from the formulation of the optimization task and consider the reduced problem. A key aspect is the efficient and accurate computation of the gradient of the discrete reduced objective. Therefore, we make use of techniques related to automatic differentiation (AD) which we discuss at the end of the chapter. Moreover, we re-consider the optimization problem from the semi-discrete perspective since big parts of the model reduction in Part II discussion are carried out in the semi-discrete setting.

In the literature several dichotomies can be found regarding numerical strategies for optimal control problems. We already distinguished the reduced and the non-reduced approach in §1.2, which is also of particular interest from the numerical perspective. A non-reduced approach in this context can also be referred to as *all-at-once approach*. Its characteristic element is that the underlying dynamic model problem and the optimization problem are solved at the same time. Thus, during the iterations in the optimization the variables might be infeasible and the underlying differential equations must be satisfied in the solution. The reduced approach from the optimization perspective can also be called a *sequential approach* since one first solves the dynamic system in each iteration and then optimizes the remaining control variables which are contained in the null space of the non-reduced problem. As before in the theoretical considerations we follow the reduced-approach.

A second dichotomy we already mentioned is the distinction of adjoint and sensitivity approach for derivative generation. One finds the alternative naming *reverse mode* and *forward mode*, typically used in the field of AD (see §2.5.1).

A further distinction that is important to our results, is whether we first discretize a considered problem and then optimize it or vice versa. A comparison of the two strategies regarding the use of POD is one of the major contributions of this thesis, therefore, we recall the basic concepts as well as advantages and disadvantages of either strategy. Having both strategies in mind the problem is first discretized in the methods proposed by us. This is reflected in the title of this thesis by the keyword 'direct' which is often used as synonym for methods where one first takes care about discretization. We comment on the naming conventions in Remark 2.1.

2.1. Discretize-then-optimize vs. optimize-then-discretize

One can distinguish two main strategies to tackle optimization problems. We refer to these as discretize-then-optimize (DTO) and optimize-then-discretize (OTD). In the DTO approach one first discretizes all infinite-dimensional parts of the problem formulation. For the optimal control example problem (OCP) this means that we carry out a discretization of the constraints $e(y,u)=0$ in the abstract form as in §1.3.1 and replace the variables y and u by finite dimensional approximations. Thus, we arrive at

$$\min_{y^{h\tau},u^{h\tau}} \quad J(y^{h\tau},u^{h\tau}) \qquad \text{s.t.} \quad e^{h\tau}(y^{h\tau},u^{h\tau})=0, \quad u^{h\tau}\in \mathcal{Q}_{ad}. \tag{2.1}$$

We use a double index $h\tau$ to make clear that space and time are handled in separate discretization steps. Problem (2.1) is now finite dimensional and has the form of a standard NLP, which can be solved using a suitable optimization method (see §2.4). In contrast, with the OTD approach one first derives infinite-dimensional optimality conditions and subsequently applies a suitable discretization. Starting again from problem (OCP) and assuming an adjoint approach, OTD means we derive the infinite-dimensional optimality system (1.16) and then apply discrete strategies to solve the coupled system.

The two possible ways are illustrated in Figure 2.1. In general it is desirable for a method to exhibit advantages of both strategies. If proper techniques are chosen both paths in the diagram yield the same result or, in short, the diagram is commutative. In this thesis we aim to establish this commutativity for our methods in the context of reduced-order modeling for optimization.

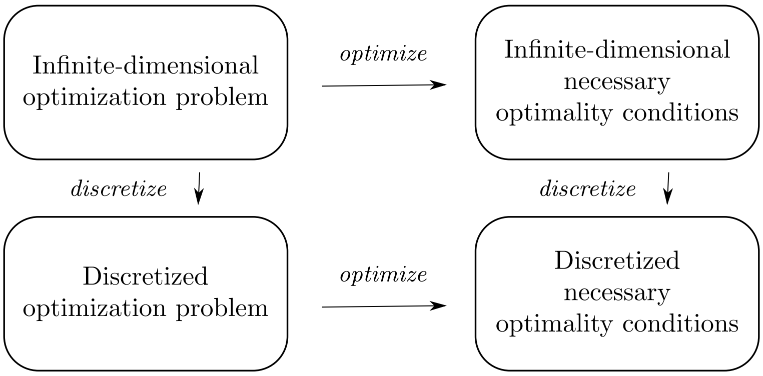
Infinite-dimensional optimization problem
optimize
Infinite-dimensional necessary optimality conditions
discretize
discretize
Discretized optimization problem
optimize
Discretized necessary optimality conditions

Figure 2.1.: Conceptual strategies to determine discretized necessary optimality conditions. We distinguish discretize-then-optimize (lower-left path) and optimize-then-discretize (upper-right path). Commutativity of the diagram is a desirable property.

A comprehensive study and comparison of these concepts is beyond the scope of this thesis, and we refer to [60] for a reflection of DTO vs. OTD in the context of optimization with PDE constraints. For ODEs one can find a discussion of the issue in [100].

There seems to be no general recipe on which approach should be chosen for a particular optimization problem. Following the OTD strategy we start from infinite-dimensional conditions for optimality. This allows to choose suitable discretizations for the occurring equations. That is, we can choose the spaces for the adjoint and the state variables properly, depending on the regularity requirements that a particular application exhibits. Also an efficient refinement of the meshes for the state and adjoint discretization is independently possible. This permits to determine highly accurate solutions and the possibility of an efficient control of the discretization error. Moreover, tackling the problem on the infinite-dimensional level allows for a deeper insight into the problem structure. On the other hand, the requirement to have a thorough understanding of the problem might often not be fulfilled for non-experts in practice. The derivation of the adjoint problem can be challenging but is the key to obtain the infinite-dimensional optimality system. Regarding convergence of an OTD type method, it is possible that the gradient computed from adjoint information is no more consistent with the objective after discretization. This drawback might slow down a derivative-based optimization algorithm or even cause a breakdown of the algorithm. However, due to the possibility of an independent error control of states and adjoints this problem can often be mitigated.

Using suitable methods for derivative computation the DTO strategy overcomes the latter deficit of OTD, since here we can guarantee consistency of objective and gradient. Moreover, in general we do not need to derive the adjoint problem as derivatives are computed on the discrete level. If not handled thoroughly, a possible disadvantage of the DTO approach is that the solution of the discretized optimization problem is not a good approximation of a solution of the continuous problem. In this case the two displayed routes in Figure 2.1 yield different discrete optimality conditions. Though, as we desire consistency between discrete and infinite-dimensional problems, in §2.2 and 2.5.2 we will employ techniques such that commutativity of the diagram is guaranteed. For us this means that we follow a DTO approach which can also be interpreted as an OTD approach. In Part II we will also carry over the commutativity property to the case where POD model reduction is involved in the discretization procedure.

In this thesis we do not keep the states explicitly in the optimization problem. Instead, analogous to the continuous optimal control scenario in §1.3, we follow a reduced approach. That is, we transform (2.1) into

$$\min_{u^{h\tau}} \quad j(u^{h\tau}) \qquad \text{s.t.} \quad u^{h\tau} \in \mathcal{Q}_{ad}, \tag{2.2}$$

where $j(u^{h\tau}) = J(y^{h\tau}(u^{h\tau}), u^{h\tau})$. The existence of the discrete solution operator $y^{h\tau}(u^{h\tau})$ is inherited from the assumption for the infinite-dimensional case, given a suitable discretization method is applied. In the following remark we comment on the difficulties regarding the terminology of DTO and OTD and the synonymous use of the terms direct and indirect approach. We make clear our understanding of the terms and point out the existing ambiguities.

Remark 2.1. Often the notion *direct method* is used in equivalence to DTO while *indirect method* is used synonymously with OTD. At the time of developing this thesis there is an ambiguity regarding these classifications. The expression 'indirect method' is historically associated with the calculus of variations, Pontryagins's maximum principle and the derivation of the adjoint problem, which is similar to the derivation of the optimality system as described in §1.3.1. Here the control is typically eliminated from the optimization problem, hence, only affects the solution algorithms in an indirect way. A 'direct method' is historically characterized by the problem transformation to an NLP and with this a direct iteration on the discretized optimal control variables (see, e.g., [117]). It is now ambiguous whether a method that derives the optimality system making use of adjoint information, but still iterates directly on the control variables in the optimization procedure, should be considered as direct or indirect. On the other hand, when one makes use of adjoint information on the discretized level this might also be considered as an indirect method in the community. While the keyword 'direct' in this thesis title unambiguously describes the method developed in this thesis, we refer to DTO vs. OTD when comparing solution methods using reduced-order models (instead of direct vs. indirect).

We choose a direct approach as we believe that for an employment of numerical optimization in industrial practice, it is beneficial to facilitate the access to a method as far as possible. With the techniques explained, relatively few knowledge on the optimization algorithm is necessary to solve a wide class of practical relevant problems. We describe our particular direct strategy in the next sections.

2.2. Problem discretization

The aim of this section is to lay out the techniques applied in order to obtain the reduced NLP (2.2), in particular the discretization methods that stand behind the solution operator $y^{h\tau}(u^{h\tau})$. We employ a method of lines, that is, we do the spatial discretization first and the integration in time subsequently. The semi-discrete problems (spatially discretized) are of crucial importance to us as it is the starting point for many investigations of model reduction presented in Chapter 3.

2.2.1. Space discretization

We start the discretization of problem (OCP) converting the infinite-dimensional spaces V and R into finite dimensional spaces V^h and R^h. In general, it is essential that we apply a suitable method that fits the requirement of a particular application. Finite Difference Methods are easy to handle, thus, often used by practitioners. Other applications require more sophisticated discretizations, e.g., for convection dominated problems Finite Volume [82] or (Nodal) Discontinuous Galerkin Methods [58] are usually chosen. These discretization methods as well as the frequently used finite element method (FEM) [24] use basis functions with local support, which allow to compute solutions on complicated geometries. On the other hand, on simple geometries one can exploit the superior convergence properties of spectral methods [58] where global basis functions are used. With regard to the applicability of POD model reduction techniques, it is beneficial that the method is or at least can be interpreted as Galerkin method. The weak form (1.2) is the natural starting point for the Galerkin discretization. We now describe briefly the basic steps that lead to the semi-discrete problem.

We discretize the Hilbert space V choosing a finite dimensional conform subspace $V^h \subset V$ of dimension N and assume $\varphi_1(r), \ldots, \varphi_N(r)$ to be a basis of V^h. Recall the most general case of $\mathcal{Q} = L^2(I, R)$ for the control function space where $R = L^2(\Omega)^{n_u}$. Hence, to obtain the semi-discrete problem formulation we also need to discretize the space R. For the sake of simplicity we assume during the discussion of the discretization that $n_u = 1$ and choose a subspace $R^h \subseteq V^h$ to approximate R. Thus, we consider a semi-discrete approximation $u^h \in L^2(I, R^h)$ of u.

With these preliminaries the Galerkin method is described as follows. For each $u^h \in L^2(I, R^h)$ and $y_s \in H$ we seek a solution $y^h(t) \in V^h,\ t \in I$ such that the system

$$\frac{\mathrm{d}}{\mathrm{d}t} \left\langle y^h(t), \varphi_j \right\rangle_H = \left\langle \mathcal{A}(y^h(t), u^h(t)), \varphi_j \right\rangle_H, \quad \left\langle y^h(0) - y_s, \varphi_j \right\rangle_H = 0, \quad j = 1, \ldots, N. \tag{2.3}$$

is satisfied. Expanding the function y^h in the space V^h we get

$$y^h(t, r) = \sum_{i=1}^{N} x_i(t) \varphi_i(r), \quad (t, r) \in I \times \Omega, \tag{2.4}$$

where $x : I \to \mathbb{R}^N$ is a vector valued time-dependent function. For control function approximations $u^h \in L^2(I, R^h)$ we consider the expansion

$$u^h(t, r) = \sum_{i=1}^{n_q} q_i(t) \varphi_i^q(r), \quad (t, r) \in I \times \Omega,$$

with $\varphi_1^q(r), \ldots, \varphi_{n_q}^q(r)$ being linear combinations of $\varphi_1(r), \ldots, \varphi_N(r)$ and a basis of R^h. We use the notation q for semi-discrete control variables, for which we have in general $q \in \mathcal{Q}^h$ with $\mathcal{Q}^h := L^2(I, \mathbb{R}^{n_q})$.

An evaluation of (2.3) yields a system of ODEs of the form

$$M\dot{x}(t) = f(t, x(t), q(t)), \quad x(0) = x_s, \ t \in I, \tag{2.5}$$

with the ODE state variable $x(t) \in \mathbb{R}^N$, a constant, symmetric, and positive-definite mass matrix $M \in \mathbb{R}^{N\times N}$ defined via

$$M_{ij} := \langle \varphi_i, \varphi_j \rangle_H, \quad 1 \leq i, j \leq N, \tag{2.6}$$

and the control $q \in \mathcal{Q}^h$. The right-hand side f and the initial value x_s are, thus, defined as

$$f(t, x(t), q(t)) := \left\langle \mathcal{A}(y^h(t), u^h(t)), \varphi \right\rangle_H, \quad x_s := M^{-1} \langle y_s, \varphi \rangle_H,$$

with $\varphi = (\varphi_1, \ldots, \varphi_N)^T$. To problem (2.5) in the following we will also refer as initial value problem (IVP). The dynamic system (2.5) has the general form of an ODE, however, originating from a Galerkin discretization it exhibits certain structure. Namely, it can essentially be separated into a linear and a nonlinear part. When we dedicate ourselves to model reduction we assume the ODE system to be of the form

$$M\dot{x}(t) = Sx(t) + F(x(t), q(t)) + C(q(t)), \quad x(0) = x_s, \ t \in I, \tag{2.7}$$

which we refer to as the *high-fidelity* (HiFi) model. The naming indicates that we assume the Galerkin discretization to be able to capture all of the essential dynamics of the model equations. In the linear part we have the constant matrix $S \in \mathbb{R}^N \times \mathbb{R}^N$ and in the nonlinear part we have $F : \mathbb{R}^N \times \mathbb{R}^{n_q} \to \mathbb{R}^N$ which results from the discretization of the nonlinearity $\Theta(y, u)$ in the PDE problem formulation (1.4). The third term $C(q)$ is an affine mapping that contains the discretization of linear controls and additional constant terms arising from, e.g., the discretization of the boundaries.

Remark 2.2. For the sake of simplicity we neglect the affine part $C(q)$ for the discussions in the semi-discrete setting. The model structure essential for this thesis is reflected by the linear and the nonlinear part. In the presented methods for model reduction $C(q)$ can be handled analogous to the linear part S.

So far we have not commented on the particular form of the space V^h and the basis $\varphi_1, \ldots, \varphi_N$. Assuming FEM as particular Galerkin discretization, the dimension of V^h can be expected to be large for relevant applications. However, conceptionally the subspace V^h is arbitrary, a fact that we exploit for model reduction, where a small solution space is constructed that is tailored for the particular application at hand. The following example illustrates the discretization steps resulting in a HiFi model of the form (2.7).

Example 2.1. We consider again the linear heat equation of Example 1.1 and apply a FEM discretization with linear basis functions. Let Ω^h be a triangulation of Ω and assume $\nu_i, \ i = 1, \ldots, n_\Omega$, to be the vertices. Note that as we impose boundary conditions weakly (see Remark 1.2) the number of degrees of freedom N after discretization is the same as n_Ω. Let the basis functions φ_j be defined by the requirements that for all $1 \leq i, j \leq n_\Omega$

$$\varphi_j(\nu_i) = \delta_{ij} \quad \text{and } \varphi_j \text{ is linear on each element in } \Omega^h.$$

The Galerkin condition now reads as

$$\int_\Omega \frac{\mathrm{d}}{\mathrm{d}t} y^h \varphi_j dr = a(y^h, \varphi_j) + \int_\Omega u^h \varphi_j \ dr + \int_{\partial\Omega} \beta_2 \varphi_j \ dr,$$
$$\int_\Omega (y^h(0) - y_s)\varphi_j \ dr = 0, \quad j = 1, \ldots, N.$$

For simplicity we assume $u^h \in R^h = V^h$. With the approximation y^h as in (2.4) and the definition of the bilinear form in Example 1.1, the parts of the right-hand side of the HiFi model are defined as

$$\begin{aligned} S_{j\cdot}x(t) &= \sum_{i=1}^{N} x_i(t) \left(-\int_\Omega \nabla\varphi_i \nabla\varphi_j \; dr - \int_{\partial\Omega} \beta_1 \varphi_i \varphi_j \; dr \right), \\ C_j(q) &= \sum_{i=1}^{N} q_i \int_\Omega \varphi_i \varphi_j \; dr + \int_{\partial\Omega} \beta_2 \varphi_j \; dr, \quad j = 1, \dots, N, \end{aligned}$$

where $C_j(q)$ is the j-th component of the affine mapping C. As the considered heat equation is linear, obviously, there is no contribution F. We will comment on the particular structure of F when dealing with the Discrete Empirical Interpolation Method in §3.2. Note that we carried out the discretization on the unit square making the evaluation of the integrals simple. For more general domains one would consider a transformation of each element to a reference element where the evaluations of the matrices are carried out (see [24] for details).

2.2.2. Time discretization

With our chosen approach, the last discretization step is the time integration of the semi-discrete problem derived in the prior section. Again we start by discretizing the controls $q \in \mathcal{Q}^h$ which are still given continuously in time. Then we discuss the time discretization scheme. We restrict ourselves to lay out the main ideas of the implicit Euler scheme which we use for the theoretical discussion of the discrete concepts. With the help of this scheme we illustrate the techniques for derivative computation via internal numerical differentiation in §2.5.2 and give error estimates in the model reduction context in Part II. Moreover, we recall some stability and consistency results as we need to pay attention to these aspects when computing derivatives.

To discretize $q \in L^2(I, \mathbb{R}^{n_q})$, we start by dividing the time domain I into $n_{\hat{q}}$ subintervals using

$$0 = t_0 < \cdots < t_{n_{\hat{q}}} = T.$$

On the time grid we replace the infinite-dimensional $q(t)$ by a finite dimensional function $q^\tau(t)$. To keep notation simple, we restrict ourselves to piecewise constant functions on the above time grid in this thesis and define $q^\tau(t)$ as

$$q^\tau(t) = \hat{q}^i \quad \text{for } t \in [t_{i-1}, t_i], \quad i = 1, \dots, t_{n_{\hat{q}}},$$

with each vector $\hat{q}^i \in \mathbb{R}^{n_q}$ characterizing the discretization on a subinterval. Finally we unite all vectors $\hat{q}^i$ in a single vector $\hat{q} \in \mathbb{R}^{n_{\hat{q}} \cdot n_q}$.

Remark 2.3. Via the control discretization in time we explicitly introduce discontinuities in the right-hand side $f(t, x, q)$ of the ODE (2.5). In general, discontinuities may appear already in the problem formulation, e.g., due to $q \in L^2(I, \mathbb{R}^{n_q})$ or due to discontinuous data $f^{\mathrm{dat}}(t)$. For the discussion of the time integration, however, we make the assumption that the right-hand side f is Lipschitz-continuous and differentiable with respect to x. The general case with discontinuities can then be retrieved by considering a piecewise integration of the problem where on each subinterval the assumption on Lipschitz-continuity and differentiability are satisfied.

Under the assumption on Lipschitz-continuity of f with respect to x, we obtain local existence and uniqueness of solutions of (2.5) from the theorem of Picard–Lindelöf. If a solution $x(t)$ on the whole time domain I exists then we have $x \in C^1(I, \mathbb{R}^N)$.

For our numerical computations we use the software tool DAESOL-II by Jan Albersmeyer [2] in which a BDF method is implemented and which is also capable of providing derivatives of arbitrary order for dynamic systems. The method uses a variety of numerical strategies such as adaptive step size, order control, and monitoring of the iteration matrix, to allow an efficient integration of IVPs. We will discuss relevant issues on basis of the implicit Euler scheme which is equivalent to a BDF scheme with order 1. A detailed description of the tool DAESOL-II and the implemented strategies can be found in [2, 14].

We consider the implicit Euler scheme with n_τ steps on the time domain I. We choose the step sizes adaptively and use the parameter τ to denote the time discretization. Consider a sequence of step sizes $\tau_n > 0$ that generate a time grid $t_0, \dots, t_{n_\tau}$ such that $t_0 = 0$, $t_n = t_{n-1} + \tau_n$, $n = 1, \dots, n_\tau$, and $t_{n_\tau} = T$. The iterates $x^n := x(t_{n-1})$ are determined as

$$F^{\mathrm{IE}}(x^{n+1}) := \quad M\frac{x^{n+1} - x^n}{\tau_n} - f(t_n, x^{n+1}, q) = 0, \quad n = 1, \dots, n_\tau, \tag{2.8}$$

where the first iterate is given as $x^1 = x(t_0) = x_s$. The indexing for the states is started at 1 to be consistent with the model reduction notation. As f is assumed to be nonlinear, in each step the nonlinear equations (2.8) must be solved. We apply a Newton-type method (see §2.4) with a fixed maximum number of iterations, which is set to three in DAESOL-II. Each iterate x^{n+1} is determined by $j \leq 3$ recursions of

$$x^{n+1,j+1} = x^{n+1,j} - \mathcal{M}F^{\mathrm{IE}}(x^{n+1,j}). \tag{2.9}$$

For simplicity we assume that the first Newton iterate is $x^{n+1,1} = x^n$. The iteration matrix $\mathcal{M}$ is an approximation of the inverse of the Jacobian of F^{IE}, that is

$$\mathcal{M} \approx \left(\frac{\mathrm{d}F^{\mathrm{IE}}(x^{n+1})}{\mathrm{d}x^{n+1}}\right)^{-1}.$$

The evaluation and decomposition of $\mathcal{M}$ is in general the most time consuming part of the integration. Hence, the monitor strategy in DAESOL-II keeps the iteration matrix fixed for several time steps and decides via a suitable error control when to rebuild $\mathcal{M}$.

Consistency, stability, and stiffness

We now address the issues of consistency, stability, and stiffness of the implicit Euler scheme. However, we restrain from going beyond a qualitative analysis. For a comprehensive study see [38, 53]. The analysis of consistency is carried out by insertion of the true solution $x(t)$ into the particular discretization scheme. We define the local truncation error of the implicit Euler scheme as

$$\sigma(t,\tau) := M\frac{x(t+\tau) - x(t)}{\tau} - f(t+\tau, x(t+\tau), q), \quad t \in I, \tau > 0.$$

A discretization scheme is called consistent of order p if for

$$\bar{\sigma}(\tau) := \max_{t \in I}\ \sigma(t,\tau)$$

$$\text{we have} \qquad \lim_{\tau \to 0} \bar{\sigma}(\tau) = 0 \quad \text{and } \bar{\sigma}(\tau) = \mathcal{O}(\tau^p).$$

Obviously, the implicit Euler scheme is of order $p = 1$.

For the analysis of stability of the integration scheme, first, we need to give a characterization of stability of the underlying ODE. For ODEs the question of stability is often traced back to the investigation of the eigenvalues of the Jacobian of the right-hand side. In this way, an IVP is considered to be stable if

$$\max_{\lambda\in\sigma(A)} \operatorname{Re}(\lambda) < 0 \quad \text{with } A = \frac{\mathrm{d}f(t,x,q)}{\mathrm{d}x}, \tag{2.10}$$

where $\sigma(A)$ denotes the spectrum of A. For a linear IVP the stability condition (2.10) can easily be checked due to A being constant. For a nonlinear problem one way to characterize stability, is to require the condition (2.10) to be satisfied in a fix point x^f, i.e., where $f(t,x^f,q) = 0$. In general stability of a solution is characterized by the fact that small perturbations in the initial data produce small errors in the solution.

With this, a discretization scheme is called stable if it carries over the stability property from the IVP to the produced sequence $x^1,\dots,x^{n_\tau}$. The stability requirement for a mapping $\Xi^n : x^n \to x^{n+1}$ that describes an iteration step is given as

$$\rho\left(\frac{\mathrm{d}\Xi^n(x^n)}{\mathrm{d}x^n}\right) < 1, \tag{2.11}$$

where $\rho(\cdot)$ denotes the spectral radius. We will show in the example below that the implicit Euler scheme is stable.

A further relevant property of IVPs is stiffness, for which multiple characterizations exist in the literature. A rigorous definition is still missing. A common characterization which is useful for our considerations is given by the following condition. We call a stable IVP stiff at time $t \in I$, if

$$\max_{\lambda\in\sigma(A)} \operatorname{Re}(\lambda) \gg \min_{\lambda\in\sigma(A)} \operatorname{Re}(\lambda) \qquad \text{with } A = \frac{\mathrm{d}f(t,x,q)}{\mathrm{d}x}.$$

Note that the stability assumption requires all $\mathrm{RE}(\lambda)$ to be negative. In a consistent discretization scheme the step size must be chosen properly to guarantee (2.11), which becomes a subtle issue for stiff problems.

Spoken qualitatively, consistency and stability of a scheme imply convergence of a discrete solution sequence $x^1,\dots,x^{n_\tau}$ to the continuous solution $x(t)$ of (2.5). In case of the implicit Euler scheme convergence can, hence, be achieved choosing the step size sufficiently small. Let us consider an example to illustrate how the terms stability and stiffness relate to a simple parabolic problem.

Example 2.2. We follow the discussion in [93] considering the 1D linear heat equation on the domain $\Omega = [0,\pi]$ with homogeneous Dirichlet boundary

$$\begin{aligned} y_t - \Delta y &= 0 && \text{in } I\times\Omega,\\ y &= 0 && \text{on } I\times\partial\Omega,\\ y(0) &= y_s && \text{on } \Omega \end{aligned} \tag{2.12}$$

and apply a finite differences discretization on an equidistant grid with $N+2$ grid points and mesh size h. Denoting with $x_i(t),\ i = 0,\dots,N+1$ the values of the states at the grid points we obtain a linear IVP of the form

$$\dot{x}(t) = Sx(t), \quad x(0) = x_s, \tag{2.13}$$

where $x = (x_1,\dots,x_N)^T$ and $S \in \mathbb{R}^{N\times N}$ is the discrete Laplacian given as

$$S = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{pmatrix}.$$

The node values of x_0 and x_{N+1} are set to zero to account for the boundary conditions. The eigenvalues λ_k and eigenvectors v^k of S have an analytic expression (see also [93]), here given as

$$\lambda_k = 2h^{-2}(\cos(kh) - 1), \quad v_i^k = \sin(ikh), \quad 1 \leq i, k, \leq N.$$

It can be now be seen that the finite difference discretization of (2.12) yields a stable IVP. For the first eigenvalue we find

$$\lambda_1 = 2h^{-2}(\cos(h) - 1) = 2h^{-2} \sum_{i=1}^{\infty} \frac{(-1)^i}{2i} h^{2i} = -1 + \mathcal{O}(h^2)$$

and that the last eigenvalue is

$$\lambda_N = \frac{2(N+1)^2}{\pi^2} \left(\cos\left(\frac{N\pi}{N+1} \right) - 1 \right) \approx -\frac{4(N+1)^2}{\pi^2}.$$

Clearly, when choosing a finer grid, $\lambda_N \to -\infty$. Thus, the IVP (2.13) becomes stiffer with increasing N.

Using the eigenvectors v^k as new basis for (2.13), we can consider the transformed problem

$$\dot{\tilde{x}}_k(t) = \lambda_k \tilde{x}_k(t), \quad \tilde{x}(0) = \tilde{x}_s, \quad k = 1, \ldots, N. \tag{2.14}$$

For $\lambda_k \ll 0$ an explicit time-stepping method has to choose very small time step sizes τ_n to guarantee a stable integration of (2.14). Instead, implicit methods should be used. With an implicit Euler scheme for all $n = 1, \ldots, n_\tau$ we have

$$(\tilde{x}_k^{n+1} - \tilde{x}_k^n)/\tau_n = \lambda_k \tilde{x}_k^{n+1} \quad \Leftrightarrow \quad \tilde{x}_k^{n+1} = \tilde{x}_k^n/(1 - \tau_n \lambda_k),$$

thus, the stability condition (2.11) is satisfied for all $\tau > 0$ as $\lambda_k < 0$.

In practice, however, we do not have the decomposition into modes as in (2.14) of the original problem (2.13). Hence, instead of solving the implicit system analytically we apply a Newton method. In analogy to (2.9) we obtain for (2.14)

$$\tilde{x}_k^{n+1,j+1} = \tilde{x}_k^{n+1,j} - \frac{1}{1 - \bar{\tau}_n \bar{\lambda}_k} \left(\tilde{x}_k^{n+1,j} - \tilde{x}_k^n - \tau_n \lambda_k \tilde{x}_k^{n+1,j} \right),$$

where we use the approximation $1 - \bar{\tau}_n \bar{\lambda}_k$ of the Jacobian of the implicit system describing the time step. According to (2.11) this recursion is stable if, for each k, we have

$$\left| 1 - \frac{1 - \tau_n \lambda_k}{1 - \bar{\tau}_n \bar{\lambda}_k} \right| < 1. \tag{2.15}$$

The condition is satisfied for all τ_n if the approximations $\bar{\tau}_n \bar{\lambda}^k$ of $\tau_n \lambda_k$ are good. For the integration scheme this means that the time instance where the iteration matrix has been built is close to the current time instance. Regarding the choice of the step size we can further transform condition (2.15). Due to $\lambda_j < 0$ it is equivalent to

$$\frac{1 - \tau_n \lambda_k}{1 - \bar{\tau}_n \bar{\lambda}_k} < 2 \quad \Leftrightarrow \quad 1 - \tau_n \lambda_k < 2 - 2\bar{\tau}_n \bar{\lambda}_k \quad \Leftrightarrow \quad \tau_n < \frac{1 + 2\bar{\tau}_n \bar{\lambda}_k}{|\lambda_k|}.$$

An adaptive integrator for stiff problems seeks to choose the step size as large as possible while keeping the scheme stable. We can see that that the step size is limited. However, τ_n can be chosen large given that only low-frequency modes $\tilde{x}_k$ are non-zero at the current state of the time evolution. This would be the case if x^n can be written as

$$x^n = \sum_{k=1}^{l} \tilde{x}_k^n v_k,$$

and $l \ll N$. In contrast when high-frequency modes with $\lambda_k \ll 0$ are present, the step size must be chosen small. The following is a common scenario: Assume high-frequency modes to be present in the initial value x_s. Due to the damping properties of the the Laplace operator, an efficient integrator would choose rather small steps with many computations of $\mathcal{M}$ at the beginning and then increase the step size and the reuse of the iteration matrix. We revisit the issue when discussing derivative computation via internal numerical differentiation in §2.5.2.

2.3. The optimization problem from the ODE perspective

We now look at optimal control problem from the semi-discrete perspective, i.e. after spatial discretization. This ODE perspective is important as big parts of the model reduction discussion are carried out in this setting. In addition we lay out the connection between the infinite-dimensional derivative equations and their semi-discrete counterparts in the second part of this section.

Inserting the semi-discrete versions y^h and u^h as defined in §2.2.1 into the abstract problem (OCP), we can define $J^h(x,q) := J(y^h, u^h)$ with the ODE states x and spatially discrete controls q. This results in an ODE-constrained optimal control problem

$$\begin{aligned} \min_{x,q} \quad & J^h(x,q) \\ \text{s.t.} \quad & M\dot{x}(t) = f(t, x(t), q(t)), \quad q \in \mathcal{Q}^h_{ad}. \end{aligned} \tag{2.16}$$

From the ODE perspective the solution approach in this thesis can be described as a *direct single shooting* method to solve (2.16). 'Direct' as we first discretize the control, which stands in contrast to an indirect approach based on Pontryagin's maximum principle. 'Single shooting' reflects the fact that we follow the reduced approach where all states are eliminated from the optimization problem. Alternative direct approaches for ODE-constrained optimal control problems include *multiple shooting* or *collocation*, where a moderate or a large number of states, respectively, is handled explicitly in the discretized optimization problem. For detailed reading on optimal control with ODE constraints we refer to [23, 48].

Assuming the existence of a solution operator $x(q)$ to (2.7) for every $q \in \mathcal{Q}^h_{ad}$, we can state the semi-discrete reduced problem

$$\min_{q} \quad j^h(q) := J^h(x(q), q) \quad \text{s.t.} \quad q \in \mathcal{Q}^h_{ad}. \tag{2.17}$$

Analogously to §1.3.1 and §1.3.2 we now seek to compute the derivative of the objective $j^h(q)$ with respect to the control. While, technically we do this via techniques based on automatic differentiation on the semi-discrete level (see §2.5), we need to have in mind our goal of commutativity of the DTO and the OTD approaches. Thus, we need to ensure that the discretely computed derivatives are consistent with their time-continuous counterparts. In the following we present semi-discrete derivative equations obtained via the adjoint and the sensitivity approach and discuss their relation to the equations derived in §1.3.

2.3.1. Relation of continuous and semi-discrete adjoints

Starting with the adjoint approach, the linear semi-discrete adjoint equation to problem (2.16) is given as

$$-\dot{z}^T(t) = z^T(t) M^{-1} f_x(t, x(t), q(t)), \quad z^T(T) = J^h_x(x,q), \tag{2.18}$$

with adjoint state variables $z : I \to \mathbb{R}^N$ and $t \in I$. Note that J^h is supposed to depend on states $x(T)$ only at final time, thus, the differentiation subscript x here stands for total differentiation with respect to $x(T)$. The derivation is similar to §1.3.1 and can be found, e.g., in [25]. Given a solution $z(t)$ of (2.18) one finds a time-continuous representation of a directional derivative of the reduced objective in a direction $\tilde{q} \in \mathcal{Q}^h$ as

$$j_q^h(q)\tilde{q} = \int_I z^T(t) M^{-1} f_q(x(t), q(t))\tilde{q} dt + J_q^h(x, q)\tilde{q}. \tag{2.19}$$

We are now interested in the relation between the semi-discrete problem (2.18) and the adjoint problem in weak form (1.15) discretized in space. More particularly, we aim to show that differentiation and discretization commute on the spatial level and their solutions are linearly related via the mass matrix.

It is well known that for pure Galerkin discretizations the DTO and OTD approaches commute, provided that the state equation is given in weak form. Let us illustrate how this can be seen for our method-of-lines approach. Consider the Galerkin approximation of the adjoint variable p

$$p^h(t, r) = \sum_{i=1}^N \tilde{z}_i(t)\varphi_i(r) \quad \in V^h \tag{2.20}$$

and u^h as in §2.2.1. With $\varphi = (\varphi_1, \ldots, \varphi_N)^T$, a Galerkin discretization of (1.15) yields

$$\left\langle -p_t^h + \mathcal{A}_y^*(y^h(t), u^h(t))p^h, \varphi \right\rangle_H = 0, \quad \left\langle p^h(T), \varphi \right\rangle_H = J_y(y^h, u^h)\varphi,$$

which is equivalent to

$$- M\dot{\tilde{z}} + \left\langle \mathcal{A}_y^*(y^h(t), u^h(t))\tilde{z}^T\varphi, \varphi \right\rangle_H = 0, \quad M\tilde{z}(T) = J_y(y^h, u^h)\varphi, \tag{2.21}$$

The following proposition describes the relation between solutions of (2.21) and (2.18).

Proposition 2.1. *Assume that a Galerkin discretization with the same subspace V^h is applied to* (1.2) *and* (1.15)*, and that the same semi-discrete control space $\mathcal{Q}^h$ is chosen. Then for solutions $z(t)$ to* (2.18) *and $\tilde{z}(t)$ to* (2.21) *we have the relation*

$$\tilde{z}(t) = M^{-1}z(t), \quad t \in I \tag{2.22}$$

Proof. With the approximation $y^h(t) \in V^h$ as in (2.4), we can write (2.5) as

$$M\dot{x} = \left\langle \mathcal{A}(y^h(t), u^h(t)), \varphi \right\rangle_H, \quad x(0) = M^{-1}\left\langle y_s, \varphi \right\rangle_H$$

We follow now the derivation process of the semi-discrete adjoint from the Galerkin perspective. To this end, consider the semi-discrete analogon to the Lagrangian (1.14)

$$\begin{aligned}\mathcal{L}^h(q, x, z) := J(y^h, u^h) - \int_I z^T\dot{x} - z^T M^{-1}\left\langle \mathcal{A}(y^h(t), u^h(t)), \varphi \right\rangle_H dt \\ - z^T(0)\left(x(0) - M^{-1}\left\langle y_s, \varphi \right\rangle_H\right),\end{aligned}$$

where $\mathcal{L}^h : \mathcal{Q}^h \times H^1(I, \mathbb{R}^N) \times L^2(I, \mathbb{R}^N) \to \mathbb{R}$ and $z \in L^2(I, \mathbb{R}^N)$ is the Lagrange multiplier. Differentiating the Lagrangian with respect to x in a direction $\tilde{x} \in H^1(I, \mathbb{R}^N)$ we obtain

$$\mathcal{L}_x^h(q, x, z)\tilde{x} = J_y(y^h, u^h)\tilde{y}^h - \int_I z^T\dot{\tilde{x}} - \left\langle \mathcal{A}_y(y^h(t), u^h(t))\tilde{y}^h, z^T M^{-1}\varphi \right\rangle_H dt - z^T(0)x(0),$$

where $\tilde{y}^h := \frac{\mathrm{d}}{\mathrm{d}x} y^h\tilde{x} = x^T\varphi$. Adjoining the spatial and temporal operators and setting $\mathcal{L}_x^h(q, x, z)\tilde{x} = 0$ yields

$$0 = \tilde{x}^T J_y(y^h, u^h)\varphi - \int_I \tilde{x}^T\left(-\dot{z} - \left\langle \varphi, \mathcal{A}_y^*(y^h(t), u^h(t))z^T M^{-1}\varphi \right\rangle_H\right) dt - \tilde{x}^T(T)z(T).$$

Requiring the last equation to hold for all $\tilde{x} \in H^1(I, \mathbb{R}^N)$ yields an adjoint solution $z \in C^1(I, \mathbb{R}^N)$ that also solves

$$-\dot{z} = \left\langle \mathcal{A}_y^*(y^h(t), u^h(t)) z^T M^{-1} \varphi, \varphi \right\rangle_H, \quad z(T) = J_y(y^h, u^h)\varphi. \tag{2.23}$$

A comparison of variables between (2.23) and (2.21) concludes the proof. q.e.d.

Key to the proof is that in either derivation of the adjoint we computed the exact derivative of the operator $\mathcal{A}(y(t), u(t))$ and used the exact adjoint in $\langle \cdot, \cdot \rangle_H$. The derivations differ only in the trial and test functions being once in V and once in V^h. The commutation property in general might get lost, e.g., when stabilization terms as streamline diffusion [34] are added.

Remark 2.4. In the DTO approach the discretization of the adjoint is automatically determined by the choice of the discretization of the states. In contrast, via OTD one has the freedom to choose a particular approximation space for the adjoint variable p.

2.3.2. Relation of continuous and semi-discrete sensitivities

We now draw our attention to the semi-discrete version of the sensitivity equations. Differentiation of the ODE problem (2.5) with respect to the control q in a direction $\tilde{q} \in L^2(I, \mathbb{R}^{n_q})$ yields

$$\dot{w}(t) = f_x(t, x(t), q) w(t) + f_q(t, x(t), q)\tilde{q}, \quad w(0) = 0. \tag{2.24}$$

with $w(t) := x_q(t)\tilde{q}$. In the ODE context (2.24) is typically called variational differential equation (VDE). The derivative of the reduced semi-discrete objective in the direction $\tilde{q}$ is given as

$$j_q^h(q)\tilde{q} = \; J_x^h(x(T), q) w(T) + J_q^h(x(T), q)\tilde{q}. \tag{2.25}$$

It can be easily seen that the VDE is the discretization of the sensitivity equation (1.20) when identical Galerkin subspaces V^h are used. Thus, choosing V^h as trial and test space to obtain a solution $\tilde{y}^h(t) = y_q^h(t)\tilde{q}$ of (1.20) with direction $\tilde{q} \in \mathcal{Q}^h$, we have the relation

$$\tilde{y}^h(t) = w(t)^T \varphi \quad \forall t \in I. \tag{2.26}$$

2.4. Newton-type methods

In this section we recall concepts of Newton-type methods that can be used to solve the optimization problem (OCP) once transformed into a finite dimensional Nonlinear Programming Problem (NLP). In particular we present the sequential quadratic programming (SQP) method and the Gauss–Newton method, which we use to solve optimal control and parameter estimation problems respectively. We restrict ourselves to basic ideas and refer the reader to the comprehensive textbook on numerical optimization by Nocedal and Wright [87]. In our practical implementation we use the software package SNOPT Version 7 [49] for optimal control problems and a Gauss–Newton method implemented in Matlab® by ourselves.

Let us consider the NLP (2.2) in the form

$$\min_q \quad j^{h\tau}(q) \quad \text{s.t.} \quad q \in \mathcal{Q}_{ad}^{h\tau}, \tag{2.27}$$

with $j^{h\tau}(q) = j(u^{h\tau})$ and $\mathcal{Q}_{ad}^{h\tau}$ being the discretized constraints. We use q from now on also for the fully discretized control vector. It will be clear from the context whether $q \in \mathcal{Q}^h$ or

$q \in \mathcal{Q}^{h\tau}$, with $\mathcal{Q}^{h\tau} = \mathbb{R}^{n_q}$ and n_q the number of discretized control variables. For the sake of simplicity we lay out the basic concepts only for the unconstrained case, as the handling of constraints is not the focus of this thesis. In presence of equality and inequality constraints standard techniques are applied in our numerical computations (see also the documentation of SNOPT [49]).

The first-order optimality conditions of an NLP can be expressed in the form

$$\mathcal{F}(q) := \nabla j^{h\tau}(q) = 0 \tag{2.28}$$

where $\mathcal{F} : \mathcal{D} \subseteq \mathbb{R}^{n_q} \to \mathbb{R}^{n_q}$ and $q \in \mathbb{R}^{n_q}$ is assumed to be a fully-discretized control vector. As $\mathcal{F}$ is in general nonlinear we compute the solution iteratively via

$$\Delta q^n := -\mathcal{M}(q^n)\mathcal{F}(q^n), \quad q^{n+1} = q^n + \alpha_n \Delta q^n, \tag{2.29}$$

where $\mathcal{M}(q^n) \in \mathbb{R}^{n_q \times n_q}$ and α_n the step length, also called the damping parameter. We refer to Newton-type methods for any method that can be expressed in the above form.

Depending on the choice of $\mathcal{M}$ we obtain specific methods. E.g., choosing $\mathcal{M} = \mathcal{J}^{-1}$, where $\mathcal{J}$ is the Jacobian of $\mathcal{F}$, gives the classic Newton method. If $\mathcal{M}$ is an approximation of the inverse of $\mathcal{J}$ one speaks of a Quasi-Newton method.

The question of global convergence arises, where the choice of the damping parameter α_k is of importance. Typically a trust-region or a line-search method is employed. The latter we briefly address in the context of the Sequential Quadratic Programming method. Regarding local convergence of Newton-type methods in §2.4.3 we recall the local contraction theorem by Bock.

2.4.1. Sequential quadratic programming

SQP is a class of optimization algorithms where the NLP is replaced by a quadratic approximation at the current iterate. In each step a quadratic subproblem is solved to find a new search direction and subsequently a globalization strategy is employed to find a new iterate. Particular methods differ in how the latter two issues are tackled. For illustration we consider the unconstrained problem

$$\min_{q} \quad f(q) := j^{h\tau}(q) \tag{2.30}$$

with the scalar objective $f : \mathbb{R}^q \to \mathbb{R}$. The quadratic programming (QP) problem in each step then is given by

$$\min_{\Delta q^n} \quad (\Delta q^n)^T H_n \Delta q^n + (\nabla f(q^n))^T \Delta q^n.$$

For the Hessian approximation H_n BFGS updating is applied in SNOPT for which superlinear convergence can be expected under certain conditions. The new iterate is found via the Newton-type iteration

$$q^{n+1} = q^n + \alpha_n \Delta q^n \quad \text{where } \Delta q^n = -H_n^{-1} \nabla f(q^n).$$

Applying a line search strategy to determine q^{n+1} one tries to find a sufficient decrease in some merit function where for simplicity we consider

$$\phi(\alpha_n) := f(q^n + \alpha_n \Delta q^{n+1}) \quad \alpha_n > 0.$$

With regard to the use of model reduction techniques, the following issues become relevant to us. Given that H_n is positive definite, which is guaranteed by the BFGS update, a decrease in $\phi(\alpha_n)$ can only be assured if $\nabla f(q^n)$ is a sufficiently good approximation of the gradient of f. While this is of importance to any type of NLP approximating the gradient properly is a challenging task in the model reduction context. The issue is addressed in §4.4. A further aspect is that due to the BFGS updating the method exhibits superlinear local convergence properties. In practice, however, the SQP solver needs a few iterations to build a good approximation of the Hessian. Thus, it is beneficial to keep the SQP optimization procedure untouched as long as possible. In our model reduction approach a series of NLPs is solved. Thus, the question of when to reconstruct the underlying NLP is important for efficiency (see §5.2).

2.4.2. Gauss–Newton method

We employ a Gauss–Newton method to solve the (reduced) parameter estimation problem (1.23), written in the form

$$\min_{q\in\mathbb{R}^{n_q}} \frac{1}{2}\left\|\mathcal{F}(q)\right\|^2. \tag{2.31}$$

Recall that for parameter estimation problems we assume also in the general PDE-constrained case that $u \in \mathbb{R}^{n_u}$. Here and in the remainder in the parameter estimation context we consider

$$\mathcal{F}_i(q) := \frac{\eta_i - h(y^h(\tilde{t}_i; q), u)}{\varsigma_i}, \quad i = 1, \dots, n_{\text{meas}}, \tag{2.32}$$

which is the semi-discretized analogon of $\mathcal{F}(u)$ as in §1.4. Notational confusion can, thus, be excluded. To $\mathcal{F}(q)$ we also refer as the residual of the parameter estimation problem.

In analogy to the SQP method we consider a linearized problem

$$\min_{\Delta q^n} \quad \frac{1}{2}\left\|\mathcal{F}(q^n) + \mathcal{J}(q^n)\Delta q^n\right\|_2^2 \quad \text{with} \ \ \mathcal{J}(q^n) := \left.\frac{\mathrm{d}\mathcal{F}(q)}{\mathrm{d}q}\right|_{q^n}. \tag{2.33}$$

Throughout, we require $\mathcal{J}$ to satisfy

$$\operatorname{rank}(\mathcal{J}(q)) = n_q \quad \forall q \in \mathbb{R}^{n_q} \tag{2.34}$$

which implies that we have at least as many measurements as parameters, i.e., $n_{\text{meas}} \geq n_q$. Under the assumptions the solution of (2.33) is given by

$$\Delta q^n = -\mathcal{J}^\dagger(q^n)\mathcal{F}(q^n) := -(\mathcal{J}^T(q^n)\mathcal{J}(q^n))^{-1}\mathcal{J}^T(q^n)\mathcal{F}(q^n), \tag{2.35}$$

where the operator $\mathcal{J}^\dagger$, often referred to as the Moore–Penrose pseudoinverse, is continuously differentiable with respect to q given $\mathcal{F} \in C^2(\mathbb{R}^{n_q})$. As before the new q^{n+1} is determined via a Newton-type iteration.

We further see that

$$\frac{\mathrm{d}}{\mathrm{d}q}\frac{1}{2}\left\|\mathcal{F}(q)\right\|^2 = \mathcal{J}^T(q)\mathcal{F}(q).$$

Due to assumption (2.34), $\mathcal{J}^T(q^n)\mathcal{J}(q^n)$ in (2.35) is positive definite. Thus, when using gobalization strategies for Gauss–Newton finding a descent direction requires a proper approximation of $\mathcal{J}^T(q)$ and $\mathcal{F}(q)$. To obtain the Jacobian $\mathcal{J}$ it is necessary to compute the sensitivities for all canonical directions $e_j \in \mathbb{R}^{n_q}$, $j = 1, \dots, n_q$ as

$$y_q^h(\tilde{t}_i, q)e_j, \quad \tilde{t}_1, \dots, \tilde{t}_{n_{\text{meas}}}, \tag{2.36}$$

which can be obtained as solution of the VDE (2.24). Thus, later in the model reduction approach we need to find good approximations not only of the gradient of the objective, but also of the sensitivities of the states y^h with respect to the parameters at all measurement time points $\tilde{t}_i$.

Note that one could also use an adjoint approach for the computation of $\mathcal{J}$, however, this would require n_{meas} adjoint directions to be propagated in the adjoint equation, which in general is less efficient as we assume $n_{\text{meas}} > n_q$.

Via the Jacobian $\mathcal{J}$ we can also analyze the statistical quality of an estimated parameter vector. As the measurements are random variables, this is also true for each solution $q^\star$ to (1.23). Thus, we consider the variance-covariance matrix C in the solution $q^\star$, given by

$$C(q^\star) := (\mathcal{J}(q^\star)^T J(q^\star))^{-1}.$$

From the diagonal entries of C one can compute linear approximations of the confidence regions of the parameter estimates. The variance-covariance matrix becomes of particular interest when solving experimental design problems, where one seeks to reduce the entries of $C(q^\star)$ by designing the experiment properly. We refer to [21, 72] for further information.

2.4.3. Local convergence of Newton-type methods

In §5.3.1 we present an a posteriori estimate for optimization solutions obtained with the reduced-order model based on the local contraction theorem by Bock [21]. Therefore, we recall a variant as in [93]. The proof is based on the Banach fixed-point theorem and can be found in [21, 93].

Let $\mathcal{F}, \mathcal{M}$, and $\mathcal{D}$ be given as in the Newton-type iterations (2.29) and $\mathcal{J}$ be the Jacobian of $\mathcal{F}$. We denote by $\mathcal{N}$ the set of Newton pairs defined as

$$\mathcal{N} := \{(q, q') \in \mathcal{P} \times \mathcal{P} |\ q' = q - \mathcal{M}(q)\mathcal{F}(q)\}. \tag{2.37}$$

and require the following two conditions on $\mathcal{J}$ and $\mathcal{M}$ (compare also [37]).

Definition 2.2 (**ω-condition**). The Jacobian $\mathcal{J}$ and the matrix $\mathcal{M}$ satisfy the ω-condition on $\mathcal{D}$ if there exists $\omega < \infty$ such that for all $\xi \in [0, 1]$ and $(q, q') \in \mathcal{N}$

$$\|\mathcal{M}(q')\left(\mathcal{J}\left(q + \xi(q' - q)\right) - \mathcal{J}(q)\right)(q' - q)\| \leq \omega\xi \|q' - q\|^2 .$$

Definition 2.3 (**κ-condition**). $\mathcal{M}$ satisfies the κ-condition on $\mathcal{D}$ if there exists $\kappa < 1$ such that for all $(q, q') \in \mathcal{N}$

$$\|(\mathcal{M}(q'))(\mathcal{F}(q) - \mathcal{J}(q)\mathcal{M}(q)\mathcal{F}(q))\| \leq \kappa \|q' - q\| .$$

Definition 2.4 (**contraction ball**). Let $\delta_n := \kappa + \frac{\omega}{2} \|\Delta q^n\|$ where $\Delta q^n = -\mathcal{M}(q^n)\mathcal{F}(q^n)$. If $\delta_0 < 1$, we define the contraction ball

$$\mathcal{D}_0 := \left\{ q \in \mathbb{R}^{n_p} |\ \|q - p_0\| \leq \frac{\|\Delta q^0\|}{1 - \delta_0} \right\}.$$

Theorem 2.5. **(Local Contraction Theorem)**
Let $\mathcal{J}$ and $\mathcal{M}$ satisfy the ω- and κ-conditions on $\mathcal{D}$. Further, assume that there is an initial guess $q^0 \in \mathcal{D}$ such that we have

$$\delta_0 < 1 \quad \text{and } \mathcal{D}_0 \subset \mathcal{D}.$$

Then the following holds:

(1) *The iterates $q^{n+1} = q^n + \Delta q^n$ are well defined and $q^n \in \mathcal{D}_0$ for all n.*

(2) *There exists $q^\star \in \mathcal{D}_0$ such that $\lim_{n\to\infty} q^n = q^\star$.*

(3) *The a priori estimate* $\left\|q^{n+l} - q^\star\right\| \leq \frac{\|\Delta q^n\|}{1-\delta_n}\delta_n^l$ *holds.*

(4) *The steps satisfy* $\left\|\Delta q^{n+1}\right\| \leq \delta_n \left\|\Delta q^n\right\|$*.*

(5) $\mathcal{M}(q^\star)\mathcal{F}(q^\star) = 0$

Bock's local contraction theorem describes the local convergence properties of Newton-type methods. The ω-condition is a measure for the nonlinearity of the problem, e.g., for linear problems we have $\omega = 0$. The κ-condition measures how well $\mathcal{M}$ approximates the inverse of the Jacobian. This can be seen in the alternative formulation (provided $\mathcal{M}$ is invertible)

$$\left\|\mathcal{M}(q')(\mathcal{M}^{-1}(q) - \mathcal{J}(q))(q'-q)\right\| \leq \kappa \left\|q'-q\right\|.$$

With Gauss–Newton we have $\mathcal{M}(q) = \mathcal{J}^\dagger(q) = (\mathcal{J}^T(q^n)\mathcal{J}(q^n))^{-1}\mathcal{J}^T(q^n)$ and $\mathcal{F}(q)$ represents the components of the least-squares objective as defined in (2.32). Due to condition (2.34) we find

$$\Delta q = -\mathcal{J}^\dagger(q^\star)\mathcal{F}(q^\star) = 0 \;\Leftrightarrow\; \mathcal{J}(q^\star)\mathcal{F}(q^\star) = \frac{\mathrm{d}}{\mathrm{d}q}\frac{1}{2}\left\|\mathcal{F}(q)\right\|^2 = 0, \tag{2.38}$$

thus, the size of the increment Δq is a suitable stopping criterion for the Gauss–Newton iterations in the unconstrained case. In contrast for general Newton-type methods the choice of a stopping criterion is less clear. Thus, we use the stopping criterion of the employed solver, providing it with the necessary objective, derivative, and constraint information (see SNOPT manual [49] for details).

Moreover, for Gauss–Newton we find that the κ-condition in a solution $q^\star$ is equivalent to

$$\frac{\left\|(\mathcal{J}^\dagger(q') - \mathcal{J}^\dagger(q^\star))\mathcal{F}(q^\star)\right\|}{\left\|q'-q^\star\right\|} \leq \kappa.$$

Requiring $\kappa < 1$ is, hence, a restriction on the nonlinearity of the problem and on the size of the parameter estimation residual $\mathcal{F}$. These conditions can be interpreted in such a way that the Gauss–Newton method tends to be attracted only by statistical relevant minima (see [21]). In addition, it can be seen that for small residuals, $\mathcal{J}^T\mathcal{J}$ is a good approximation of the Hessian of the parameter estimation objective (see, e.g., [87]) which allows for fast convergence of the method. The latter two aspects motivate our choice to use Gauss–Newton. Note that with Gauss–Newton in general the numerical effort for the computation of the increment is larger than providing the gradient via an adjoint approach as for optimal control problems.

2.5. Numerical Differentiation

In practice, we face the problem of computing a variety of derivatives that are necessary not only for computing, e.g., the gradient of the NLP (2.2), but also for simulation of the states, provided that a implicit time-stepping scheme is used. For an efficient use in optimization we require a fast and accurate computation of derivatives. We do not want to provide the derivatives in analytical form by hand, as this is error prone and barely tractable for relevant real-world problems which may contain complicated nonlinear model functions. A common technique to evaluate derivatives in a convenient way is finite differences where, however, we introduce truncation errors which is often unacceptable for the accuracy demands in optimization.

In our direct approach we need to compute derivatives on two different levels. On the one hand, we have to provide derivatives of the right-hand sides f of the dynamic systems to the integrator. For this purpose we use AD which allows us to obtain derivatives that are exact up to machine precision in an almost 'automatic' way. The basic concepts are presented in the first part of this section. On the other hand, due to the direct approach we need to compute derivatives of the reduced objective in the NLP (2.2), meaning that we need to differentiate the integration scheme used to solve the underlying dynamic system. For the discretely computed derivatives, in addition, we have the goal (see §2.1) that they are consistent with solutions of the adjoint and sensitivity equations which we introduced in §2.3. We achieve this by following the principle of internal numerical differentiation (IND) [19] which is closely related to AD and which we discuss in §2.5.2. In the last part of this chapter we point out difficulties when applying IND that arise in particular with ordinary differential equations (ODEs) stemming from PDE discretizations.

2.5.1. Automatic differentiation

The technique of automatic differentiation (AD), often also referred to as *Algorithmic Differentiation*, is based on the fact that every mathematical function implemented as computer code is a sequence of basic mathematical operations of which the exact derivatives are known. Combining this with an efficient application of the chain rule, we obtain a variety of techniques to obtain a desired directional derivative. The technique is appealing due to the following properties:

- AD computes the numerically exact derivatives.
- Provided the code is given in a common programming language (C/C++, Fortran, Matlab®), there are tools available to compute the derivatives in a quasi automatic way.
- Derivatives can be computed in an efficient way when structures are exploited.

For a comprehensive study of the topic we suggest the textbook by Griewank [51].

From an illustrative perspective, each mathematical program can be considered as a directed graph with a set of input nodes, output nodes, and inner nodes that represent basic elementary operations. In the AD context, input nodes represent the *independent variables* and output nodes are called *dependent variables*. The situation is depicted in Figure 2.2. Two basic ways to compute directional derivatives are distinguished. Traversing the function graph in Fig. 2.2 from the left to the right, computing the function values together with the derivatives at each node and applying the chain rule is called the *forward mode* of AD. This is the analogon to the sensitivity approach for the computation of derivatives for PDEs as in §1.3.2. In the so-called *reverse mode* of AD, the graph is first evaluated,

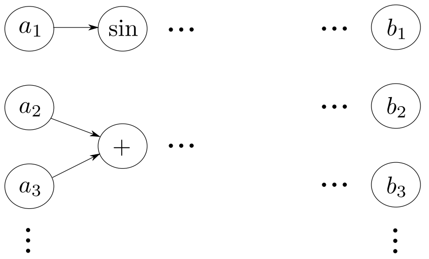

Figure 2.2.: Directed graph representation of mathematical a function. Independent variables (left) are mapped to the dependent variables (right) via elementary mathematical operations.

remembering all intermediate values. Subsequently traversing from the right to the left, evaluating derivatives, and applying the chain rule backwards, one obtains adjoint directional derivatives. Hence, the reverse mode is often also called the *adjoint mode* of AD, which is related to the adjoint approach as described in §1.3.1.

Expressed in a formal way, we consider a function $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ which maps the independent variables a to the dependent variables b. Given a direction $\dot{a} \in \mathbb{R}^n$, with the forward mode we compute

$$\dot{b} = \frac{\mathrm{d}\Phi(a)}{\mathrm{d}a}\dot{a} \quad \in \mathbb{R}^m.$$

To compute the full Jacobian of Φ, n directional derivatives need to be computed, thus, the complexity of the forward mode is $\mathcal{O}(n)$. The reverse mode computes for a given adjoint direction $\bar{b} \in \mathbb{R}^m$

$$\bar{a}^T = \bar{b}^T \frac{\mathrm{d}\Phi(a)}{\mathrm{d}a} \quad \in \mathbb{R}^n.$$

Here the complexity to compute the full Jacobian is given as $\mathcal{O}(m)$. For a common optimization problem we have $m = 1$, hence, the adjoint mode should be chosen. In contrast when we are interested in directional derivatives of a given function with many outputs, e.g., in parameter estimation problems, then the forward mode is the method of choice.

2.5.2. Principle of internal numerical differentiation

In this section we present the concepts for an efficient differentiation of an integration scheme that is used to solve the ODE (2.5) in §2.2.1. Differentiation of the scheme is necessary as it is part of the evaluation of the reduced objective function $j^{h\tau}(q)$ in (2.27) and we require the gradient $\nabla j^{h\tau}(q)$ for the optimization. After a full discretization of the problem, $j^{h\tau}(q)$ can also be considered as a black-box function consisting of a sequence of basic mathematical operations. Thus, one could apply techniques such as finite differences or AD to this function as discussed above. We refer to this strategy as external numerical differentiation (END). Besides the issue of low accuracy with finite differences, the problem of the END approach is that the procedure to evaluate $j^{h\tau}(q)$ contains operations that are not differentiable, e.g., the choice of the step size during the time integration. This may lead the gradient $\nabla j^{h\tau}(q)$ to be inconsistent with $j^{h\tau}(q)$. Moreover, we have the goal to be able to interpret our DTO approach also as an OTD approach as discussed in §2.1.

To this end, in this thesis we follow the principle of internal numerical differentiation (IND) which was first proposed by Bock [19]. With the IND principle, on the one hand, we achieve the goal of commutativity of DTO and OTD on the semi-discrete level and, on the other hand, due to its close relation to AD we can expect an efficient and accurate computation of the necessary derivatives. Further reading on IND and its application to BDF methods as well as the implementation in the software package DAESOL-II can be found in [2, 3, 14].

As discussed in §2.2.2 we apply implicit time-stepping methods to solve the ODE problem (2.5). We assume these methods to be adaptive, e.g., in the choice of the step size or the number of evaluations of the iteration matrices necessary to solve the implicit systems in each step. Conceptionally we apply the forward and reverse mode of AD to the sequence of operations that are specified by the choice of the particular time discretization method. We refer to this as either *forward IND* or *adjoint IND* respectively. The following two aspects define what we understand as the principle of IND:

1. When differentiating the integration scheme, all adaptive components remain fixed.

2. The choice of the adaptive components should be oriented towards the stability requirements of the state problem and the adjoint/sensitivity problems simultaneously.

With this definition we are aligned with the characterization of IND as, e.g., in [93]. For a successful and efficient application of IND one should consider three additional important aspects of IND in a particular implementation. For a compact overview we also add these aspects to the characterization of IND:

3. If matrix decompositions are used for the solutions of the linear subproblems, these should be reused.
4. The time integration scheme should be chosen such that consistency of discrete and continuous derivatives can be achieved.
5. Close to an optimal solution it may be necessary to fix the discretization scheme for several iterations of the optimization.

We now discuss these topics, their relation, and their implications for the resulting derivatives.

In aspect 1. we require the discretization scheme to remain frozen when differentiating the time integration. This is crucial for an efficient derivative computation for dynamic systems based on AD and is also discussed in [40]. Firstly, with this we overcome the problem of non-differentiable operations in the scheme typically introduced by the controllers that determine the adaptive components. Secondly, by freezing the scheme and assuming numerically exact derivatives of the right-hand sides (e.g., via AD) we obtain the numerically exact derivative of a fixed integration scheme. This was referred to by Bock as the *analytical limit of IND* [20]. Thirdly, for certain integration schemes (see aspect 4.) it can be shown that by following aspect 1., the discretely computed derivatives are a solution to a consistent integration scheme applied to the VDE (2.24) or the adjoint ODE (2.18) respectively. With forward IND this is guaranteed if a linear time-stepping method is used [19]. The interpretation of discrete IND adjoints as solutions of (2.18) via a consistent scheme is more complicated. For Runge–Kutta methods consistency of IND adjoints was shown first by Bock in [21] and later by [101, 118]. In [102] it is shown for linear multi-step methods that discrete adjoints in general are inconsistent with the continuous adjoint solutions. In [16] BDF methods are discussed and it is analyzed how this inconsistency can be overcome. It is shown that a Hilbert space setting is not suitable to establish consistency between IND adjoints an their continuous counterpart and, therefore, it is necessary to interpret the discrete adjoints in a Banach space setting. We illustrate in Example 2.3 that the IND adjoint scheme of the implicit Euler is again an implicit Euler scheme.

In aspect 2. we require that the discretization scheme allows a stable integration of the adjoint ODE or VDE respectively. While for VDEs this could be achieved by controlling also the error in the VDE variables during the integration (see [2]), for the computation of the discrete adjoints the scheme is already fixed on the whole time horizon when starting the reverse sweep. Hence, the stability of the adjoint scheme can only be influenced during the forward integration. At the end of §2.5.3 we present heuristics to attenuate this problem.

Regarding computational efficiency of IND, the reuse of the matrix decompositions in the integration scheme is recommended in the third aspect above. The Jacobians of the right-hand side of the state problem and the adjoint or VDE problem respectively are identical. As the computation of the Jacobian of the right-hand sides is often the most time consuming part this can be a significant improvement in computation time.

Regarding the use of IND in an optimization algorithm the following becomes important. As the time integration scheme is chosen adaptively, the considered NLP slightly alters in each optimization iteration, as possibly a different scheme is used due to the changing optimization variable. Thus, in aspect 5. we suggest to fix the scheme over multiple optimization iterations which fixes the NLP and makes the IND derivatives exact up to machine precision. This allows to obtain fast convergence and allows the optimization algorithm to satisfy small termination tolerances.

We sum up interpreting IND from two different viewpoints. As motivated, IND is an application of AD to the integration scheme, where the dependence of the adaptively chosen components with respect to the controls is neglected. Hence, an efficient and accurate computation of the derivatives can be expected. Secondly, under the assumptions the discrete derivatives are computed with a stable scheme and are consistent approximations of the adjoint ODE or the VDE respectively (with a possible adaption of the space in which solutions are considered).

Remark 2.5. Regarding our overall solution approach we can summarize the following. By virtue of the identities (2.22) and (2.26) we can see that the semi-discrete adjoint and sensitivity problems are consistent approximations of the infinite-dimensional problems. Thus, spatial discretization and differentiation commute under the assumptions in this thesis. In this section we have shown that by following the IND principle we obtain an analogous result also for the time discretization step. Hence, the discrete derivatives computed for the NLP (2.1) are also a consistent approximation of the infinite-dimensional optimality conditions stated in §1.3. We can conclude that for the chosen solution approach, so far discretize-then-optimize and optimize-then-discretize commute.

Example 2.3. Let us illustrate a few aspects of forward IND using an implicit Euler method with variable step size and an evaluation of the Newton iteration matrix in each time integration step. For simplicity assume that one Newton step is performed to determine the solution x^{n+1} of the implicit Euler system, the initial guess for the Newton iteration is the current state approximation x^n, and $M = \mathbb{I}$. Writing down the rule including all dependencies, we arrive at

$$\begin{aligned} x^{n+1}(q) &= x^n(q) + \mathcal{M}_n(q)\tau_n(q)f(t_{n-1}, x^n(q), q) \\ \text{where } \ \mathcal{M}_n(q) &= \left(\mathbb{I} - \tau_n(q)f_x(t_{n-1}, x^n(q), q)\right)^{-1}. \end{aligned}$$

One can easily see that the scheme is consistent for bounded f_x. We are now interested in the sensitivity of x^{n+1} with respect to q, thus, in AD notation we have now $b = x^{n+1}$ and $a = q$ and we consider the Newton step as an elementary function. Clearly, the elements τ_n and $\mathcal{M}_n$ defining the scheme also depend on q. However, following aspect 1. of IND we neglect this influence on the adaptive components and obtain the derivative

$$\begin{aligned} x_q^{n+1}(q)\tilde{q} = x_q^n(q)\tilde{q} &+ \mathcal{M}_n\tau_n(q)f_x(t_{n-1}, x^n(q), q)x_q^n(q)\tilde{q} \\ &+ \mathcal{M}_n\tau_n(q)f_q(t_{n-1}, x^n(q), q)\tilde{q}, \end{aligned}$$

with a direction $\dot{a} = \tilde{q} \in \mathbb{R}^{n_q}$. Setting $w^n = x_q^n(q)\tilde{q}$ this is obviously the same as applying the chosen discretization scheme to the VDE (2.24). Moreover, we can see that the influence of $\mathcal{M}_n$ on x^{n+1} vanishes when a fix point is found in the Newton iteration. Hence, the derivative becomes more accurate if the implicit system is solved with higher precision.

2.5.3. Adjoint IND and its application to semi-discrete PDEs

We now briefly investigate some peculiarities of adjoint IND. In particular we consider the case of an implicit Euler method and discuss its application to systems stemming from PDE discretizations.

We use again the implicit Euler scheme as in Example 2.3. Differentiation of

$$x^{n+1}(x^n) = x^n + \mathcal{M}_n\tau_n f(t_{n-1}, x^n, q)$$

with respect to x^n, multiplying with an adjoint direction $\bar{x}^{n+1} \in \mathbb{R}^N$, and following the IND principle, we get

$$\begin{aligned}(\bar{x}^n)^T &= (\bar{x}^{n+1})^T + (\bar{x}^{n+1})^T \mathcal{M}_n \tau_n f_x(t_{n-1}, x^n, q)\\ &= (\bar{x}^{n+1})^T \mathcal{M}_n \left(\mathcal{M}_n^{-1} + \tau_n f_x(t_{n-1}, x^n, q)\right)\\ &= (\bar{x}^{n+1})^T \mathcal{M}_n,\end{aligned} \tag{2.39}$$

where we set $(\bar{x}^n)^T := (\bar{x}^{n+1})^T \frac{\mathrm{d}}{\mathrm{d}x^n} x^{n+1}$. The last equality holds as we assume $\mathcal{M}_n$ to be built in every time step. Now consider the same discretization scheme applied to the adjoint problem (2.18) integrated backwards in time

$$\begin{aligned}(z^n)^T &= (z^{n+1})^T + (z^{n+1})^T \tau_n f_x(t_n, x^{n+1}, q) \tilde{\mathcal{M}}_n\\ &= (z^{n+1})^T \tilde{\mathcal{M}}_n,\end{aligned}$$

where $\tilde{\mathcal{M}}_n = (\mathbb{I} - \tau_n f_x(t_n, x^{n+1}, q))^{-1}$ and $z^n := z(t_{n-1})$ as for the states. Obviously, the adjoint IND scheme (2.39) and the implicit Euler scheme applied to the adjoint equation yield the same rule for the time stepping, differing only in the particular time instances where the Jacobian of the right-hand side for the iteration matrix is evaluated. I.e., for adjoint IND $\mathcal{M}_n$ is evaluated at an earlier time instance t_{n-1} in comparison to t_n for $\tilde{\mathcal{M}}$. Clearly, both are consistent schemes which can be traced back to the linear nature of the adjoint problem.

We now review the stability issues in Example 2.2. With adjoint IND we encounter the following situation. According to the discussion in Example 2.2 one would expect a variable step size τ_n to be rather large at the end of the time horizon and the last decomposition point to lie back a few iterations. We assume this due to the high-frequency modes of the system having been damped out due to the properties of the diffusion operator. When interpreting adjoint IND as integration scheme of the adjoint problem this means that we start with a large step size and a possibly bad approximation of the iteration matrix. Now assume that we want to solve an optimization problem with objective

$$J(y,u) = \int_\Omega (y - y_\Omega)^2 dr, \quad y_\Omega(r) = \begin{cases} 1, & \text{if } r > 0.5,\\ 0, & \text{otherwise,}\end{cases}$$

on the domain $\Omega = (0,1)$. Assuming $y_\Omega \in V^h$, the adjoint direction is then given as $z(T) = M(x(T) - x_\Omega(T))$, thus, the first adjoint iterate $z^{n_\tau+1}$ contains high-frequency modes stemming from the discontinuity in y_Ω. Hence, the situation may result in the discrete IND scheme to be unstable. An analogous situation is discussed for forward IND in [93].

In our numerical results we use the software tool DAESOL-II. We cope with the issue of stability applying the following measures depending on the requirements of a particular problem.

- Reducing the order of the BDF method to one, as the stability properties of the method decrease with increasing order (BDF schemes are stable only for order ≤ 6).

- Forcing the iteration matrix to be rebuild more often. In DAESOL-II we achieve this by decreasing the error tolerance of the Newton method with which the implicit systems are solved.

- Forcing the iteration matrix to be rebuild in each step of the integration.

- Fixing the step size to a sufficiently small value.

We apply these strategies based on our experience with the particular problem we consider in the applications. For the numerical tests regarding error estimates for proper orthogonal decomposition (POD) we use an implicit Euler method, which we enforce by reducing the maximum order to one, using a sufficiently small maximum step size, and setting the relative error tolerance of the integration to a large value.

Part II.

Model Order Reduction

3. POD/DEIM Reduced Order Models

The expression model order reduction (MOR), also referred to as *reduced-order modeling*, comprises a variety of methods to tackle the problem of large dimensions in discretized dynamic problems. Essentially one is interested in simplifying a dynamic system such that its evaluation costs are reduced, however, preserving the input-output behavior. From a broad perspective, MOR techniques first appeared in the fields of system and control theory with major developments in the 1980s. Among the important contributions we find the articles [84] by Lumley and [107] by Sirovich, which are relevant to us as there the proper orthogonal decomposition (POD) method for model reduction was introduced as it is conceptionally used today for time-dependent PDEs. In subsequent years POD MOR became also of interest to numerical mathematicians which resulted in a further fast development in the field.

A division into three classes of MOR methods is done in [5]. The first class are singular value decomposition (SVD) based methods which contain, e.g., POD and balanced truncation methods [124]. They exhibit the property that the essential characteristics such as stability of the original system are preserved. This comes comes at the cost of an increased effort for the construction of the so-called reduced-order model (ROM). In contrast, Krylov methods [10] typically do not exhibit these properties. However, due to their iterative nature they are applicable to problems of larger size and their implementation is typically easier. As a third class one finds methods where a combination of both techniques is applied. In this thesis we are exclusively interested in model reduction based on POD.

Due to their importance in the MOR community, we also mention the family of reduced basis methods [89] which overlap with the above three classes. These methods are applied to parametrized systems, where one is interested in system responses for a variety of parameter configurations, the so-called *snapshots*. One characteristic of the reduced basis family is that one distinguishes between an offline and an online phase. Through the distinction into different phases these methods became of particular interest in the real-time context. Applying POD techniques in a reduced-basis-method fashion is also referred to as *interpolation based POD*. Typically, this is done for stationary problems. In an offline phase the reference system is first solved for certain choices of parameter configurations and then POD is applied to the obtained snapshots to construct a reduced basis. However, with regard to efficiency in the optimization context we cannot afford the offline phase to be too extensive. In this thesis we consider only instationary problems where one follows a different approach to obtain the data for the decomposition. Here the dynamic system is solved once and the state solution at certain time instances are used as snapshots. This is sometimes also referred to as *projection based POD*.

Another dichotomy to be mentioned in the MOR context are linear and nonlinear problems. Methods suitable for linear problems are, e.g., balanced truncation or the class of Krylov methods. To date the POD MOR techniques appear to be the state of the art for nonlinear problems. An important contribution to the handling of the nonlinearity in POD reduced-order models is the discrete empirical interpolation method (DEIM) [31]. With DEIM the cost of the evaluation of the nonlinearity can be further reduced which typically makes the overall simulation cost of the reduced-order model negligible in comparison to the high-dimensional discretized dynamic system.

While different from their perspectives there is a unifying feature in MOR techniques. Most methods can be interpreted as a projection technique, i.e., the truncation of the state solution in an appropriate basis. Thus, we can introduce MOR techniques in a general and abstract way by considering a transformation $T \in \mathbb{R}^{N\times N}$ applied to the states x such that $\bar{x} = Tx$, where $\bar{x} \in \mathbb{R}^N$. We define

$$\begin{aligned} \bar{x} := \begin{pmatrix} \widehat{x} \\ x' \end{pmatrix}, \quad T := \begin{bmatrix} W^T \\ T_1^T \end{bmatrix}, \quad T^{-1} := \begin{bmatrix} U & T_2 \end{bmatrix}, \\ \text{with} \qquad \widehat{x} \in \mathbb{R}^k, \quad U, W \in \mathbb{R}^{N\times k}, \quad k < N, \end{aligned}$$

hence, as $W^T U = \mathbb{I}$ we find that

$$\Pi = UW^T$$

is an oblique projection along the kernel of W^T onto the k-dimensional subspace spanned by the columns of U. Insertion into the ODE system (2.5) yields

$$\begin{pmatrix} \dot{\widehat{x}} \\ \dot{x}' \end{pmatrix} = \begin{pmatrix} W^T f(U\widehat{x} + T_2 x', t) \\ T_1^T f(U\widehat{x} + T_2 x', t) \end{pmatrix},$$

noting that this is still an exact expression. The reduction step consists in only considering the first k equations and omitting $T_2 x'$. The MOR methods distinguish in the particular choices of W and U and pursue the goal of minimizing the neglected parts. In this thesis we consider the POD case with projection $\Pi = UW^T$ and $W^T = \Psi^T M$, $U = \Psi$. and the case of DEIM, where the nonlinearity F as defined in (2.7) is substituted by the projection ΠF with $W^T = (P^T\Phi)^{-1}P^T$ and $U = \Phi$. In §3.1 and §3.2 we take a closer look on the details for the computation of the matrices Ψ, Φ and P and in chapter 4 we present novel techniques for the enhancement of these matrices for the purpose of optimal control and parameter estimation.

To any ROM we present here, we also refer to as *surrogate model*, especially in the optimization context where we seek to use it as prediction instead of some given reference model. To the 'full-order' reference model that is subject of the reduction step we refer as *high-fidelity* or, short, HiFi model. Its particular form and how it is obtained by spatial discretization of the PDE we presented in §2.2.1.

In [103] a broad collection of articles concerning the MOR topic can be found. An overview is given in the textbook [5]. The use of model reduction techniques in the context of robust optimal control is handled in [124]. Further online information can be found, e.g., within the MoRePaS project[1].

3.1. Proper orthogonal decomposition

Proper orthogonal decomposition for model reduction is nowadays often briefly summed up as a popular method, showing competitive results in practical linear and nonlinear applications "despite of a certain heuristic flavor"[112]. In the following sections we explain basic concepts of POD, rediscover some of the heuristic flavor, and present results that give a rigorous mathematical fundament to certain heuristics that have been applied in earlier contributions.

The basic concept of POD was first introduced by Pearson [90]. It is also known under the expressions Karhunen–Lòeve expansion or principal component analysis, depending on the field where it is applied. Historically the most important fields of application are in fluid dynamics [18, 50, 63, 84] and in image processing and pattern recognition [44, 98].

Conceptionally starting from a given set of data vectors, one tries to find a basis of low dimension such that the data can be expressed in that basis while minimizing the approximation error in a least-squares sense in an appropriate norm. To this basis we also

[1] http://www.morepas.org/

refer as *POD modes*. The reduced-order model can then be obtained either by projection onto the subspace, spanned by the optimal basis or by performing a Galerkin discretization using the optimal basis as trial and test functions. Applied properly both yields the same result. While the provenance of the data can be arbitrary, we are here interested in data that is obtained via simulation of the full dynamic system. This points towards an important property of POD in model reduction. I.e., by using information obtained from a solution of the dynamic system to construct the basis, it is legitimate to assume that the surrogate still reflects the essential physical properties of the system. This stands in contrast to common Galerkin discretizations as, e.g., finite element methods (FEMs) where the basis does not contain any a priori information of the physical behavior of the system. A further characteristic of the POD approach is that the Galerkin trial and test functions are optimal in a least-squares sense. Often only a small number of modes is necessary to capture the energy of the system. The method also stands out for its ease of applicability to nonlinear problems as only basic matrix operations are used. Even though the reduction step is performed by an oblique projection onto a linear subspace, the resulting reduced-order model is, however, still nonlinear.

Under the assumption that the POD subspace is sufficiently rich, the model reduction approach can be highly efficient. However, here one aspect of the heuristic flavor comes into play. While for many practical examples POD model reduction performs well, for other applications the assumption is not satisfied and the lack of a priori error estimates in the POD approach can destroy its efficiency. The problem is revisited in this thesis and we present novel approaches to deal with it.

An overview to POD model reduction is found in [63], a detailed technical description of the reduction steps is given in [74], and a recent comprehensive literature regarding POD can be found in the lecture notes of Volkwein [116]. We first present the basic techniques for computing a POD basis in a time-discrete and a time-continuous setting. Then we show in more detail how the reduced-order models are obtained. In the last section of this chapter we present theoretical bounds for surrogate models based on POD and DEIM on different levels of discretization. These estimates describe how well a solution of a surrogate model approximates the HiFi model solution and they are the basis for further investigations of the derivative-enhancement in chapter 4.

3.1.1. Time-discrete POD

With the notation as in chapter §1, assume that we have m snapshots $y^j := y(t_{j-1}) \in V$ at time instances $t_0, \dots, t_{m-1}$. We define

$$\tilde{V} := \operatorname{span}\{y^j\}_{i=1}^m \quad \text{and} \quad \dim \tilde{V} = d,$$

with $d \leq m$, hence, $\tilde{V} \subset V$. Let $\psi_1, ..., \psi_d \in V$. We call $\psi_1, ..., \psi_k$ a *POD basis* of order k if for every $k \leq d$ it is a solution of the optimization problem

$$\begin{aligned} \min_{\psi_1,\dots,\psi_k} \quad & \sum_{i=1}^m \gamma_i \left\| y^i - \sum_{j=1}^k (y^i, \psi_j)_H \psi_j \right\|_H^2 \\ \text{s.t.} \quad & (\psi_n, \psi_l)_H = \delta_{nl}, \; 1 \leq n, l \leq k, \end{aligned} \tag{3.1}$$

with the weights

$$\gamma_1 = (t_1 - t_0)/2, \quad \gamma_i = (t_i - t_{i-2})/2, \quad 2 \leq i \leq m-1, \quad \gamma_m = (t_{m-1} - t_{m-2})/2. \tag{3.2}$$

Due to the orthonormality of the set $\{\psi_i\}_{i=1}^k$, an equivalent characterization of the POD basis is given by the maximization problem

$$\max_{\psi_1,\dots,\psi_k} \quad \sum_{i=1}^{m}\sum_{j=1}^{k}\gamma_i(y^i,\psi_j)_H^2 \qquad \text{s.t.} \quad (\psi_n,\psi_l)_H=\delta_{nl}, \quad 1\le n,l\le k.$$

To a given POD basis $\psi_1,\dots,\psi_k \in V$ we can also consider the *POD subspace* which is defined by

$$V^k := \operatorname{span}\{\psi_i\}_{i=1}^k \subseteq \tilde{V} \subseteq V. \tag{3.3}$$

Regarding the computation of a POD basis we define the linear operator $\mathcal{R}_m : V \to \tilde{V} \subset V$ as

$$\mathcal{R}_m\phi := \sum_{i=1}^{m}\gamma_i(y^i,\phi)_H y^i \quad \forall\phi\in V,$$

noting that $\mathcal{R}_m$ is bounded, compact, symmetric and non-negative. Consider the eigenvalue problem

$$\mathcal{R}_m\phi=\lambda\phi, \quad \phi\in V. \tag{3.4}$$

From the Hilbert-Schmidt theorem we know that there exists an orthonormal set $\{\psi_i\}_{i=1}^d$ and a sequence of eigenvalues $\{\lambda_i\}_{i=1}^d$ such that

$$\mathcal{R}_m\psi_i=\lambda_i\psi_i, \quad \lambda_1>\cdots>\lambda_d>0.$$

Note that, thus, $\psi_1,...,\psi_d$ is an orthonormal basis of $\tilde{V}$.

Theorem 3.1. *Let $\{\psi_i\}_{i=1}^d$ be a solution to* (3.4) *and $\{\lambda_i\}_{i=1}^d$ the corresponding eigenvalues. Then for every $k\le d$ the eigenfunctions $\psi_1,\dots,\psi_k$ are a solution to the optimization problem* (3.1). *The residual in the solution is given as*

$$\sum_{i=1}^{m}\gamma_i\left\|y^i-\sum_{j=1}^{k}(y^i,\psi_j)_H\psi_j\right\|_H^2=\sum_{i=k+1}^{d}\lambda_i. \tag{3.5}$$

To the residual we will also refer as *POD projection error.* The proof to Theorem 3.1 can be found in [114].

We now draw our attention to time-discrete POD in the semi-discrete context which is also the situation we typically face in a practical application. For convenience, we introduce the abbreviation $x^j := x(t_{j-1})$ for the state solutions of (2.7) analogous to $y^j\in V$. Consider the *POD snapshot matrix*

$$\mathcal{S} := \begin{bmatrix}\sqrt{\gamma_1}x^1 & \dots & \sqrt{\gamma_m}x^m\end{bmatrix} \quad \in\mathbb{R}^{N\times m}$$

where in analogy to the dimension of the space $\tilde{V}$ we denote its column rank by $d\le N$. We now solve the POD optimization problem in finite dimension, i.e., we seek to find the *POD projection matrix* $\Psi\in\mathbb{R}^{N\times k}$, obtained as solution of

$$\min_{\Psi\in\mathbb{R}^{N\times k}} \quad \sum_{i=1}^{m}\gamma_i\left\|x^i-\Psi\Psi^TMx^i\right\|_X^2 \quad \text{s.t.} \qquad \Psi^TM\Psi=\mathbb{I}, \tag{3.6}$$

where we use the norm $\|\cdot\|_X$ belonging to the M-inner product $\langle\cdot,\cdot\rangle_X$, with the mass matrix M as introduced in §2.2.1. The objective in (3.6) is right away obtained from (3.1) under the assumption that $y^i\in V^h$, $i=1,\dots,m$ and $\psi_i\in V^h$, $i=1,\dots,k$. The matrix

$$\mathbf{R}_m := \mathcal{S}\mathcal{S}^TM, \tag{3.7}$$

is the spatially discretized equivalent to the operator $\mathcal{R}_m$. A solution to problem (3.6) is, hence, given by the eigenvectors of the problem

$$\mathbf{R}_m \phi = \lambda \phi, \quad \phi \in \mathbb{R}^N, \tag{3.8}$$

which is either obtained via Galerkin projection of (3.4) onto the space V^h or as optimality conditions to (3.6). With the square root $M^{1/2}$ of the symmetric and positive definite matrix M, defined as $M^{1/2}M^{1/2} = M$, we set $\bar{\phi} := M^{1/2}\phi$, $\bar{\mathcal{S}} := M^{1/2}\mathcal{S}$ and multiply (3.8) from the left with $M^{1/2}$ to get

$$\bar{\mathcal{S}}\bar{\mathcal{S}}^T \bar{\phi} = \lambda \bar{\phi}, \quad \bar{\phi} \in \mathbb{R}^N.$$

There are two common ways to solve the finite dimensional eigenvalue problem related to the POD optimality (3.6).

Variant I: We apply a SVD on the matrix

$$\bar{\mathcal{S}} = A \Sigma B^T$$

with orthonormal matrices $A \in \mathbb{R}^{N \times d}$, $B \in \mathbb{R}^{m \times d}$, and a diagonal matrix $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_d)$ containing the singular values of $\bar{\mathcal{S}}$. Defining $A_k := [A_{.1}, \dots, A_{.k}]$, the POD projection matrix is given by

$$\Psi := M^{-1/2} A_k \ \in \mathbb{R}^{N \times k}$$

and for the eigenvalues we have $\lambda_i = \sigma_i^2, \ i = 1, \dots, d$.

Variant II: We solve the $m \times m$ eigenvalue problem

$$\bar{\mathcal{S}}^T \bar{\mathcal{S}} \tilde{\phi} = \mathcal{S}^T M \mathcal{S} \tilde{\phi} = \lambda \tilde{\phi}, \quad \tilde{\phi} \in \mathbb{R}^N.$$

The solution is given by the columns of the matrix B that appears in the SVD of $\bar{\mathcal{S}}$ in Variant I. Due to the properties of the SVD, we have

$$A = \bar{\mathcal{S}} B \Sigma^{-1},$$

thus, the projection matrix is given as

$$\Psi_{.i} = \frac{1}{\lambda_i} M^{-1/2} \bar{\mathcal{S}} B = \frac{1}{\lambda_i} \mathcal{S} B, \quad i = 1, \dots, k.$$

The second variant of computing the projection matrix is often referred to as the *method of snapshots* proposed by Sirovich [107]. Note that the naming is rather historical, as in either variant the computation of snapshots precedes the decomposition. A discussion of snapshot POD vs. classical POD is carried out in [59].

With fine discretizations of the underlying PDE we have $N \gg m$, hence, solving for the eigenvalues of $\bar{\mathcal{S}}^T \bar{\mathcal{S}}$ in Variant II is numerically more efficient. Moreover, the computation of $M^{1/2}$ and $M^{-1/2}$ requires additional effort, which is not needed in the method of snapshots. In contrast, SVD has better stability properties, hence, the projection matrix can be computed more accurately, which is of interest for the numerical investigation of approximation errors. The relation between Ψ and the POD modes in a Galerkin space V^h is described by

$$\psi_j(r) = \sum_{i=1}^{N} \Psi_{ij} \varphi_i(r), \qquad j = 1, \dots, k,$$

and the residual of (3.6) is given by the eigenvalues of (3.8)

$$\sum_{i=1}^{m} \gamma_i \left\| x^i - \Psi \Psi^T M x^i \right\|_X^2 = \sum_{i=k+1}^{d} \lambda_i. \tag{3.9}$$

Note that we have the relations

$$\langle \psi_i, \psi_j \rangle_H = \langle \Psi_{.i}, \Psi_{.j} \rangle_X = \langle \Psi_{.i}, M \Psi_{.j} \rangle = \delta_{ij}, \quad 1 \le i, j \le k.$$

Remark 3.1. For the POD subspace, after Galerkin discretization we have $V^k \subseteq V^h$. Moreover, $V^k = V^h$ if $d = N$ and we choose $k = d$. In this case using a POD basis reduces to a basis transformation.

Remark 3.2. The choice of k is crucial for the performance of POD. We consider three possibilities. Often the ratio between the modeled and the total energy is considered, given by

$$\mathcal{E}(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq 1 - \mathcal{E}_{TOL},$$

where $\mathcal{E}_{TOL} > 0$ (compare [74]). For the investigations of the POD reconstruction error in §3.3.5 and 4.5 we use a similar criterion. We choose k such that

$$\left(\sum_{i=k+1}^{d} \lambda_i \right)^{\frac{1}{2}} \leq \epsilon_P,$$

i.e., we want the square root of the projection error to be below a certain threshold ϵ_P. Thus, the criterion ϵ_P is closely related to the error with which the snapshots are approximated in their natural norm. In the optimization context it may be more efficient to only compute a moderate number of eigenvalues of $\mathcal{S}^T M \mathcal{S}$. Then, we rely on the mere value of the $(k+1)$-th eigenvalue, i.e., we choose k such that

$$\sqrt{\lambda_{k+1}} \leq \lambda_{TOL} \quad \text{with} \quad \lambda_{TOL} > 0.$$

The actual values of the eigenvalues are highly problem dependent. An exponential decay is observed, for example, in many problems in fluid dynamics and diffusion dominated parabolic PDEs. Thus, with a fast decay, already for small k the projection error $\sum_{i=k+1}^{d} \lambda_i \ll 1$ and only a small number k of POD modes is necessary to capture most of the information given by the snapshots.

3.1.2. Time-continuous POD

So far we have not commented on the time weights $\gamma_1, \dots, \gamma_k$ in the definition of the POD basis. Their choice can be motivated by introducing a continuous version of POD. Moreover, by the introduction of the time-continuous setting we are independent of the concrete choice of the snapshots and the discretization scheme that is used to solve the underlying ODE. Assume that we have a continuous solution $x(t)$ of the HiFi problem (2.7). A further characterization of the POD basis is then given as solution of the optimization problem

$$\min_{\Psi \in \mathbb{R}^{N \times k}} \int_I \left\| x(t) - \Psi \Psi^T M x(t) \right\|_X^2 dt \quad \text{s.t.} \quad \Psi^T M \Psi = \mathbb{I}. \tag{3.10}$$

The problem (3.6) is obtained as discretization of (3.10) using the trapezoidal rule. The relation between solutions to (3.10) and (3.6) is investigated in [76]. It is shown that the first k dominant eigenvectors of the matrix $\mathbf{R} = \int_I x(t) x(t)^T M dt$ are a solution to (3.10). Note that $\mathbf{R}_m$ is obtained applying the trapezoidal rule to $\mathbf{R}$. Let $\bar{\lambda}_1 \geq \ldots \geq \bar{\lambda}_N \geq 0$ be the eigenvalues of $\mathbf{R}$, where for $d = \operatorname{rank}(\mathbf{R}) < N$, $\bar{\lambda}_j = 0$, $d < j \leq N$. Analogous to the time-discrete projection error we find that

$$\int_I \left\| x(t) - \Psi \Psi^T M x(t) \right\|_X^2 dt = \sum_{i=k+1}^{N} \bar{\lambda}_i. \tag{3.11}$$

The eigenvalues of the discrete and the time-continuous version of POD obey the following relation (see also [116]). Assuming a fixed and sufficiently small time step τ between snapshots, we have

$$\sum_{i=k+1}^{d} \lambda_i \leq 2 \sum_{i=k+1}^{N} \bar{\lambda}_i, \quad \text{and} \quad \lambda_j \to \bar{\lambda}_j,\ 1 \leq j \leq k \ \text{ as } \tau \to 0. \tag{3.12}$$

Note that d depends on the size of the time step which corresponds to the number of snapshots m.

3.1.3. POD Galerkin discretization

Assume now that the m snapshots $x^1, \dots, x^m \in \mathbb{R}^N$ are evaluations at time instances $t_0, \dots, t_{m-1}$ of a solution $x(t)$ of the HiFi problem (2.7). Neglecting the constant part as discussed in Remark 2.2 the HiFi problem is given as

$$M\dot{x}(t) = Sx(t) + F(x(t), q(t)), \quad x(0) = x_s,\ t \in I.$$

Via the POD method in §3.1.1 we compute the matrix Ψ and the POD basis $\psi_j \in V^h,\ j = 1, \dots, k$. There are now two ways to construct the reduced-order model, both leading to the same result. Processing as described at the beginning of the chapter, we employ a projection of the HiFi model by setting $x = \Psi\widehat{x}$, where $\widehat{x}$ is a vector-valued function from I to $\mathbb{R}^k$, and multiplying (2.7) from the left by Ψ^T to obtain

$$\dot{\widehat{x}}(t) = \widehat{S}\widehat{x}(t) + \Psi^T F(\Psi\widehat{x}(t), q(t)), \quad \widehat{x}(0) = \Psi^T M x^0, \tag{P-ROM}$$

with $\widehat{S} := \Psi^T S \Psi$. Alternatively we consider the POD Galerkin approximation

$$y^k(t, r) := \sum_{i=1}^{k} \widehat{x}_i(t)\psi_i(r), \quad (t, r) \in I \times \Omega.$$

Insertion of y^k in the weak form (1.2) and using the POD modes for the Galerkin discretization in the space V^k yields

$$\frac{\mathrm{d}}{\mathrm{d}t} \left\langle y^k(t), \psi_j \right\rangle_H = \left\langle A(y^k(t), u^h(t)), \psi_j \right\rangle_H, \quad \left\langle y^k(0) - y_s, \psi_j \right\rangle_H, \quad j = 1, \dots, k.$$

An exact evaluation of the last equation results again in (P-ROM). We can see that the procedure can also be interpreted as a projection of V^h on the POD subspace V^k. Note that the mass matrix for the reduced-order model turns out to be the identity, which is due to the orthonormality constraint in the POD optimality. Moreover, note that the model still contains the same nonlinearity F, evaluated at all discrete state discretization points. We dedicate ourselves to the reduction of the evaluation cost of the nonlinear part in the next section.

Remark 3.3. In practical applications, sometimes the mass and stiffness matrices are not available, e.g., when using a software package to solve the PDE without access to the internal data structures. While then we cannot do the projection step, it is still possible to construct (P-ROM) using the Galerkin discretization. Note that then we have to rely on quadrature rules to evaluate the stiffness matrix which is an additional error source for approximations via POD surrogate models.

Remark 3.4. Due to the fact that we require the model to be given in weak form with boundary conditions already included as in (1.2), we obtain a direct relation between HiFi and reduced-order model via projection. This allows for highly accurate POD approximations under certain assumptions that will be investigated in §3.3. A similar discussion related to this issue is found in [70], where the authors show an a priori stability guarantee for POD surrogate models with weakly imposed boundary conditions.

3.2. Discrete empirical interpolation method

In this section we have the objective of reducing the computational cost for the evaluation of the nonlinear term in the reduced-order model. We follow the ideas presented by Chaturantabut and Sorensen [31], which are closely related to the *empirical interpolation method* (EIM) proposed in [13]. While the latter is purely based on a greedy algorithm, in the DEIM approach first a POD basis is computed and then a greedy algorithm is applied to select the interpolation indices. Other approaches to reduce the complexity of the evaluation of the nonlinearities are found, e.g., in [7, 33, 113].

3.2.1. DEIM projection

Consider again the POD reduced-order model (P-ROM)

$$\dot{\widehat{x}} = \underbrace{\widehat{S}}_{k\times k}\ \widehat{x} + \underbrace{\Psi^T}_{k\times N} \underbrace{F(\Psi\widehat{x}, q)}_{N\times 1},$$

of dimension $k \ll N$. Obviously, the reduced linear part is cheap to evaluate, however, in the nonlinear part we still have evaluation costs of complexity N of the HiFi model.

Let us consider the nonlinear part F as a mapping of data to the vector $\mathrm{f}(t) \in \mathbb{R}^N,\ t \in I$. By projection of $\mathrm{f}(t)$ onto a subspace $R^\ell = \operatorname{span}\{\Phi_{\cdot 1}, \ldots, \Phi_{\cdot \ell}\} \subseteq \mathbb{R}^N, \Phi \in \mathbb{R}^{N\times\ell}$ of dimension $\ell \ll N$ we have its approximation as

$$\mathrm{f}(t) \approx \Phi c(t), \quad c(t) \in \mathbb{R}^\ell. \tag{3.13}$$

To determine the corresponding coefficients $c(t)$ from the overdetermined system $\mathrm{f}(t) = \Phi c(t)$, we define

$$P := [e_{\wp_1}, \ldots, e_{\wp_\ell}] \in \mathbb{R}^{N\times\ell}$$

where $e_{\wp_i}$ is the $\wp_i$-th column of the identity matrix $\mathbb{I} \in \mathbb{R}^{N\times N}$. Hence, the matrix multiplication $P^T\Phi$ corresponds to a selection of ℓ rows of Φ. If $P^T\Phi$ is nonsingular, we can determine $c(t)$ uniquely from the system

$$P^T\mathrm{f}(t) = P^T\Phi c(t),$$

to obtain $c(t) = (P^T\Phi)^{-1}P^T\mathrm{f}(t)$. Thus, a DEIM approximation $\widehat{\mathrm{f}}(t)$ is given as

$$\widehat{\mathrm{f}}(t) := \Phi(P^T\Phi)^{-1}P^T\mathrm{f}(t) = \mathbb{P}\mathrm{f}(t),$$

with $\mathbb{P} := \Phi(P^T\Phi)^{-1}P^T$. Insertion of the approximation into the model (P-ROM) yields the POD/DEIM reduced-order model

$$\dot{\widehat{x}}(t) = \widehat{S}\widehat{x}(t) + \Psi^T\mathbb{P}F(\Psi\widehat{x}(t), q(t)), \quad \widehat{x}(0) = \Psi^T M x^0. \tag{PD-ROM}$$

As we can see, the complexity of the evaluation of the nonlinear part has been reduced from N to ℓ. The expression $\Psi^T\Phi(P^T\Phi)^{-1} \in \mathbb{R}^{k\times\ell}$ can be computed when constructing the surrogate model and only ℓ entries of the nonlinear vector F need to be evaluated.

A particular DEIM projection $\mathbb{P}$ is obtained by

1. Selecting the interpolation indices $\wp_1, \dots, \wp_\ell$ and the matrix P.
2. Choosing the DEIM projection matrix Φ.

For the interpolation we employ the Greedy algorithm presented in [31], which we recall in Algorithm 1. DEIM, in contrast to the empirical interpolation method, considers only discrete evaluation points in the domain Ω and selects the indices inductively limiting the error bound given in Lemma 3.2.

Algorithm 1 Greedy algorithm for DEIM

```
Require: $\phi_1, \dots, \phi_\ell \in \mathbb{R}^N$ linearly independent
    $[\rho, \wp_1] = \mathtt{max}(\mathtt{abs}(\phi_1))$
    $\Phi = \begin{bmatrix}\phi_1\end{bmatrix}, \; P = \begin{bmatrix}e_{\wp_1}\end{bmatrix}$
    for $n = 2, \dots, \ell$ do
        Solve $(P^T\Phi)c = P^T\phi_n$ for $c$
        $\tilde{r} = \phi_n - \Phi c$
        $[\rho, \wp_n] = \mathtt{max}(\mathtt{abs}(\tilde{r}))$
        $\Phi = \begin{bmatrix}\Phi & \phi\end{bmatrix}, \; P = \begin{bmatrix}P & e_{\wp_n}\end{bmatrix}$
    end for
return $P$
```

In Algorithm 1 we use a Matlab® oriented notation, i.e., `abs` returns the absolute values for each vector element and `max` chooses the maximum value of a vector return the value and the index of the vector element.

The projection matrix Φ is computed by applying POD to the DEIM snapshot matrix

$$\mathcal{D} := \begin{bmatrix}\sqrt{\gamma_1}F(x^1, q(t_0)) & \dots & \sqrt{\gamma_m}F(x^m, q(t_{m-1}))\end{bmatrix} \tag{3.14}$$

which are the evaluations of the nonlinearities in the HiFi model at the snapshot locations weighted with the time weights. For the sake of simplicity, in the following we omit the time argument of the control $q \in L^2(I, \mathbb{R}^{n_q})$ in the nonlinearity F and assume the control to be evaluated at the time instance corresponding to the snaphots. Φ is obtained from $\mathcal{D}$ analogous to §3.1, setting $M = \mathbb{I}$. The relation between the DEIM projection and the POD optimality is established in the next section.

Remark 3.5. The nonlinear part $F(x(t), q)$ in general is of the form

$$F(x(t), q) = B_1\tilde{F}(B_2x(t), q), \quad B_1 \in \mathbb{R}^{N\times n_f}, \; B_2 \in \mathbb{R}^{n_f\times N},$$

where n_f is the number of quadrature evaluation points and the Jacobian of the nonlinearity $\tilde{F}$ w.r.t. x is a diagonal matrix, i.e., each entry of the vector $\tilde{F} \in \mathbb{R}^{n_f}$ depends on exactly one entry of B_2x and $\tilde{F}$ is properly ordered. Then it is more efficient to construct the (PD-ROM) as

$$\dot{\widehat{x}}(t) = \widehat{S}\widehat{x}(t) + \Psi^T B_1\mathbb{P}\tilde{F}(B_2\Psi\widehat{x}(t), q),$$

with $\mathbb{P} \in \mathbb{R}^{n_f}$, the selection matrix P is chosen as in Algorithm 1 and instead of the matrix $\mathcal{D}$ we use

$$\tilde{\mathcal{D}} := \left[\sqrt{\gamma_1}\tilde{F}(B_2x^1, q), \dots, \sqrt{\gamma_m}\tilde{F}(B_2x^m, q)\right].$$

3.2.2. Error estimates for DEIM

The following Lemma quantifies the error of the DEIM approximation $\widehat{\mathrm{f}}$.

Lemma 3.2. *Let* $\mathrm{f} \in \mathbb{R}^N$ *and* $\Phi \in \mathbb{R}^{N\times\ell}$ *be a matrix with orthonormal columns,* $\widehat{\mathrm{f}} = \mathbb{P}\mathrm{f}$ *be the DEIM approximation with* $\mathbb{P} = \Phi(P^T\Phi)^{-1}P^T$*, and* P *the result of Algorithm 1. Then we have the estimate*

$$\left\|\widehat{\mathrm{f}} - \mathrm{f}\right\| \leq C_D \left\|\mathrm{f} - \Phi\Phi^T\mathrm{f}\right\| \tag{3.15}$$

with $C_D = \left\|(P^T\Phi)^{-1}\right\|$.

Proof. We have

$$\begin{aligned} \left\|\widehat{\mathrm{f}} - \mathrm{f}\right\| &= \left\|\mathbb{P}\mathrm{f} + \Phi\Phi^T\mathrm{f} - \mathrm{f} - \Phi\Phi^T\mathrm{f}\right\| \\ &= \left\|\mathbb{P}\mathrm{f} + \mathbb{P}\Phi\Phi^T\mathrm{f} - \mathrm{f} - \Phi\Phi^T\mathrm{f}\right\| \\ &= \left\|(\mathbb{P} - \mathbb{I})\mathrm{w}\right\| \\ &\leq \left\|\mathbb{P} - \mathbb{I}\right\| \left\|\mathrm{w}\right\|, \end{aligned}$$

where we defined $\mathrm{w} = \mathrm{f} - \Phi\Phi^T\mathrm{f}$. Moreover,

$$\begin{aligned} \left\|\mathbb{P} - \mathbb{I}\right\| = \left\|\mathbb{P}\right\| &= \left\|\Phi(P^T\Phi)^{-1}P^T\right\| \\ &\leq \left\|(P^T\Phi)^{-1}\right\|, \end{aligned}$$

where for the first equation we use the results in [109] and exploit $\|\Phi\| = \|P\| = 1$ for the inequality. q.e.d.

Note that due to the orthogonal projection $\Phi\Phi^T$, $\Phi\Phi^T\mathrm{f}$ yields the best approximation of f in the space spanned by the columns of Φ. As a result of Lemma 3.2 in particular we have

$$\mathrm{f} - \mathbb{P}\mathrm{f} = (\mathbb{I} - \mathbb{P})\mathrm{w}, \tag{3.16}$$

with $\mathrm{w} := \mathrm{f} - \Phi\Phi^T\mathrm{f}$, which will be of use for the error estimates in §3.3. The estimate in Lemma 3.2 motivates the choice to employ POD to compute the DEIM subspace R^ℓ. Using the Euclidean norm in the POD optimality (3.6), the matrix Φ satisfies

$$\min_{\Phi\in\mathbb{R}^{N\times\ell}} \quad \sum_{i=1}^{m} \gamma_i \left\|F(x^i, q) - \Phi\Phi^T F(x^i, q)\right\|^2 \quad \text{s.t.} \qquad \Phi^T\Phi = \mathbb{I}, \tag{3.17}$$

which is the same as considering the snapshot matrix $\mathcal{D}$ for the POD method. Equivalently to POD model reduction, choosing a sufficient number of snapshots and ℓ sufficiently large we assume the space R^ℓ to reflect the essential dynamic behavior of the nonlinear F part of the HiFi model. Analogously we have an estimate

$$\sum_{i=1}^{m} \gamma_i \left\|F(x^i, q) - \Phi\Phi^T F(x^i, q)\right\|^2 = \sum_{i=\ell+1}^{\tilde{d}} s_i, \tag{3.18}$$

for the eigenvalues $s_1, \ldots, s_{\tilde{d}}$ of $\mathcal{D}\mathcal{D}^T$ with $\tilde{d} = \operatorname{rank}(\mathcal{D})$. Finally, we pose the continuous POD optimality for the DEIM projection, which we use for the estimates in §3.3.3. The projection matrix Φ is supposed to satisfy

$$\min_{\Phi\in\mathbb{R}^{N\times\ell}} \quad \int_I \left\|F(x(t), q) - \Phi\Phi^T F(x(t), q)\right\|^2 dt \quad \text{s.t.} \qquad \Phi^T\Phi = \mathbb{I} \tag{3.19}$$

and the estimate for the projection error is

$$\int_I \left\|F(x(t), q) - \Phi\Phi^T F(x(t), q)\right\|^2 dt = \sum_{i=\ell+1}^{N} \bar{s}_i, \tag{3.20}$$

with corresponding eigenvalues $\bar{s}_1, \ldots, \bar{s}_N$ analogous to §3.1.1

3.3. Error analysis

In the following we recall error estimates for POD/DEIM reduced-order models, which we extend to a more general scenario for its application to semi-discrete PDEs. With the estimates we assess the error between solutions of the full problems and the reduced-order problems for various scenarios. The estimates are of a priori type as the bounds are not affected by the surrogate solutions. However, by construction the estimates involve information of solutions of the HiFi model.

We start explaining the types of error we are concerned with, namely reconstruction and prediction errors. In [75, 76] general a priori error estimates in function space for reconstruction errors of reduced-order models of type (P-ROM) are given for linear and semi-linear parabolic PDEs. Their results are briefly recalled in §3.3.2. In §3.3.3 and §3.3.4 we give a priori estimates for the spatially discretized version as in [32], which also comprise the error introduced by the DEIM projection as in (PD-ROM). We extend their results to oblique POD projections with a weighted inner product norm.

3.3.1. Reconstruction vs. prediction

In this thesis we distinguish between two types of error that we are interested in when using surrogate models. We refer to these as the *reconstruction error* and the *prediction error*.

Definition 3.3 (Reconstruction error). For $\bar{q} \in \mathcal{Q}^h$, let $x(\bar{q})$ be the solution operator satisfying the HiFi problem (2.7) and $\widehat{x}(\bar{q})$ the solution operator that satisfies the surrogate model (PD-ROM). Assume $\xi(\bar{q})$ to be an operator depending on $\bar{q}$ via $x(\bar{q})$ and $\widehat{\xi}(\bar{q})$ to be the same operator depending on $\bar{q}$ via $\Psi\widehat{x}(\bar{q})$. The *POD reconstruction error* is then defined as

$$RE\,(\xi) := \frac{\left\|\xi(\bar{q}) - \widehat{\xi}(\bar{q})\right\|}{\|\xi(\bar{q})\|}, \tag{3.21}$$

where the surrogate model is constructed at $\bar{q}$.

The norm is chosen according to the codomain of the operator ξ, e.g., we consider $(\int_I \|\cdot\|_X^2 \, dt)^{\frac{1}{2}}$ for ξ being the identity and the absolute value for the quantity $j^h(x(\bar{q}))$. In particular the reconstruction error is evaluated for the same control $\bar{q} \in \mathcal{Q}_{ad}^h$ for which the reduced basis is constructed. In case of ξ being the gradient it also depends on either adjoint or sensitivity information. The reconstruction errors for ξ is then evaluated analogously to the states with the respective HiFi or surrogate approximation.

Definition 3.4 (Prediction error). With the definition of $\xi(q)$ as in the reconstruction error, we define the *POD prediction error* between HiFi and surrogate model as

$$PE\,(\xi) := \frac{\left\|\xi(q) - \widehat{\xi}(q)\right\|}{\|\xi(q)\|}, \qquad q, \bar{q} \in \mathcal{Q}^h, \ q \neq \bar{q}, \tag{3.22}$$

where the surrogate model used to evaluate the approximation $\widehat{x}(q)$ is constructed at $\bar{q}$.

Clearly, the prediction error is of major interest when solving optimal control problems, where the control variable changes in each iteration of the optimization algorithm. We discuss the prediction error and the use of surrogate models for optimization in more detail in chapter 5.

The term 'reconstruction' reflects that a first interest in POD model reduction is to approximate the already given HiFi information as good as possible with the surrogate model.

Therefore, the investigation of the reconstruction error is an analysis of the consistency between the HiFi model and its surrogate at a given control $\bar{q}$. In §3.3.3 we give a bound on the reconstruction error with the operator ξ being the identity. For the estimate we assume exact solutions to the HiFi and the surrogate model equations. After discretization the operators $x(q)$ and $\widehat{x}(q)$ are no more exact solutions of their corresponding model equations. Thus, to analyze the reconstruction error we need to take into account the error introduced by the time integration scheme. For the estimates in §3.3.4 it is assumed that an equidistant time grid and an implicit Euler method is applied to solve the semi-discrete problems. Thus, we conserve the close relation between HiFi and surrogate model which allows highly accurate reconstruction errors. Examples are given in §3.3.5. If an error controlled method is used to integrate the HiFi and surrogate models independently, the reconstruction error is affected by the approximation error of the time integration scheme.

3.3.2. Reconstruction errors in function space

We consider an implicit Euler scheme in the POD subspace V^k to solve the semi-linear PDE given in weak form as in (1.2). Therefore, we seek a sequence $y_1^k, \dots, y_{n_\tau+1}^k \in V^k$ that satisfies

$$\begin{aligned} \frac{\langle y_{n+1}^k - y_n^k, \psi \rangle_H}{\tau} &= \langle A(y_{n+1}^k, u^h(t_n)), \psi \rangle_H, \\ \langle y_1^k, \psi \rangle_H = \langle y_s, \psi \rangle_H & \quad \text{for all } \psi \in V^k, \end{aligned} \tag{3.23}$$

with a constant step size $\tau := T/n_\tau$ and $n = 1, \dots, n_\tau$. The POD subspace V^k is assumed to be given according to (3.3). The snapshots to construct the POD basis are $\{y^i\}_{i=1}^m$, where $y \in Y$ is a solution to (1.2) with $m = n_\tau + 1$ and the time grid $t_j = \tau j,\ j = 0, \dots, n_\tau$. For the next estimate we assume the initial condition $y_s \in V$ which allows solutions y to (1.2) in $C(I, V)$ and we require that the bilinear form defined in (1.3) is V-elliptic and continuous.

Theorem 3.5. *Let $y \in C(I, V)$ be a unique solution to (1.2). Assume in addition $y \in W^{2,2}(I, H) := \{\phi \in L^2(I, H) \ : \ \phi_t, \phi_{tt} \in L^2(I, H)\}$, and let $\{y_i^k\}_{i=0}^{n_\tau}$ be a unique solution of (3.23) with*

$$\max_{0 \le i \le m} \left\| y_i^k \right\|_H \le C_B$$

and a constant $C_B > 0$ independent of m. If the nonlinearity $\Theta(y)$ of problem (1.2) is Lipschitz-continuous in y and the step size τ is sufficiently small, then the estimate

$$\sum_{i=1}^m \left\| y_i^k - y(t_i) \right\|_H^2 \le \left(C_1 + C_2 m + \frac{C_3}{\tau^2} \right) \sum_{i=k+1}^d \lambda_i + C_4 \tau^2 \tag{3.24}$$

holds with constants C_1, C_2, C_3, and C_4 independent of m and k.

Proof. The proof can be found in [75, Theorem 10]. q.e.d.

We give here a slightly altered form of the estimate in [75, Theorem 10]. In the original version the estimate depends also on a term for the projection error of the initial value y_s. As described there, it can be replaced by $C_2 m \sum_{i=k+1}^d \lambda_i$. We also put in the additional term $\frac{C_3}{\tau^2} \sum_{i=k+1}^d \lambda_i$ to account for the fact that, in contrast to [75], we do not include the n_τ additional difference quotient snapshots $(y^{i+1} - y^i)/\tau,\ i = 1, \dots, n_\tau$, in the snapshot matrix (compare also [76, Theorem 4.7]). In numerical computations presented by the authors these additional snapshots yielded significant improvements only for coarse time grids. In [64] improved numerical results are reported when including finite difference quotients. Note that the difference quotients are contained in the space $\operatorname{span}\{y^0, \dots, y^{n_\tau}\}$, however, the POD basis may differ whether they are included or not. In this thesis we abstain from this inclusion. In our numerical results we never experienced the snapshot time grid to be a limiting factor for the reconstruction error.

Similar estimates are given for the explicit Euler and Crank–Nicholson scheme in [75]. In [76] the estimate is extended to general evolution problems and the restriction on using the same time grid for the implicit Euler method and the snapshots is removed. The bound in Theorem 3.5 and its extension in [76] are the most general reconstruction error estimates available for a particular time integration method.

Remark 3.6. For the practical application of POD we can conclude the following guidelines from Theorem 3.5. The reconstruction errors for POD surrogate models can assumed to be small if

1. The number of POD basis functions is sufficiently large.
2. The step size of the time integration is sufficiently small.
3. The number of POD snapshots is sufficiently large.

In the following we analyze the difference between time-continuous solutions of the HiFi and the surrogate model that also include the DEIM projection. Then we recall estimates for time-discrete solutions obtained with the implicit Euler scheme (2.8).

3.3.3. Reconstruction errors for time-continuous POD

We recall error estimates for the (PD-ROM) as given by Chaturantabut and Sorensen in [32], which correspond to the reconstruction error as defined in (3.21). In contrast to [32] we state an extension to M-orthonormal projection matrices Ψ and assume the states and adjoints as elements of Hilbert spaces with a corresponding weighted norm.

Let X and Z be Hilbert spaces. Assume X to be equipped with the norm $\|\cdot\|_X$ defined for some $x \in X$ as $\|x\|_X^2 = \langle x, x\rangle_M$, which is inherited from the Galerkin discretization. Moreover, we consider the norm $\|\cdot\|_Z$ of Z defined as

$$\|z\|_Z^2 = \langle x, x\rangle_{M^{-1}} \tag{3.25}$$

for some $z \in Z$, noting that the inverse of M is again symmetric and positive definite. Considering M as a mapping from X to Z the norms $\|\cdot\|_Z$ and $\|\cdot\|_X$ are closely related according to

$$\|Mx\|_Z = \|x\|_X \,.$$

In §4.1.1 we assume solutions z of the semi-discrete adjoint problem (2.18) to be in the space Z. Given the relation $x = \Psi\widehat{x}$ of the POD projection, we also find

$$\|x\|_X = \|\widehat{x}\| \,.$$

Note that the 2-norm for the reduced states $\widehat{x}$ is inherited from the L^2-norm in the POD subspace where the mass matrix is the identity. Further, we use the Lipschitz constant L_f to estimate the nonlinearities, defined for some $x, x' \in X$ and $F : X \times \mathbb{R}^{N_q} \to \mathbb{R}^N$ as

$$\|F(x,q) - F(x',q)\| \le L_f \|x - x'\|_X \,.$$

We use the 2-norm on the left as we do not make any assumptions on the codomain of the nonlinearity. Using a particular norm would result in a POD optimality for the DEIM basis in the corresponding norm.

The matrix norms in the subsequent are assumed to be induced by the norm of the vector space. E.g., for $S : X \to Z$ we have

$$\|S\| := \sup_{x \in X} \frac{\|Sx\|_Z}{\|x\|_X}.$$

Also we need the logarithmic norm, defined for a matrix $A \in \mathbb{R}^{N \times N}$ together with a matrix norm $\|\cdot\|$ as

$$\mu(A) := \lim_{h \to 0} \frac{\|\mathbb{I} + hA\| - 1}{h}.$$

We use the alternative formulation [108] with a compatible scalar product $\langle \cdot, \cdot \rangle$ and

$$\mu(A) = \sup_{v \in \mathbb{R}^N} \frac{\langle Av, v \rangle}{\langle v, v \rangle}. \tag{3.26}$$

Moreover, we make the following basic assumptions on the matrices Ψ and Φ.

$$\begin{aligned} &\Psi \in \mathbb{R}^{N \times k} \text{ with } \Psi^T M \Psi = \mathbb{I} \text{ , thus, } \Psi\Psi^T M \text{ is an oblique projection,} \\ &\Phi \in \mathbb{R}^{N \times \ell} \text{ with } \Phi^T \Phi = \mathbb{I} \text{ , thus, } \Phi\Phi^T \text{ is an orthogonal projection.} \end{aligned} \tag{A3.26}$$

If not indicated differently, in the reconstruction error bounds assumption (A3.26) is the only requirement on the matrices Ψ and Φ needed for the construction of the reduced-order model (PD-ROM).

Theorem 3.6. *For a fix $q \in \mathcal{Q}^h$ and $t \in I$ let $x(t)$ and $\widehat{x}(t)$ be the solutions of the HiFi problem* (2.7) *and* (PD-ROM) *respectively. Let* (A3.26) *hold. If the nonlinearity $F(x(t), q)$ is Lipschitz-continuous in x, then there is a constant $\bar{C}_x < \infty$ such that we have*

$$\int_I \|x(t) - \Psi\widehat{x}(t)\|_X^2 \, dt \le \bar{C}_x(\bar{\mathcal{E}}_x + \bar{\mathcal{E}}_{\mathrm{f}}) \tag{3.27}$$

with

$$\bar{\mathcal{E}}_x := \int_I \left\|x(t) - \Psi\Psi^T M x(t)\right\|_X^2 dt, \quad \bar{\mathcal{E}}_{\mathrm{f}} := \int_I \left\|F(x(t), q) - \Phi\Phi^T F(x(t), q)\right\|^2 dt.$$

If the projection matrices Ψ and Φ are computed according to (3.10) *and* (3.19)*, we have the bound*

$$\bar{\mathcal{E}}_x = \sum_{i=k+1}^{N} \bar{\lambda}_i, \qquad \bar{\mathcal{E}}_{\mathrm{f}} = \sum_{i=\ell+1}^{N} \bar{s}_i.$$

Proof. We extend the proof in [32, Theorem 4.2] to the case of a weighted norm $\|\cdot\|_X$ and a projection with $x = \Psi\widehat{x}$ and $\Psi M \Psi = \mathbb{I}$. Consider the transformed HiFi problem

$$\dot{\tilde{x}}(t) = SM^{-1}\tilde{x}(t) + F(M^{-1}\tilde{x}(t), q), \quad \tilde{x}(0) = Mx_0, \tag{3.28}$$

with $\tilde{x} := Mx$. We define the point wise error $e(t) = \tilde{x}(t) - M\Psi\widehat{x}(t)$ and consider

$$e(t) = \rho(t) + \theta(t),$$

where $\rho(t) := \tilde{x}(t) - M\Psi\Psi^T\tilde{x}(t)$ and $\theta(t) := M\Psi\Psi^T\tilde{x}(t) - M\Psi\widehat{x}(t)$. We now seek an estimate for $\|\dot{\theta}(t)\|$ such that we can apply Gronwall's lemma. For $\dot{\theta}(t) = M\Psi\Psi^T\dot{\tilde{x}}(t) - M\Psi\dot{\widehat{x}}(t)$ inserting (3.28) and (PD-ROM) and omitting the second argument in F, we obtain

$$\begin{aligned} \dot{\theta}(t) &= M\Psi\Psi^T \left[AM^{-1}(\tilde{x}(t) - M\Psi\widehat{x}) + F(x(t)) + \mathbb{P}F(\Psi\widehat{x}(t))\right] \\ &= M\Psi\Psi^T \left[AM^{-1}(\rho(t) - \theta(t)) + F(x(t)) - \mathbb{P}F(\Psi\widehat{x}(t))\right]. \end{aligned}$$

Defining $\widehat{\theta}(t) = \Psi^T \theta(t)$, noting that $\theta(t) = M\Psi\widehat{\theta}(t)$, we have

$$\begin{aligned}\dot{\widehat{\theta}}(t) &= \Psi^T A M^{-1}(\rho(t) - \theta(t)) + \Psi^T F(x(t)) - \Psi^T \mathbb{P} F(\Psi \widehat{x}(t)) \\ &= \widehat{S}\widehat{\theta}(t) + G(t),\end{aligned}$$

with $G(t) := \Psi^T S M^{-1}\rho(t) + \Psi^T F(x(t)) - \Psi^T \mathbb{P} F(\Psi\widehat{x}(t))$. We can estimate the norm of $G(t)$ as

$$\begin{aligned}\|G(t)\| = &\ \left\|\Psi^T S M^{-1}\rho(t) + \Psi^T F(x(t)) - \Psi^T \mathbb{P} F(x(t)) + \Psi^T \mathbb{P} F(x(t)) - \Psi^T \mathbb{P} F(\Psi\widehat{x}(t))\right\| \\ \leq &\ \left\|\Psi^T S M^{-1}\rho(t)\right\| + \left\|\Psi^T(\mathbb{I} - \mathbb{P})\mathrm{w}(t)\right\| + L_f \left\|\Psi^T \mathbb{P}\right\| \left\|x(t) - \Psi\widehat{x}(t)\right\|_X , \\ \leq &\ c_1 \left\|\rho(t)\right\|_Z + c_2 \left\|\mathrm{w}(t)\right\| + c_3 \left\|\theta(t)\right\|_Z ,\end{aligned} \tag{3.29}$$

with the Lipschitz constant $L_f < \infty$ of F, $c_1 := \left\|\Psi^T S\right\| + L_f \left\|\Psi^T \mathbb{P}\right\|$, $c_2 := \left\|\Psi^T(\mathbb{I} - \mathbb{P})\right\|$ and $c_3 := L_f \left\|\Psi^T \mathbb{P}\right\|$. To estimate the DEIM projection we apply (3.16) on $\Psi^T F(x(t)) - \Psi^T \mathbb{P} F(x(t)) = (\mathbb{I} - \mathbb{P})\mathrm{w}(t)$, where

$$\mathrm{w}(t) := F(x(t)) - \Phi\Phi^T F(x(t)).$$

Furthermore, note $\left\|M^{-1}\rho(t)\right\|_X = \left\|\rho(t)\right\|_Z$ and $\left\|x\right\|_X = \left\|\tilde{x}\right\|_Z$. With

$$\|\widehat{\theta}(t)\| \frac{d}{dt} \|\widehat{\theta}(t)\| = \frac{1}{2}\frac{d}{dt}\left(\|\widehat{\theta}(t)\|\right)^2 = \left\langle \widehat{\theta}(t), \dot{\widehat{\theta}}(t)\right\rangle$$

we get

$$\begin{aligned}\|\dot{\widehat{\theta}}(t)\| &\leq \frac{1}{\|\widehat{\theta}(t)\|}\left\langle \widehat{\theta}(t), \dot{\widehat{\theta}}(t)\right\rangle = \frac{1}{\|\widehat{\theta}(t)\|}\left\langle \widehat{\theta}(t), \widehat{S}\widehat{\theta}(t) + G(t)\right\rangle \\ &= \frac{\|\widehat{\theta}(t)\|}{\|\widehat{\theta}(t)\|^2}\left\langle \widehat{\theta}(t), \widehat{S}\widehat{\theta}(t)\right\rangle \frac{1}{\|\widehat{\theta}(t)\|}\left\langle \widehat{\theta}(t), G(t)\right\rangle \\ &= \frac{\|\widehat{\theta}(t)\|}{\|\widehat{\theta}(t)\|^2}\left\langle \widehat{\theta}(t), \widehat{S}\theta(t)\right\rangle + \frac{1}{\|\widehat{\theta}(t)\|}\left\langle \widehat{\theta}(t), G(t)\right\rangle \\ &\leq \mu(\widehat{S})\|\widehat{\theta}(t)\| + \|G(t)\| \\ &\leq a\|\widehat{\theta}(t)\| + b(t),\end{aligned}$$

where $a := \mu(\widehat{S}) + c_3$ and $b(t) := c_1 \left\|\rho(t)\right\|_Z + c_2 \left\|\mathrm{w}(t)\right\|$. In the last two inequalities we use the identity $\left\|\theta(t)\right\|_Z = \|\widehat{\theta}(t)\|$ and the property of $\mu(\widehat{S})$ in (3.26). Applying Gronwall's lemma we obtain

$$\begin{aligned}\left\|\theta(t)\right\|_Z = \|\widehat{\theta}(t)\| \leq \left\|\theta(0)\right\|_Z e^{at} + \int_0^t e^{a(t-s)b(s)} ds \\ \leq \left(\int_0^t e^{2a(t-s)} ds\right)^{\frac{1}{2}} \left(2\int_0^t c_1^2 \left\|\rho(t)\right\|_Z^2 + c_2^2 \left\|\mathrm{w}(t)\right\|^2 ds\right)^{\frac{1}{2}} ,\end{aligned}$$

using Cauchy-Schwarz and the second binomial formula. The initial value is $\theta(0) = 0$ by definition. Thus,

$$\left\|\theta(t)\right\|_Z^2 \leq \ \zeta\left(\int_I c_1^2 \left\|\rho(t)\right\|_Z^2 + c_2^2 \left\|\mathrm{w}(t)\right\|^2 ds\right),$$

with $\zeta = 2\int_I e^{2a(t-s)} ds$, which gives

$$\int_I \|\theta(t)\|_Z^2 \leq \; T\zeta \left(\int_I c_1^2 \|\rho(t)\|_Z^2 + c_2^2 \|\mathrm{w}(t)\|^2 \, ds \right).$$

Applying the last inequality to the total error $\int_I \|x(t) - \Psi\widehat{x}(t)\|_X^2 \, dt = \int_I \|e(t)\|_Z^2 \, dt$ yields the error bound (3.31) with

$$\bar{C}_x := \max\{1 + T\zeta c_1^2, \; T\zeta c_2^2\},$$

noting that $\int_I \|\rho(t)\|_Z^2 \, dt = \int_I \left\|x(t) - \Psi\Psi^T M x(t)\right\|_X^2 dt.$ q.e.d.

For (P-ROM) without DEIM projection the error estimate holds with $\bar{\mathcal{E}}_\mathrm{f} = 0$.

Remark 3.7. Due to the application of Gronwall's lemma we need the state trajectories to be continuously differentiable which we obtain from the assumption of Lipschitz-continuity on the nonlinearity. Thus, while the estimate is stated for a general $q \in \mathcal{Q}^h = L^2(I, \mathbb{R}^{n_q})$ it must be chosen such that the Lipschitz condition is satisfied.

Remark 3.8. The reconstruction error definition $RE\,(x)$ is related to estimate (3.27) when using the norm $\|x\|_{L^2(I,X)}$ via

$$\|x(t) - \Psi\widehat{x}(t)\|_{L^2(I,X)}^2 = \int_I \|x(t) - \Psi\widehat{x}(t)\|_X^2 \, dt. \tag{3.30}$$

Thus, we will refer to the estimates also simply as reconstruction errors.

Remark 3.9. From the reconstruction error also a point-wise bound for a certain time instance t can be deduced. E.g., for solutions $y^h(t) \in V^h$ at final time we find that

$$\left\|y^h(T) - y^k(T)\right\|_H^2 \to 0$$

by increasing k and ℓ. This is inherited from the assumption that the state solution space Y is continuously embedded in the space $C(I, H)$.

3.3.4. Reconstruction errors for implicit Euler time-stepping

We now consider the implicit Euler scheme as defined in (2.8) with constant step size $\tau := T/n_\tau$ and the time grid $t_j = j\tau, \; j = 0, \dots, n_\tau$. The initial value is $x^1 = x(t_0) = x_s$, thus, the sequence yields $m = n_\tau + 1$ snapshots. As for the time-continuous case, we extend the results in [32] to oblique projections and a weighted norm.

Theorem 3.7. *For a fix $q \in \mathcal{Q}^{h\tau}$, let $\{x^n\}_{n=1}^m$ and $\{\widehat{x}^n\}_{n=1}^m$ be discrete solutions of the HiFi problem* (2.7) *and* (PD-ROM) *obtained via the implicit Euler scheme* (2.8)*. Let* (A3.26) *hold. If the nonlinearity $F(x, q)$ is Lipschitz-continuous in x and the step size τ is chosen sufficiently small, then there is a constant $C_x < \infty$ such that we have*

$$\sum_{n=1}^m \|x^n - \Psi\widehat{x}^n\|_X^2 \leq C_x(\mathcal{E}_x + \mathcal{E}_\mathrm{f}) \tag{3.31}$$

with

$$\mathcal{E}_x := \sum_{n=1}^m \left\|x^n - \Psi\Psi^T M x^n\right\|_X^2, \quad \mathcal{E}_\mathrm{f} := \sum_{n=1}^m \left\|F(x^n, q) - \Phi\Phi^T F(x^n, q)\right\|^2$$

If the projection matrices Ψ *and* Φ *are computed as in* (3.6) *and* (3.17)*, we have*

$$\mathcal{E}_x = \sum_{i=k+1}^{d} \lambda_i, \qquad \mathcal{E}_{\mathrm{f}} = \sum_{i=\ell+1}^{\tilde{d}} s_i,$$

where d *and* $\tilde{d}$ *are the rank of the corresponding snapshot matrix.*

Proof. We give a proof similar to [32, Theorem 4.2], however, exploiting the presence of a linear part in the dynamic problems. Consider again the transformed HiFi model (3.28), where $\tilde{x} = Mx$. With the $\tilde{x} = M\Psi\widehat{x}$ we have

$$\frac{\tilde{x}^n - \tilde{x}^{n-1}}{\tau} = SM^{-1}\tilde{x}^n + F(x^n) \quad \text{and} \quad \frac{\widehat{x}^n - \widehat{x}^{n-1}}{\tau} = \widehat{S}\widehat{x}^n + \Psi^T \mathbb{P} F(\Psi\widehat{x}^n),$$

where we omit the q argument in F. We define $E_n := \tilde{x}^n - M\Psi\widehat{x}^n$ and consider

$$E_n = \rho_n + \theta_n,$$

where $\rho_n := \tilde{x}^n - M\Psi\Psi^T\tilde{x}^n$ and $\theta_n := M\Psi\Psi^T\tilde{x}^n - M\Psi\widehat{x}^n$. Analogously to the proof of Theorem 3.6, we seek an expression for the evolution of θ_n. Therefore, we define $\widehat{\theta}_n = \Psi^T\theta_n$ from which follows $\theta_n = M\Psi\widehat{\theta}_n$ and consider

$$\begin{aligned}
\frac{\widehat{\theta}_n - \widehat{\theta}_{n-1}}{\tau} &= \Psi^T \frac{\tilde{x}^n - \tilde{x}^{n-1}}{\tau} - \frac{\widehat{x}^n - \widehat{x}^{n-1}}{\tau} \\
&= \Psi^T SM^{-1}\tilde{x}^n + \Psi^T F(x^n) - \widehat{S}\widehat{x}^n - \Psi^T \mathbb{P} F(\Psi\widehat{x}^n) \\
&= \Psi^T SM^{-1}(\rho_n + \theta_n) + \Psi^T F(x^n) - \Psi^T \mathbb{P} F(\Psi\widehat{x}^n) \\
&= \widehat{S}\widehat{\theta}_n + G_n,
\end{aligned}$$

with $G_n := \Psi^T SM^{-1}\rho_n + \Psi^T F(x^n) - \Psi^T\mathbb{P}F(\Psi\widehat{x}^n)$. As in (3.29) we can estimate the norm of G_n via

$$\begin{aligned}
\|G_n\| \leq\; & \left\|\Psi^T SM^{-1}\rho_n\right\| + \left\|\Psi^T(\mathbb{I} - \mathbb{P})\mathrm{w}_n\right\| + L_f \left\|\Psi^T\mathbb{P}\right\| \|\tilde{x}^n - M\Psi\widehat{x}^n\|_Z, \\
\leq\; & c_1 \|\rho_n\|_Z + c_2 \|\mathrm{w}_n\| + c_3 \|\theta_n\|_Z,
\end{aligned}$$

with the Lipschitz constant L_f, $\mathrm{w}_n := F(x^n) - \Phi\Phi^T F(x^n)$, $c_1 := \left\|\Psi^T S\right\| + L_f \left\|\Psi^T\mathbb{P}\right\|$, $c_2 := \left\|\Psi^T(\mathbb{I} - \mathbb{P})\right\|$ and $c_3 := L_f \left\|\Psi^T\mathbb{P}\right\|$. Therefore, we have

$$\begin{aligned}
\frac{\|\widehat{\theta}_n\| - \|\widehat{\theta}_{n-1}\|}{\tau} &\leq \frac{1}{\tau}\left(\frac{\langle\widehat{\theta}_n, \widehat{\theta}_n\rangle}{\|\widehat{\theta}_n\|} - \frac{\langle\widehat{\theta}_n, \widehat{\theta}_{n-1}\rangle}{\|\widehat{\theta}_n\|}\right) \\
&= \frac{1}{\|\widehat{\theta}_n\|}\left\langle \widehat{\theta}_n, \frac{\widehat{\theta}_n - \widehat{\theta}_{n-1}}{\tau}\right\rangle \\
&= \frac{\|\widehat{\theta}_n\|}{\|\widehat{\theta}_n\|^2}\left\langle \widehat{\theta}_n, \widehat{S}\widehat{\theta}_n\right\rangle + \frac{1}{\|\widehat{\theta}_n\|}\left\langle \widehat{\theta}_n, G_n\right\rangle \\
&\leq \mu(\widehat{S})\|\widehat{\theta}_n\| + \|G_n\| \\
&\leq a\|\widehat{\theta}_n\| + b_n,
\end{aligned}$$

where $a := \mu(\widehat{S}) + c_3$ and $b_n := c_1 \|\rho_n\|_Z + c_2 \|\mathrm{w}_n\|$. Recall $\|\theta_n\|_Z = \|\widehat{\theta}_n\|$ and define

$\tilde{\zeta} := 1/(1-\tau a)$, which gives $\tilde{\zeta} > 0$ for τ sufficiently small. For $2 \leq n \leq m$ follows

$$\begin{aligned}\|\theta_n\|_Z &\leq \tilde{\zeta}\left(\|\theta_{n-1}\|_Z + \tau b_n\right) \leq \tilde{\zeta}^{n-1}\|\theta_1\|_Z + \tau\sum_{i=1}^{n-1}\tilde{\zeta}^i b_{n-i+1}\\ &\leq \tau\left(\zeta\sum_{i=2}^{n} b_i^2\right)^{\frac{1}{2}},\end{aligned}$$

defining $\zeta := \sum_{i=1}^{m-1}\tilde{\zeta}^{2i}$ and noting $\theta_1 = 0$. Thus,

$$\|\theta_n\|_Z^2 = \tau^2\zeta\sum_{i=2}^{n} b_i^2 \leq 2\tau^2\zeta\left(\sum_{i=2}^{n} c_1^2\|\rho_i\|_Z^2 + c_2^2\|\mathrm{w}_i\|^2\right).$$

Applying the last estimate to the total error $\sum_{n=1}^{m}\|E_n\|_Z^2 = \sum_{n=1}^{m}\left(\|\rho_n\|_Z^2 + \|\theta_n\|_Z^2\right)$ we obtain the error bound (3.31) with

$$C_x := \max\{1 + 2T\tau\zeta c_1^2,\ 2T\tau\zeta c_2^2\},$$

noting that $\sum_{n=1}^{m}\|E_n\|_Z^2 = \sum_{n=1}^{m}\|x^n - \Psi\widehat{x}^n\|_X^2$ and

$$\sum_{n=1}^{m}\|\rho_n\|_Z^2 = \sum_{n=1}^{m}\left\|x^n - M\Psi\Psi^T x^n\right\|_X^2.$$

q.e.d.

Regarding the applicability of the estimate, one must note that here we assume the implicit equations in each Euler step to hold exactly for $q \in \mathcal{Q}^{h\tau}$ and we make no assertions on how to solve these. Thus, in practice one has to take care that the accuracy with which the implicit equations are solved is sufficiently high and the control q is sufficiently regular such that the equations can be solved at all (compare Remark 3.7). It must be further noted that the constant C_x is a rough overestimate. In practice we observe often $\bar{C}_x$ to be in an order of magnitude close to 1 (compare §3.3.5).

The estimate (3.31) is in agreement with the more general result of §3.3.2, i.e., the dependence of the error on the snapshots included, the step size τ, and the POD/DEIM eigenvalues. However, note that in contrast to estimate (3.24) we lack the occurrence of the factor $1/\tau^2$ in (3.31). In Theorem 3.7 discrete solutions of the HiFi model are compared with discrete solutions of the (PD-ROM) each obtained via an identical implicit Euler scheme. In contrast, in Theorem 3.5 discrete solutions of the surrogate model are compared to continuous solutions of the HiFi model, hence, the truncation error τ^2 of the implicit Euler method must be included.

Remark 3.10. By including appropriate time weights $\gamma_1, \ldots, \gamma_m$, estimate (3.31) is an approximation of the continuous estimate (3.27). The estimate (3.31) then becomes

$$\sum_{n=1}^{m}\gamma_n\|x^n - \Psi\widehat{x}^n\|_X^2 \leq C_x(\mathcal{E}_x + \mathcal{E}_\mathrm{f})$$

with

$$\mathcal{E}_x = \sum_{n=1}^{m}\gamma_n\left\|x^n - \Psi\Psi^T M x^n\right\|_X^2, \quad \mathcal{E}_\mathrm{f} = \sum_{n=1}^{m}\gamma_n\left\|F(x^n,q) - \Phi\Phi^T F(x^n,q)\right\|^2.$$

Remark 3.11. From estimate (3.31) also an error bound for the reconstruction error of a single time instance t_i of the implicit Euler scheme can be deduced, in analogy to Remark 3.9. Obviously, we have, e.g., at final time T

$$\|x^m - \Psi \widehat{x}^m\|_X^2 \le \sum_{n=1}^{m} \|x^n - \Psi \widehat{x}^n\|_X^2 \le C_x(\mathcal{E}_x + \mathcal{E}_{\mathrm{f}}).$$

3.3.5. Examples

We now give numerical examples to provide the reader without POD experience with an idea of common practical approximation quality. The results reflect the theoretical reconstruction results of this section and serve as a basis for further investigations when we dedicate ourselves to derivative approximations. One must note that the reconstruction errors become zero if the number of POD and DEIM basis functions is N which is the dimension of the HiFi model. In this case the model reduction step via POD (without DEIM) reduces to a basis transformation. However, to efficiently apply POD we hope for the number of necessary basis functions to be small. The software setting where the computations are carried out is described in chapter 6.

We consider the following convection-diffusion-reaction example problem on the unit square $\Omega = (0,1)^2$, with time horizon $I = [0, 0.5]$, and mixed boundary conditions of Robin and homogeneous Neumann type

$$\begin{aligned} y_t = -a^T \nabla y + D\Delta y + \Theta(y, u_1, u_2) + u_3 \qquad & \text{in } I \times \Omega, \\ \partial_\nu y + \beta_1 y = \beta_2 \qquad & \text{on } I \times \Gamma_1, \\ \partial_\nu y = 0 \qquad & \text{on } I \times \Gamma_2, \\ y(0) = 0 \qquad & \text{on } \Omega. \end{aligned} \tag{3.32}$$

The boundary is defined by the constant velocity vector $a \in \mathbb{R}^2$ and

$$\Gamma_1 = \{r \in \partial\Omega \ : \ a^T \mathbf{n}(r) < 0\} \quad \text{and} \quad \Gamma_2 = \{r \ : \ a^T \mathbf{n}(r) \ge 0\},$$

where $\mathbf{n}(r)$ is the outer normal vector at $r \in \partial\Omega$. In the following we distinguish two particular cases.

Case A: In a convection dominated setting with use

$$a = (0.2, -1)^T, \quad D = 0.02, \quad \beta_1 = \beta_2 = 10^3,$$

the nonlinearity

$$\Theta(y, u_1, u_2) = e^{u_2 \sin(y r_1)} - (\cos(\pi u_2) + u_2 y^2),$$

where $u_1, u_2 \in \mathbb{R}$, and a spatially distributed control $u_3 = q_3 \phi_S(r)$ with shape function $\phi_S(r)$ given as

$$\phi_S(r) := \max(0, -0.25 + e^{-25((r_1 - 0.25)^2 + (r_2 - 0.25)^2)}). \tag{3.33}$$

An illustration of the shape function ϕ_S is given in Figure 3.1.

Case B: In a diffusion dominated setting with use

$$\begin{gathered} a = (0.004, -0.02)^T, \quad D = 1, \quad \beta_1 = \beta_2 = 10^2, \\ \Theta(y, u_1, u_2) = 0.01 e^{u_2 \sin(y r_1)} - 0.01(\cos(\pi u_2) + u_2 y^2), \end{gathered}$$

and $u_3 = q_3 \phi_S(r)$ as above.

We choose different values in the two cases, such that for case B we can expect better reconstruction properties in comparison to case A. The choices are based on practical experience with POD. Roughly spoken, we decrease the nonlinearity and stiffness as well as the amount of energy contained in the system. Later we are also interested in adjoints and sensitivities with respect to the controls. For the state reconstruction tests we set $u_1 = u_2 = 1$ and $q_3 = -1$. Note that after discretization we have $u_1 = q_1$ and $u_2 = q_2$.

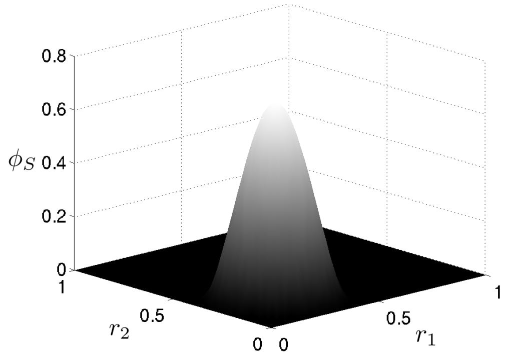

Figure 3.1.: Shape function ϕ_S defining u_2.

To discretize the problem in space we employed linear finite elements with $N = 289$ based on the software deal-ii [11]. The time integration is carried out with DAESOL-II. For the reconstruction tests we adapted the configurations such that we do an implicit Euler method with fixed step size $\tau = 10^{-3}$, a decomposition of the Newton iteration matrix in each step, and one Newton iteration per time step. This allows for small reconstruction errors close to machine precision in case B. Note that the discretization parameters are tailored to show effects of POD model reduction in a discrete setting, thus, temporal discretization quality is chosen higher than spatial discretization accuracy.

The reduced-order models are constructed according to §3.1.3 and 3.2.1. The projection matrices are determined via the optimality (3.6) for the POD projection and (3.17) for the DEIM projection, however, we use an additional normalization of the snapshots. We divide each POD snapshot by $\sum_{l=1}^{m} \gamma_l \left\|x^l\right\|_X$ and each DEIM snapshot by $\sum_{l=1}^{m} \gamma_l \left\|F(x^l, q)\right\|$. Normalization issues are discussed in detail the next chapter. We apply Variant I to compute the eigenvectors of the snapshot matrices. Thus, using SVD, which guarantees an accurate POD basis approximation close to machine precision. Information is taken from all time instances, thus, the snapshot matrices are $\mathcal{S}, \mathcal{D} \in \mathbb{R}^{289\times 501}$. In the following the reconstruction errors $RE\left(x\right)$ are compared using the norm $\|x\|_{L^2(I,X)}$ which corresponds to the square root of the reconstruction error estimates in Theorem 3.7 (see also Remark 3.8). The error is evaluated using a quadrature rule implied by the POD weights γ_i as in (3.2).

Figure 3.2 shows the reconstruction results of the example problem (3.32) for the cases A and B. We increased the number of POD basis functions used to build (P-ROM) and computed the reconstruction error $RE\left(x\right)$ for each instance. In addition the square root over the sum of neglected eigenvalues is plotted, which for a purely state dependent POD basis corresponds to the square root of the discrete POD projection error $\mathcal{E}_x$ according to (3.9).

We observe an exponential decay in the reconstruction and the projection error in both cases. The fast decay of eigenvalues is typical and essential for a successful application of POD. Furthermore, we see that the projection error and the reconstruction error are closely related which is expected due to the reconstruction results of §3.3.4. This is a behavior

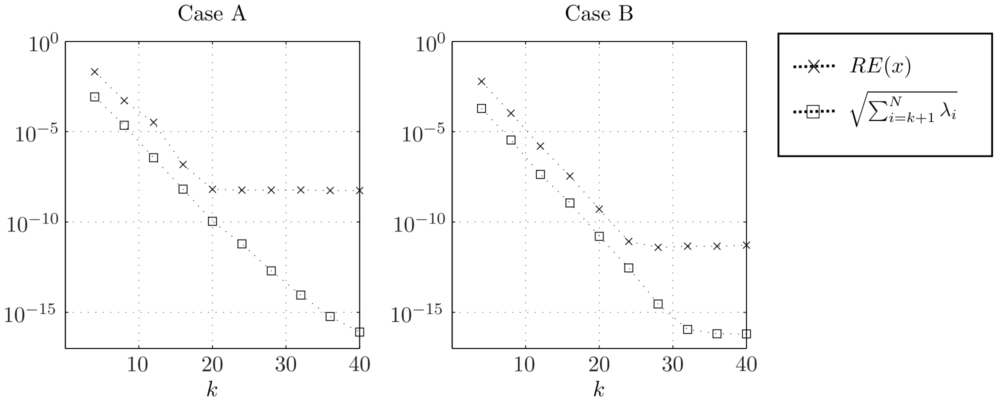

Figure 3.2.: POD reconstruction errors $RE(x)$ using the norm $\|x\|_{L^2(I,X)}$ and the square root of the projection error $\mathcal{E}_x$ corresponding to the neglected eigenvalues λ_i.

we observe for all tested examples. While for the general reconstruction result of §3.3.2 the constants $C_1, \ldots, C_4$ are independent of k, the constant C_x in (3.31) is technically not. However, the numerical results indicate that the value of C_x is also independent of the number of POD basis functions. Moreover, we see that the decay of the reconstruction error stops after a certain value of k. Recalling the main influence factors for POD reconstruction in Remark 3.6, we observe that the POD projection error dictates the reconstruction error for small POD bases and for larger bases the time discretization error becomes dominant. The number of snapshots included is obviously not a factor as all available snapshots have been used. We observed that the errors can be driven close to machine precision by choosing τ sufficiently small, which is not explicitly shown here. Moreover, as expected the error decay is faster in case B where also the time discretization error starts to dominate not before a reconstruction error of about $\approx 10^{-12}$ in contrast to about 10^{-8} in case A. Note that high accuracies can be obtained as the reduced-order models are an exact (up to the precision of Ψ) projection of the HiFi model.

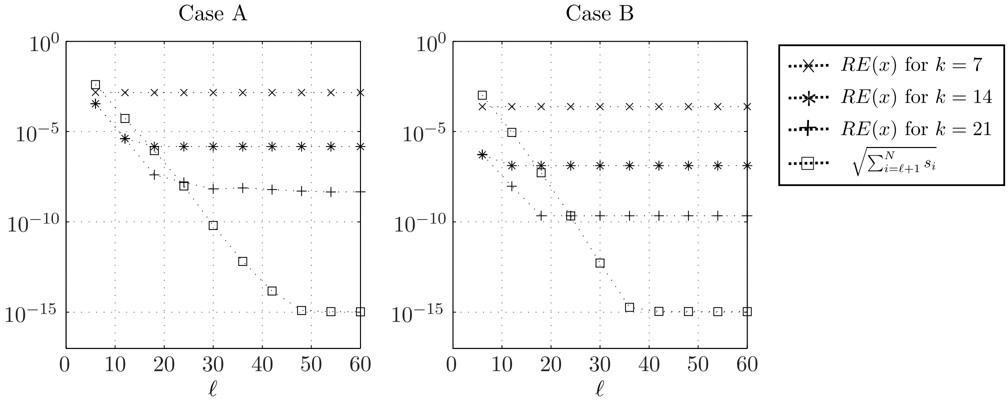

Figure 3.3.: Decay of the DEIM reconstruction errors for different instances of POD reduced-order models for the norm $\left(\int_I \|x\|_X^2 \, dt\right)^{\frac{1}{2}}$. In addition the the DEIM projection error is plotted determined via the neglected eigenvalues s_i.

In Figure 3.3 we display analogous reconstruction results of the DEIM approximation for both example cases and the same norm. In addition we distinguish different sizes of POD reduced-order models that the DEIM approximation is applied to, i.e., for $k = 12, 24, 36$. As before we observe an exponential decay of eigenvalues and reconstruction errors. However, with a certain size of ℓ the reconstruction cannot be improved further, even for relatively large errors. Obviously, this is due to the fact that we achieved the best possible approximation quality determined by the size of k (compare to the corresponding value of k in Figure 3.2). From a practical point of view, it is desirable to choose ℓ such that the best possible reconstruction error of a (P-ROM) can be achieved.

Remark 3.12. Note that we cannot expect to reduce the error exactly until the POD reconstruction limit as the DEIM space is constructed from HiFi data and the nonlinear evaluations are done with surrogate data. To remove this discrepancy one would have to construct the DEIM space from data of a pure POD surrogate, i.e., using snapshots

$$F(\Psi\widehat{x}^1, q), \ldots, F(\Psi\widehat{x}^m, q).$$

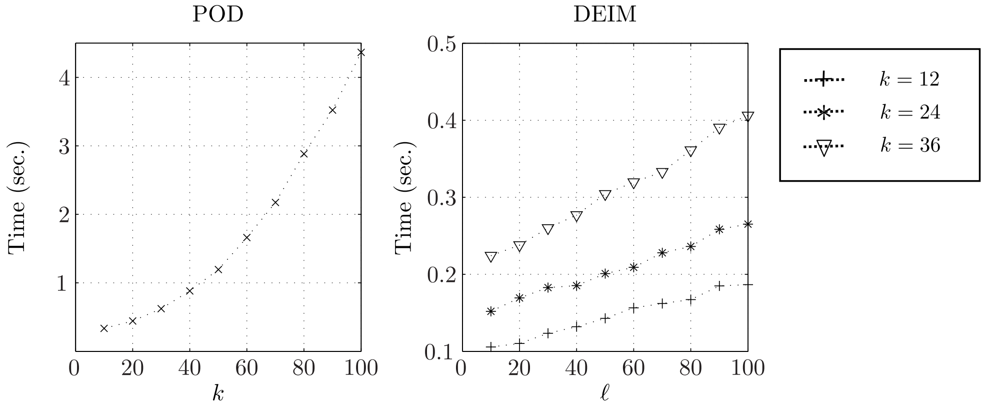

Figure 3.4.: Runtimes for simulations of (P-ROM) (left) and (PD-ROM) for different instances of POD reduced-oder models (right). Runtime for a HiFi simulation is 2.5 seconds.

In Figure 3.4 we show runtimes for a single simulation of the reduced-order models (P-ROM) (left) and (PD-ROM) (right) for case A. The software environment and technical aspects are described in chapter 6. The simulation time for the HiFi model was approximately 2.5 seconds. As one would expect, with increasing number of basis functions the runtime increases. In case of POD the evaluation of the reduced-order model for $k = 100$ is almost as expensive as the HiFi model, even though a still significantly smaller system is used. One must recall that the POD basis uses global modes, thus, dense linear algebra and matrix multiplications must be used. Also the evaluation of the nonlinearities is still as expensive as for the HiFi model. Hence, we can conclude that the efficiency of the POD approach gets lost when the basis is not able to capture the essential information with a moderate number of modes. With DEIM the runtimes can be further reduced. Also for relatively large numbers of DEIM basis functions, provided that the POD basis is moderate, the evaluations are cheap. This will be important as later when including adjoint derivatives we need a lot more information in the DEIM subspace. Note that by applying DEIM, the cost of the reduced-order model is completely independent of the size of the HiFi model. Hence, the same runtimes are obtained even for HiFi models with much finer discretization.

4. Derivative-Extended Reduced-Order Models

In the direct approach we follow we want to use the POD/DEIM surrogate models (P-ROM) and (PD-ROM) to solve the reduced optimal control problem (2.17) of the semi-discrete setting, applying the concepts for derivative computation as explained in §2.5. To this end, it is desirable to compute good POD/DEIM approximations of the reduced objective and its gradient. While it is challenging to assure this for a fixed surrogate model and all q (the problem of POD prediction), we should at least guarantee small approximation errors locally, i.e., small reconstruction errors $RE\left(j^h(\bar{q})\right)$ and $RE\left(\nabla j^h(\bar{q})\right)$ for a reference $\bar{q}$ where the surrogate model is constructed.

The reconstruction properties of the objective are inherited from the reconstruction properties of the states. We can use the estimates in §3.3 to find error bounds for the objective, depending on the setting we consider. As we assume the objective to depend on the state at final time only we obtain from Theorem 3.6 and Remark 3.9 that

$$RE\left(j^h(q)\right) \leq \bar{C}_J(\bar{\mathcal{E}}_x + \bar{\mathcal{E}}_{\mathrm{f}}) = \bar{C}_J\left(\sum_{i=k+1}^{N} \bar{\lambda}_i + \sum_{i=\ell+1}^{N} \bar{s}_i\right), \tag{4.1}$$

for some constant $\bar{C}_J < \infty$. The equation in (4.1) holds for POD and DEIM bases built according to (3.10) and (3.19). For the discrete setting, using the implicit Euler scheme (2.8) to solve HiFi and surrogate model, we obtain due to Theorem 3.7 and Remark 4.2 that

$$RE\left(j^{h\tau}(q)\right) \leq C_J(\mathcal{E}_x + \mathcal{E}_{\mathrm{f}}) = C_J\left(\sum_{i=k+1}^{d} \lambda_i + \sum_{i=\ell+1}^{\tilde{d}} s_i\right), \tag{4.2}$$

for some constant $C_J < \infty$.

However, the reconstruction properties for the adjoints and the sensitivity equations are yet unclear and with this the reconstruction properties of the gradient. In this chapter we shed light on the relations between the HiFi model, the reduced-order models (P-ROM) and (PD-ROM), and their corresponding adjoint and sensitivity problems. The idea we follow is to include additional derivative information in the snapshot matrix for POD projection as well as for the DEIM projection. The forward case for pure (P-ROM) without DEIM projection was presented in [104]. We extend the ideas to the adjoint case and to surrogate models with DEIM projection. Note that we have in mind the overall goal to use the adjoint approach to solve optimal control problems while with the sensitivity approach we seek to solve parameter estimation problems, where the number of degrees of freedom for the optimization variable is rather small to moderate.

4.1. Reconstruction error estimates

4.1.1. Adjoint equations

We extend the estimates in §3.3 to the adjoint problem of the HiFi model (2.7), which contain, due to the nonlinearity, solutions of the state equation. Analogous to (2.18) the HiFi adjoint is obtained as solution of

$$-\dot{z}^T(t) = z^T(t)M^{-1}(S + F_x(x(t), q)), \quad z^T(T) = J^h_x(x, q). \tag{4.3}$$

An overview of the relations between the models and their adjoints is given in Figure 4.1. By 'discretize' we mean the Galerkin discretization as in §2.2.1, by 'adjoining' the derivation of the adjoint, and by POD and DEIM projection we refer to the construction of the surrogate models as explained in §3.1.3 and §3.2.1. Recalling Remark 2.4 the adjoint problems are defined by the corresponding state problems. With the direct approach we seek to construct the reduced problems (P-ROM) and (PD-ROM) such that their semi-discrete adjoint problems are POD/DEIM projections of the HiFi adjoint problem in the sense of POD/DEIM model reduction as described in Chapter 3. Thus, the question arises what the projections look like and how the subspaces must be constructed.

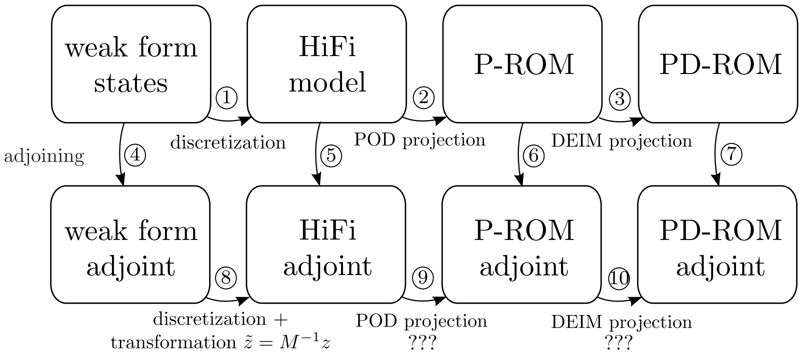

Figure 4.1.: Overview of the relations between infinite-dimensional problems in weak form, semi-discrete models, POD reduced-order models, and POD/DEIM reduced-order models for states and adjoints.

We start by analyzing under which conditions the diagram in Figure 4.1 is commutative. Commutativity of the left part (path ①+⑤ = path ④+⑧) was shown in §2.3, Proposition 2.1. Following the techniques for the derivation of the adjoint in the semi-discrete case as in §2.3, the adjoint of (P-ROM) together with the objective $J^h(\Psi\widehat{x}, q)$ is

$$-\dot{\widehat{z}}^T(t) = \widehat{z}^T(t)\left(\widehat{S} + \Psi^T F_{\widehat{x}}(\Psi\widehat{x}(t), q)\right), \quad \widehat{z}(T)^T = J^h_{\widehat{x}}(\Psi\widehat{x}, q). \tag{4.4}$$

For now we do not make any assumptions on the subspaces that are used for the POD/DEIM and consider only the formal projection step. Under the basic assumption (A3.26) on the projection matrices, the following results are obtained immediately.

Lemma 4.1. *If the relation $z = M\Psi\widehat{z}$ is used for the POD projection of the HiFi adjoint equation* (4.3)*, then the paths ②+⑥ and ⑤+⑨ yield the same result.*

The adjoint problem of (PD-ROM) is analogously given as

$$-\dot{\widehat{z}}^T(t) = \widehat{z}^T(t)\left(\widehat{S} + \Psi^T \mathbb{P} F_{\widehat{x}}(\Psi\widehat{x}(t), q)\right), \quad \widehat{z}(T)^T = J^h_{\widehat{x}}(\Psi\widehat{x}, q). \tag{4.5}$$

Lemma 4.2. *If the DEIM projection is applied simultaneously on all k columns of $F_{\widehat{x}}(\Psi\widehat{x}(t), q)$ in* (4.4)*, then the paths ③+⑦ and ⑥+⑩ yield the same result.*

Commutativity of the diagram follows then directly from Lemmas 4.1 and 4.2. Note that it is not a priorily clear how to do the DEIM projection. We will discuss this issue later in §5.4. Now we present error estimates for solutions of the HiFi and the (PD-ROM) adjoint. Recall the definition of the norm $\|\cdot\|_Z = \langle\cdot,\cdot\rangle_{M^{-1}}$ in (3.25).

Theorem 4.3. *For a fix $q \in \mathcal{Q}^h$ and $t \in I$ let $x(t)$ and $z(t)$ be the solutions of the state* (2.7) *and adjoint problem* (4.3) *and $\widehat{x}(t)$ and $\widehat{z}(t)$ the solutions of* (PD-ROM) *and its adjoint* *problem* (4.5) *respectively. Let* (A3.26) *hold. Assume the matrix of directional derivatives* $F_x(x(t), q)\Psi$ *and* $F(x(t), q)$ *to be Lipschitz-continuous in* x*. If the solution* $z(t)$ *is bounded with* $z_{max} = \sup_{t\in I} \|z(t)\|_Z$*, then there is a constant* $\bar{C}_z < \infty$ *such that we have*

$$\int_I \|z(t) - M\Psi\widehat{z}(t)\|_Z^2 \, dt \le \bar{C}_z \left(\bar{\mathcal{E}}_x + \bar{\mathcal{E}}_{\mathrm{f}} + \bar{\mathcal{E}}_z + \sum_{i=1}^{k} \bar{\mathcal{E}}_{\mathrm{fz}}^i \right) \tag{4.6}$$

with $\bar{\mathcal{E}}_x$ *and* $\bar{\mathcal{E}}_{\mathrm{f}}$ *as in Theorem 3.6 and*

$$\begin{aligned} \bar{\mathcal{E}}_z &:= \int_I \left\| z(t) - M\Psi\Psi^T z(t) \right\|_Z^2 dt, \\ \bar{\mathcal{E}}_{\mathrm{fz}}^i &:= \int_I \left\| F_x(x(t), q)\Psi_{\cdot i} - \Phi\Phi^T F_x(x(t), q)\Psi_{\cdot i} \right\|^2 dt, \quad i = 1, \ldots, k. \end{aligned}$$

Note that $\|\cdot\|_Z$ is the natural norm inherited from the Galerkin discretization of the continuous adjoint (1.15) in the space V^h and the relation $\tilde{z} = M^{-1}z$ established in Proposition 2.1.

Proof. Similar to the proof in Theorem 3.6 we decompose the point wise error $e(t) := z(t) - M\Psi\widehat{z}(t)$ into

$$e(t) = \rho(t) + \theta(t)$$

with $\rho(t) := z(t) - M\Psi\Psi^T z(t)$ and $\theta(t) := M\Psi\Psi^T z(t) - M\Psi\widehat{z}(t)$. Again omitting the control argument q in F_x we have

$$\dot{\theta}(t) = M\Psi\Psi^T \left[S^T M^{-1} (\rho(t) + \theta(t)) + F_x(x(t))^T z(t) - (\mathbb{P}F_x(\Psi\widehat{x}(t)))^T \, \Psi\widehat{z}(t) \right]$$

where we used $F_x(\Psi\widehat{x}(t))\Psi = F_{\widehat{x}}(\Psi\widehat{x}(t))$. Defining $\widehat{\theta}(t) := \Psi^T\theta(t)$ and multiplying the expression for $\dot{\theta}(t)$ from the left with Ψ^T, one obtains

$$\dot{\widehat{\theta}}(t) = \widehat{S}\widehat{\theta}(t) + G(t)$$

with

$$G(t) := \Psi^T \left[S^T M^{-1}\rho(t) + F_x(x(t))^T M^{-1} z(t) - (\mathbb{P}F_x(\Psi\widehat{x}(t)))^T \, \Psi\widehat{z}(t) \right].$$

Using (3.16) with $F_x(x(t))\Psi - \mathbb{P}F_x(x(t))\Psi = (\mathbb{I} - \mathbb{P})\mathrm{w}_z(t)$ and $\mathrm{w}_z(t) := F_x(x(t))\Psi - \Phi\Phi^T F_x(x(t))\Psi$, the norm of $G(t)$ is bounded by

$$\begin{aligned} \|G(t)\| \le\ & \big\| \Psi^T S^T M^{-1}\rho(t) + \big[(F_x(x(t))\Psi)^T - (\mathbb{P}F_x(x(t))\Psi)^T \\ & + (\mathbb{P}F_x(x(t))\Psi)^T - (\mathbb{P}F_x(\Psi\widehat{x}(t))\Psi)^T \big] M^{-1} z(t) \\ & + (\mathbb{P}F_x(\Psi\widehat{x}(t))\Psi)^T M^{-1} (z(t) - M\Psi\widehat{z}(t)) \big\| \\ \le\ & \left\| \Psi^T S^T M^{-1}\rho(t) + (\mathbb{P}F_x(\Psi\widehat{x}(t))\Psi)^T M^{-1} (\rho(t) + \theta(t)) \right\| \\ +\ & \left\| z(t)^T M^{-1} (\mathbb{I} - \mathbb{P})\mathrm{w}_z(t) \right\| + \left\| z^T(t) M^{-1} \mathbb{P} \left(F_x(x(t))\Psi - F_x(\Psi\widehat{x}(t))\Psi \right) \right\| \\ \le\ & c_1 \|\rho(t)\|_Z + c_2 \|\theta(t)\|_Z + c_3 \|\mathrm{w}_z(t)\| + c_4 \|x(t) - \Psi\widehat{x}(t)\|_X \end{aligned} \tag{4.7}$$

with constants $c_1 := \|S\Psi\| + F_{max}\|\mathbb{P}\|$, $c_2 := F_{max}\|\mathbb{P}\|$, $c_3 := z_{max}\|(\mathbb{I} - \mathbb{P})\|$, $c_4 := z_{max}L_f\|\mathbb{P}\|$, where $F_{max} = \sup_{t\in I}\|F_x(x(t))\Psi\|$ and L_f the Lipschitz constant defined as

$$\|(F_x(x(t))\Psi - F_x(\Psi\widehat{x}(t))\Psi)\| \le L_f \|x(t) - \Psi\widehat{x}(t)\|_X.$$

Note $\|M^{-1}z\|_X = \|z\|_Z$ and accordingly for ρ and θ. For the temporal evolution of $\|\widehat{\theta}\|$ we obtain as before

$$\begin{aligned}\|\dot{\widehat{\theta}}(t)\| &\leq \mu(\widehat{S})\|\widehat{\theta}(t)\| + \|G(t)\| \\ &\leq a\|\widehat{\theta}(t)\| + b(t)\end{aligned}$$

using the logarithmic norm $\mu(\cdot)$ and $a := \mu(\widehat{S}) + c_2,\quad b(t) := c_1 \|\rho(t)\|_Z + c_3 \|\mathrm{w}_z(t)\| + c_4 \|x(t) - \Psi\widehat{x}(t)\|_X$. From here we continue as in the proof to Theorem 3.6, applying Gronwall's lemma to the last inequality. q.e.d.

We now state the results for the adjoint problem solved with the implicit Euler scheme (2.8). We use again the abbreviation $z^j := z(t_{j-1})$ as for the states.

Theorem 4.4. *For a fix $q \in \mathcal{Q}^{h\tau}$ let $\{x^n\}_{n=1}^m$ and $\{z^n\}_{n=1}^m$ be solutions of* (2.7) *and* (4.3) *obtained via the implicit Euler scheme* (2.8). *Let $\{\widehat{x}^n\}_{n=1}^m$ and $\{\widehat{z}^n\}_{n=1}^m$ be solutions of* (PD-ROM) *and* (4.5) *respectively. Let* (A3.26) *hold. Assume the matrix of directional derivatives $F_x(x,q)\Psi$ and $F(x,q)$ to be Lipschitz-continuous in x. If $z_{max} = \max\{z^1,\dots,z^m\} < \infty$ and τ is chosen sufficiently small, then there is a constant $C_z < \infty$ such that we have*

$$\sum_{n=1}^{m} \gamma_n \|z^n - M\Psi\widehat{z}^n\|_Z^2 \leq C_z \left(\mathcal{E}_x + \mathcal{E}_\mathrm{f} + \mathcal{E}_z + \sum_{i=1}^{k} \mathcal{E}_{\mathrm{f}z}^i\right) \tag{4.8}$$

with $\mathcal{E}_x$ and $\mathcal{E}_\mathrm{f}$ as in Remark 3.10 and

$$\mathcal{E}_z := \sum_{n=1}^{m} \gamma_n \left\|z^n - \Psi\Psi^T M z^n\right\|_Z^2,$$

$$\mathcal{E}_{\mathrm{f}z}^i := \sum_{n=1}^{m} \gamma_n \left\|F_z(x^n,q)\Psi_{\cdot i} - \Phi\Phi^T F_z(x^n,q)\Psi_{\cdot i}\right\|^2, \quad i = 1,\dots,k.$$

Proof. The proof is similar to Theorem 3.7 using a bound for G_n analogous to estimate (4.7) in Theorem 4.3 and noting that z_n are in reverse order due to backward integration in time. q.e.d.

Clearly, both theorems also hold for the adjoint problem (4.4) of (P-ROM) with $\bar{\mathcal{E}}_\mathrm{f} = \bar{\mathcal{E}}_{\mathrm{f}z}^i = \mathcal{E}_\mathrm{f} = \mathcal{E}_{\mathrm{f}z}^i = 0$.

4.1.2. Sensitivity equations

In this section errors between solutions of the sensitivity equations and their surrogate approximations are assessed. The VDE to the HiFi problem (2.7) in a direction $\tilde{q}$ is given as

$$\dot{w}(t) = (S + F_x(x(t),q))\, w(t) + F_q(x(t),q)\tilde{q}, \quad w(0) = 0. \tag{4.9}$$

The relation of the HiFi and surrogate models in a sensitivity based approach are shown in Figure 4.2 in analogy to the adjoint case. The VDE of (P-ROM) according to §2.3 is given as

$$\dot{\widehat{w}}(t) = \left(\widehat{S} + \Psi^T F_{\widehat{x}}(\Psi\widehat{x}(t),q)\right) \widehat{w}(t) + \Psi^T F_q(\Psi\widehat{x}(t),q)\tilde{q}, \quad \widehat{w}(0) = 0, \tag{4.10}$$

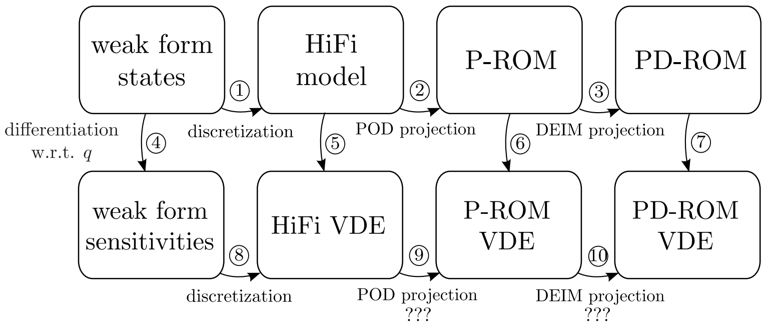

Figure 4.2.: Overview of the relations between infinite-dimensional problems in weak form, semi-discrete models, POD reduced-order models, and POD/DEIM reduced-order models for states and sensitivity equation/VDE.

with the VDE state variable $\widehat{w} = \widehat{x}_q \tilde{q} \in \mathbb{R}^k$ and a direction $\tilde{q} \in \mathcal{Q}^h$. The VDE of the (PD-ROM) is

$$\dot{\widehat{w}}(t) = \left(\widehat{S} + \Psi^T \mathbb{P} F_{\widehat{x}}(\Psi \widehat{x}(t), q)\right) \widehat{w}(t) + \Psi^T \mathbb{P} F_q(\Psi \widehat{x}(t), q)\tilde{q}, \quad \widehat{w}(0) = 0. \tag{4.11}$$

As before we consider only the formal projection step and require the projection matrices Ψ, Φ to only satisfy (A3.26). Commutation of the left part is a result of §2.3.2. Commutativity of the diagram is then a result of the following two Lemmata. Their proofs are obtained via standard calculus.

Lemma 4.5. *Applying a POD projection on* (4.9) *with the relation $w = \Psi \widehat{w}$, then the path ②+⑥ yields the same result as ⑤+⑨.*

Lemma 4.6. *Applying a DEIM projection on the sum $F_{\widehat{x}}(\Psi \widehat{x}(t), q)\widehat{w}(t) + F_q(\Psi \widehat{x}(t), q)\tilde{q}$ in* (4.10)*, then the path ③+⑦ yields the same result as ⑥+⑩.*

We now give estimates for the error between solutions to (4.9) and (4.11). For fix $q, \tilde{q} \in \mathcal{Q}^h$ in the following we use the abbreviation

$$\tilde{F}(x, w) := F_x(x, q)w + F_q(x, q)\tilde{q}, \quad \forall x, w \in X. \tag{4.12}$$

Theorem 4.7. *For fix $q, \tilde{q} \in \mathcal{Q}^h$ and $t \in I$ let $x(t)$ and $w(t)$ be the solutions of the state* (2.7) *and VDE problem* (4.9) *and let $\widehat{x}(t)$ and $\widehat{w}(t)$ be the solutions of the* (PD-ROM) *and its VDE* (4.11) *respectively. Let* (A3.26) *hold. If $F_x(x(t), q)w(t)$, $F_q(x(t), q)\tilde{q}$, and $F(x(t), q)$ are Lipschitz-continuous in x, then there is a constant $\bar{C}_w < \infty$ such that we have*

$$\int_I \|w(t) - \Psi \widehat{w}(t)\|_X^2 \, dt \leq \bar{C}_w (\bar{\mathcal{E}}_x + \bar{\mathcal{E}}_{\mathrm{f}} + \bar{\mathcal{E}}_w + \bar{\mathcal{E}}_{\mathrm{f}w}) \tag{4.13}$$

with $\bar{\mathcal{E}}_x$ and $\bar{\mathcal{E}}_{\mathrm{f}}$ as in Theorem 3.6 and

$$\bar{\mathcal{E}}_w := \int_I \left\| w(t) - \Psi \Psi^T M w(t) \right\|_X^2 dt,$$
$$\bar{\mathcal{E}}_{\mathrm{f}w} := \int_I \left\| \tilde{F}(x(t), w(t)) - \Phi \Phi^T \tilde{F}(x(t), w(t)) \right\|^2 dt.$$

Note that $\|\cdot\|_X$ is the natural norm for solutions of (4.9) inherited from the discretization of the weak form of the sensitivity problem (1.20).

Proof. The proof is analogous to Theorem 3.6, considering a transformed HiFi VDE

$$\dot{\tilde{w}}(t) = (S + F_x(x(t), q))\, M^{-1}\tilde{w}(t) + F_q(x(t), q)\tilde{q}, \quad w(0) = 0,$$

with $\tilde{w} = Mw$ and seeking a bound for the error $e(t) = \rho(t) + \theta(t)$, where $\rho(t) := \tilde{w}(t) - M\Psi\Psi^T\tilde{w}(t)$ and $\theta(t) := M\Psi\Psi^T\tilde{w}(t) - M\Psi\widehat{w}(t)$. The crucial step is to find an estimate for $\|G(t)\|$ in the temporal evolution of $\widehat{\theta}(t) = \Psi^T\theta(t)$, given as

$$\dot{\widehat{\theta}}(t) = \widehat{S}\widehat{\theta}(t) + G(t). \tag{4.14}$$

With

$$G(t) = \Psi^T SM^{-1}\rho(t) + \Psi^T\tilde{F}(x(t), w(t)) - \Psi^T\mathbb{P}\tilde{F}(\Psi\widehat{x}(t), \Psi\widehat{w}(t))$$

a bound is given by

$$\|G(t)\| \leq \left\|\Psi^T SM^{-1}\rho(t)\right\| + \left\|\Psi^T\tilde{F}(x(t), w(t)) - \Psi^T\mathbb{P}\tilde{F}(\Psi\widehat{x}(t), \Psi\widehat{w}(t))\right\| \tag{4.15}$$

where the second term is bounded by

$$\begin{aligned}
&\left\|\Psi^T\tilde{F}(x(t), w(t)) - \Psi^T\mathbb{P}\tilde{F}(\Psi\widehat{x}(t), \Psi\widehat{w}(t))\right\| \\
&\leq \left\|\Psi^T\tilde{F}(x(t), w(t)) - \Psi^T\mathbb{P}\tilde{F}(x(t), w(t))\right\| \\
&\quad + \left\|\Psi^T\mathbb{P}\tilde{F}(x(t), w(t)) - \Psi^T\mathbb{P}\tilde{F}(\Psi\widehat{x}(t), w(t))\right\| \\
&\quad + \left\|\Psi^T\mathbb{P}\tilde{F}(\Psi\widehat{x}(t), w(t)) - \Psi^T\mathbb{P}\tilde{F}(\Psi\widehat{x}(t), \Psi\widehat{w}(t))\right\| \\
&\leq \left\|\Psi^T(\mathbb{I} - \mathbb{P})\right\| \|\tilde{\mathrm{w}}(t)\| + L_f \left\|\Psi^T\mathbb{P}\right\| \|x(t) - \Psi\widehat{x}(t)\|_X \\
&\quad + F_{max} \left\|\Psi^T\mathbb{P}\right\| \|\tilde{w}(t) - M\Psi\widehat{w}(t)\|_Z\,,
\end{aligned}$$

with

$$\mathrm{w}_w(t) := \tilde{F}_x(x(t), w(t)) - \Phi\Phi^T\tilde{F}_x(x(t), w(t)),$$

$F_{max} := \max_{t\in I}\left(\left\|F_x(\Psi\widehat{x}(t), q)M^{-1}\right\|\right)$, and L_f the Lipschitz constant defined as

$$\left\|\tilde{F}_x(x(t), w(t)) - \tilde{F}_x(\Psi\widehat{x}(t), w(t))\right\| \leq L_f \|x(t) - \Psi\widehat{x}(t)\|_X\,,$$

which exists to the Lipschitz continuity of $F_x(x(t), q)w(t)$ and $F_q(x(t), q)\tilde{q}$. Using (4.14) we can proceed as in the proof of Theorem 3.6. q.e.d.

For the time-discrete case we use again the notation $w^j := w(t_{j-1}),\ j = 1, \dots, m$.

Theorem 4.8. *For fix $q, \tilde{q} \in \mathcal{Q}^{h\tau}$ let $\{x^n\}_{n=1}^m$ and $\{w^n\}_{n=1}^m$ be solutions of* (2.7) *and* (4.9) *obtained via the implicit Euler scheme* (2.8)*. Let $\{\widehat{x}^n\}_{n=1}^m$ and $\{\widehat{w}^n\}_{n=1}^m$ be solutions of* (PD-ROM) *and* (4.11) *respectively. Assume* (A3.26) *to hold. If $F_x(x^i, q)w^i$, $F_q(x^i, q)\tilde{q}$, and $F(x^i, q)$, $i = 1, \dots, m$, are Lipschitz-continuous in x, then there is a constant $C_w < \infty$ such that we have*

$$\sum_{n=1}^{m} \gamma_n \|w^n - \Psi w^n\|_X^2 \leq C_w(\mathcal{E}_x + \mathcal{E}_\mathrm{f} + \mathcal{E}_w + \mathcal{E}_\mathrm{fw}) \tag{4.16}$$

with $\mathcal{E}_x$ and $\mathcal{E}_\mathrm{f}$ as in Remark 3.10 and

$$\begin{aligned}
\mathcal{E}_w &:= \sum_{n=1}^{m} \gamma_n \left\|w^n - \Psi\Psi^T Mw^n\right\|_X^2\,, \\
\mathcal{E}_\mathrm{fw} &:= \sum_{n=1}^{m} \gamma_n \left\|\tilde{F}(x^n, w^n) - \Phi\Phi^T\tilde{F}(x^n, w^n)\right\|^2.
\end{aligned}$$

Proof. The proof is similar to Theorem 3.7 using a bound for G_n analogous to estimate (4.15) in Theorem 4.7. q.e.d.

4.2. Enhancement of the POD subspace

In the literature on POD we find several approaches where the use of derivative information for the enhancement of reduced-order models is suggested. Typically, the additional derivative information is used to improve the POD prediction properties of a surrogate model and, thus, obtain better performance of POD surrogate models when varying the control variables.

We find the first use of adjoint information in POD reduced-order models in [39]. The authors observe an improved numerical performance solving optimal control problems, when solutions of the adjoint states are included in the POD basis. In [62] adjoint information is also added to the POD basis which leads to better results for error estimation of suboptimal solutions of a linear-quadratic optimal control problem. Also in [112, Remark 4.10] a POD optimality is considered where states and adjoints are included. In this thesis we follow a similar idea for an inclusion of adjoint information in the POD basis. In addition, we suggest a particular method of how to include the additional adjoint (and sensitivity) snapshots using proper weighting. We extend the results to the DEIM projection and analyze the properties of the extended bases which gives an explanation of the superior performance of the enhanced surrogate model.

Sensitivity information for model reduction with reduced-basis methods was already used in the 1980s [88, 91, 92] in the context of parametrized PDEs without time dependence. Extensions to the POD methodology are found in [30, 122]. The authors follow an interpolation based POD approach. If only state information is used for the construction of the basis, they refer to it as Lagrange approach. When a subspace that also contains the derivatives with respect to parameters at different configurations is used, they refer to this as Taylor or Hermite approach. It is shown in [30] that the inclusion of derivatives results in a model which is more robust with respect to parameter changes and more efficient to compute. We discuss the topic more thoroughly in §5.1. In [125] the inclusion of derivatives is motivated by the importance of a ROM to be able to reflect also derivate information, which is aligned with our superior motivation to construct derivative-extended ROMs.

In [57, 95] time-dependent systems are considered and the sensitivity of the POD modes on parameter changes are investigated. The authors in [57] use these POD mode sensitivities to improve POD prediction by either extrapolation or by including them in the POD basis. The latter, however, might increase the order of the reduced basis significantly, that is by k times the number of parameters which is rather undesirable and stands in contrast to the extended POD optimality (4.26) of the sensitivity case. Their results are carried over to fluid flows in [56]. In [121] the authors follow a similar approach including POD mode sensitivity in the basis and truncating the enlarged basis using a Ritz approach.

Remark 4.1. In the case of time-dependent PDEs, typically already a lot of information of the system is already obtained with one single time integration for a fixed control variable q. Thus, an excessive offline phase, i.e., the computation of snapshots and possibly derivatives for different control configurations is uncommon for dynamic problems. Moreover, in the context of optimization the solution algorithm often converges in a moderate number of steps. Hence, using snapshots of multiple parameter configurations to compute (P-ROM) might destroy the efficiency of the model reduction approach for the optimization purpose.

In the following we present methods for the inclusion of adjoint and sensitivity information in pure POD models without DEIM projection. The enhancement of the DEIM basis is presented in §4.3. We start with the time-continuous case and then present the discrete case.

4.2.1. Adjoint case

We first consider (P-ROM) (without DEIM projection) together with its adjoint problem. From Lemma 4.1, we know that the adjoint equation (4.4) is formally a POD projection of the semi-discretized adjoint (2.21) with the relation $M^{-1}z = \Psi\widehat{z}$ between reduced and HiFi states. Due to $M^{-1}z = \tilde{z}$, we find that the POD approximation of the infinite-dimensional adjoint variable p is

$$p^k(t) = \sum_{i=1}^{N} \widehat{z}_i(t)\psi_i \quad \in V^k,$$

which is a projection of p^h as defined in (2.20) from the space V^h onto V^k. However, the space V^k is so far constructed from snapshots of the states only, hence, we cannot expect p^k to reflect the dynamics of the adjoint p^h. To overcome this problem we need to use an extended POD subspace V_z^k that contains the dynamics of the HiFi and the adjoint problem.

Recall the errors $\bar{\mathcal{E}}_x$ and $\bar{\mathcal{E}}_z$ of Theorem 4.3. For $\bar{\mathcal{E}}_z$ we have

$$\bar{\mathcal{E}}_z = \int_I \left\|z(t) - M\Psi\Psi^T z(t)\right\|_Z^2 dt = \int_I \left\|\tilde{z}(t) - \Psi\Psi^T M\tilde{z}(t)\right\|_X^2 dt.$$

In the continuous POD case the problem of finding a subspace V_z^k that minimizes $\bar{\mathcal{E}}_z$ and $\bar{\mathcal{E}}_x$ at the same time corresponds to the extended POD optimization problem

$$\begin{aligned} \min_{\Psi\in\mathbb{R}^{N\times k}} \quad & \int_I \zeta \left\|x(t) - \Psi\Psi^T Mx(t)\right\|_X^2 dt + \int_I \zeta^z \left\|\tilde{z}(t) - \Psi\Psi^T M\tilde{z}(t)\right\|_X^2 dt, \\ \text{s.t.} \quad & \Psi^T M\Psi = \mathbb{I}, \end{aligned} \tag{4.17}$$

with weights $\zeta, \zeta^z \in \mathbb{R}$, for which we give particular choices only in the time-discrete case. After suitable transformation, the POD projection error of the extended problem, and with this the POD related errors $\bar{\mathcal{E}}_x$ and $\bar{\mathcal{E}}_z$, can be estimated analogous to (3.5) as

$$\zeta\bar{\mathcal{E}}_x + \zeta^z\bar{\mathcal{E}}_z = \sum_{i=k+1}^{N} \bar{\lambda}_i^z, \tag{4.18}$$

where $\bar{\lambda}_1^z \geq \bar{\lambda}_2^z \geq \cdots \geq 0$ are the eigenvalues of $\mathbf{R} = \int_I x(t)x(t)^T M dt + \int_I \tilde{z}(t)\tilde{z}(t)^T M dt$.

In the time-discrete setting we proceed as follows. We take snapshots for states and adjoints at the same time instances $t_0, \ldots, t_{n_\tau} \in I$. Assume the snapshots $x^1, \ldots, x^m$ of the states and the adjoints $z^1, \ldots, z^m$ to be obtained with the implicit Euler scheme (2.8). The extended snapshot matrix $\mathcal{S}_z \in \mathbb{R}^{N\times 2\cdot m}$ then consists of the snapshot samples

$$\left\{\sqrt{\zeta_i}x^i\right\}_{i=1}^m, \quad \left\{\sqrt{\zeta_i^z}M^{-1}z^i\right\}_{i=1}^m \tag{4.19}$$

with positive weights ζ_j, ζ_j^z, $j = 1, \ldots, m$. It is important to choose these weights carefully, as different magnitudes in the derivatives and the nominal solutions can deteriorate the quality of the POD basis. Considering the general POD projection error (3.5) it is clear that the magnitude of the eigenvalues corresponds to the magnitude of the snapshots. Hence, when choosing a certain k, modes belonging to smaller eigenvalues are thrown out and the corresponding information is not captured. We propose to weight each snapshot by

$$\zeta_i = \frac{\gamma_i}{\left(\epsilon_s + \sum_{l=1}^m \gamma_l \left\|x^l\right\|_X\right)^2}, \quad \zeta_i^z = \frac{\theta\gamma_i}{\left(\epsilon_s + \sum_{l=1}^m \gamma_l \left\|M^{-1}z^l\right\|_X\right)^2}, \quad i = 1, \ldots, m, \tag{4.20}$$

where $\gamma_1, \ldots, \gamma_m$ are the snapshot time weights as in (3.2), ϵ_s is a small constant, and $\theta \in (0, 1]$. Thus, the snapshots are normalized dividing by an approximation of $\int_I \|x(t)\|_X \, dt$, while ϵ_s prevents division by zero. With the additional adjoint snapshot factor θ we have the possibility to control the amount of adjoint information added and, thus, give the states more importance in the basis. Clearly, we aim to include the additional information without corrupting the projection error for the states for a standard POD basis. We assess the issue in Remark 4.2.

Replacing the standard weights γ_i by $\zeta_1, \ldots, \zeta_m$ and $\zeta_1^z, \ldots, \zeta_m^z$ in the reconstruction estimate of Theorem 4.4 we have

$$\mathcal{E}_x = \sum_{i=1}^{m} \zeta_i \left\| x^i - \Psi\Psi^T M x^i \right\|_X^2 dt, \quad \mathcal{E}_z = \sum_{i=1}^{m} \zeta_i^z \left\| \tilde{z}^i - \Psi\Psi^T M \tilde{z}^i \right\|_X^2 dt.$$

Using the extended snapshot matrix $\mathcal{S}_z$ for the computation of the basis corresponds to solving the extended POD optimality problem

$$\min_{\Psi \in \mathbb{R}^{N \times k}} \mathcal{E}_x + \mathcal{E}_z \qquad \text{s.t.} \quad \Psi^T M \Psi = \mathbb{I}. \tag{4.21}$$

We denote the extended subspace resulting from either (4.21) or (4.21) as $V_z^k \subseteq V^h$. The residual of (4.21) in the solution is

$$\mathcal{E}_x + \mathcal{E}_z = \sum_{i=k+1}^{d} \lambda_i^z, \tag{4.22}$$

with $\lambda_1^z, \ldots, \lambda_d^z$ being the eigenvalues of the matrix $\mathbf{R}_m = \mathcal{S}_z \mathcal{S}_z^T M$ and $d = \operatorname{rank}(\mathcal{S}_z)$.

Remark 4.2. By determining the number of basis functions for the extended snapshot matrix $\mathcal{S}_z$ according to a given tolerance criterion (see Remark (3.2)), the inclusion of additional information in the basis does not come at the cost of losing approximation quality of the states. I.e., using the ϵ_P criterion, obviously, we have

$$\mathcal{E}_x \leq \mathcal{E}_x + \mathcal{E}_z = \sum_{i=k+1}^{d} \lambda_i^z \leq \epsilon_P.$$

Clearly, the number of necessary modes k will in general be larger, however, it is still an optimal choice in the sense of (4.21).

Remark 4.3. In practice, the adjoint snapshots are computed using automatic differentiation (AD) following the principle of internal numerical differentiation. The need to multiply each adjoint snapshot z^j by M^{-1} comes quite natural from the PDE perspective. However, it is less obvious from the AD perspective, where the interpretation of adjoint information is unclear in general.

4.2.2. Sensitivity case

In the sensitivity case we enhance the subspace V^k such that we obtain a bound for the states and all VDE solutions for a given set of control variable directions. We consider a POD reduced-order model without DEIM projection first. The Galerkin approximation of a directional derivative of the states with respect to a $\tilde{q} \in \mathcal{Q}^h$ is given as

$$y_q^h(t)\tilde{q} = \sum_{i=1}^{N} w_i(t)\varphi_i \ \in V^h.$$

Accordingly, with solutions $\widehat{w}(t)$ of the VDE (4.10) to (P-ROM) and recalling Lemma 4.5, the approximation of the directional derivative in V^k is

$$y_q^k(t)\tilde{q} = \sum_{i=1}^{k} \widehat{w}_i(t)\psi_i \ \in V^k.$$

Obviously, the sensitivities are in the same space as the states. However, V^k is constructed from state information only, hence, we cannot expect $y_q^k\tilde{q}$ to be a good approximation of $y_q^h\tilde{q}$.

Assume that we have continuous solutions $w^1(t), \dots, w^{n_q}(t)$ of (4.9) for a set of directions $\tilde{q}_1, \dots, \tilde{q}_{n_q} \in \mathcal{Q}^h$. We want to construct an extended basis V_w^k that minimizes the errors $\bar{\mathcal{E}}_x$ and $\bar{\mathcal{E}}_w$ in Theorem 4.7 for the state and all VDE solutions. Therefore, we denote the error related to each solution $w^j(t)$ by

$$\bar{\mathcal{E}}_w^j = \int_I \left\| w^j(t) - \Psi\Psi^T M w^j(t) \right\|_X^2 dt, \quad j = 1, \dots, n_q,$$

and consider the extended POD optimality

$$\min_{\Psi \in \mathbb{R}^{N\times k}} \zeta\bar{\mathcal{E}}_x + \sum_{j=1}^{n_q} \zeta^{w,j}\bar{\mathcal{E}}_w^j \qquad \text{s.t.} \quad \Psi^T M \Psi = \mathbb{I}, \tag{4.23}$$

with weights $\zeta, \zeta^{w,1}, \dots, \zeta^{w,n_q} \in \mathbb{R}$, for which we give particular choices in the time-discrete setting. The POD projection error is

$$\zeta\bar{\mathcal{E}}_x + \sum_{j=1}^{n_q} \zeta^{w,j}\bar{\mathcal{E}}_w^j = \sum_{i=1}^{N} \bar{\lambda}_i^w,$$

with the eigenvalues $\bar{\lambda}_i^w$ of the operator $\mathbf{R}$ belonging to (4.23) analogous to (3.10).

For the discrete case we assume again that the snapshots are obtained from the implicit Euler scheme (2.8) on the grid $t_0, \dots, t_{n_\tau} \in I$. Let

$$w^{i,j} := w^j(t_{i-1}), \quad i = 1, \dots, m, \quad j = 1, \dots, n_q$$

be the VDE snapshots and $x^1, \dots, x^m$ the state snapshots as for standard POD. We consider the sensitivity-extended snapshot matrix $\mathcal{S}_w \in \mathbb{R}^{N\times m + n_q\cdot m}$ consisting of the snapshots

$$\left\{\sqrt{\zeta_i}x^i\right\}_{i=1}^m, \left\{\sqrt{\zeta_i^{w,1}}w^{i,1}\right\}_{i=1}^m, \dots, \left\{\sqrt{\zeta_i^{w,n_q}}w^{i,n_q}\right\}_{i=1}^m \tag{4.24}$$

The weights are chosen as follows

$$\begin{gathered}\zeta_i = \frac{\gamma_i}{\left(\epsilon_s + \sum_{l=1}^m \gamma_l \left\|x^l\right\|_X\right)^2}, \quad \zeta_i^{w,j} = \frac{\theta\gamma_i}{\left(\epsilon_s + \sum_{l=1}^m \gamma_l \left\|w^{l,j}\right\|_X\right)^2}, \\ i = 1, \dots, m, \quad j = 1, \dots, n_q,\end{gathered} \tag{4.25}$$

using the time weights $\gamma_1, \dots, \gamma_m$. As before we introduce a factor $\theta \in (0, 1]$ to control the amount of derivative information in the basis and $\epsilon_s > 0$ to prevent division by zero.

Proper orthogonal decomposition of the matrix $\mathcal{S}_w$ is equivalent to solving the extended POD optimality problem in the discrete case, which is given as

$$\min_{\Psi \in \mathbb{R}^{N\times k}} \mathcal{E}_x + \sum_{j=1}^{n_q} \mathcal{E}_w^j \qquad \text{s.t.} \quad \Psi^T M \Psi = \mathbb{I}, \tag{4.26}$$

where

$$\mathcal{E}_x = \sum_{i=1}^{m} \zeta_i \left\| x^i - \Psi\Psi^T M x^i \right\|_X^2, \quad \mathcal{E}_w^j = \sum_{i=1}^{m} \zeta_i^{w,j} \left\| w^{i,j} - \Psi\Psi^T M w^{i,j} \right\|_X^2.$$

The dimension of the subspace V_w^k is again chosen according to Remark 3.2, thus, we guarantee that adding additional snapshots does not come at the cost of a loss of dynamic information of the state problem (see also Remark 4.2).

In the sensitivity case we might face the problem that the number of snapshots causes the decomposition of the snapshot matrix to be slow. We deal with this issue later in the applications part.

4.3. Enhancement of the DEIM subspace

An exploitation of derivative information in the context of DEIM approximations for reduced-order modeling is not documented so far in literature. The authors in [41] present results for parametric derivative approximations with EIM. It is shown that, provided the EIM approximation scheme is convergent, this also holds for parametric derivatives under certain additional regularity assumptions. However, as DEIM only operates on the discrete level, the relation to their results is beyond the focus of this thesis.

We proceed similar as for the POD subspace enhancement, first using additional information to enrich the DEIM basis in the adjoint case and subsequently in the sensitivity case, first dealing with the continuous and then with the time-discrete case respectively.

4.3.1. Adjoint case

We seek to enhance the DEIM subspace R^ℓ such that we obtain small POD reconstruction errors between solutions of the HiFi adjoint (4.3) and the adjoint (4.5) of (PD-ROM). However, in the adjoint case we also need to have in mind the overall goal of minimizing the reconstruction error of the objective and its gradient in the underlying optimization problem. Thus, additional sensitivity information with respect to q must be added that is not related directly to the adjoint problem.

Assume first that an extended POD optimality problem is solved that includes $\bar{\mathcal{E}}_{\mathrm{f}}$ and $\bar{\mathcal{E}}_{\mathrm{f}z}$ as given in Theorem 4.3 in analogy to the above sections. This would enable us to estimate the two errors with the neglected eigenvalues of a corresponding operator $\mathbf{R}$ and with this the reconstruction error for the adjoints. However, for a reconstruction error bound for the gradient, inclusion of adjoint information is not enough. Therefore, we consider the POD approximation $\widehat{j}_q^h(q)\tilde{q}$ of a derivative of the reduced objective j^h in a direction $\tilde{q}$ based on solutions $\widehat{x}$ of (PD-ROM). Analogous to (2.19) it is given as

$$\widehat{j}_q^h(q)\tilde{q} = \int_I \widehat{z}^T(t)\Psi^T \mathbb{P} F_q(\Psi\widehat{x}(t), q)\tilde{q}\; dt + J_q^h(\Psi\widehat{x}, q)\tilde{q}. \tag{4.27}$$

As the DEIM projection also affects $F_q(\Psi\widehat{x}(t), q)$, we need to use a DEIM basis that also includes sensitivity information with respect to q. For the error related to $F_q(\Psi\widehat{x}(t), q)$ and a set of directions $\tilde{q}_1, \ldots, \tilde{q}_{n_q}$ we introduce

$$\bar{\mathcal{E}}_{\mathrm{f}q}^j := \int_I \left\| F_q(x(t), q)\tilde{q}_j - \Phi\Phi^T F_q(x(t), q)\tilde{q}_j \right\|^2 \, dt, \quad j = 1, \ldots, n_q. \tag{4.28}$$

Thus, with $\bar{\mathcal{E}}_{\mathrm{f}}$, $\bar{\mathcal{E}}_{\mathrm{f}z}^i$ of Theorem 4.3 and weights $\tilde{\zeta}, \tilde{\zeta}^z, \tilde{\zeta}^{q,1}, \dots, \tilde{\zeta}^{q,n_q} \in \mathbb{R}$ we consider the extended POD optimality for DEIM

$$\min_{\Phi \in \mathbb{R}^{N \times \ell}} \; \tilde{\zeta}\bar{\mathcal{E}}_{\mathrm{f}} + \tilde{\zeta}^z \sum_{i=1}^{k} \bar{\mathcal{E}}_{\mathrm{f}z}^i + \sum_{j=1}^{n_q} \tilde{\zeta}^{q,j} \bar{\mathcal{E}}_{\mathrm{f}q}^j \qquad \text{s.t. } \; \Phi^T\Phi = \mathbb{I}, \tag{4.29}$$

where in contrast to a standard POD optimality we need to approximate $1 + k + n_q$ time-dependent vectors. We specify the weights only in the time-discrete setting. Note that we use only one weight $\tilde{\zeta}^z$ for all $\bar{\mathcal{E}}_{\mathrm{f}z}^i$ as we know that the directions $\Psi_{\cdot i}$ are M-orthonormal. With eigenvalues $\bar{s}_1^z, \dots, \bar{s}_N^z$ of the operator $\mathbf{R}$ belonging to (4.29) the residual is given by

$$\tilde{\zeta}\bar{\mathcal{E}}_{\mathrm{f}} + \tilde{\zeta}^z \sum_{i=1}^{k} \bar{\mathcal{E}}_{\mathrm{f}z}^i + \sum_{j=1}^{n_q} \tilde{\zeta}^{q,j} \bar{\mathcal{E}}_{\mathrm{f}q}^j = \sum_{n=\ell+1}^{N} \bar{s}_n^z. \tag{4.30}$$

Proceeding analogously in the time-discrete case, we now need to approximate $(1 + k + n_q) \cdot m$ snapshot vectors. To build the (PD-ROM) efficiently, in practice the number of snapshots needs to be decreased which we handle in chapter 6. For adjoint-extended DEIM we consider the snapshot matrix $\mathcal{D}_z$ that contains the vectors

$$\begin{gathered} \left\{ \sqrt{\tilde{\zeta}_i} F(x^i, q) \right\}_{i=1}^{m}, \quad \left\{ \sqrt{\tilde{\zeta}_i^z} F_x(x^i, q)\Psi \right\}_{i=1}^{m}, \\ \left\{ \sqrt{\tilde{\zeta}_i^{q,1}} F_q(x^i, q)\tilde{q}_1 \right\}_{i=1}^{m}, \; \dots \; , \left\{ \sqrt{\tilde{\zeta}_i^{q,n_q}} F_q(x^i, q)\tilde{q}_{n_q} \right\}_{i=1}^{m}. \end{gathered} \tag{4.31}$$

We choose the weights as

$$\begin{gathered} \tilde{\zeta}_i = \frac{\gamma_i}{(\epsilon_s + \sum_{l=1}^m \gamma_l \left\| F(x^l, q) \right\|)^2}, \quad \tilde{\zeta}_i^z = \frac{\theta_z \gamma_i}{(\epsilon_s + \sum_{l=1}^m \gamma_l \left\| F_x(x^l, q)\Psi_{\cdot 1} \right\|)^2}, \\ \tilde{\zeta}_i^{q,j} = \frac{\theta_q \gamma_i}{(\epsilon_s + \sum_{l=1}^m \gamma_l \left\| F_q(x^l, q)\tilde{q}_j \right\|)^2}, \quad i = 1, \dots, m, \quad j = 1, \dots, n_q. \end{gathered}$$

As for the extended POD basis, we use a $\epsilon_s > 0$ and choose factors $0 < \theta_z, \theta_q \leq 1$. Including the above weights in the DEIM related time-discrete errors we get

$$\begin{aligned} \mathcal{E}_{\mathrm{f}} &= \sum_{i=1}^{m} \tilde{\zeta}_i \left\| F(x^l, q) - \Phi\Phi^T F(x^i, q) \right\|^2, \\ \mathcal{E}_{\mathrm{f}z}^i &= \sum_{i=1}^{m} \tilde{\zeta}_i^z \left\| F_z(x^i, q)\Psi_{\cdot i} - \Phi\Phi^T F_z(x^i, q)\Psi_{\cdot i} \right\|^2, \\ \mathcal{E}_{\mathrm{f}q}^j &= \sum_{i=1}^{m} \tilde{\zeta}_i^{q,j} \left\| F_q(x^i, q)\tilde{q}_j - \Phi\Phi^T F_q(x^i, q)\tilde{q}_j \right\|^2. \end{aligned}$$

The extended POD optimality in the discrete setting is

$$\min_{\Phi \in \mathbb{R}^{N \times \ell}} \mathcal{E}_{\mathrm{f}} + \sum_{i=1}^{k} \mathcal{E}_{\mathrm{f}z}^i + \sum_{j=1}^{n_q} \mathcal{E}_{\mathrm{f}q}^j \qquad \text{s.t. } \; \Phi^T\Phi = \mathbb{I}. \tag{4.32}$$

We denote the DEIM subspace spanned by a POD basis constructed from $\mathcal{D}_z$ by R_z^ℓ. With the eigenvalues $s_1^z, \dots, s_{\tilde{d}}^z$ corresponding to the snapshot matrix $\mathcal{D}_z$ and $\tilde{d} = \operatorname{rank}(\mathcal{D}_z)$ in a solution of (4.32) the POD projection error is

$$\mathcal{E}_{\mathrm{f}} + \sum_{i=1}^{k} \mathcal{E}_{\mathrm{f}z}^i + \sum_{j=1}^{n_q} \mathcal{E}_{\mathrm{f}q}^j = \sum_{n=\ell+1}^{\tilde{d}} s_n^z \tag{4.33}$$

Remark 4.4. The inclusion of an error term $\bar{\mathcal{E}}_{\mathrm{f}q}^{j}$ in the extended POD optimality is only necessary when there is a nonlinear dependence of F on q in a direction $\tilde{q}_j$. E.g., for $\tilde{q}_j = e_j$ and q_j entering linearly, $F_q(x(t),q)\tilde{q}_j$ only differs from $F(x(t),q)$ by a constant factor and should already be well represented in the DEIM subspace.

Remark 4.5. In the presentation of the dynamic model problems we have pointed out that with DEIM we face additional difficulties when the control is spatially distributed and enters nonlinearly. According to our strategy to compute the extended DEIM basis, we need to include as many snapshot sets as we have discretized control parameters q. In this situation we cannot expect to obtain a small DEIM subspace. The problem is inherent, as in this situation we seek to find a small subspace, but still preserve all properties of the possibly large control discretization.

4.3.2. Sensitivity case

The last scenario of subspace enrichment in the sensitivity case for DEIM follows mostly analogous. Assume that we seek a set of directional derivatives of the objective with directions $\tilde{q}_1, \dots, \tilde{q}_{n_q}$. The continuous extended POD optimality for the DEIM basis is then

$$\min_{\Phi \in \mathbb{R}^{N \times \ell}} \tilde{\zeta} \bar{\mathcal{E}}_{\mathrm{f}} + \sum_{j=1}^{n_q} \tilde{\zeta}^{w,j} \bar{\mathcal{E}}_{\mathrm{f}w}^{j} \qquad \text{s.t.} \;\; \Phi^T \Phi = \mathbb{I}, \tag{4.34}$$

with the scalar weights $\tilde{\zeta}, \tilde{\zeta}^{w,1}, \dots, \tilde{\zeta}^{w,n_q}$, the error $\bar{\mathcal{E}}_{\mathrm{f}}$ as in Theorem 3.6, and

$$\bar{\mathcal{E}}_{\mathrm{f}w}^{j} := \int_I \left\| \tilde{F}(x(t), w^j(t)) - \Phi\Phi^T \tilde{F}(x(t), w^j(t)) \right\|^2 \, dt,$$

where

$$\tilde{F}(x(t), w^j(t)) = F_x(x(t), q) w^j(t) + F_q(x(t), q)\tilde{q}_j$$

and $w^1(t), \dots, w^{n_q}(t)$ are solutions to (4.9) for a set of directions $\tilde{q}_1, \dots, \tilde{q}_{n_q} \in \mathcal{Q}^h$. The POD projection error in the solution of (4.34) is given as

$$\tilde{\zeta} \bar{\mathcal{E}}_{\mathrm{f}} + \sum_{j=1}^{n_q} \tilde{\zeta}^{w,j} \bar{\mathcal{E}}_{\mathrm{f}w}^{j} = \sum_{i=\ell+1}^{N} \bar{s}_i^w,$$

where $\bar{s}_1^w, \dots, \bar{s}_N^w$ are the eigenvalues of the corresponding operator.

In the time-discrete case we use a snapshot matrix $\mathcal{D}_w$ that consists of

$$\left\{ \sqrt{\tilde{\zeta}_i} F(x^i, q) \right\}_{i=1}^{m}, \quad \left\{ \sqrt{\tilde{\zeta}_l^{q,j}} \tilde{F}(x^i, w^{i,j}) \right\}_{i=1}^{m}, \quad j = 1, \dots, n_q. \tag{4.35}$$

The weights are chosen as

$$\tilde{\zeta}_i = \frac{\gamma_i}{(\epsilon_s + \sum_{l=1}^m \gamma_l \left\| F(x^l, q) \right\|)^2}, \quad \tilde{\zeta}_i^{w,j} = \frac{\theta_w \gamma_i}{(\epsilon_s + \sum_{l=1}^m \gamma_l \| \tilde{F}(x^l, w^{l,j}) \|)^2},$$
$$i = 1, \dots, m, \quad j = 1, \dots, n_q,$$

where $\theta_w \in (0, 1]$ and $\epsilon_s > 0$. The time-discrete POD optimality is

$$\min_{\Phi \in \mathbb{R}^{N \times \ell}} \mathcal{E}_{\mathrm{f}} + \sum_{j=1}^{n_q} \mathcal{E}_{\mathrm{f}w}^{j} \qquad \text{s.t.} \;\; \Phi^T \Phi = \mathbb{I}, \tag{4.36}$$

where

$$\mathcal{E}_{\mathrm{f}w}^{j} := \sum_{i=1}^{m} \tilde{\zeta}_i^{w,j} \left\| \tilde{F}(x^i, w^{i,j}) - \Phi\Phi^T \tilde{F}^j(x^i, w^{i,j}) \right\|^2 ,$$

We denote the DEIM subspace spanned by a POD basis constructed from $\mathcal{D}_w$ by R_w^ℓ. With corresponding eigenvalues $s_1^w, \dots, s_{\tilde{d}}^w$ and $\tilde{d} = \operatorname{rank}(\mathcal{D}_w)$, for the POD projection error we have

$$\mathcal{E}_{\mathrm{f}} + \sum_{j=1}^{n_q} \mathcal{E}_{\mathrm{f}w}^{j} = \sum_{l=\ell+1}^{\tilde{d}} s_l^w .$$

4.4. Reconstruction error estimates for the gradient

We now make use of the results in §4.2 and §4.3 to present reconstruction error estimates for the gradient of the reduced objectives $j^h(q)$ and $j^{h\tau}(q)$.

Definition 4.9 (DEPOD/DEDEIM). We refer to the approach of using the extended POD optimalities (4.17), (4.21), (4.23), or (4.26) for the POD basis as derivative-extended proper orthogonal decomposition (DEPOD). Using either (4.29), (4.32), (4.34), or (4.36) for the DEIM basis, we refer to as derivative-extended discrete empirical interpolation method (DEDEIM). We call the enriched subspaces V_z^k, V_w^k *DEPOD subspaces* and the enriched subspaces R_z^ℓ, R_w^ℓ *DEDEIM subspaces.*

Whether we mean the adjoint or sensitivity case and the time-continuous or time-discrete case will be mentioned explicitly or clear from the context. We start with the time-continuous case and the derivative representation of the objective using adjoints.

Theorem 4.10. *For a fix $q \in \mathcal{Q}^h$ assume the solutions $x(t), z(t)$ of* (2.7) *and* (4.3) *and the solutions $\widehat{x}(t), \widehat{z}(t)$ of* (PD-ROM) *and* (4.5) *to satisfy the reconstruction estimates* (3.27) *and* (4.6). *Let $\tilde{q} \in \mathcal{Q}^h$ and $F_q(x,q)\tilde{q}$ be Lipschitz-continuous in x. If the projection matrices Ψ and Φ and corresponding eigenvalues $\bar{\lambda}_1^z, \dots, \bar{\lambda}_N^z$ and $\bar{s}_1^z, \dots, \bar{s}z_N$ are computed according to the extended POD optimalities* (4.17) *and* (4.29), *then there is a constant $\bar{C}_{Jz} < \infty$ such that*

$$RE\left(j_q^h(q)\tilde{q}\right) \leq \bar{C}_{Jz} \left(\sum_{i=k+1}^{N} \bar{\lambda}_i^z + \sum_{j=\ell+1}^{N} \bar{s}_j^z \right). \tag{4.37}$$

Proof. By definition for the reconstruction error we have

$$RE\left(j_q^h(q)\tilde{q}\right) = \left\| j_q^h(q)\tilde{q} - \widehat{j}_q^h(q)\tilde{q} \right\| / c_0,$$

with $c_0 := \|j_q^h(q)\tilde{q}\|$ and the directional derivatives $j_q^h(q)\tilde{q}$ and $\widehat{j}_q^h(q)\tilde{q}$ as in (2.19) and (4.27). Exploiting that $J_q^h(x,q)\tilde{q}$ is independent of x gives

$$\begin{aligned} RE\left(j^h(q)_q\tilde{q}\right) \leq & \int_I \left\| z^T(t) M^{-1} F_q(x(t),q)\tilde{q} - \widehat{z}^T(t)\Psi^T \mathbb{P} F_q(\Psi\widehat{x}(t),q)\tilde{q} \right\| \, dt/c_0 \\ \leq & \int_I \left\| z^T(t) M^{-1} F_q(x(t),q)\tilde{q} - z^T(t) M^{-1} \mathbb{P} F_q(x(t),q)\tilde{q} \right\| \\ & + \left\| z^T(t) M^{-1} \mathbb{P} F_q(x(t),q)\tilde{q} - z^T(t) M^{-1} \mathbb{P} F_q(\Psi\widehat{x}(t),q)\tilde{q} \right\| \\ & + \left\| z^T(t) M^{-1} \mathbb{P} F_q(\Psi\widehat{x}(t),q)\tilde{q} - \widehat{z}^T(t)\Psi^T \mathbb{P} F_q(\Psi\widehat{x}(t),q)\tilde{q} \right\| \, dt/c_0 \\ \leq & \, c_1 \int_I \left\| F_q(x(t),q)\tilde{q} - \Phi\Phi^T F_q(x(t),q)\tilde{q} \right\| \, dt \\ & + c_2 \int_I \|x(t) - \Psi\widehat{x}(t)\|_X \, dt + c_3 \int_I \left\| M^{-1} z(t) - \Psi\widehat{z}(t) \right\|_X \, dt \end{aligned}$$

with constants $c_1 := z_{max} \|\mathbb{I} - \mathbb{P}\| / c_0$, $c_2 := z_{max} \|\mathbb{P}\| L_f / c_0$, $c_3 := \sup_t(\|\mathbb{P} F_q(\Psi\widehat{x}(t))\|)/c_0$, the Lipschitz constant L_f of $F_q(x(t), q)\tilde{q}$, and $z_{max} := \sup_t(\left\|z(t)^T\right\|_Z)$. For the first term we applied (3.16). Using the reconstruction estimates (3.27) and (4.6), the definition of $\bar{\mathcal{E}}^j_{\mathrm{f}q}$ in (4.28), and the mean value theorem, then there are constants $C_1, C_2, C_3 < \infty$ such that

$$\int_I \left\|F_q(x(t), q)\tilde{q} - \Phi\Phi^T F_q(x(t), q)\tilde{q}\right\| \, dt \leq C_1 \bar{\mathcal{E}}_{\mathrm{f}q},$$

$$\int_I \|x(t) - \Psi\widehat{x}(t)\|_X \, dt \leq C_2 \left(\bar{\mathcal{E}}_x + \bar{\mathcal{E}}_{\mathrm{f}}\right),$$

$$\int_I \left\|M^{-1} z(t) - \Psi\widehat{z}(t)\right\|_X \, dt \leq C_3 \left(\bar{\mathcal{E}}_x + \bar{\mathcal{E}}_{\mathrm{f}} + \bar{\mathcal{E}}_z + \sum_{l=1}^{k} \bar{\mathcal{E}}^l_{\mathrm{f}z}\right).$$

Due to the extended POD optimalities, the errors on the right are bounded according to the projection error estimates (4.18) and (4.30). Thus, we can find a constant $\bar{C}_{Jz} < \infty$ such that

$$\begin{aligned} & c_1 \int_I \left\|F_q(x(t), q)\tilde{q} - \Phi\Phi^T F_q(x(t), q)\tilde{q}\right\| \, dt \\ & + \; c_2 \int_I \|x(t) - \Psi\widehat{x}(t)\|_X \, dt + \; c_3 \int_I \left\|M^{-1} z(t) - \Psi\widehat{z}(t)\right\|_X \, dt \\ & \leq \; \bar{C}_{Jz} \left(\zeta\bar{\mathcal{E}}_x + \zeta^z\bar{\mathcal{E}}_z + \tilde{\zeta}\bar{\mathcal{E}}_{\mathrm{f}} + \tilde{\zeta}^z \sum_{l=1}^{k} \bar{\mathcal{E}}^l_{\mathrm{f}z} + \tilde{\zeta}^q \bar{\mathcal{E}}_{\mathrm{f}q}\right) = \; \bar{C}_{Jz} \left(\sum_{i=k+1}^{N} \bar{\lambda}^z_i + \sum_{j=\ell+1}^{N} \bar{s}^z_j\right) \end{aligned}$$

which concludes the proof. q.e.d.

Theorem 4.10 is a direct consequence of the Theorems 3.6 and 4.3 for the reconstruction error of states and adjoints.

In the time-discrete setting the integral in the reduced objective must be approximated by a quadrature rule. Considering the implicit Euler scheme (2.8) with n_τ steps and using the trapezoidal weights $\gamma_1, \ldots, \gamma_m$ as in (3.2) for the $m = n_\tau + 1$ snapshot locations, we get

$$j^{h\tau}_q(q)\tilde{q} = \sum_{n=1}^{m} \gamma_n (z^n)^T M^{-1} F_q(x^n, q)\tilde{q} + J^{h\tau}_q(x^n, q)\tilde{q}. \tag{4.38}$$

We here assume fixed $q, \tilde{q} \in \mathcal{Q}^{h\tau}$ and require $q, \tilde{q}$ to be given such that the time-discrete reconstruction estimates for states and adjoints hold and the implicit equations in each Euler step are satisfied (compare also Remark 2.3). The reconstruction estimate for the gradient follows analogous to the time-continuous case and we obtain

$$RE\left(j^{h\tau}_q(q)\tilde{q}\right) \leq C_{Jz} \left(\sum_{i=k+1}^{d} \lambda^z_i + \sum_{j=\ell+1}^{\tilde{d}} s^z_j\right), \tag{4.39}$$

with a constant $C_{Jz} < \infty$.

Directional derivatives of the reduced objective in the sensitivity case using (PD-ROM) for its approximation are given as (compare (2.25))

$$\widehat{j}^h_q(q)\tilde{q} = \; J^h_{\widehat{x}}(\Psi\widehat{x}(T))\widehat{w}(T) + J^h_q(\Psi\widehat{x}(T), q)\tilde{q}.$$

Theorem 4.11. *Let the set of directions $\tilde{q}_1, \ldots, \tilde{q}_{n_q}$ form a basis of $\mathcal{Q}^{h\tau}$. Assume that the corresponding VDE solutions $w^1(t), \ldots, w^{n_q}(t)$ and $\widehat{w}^1(t), \ldots, \widehat{w}^{n_q}(t)$ of (4.9) and (4.11)* *and solutions $x(t)$ and $\widehat{x}(t)$ of* (2.7) *and* (PD-ROM) *satisfy the reconstruction estimates* (4.13) *and* (3.27) *at $q \in \mathcal{Q}^h$ for all $\tilde{q}_i$. If Ψ,Φ,$\lambda_1^w, \dots, \lambda_N^w$, and $s_1^w, \dots, s_N^w$ are computed according to the extended POD optimalities* (4.23) *and* (4.34)*, then there is a constant $\bar{C}_{Jw} < \infty$ such that for all $\tilde{q} \in \mathcal{Q}^{h\tau}$*

$$RE\left(j_q^h(q)\tilde{q}\right) = \bar{C}_{Jw}\left(\sum_{i=k+1}^{N} \bar{\lambda}_i^w + \sum_{j=\ell+1}^{N} \bar{s}_j^w\right). \tag{4.40}$$

Proof. As the set of directions $\tilde{q}_1, \dots, \tilde{q}_{n_q}$ form a basis we can decompose $\tilde{q} = \sum_{i=1}^{n_q} c_i\tilde{q}_i$. Thus,

$$RE\left(j_q^h(q)\tilde{q}\right) = RE\left(\sum_{i=1}^{n_q} j_q^h(q)c_i\tilde{q}_i\right)$$

The assertion is now directly obtained applying the reconstruction estimates for the states and VDE solutions to each term of the sum and using the continuous embedding of Y in $C(I, H)$. q.e.d.

Note that here we consider a finite-dimensional control variable space $\mathcal{Q}^{h\tau}$ already in the time-continuous setting. In the time-discrete setting we have

$$\widehat{j}_q^{h\tau}(q)\tilde{q} = J_{\widehat{x}}^h(\Psi\widehat{x}^m)\widehat{w}^m + J_q^h(\Psi\widehat{x}^m, q)\tilde{q}.$$

Analogously we find a constant C_{Jw} such that

$$RE\left(j_q^{h\tau}(q)\tilde{q}\right) = C_{Jw}\left(\sum_{i=k+1}^{d} \lambda_i^w + \sum_{j=\ell+1}^{\tilde{d}} s_j^w\right), \tag{4.41}$$

where $q, \tilde{q} \in \mathcal{Q}^{h\tau}$. The estimate follows directly from Theorems 3.7 and 4.8 assuming that the set of directions $\tilde{q}_1, \dots, \tilde{q}_{n_q}$ used to construct the sensitivity-extended basis form a basis of $\mathcal{Q}^{h\tau}$.

Remark 4.6. In the time-continuous case the reconstruction error $RE\left(j_q^h(q)\tilde{q}\right)$ for the gradient is given for analytical expressions and, thus, is not affected by a particular discretization of the equations it depends on. Assume that (PD-ROM) with either adjoint or sensitivity-extended DEPOD/DEDEIM subspaces is used and we want to compute the POD approximation of the directional derivative $\widehat{j}_q^h(q)\tilde{q}$. If $\widehat{j}_q^h(q)\tilde{q}$ is approximated sufficiently well using any method for derivative computation, the bounds (4.37) and (4.40) still hold up to the approximation error of the method used to generate the derivative. I.e., in practice we could use either finite differences, the sensitivity, or the adjoint approach to compute $\widehat{j}_q^h(q)\tilde{q}$ independent of the particular enhancement method. In either case given a DEPOD/DEDEIM basis is used, we can expect good approximations of the HiFi derivatives. Thus, by adding either sensitivity or adjoint information we make the ability to allow derivative approximations of the reduced objective to an inherent property of the surrogate model.

4.5. Numerical results

We conclude the chapter on derivative-extended reduced-order models presenting numerical results that reflect the theoretical assertions and that illustrate the practical reconstruction properties of DEPOD/DEDEIM reduced-order models. We recycle the example problem introduced in §3.3.5 for state reconstruction and the cases A and B. As objective function we use

$$J(y,u) = \frac{1}{2}\left\|y(T) - y_\Omega\right\|_\Omega^2, \quad \text{where} \quad y_\Omega(r_1,r_2) := \begin{cases} 1 & \text{if } r_2 > 0.5 \\ 0 & \text{else.} \end{cases} \tag{4.42}$$

The tests are carried out in analogy to §3.3.5, thus, the same discretization ($N = 289$, $\tau = 10^{-3}$) is used and all available snapshots are used to compose the according snapshot matrix. The additional weights θ_z and θ_w for the extended bases are set to one. The derivatives in the adjoint and in the forward case are computed following the discrete concepts for derivative computation as described in §2.5. Note that, hence, any of the discrete adjoints and sensitivities in the following are not explicit discretizations of the time-continuous problem, however, we have consistent approximations due to the use of internal numerical differentiation (IND).

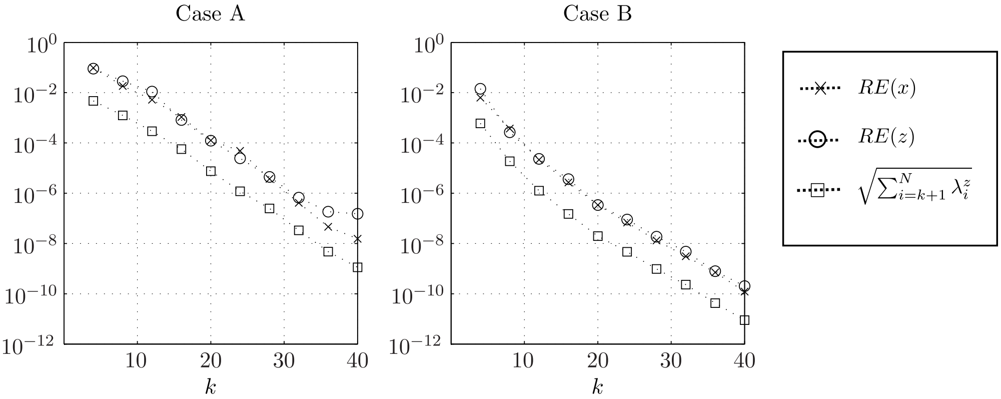

Figure 4.3.: Reconstruction results for states and adjoints with DEPOD basis using the norms $\|\cdot\|_{L^2(I,X)}$ and $\|\cdot\|_{L^2(I,Z)}$ and the projection error expressed via the neglected eigenvalues corresponding to $\mathcal{S}_z$.

Adjoint DEPOD reconstruction

In Figure 4.3 we show the POD reconstruction results for state and adjoint variables in dependence of the number of DEPOD basis functions used to construct (P-ROM). The reconstruction error is evaluated as in the tests for the states using the norms

$$\|x\|_{L^2(I,X)} = \left(\int_I \|x\|_X^2 \, dt\right)^{\frac{1}{2}} \quad \text{and} \quad \|z\|_{L^2(I,Z)} = \left(\int_I \|z\|_Z^2 \, dt\right)^{\frac{1}{2}}.$$

As before in the case of pure state reconstruction we observe for both cases an exponentially decreasing error that is closely connected to the decay of the neglected eigenvalues of the enriched basis. The number of necessary basis functions for a certain accuracy is now larger as more information needs to be contained in the POD subspace. However, in either case the size of the surrogate model is still tractable (compare to Figure 3.4). Moreover, the reconstruction errors of states and adjoints are of comparable size in this case caused by the common dominating projection error and the normalization of the snapshots. In general, however, the difference may be larger.

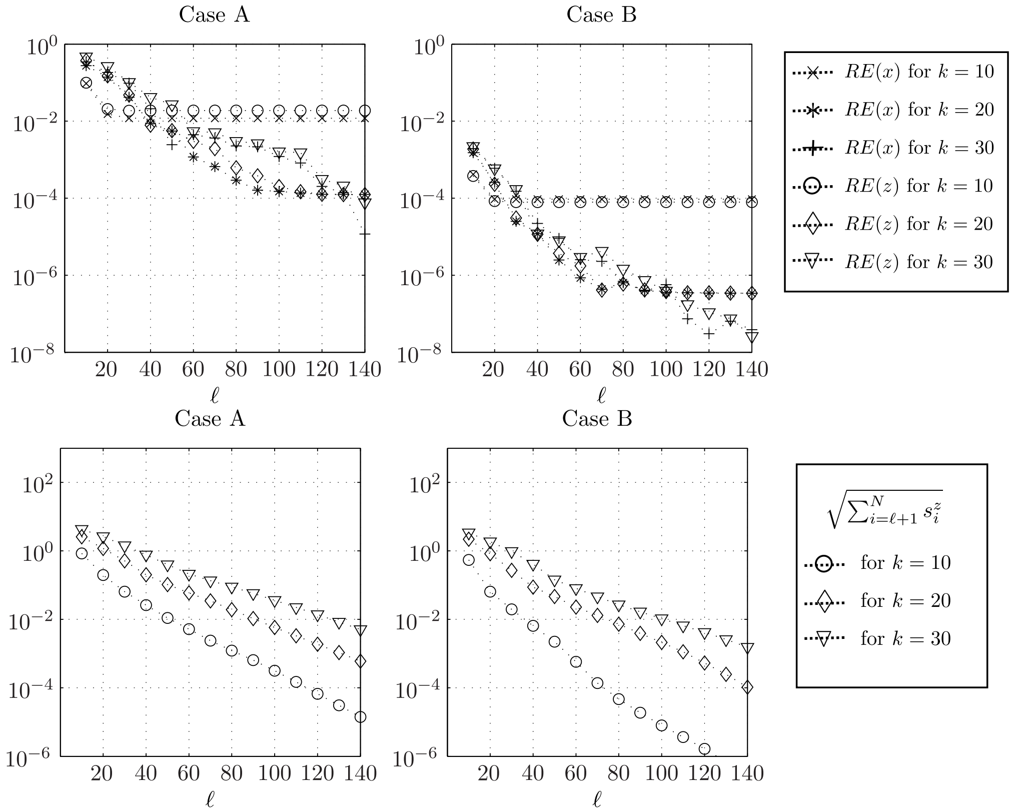

Figure 4.4.: Reconstruction results for DEDEIM projection for different reference DEPOD models with $k = 10, 20, 30$ (top) and corresponding projection errors (bottom).

Adjoint DEPOD/DEDEIM reconstruction

In a next test we compare the reconstruction behavior of (PD-ROM) where DEPOD and a DEDEIM basis are used for different numbers of k and ℓ. The results are displayed in Figure 4.4. In the upper half we compare reconstruction errors for states and adjoints for both cases and in the bottom half the corresponding square root of the projection errors are plotted. As before we carry out the test for different (P-ROM) reference models ($k =$ 10, 20, 30). First we observe in analogy to Figure 3.3 that the errors decay exponentially and after a certain threshold for ℓ the reconstruction cannot be improved further, as the POD related part of the reconstruction error starts to dominate. Also states and adjoints behave alike. In contrast to pure state reconstruction we now have different decay behaviors of the DEIM projection errors as the information in the DEIM snapshot matrix R_z^ℓ depends on the number of POD basis functions k. For large numbers of k we have more slowly decaying eigenvalues and a higher accuracy demand on the DEIM projection error, resulting in a fast growing number of DEIM basis functions that are necessary to achieve the best possible reconstruction. Note that also derivative information with respect to the controls is included in the DEIM basis (for POD approximations of the objective), which would not be needed for the approximation of the adjoints. In comparison, for state reconstruction the projection error already is close to machine precision with $\ell \approx 48$ (case A) and $\ell \approx 36$ (case B). However, a satisfying reconstruction accuracy of, e.g., 10^{-4} is obtained for $k = 30$, $\ell = 130$ (case A) and $k = 10$, $\ell = 20$ (case B) which would be good enough for many practical applications. Recalling the runtime results for a (PD-ROM) in Figure 3.4, the evaluation costs in both cases are still much cheaper in comparison to a HiFi evaluation. Note also that while the reconstruction results of cases A and B are significantly different, the decay of the eigenvalues is of comparable magnitude. Thus, finding appropriate thresholds to determine the number of POD and DEIM basis functions is highly problem dependent.

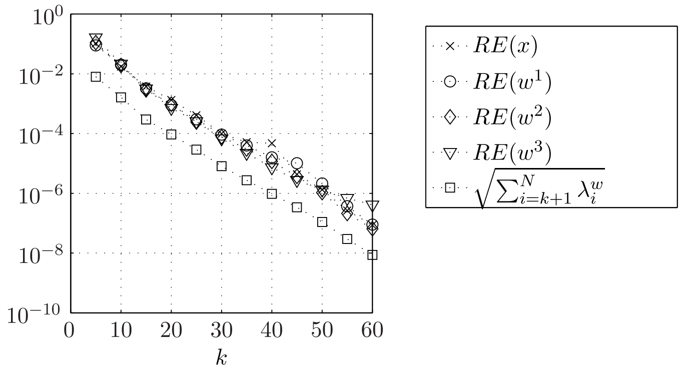

Figure 4.5.: Reconstruction results for DEPOD projection for the sensitivity case and the corresponding projection error for case A.

Sensitivity DEPOD reconstruction

In Figure 4.5 we display the results of reconstruction for states and sensitivities with respect to the three controls q_1, q_2 and q_3, restricting ourselves to case A. For the VDE solutions w^1, w^2 and w^3 the same norm as for the states above is used. Like for the adjoint case, states and derivatives decay in a similar way. In the sensitivity case to obtain a reconstruction accuracy of about 10^{-4}, now approximately $k = 30$ basis functions are necessary instead of $k = 20$ for adjoint DEPOD. This is to be expected as we need to add as many snapshot sets as controls (here three).

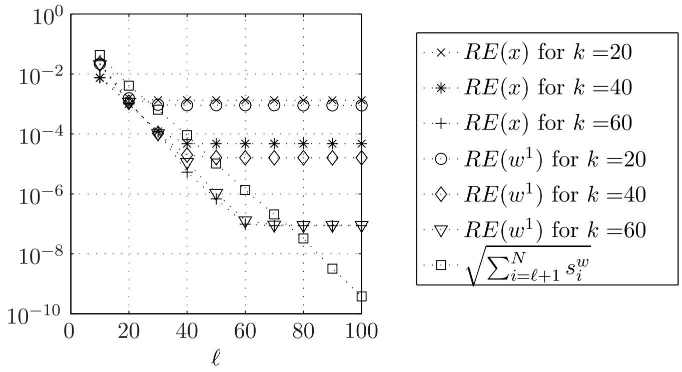

Figure 4.6.: Reconstruction results for DEDEIM projection with sensitivity-extended basis and the corresponding projection error for case A.

Sensitivity DEPOD/DEDEIM reconstruction

For the sensitivity case reconstruction errors for DEDEIM projection are also compared. The results for states and the first VDE variable w^1 are given in Figure 4.4 (w^2 and w^3 behave similarly and are not shown). The decay of the eigenvalues is now independent of the number of POD basis functions used in the reference POD reduced-order model as for the pure state reconstruction. Thus, relatively few DEIM basis functions are necessary, e.g., 10^{-4} is achieved with approximately $\ell = 30$ (compare to $\ell \approx 120$).

Comparison of different types of POD/DEIM basis

Further, we investigated the behavior of the projection error $\mathcal{E}_x$ and the reconstruction error $RE(x)$ in dependency of the type of basis used (POD, DEPOD with adjoints, DEPOD with sensitivities). The number of basis functions is chosen using a threshold ϵ_P (see also Remark 3.2), i.e., we choose k for the POD basis such that

$$\sqrt{\sum_{i=k+1}^{N} \lambda_i} < \epsilon_P$$

and analogously for λ_i^z, λ_i^w for DEPOD. We consider case A and display the results in Figure 4.7. As we expect, the projection and reconstruction errors are almost the same respectively for all three basis choices respectively. We observe that the reconstruction errors for states in case of DEPOD with sensitivities are actually slightly smaller, which is in accordance to Remark 4.2. The results show that the inclusion of derivative information has no compromising effect on the POD reconstruction quality of the states.

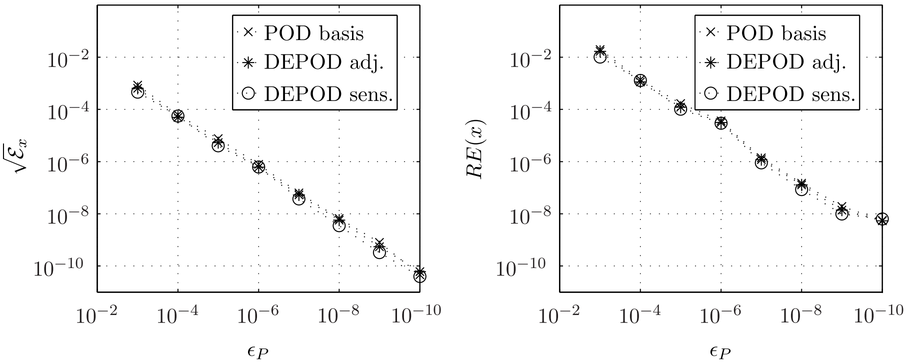

Figure 4.7.: Projection and reconstruction errors of the states for different types of POD basis in dependency of the criterion ϵ_P to choose the size of the basis.

Reconstruction results for reduced objective and gradient

Finally, we turn towards the optimization perspective and show results for the reconstruction error of the reduced objective and its gradient, restricting ourselves to case A once again. In Figure 4.8 we display the reconstruction results where we use the 2-norm for either quantity. In the graphic we compare the decay of the reconstruction error for different POD bases used to obtain (P-ROM). For the POD basis without derivatives and the adjoint DEPOD basis an exponential decay up to a certain best possible accuracy is observed. For pure POD the decay is fastest for the objective, however, we observe that the error in the gradient does not get smaller than approximately 10^{-2}. Obviously, the POD ROM is not able to reflect the derivative information properly. As one would expect the decay for both objective and derivatives is smaller with an adjoint enhanced basis in comparison to the sensitivity case. Here we observe the superior properties of the the adjoint-extended basis, as in this case both enhancement strategies are used for reconstruction of the same quantity of interest.

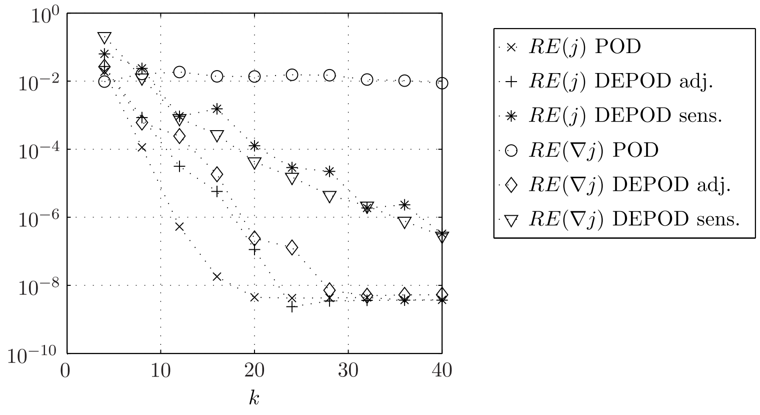

Figure 4.8.: Reconstruction results for the objective and the gradient with respect to the three controls q_1, q_2 and q_3 for case A.

In the last figure we show the reconstruction results for objective and gradient in dependence of the number of DEIM basis functions used. We give again only the results for case A and restrict ourselves to a POD reference basis with $k = 20$ as the other scenarios behave alike. As before for the POD projection we have a fast decay for the objective if no derivatives are used while the reconstruction error for the gradient does not get smaller than, in this case, approximately 0.1. As for the state reconstruction the errors related to the DEIM projection decay faster for a sensitivity-enhanced basis in comparison to the adjoint enhancement. For adjoint DEDEIM with roughly $\ell = 140$ basis functions we are close to the best possible accuracy determined by the POD surrogate model without derivative enhancement. The actual limit cannot be achieved even for much larger ℓ due to the basis built from HiFi information instead of information of (P-ROM) (see also Remark 3.12). Moreover, note that while we choose different derivative enhancement strategies, in either case the gradient is evaluated using adjoint IND. Due to its AD character the results are identical to an evaluation of the gradient with the sensitivity approach and IND. Thus, we can also confirm the assertion of Remark 4.6 numerically. Moreover, we observe an oscillating behavior for varying ℓ which we explain by the nonlinear affects, as the plots become smoother when reducing the nonlinearity (not shown explicitly).

We conclude from the numerical results that we are able to construct (PD-ROM) using DEPOD and DEDEIM subspaces without increasing the size of the necessary basis functions too much to achieve a practically satisfying accuracy. In the adjoint case we observe that less than 20 POD and less than 100 DEIM basis functions are enough to achieve an accuracy of about 10^{-4} for the reconstruction error of the gradient. In the sensitivity case this is achieved for approximately 20 POD and about 30 DEIM basis functions. Comparing this to the runtime results of §3.3.5, Figure 3.4, the values are clearly small enough to obtain a significant reduction in the evaluation costs of the objective and the gradient. Moreover, we have shown numerically that the inclusion of additional information does not harm the standard reconstruction properties of POD/DEIM, besides the fact that the basis has to be increased. We have shown that the size of the basis can be chosen adaptively such that

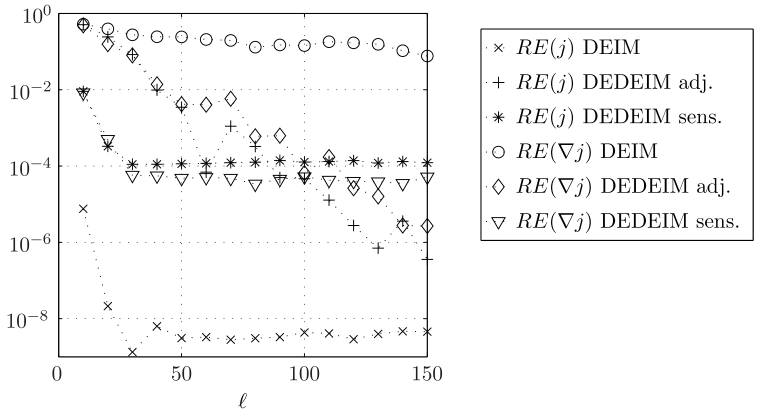

Figure 4.9.: Reconstruction results for DEIM projection of the objective and the gradient with respect to the three controls q_1, q_2 and q_3 for case A and a POD basis with $k = 20$.

for a certain criterion ϵ_P the reconstruction accuracy of a POD/DEIM surrogate without derivatives can again be achieved with DEPOD/DEDEIM.

5. POD/DEIM Reduced-Order Models in Optimization

In this section we present novel algorithms to solve optimal control and parameter estimation problems. We exploit the results of the previous sections and analyze the distinguished properties of the approaches. An aspect that is of paramount importance is the improvement of POD prediction using DEPOD and DEDEIM subspaces (see Definition 4.9) for the construction of the surrogate models, which we discuss in §5.1. In addition, we present an a posteriori error estimate for parameter estimation problems, which can be used to evaluate the distance between a suboptimal solution obtained via the surrogate model and the solution of the high-fidelity (HiFi) optimization problem. The latter result is published by the author of this thesis in [104] and it is included here for completeness.

In literature we find multiple employments of POD in optimization with the aim of reducing the computational costs. A general overview is given by Sachs [99]. POD in the context of optimal control for particular applications is found in [8, 79, 106]. Below several articles containing concepts for optimal control problems, typically of the general linear-quadratic type, are briefly assessed and their relation to our work is discussed. There is also a variety of applications of POD in the context of inverse problems [12, 47, 68, 120, 122]. While the motivations in inverse and parameter estimation problems are similar, inverse problems typically contain a distributed parameter space. Hence, they are often handled similarly to optimal control problems using an adjoint approach for their solution. A parameter estimation approach related to this thesis was taken in [80] where a Gauss–Newton method is also employed to solve the least-squares problem. Other examples of applications of POD to estimate scalar parameters are, e.g., [67, 97]. Application of DEIM is getting popular in the MOR context, however, considering the recent appearance of this method, only little experience is available in combination with optimization (see [26, 81]).

With POD applied to an optimization problem in a straightforward way, the authors often observe an outstanding performance with the model reduction approach. If the HiFi model is evaluated only once and the optimization process, based on one fixed surrogate model, is able to find the solution of the original problem, tremendous runtime improvements can be made. Under the assumption that the subspaces V^k and R^ℓ constructed from a particular snapshot set are sufficiently rich for the optimization purpose, the approach is powerful. This assumption is equivalent to the fact that we assume a good POD prediction of the (P-ROM) or (PD-ROM) for relevant control configurations, as defined in §3.3.1. In practice we often have a priori knowledge on the process which can be exploited in the POD approach for optimization, e.g., [17].

However, the assumption on the subspaces is often not satisfied. Due to the lack of reliable a priori estimation for POD, it is not clear how to enhance the space in general to satisfy the assumption. Several strategies have been suggested to overcome these difficulties which are directly related to the POD prediction problem. There are conceptionally two major ways to tackle this problem. One strategy is to enhance the model such that good POD predictions are possible for a certain region of control variable configurations. This might require a large effort to construct an appropriate basis. Hence, the procedure is commonly separated into an offline and an online phase. A second conceptual strategy is to recompute the HiFi solutions and adapt the model over time, thus, breaking the clear separation between offline and online phase. Typically, a suboptimal solution via the reduced-order model is computed and then either the optimization step is repeated with the suboptimal solution as new initial guess or other post processing steps are performed (see, e.g., [1, 61, 77, 115, 122]). If one is interested in a long-term use of the surrogate model with fast simulations in a certain control domain of interest, the first strategy should obviously be used. Recalling Remark 4.1 in optimization we are, however, not willing to invest too much time in the construction of the model. Our methods combine aspects of both approaches.

We already mentioned the beneficial effects of the inclusion of derivative information in the reduced basis on the POD prediction problem at the beginning of §4.2. In §5.1 we give an estimate for POD prediction of the reduced objective and of the states which is based on the use of DEPOD and DEDEIM bases. The estimate makes assertions for the local behavior of DEPOD/DEDEIM surrogates, i.e., the analysis is valid for perturbations in the data of which the model is constructed from. To our knowledge the prediction estimate is the only existing one so far for reduced-order models based on POD and DEIM. While the issues are closely related, note that our prior motivation to include derivatives is to enhance the reduced-order model with the ability to approximate not only state, but also derivative information. In [65, 105] error estimates for POD prediction with standard POD surrogate models are presented. Using an adjoint analysis, the authors are able to define so called *regions of validity* around a given reference configuration, where the model is constructed. To define these regions several solutions of the adjoint of the HiFi problem are necessary, hence, one must decide whether the additional effort pays off in optimization. Our results show that for a variety of applications, only few HiFi solutions are necessary to solve the optimization problem. Other approaches to tackle the POD prediction problem are described in [94] where POD is combined with a dynamic iteration procedure, in [4] where interpolation techniques for POD subspaces are applied, and in [86] where concepts of dual-weighted-residual error estimation are used to measure the difference in the objective, evaluated with the HiFi and surrogate model.

Among the adaptive strategies we find [1], where after determining a suboptimal solution the HiFi model is reevaluated and the state vectors are added to the snapshot matrix $\mathcal{S}$ and (P-ROM) is rebuilt. A similar algorithm is used in [61]. The adaption strategy in [64] is to use a few POD modes of the old basis when the basis is reconstructed at the suboptimal solution. In any of the proposed algorithms the stopping criterion is to check whether a suboptimal solution $\widehat{u}^\star_{(n)}$ obtained in iteration n has changed significantly in comparison to $\widehat{u}^\star_{(n-1)}$, i.e., test if

$$\left\| \widehat{u}^\star_{(n)} - \widehat{u}^\star_{(n-1)} \right\|_{\mathcal{Q}} < TOL. \tag{5.1}$$

A related article is [96], where an SQP algorithm is used to solve the optimization problem in an all-at-once fashion, updating the POD basis in each SQP step. In [6, 42, 73] trust region methods are presented which the authors denote by *trust region POD* (TR-POD) and which we discuss more thoroughly below.

Recently a priori and a posteriori error estimates were presented for POD and linear-quadratic optimal control problems. The estimates indicate how close a suboptimal control solution is to the solution of the original problem. Hinze and Volkwein [62] derive a priori estimates, but the POD basis is computed utilizing the optimal solution obtained via the HiFi problem. This drawback is overcome by the a posteriori analysis in [112] by Tröltzsch and Volkwein. Their estimates are related to the a posteriori analysis we present in §5.3.1.

Several extensions to the topic can be found. In [81] a priori error estimates are presented for the POD Galerkin schemes for nonlinear elliptic-parabolic systems. In [71] a posteriori estimates similar to [112] are given where second-order information is used. The authors in [69] apply the technique to bilinear elliptic optimal control problems solved by an inexact SQP method, where the inexactness originates in the approximation by the (P-ROM). In optimal control algorithms used in combination with the mentioned a posteriori estimates, typically the number of POD basis functions is increased according to the estimate. We have seen before that this strategy might destroy the efficiency of the reduced-order modeling approach. We investigate this in §5.4.1.

In [77] Kunisch and Volkwein tackle the POD prediction problem by *optimality systems POD*. The reduced optimal control problem (analogous to (5.5)) is extended by additional constraints, namely the original HiFi problem (2.7) and the condition for the construction of the reduced problem (3.4). To regain efficiency an operator splitting is performed into reduced-order and HiFi variables. The algorithm yields a correction of the suboptimal solution proposed with (P-ROM) into the direction of the true HiFi solution. In [115] the OS-POD idea is combined with the a posteriori error estimates in [112]. A related goal-oriented approach to construct the POD basis was taken in [28] where the authors consider a POD optimality similar to (3.1) that contains a quantity of interest and the governing equations as constraints for a set of parameter configurations. However, the POD basis is quite costly to compute, i.e., the HiFi problem must be solved for each parameter configuration in each CG iteration which is used to solve the extended POD optimality problem.

We start presenting the POD prediction estimate and then describe particular algorithms to solve optimal control and parameter estimation problems. In §5.4 we discuss POD MOR from the perspective of discretize-then-optimize vs. Optimize-then-Discretize, slightly altering the naming convention there to adapt to the peculiarities.

5.1. POD prediction for derivative-extended reduced-order models

In this section we analyze the POD prediction properties of surrogate models with DE-POD/DEDEIM basis. We show that we obtain local first-order accurate approximations of the objective evaluated with the HiFi model given that the reduced-order models are constructed as proposed in Chapter 4. The results can also be interpreted as a local robustification with respect to the data that the surrogate model is constructed from. To our knowledge the estimates are the first of its kind for the POD prediction problem in the time-dependent case.

We extend the discussion in [30], carried out for stationary problems in the context of interpolation based POD. The result is summarized in the following theorem, where we establish an estimate for the POD prediction error of $j^h(q)$ in a neighborhood of a given control configuration $\bar{q}$ for adjoint enriched surrogates in the time-continuous setting.

Theorem 5.1. *For a reference control $\bar{q} \in \mathcal{Q}^h$ where* (PD-ROM) *is constructed let the reconstruction estimate for the reduced objective* (4.1) *and its gradient* (4.37) *hold. Assume $\mathcal{Q}^h$ to be convex. Then there is a constant $\bar{C}_{Tz}$ such that for every $q \in \mathcal{Q}^h$ the POD prediction error of the objective can be estimated as*

$$PE\left(j^h(q)\right) \leq \bar{C}_{Tz}\left(\sum_{i=k+1}^{N} \bar{\lambda}_i^z + \sum_{j=\ell+1}^{N} \bar{s}_j^z\right) \|q-\bar{q}\|_{\mathcal{Q}^h} + \mathcal{O}(\|q-\bar{q}\|_{\mathcal{Q}^h}^2). \tag{5.2}$$

Proof. Due to the convexity of $\mathcal{Q}^h$ we can write down the Taylor approximations of the objectives $j^h(q)$ and its POD approximation $\widehat{j}^h(q)$ for any $\bar{q}$. Using $h := q-\bar{q}$ and $c_0 := 1/\left\|j^h(q)\right\|$ we obtain

$$\begin{aligned} c_0\left\|j^h(q)-\widehat{j}^h(q)\right\| &= c_0\left\|j^h(\bar{q})+j_q^h(\bar{q})h-\widehat{j}^h(\bar{q})-\widehat{j}_q^h(\bar{q})h+\mathcal{O}(\|h\|_{\mathcal{Q}^h}^2)\right\| \\ &\leq c_0\left\|j^h(\bar{q})-\widehat{j}^h(\bar{q})\right\|+c_0\left\|j_q^h(\bar{q})h-\widehat{j}_q^h(\bar{q})h\right\|+\mathcal{O}(\|h\|_{\mathcal{Q}^h}^2) \\ &\leq \bar{C}_{Tz}\left(\sum_{i=k+1}^{N} \bar{\lambda}_i^z + \sum_{j=\ell+1}^{N} \bar{s}_j^z\right)\|h\|+\mathcal{O}(\|h\|_{\mathcal{Q}^h}^2). \end{aligned}$$

For the last inequality the reconstruction error (up to the normalizing constant) of the gradient is estimated according to Theorem 4.10. For the error in the objective the reconstruction estimate (4.1) is used, noting that it still applies with the eigenvalues $\bar{\lambda}_i, \bar{s}_j$ obtained from the extended optimalities (see also Remark 4.2). q.e.d.

The time-discrete case for the adjoint approach can be carried out analogously with eigenvalues $\bar{\lambda}_i^w, \bar{s}_j^w$ instead of $\bar{\lambda}_i^z, \bar{s}_j^z$ and $q \in \mathcal{Q}^{h\tau}$.

In the sensitivity case, we can even give a POD prediction estimate for the states. Again we need to restrict the possibly infinite-dimensional space $\mathcal{Q}^h$ to the finite dimensional space $\mathcal{Q}^{h\tau}$ already in the time-continuous setting. Note that in practice we consider the sensitivity case only for parameter estimation problems where the dimension of the optimization variable space is assumed to be small. We assume now that there are Fréchet differentiable solution operators $q \mapsto y^h(q)$ and $q \mapsto y^k(q)$ with $y^h(q), y^k(q) \in C^1(I, V^h)$ that can be obtained by solutions of the HiFi (2.7) and surrogate model (PD-ROM) for a given control $q \in \mathcal{Q}^{h\tau}$ respectively.

Theorem 5.2. *Let the set of directions $\tilde{q}_1, \dots, \tilde{q}_{n_q}$ form a basis of $\mathcal{Q}^{h\tau}$. Assume that the corresponding VDE solutions $w^1(t), \dots, w^{n_q}(t)$ and $\widehat{w}^1(t), \dots, \widehat{w}^{n_q}(t)$ of* (4.9) *and* (4.11) *and solutions $x(t)$ and $\widehat{x}(t)$ of* (2.7) *and* (PD-ROM) *satisfy the reconstruction estimates* (4.13) *and* (3.27) *for all $\tilde{q}_i$ at $\bar{q} \in \mathcal{Q}_0^{h\tau} \subseteq \mathcal{Q}^{h\tau}$, $\mathcal{Q}_0^{h\tau}$ open. If Ψ, Φ, $\lambda_1^w, \dots, \lambda_N^w$, and $s_1^w, \dots, s_N^w$ are computed according to the extended POD optimalities* (4.23) *and* (4.34), *then there is a constant $\bar{C}_{Jw}$ such that for every $q \in \mathcal{Q}_0^{h\tau}$*

$$\int_I \left\| y^h(q) - y^k(q) \right\|_H dt \le \bar{C}_y \left(\sum_{i=k+1}^{N} \bar{\lambda}_i^w + \sum_{j=\ell+1}^{N} \bar{s}_j^w \right) \|q - \bar{q}\|_{\mathcal{Q}^{h\tau}} + \mathcal{O}(\|q - \bar{q}\|_{\mathcal{Q}^{h\tau}}^2). \quad (5.3)$$

Proof. Due to the differentiability of the solution operators $y^h(q)$ and $y^k(q)$ we can write down their Taylor expansion as functions from $\mathcal{Q}_0^{h\tau}$ to $C^1(I, V^h)$. Using $h = q - \bar{q}$ we get

$$\begin{aligned} &\int_I \left\| y^h(q) - y^k(q) \right\|_H dt \\ &= \int_I \left\| y^h(\bar{q}) - y^k(q) + y_q^h(\bar{q})h - y_q^k(\bar{q})h + \mathcal{O}(\|q - \bar{q}\|_{\mathcal{Q}^{h\tau}}^2) \right\|_H dt. \end{aligned}$$

Here $y_q^h(\bar{q})h$ and $y_q^k(\bar{q})h$ are the derivatives of the solution operators in the direction h. These can be expressed by sensitivities obtained as solutions of the corresponding VDE problems for given $\bar{q}$. As the directions used to build the DEPOD/DEDEIM subspaces form a basis of $\mathcal{Q}^{h\tau}$, the direction h and with this the difference between $y_q^h(\bar{q})h$ and $y_q^k(\bar{q})h$ can be decomposed such that

$$y_q^h(\bar{q})h - y_q^k(\bar{q})h = \sum_{i=1}^{n_q} \left(y_q^h(\bar{q}) - y_q^k(\bar{q}) \right) c_i \tilde{q}_i,$$

with coefficients $c_1, \dots, c_{n_q} \in \mathbb{R}$. Applying the reconstruction estimate in Theorem 4.7 to the decomposition of the directional derivatives and the reconstruction estimate in Theorem 3.6 to the difference in the states, the assertion follows immediately. q.e.d.

The time-discrete case can be obtained analogously. Moreover, a POD prediction estimate for the objective in (5.2) follows directly for sensitivity enhanced DEPOD/DEDEIM via the reconstruction results of Chapter 4.

DEPOD/DEDEIM bases enhanced with sensitivity information contain more information in comparison to an enhancement with adjoints, as we additionally obtain a POD prediction estimate for the states. The result is relevant for the solution of the parameter estimation problem via Gauss–Newton, where we need to approximate the function $\mathcal{F}(q)$, i.e., approximations of the states

$$y^h(\tilde{t}_1; q), \ldots, y^h(\tilde{t}_{n_{\text{meas}}}; q).$$

However, the advantage comes at the cost that we possibly need to solve many more HiFi problems than we would have to in the adjoint case, where only one adjoint problem has to be solved. With regard to the underlying optimization problem the adjoint DEPOD/DEDEIM enhancement can be interpreted as a goal-oriented approach, as we only extend the model to give good approximations for a quantity of interest (the objective in the optimal control case) in a region around the reference configuration.

5.2. A DEPOD algorithm for optimal control

The aim of this section is to present an algorithmic framework for an efficient solution of the optimal control problem (OCP), transformed into an NLP as described in Chapter 2. I.e., we want to solve

$$\min_{q \in \mathcal{Q}_{ad}^{h\tau}} \quad j^{h\tau}(q) \;=\; J^{h\tau}(x(q), q), \tag{5.4}$$

where $x(q)$ is determined such that it satisfies the HiFi model equation (2.7). We seek to gain efficiency by considering the same problem, however, approximating $x(q)$ using the surrogate, i.e., solving

$$\min_{q \in \mathcal{Q}_{ad}^{h\tau}} \quad \widehat{j}^{h\tau}(q) \;=\; J^{h\tau}(\Psi \widehat{x}(q), q), \tag{5.5}$$

with $\widehat{x}(q)$ being a solution to (PD-ROM). We use an SQP algorithm, as described in §2.4.1, to solve each instance of optimal control problem, i.e., the HiFi and (PD-ROM) for different $\bar{q}$ where the model is constructed. Throughout we assume that there exists a solution $q^\star$ of (5.4) and solutions $\widehat{q}^\star$ of (5.5) for each surrogate model instance. We use DEPOD/DEDEIM with adjoint information. For the computation of the gradient of either (5.4) or (5.5), the adjoint techniques for derivative computation are used as described in §2.3 and §2.5. We take advantage of the superlinear convergence properties of the SQP method with BFGS updating. Note, however, that with the proposed concepts any algorithm that is able to solve (5.4) could be used (see also Remark 4.6).

We assume that the time discretization uses as many snapshots as necessary to capture the essential dynamics and that a sufficiently accurate time integration scheme is employed. Recalling the main influences for POD reconstruction in Remark 3.6, the occurring reconstruction errors essentially depend on the POD/DEIM eigenvalues.

We present a *DEPOD optimal control* (DEPOD-OC) algorithm which is closely related to the adaptive algorithms in [1, 61]. Two essential differences are the extended POD and DEIM basis for the reduced-order model and the stopping criterion. For a particular implementation of Algorithm 2 further specification of some steps is necessary which we discuss now. The algorithm consists of two levels, the major iterations which account for the problem of POD prediction, and the inner optimization loop where the reduced optimal control problem is solved using the surrogate model. Conceptually, one can compare this approach to an SQP-like method, where instead of a quadratic approximation of the objective, an approximation based on the surrogate is used. On the one hand, according to Theorem 5.1 we have first-order approximation quality for the surrogate approximation of the objective up to a reconstruction accuracy, while for SQP with, e.g., exact Hessian we have second order. On the other hand, estimate (5.2) is only an upper bound on the prediction error. In fact, the approximation will usually be much better if the DEPOD/DEDEIM

Algorithm 2 DEPOD-OC

```
Require: $q_{(0)} \in \mathbb{R}^{n_q}$
 1: for $n = 0, 1, \ldots$ do
 2:     Compute $\mathcal{S}_z$ and $\mathcal{D}_z$ as in (4.19) and (4.31) at $q_{(n)}$
 3:     Build the surrogate model using $\mathcal{S}_z$ and $\mathcal{D}_z$
 4:     Choose $k$ and $\ell$ such that $\sqrt{\lambda^z_{k+1}} < \lambda_{TOL}$ and $\sqrt{s^z_{\ell+1}} < s_{TOL}$
 5:     if stopcrit($q_{(n)}$) $\leq TOL_n$ then
 6:         $\widehat{q}^\star \leftarrow q_{(n)}$
 7:         Break
 8:     else
 9:         Solve (5.4) using the surrogate model to obtain $\widehat{q}^\star$
10:         $q_{(n+1)} \leftarrow \widehat{q}^\star$
11:     end if
12: end for
return $\widehat{q}^\star$
```

subspaces allow approximations of higher order derivatives as well. When the subspaces are sufficiently rich, then the POD surrogate may even provide good approximations in the whole domain of interest $\mathcal{Q}^{h\tau}_{ad}$. As described at the beginning of the chapter this is often the case in practice.

In line 5 of the DEPOD-OC algorithm we use the stopping criterion stopcrit($q_{(n)}$) corresponding to the convergence criterion of the Newton-type method employed. It is typically based on the optimality criterion (1.8), which itself depends on the gradient $\nabla j^{h\tau}$ at $q_{(n)}$ in the unconstrained case. However, the stopping criterion used in a particular NLP solver might not be available for the user. Hence, we suggest to use TOL_n as tolerance for the Newton-type algorithm in the inner loop and check if it returns successfully in one iteration. If we can guarantee a sufficiently good local approximation of the HiFi model via the surrogate model, convergence of Algorithm 2 is equivalent to $\widehat{q}^\star$ being a local optimum of the original HiFi problem. This property stands in contrast to the common stopping criterion (5.1) in POD algorithms for optimal control, as a priori there is no guarantee that a small change in the control means that we are close to the solution. Note that in the optimal control case we need an extra evaluation of the HiFi model to carry out the final convergence check.

Given the choice of the stopping criterion, the local POD prediction quality of the reduced objective at iterate $q_{(n)}$ is of paramount importance. Due to the POD prediction estimate in Theorem 5.1 it becomes clear that we need to choose the tolerance criteria λ_{TOL} and s_{TOL} in line 4 properly (compare also to Remark 3.2). Under the assumptions on the time integration and the number of snapshots, we have first-order approximation quality if k and ℓ are large enough. The choices of λ_{TOL} and s_{TOL} are problem dependent. For our numerical computations we rely on experience to choose the tolerances. However, one could afford to use an a posteriori control of the reconstruction error to choose k and ℓ, as evaluation costs of the surrogate model are often negligible in comparison to the overall costs. Note that the algorithm may converge if λ_{TOL} and s_{TOL} are chosen poorly or a conventional POD basis is used. We observe this later in the applications part. The reason is that if a suboptimal solution $\widehat{q}^\star$ is found and the POD/DEIM bases are rebuilt, the new surrogate model may still have the same structure as the previous one and the inner loop will successfully return in one step. However, as for general POD surrogates we have no a priori knowledge about its local behavior we do not know if a good approximation of the HiFi solution was found.

In line 9 we apply an SQP algorithm. Due to the updating of the Hessian approximation we follow the strategy to do as many iterations as necessary to obtain a solution satisfying the termination tolerance of the inner loop. Regarding global and local convergence of the inner optimization, this can be efficiently handled by common NLP techniques. Due to the derivative computation based on automatic differentiation, for any $q \in \mathcal{Q}_{ad}^{h\tau}$ the gradient $\nabla \widehat{j}^{h\tau}(q)$ is an accurate approximation of the gradient of $\widehat{j}^{h\tau}(q)$, independent of the particular surrogate model used.

Regarding overall convergence of the DEPOD-OC algorithm several possible enhancements can be found in literature. A possibility is the use of information from the previous reduced basis similar to [1, 64] to prevent cycling. An alternative is the TR-POD, which we discuss briefly here. In [6] TR-POD is presented together with a convergence result for optimal control problems using POD reduced-order models. The authors use the assumption

$$\frac{\left\|\nabla j^{h\tau}(q) - \nabla \widehat{j}^{h\tau}(q)\right\|}{\left\|\nabla \widehat{j}^{h\tau}(q)\right\|} \leq c_{\text{TR}} \tag{5.6}$$

for some user-defined constants $c_{\text{TR}} > 0$. In general, this cannot be guaranteed for standard reduced-order models. However, (5.6) is basically the reconstruction error $RE\left(\nabla j^{h\tau}(q)\right)$ for the gradient as defined in §3.3.1, only differing in the normalizing term where the surrogate approximation is used instead. Thus, due to Theorem 4.10 it can be satisfied for DEPOD/DEDEIM. Hence, the convergence result of TR-POD could immediately be carried over to Algorithm 2 (for details see [6, 73]). However, due to our numerical experience with DEPOD/DEDEIM we do not recommend to restrict the search region at first hand, as the efficiency of the reduced-order modeling approach might be significantly reduced.

We briefly recall the a posteriori estimate in [112] and adapt it to our situation. To this end, assume (2.17) to result from a linear-quadratic optimal control problem and let $q^\star$ be its optimal solution. If $\widehat{q}^\star$ is a suboptimal solution obtained via (P-ROM) (without DEIM projection), then one can show that

$$\|q^\star - \widehat{q}^\star\| \leq \frac{1}{\gamma} \|\widehat{\zeta}\|, \tag{5.7}$$

for some perturbation $\widehat{\zeta} \in \mathbb{R}^{n_q}$ that is computed via solutions of the HiFi model and its adjoint at $\widehat{q}^\star$. Given that the snapshots form a basis of V^h one has $\widehat{\zeta} \to 0$ for $k \to N$. The authors in [112] use the estimate to solve a linear-quadratic optimal control problem, evaluating the quality of a suboptimal solution and subsequently increasing the number of POD basis functions if necessary.

In our approach, we do not follow this strategy for the following reasons: To evaluate the estimate (5.7) a HiFi state plus adjoint solution is necessary. Our results suggest that, if state and adjoint information is available at a suboptimal solution $\widehat{q}^\star$, the information should be used or added to the POD basis. In [69] convergence to the full-order problem using this strategy is shown. In case of rebuilding the basis completely, the convergence result cannot be applied anymore as the problem changes in every major iteration. However, our numerical results indicate that the DEPOD approach is in terms of efficiency superior to an increase of the number of basis functions.

5.3. A DEPOD algorithm for parameter estimation

In this section we present a *DEPOD parameter estimation* (DEPOD-PE) algorithm to solve the reduced parameter estimation problem (1.23) as described in §1.4. Again we compare the problems with the states being once approximated by $x(q)$ as in (2.31) and once by the POD approximation $\Psi\widehat{x}(q)$, i.e., solving

$$\min_q \frac{1}{2}\left\|\widehat{\mathcal{F}}(q)\right\|^2, \quad \widehat{\mathcal{F}}_i(q) := \frac{\eta_i - h(y^k(\tilde{t}_i; q), u)}{\varsigma_i}, \quad i = 1, \ldots, n_{\text{meas}}. \tag{5.8}$$

In the surrogate models we use a DEPOD/DEDEIM basis with sensitivity information. Again we assume a sufficiently accurate time integration and a sufficiently large number of snapshots. The solution of (2.31) and each instance of (5.8) is based on the Gauss–Newton method presented in §2.4.2. As above we assume that a solution of each problem instance exists. Due to the need of computing the vector $\mathcal{F}(q)$ and the Jacobian $\mathcal{J}(q)$ in the Gauss–Newton algorithm, the sensitivity approach is more efficient under the assumption $n_q < n_m$. Even though the concepts are independent of the number of parameters, for a practical concern we think of a moderate number, e.g., $n_q \leq 20$.

Algorithm 3 DEPOD-PE

```
Require: $q_{(0)}$
     for $n = 0, 1, \dots$ do
 2:      Compute $\mathcal{S}_w$ and $\mathcal{D}_w$ as in (4.24) and (4.35) at $q_{(n)}$
         Build surrogate model using $\mathcal{S}_w$ and $\mathcal{D}_w$
 4:      Choose $k$ and $\ell$ such that $\sqrt{\lambda^w_{k+1}} < \lambda_{TOL}$ and $\sqrt{s^w_{\ell+1}} < s_{TOL}$
         $\widehat{q}^{(0)} \leftarrow q_{(n)}$
 6:      for $i = 0, 1, \dots$ do
             Compute $\Delta q$ from (2.33) using the surrogate model
 8:          $\widehat{q}^{(i+1)} \leftarrow \widehat{q}^{(i)} + \Delta q$
             if $\|\Delta q\| \leq TOL_{\mathrm{GN}}$ then
10:              $\widehat{q}^{\star} \leftarrow \widehat{q}^{(i+1)}$, break
             end if
12:      end for
         if $\left\|\widehat{q}^{\star} - q_{(n)}\right\| \leq TOL_n$ then
14:          break
         else
16:          $q_{(n+1)} \leftarrow \widehat{q}^{\star}$
         end if
18:  end for
return $\widehat{q}^{\star}$
```

Algorithm 3 follows the basic structure of the DEPOD-OC Algorithm 2. We only comment on the differences. In the parameter estimation case we treat the inner loop iterations explicitly. For the sake of simplicity, here we do not consider any globalization techniques (e.g., restrictive monotonicity test [22] or a line search) and assume that we are in the area of local convergence of the Gauss–Newton method. Strategies for the particular implementation are discussed in chapter 6.

In contrast to the DEPOD-OC algorithm, the stopping criterion is based on the change of the parameter values. In Theorem (5.4) we give an a posteriori estimate which states that if Algorithm 3 terminates with the criterion as in line 13 then we have also found a solution of the HiFi problem up to a certain tolerance and reconstruction error. A direct check of the parameter change is desirable for parameter estimation problems, as the choice of the tolerance TOL_n should in practice be motivated by the statistical properties of the underlying problem, e.g., the variance of the parameter estimate. Moreover, using the stopping criterion as in Algorithm 3 saves one additional computation of HiFi information.

5.3.1. An a posteriori error estimate

In this section we make use of the local convergence properties of the Gauss–Newton method applied to parameter estimation problems (see also §2.4.3). Conceptually, we exploit the fact that the Gauss–Newton method only needs first-order derivative information of the current step. This is different, e.g., from SQP and BFGS updating which makes use of information of previous steps that enters the increments. The estimate is similar to the a posteriori estimate (5.7) for linear-quadratic optimal control problems.

If the solution operator is evaluated with the HiFi model we refer to this as *HiFi Gauss–Newton*, otherwise when the surrogate model is used we call it *POD Gauss–Newton*. The following lemma establishes an estimate for the quality of a solution $\widehat{u}^\star$ of POD Gauss–Newton, given a DEPOD/DEDEIM basis constructed from sensitivity information is used. For brevity we write

$$\begin{aligned}\Delta q(\bar q) &:= -\mathcal{J}^\dagger(\bar q)\mathcal{F}(\bar q),\\ \Delta \widehat{q}(\bar q) &:= -\widehat{\mathcal{J}}^\dagger(\bar q)\widehat{\mathcal{F}}(\bar q),\end{aligned}$$

for $\bar q \in \mathcal{Q}^{h\tau}$. These are the increments computed with the HiFi and the surrogate model at $\bar q$ respectively. Let $\widehat{\mathcal{Q}}_0^h$ be the contraction ball of POD Gauss–Newton as in the local contraction theorem.

Lemma 5.3. *For a given $\bar q$ at which the DEPOD/DEDEIM basis is built, assume that the conditions of Theorem 2.5 are satisfied for POD Gauss–Newton on $\mathcal{Q}^{h\tau}$ for $\widehat{q}_{(0)} = \bar q$. Let $\widehat{q}^\star$ be the limit point from Theorem 2.5 (2). Moreover, assume $\mathcal{J}^\dagger, \widehat{\mathcal{J}}^\dagger \in C^1(\widehat{\mathcal{Q}}_0^{h\tau})$. Then there is a constant $C(\bar q) < \infty$ independent of $\widehat{q}^\star$ such that we have*

$$\|\Delta q(\widehat{q}^\star)\| \leq \varepsilon_1 + (\varepsilon_2 + C(\bar q))\,\|\widehat{q}^\star - \bar q\| + \mathcal{O}(\|\widehat{q}^\star - \bar q\|^2) \tag{5.9}$$

where ε_1 and ε_2 are constants depending on the DEPOD reconstruction errors at $\bar q$.

Proof. As $\mathcal{J}^\dagger, \widehat{\mathcal{J}}^\dagger \in C^1(\widehat{\mathcal{Q}}_0^{h\tau})$ and $\widehat{\mathcal{Q}}_0^{h\tau}$ is compact we have $\left\|\frac{\mathrm{d}}{\mathrm{d}q}\mathcal{J}^\dagger(\bar q)\right\| < \infty$ and $\left\|\frac{\mathrm{d}}{\mathrm{d}q}\widehat{\mathcal{J}}^\dagger(\bar q)\right\| < \infty$. Furthermore, $\Delta\widehat{q}(\widehat{q}^\star) = 0$ holds in the solution of POD Gauss–Newton due to Theorem 2.5. A Taylor expansion of the Gauss–Newton increments at $\bar q$ (defining $h := \widehat{q}^\star - \bar q$ and omitting the argument $\bar q$ of $\mathcal{F}, \mathcal{J}^\dagger, \widehat{\mathcal{F}}$ and $\widehat{\mathcal{J}}^\dagger$) yields

$$\begin{aligned}
\|\Delta q(\widehat{q}^\star)\| =\ & \|\Delta q(\widehat{q}^\star) - \Delta\widehat{q}(\widehat{q}^\star)\|\\
\leq\ & \left\| -\left(\mathcal{J}^\dagger\mathcal{F} + \frac{\mathrm{d}(\mathcal{J}^\dagger\mathcal{F})}{\mathrm{d}q}h\right) + \left(\widehat{\mathcal{J}}^\dagger\widehat{\mathcal{F}} + \frac{\mathrm{d}(\widehat{\mathcal{J}}^\dagger\widehat{\mathcal{F}})}{\mathrm{d}q}h\right)\right\| + \mathcal{O}(\|h\|^2)\\
=\ & \left\|\left(\widehat{\mathcal{J}}^\dagger\widehat{\mathcal{F}} - \mathcal{J}^\dagger\mathcal{F}\right) + \left(\frac{\mathrm{d}\widehat{\mathcal{J}}^\dagger}{\mathrm{d}q}\widehat{\mathcal{F}} - \frac{\mathrm{d}\mathcal{J}^\dagger}{\mathrm{d}q}\mathcal{F}\right)h + \left(\widehat{\mathcal{J}}^\dagger\widehat{\mathcal{J}} - \mathcal{J}^\dagger\mathcal{J}\right)h\right\| + \mathcal{O}(\|h\|^2)\\
=\ & \left\|\left(\widehat{\mathcal{J}}^\dagger\widehat{\mathcal{F}} - \mathcal{J}^\dagger\mathcal{F}\right) + \left(\frac{\mathrm{d}\widehat{\mathcal{J}}^\dagger}{\mathrm{d}q}\left(\widehat{\mathcal{F}} - \mathcal{F} + \mathcal{F}\right) - \frac{\mathrm{d}\mathcal{J}^\dagger}{\mathrm{d}q}\mathcal{F}\right)h + \left(\widehat{\mathcal{J}}^\dagger\widehat{\mathcal{J}} - \mathcal{J}^\dagger\mathcal{J}\right)h\right\| + \mathcal{O}(\|h\|^2)\\
\leq\ & \underbrace{\left\|\left(\widehat{\mathcal{J}}^\dagger\widehat{\mathcal{F}} - \mathcal{J}^\dagger\mathcal{F}\right)\right\|}_{=\varepsilon_1} + \underbrace{\left\|\left(\frac{\mathrm{d}\widehat{\mathcal{J}}^\dagger}{\mathrm{d}q}\mathcal{F} - \frac{\mathrm{d}\mathcal{J}^\dagger}{\mathrm{d}q}\mathcal{F}\right)\right\|}_{=C(\bar q)}\|h\|\\
& + \underbrace{\left\|\frac{\mathrm{d}\widehat{\mathcal{J}}^\dagger}{\mathrm{d}q}(\widehat{\mathcal{F}} - \mathcal{F}) + \widehat{\mathcal{J}}^\dagger\widehat{\mathcal{J}} - \mathcal{J}^\dagger\mathcal{J}\right\|}_{=\varepsilon_2}\|h\| + \mathcal{O}(\|h\|^2)\\
=\ & \varepsilon_1 + (\varepsilon_2 + C(\bar q))\,\|h\| + \mathcal{O}(\|h\|^2).
\end{aligned}$$

q.e.d.

The reconstruction properties of $\mathcal{F}, \mathcal{J}$ and $\mathcal{J}^\dagger$ follow directly from the state and sensitivity reconstruction errors, given in Theorems 3.6 and 4.7, and Remark 3.9. Thus, with $C_{\varepsilon_1}, C_{\varepsilon_2} <$

∞ and the reconstruction errors ε_1 and ε_2 are bounded according to

$$\varepsilon_1 \leq C_{\varepsilon_1} \left(\sum_{i=k+1}^{d} \lambda_i^w + \sum_{j=\ell+1}^{\tilde{d}} s_i^w \right), \quad \varepsilon_2 \leq C_{\varepsilon_2} \left(\sum_{i=k+1}^{d} \lambda_i^w + \sum_{j=\ell+1}^{\tilde{d}} s_i^w \right).$$

Assuming ε_1 and ε_2 to be negligible, the estimate (5.9) is dominated by the constant $C(\bar{q})$ which is an upper bound for the difference between the derivatives of the pseudoinverses, computed with the HiFi and the surrogate model. We can assume in practice that $C(\bar{q}) \ll \infty$ for two reasons. On the one hand, the second-order derivatives $\frac{\mathrm{d}}{\mathrm{d}q}\mathcal{J}^\dagger(\bar{q})$ and $\frac{\mathrm{d}}{\mathrm{d}q}\widehat{\mathcal{J}}^\dagger(\bar{q})$ are bounded due to the κ-conditions and we require $\kappa < 1$ for convergence of Gauss–Newton. On the other hand, the residual $\mathcal{F}$ can be assumed to be small if the data can be fitted well through the model.

We now formulate the central theorem which yields an estimate on how far a suboptimal solution deviates from the HiFi solution.

Theorem 5.4. *For a given $\bar{q}$ where the DEPOD/DEDEIM basis is built, assume that the conditions of the local contraction theorem hold on $\mathcal{Q}^{h\tau}$ for HiFi and POD Gauss–Newton with $q_{(0)} \in \widehat{\mathcal{Q}}_0^{h\tau}$ and $\widehat{q}_{(0)} = \bar{q}$. Let $q^\star$ and $\widehat{q}^\star$ be the corresponding limit points of Theorem 2.5 (2). Moreover, require $\mathcal{J}^\dagger, \widehat{\mathcal{J}}^\dagger \in C^1(\widehat{\mathcal{Q}}_0^{h\tau})$. Then we have*

$$\|\widehat{q}^\star - q^\star\| \leq \frac{1}{1-\delta(\widehat{q}^\star)} \left(\varepsilon_1 + (\varepsilon_2 + C(\bar{q}))\, \|\widehat{q}^\star - \bar{q}\|\right) + \mathcal{O}(\|\widehat{q}^\star - \bar{q}\|^2) \tag{5.10}$$

where $\delta(\widehat{q}^\star) := \kappa + \frac{\omega}{2}\Delta q(\widehat{q}^\star) < 1$ and ω and κ as in (2.2) *and* (2.3) *for the HiFi, and ε_1, ε_2 and $C(\bar{q})$ as in Lemma 5.3.*

Proof. The theorem follows directly from Lemma 5.3 and the a priori estimate (3) in the local contraction theorem, where we set $n = l = 0$ and choose $q_0 = \widehat{q}^\star$ for HiFi Gauss–Newton. q.e.d.

From a practical point of view, Theorem 5.4 states that if Algorithm 3 terminates, then $\|\widehat{q}^\star - \bar{q}\| < TOL$ and with this the distance $\|q^\star - \widehat{q}^\star\|$ is small. Thus, we have found a solution of the HiFi parameter estimation problem up to some given termination tolerance and reconstruction errors.

Remark 5.1. Note that for negligible reconstruction errors estimate (5.10) is of first-order due to the DEPOD/DEDEIM enrichment. The estimate does not hold for standard POD ROMs as ε_1 already contains sensitivity information. Similar to the discussion in the optimal control case, estimate (5.10) is actually overly pessimistic and may in practice be much smaller.

Reconsidering the question of overall convergence of Algorithm 3, it is clear that for every first Gauss–Newton step after a DEPOD/DEDEIM basis is built ($q_{(0)} = \widehat{q}_{(0)} = \bar{q}$), we have

$$\|\Delta q(\bar{q}) - \Delta\widehat{q}(\bar{q})\| = \varepsilon_1,$$

where ε_1 depends on reconstruction errors. After the first iteration the difference between the Gauss–Newton directions depends on how much the dynamics of the underlying PDE system change. The a posteriori error estimate is not explicitly computed to test convergence of the DEPOD-PE later in the numerical results as this would involve an estimation of κ and ω which is beyond the scope of this thesis.

5.4. Approximate-then-optimize vs. optimize-then-approximate

In §2.1 we presented two conceptual approaches to solve optimization problems, i.e., the discretize-then-optimize (DTO) and the optimize-then-discretize (OTD) approach. In this section we discuss how optimization using reduced-order models relates to these approaches. We distinguish between the following two conceptual strategies to solve optimization problems using POD/DEIM reduced-order models. The dichotomy can be understood as an analogon to the DTO and OTD dichotomy.

1. approximate-then-optimize (ATO): First the dynamic model problem is approximated via a surrogate model which is subsequently used in the optimization process.
2. optimize-then-approximate (OTA): First the necessary optimality conditions are derived for the optimization problem. These are subsequently approximated by suitable surrogate approximations.

In [8] a similar distinction is made using the expressions 'approximate-then-design' and 'design-then-approximate', however, as we deal with general optimization problems, the naming is adapted. In the following we analyze the difficulties that can appear with either of the strategies. While we suggest to consider the above dichotomy, actually, there are many mixed forms found in literature. However, concerning the improvements we suggest, a classification into one of the two categories can be made.

In the ATO strategy one starts to construct the POD reduced-order model following the common techniques as described in §3 and in a subsequent step one deals with the optimization problem independently (see [83, 110]). Often the POD subspace is sufficiently rich and one obtains satisfying results from numerical computations. Typically in an ATO approach one does not explicitly construct the model for the optimization purpose. The advantage is that the models can be built via standard techniques and common optimization algorithms can be applied, independently of the particular surrogate model. If the direct approach and concepts of automatic differentiation are used, we always find an accurate descent direction and, thus, the discretized optimization problem can be solved with standard NLP techniques. However, as discussed before, the solutions obtained with this approach are typically suboptimal due to the POD prediction problem.

The problem with the ATO approach is that in general we lack consistency with the necessary optimality conditions of the HiFi problem in a suboptimal solution and, hence, with the infinite-dimensional problem. This may lead to the following situation: Assume the HiFi necessary optimality conditions to be satisfied at $q^\star$ and the reduced basis to be constructed in $q^\star$ from state snapshots only. Then it is possible that the optimality conditions of the surrogate model do not hold in this point. The reason is that by projecting onto subspaces we actually change the problem. From a practical point of view, this may cause that, even though constructing the surrogate and starting the optimization in the 'true' solution $q^\star$, the algorithm may converge towards a significantly different solution $\widehat{q}^\star$.

In contrast, with the OTA strategy one explicitly considers the optimality system and then derives approximations of the occurring equations (see,e.g, [8, 66, 112]). In case of the optimality system (1.16) in the adjoint approach, this means that there is a surrogate approximation y^k with subspaces V^k, R^ℓ for the states and a surrogate $p^{\tilde{k}}$ with subspaces $V_p^{\tilde{k}}, R_p^{\tilde{\ell}}$ for the adjoints, each computed via the POD optimality (3.1) and corresponding snapshots. The advantage is that the POD approximations of states and adjoints are by construction consistent with the original problem, thus, also the necessary optimality conditions are satisfied. However, we face the dilemma that the gradient $\nabla j^{h\tau}(q)$ approximated by p^k might not be a descent direction for the objective $j^{h\tau}(q)$ approximated by y^k. While in an OTD approach the inconsistency can be mitigated by suitable error control, the lack of reliable prediction error estimates for POD makes this a challenging task.

The inclusion of derivative information as suggested in Chapter 4 is a remedy to overcome the problem of either of the conceptual approaches. In case of OTA, using a DEPOD/DEDEIM basis yields first-order local approximation quality of the surrogate model due to Theorem 5.1. Thus, with improved local POD prediction we can guarantee that at least at the reference configuration we can find a descent direction for the objective $j^{h\tau}(q)$ approximated with y^k. In case of ATO, extending the POD and DEIM bases with derivative information yields a surrogate model that allows for small reconstruction errors of the objective and the gradient. Hence, the surrogate optimization problem constructed in a solution has the same local optimum as the HiFi optimization problem and the two problems have consistent optimality conditions. Algorithms 2 and 3 arise from a direct optimization perspective, thus, we classify our approach as ATO. We have also shown that our discrete HiFi optimality conditions are consistent approximations of the infinite-dimensional problem (see Remark 2.5). If we use DEPOD/DEDEIM bases for the construction of the surrogate model, then our direct approach also yields consistent POD approximations of the infinite-dimensional necessary optimality conditions. Hence, the ATO strategy we follow can also be interpreted as an OTA approach where joint subspaces for states and adjoints are used.

In Theorem 4.3 we gave an error estimate which depends on the DEIM projections $\bar{\mathcal{E}}^i_{\mathrm{fz}}$, $i = 1, \ldots, k$, where in contrast to a standard application of DEIM, instead of a single vector $F(x(t), q)$ the k vectors $F_x(x(t), q)\Psi_{.i}$ must be approximated. However, one could find an alternative DEIM projection for the adjoint equation of (P-ROM) (4.4). Recalling the equation

$$-\widehat{z}_t^T(t) = \widehat{z}^T(t)\widehat{S} + \widehat{z}^T(t)\Psi^T F_x(\Psi\widehat{x}(t), q)\Psi$$

one could also consider applying a projection matrix $\mathbb{P}$ to the nonlinear part such that

$$-\widehat{z}_t^T(t) = \widehat{z}^T(t)\widehat{S} + \left(\Psi^T\mathbb{P}\tilde{F}(t)\right)^T \quad \text{with} \quad \tilde{F}(t) := F_x(\Psi\widehat{x}(t), q)^T\Psi\widehat{z}(t). \tag{5.11}$$

In contrast to (PD-ROM), here the DEIM projection approximates the adjoint nonlinearity $\tilde{F}(t) \in \mathbb{R}^N$, which can be interpreted as an adjoint directional derivative of $F_x(x(t), q)$ with direction $z(t)^T \approx (\Psi\widehat{z}(t))^T$. Obviously, in terms of DEIM projection efficiency, it is better to consider only the dynamics of one instead of k vectors. Hence, (5.11) would be the natural choice when proceeding in an OTA fashion, where the projection of the adjoint equation is carried out separately. The problem is that

$$\left(\Psi^T\mathbb{P}\tilde{F}(t)\right)^T \neq \widehat{z}^T\Psi^T\mathbb{P}F_{\widehat{x}}(\Psi\widehat{x}, q),$$

where on the right-hand side we have the exact derivative of the nonlinear part of (PD-ROM) which is used in the adjoint equation (4.5). Thus, while we would be able to establish a reconstruction estimate also for (5.11), the approximated gradient would, however, not be consistent with the objective. In contrast, it is guaranteed for (4.5).

5.4.1. Examples

In the following we investigate two dilemmas of conventional ATO and OTA approaches on concrete optimal control examples and how the shortcomings can be overcome with DEPOD/DEDEIM. Similar problems are detected also for parameter estimation problems. An example is given in [104]. We vary the dynamic model problem (3.32) slightly, first considering

$$\begin{gathered} y_t = -a^T\nabla y + D\Delta y + \Theta(y, u_1) + u_2 \quad \text{in } I \times \Omega, \\ \partial_\nu y + \beta_1 y = \beta_2 \quad \text{on } I \times \Gamma_1, \qquad \partial_\nu y = 0 \quad \text{on } I \times \Gamma_2, \end{gathered}$$

with $\Omega = (0,1)^2$, $I = [0, 0.5]$, and Γ_1 and Γ_2 as before in in §3.3.5. Here we use an initial condition $y(0) = 0.5$, the values

$$a = (0.2, -1)^T, \quad D = 0.1, \quad \beta_1 = \beta_2 = 10^3,$$

the nonlinearity

$$\Theta(y, u_1) = e^{\sin(y r_1)} + u_1 y \sin(2\pi r_1),$$

and a u_2 that is spatially distributed with the shape function $\phi_S(r)$ in (3.33) which is also shown in Figure 3.1. To define the optimal control problem we use again the quadratic objective as in (4.42). Bounds on the two controls are never active in our computations. The discretization is analogous to the reconstruction tests, however, we now use $N = 1089$ degrees of freedom for the finite element method and $\tau = 2.5 \times 10^{-3}$ for the implicit Euler scheme.

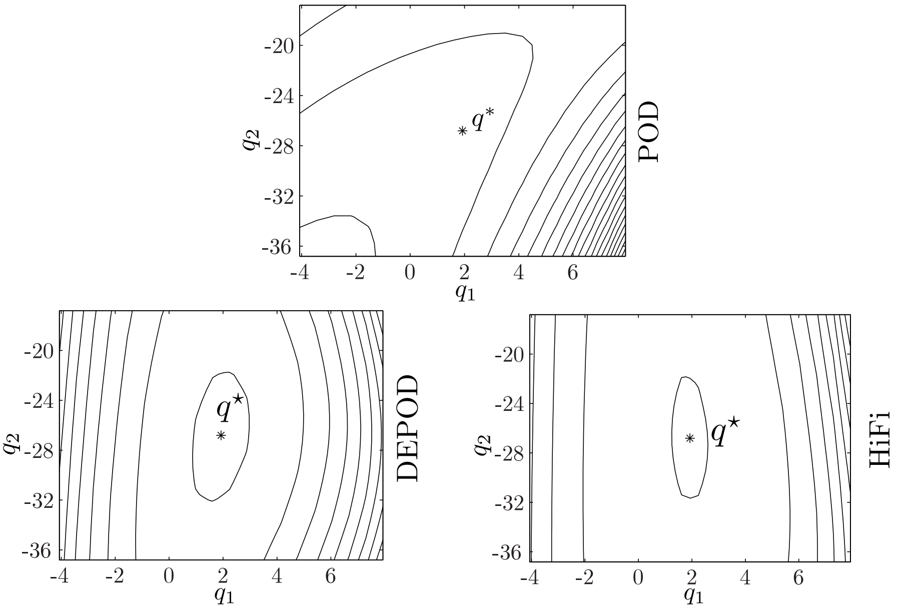

Figure 5.1.: Isolines of the objective $j^{h\tau}$ evaluated with a POD ROM and DEPOD ROM that are constructed in the solution point $q^\star$, and evaluated with the HiFi model.

ATO perspective

We start the investigations from the ATO perspective. The optimal solution was computed with the HiFi model discretization, solving the resulting NLP with SNOPT with a tolerance close to machine precision. The reference optimal control is $q^* \approx (1.92, -26.81)^T$. We now encounter the following situation: Assume that a surrogate optimization is carried out starting with the known optimal HiFi solution $q^\star$ as initial guess, which corresponds to an inner optimization loop in Algorithm 2. A surrogate (P-ROM) constructed in $q^\star$ without DEIM projection is used and a fixed number of $k = 15$ POD basis functions is chosen such that the reconstruction error at q^* for the conventional POD surrogate without derivative information is

$$RE\left(j^{h\tau}\right) = 7.1 \times 10^{-7}, \quad RE\,(x) = 2.4 \times 10^{-7},$$

using the norm $L^2(I, X)$ for the ODE states x. All snapshots available are included in the bases. We compare the behavior of the objective $\widehat{j}^{h\tau}(q)$ around the solution point with the POD basis constructed with and without adjoint information. To this end, in Figure 5.1 we visualize the isolines of each of the problem instances. We observe that the objective, if evaluated with a pure POD ROM, shows an entirely different behavior in comparison to the HiFi objective. Thus, even though the best possible control configuration guess was provided to the surrogate optimization, using it in an ATO fashion results in a significantly different solution. On the other hand, when a DEPOD basis is used, HiFi and surrogate model possess the same local optimum. Moreover, we see that the DEPOD ROM behaves similarly to the HiFi model around the construction point $q^\star$, which underlines the superior POD prediction properties of DEPOD.

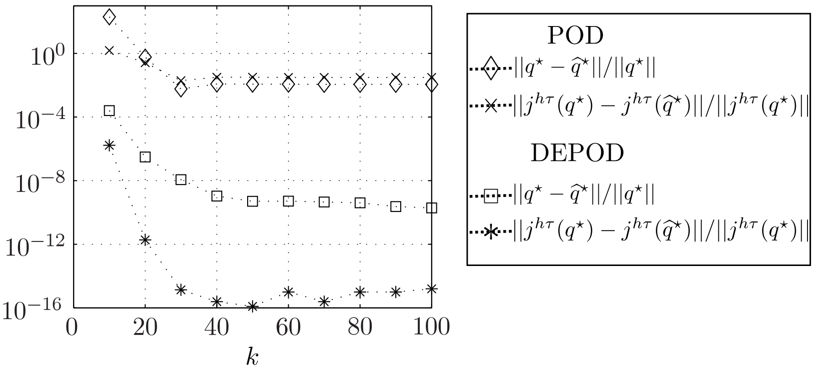

Figure 5.2.: Distance of suboptimal solutions $\widehat{q}^\star$ and $q^\star$ for surrogate optimization started in the numerically exact solution $q^\star$ and corresponding difference of the objective evaluated with the HiFi model.

A quantification of the difference between suboptimal solutions $\widehat{q}^\star$ from the surrogate optimization and the numerically exact HiFi solution $q^\star$ is shown in Figure 5.2. We performed the following test: The surrogate models with POD and DEPOD basis are constructed with an increasing number of basis functions in the HiFi solution $q^\star$. Then we solve the resulting NLPs up to machine precision, which we can do due to accurate derivative computation. We plot the difference

$$\frac{\|q^\star - \widehat{q}^\star\|}{\|q^\star\|}, \qquad \frac{\left\|j^{h\tau}(q^\star) - j^{h\tau}(\widehat{q}^\star)\right\|}{\|j^{h\tau}(q^\star)\|},$$

between HiFi and suboptimal solutions as well as the difference between objectives evaluated using the HiFi model at the HiFi and suboptimal solution respectively. We observe that with a standard POD ROM with $k = 10$ and a reasonable state reconstruction error $\left(RE\left(j^{h\tau}\right) \approx 5 \times 10^{-5}\right)$ one finds a suboptimal solution that differs from the reference solution by more than 100%. The resulting objective even differs by a factor of 200 from the objective value of the reference solution which shows that a conventional POD ROM may be highly unrobust. As described in §5.2 a common way to deal with the issue is to increase the number of POD basis functions. However, we can see that even by increasing the number of basis functions to $k = 100$, $\widehat{q}^\star$ does not get closer to $q^\star$ than approximately 3×10^{-2}. In contrast, with DEPOD we obtain a difference of about 10^{-4} already for $k = 10$. As in the reconstruction tests we observe an exponential decay that reaches an asymptotic limit, in this case determined by the accuracy of the time integration.

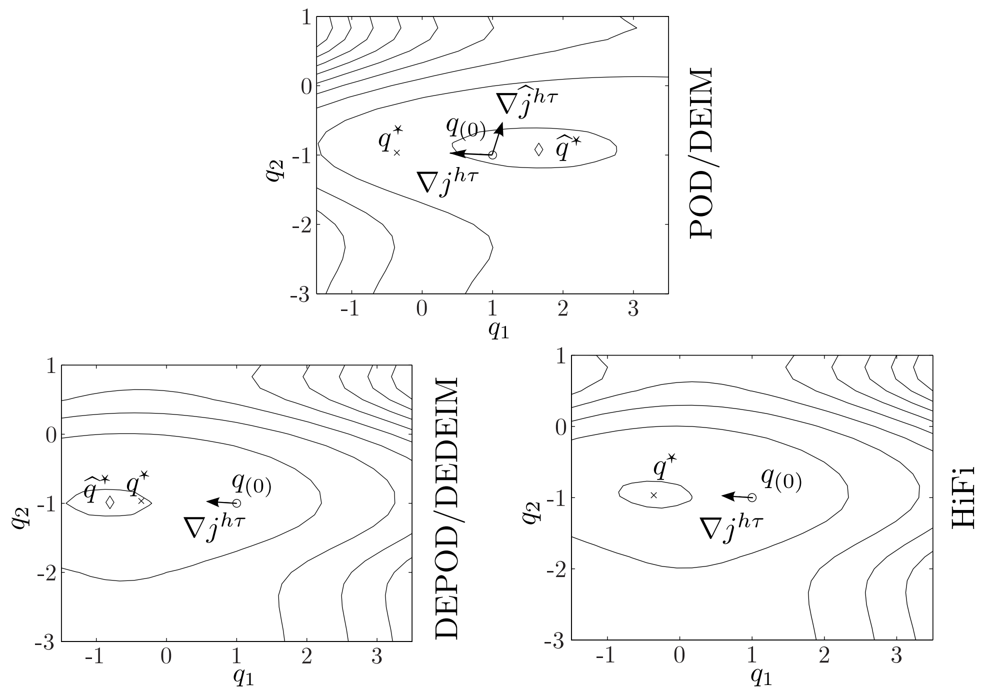

Figure 5.3.: Isolines of objective at initial guess $q_{(0)}$ for a POD/DEIM ROM (top), DEPOD/DEDEIM ROM (left), and the HiFi model (right) and the respective gradients and suboptimal solutions.

OTA perspective

In a second example we turn towards an OTA perspective on POD. The same example problem and discretization is used, however, we remove the spatially distributed control and replace the function Θ by

$$\Theta(y, u_1, u_2) = u_1 y \cos(\pi r_1) + y \sin(u_2 \pi r_2).$$

Consider the optimality system (1.16). The OTA strategy we employ is to construct (PD-ROM) for the state variables y, which we subsequently use to evaluate the objective. The computation of the gradient $\nabla j^{h\tau}$ is done without a model reduction step, i.e., the adjoint equation and the gradient equation in (1.16) are evaluated from HiFi information only. We then follow a reduced approach to solve the OCP, providing objective and gradient to the SQP method implemented in SNOPT.

The reference solution is $q^\star \approx (-0.36, -0.96)$ obtained with the HiFi model and an optimality tolerance close to machine precision. Clearly, when starting in $q^\star$, the optimality conditions are satisfied and the algorithm returns after the first convergence check. However, we now analyze the situation when starting the optimization at an initial guess $q_0 = (1, -1)$. We compare now different bases used in the reduced-order model to the case where the states are evaluated from HiFi information. We compute the surrogates with a fixed number of POD and DEIM basis functions, setting $k = 15$ and $\ell = 40$. Thus, the reconstruction errors using state information only for the POD/DEIM bases are

$$RE\left(j^{h\tau}\right) = 1.1 \times 10^{-6}, \quad RE\,(x) = 8.6 \times 10^{-7}.$$

In Figure 5.3 we show the isolines of the objective around the initial guess $q_{(0)}$ for a surrogate with POD/DEIM basis (top), with DEPOD/DEDEIM basis (left), and for the dynamic model evaluated with the HiFi model (right). As one can see, the suboptimal solution $\widehat{q}^\star$ in the scenario on the top is in opposite direction of the 'true' solution $q^\star$ relative to the initial guess. As the exact gradient information is provided $\nabla\widehat{j}^{h\tau}$ points towards $q^\star$, however, the actual steepest descent for the objective is $\nabla\widehat{j}^{h\tau}$. Thus, as the POD approximated gradient does not point in a direction where a decrease in the objective can be achieved, i.e.,

$$(\nabla j^{h\tau})^T \nabla\widehat{j}^{h\tau} < 0,$$

the solver cannot advance from this point and the OTA strategy we use gets stuck in this point. Obviously, this problem is inherent to the approach and not to the solver, as any gradient-based method that uses a line search will face the same difficulties. In contrast, with DEPOD/DEDEIM both gradient vectors point in the same direction up to the reconstruction error, which is $RE\left(\nabla j^{h\tau}\right) \approx 10^{-4}$. Moreover, we see that in the POD/DEIM case the objective behaves entirely different in comparison to the HiFi case while with DEPOD/DEDEIM the isolines are alike and we find a suboptimal solution already close to $q^\star$.

	(sub)optimal solution
HiFi reference solution $q^\star$	$(-0.36, -0.96)$
(PD-ROM) with $k = 15, \ell = 40$	$(1.46, -0.88)$
(PD-ROM) with $k = 15, \ell = 100$	$(2.74, -1.25)$
(PD-ROM) with $k = 15, \ell = 200$	$(1.25, -0.63)$

Table 5.1.: Comparison of HiFi solution of the optimization problem and suboptimal solutions obtained for different surrogate model instances dependent on ℓ with initial guess $q_{(0)} = (1, -1)^T$. The suboptimal solutions are still further from $q^\star$ than $q_{(0)}$ with $\ell = 200$

In Table 5.4.1 the optimal solution $q^\star$ obtained with the HiFi model is compared with suboptimal solutions when more information is added to the DEIM subspace. As we can see, with $\ell = 200$ basis functions the suboptimal $\widehat{q}^\star$ is still even further from $q^\star$ than the initial guess $q_{(0)}$. Without the DEIM projection, the pure POD ROM is able to find a solution $(-0.32, -0.90)$. Thus, in this example problem the large deviations in the suboptimal solution and, thus, the complete failure of the model reduction approach are caused by the DEIM projection step.

In either of the examples we are able to find a suboptimal solution via Algorithm 2 in less than five major iterations, using a stopping tolerance of $TOL_n = 10^{-4}$. We demonstrate the performance of the DEPOD-OC and DEPOD-PE algorithms in more detail in Part III.

Part III.

Applications

In the last part of this thesis we apply the developed algorithms to three applications, each with its own particular challenges for the model reduction approach. The aim of this part is to give an idea of how the derivative-extended proper orthogonal decomposition (DEPOD) algorithms behave for practical applications and how big the computational savings are.

The results are compared regarding to their approximation quality and the runtime. There is no state-of-the-art optimization algorithm for POD model reduction to test the developed algorithms against. Thus, we compare the DEPOD-OC and DEPOD-PE algorithm as presented in §5.2 and 5.3 against the same algorithms where standard POD and DEIM bases are used. This is very similar to the adaptive algorithms suggested in [1, 61]. In future implementations the ideas of DEPOD should, however, be combined with other POD optimization strategies such as optimality systems POD [77] or trust-region POD [6, 42, 73]. To guarantee accurate solutions to the underlying infinite-dimensional problem, methods for adaptive mesh refinement and error control on the HiFi discretization level must also be included.

We start by discussing aspects of the implementation and explain how the software components interact. A key aspect is the efficient implementation of the ODE right-hand side and their differentiation for both the HiFi and the reduced-order model. Thus, we can compare the runtimes such that they reflect the algorithmic superiority of POD without blurring the results due to an inefficient implementation.

We then present three different applications. In the first 2D heat transport problem we deal with a spatially distributed control function with high-dimensional control discretization and solve a nonlinear parameter estimation problem with four parameters. In the second application we consider a variation of the Lotka-Volterra predator-prey dynamics which we extend to a system of 2D partial differential equations. In the optimal control problem we deal with a time-dependent boundary control function and a nonlinear time-dependent control. In the last application we use a model from industry that contains a large system of 1D PDE equations. The challenges are the large system size, the complicated nonlinear contributions, and the handling of transient and 'quasi steady-state' phases during the time integration.

To facilitate the distinction between optimal control problems and parameter estimation problems, in the following we use the infinite-dimensional notation u for the variables in the optimal control case and the discrete notation q in the parameter estimation problems where $q \in \mathbb{R}^{n_q}$.

6. Implementation

We have implemented the optimal control Algorithm 2, the parameter estimation Algorithm 3, and the derivative-extended model reduction techniques described in Chapter 4. In the following we briefly describe the most important features of the implementation and discuss minor algorithmic aspects we have to deal with in practical computations.

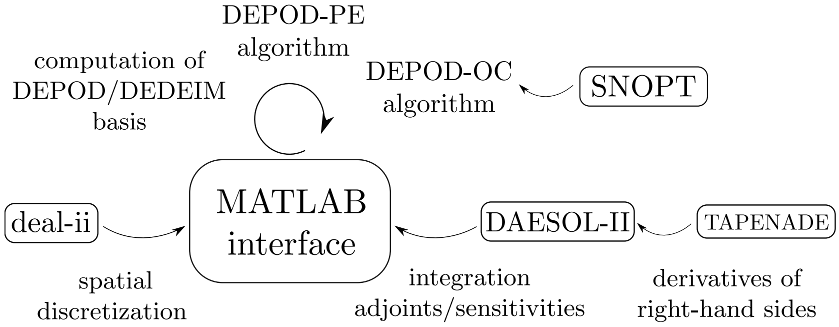

Figure 6.1.: Overview on the structure and additional software components in the implementation.

All results are obtained using a notebook computer with an Intel® Core™ i5-3317U CPU with 1.70GHz and 4GB of RAM under Ubuntu 13.10. The model reduction techniques and the optimization framework are implemented in Matlab® (8.1.0). The spatial discretization is obtained via the software package deal-ii [11], a finite element library to solve partial differential equations. The simulation and derivative computation of the HiFi and the surrogate model is carried out using a mex interface to the Backward Differentiation Formula (BDF) integrator DAESOL-II [2]. The derivatives of the right-hand sides of the dynamic systems (HiFi and surrogate) required by DAESOL-II are computed using the automatic differentiation (AD) source code transformation tool TAPENADE [55]. To solve the nonlinear programming problem (NLP) resulting from the optimal control problem discretization we use the sequential quadratic programming (SQP) algorithm implemented in the software package SNOPT [49]. An overview of the interacting software components is given in Figure 6.1. We now discuss various aspects of the implementation one by one.

Implementation and integration of semi-discrete PDEs in DAESOL-II

The integrator tool DAESOL-II requires the user to provide the ODE right-hand and left-hand side in C/C++. By default the required derivatives are then computed using ADOL-C [119], an AD tool based on operator overloading in C/C++. As an efficient evaluation of the right-hand sides and its derivatives is important to analyze the practical potential of our model-reduction approach, we replace the evaluation of the right-hand side by calls to Fortran routines. This allows the computation of adjoint and forward derivatives by source code transformation via TAPENADE. Details are discussed below.

The implementation of the HiFi right-hand side is structured into a linear and a nonlinear part according to (2.7). The components of the linear part S are evaluated using matrices and vectors computed with deal-ii before compiling the right-hand side code. The nonlinear part is implemented in a separate subroutine, such that it can be used for both the HiFi and the surrogate model. The POD/DEIM projection matrices and the reduced linear part are evaluated in Matlab® and then provided to the surrogate right-hand side during runtime.

In the numerical tests we vary the following integrator options of DAESOL-II: The relative tolerance TOL_{rel} is used to control the integration accuracy. We restrict the order $\max_{\text{order}}$ of the BDF-method if necessary to improve stability of the integration. Note that BDF schemes in practice are limited by the order 6 for stability reasons [54]. The value of TOL_{Newton} can be used to increase the accuracy with which the implicit systems in each step are solved. Essentially the value forces the Jacobian of the right-hand side to be evaluated more often (see also §2.5.3). When using an implicit Euler method in DAESOL-II we choose a maximum step size $\tau_{\max}$ and a large value for TOL_{rel} such that the actual step size τ of each step is $\tau_{\max}$. Moreover, we set $\max_{\text{order}} = 1$ and enforce the Jacobian matrix to be rebuild in every integration step.

Time dependent controls and checkpointing techniques

For problems with time-dependent control variables we subdivide the time horizon $[0, T]$ into subintervals $[t_{i-1}, t_i],\ i = 1, \dots, n_{\hat{q}}$ as explained in §2.2.2. We compute the gradient of the reduced objective in optimal control problems using the adjoint approach. To this end, first the system is integrated in time on each subinterval in a forward sweep. Then we do a reverse sweep which means going backward in time and computing the adjoint sensitivities and the gradient with respect to the control variables that are active in the current time interval.

For the application of the adjoint mode of AD we need to save all intermediate steps on the so-called tape, a term introduced in [51] to denote the memory space where information necessary for a reverse sweep is stored. This may result in excessive memory requirements, thus, we follow a checkpointing strategy (see also [51]). This means, we only remember the initial value conditions of the integration on each subinterval in the forward sweep. In the backward sweep we then recompute all intermediate steps on the active interval and store the required information on the tape. Thus, we avoid possible memory issues, however, the checkpointing strategy comes at the cost that we need to evaluate the forward problem twice (once in the forward and once in the backward sweep).

Source code transformation with TAPENADE

The derivatives required by DAESOL-II are provided by TAPENADE which uses source code transformation. The respective right-hand side Fortran code is processed by TAPENADE before compile time and Fortran routines for each type of derivative are created. The advantage is that the code can then be optimized by the compiler which yields significant runtime improvements.

In Table 6.1 the runtimes of a HiFi and a corresponding surrogate model for different GNU Fortran compiler flag options are shown. As one would expect, the runtimes improve with increasing compiler optimization. However, the savings are much larger in case of the surrogate model. Moreover, we observe that the gap in runtime improvement between HiFi and surrogate model becomes larger when increasing the number of POD modes. With $k = 30$ and compile flag -g in the surrogate we have half the cost of the HiFi model. Using compiler optimization it is possible to reduce the runtime by a factor > 10. We conclude that in case of a moderately sized HiFi problem an efficient implementation of the right-hand side is crucial to exploit the potential of the model reduction approach.

	Runtimes (in seconds)		
Compiler flag	-g	-O1	-O3
HiFi model $N = 1089$	5.41	4.65	4.57
surrogate model $k = 23, \ell = 174$	1.20	0.38	0.33
surrogate model $k = 30, \ell = 237$	2.09	0.54	0.41

Table 6.1.: Comparison of simulation costs for the HiFi model vs. the surrogate model in dependence of GNU Fortran compiler flag options.

Computation of right-hand side Jacobians using seed matrices

For the computation of the Jacobian of the right-hand side of the HiFi model we exploit the sparsity structure of the PDE spatial discretization. Let f be the differential right-hand side of the HiFi model. Then instead of evaluating each column of $\mathrm{d}f/\mathrm{d}x$ independently we compute n_{seed} directional derivatives

$$\frac{\mathrm{d}f}{\mathrm{d}x}A_{\text{seed}}, \qquad \text{with} \quad A_{\text{seed}} \in \mathbb{R}^{N \times n_{\text{seed}}}$$

where we typically have $n_{\text{seed}} \ll N$. The seed matrix A_{seed} can be determined using graph coloring algorithms. In the graph each node corresponds to a column of the Jacobian matrix and nodes are connected if the corresponding columns have nonzero entries at the same column entries. The full Jacobian can be reconstructed from the n_{seed} directional derivatives. For details see [51]. An efficient derivative compuation is important to obtain meaningful runtime results when comparing HiFi optimization with surrogate optimization.

Reduction of snapshot sets

The runtimes for the decomposition of the snapshot matrices strongly depend on the number of snapshots included. When computing the POD basis according to Variant II in §3.1.1, the size of the eigenvalue problem is determined by the number of snapshots, thus, the decomposition time starts to dominate the overall costs after a certain threshold. Thus, to construct the derivative-extended reduced-order models we need to deal with the possibly large number of snapshot sets. We experience the choice of the snapshot locations to have no significant effects on our numerical results. Therefore, we apply a heuristic to reduce the snapshot sets. If the snapshot choice becomes an issue one should consider methods for optimal snapshot locations as proposed in [78] or exploit a priori knowledge about the process.

Let $t_1, \dots, t_m$ be given snapshot time instances and m_{set} the number of overall snapshot sets to be included in the snapshot matrix. Then, if $m \cdot m_{\text{set}} > 1000$, we choose a number of reduced snapshot locations $\tilde{m}$ such that

$$m_{\text{set}} \cdot \tilde{m} \approx 1000.$$

With this choice we guarantee that the cost of the computation of the POD basis is negligible in comparison to the overall algorithm (on our machine less than 2 seconds). The number of $\tilde{m}$ is determined by choosing an integer valued reduction factor $m_{\text{red}} \approx (m \cdot m_{\text{set}})/900$ which we use to select every m_{red}-th time instance of $t_1, \dots, t_m$. Thus, we obtain a subset of $\tilde{m}$ time instances

$$\{t_{\pi_1}, \dots, t_{\pi_{\tilde{m}}}\} \subseteq \{t_1, \dots, t_m\}.$$

The POD snapshot weights $\tilde{\gamma}_{\pi_1}, \dots, \tilde{\gamma}_{\pi_{\tilde{m}}}$ for the reduced number of snapshot locations are given as

$$\tilde{\gamma}_{\pi_1} = (t_{\pi_2} - t_{\pi_1})/2, \quad \tilde{\gamma}_{\pi_i} = (t_{\pi_{i+1}} - t_{\pi_{i-1}})/2, \quad 2 \leq i \leq \tilde{m} - 1, \quad \tilde{\gamma}_{\pi_{\tilde{m}}} = (t_{\pi_{\tilde{m}}} - t_{\pi_{\tilde{m}-1}})/2,$$

analogous to the common POD snapshot weights in (3.2). The number of snapshot sets is not known a priori, as due to the dependency structure of the models some sets might contain only zero values. These sets we discard before the actual snapshot location reduction.

Settings for the SQP method

To solve optimal control problems we use an SQP algorithm with BFGS updates implemented in the software tool SNOPT Version 7 [49]. We access SNOPT via its Matlab® interface. Only bound constraints are considered. The tolerances for 'Major feasibility tolerance', 'Major optimality tolerance', and 'Minor feasibility tolerance' are set to one single value TOL_{snopt}. All other specifications for SNOPT are set to the standard value. As discussed in §5.2 we speak of convergence of the DEPOD-OC algorithm if the inner optimization loop returns successfully after the first iteration, which here means that when SNOPT returns with a successful info flag after the first convergence check.

Settings for the Gauss–Newton method

We implemented the Gauss–Newton algorithm as described in §2.4.2 in Matlab®. The examples required additional globalization strategies to be employed. We enforce the parameters to stay within certain lower and upper bounds to avoid an evaluation in infeasible regions. This is done by halving the length of the suggested increment Δq until no more bounds are violated. To guarantee global convergence a backtracking line search is used. We halve the increment Δq until we find a reduction in the norm of the residual $\|\mathcal{F}(q)\|$. The line search globalization is deactivated as soon as the increment is sufficiently small ($\|\Delta q\| > 0.5$).

7. 2D Heat Transport

In the first application of algorithms DEPOD-OC and DEPOD-PE we consider a convection-diffusion-reaction problem which can be interpreted as 2D model for heat transport. For example, the model could describe the temperature of a gas streaming through a tubular reactor with additional heat generation due to chemical reactions influencing the temperature. The reaction part consists of the nonlinear contribution part Θ which is affected by the DEIM reduction. We consider two different optimization problems. In the optimal control problem a linear spatially distributed control function must be determined such that a specific temperature profile is achieved at final time. In the parameter estimation problem we estimate the values of four parameters in the nonlinear contributions from artificially generated measurement data.

7.1. Model description

Assuming radial symmetric behavior on the unit square $\Omega = (0,1)^2$ and the time horizon $I = [0,T]$ with $T = 1$ the model equation is given as

$$\begin{gathered} y_t = -a^T \nabla y + D\Delta y + \Theta(y) + u \quad \text{in } I \times \Omega, \\ \partial_\nu y + \beta_{1,1} y = \beta_{2,1} \quad \text{on } I \times \Gamma_1, \\ \partial_\nu y + \beta_{1,2} y = \beta_{2,2} \quad \text{on } I \times \Gamma_2, \\ \partial_\nu y = 0 \quad \text{on } I \times \Gamma_3 \cup \Gamma_4, \qquad y(0) = 320 \quad \text{on } \Omega, \end{gathered} \tag{7.1}$$

and the boundaries are defined as

$$\begin{aligned} \Gamma_1 &:= \{r \in \Omega \ : \ r_2 = 1\}, \quad \Gamma_2 := \{r \in \Omega \ : \ r_1 = 1\}, \\ \Gamma_3 &:= \{r \in \Omega \ : \ r_2 = 0\}, \quad \Gamma_4 := \{r \in \Omega \ : \ r_1 = 0\}. \end{aligned}$$

The boundary part Γ_1 can be interpreted as reactor inflow, Γ_3 the outflow, γ_2 the reactor wall, and Γ_4 the center of the reactor along the axis. The nonlinear part Θ which is affected by the DEIM projection is given as

$$\begin{gathered} \Theta(y) = 1.5/4(-K_1 y + K_2 y + K_3), \quad \text{with} \\ K_1 = 5p_1 \exp(-1/y), \quad K_2 = (p_3 y/200 + p_2 \cos(y/300))^2, \quad K_3 = 0.6(p_4 - 1)\sin(2\pi r_1) y. \end{gathered}$$

The control function $u \in L^2(\Omega)$ is linear and distributed on the whole domain Ω. We choose the values defining the spatial operators and the boundary conditions as

$$\begin{gathered} a = (0,-1)^T, \quad D = 0.1, \quad \beta_{1,1} = 10^3, \quad \beta_{2,1} = 10^3 y_{\Gamma_1}, \\ \beta_{1,2} = 10, \quad \beta_{2,2} = 10 y_{\Gamma_2}, \quad y_{\Gamma_1} = 350, \quad y_{\Gamma_2} = 310. \end{gathered}$$

7.2. Optimal control results

To test the DEPOD-OC algorithm we optimize the final state distribution $y(T)$ by finding an optimal control function u. To this end, we consider the optimal control problem with quadratic objective

$$\min_{y,u} \ J(y,u) = \frac{1}{2} \left\| y(T) - y_\Omega \right\|_\Omega^2 + \frac{0.02}{2} \cdot \|u\|_\Omega^2 \qquad \text{s.t.} \quad (7.1). \tag{7.2}$$

We distinguish two particular problem instances differing in the shape function y_Ω for which we consider

$$y_\Omega^{(1)}(r) := \begin{cases} 360 & \text{if } r_1 < 0.5 \\ 300 & \text{else.} \end{cases}, \qquad y_\Omega^{(2)}(r) := 300 + 60\cos\left(\frac{\pi}{2} r_1\right).$$

Bounds on the controls are never active during the computations. The values of the four parameters $p_1, \ldots, p_4$, to be estimated later in the parameter estimation problem are set to one in the optimal control scenario.

We use linear finite elements for the spatial discretization with $N = 1089$ degrees of freedom and a time step size $\tau = 5 \cdot 10^{-3}$ for the implicit Euler method in DAESOL-II. With these choices the approximation quality of the reconstruction error essentially depends only on the number of POD basis functions.

The control function is discretized on the same grid as the states, i.e.,

$$u = \sum_{i=1}^{N} q_i \varphi_i \quad \in V^h.$$

The reference solutions $u^\star$ of the optimal control problems we compute using the HiFi model in SNOPT setting the tolerance to $TOL_{\text{snopt}} = 10^{-6}$. The optimization is started in either case with $u(r) = 1$. The reference optimal control function in the solution of problem (7.2) for the shape function $y_\Omega^{(1)}$ is displayed on the upper-left of Figure 7.2.

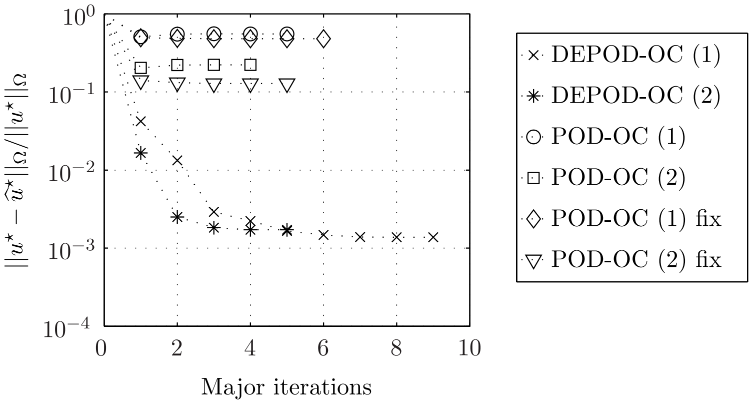

Figure 7.1.: Distance between reference solutions and surrogate suboptimal solutions in each major iteration of Algorithm 2. (1) and (2) refer to the objective where $y_\Omega^{(1)}$ and $y_\Omega^{(2)}$ is used respectively. 'POD-OC' means the DEPOD-OC algorithm with standard POD/DEIM bases. In 'POD-OC fix' we select the same $k = 26$ as for the last major iteration of DEPOD-OC.

In the first test we investigate the behavior of the major iterates of Algorithm 2 and compare them to the same algorithm where a standard POD/DEIM basis is used in the surrogate models instead of DEPOD/DEDEIM. The relative distances of suboptimal solutions $\widehat{u}^\star$ to the reference solutions $u^\star$, given as

$$\frac{\|\widehat{u}^\star - u^\star\|_\Omega}{\|u^\star\|_\Omega}, \tag{7.3}$$

in each major iteration are shown in Figure 7.1. Here 'DEPOD-OC (i)' refers to the DEPOD-OC algorithm applied to the problem instance where the shape function $y_\Omega^{(i)}$, $i = 1, 2$ is

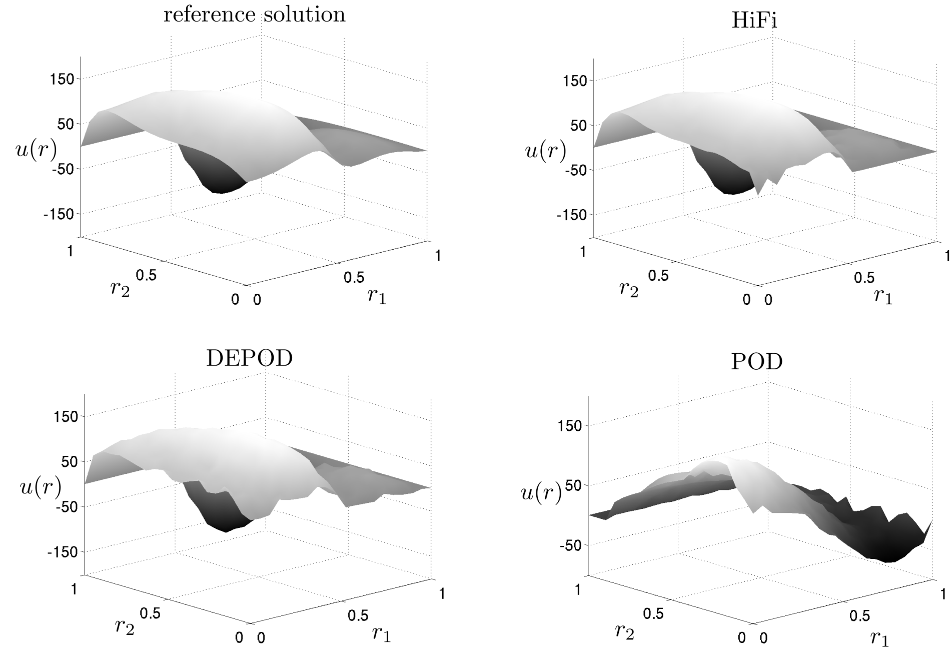

Figure 7.2.: Solutions of the optimal control problem (7.2) for the shape function $y_\Omega^{(1)}$. The reference control computed with the HiFi model and $TOL_{\text{snopt}} = 10^{-6}$ is shown on the upper-left. The HiFi, the DEPOD, and the POD solutions are computed with $TOL_{\text{snopt}} = 5 \times 10^{-5}$.

used. The same applies to 'POD-OC (i)' where the POD/DEIM basis is used instead in the surrogate model. For DEPOD-OC and POD-OC the bases are constructed using the tolerances $\lambda_{TOL} = 10^{-5}$ and $s_{TOL} = 10^{-4}$, i.e., we choose k and ℓ such that (compare to Remark 3.2)

$$\sqrt{\lambda_{k+1}} \leq \lambda_{TOL} \quad \text{and} \quad \sqrt{s_{\ell+1}} \leq s_{TOL}. \tag{7.4}$$

In 'POD-OC (i) fix' we choose the same number of basis functions $k = 26$ as in the last DEPOD-OC major iteration as a POD basis for a certain λ_{TOL} is by construction smaller than a DEPOD basis (compare Remark 4.2). In the last DEPOD surrogate model we have $\ell = 230$, thus, we use the full DEIM space for the fixed POD-OC algorithm as there are not enough snapshots available.

The tolerance for SNOPT is set to $TOL_{\text{snopt}} = 5 \times 10^{-5}$. We observe that either of the algorithms converge in less than 10 iterations. The POD-OC algorithms even converge after only 5 and 4 major iterations, however, the error between the reference solution and the suboptimal solutions at the final iteration are still $> 10^{-1}$. The suboptimal solution of POD-OC (1) is shown in the lower-right of Figure 7.2. With a POD basis only, we are not able to determine the essential features of the reference solution $u^\star$. In contrast with the DEPOD-OC algorithm the distance to $u^\star$ is almost as small as 10^{-3} for either problem instance. The suboptimal solution after the final iteration of DEPOD-OC (1) is shown in the lower-left part of Figure 7.2. We can see that the essential features of the reference control solution are reached. For comparison we show the solution obtained with the HiFi problem for $y_\Omega^{(1)}$ and the same SNOPT tolerance $TOL_{\text{snopt}} = 5 \times 10^{-5}$. The HiFi solution has a relative distance to $u^\star$ of 6.6×10^{-4} and is slightly smoother than the DEPOD-OC solution. However, we are nearly as good as the HiFi solution with DEPOD-OC for the given tolerance where we have a distance of 1.3×10^{-3} (compare also Table 7.1).

In the POD-OC algorithm with a fixed number of basis functions the POD and DEPOD subspaces have the same dimension (no DEIM projection is carried out in POD-OC fix). We observe that the additional snapshots (before in POD-OC we have $k \approx 12$ and $\ell \approx 13$) barely yield an improvement of the POD-OC results where the number of basis functions is chosen adaptively. We also have compared the algorithms to the optimize-then-discretize approach we used in §5.4.1. However, SNOPT returns with an error message after a few iterations as the line search cannot find a descent in the objective. Thus, no reasonable comparison is possible.

We also confirmed that the DEPOD-OC solutions at the final iteration are actually solutions to the HiFi problems within the given tolerance by starting the HiFi optimization in the solution of DEPOD-OC. This result can be expected as the DEPOD surrogates are constructed such that the reconstruction error of the gradient in the norm $\|\cdot\|_\Omega$ ($RE\left(\nabla j\right) \approx 4 \times 10^{-6}$) is below the accuracy requirements of the optimization ($TOL_{\text{snopt}} = 5 \times 10^{-5}$). It must be recalled that convergence of either of the algorithms is not necessarily equivalent to having found a solution of the HiFi problem. As discussed in §5.2 we must take care that the derivatives are approximated sufficiently well. The reconstruction errors with DEPOD are about $RE\left(x\right) \approx 10^{-5}$ and $RE\left(z\right) \approx 10^{-5}$.

	Objective with $y_\Omega^{(1)}$			Objective with $y_\Omega^{(2)}$		
	HiFi	DEPOD	DEPOD	HiFi	DEPOD	DEPOD
λ_{TOL}/s_{TOL}	-	$10^{-5}/10^{-4}$	$10^{-6}/10^{-5}$	-	$10^{-5}/10^{-4}$	$10^{-6}/10^{-5}$
k/ℓ in last iter.	-	26/230	33/287	-	25/196	31/254
Major iterations	-	9	7	-	5	4
Minor iterations	102	279	238	108	162	143
Minor iter. time	1358s	158s	182s	1431s	79s	92s
Total time	1358s	340s	342s	1431s	181s	183 s
Distance to $u^\star$	6.6×10^{-4}	1.3×10^{-3}	1.4×10^{-3}	3.0×10^{-4}	1.7×10^{-2}	1.2×10^{-3}
Error in J	2.4×10^{-4}	5.0×10^{-4}	4.9×10^{-4}	2.9×10^{-4}	1.1×10^{-2}	1.0×10^{-3}

Table 7.1.: Comparison of HiFi optimization ($N = 1089$) and the DEPOD-OC algorithm for two different objectives and different choices of eigenvalue criteria to choose k and ℓ. The tolerance for SNOPT is chosen as $TOL_{\text{SNOPT}} = 5 \times 10^{-5}$ and the initial guess is $u(r) = 1$.

In Table 7.1 we show the results of a comparison of DEPOD-OC and HiFi optimization for the two optimal control problem instances and different choices of eigenvalue criteria. The termination tolerance is again set to $TOL_{\text{SNOPT}} = 5 \times 10^{-5}$ and the results of each optimization are compared regarding their relative distance to $u^\star$ according to (7.3) and the relative error in the objective. For the latter we compare the objective value of the HiFi at the reference control with the objective value of the HiFi at the suboptimal solution of the DEPOD-OC algorithm. The total number of evaluations of the HiFi model necessary to construct the surrogate is the number of major iterations +1. The minor iterations correspond to the overall number of iterations needed in SNOPT. Note that this number does not necessarily correspond to the number of function evaluations which may be larger.

The results show that we reduced the overall runtime of the optimization by a factor 4 in case of the objective with $y_\Omega^{(1)}$ and almost a factor of 10 for $y_\Omega^{(2)}$. The loss of solution quality is relatively small, in particular when the results are compared to the case where standard POD and DEIM bases are used (see Figure 7.1). When comparing the behavior of the DEPOD-OC algorithm with regard to different choices of eigenvalue criteria λ_{TOL} and s_{TOL}, in either problem the number of major iterations can be reduced using a larger basis. However, the overall costs stay roughly the same which is due to the fact that the minor iterations become more expensive with larger DEPOD/DEDEIM bases. Moreover, using a larger basis did not significantly improve the outcome of the DEPOD-OC algorithm. However, the larger POD basis would allow the DEPOD-OC algorithm to satisfy a smaller tolerance criterion TOL_{SNOPT}. The issue will be further investigated in the application in chapter 8. As long as the necessary reconstruction accuracy for the surrogate model is achieved, it is in general hard to tell a priori how to choose the size of the reduced basis in order to obtain an optimal overall runtime performance.

The different shape functions $y_\Omega^{(1)}$ and $y_\Omega^{(2)}$ also had a significant impact on the behavior of Algorithm 2. While the number of HiFi optimization iterations were roughly the same, the surrogate optimization took more major and minor iterations in case of the discontinuous shape functions $y_\Omega^{(2)}$ in comparison to the smooth shape function $y_\Omega^{(2)}$.

7.3. Parameter estimation results

In the parameter estimation example we estimate the values of $p_1, \dots, p_4$ from measurement data with normally distributed random noise, thus, the optimization variables are $q_j = p_j,\ j = 1, \dots, 4$. The same model (7.1) as in the optimal control example is used, with the differences

$$\Theta(y) = 1.5(-K_1 y + K_2 y + K_3), \quad a = (0, -4)^T, \quad D = 0.4.$$

Space and time discretization are identical to the optimal control scenario. Let the four parts $\Omega_1, \dots, \Omega_4$ of the domain Ω be defined as

$$\Omega_1 := \{r \in \Omega \ : \ r_1 \le 0.5 \wedge r_2 \le 0.5\}, \quad \Omega_2 := \{r \in \Omega \ : \ r_1 \ge 0.5 \wedge r_2 \le 0.5\},$$
$$\Omega_3 := \{r \in \Omega \ : \ r_1 \le 0.5 \wedge r_2 \ge 0.5\}, \quad \Omega_4 := \{r \in \Omega \ : \ r_1 \ge 0.5 \wedge r_2 \ge 0.5\},$$

then we consider the 4 measurement functions

$$h_1(y(t)) = \int_{\Omega_1} y(t)dr, \qquad h_2(y(t)) = \int_{\Omega_2} y(t)dr,$$
$$h_3(y(t)) = \int_{\Omega_3} y(t)dr, \qquad h_4(y(t)) = \int_{\Omega_4} y(t)dr,$$

and a measurement error standard deviation of $\varsigma = 6$, which is a relative error of about 2%. In the numerical computations we approximate the integrals by the mean value of the node values in the corresponding region. Thus, the approximation is no more grid independent which has, however, no significant impact on the results as all computations are done on the same equidistant grid. In the experiment we take measurements at 10 different time instances $t_1 . \dots, t_{10}$ defined as

$$t_n = \sum_{i=1}^{n} \Delta t_i, \quad n = 1, \dots, 10, \qquad \Delta t_i = T/10,$$

and increase the value of y_{Γ_1} at the inflow boundary successively from $y_{\Gamma_1} = 300$ on $[t_0, t_1]$ to $y_{\Gamma_1} = 440$ on $[t_9, t_{10}]$ with linearly interpolated values.

We compute 2 sets of measurement data with the parameters set to 1 and add normally distributed errors with standard deviation ς to the value of the measurement function. The reference solutions $q^\star_{(1)}, q^\star_{(2)} \in \mathbb{R}^4$ for each set of measurement data are computed with the HiFi model and the Gauss–Newton algorithm and a termination tolerance of TOL_{GN} close to machine precision.

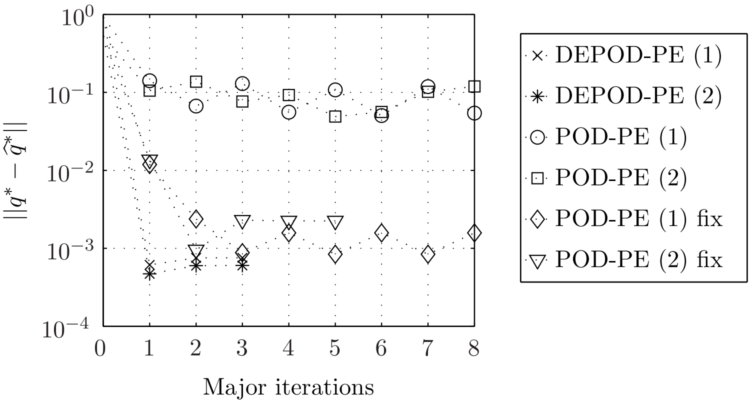

Figure 7.3.: Distance to reference solution for each major iteration of the surrogate optimization. The number in brackets refers to the measurement data set used in the parameter estimation. 'POD-PE' refers to the DEPOD-PE algorithm with a standard POD/DEIM basis used. In 'POD-PE fix' we choose the same number of basis functions as in DEPOD-OC.

As for the DEPOD-OC algorithm we investigate the behavior of the major iterates of DEPOD-PE and compare them to the same algorithm where a standard POD/DEIM basis is used which we denote by POD-PE. Again we choose a fixed number of basis functions in 'POD-PE fix' corresponding to the size of the bases in DEPOD-OC. The distances of the suboptimal solutions in each major iteration to the reference solutions are shown in Figure 7.3. We give here the absolute error as the parameters have an order of magnitude of one. The number in brackets refers to the different measurement data sets used to estimate $q_1, \ldots, q_4$. We set the Gauss–Newton tolerance to $TOL_{\mathrm{GN}} = 10^{-6}$ and the major iteration stopping tolerance of DEPOD-PE to $TOL_n = 10^{-5}$. The initial values of the parameters are set to 0.6 in either case. The tolerances for the construction of the surrogate model are $\lambda_{TOL} = 2 \times 10^{-4}$ and $s_{TOL} = 2 \times 10^{-3}$. We can see from the plots that the DEPOD-PE algorithm converges in only 3 major iterations for either measurement data set used. The approximation quality of the parameters is well within the confidence region, which is approximately given by the diagonal elements of the covariance matrix $C(q)$. In the solution $q^*_{(1)}$ of the first measurement data set these are given as

$$\operatorname{diag}(C(q^*_{(1)})) \approx (0.48, 0.47, 0.20, 0.07).$$

In contrast with the POD-PE algorithm there is no convergence at all. The algorithm only stops as we limit the maximum number of major iterations to 8. The suboptimal solutions obtained are not or just barely within the approximate confidence region. When the same number of basis functions is used the results of standard POD can be significantly improved in this case (fixed basis $k = 18$, $\ell = 36$ and before in POD-OC $k = 8$, $\ell = 10$). However, with the measurement data set (2) it takes five major iterations to converge and with (1) the algorithm does not converge at all. We also tested an Optimize-Then-Discretize approach analogous to §5.4.1, providing the algorithm with the HiFi Jacobian and a surrogate approximation of the residual $\mathcal{F}$ without derivative enhanced basis. The results are very similar to the POD-PE iterates and are, thus, not explicitly shown. The problem of incompatible gradient information here did not make the algorithm break down as we do full Gauss–Newton steps close to the solution, thus, no more descent must be guaranteed.

Further we see that the error in the suboptimal solutions of DEPOD-PE are the smallest after the first iteration. One must note that many discrete decisions are involved in the reduced-order model construction process and we cannot make any assertions on suboptimal solutions far from the initial guess. However, this numerical artifact vanishes when the POD approximation quality is increased (not explicitly shown here).

	HiFi GN	DEPOD-PE			
λ_{TOL}/s_{TOL}	-	$2 \cdot 10^{-4}/2 \cdot 10^{-3}$	$10^{-5}/10^{-4}$	$10^{-6}/10^{-5}$	$10^{-7}/10^{-6}$
k/ℓ in last iter	-	18/36	27/57	39/79	52/96
Major Iterations	-	3	3	2	2
Minor Iterations	10	16	14	12	12
Minor Iter. Time	251s	19s	18s	18s	21s
Total Time	251s	64s	62s	52s	55s
Distance to $q^\star_{(1)}$	2.3×10^{-8}	7.5×10^{-4}	6.8×10^{-5}	6.7×10^{-6}	7.3×10^{-7}
Residual error	2.3×10^{-13}	1.1×10^{-6}	7.6×10^{-9}	1.4×10^{-11}	7.1×10^{-12}

Table 7.2.: Comparison of HiFi Gauss–Newton ($N = 1089$) and the DEPOD-PE algorithm for different choices of eigenvalue criteria to choose k and ℓ. The optimization tolerances are $TOL_{\mathrm{GN}} = 10^{-6}$ and $TOL_n = 10^{-5}$. Initial values for the parameters are set to 0.6.

In Table 7.2 we show the runtime results of HiFi Gauss–Newton and the DEPOD-PE algorithm for different values of λ_{TOL} and s_{TOL} to choose k and ℓ. The computations are carried out with the first measurement set with reference solution $q^\star_{(1)}$. The overall runtime can be improved by a factor of about four with either version of the surrogate model. Three major iterations are necessary in the first two surrogate optimizations and two when the eigenvalue criteria are decreased. The distance to the reference solutions is measured in the 2-norm and the residual error is the difference between the reference residual $0.5 \left\|\mathcal{F}(q^\star)\right\|^2$ and the HiFi residual evaluated with the corresponding solution. In contrast to the optimal control problem the distance between the HiFi solution and the reference solution $q^\star$ is significantly smaller than with the DEPOD-PE algorithm. The reason is that we defined a well-posed parameter estimation problem, i.e., the parameters can be identified within relatively small confidence regions. This leads to fast convergence of the Gauss–Newton method close to the solution. It only takes 10 Gauss–Newton iterations for the HiFi optimization to converge and the reference solution (TOL_{GN} close to machine precision) is obtained with only three more iterations (compare to the discussion in §2.4.3). We can, however, see that we obtain a suboptimal solution error close to the HiFi solution error by decreasing λ_{TOL} and s_{TOL}. The results show that we can expect a small reconstruction error to yield a small optimization solution error when a DEPOD/DEDEIM basis is used.

8. 2D Predator-Prey

In this chapter we apply the DEPOD optimization algorithms to the well-known Lotka–Volterra equations which describe the population dynamics of predators and prey. We extend the standard ODE model to a system of 2D partial differential equations. The population count in this problem formulation can be influenced on certain parts of the boundary of the domain and by an additional nonlinear grow rate term that we introduce.

The challenges in this example are the application of DEPOD/DEDEIM to systems of partial differential equations and the optimal control of time-dependent controls one of which enters nonlinearly. Moreover, we control the error via the strategies in DAESOL-II, thus, the results of this chapter are time-grid independent. We start with a description of the underlying model and then discuss the strategies that we employ to deal with PDE systems. Finally, we present numerical results for a sample optimal control and parameter estimation problem.

8.1. Model description

On the domain $\Omega = (0,1) \times (0,2)$ we consider the differential states y_1 (prey) and y_2 (predators) defined by the coupled system of partial differential equations

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} y_1 &= -a^T \nabla y_1 + D\Delta y_1 + \Theta_1(y_1, y_2), \\ \frac{\mathrm{d}}{\mathrm{d}t} y_2 &= D\Delta y_2 + \Theta_2(y_1, y_2), \\ \partial_\nu D y_1 + \beta_{1,1} y_1 = \beta_{2,1} \quad &\text{on } I \times \Gamma_1, \qquad \partial_\nu D y_1 = 0 \quad \text{on } I \times \Gamma_3, \\ \partial_\nu D y_2 + \beta_{1,2} y_2 = \beta_{2,2} \quad &\text{on } I \times \Gamma_2, \qquad \partial_\nu D y_2 = 0 \quad \text{on } I \times \Gamma_3, \\ y_1(0) = 0.2 \quad \text{on } \Omega, &\qquad y_2(0) = 0.1 \quad \text{on } \Omega. \end{aligned} \tag{8.1}$$

The time horizon is $I = [0, T]$ with $T = 6$ in the optimal control and $T = 12$ in the parameter estimation example. The nonlinearities contain the Lotka-Volterra dynamic with the growth and death rate of predators and prey, given as

$$\begin{aligned} \Theta_1(y_1, y_2) &= y_1(p_1 - p_2 y_2), \\ \Theta_2(y_1, y_2) &= -y_2(p_3 - p_4 y_1 - K), \\ K &= \frac{1}{(k_0 y_2)^2 + 0.1}. \end{aligned}$$

We add an additional nonlinear grow rate term K for the predators which depends on k_0. The boundary $\Gamma = \partial\Omega$ is subdivided according to Figure 8.1. We consider a boundary Γ_1 where the prey population is affected by a term y_Γ^{prey}, a boundary Γ_2 where the predator population is affected by a term y_Γ^{pred}, and $\Gamma_3 = \Gamma \backslash (\Gamma_1 \cup \Gamma_2)$. The spatial operators and boundary conditions in the optimal control case are defined via

$$\begin{gathered} a = (0,0)^T, \quad D = 0.2, \quad \beta_{1,1} = 1, \quad \beta_{2,1} = y_\Gamma^{\text{prey}}, \\ \beta_{1,2} = 1, \quad \beta_{2,2} = y_\Gamma^{\text{pred}}. \end{gathered}$$

The values $p_1, \dots, p_4$ in the nonlinear contributions are set to

$$p_1 = 3.0, \quad p_2 = 2.0, \quad p_3 = 2.0, \quad p_4 = 2.8.$$

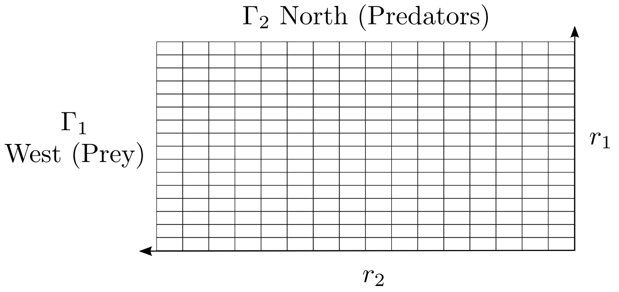

Figure 8.1.: Domain Ω and grid of the predator-prey 2D PDE problem.

8.2. DEPOD/DEDEIM for PDE systems

In the predator-prey application we apply our proposed model reduction techniques to a system of PDEs. Conceptually, POD can be applied to any ODE system via the projection step described in §3.1.3. However, in practice we rely on the good prediction properties of POD surrogate models which are particularly favorable when applied to parabolic problems. Recall that the prediction estimates in §5.1 only make an assertion on the error close to the control configuration where the basis is constructed, thus, with DEPOD we still need to rely on good prediction properties further away from the reconstruction point for an efficient application.

Therefore, we decompose the snapshot matrix for each of the components of a system independently. Consider the semi-discrete system of PDEs of size $n_x = N \cdot n_c$ with n_c components

$$\begin{gathered}\tilde{M}\dot{x}(t) = Sx(t) + F(x(t)),\\ x(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_{n_c}(t) \end{pmatrix}, \quad F(x(t)) = \begin{pmatrix} F_1(x(t)) \\ \vdots \\ F_{n_c}(x(t)) \end{pmatrix}, \\ x_i(t) \in \mathbb{R}^N, \quad F_i : \mathbb{R}^{n_x} \to \mathbb{R}^N, \quad i = 1, \dots, n_c,\end{gathered} \tag{8.2}$$

where $\tilde{M}, S \in \mathbb{R}^{n_x \times n_x}$ are block diagonal matrices with blocks of size $N \times N$. In a standard application of POD model reduction to (8.2) one would consider the snapshot matrix $\mathcal{S} \in \mathbb{R}^{n_x \times m}$ consisting of snapshots $x^1, \dots, x^m$ and use the mass matrix $\tilde{M}$ to obtain the projection matrix Ψ. Instead we apply POD on sub-matrices $\mathcal{S}_i \in \mathbb{R}^{N \times m}$, $i = 1, \dots, n_c$ where each $\mathcal{S}_i$ consists of snapshots of the components $x_i^1, \dots, x_i^m$. This results in a set of projection matrices $\Psi_1, \dots, \Psi_{n_c}$ from which we can build the overall projection matrix

$$\Psi = \begin{pmatrix} \Psi_1 & 0 & \\ 0 & \ddots & 0 \\ & 0 & \Psi_{n_c} \end{pmatrix}.$$

With a POD approximation $x = \Psi\widehat{x}$ we can then construct the surrogate model as described in §3.1.3 with the only difference that Ψ and the resulting linear part $\widehat{S}$ are now block diagonal. This can immediately be carried over to derivative-extended snapshot matrices.

For the application of DEIM we need to be more cautious and take a closer look at the structure of F. In case of a PDE system, $F(x)$ results from the discretization of the nonlinear part $\Theta(y_1, \dots, y_{n_c})$. Thus, each component $F_i(x)$ of $F(x)$ in general may depend on all of the components $x_i,\ i = 1, \dots, n_c$. As DEIM selects a number of indices among the quadrature evaluation points, these must be the same for all components of the system. Hence, we construct one single DEIM projection matrix Φ for all system components from the snapshots

$$\left\{F_1(x^j)\right\}_{j=1}^m, \dots, \left\{F_{n_c}(x^j)\right\}_{j=1}^m$$

included in $\mathcal{D}$. We can then determine the DEIM interpolation points via Algorithm 1 for all components simultaneously. It is also possible to construct the matrices $\Phi_1, \dots, \Phi_{n_c}$ independently, however, then it is unclear how to guarantee common interpolation points for all components.

To construct a DEDEIM basis we need to have a closer look at the term $F_{\widehat{x}}(\Psi\widehat{x}) = F_x(\Psi\widehat{x})\Psi$ which appears in the adjoint of the POD/DEIM reduced-order model in equation (4.5). According to our strategy to compute a single DEIM projection matrix Φ this requires the inclusion of the snapshots

$$\left\{\frac{\mathrm{d}F_1(\Psi\widehat{x}^j)}{\mathrm{d}x}\Psi\right\}_{j=1}^m, \dots, \left\{\frac{\mathrm{d}F_{n_c}(\Psi\widehat{x}^j)}{\mathrm{d}x}\Psi\right\}_{j=1}^m$$

in the snapshot matrix $\mathcal{D}$ which would mean $n_c \cdot m \cdot k$ additional vectors. However, due to the block diagonal structure of Ψ we find that

$$\frac{\mathrm{d}F_i(\Psi\widehat{x})}{\mathrm{d}x}\Psi = \begin{pmatrix} \frac{\partial F_i(\Psi\widehat{x})}{\partial x_1}\Psi_1 & \dots & \frac{\partial F_i(\Psi\widehat{x})}{\partial x_{n_c}}\Psi_{n_c} \end{pmatrix}, \quad i = 1, \dots, n_c.$$

In practical applications the components F_i are likely to depend only on a few of the state components. Thus, many of the partial derivatives are zero and the corresponding vectors should be removed from the matrix $\mathcal{D}$ before its decomposition. The case for control dependent F in the adjoint case and the sensitivity-extended DEIM basis is obtained analogously.

8.3. Optimal control results

The goal of the optimal control problem is to influence the dynamic system such that the population amount of prey on the whole spatial domain is close to one over the whole time horizon. The optimization variables are the boundary control $y_\Gamma^{\text{prey}} =: u_1$ and the factor $k_0 =: u_2$ in the nonlinear grow factor K. Both are time dependent, thus, we consider the control variable $u \in L^2(I, \mathbb{R}^2)$ and the problem

$$\begin{aligned} \min_{y,u} J(y,u) &= \frac{1}{2}\int_I \|y_1(t) - 1\|_\Omega^2\, dt + \frac{0.0005}{2}\|u\|_{\mathcal{Q}}^2 \\ \text{s.t.} \quad (8.1), &\qquad 0 \le u_1(t) \le 20, \quad 0 \le u_2(t) \le 5, \quad t \in I. \end{aligned}$$

We use the norm

$$\|u\|_{\mathcal{Q}} = \left(\int_I (u_1)^2 dt\right)^{\frac{1}{2}} + \left(\int_I (u_2)^2 dt\right)^{\frac{1}{2}}.$$

Over time the value of y_Γ^{pred} in the boundary conditions for the predator is varied according to

$$y_\Gamma^{\text{pred}}(t) = \begin{cases} 0.1 & \text{for } 0 \le t < 2, \\ 2.1 & \text{for } 2 \le t < 4, \\ 0.05 & \text{for } 4 \le t \le 6. \end{cases} \tag{8.3}$$

We discretize in space with linear finite elements and $N = 289$, thus, the dimension of the system is $n_x = 579$. For the time integration we use the settings $\max_{\text{order}} = 4$ and $TOL_{\text{Newton}} = 10^{-3}$ and we take advantage of the monitor strategy in DAESOL-II to decide when to rebuild the iteration matrices. For the relative error TOL_{rel} we tested different values in combination with different optimization tolerances. The discretization of the control variable is done with piecewise constant controls on 60 equally sized subintervals. The initial guesses are $u_1(t) = 1$ and $u_2(t) = 1$. The reference optimal control trajectories $u_1^\star$ and $u_2^\star$ are shown in Figure 8.2, which we obtained via the HiFi model, an SNOPT tolerance of $TOL_{\text{snopt}} = 10^{-6}$, and a relative time integration error of $TOL_{\text{rel}} = 10^{-7}$.

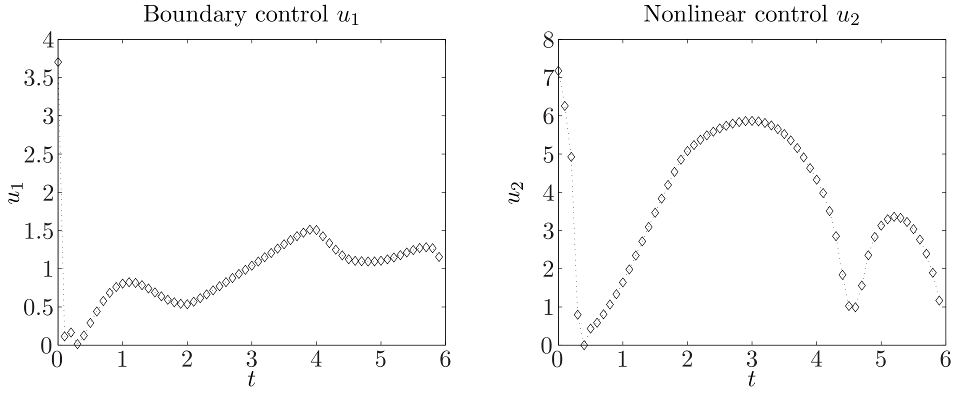

Figure 8.2.: Optimal solution trajectories of control functions.

We carry out a HiFi optimization via SNOPT, the DEPOD-OC algorithm, and the DEPOD-OC algorithm with standard POD/DEIM basis (POD-OC) to solve the optimal control problem with the two settings in Table 8.3.

	TOL_{rel}	TOL_{snopt}	λ_{TOL}	s_{TOL}
Settings (1)	4×10^{-5}	10^{-4}	5×10^{-3}	2×10^{-2}
Settings (2)	1×10^{-6}	10^{-6}	2×10^{-3}	1×10^{-2}

Table 8.1.: Settings for optimal control problem tests

The optimization results are shown in Figure 8.3. In the diagram with the dark gray bar we display the distance between the solutions of the respective algorithm and the reference solution $u^\star$ with respect to the norm $\|\cdot\|_Q$. The light gray bar reflects the distance $\|j(u^\star) - j(\widehat{u}^\star)\|$ between the objective value of the reference solution ($j(u^\star) \approx 1.14$) and the HiFi objective value evaluated with the respective solution. We observe that the solutions have roughly the same quality for Settings (1). The error is the smallest with the HiFi model and DEPOD shows a slight improvement in the control solution in comparison to POD. In contrast when the accuracy requirements are higher, POD-OC is unable to find a solution with an error close to the HiFi problem. With Settings (2) the solution quality is even worse than with Settings (1). With DEPOD-OC we are able to obtain a solution almost as good as the HiFi solution. Note that we only slightly decreased the eigenvalue tolerances from settings (1) to (2) which resulted in $k = 22, \ell = 81$ in comparison to $k = 19, \ell = 68$ (compare Table 8.2). The approximation quality of the DEPOD ROM here is mainly improved due to the reduced error in the time integration. Note also that in contrast to the previous application all solutions are computed on different time grids determined by the step size control of DAESOL-II. In this sense the results presented in this chapter are time-grid independent.

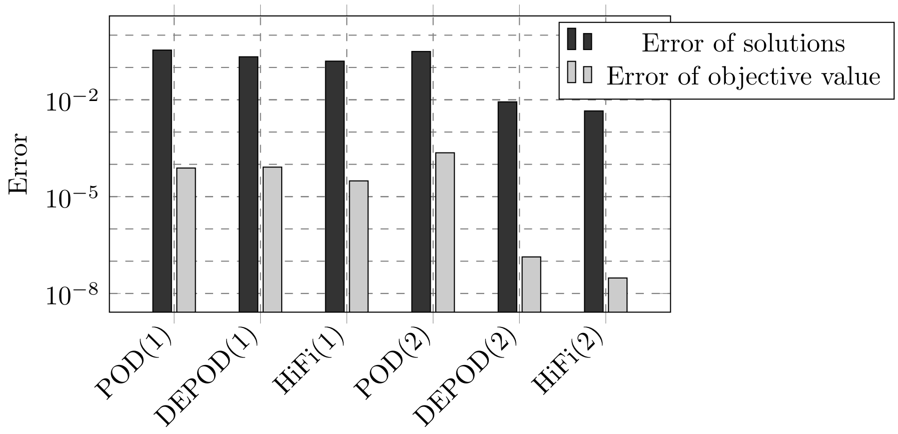

Figure 8.3.: Error of optimal control solutions in norm $\|\cdot\|_Q$ and objective error, i.e., the distance between the reference objective and the HiFi objective evaluated with the respective solution, for Settings (1)/(2) and POD, DEPOD, and HiFi optimization.

In Table 8.2 we compare runtimes of a HiFi optimization and the DEPOD-OC algorithm for the two tolerance settings in Table (8.3). In the first setting the overall runtime is reduced by a factor of approximately 8.5 and in the second by a factor of 7. The gain in runtime is smaller for the higher accuracy demand even though the number of minor iterations is doubled in the HiFi optimization while the number of minor iterations increases only by about 40% for DEPOD-OC. One reason for this is that the cost of the surrogate evaluation for the objective and the gradient grows faster when decreasing TOL_{rel} than for the HiFi model. Evaluation costs for Settings (1) are approximately 13.5 seconds (HiFi) and 0.5 seconds (surrogate) while for Settings (2) we have 22 seconds for the HiFi and 1.4 seconds for the surrogate. On the one hand, this can be explained by the larger size of the POD basis. On the other hand, in DAESOL-II we observe a faster increase of the number right-hand side Jacobian rebuilds for the surrogate model than for the HiFi. We will further discuss the issue of runtimes within DAESOL-II in the last application.

The number of DEPOD/DEDEIM basis functions are $k = 18/\ell = 68$ with Settings (1) and $k = 21/\ell = 81$ with Settings (2). Recall that here actually a POD basis for each of the components is computed, which takes 9 (10) basis functions for the first component and 9 (11) for the second one respectively.

8.4. Parameter estimation results

In the parameter estimation problem we estimate the Lotka–Volterra grow and death rate factors $p_1, \ldots, p_4$ which we now refer to by the optimization variables $q_j = p_j,\ j = 1, \ldots, 4$. We again generate randomly perturbed measurement data as in the 2D heat transport application to estimate the parameter values. For the parameter estimation problem the

	Settings (1)		Settings (2)	
	HiFi	DEPOD	HiFi	DEPOD
Major iterations	-	3	-	3
Minor iterations	52	99	98	138
Minor iteration time	1371s	77s	3962s	442s
Total time	1371s	160s	3962s	572s
k, ℓ in last iteration	-	18 / 68	-	21 / 81

Table 8.2.: Runtime results for HiFi optimization ($n_x = 578, N = 289$) and DEPOD-OC for the two settings in Table (8.3).

term K in the nonlinear part Θ is set to zero and we choose

$$a = (0, -0.2)^T, \quad D = 0.2, \quad \beta_{1,1} = 0.2, \quad \beta_{2,1} = 0.2 \cdot y_\Gamma^{\text{prey}},$$
$$\beta_{1,2} = 1, \quad \beta_{2,2} = y_\Gamma^{\text{pred}},$$

and the final time is now $T = 12$. We consider two measurement functions h_1, h_2 defined as

$$h_1(y(t)) = \int_{\Gamma_{h_1}} y(t)dr, \qquad \Gamma_{h_1} := \{r \in \Omega \ : \ r_1 \leq 0.5 \wedge r_2 = 2\},$$
$$h_2(y(t)) = \int_{\Gamma_{h_2}} y(t)dr, \qquad \Gamma_{h_2} := \{r \in \Omega \ : \ r_1 > 0.5 \wedge r_2 = 2\},$$

which means that we can count the amount of prey on two parts of the boundary Γ_1 of the domain. The measurement error standard deviation is set to $\varsigma = 0.1$. As above, we approximate the integrals by the mean value of the node values in the corresponding region in the numerical computations. During the experiment we take 12 measurements at the time instances $1, \dots, 12$ and the boundary value for the predators is again varied over time according to (8.3) in the optimal control example. The parameter initial values are set to $q = (2, 2, 2, 2)^T$. The space and time discretization is the same as for the optimal control scenario, however, we again vary the integration tolerance using the settings as in Table 8.3. We compute a reference solution $q^\star \approx (2.92, 1.80, 2.07, 2.90)^T$ with HiFi Gauss–Newton, an integration tolerance of $TOL_{\text{rel}} = 1 \times 10^{-7}$, and a termination tolerance of TOL_{GN} close to machine precision.

	TOL_{rel}	TOL_{GN}	TOL_n	λ_{TOL}	s_{TOL}
Settings (1)	10^{-4}	10^{-4}	5×10^{-4}	5×10^{-3}	5×10^{-2}
Settings (2)	10^{-6}	10^{-6}	5×10^{-6}	5×10^{-4}	5×10^{-3}

Table 8.3.: Settings for parameter estimation tests

We compare again the HiFi optimization with the DEPOD-PE algorithm and the same algorithm where a standard POD/DEIM basis is used. Analogous to the optimal control case in the diagram in Figure 8.4 we display the distance to the reference solutions in the 2-norm (dark gray bars) and the residual error (light grey bar) which is the difference between the least squares functional $0.5 \cdot \|\mathcal{F}(q^\star)\|^2$ of the reference solution and the HiFi least squares

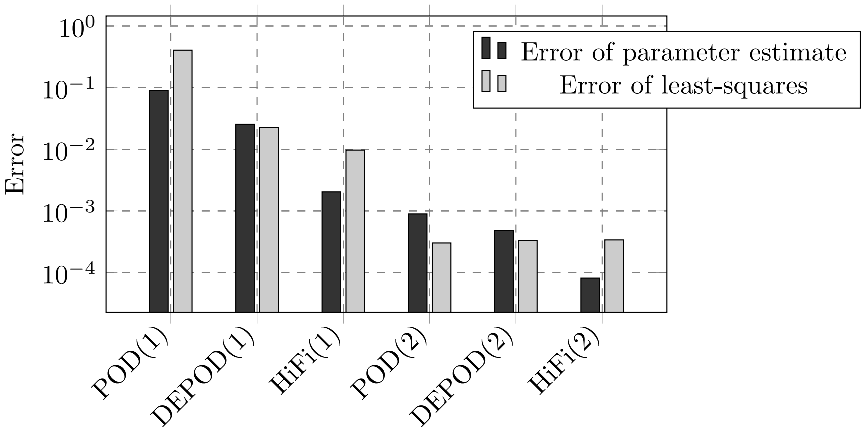

Figure 8.4.: Error of parameter estimates in 2-norm and residual errors, i.e., the distance between the reference residual $0.5 \cdot \|\mathcal{F}(q^\star)\|^2$ and the HiFi residual evaluated with the estimate, for Settings (1)/(2) and POD, DEPOD, and HiFi optimization.

functional evaluated with the corresponding solution. In both settings the DEPOD-PE algorithm is able to find better approximations than the POD-PE. Also similar to the first application in chapter 7 the parameters in the problem are identified well. Thus, we have fast local convergence which allows the HiFi optimization to make great progress in the last step before the termination criterion is satisfied while the surrogates are limited by their approximation properties.

	Settings (1)		Settings (2)	
	HiFi	DEPOD	HiFi	DEPOD
Major Iterations	-	3	-	4
Minor Iterations	17	22	19	25
Minor Iter. Time	123s	26s	183s	41s
Total Time	123s	46s	183s	84s
k, ℓ in last iteration		22 / 38	-	33 / 59

Table 8.4.: Comparison of HiFi Gauss–Newton and the DEPOD-PE algorithm for the predator-prey example and two different accuracy settings.

In Table 8.4 we show the runtime comparison between HiFi Gauss–Newton and the DEPOD-PE algorithm for the two settings. Due to the fast local convergence the number of Gauss–Newton steps is again relatively small in either case. We obtain savings from the DEPOD-PE algorithm which become smaller with increasing accuracy demands for the same reasons as in the optimal control case.

We conclude the chapter noting that for the Lotka-Volterra problem we are able to gain runtime savings from the DEPOD-OC and DEPOD-PE algorithms and a superior performance of a derivative-extended basis in comparison to the standard POD basis. As the number of required POD basis functions for each component is relatively small, the overall POD system is still small and the reduced-order modeling approach efficient. In the next section we deal with a much larger PDE systems which results in a significant increase of the surrogate system size and with this the evaluation costs of the surrogate model.

9. Heterogeneous Catalysis in Tubular Reactors

In the following application we consider a model for a reactive gas flow in a tubular reactor. The model is of heterogeneous catalysis type, i.e., the components in the process are in a different phase than the catalyst they react with. Here the flowing reactants are in gas phase. For the reactions to take place they must diffuse into the pores of the catalyst which is an amorphous porous metal oxide. The process is run in a continuously-fed fixed-bed reactor, meaning it is packed with solid catalyst particles plus some additional solid inert material and the gas is pumped in and out all the time. The process is relevant for the petrochemical industry and is based on the oxidation of the main reactant in the presence of air.

Heterogeneous catalysis models are widely used in chemical engineering and a general introduction into this model class is found in [43]. The model was developed by our industrial partners at the Scientific Computing Group of BASF SE and details of the model that we use are confidential. Thus, we restrict ourselves to the description of the model structure and a qualitative analysis of the results. Other applications in the field of heterogeneous catalysis in tubular reactors are, e.g., the modeling of Diesel Oxidation Catalysts with the goal of reduction of car exhaust (see, e.g., [52] where a detailed model description is provided).

The new challenges arising in this application are the large system size comprising 22 components, the complicated nonlinearity, and the handling of time integration where the system frequently changes between a transient and a steady-state phase.

9.1. Model description

Component type	Notation
Mole fractions in gas phase	y_j^{gas}
Temperature in gas phase	y_T^{gas}
Mole fractions in solid catalyst phase	y_j^{cat}
Temperature in solid catalyst phase	y_T^{cat}
Temperature in solid inert	y_T^{inert}
Catalyst activity	y^{act}

Table 9.1.: Overview of state variables in the heterogeneous catalysis model. y consists of six types of states and an overall of 22 state components.

Under the assumption that the gas velocity is constant along any cross section of the tubular reactor ('plug flow'), we consider a one-dimensional spatial domain $\Omega = (0, L)$ where $L = 2.81$ meters is the reactor length. The behavior in the gas phase of the nine reacting components y_j^{gas}, $j = 1, \ldots, 9$, and the temperature y_T^{gas} is modeled by a system of convection-diffusion-reaction equations. The same number of components appears again in the catalyst phase, represented by the states y_j^{cat} and y_T^{cat}. The remaining two components are the temperature in the non-reacting inert solid of the reactor fill y_T^{inert} and the activity of the catalyst y^{act}.

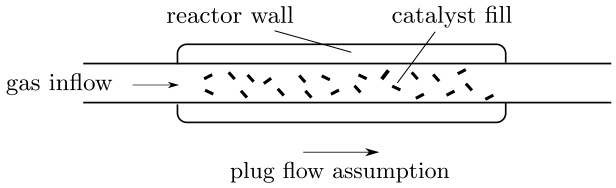

Figure 9.1.: Tubular reactor, gas flow, and catalyst fill.

We are interested in the long term behavior of the system on the time horizon $I = [0, T]$ with $T = 6652800$. The considered unit of t is seconds, thus, the process is running for 77 days. The transport, the exchange between the gas and the solid phase, and the reactions take place on a relatively small time scale. The decay of the catalyst and with this its deactivation are relatively slow. The catalyst activation reduces to about 50% over the 77 days. Thus, the simulated system is in a 'quasi steady-state' most of the time with only the catalyst activity decreasing slowly over time. The governing equations are given as

$$\begin{gathered}
\frac{\mathrm{d}y_j^{\text{gas}}}{\mathrm{d}t} = -\frac{a}{p\varepsilon_g}\frac{\partial y_j^{\text{gas}}}{\partial r} + \frac{D}{p\varepsilon_g}\frac{\partial^2 y_j^{\text{gas}}}{\partial r^2} + \Theta_j^{\text{gas}}(y), \qquad j = 1, \dots, 9, \\
\frac{\mathrm{d}y_{\mathcal{T}}^{\text{gas}}}{\mathrm{d}t} = -\frac{a}{\varepsilon_g}\frac{\partial y_{\mathcal{T}}^{\text{gas}}}{\partial r} + \frac{D}{\varepsilon_g}\frac{\partial^2 y_{\mathcal{T}}^{\text{gas}}}{\partial r^2} + \Theta_{\mathcal{T}}^{\text{gas}}(y, u), \\
\frac{\mathrm{d}y_j^{\text{cat}}}{\mathrm{d}t} = \Theta_j^{\text{cat}}(y), \qquad j = 1, \dots, 9, \\
\frac{\mathrm{d}y_{\mathcal{T}}^{\text{cat}}}{\mathrm{d}t} = \Theta_{\mathcal{T}}^{\text{cat}}(y), \qquad \frac{\mathrm{d}y_{\mathcal{T}}^{\text{inert}}}{\mathrm{d}t} = \Theta_{\mathcal{T}}^{\text{inert}}(y), \qquad \frac{\mathrm{d}y_{\text{act}}}{\mathrm{d}t} = \Theta_{\text{act}}(y), \\
\partial_\nu \frac{a}{p\varepsilon_g} y_j^{\text{gas}} + \beta_1 y_1 = \beta_2 y_{j,\text{in}}^{\text{gas}} \quad \text{on } I \times \Gamma_{\text{in}}, \qquad \partial_\nu \frac{a}{p\varepsilon_g} y_j^{\text{gas}} = 0 \quad \text{on } I \times \Gamma_{\text{out}}, \\
\partial_\nu \frac{a}{\varepsilon_g} y_{\mathcal{T}}^{\text{gas}} + \beta_1 y_1 = \beta_2 y_{\mathcal{T},\text{in}}^{\text{gas}} \quad \text{on } I \times \Gamma_{\text{in}}, \qquad \partial_\nu \frac{a}{\varepsilon_g} y_{\mathcal{T}}^{\text{gas}} = 0 \quad \text{on } I \times \Gamma_{\text{out}}.
\end{gathered} \tag{9.1}$$

The initial value condition for the catalyst activity is set to $y_s^{\text{act}} = 1$. The diffusion constant is $D = 0.1$, the impulse is $p = 3900$, and the flow velocity $a \approx 3$ is varying slightly with temperature at the reactor inflow, thus, it is constant in space. Due to the long time horizon of the model, initial value conditions for the other components have essentially no impact and are set according to the value of the corresponding state at the inflow boundary. Inflow boundary conditions are imposed weakly setting $\beta_1 = \beta_2 = 100$. Only the components in the gas phase are directly affected by spatial operators. The other components are defined by the corresponding terms Θ only, which is the part to be reduced by the DEIM projection. It is in general assumed to be nonlinear, however, for a more convenient handling of the large system we also included parts that are linear in the states. The contributions Θ to the gas phase components only contain linear operations which are the exchange rates between different phases. They are given as

$$\begin{aligned}
\Theta_j^{\text{gas}} &= \frac{k_m A_{\text{cat}}}{p\varepsilon_g}\left(\frac{y_{\mathcal{T}}^{\text{gas}}}{y_{\mathcal{T}}^{\text{cat}}} y_j^{\text{cat}} - y_j^{\text{gas}}\right), \qquad j = 1, \dots, 9, \\
\Theta_{\mathcal{T}}^{\text{gas}} &= \frac{4U}{d\varepsilon_g\rho_g C_{\text{gas}}}\left(y_{\mathcal{T}}^{\text{gas}} - \mathcal{T}_{\text{wall}}\right) + \frac{k_h A_{\text{cat}}}{\varepsilon_g\rho_g C_{\text{gas}}}\left(y_{\mathcal{T}}^{\text{gas}} - y_{\mathcal{T}}^{\text{cat}}\right) \\
&\quad + \frac{k_h A_{inert}}{\varepsilon_g\rho_g C_{\text{gas}}}\left(y_{\mathcal{T}}^{\text{gas}} - y_{\mathcal{T}}^{\text{inert}}\right),
\end{aligned}$$

The contributions Θ to the remaining components contain exchange rates between the phases as well as the reaction kinetics and the decay rate of the catalyst.

$$\begin{aligned}
\Theta_j^{\text{cat}} &= \frac{k_m A_{\text{cat}}}{p\varepsilon_p\varepsilon_s(1-\varepsilon_g)} \left(\frac{y_{\mathcal{T}}^{\text{gas}}}{y_{\mathcal{T}}^{\text{cat}}} y_j^{\text{gas}} - y_j^{\text{cat}} \right) \\
&\quad + \frac{1}{p\varepsilon_p\varepsilon_s(1-\varepsilon_g)} R_j(y^{\text{cat}}, y_{\mathcal{T}}^{\text{cat}}), \qquad j = 1, \dots, 9, \\
\Theta_{\mathcal{T}}^{\text{cat}} &= \frac{\rho_{g,\text{cat}}}{H_{\text{tot}}(1-\varepsilon_g)\varepsilon_s} \sum_{j=1}^{9} H_f R_j(y^{\text{cat}}, y_{\mathcal{T}}^{\text{cat}}) + \frac{k_h A_{\text{cat}}}{H_{\text{tot}}(1-\varepsilon_g)\varepsilon_s} \left(y_{\mathcal{T}}^{\text{gas}} - y_{\mathcal{T}}^{\text{cat}} \right), \\
\Theta_{\mathcal{T}}^{\text{inert}} &= \frac{k_h A_{\text{inert}}}{C_{\text{cat}}\rho_{\text{cat}}(1-\varepsilon_g)(1-\varepsilon_s)} \left(y_{\mathcal{T}}^{\text{gas}} - y_{\mathcal{T}}^{\text{inert}} \right), \\
\Theta_{\text{act}} &= -k_0 \exp\left(-\frac{E_s}{R_g y_{\mathcal{T}}^{\text{cat}}} \right) y_{\text{cat}}.
\end{aligned}$$

The reactions taking place in the process are shown in Figure 9.2. Each arrow corresponds to a reaction with the reactants at the start and next to the arrow while the products are placed at the arrowhead. The main product we are interested in is y_6 gained by reaction of y_1 and y_5.

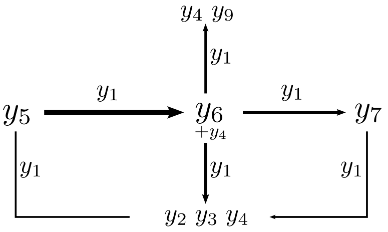

Figure 9.2.: Reactions considered in the heterogeneous catalysis model. Each arrow corresponds to a reaction with the reactants at the start and next to the arrow and the products at the arrowhead.

Over time the temperature $\mathcal{T}_{\text{wall}} = y_{\mathcal{T},\text{in}}^{\text{gas}}$ and the gas mole fractions $y_{j,\text{in}}^{\text{gas}}$, $j = 1, \dots, 7$, at the reactor inflow are varied following a given time profile. We consider a subdivision of the time horizon into 38 equally sized subintervals, defined by the time instances $0 = t_0 < \dots < t_{38} = T$, and time profiles that are piecewise constant functions on each subinterval. The time profiles for $y_{1,\text{in}}^{\text{gas}}$ (u_1), $y_{4,\text{in}}^{\text{gas}}$ (u_2) and $\mathcal{T}_{\text{wall}}$ (u_3) are shown in the top row of Figure 9.4. The state trajectories at final time T of the main product component y_6, the catalyst activity y^{act}, and the temperature in gas phase $y_{\mathcal{T}}^{\text{gas}}$ are shown in Figure 9.3.

Scaling of the problem

A scaling of the system is necessary as the order of magnitude of the temperature states is about 600 while the species are of an order of magnitude between 10^{-4} and 10^{-1}. Let $A \in \mathbb{R}^{n_x \times n_x}$ be a diagonal scaling matrix in the semi-discrete setting with n_x being the dimension of the ODE model. By setting $x = A\tilde{x}$, the HiFi problem (2.7) can be transformed into

$$MA\dot{\tilde{x}}(t) = SA\tilde{x}(t) + F(A\tilde{x}(t), q), \quad \tilde{x}(0) = A^{-1}x_s, \ t \in I.$$

Integration of this problem yields snapshots $\tilde{x}^1, \dots, \tilde{x}^m$ which have then entries of the same order of magnitude for all components if A is chosen carefully. We use the approximation

Component type	Notation
Mass transfer coefficient	k_m
Heat transfer coefficient	k_h
Catalyst surface area	A_{cat}
Inert surface area	A_{inert}
Impulse	p
Gas phase volume fraction	ε_g
Catalyst fraction of solid material	ε_s
Pore fraction of active catalysts	ε_p
Heat transfer at reactor wall	U
Tubular reactor diameter	d
Gas density	ρ_g
Density of catalyst skeleton	ρ_{cat}
Density of gas in catalyst pores	$\rho_{g,\text{cat}}$
Average heat capacity	C_{gas}
Heat capacity of catalyst solid	C_{cat}
Volume heat capacity in catalyst	H_{tot}
Standard enthalpy of formation	H_f
Pre-exponential factor for deactivation	k_0
Activation energy for deactivation	E_s
Ideal gas constant	R_g
Reaction balances for component j	R_j
Reactor wall and inflow temperature	$\mathcal{T}_{\text{wall}}$

Table 9.2.: Entities in the (nonlinear) contributions Θ.

$\tilde{x} = \Psi\widehat{x}$ of the transformed states in the POD projection and carry out the model reduction steps analogously. Note that we now use the transformed mass matrix $\tilde{M} := MA$ for the decomposition and that with this we have orthonormality of the projection matrices Ψ with respect to $\tilde{M}$.

9.2. Optimization goals

The goal of the optimal control problem is to maximize the overall amount of the gas component y_6^{gas} which is the main product of the whole process. At the same time we want to keep the deactivation of the catalyst activity as small as possible as this allows to continue running the production process efficiently. To this end, we consider the objective

$$J(y,u) = -800 \int_I y_6^{\text{gas}}(t,L)dt + \int_\Omega y_{\text{act}}(T,r)dr + \frac{5\times 10^{-6}}{2} \sum_{i=1}^{3} \int_I (u_i(t) - u_{\Omega,i})^2 dt.$$

The additional factor 800 is used to account for the larger importance of the product maximization goal and the smaller order of magnitude of y_6^{gas} in comparison to y_{act}. The optimization variable is a time-dependent vector valued function $u \in L^2(I,\mathbb{R}^3)$ consisting of the time profiles of the wall temperature $\mathcal{T}_{\text{wall}}$ and the components $y_{1,\text{in}}^{\text{gas}}$ and $y_{4,\text{in}}^{\text{gas}}$. We express this via time-dependent control functions which are piecewise constant on the 38

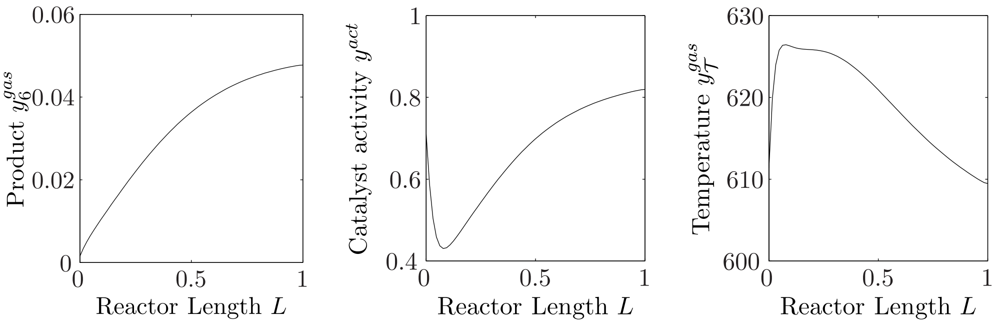

Figure 9.3.: States at final time T (in quasi steady-state) of main product component y_6, the catalyst activity y^{act}, and the temperature in gas phase $y_\mathcal{T}^{\text{gas}}$.

time subintervals, i.e., we have

$$u_1(t) = \mathcal{T}_{\text{wall}}(t), \quad u_2(t) = y_{1,\text{in}}^{\text{gas}}(t), \quad u_3(t) = y_{4,\text{in}}^{\text{gas}}(t),$$
$$u_{1,2,3}(t) \text{ constant on } t \in [t_{i-1}, t_i], \quad i = 1, \ldots, 38.$$

The time profiles before optimization that serve as initial values in the optimization are shown in the top row of Figure 9.4. We include a regularization term in the objective with constants

$$u_{\Omega,1} = 380, \quad u_{\Omega,2} = 30, \quad u_{\Omega,3} = 280.$$

As bounds we use

$$u_1^{lo} = 231.0,\ u_1^{up} = 594.1,\ u_2^{lo} = 19.6,\ u_2^{up} = 48.1,\ u_3^{lo} = 280,\ u_3^{up} = 350.$$

In the parameter estimation problem we estimate nine parameters. Among these are the pre-exponential factor k_0 and the activation energy E_s of the catalyst deactivation. The other seven parameters are not explicitly listed here. They are four pre-exponential factors and three activation energy values which enter in the computation of the balances $R_j,\ j = 1, \ldots, 9$.

During the experiment we measure the concentrations of the six components in the gas phase at the reactor outlet and the temperature values in the gas phase at 33 equally distributed locations along the reactor length. In particular we have

$$h_1(y(t)) = y_1^{\text{gas}}(t, L), \quad h_2(y(t)) = y_2^{\text{gas}}(t, L), \quad h_3(y(t)) = y_3^{\text{gas}}(t, L),$$
$$h_4(y(t)) = y_5^{\text{gas}}(t, L), \quad h_5(y(t)) = y_6^{\text{gas}}(t, L), \quad h_6(y(t)) = y_7^{\text{gas}}(t, L),$$
$$h_{6+i}(y(t)) = y_\mathcal{T}^{\text{gas}}(t, L_i), \quad 0 = L_1 < \cdots < L_{33} = L$$

with $L_{i+1} - L_i$ equally sized. While the measurement function $h(y)$ according to its definition takes arguments from H, here we measure only at the boundary. Thus, we consider $h : \mathbb{R} \to \mathbb{R}$. The measurement errors are assumed to be normally distributed with 1% standard deviation ς of the corresponding entity. Measurements are taken at the 38 time instances $\tilde{t}_i = t_i,\ i = 1, \ldots, 38$ for the t_i as defined above. We use again self-generated measurement data, i.e., we randomly perturb the measurement values with variance ς^2 and mean zero. Initial values are all set to 0.7.

9.3. Results

For the discretization we use linear finite elements in space with $N = 65$ degrees of freedom for each component. The resulting ODE system has a size of $n_x = 22 \cdot N = 1430$. The time

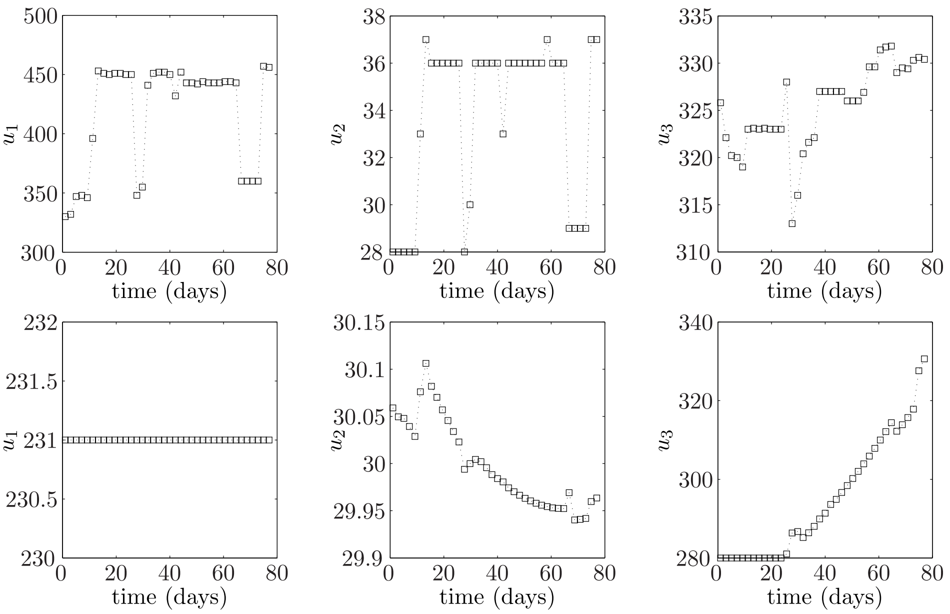

Figure 9.4.: Given initial time profiles (top row) of control functions $u_{1,2,3}$ and optimized time profiles (bottom row).

integration is done with DAESOL-II. In this application we force DAESOL-II to rebuild the right-hand side Jacobian in each time integration step. Thus, we ensure a stable integration of states, adjoints, and sensitivities (see also the discussion in Example 2.2 and in §2.5.3). The maximum order of DAESOL-II is set to $\max_{\text{order}} = 3$.

The reference solution of the optimal control problem is computed with an integration tolerance of $TOL_{\text{rel}} = 1 \times 10^{-7}$ and an SNOPT termination tolerance of $TOL_{\text{snopt}} = 5 \times 10^{-7}$. The solution trajectories are shown in the bottom row of Figure 9.4. The second control u_2 has little influence on the practical optimization goals and is mainly determined by the regularization term. The optimization with the HiFi model and the DEPOD-OC algorithm are carried out with

$$TOL_{\text{rel}} = 1 \times 10^{-6}, \qquad TOL_{\text{snopt}} = 10^{-6}.$$

The reference solution for the parameter estimation problem is computed with $TOL_{\text{rel}} = 1 \times 10^{-7}$ and $TOL_{\text{GN}} = 10^{-9}$. It is given as

$$q^\star \approx (1.59,\ 0.64,\ 0.65,\ 0.73,\ 0.79,\ 0.89,\ 1.54,\ 0.73,\ 1.23)^T.$$

The settings for the comparison between HiFi Gauss–Newton and DEPOD-PE are

$$TOL_{\text{rel}} = 1 \times 10^{-5}, \qquad TOL_{\text{GN}} = 10^{-3}, \qquad TOL_n = 10^{-2}.$$

In Table 9.3 we show the results of the comparison between a HiFi optimization and an optimization using DEPOD ROMs for the optimal control and the parameter estimation problem. We are able to reduce the runtime by a factor of almost three in the optimal control case and roughly a factor of two in the parameter estimation case. In case of DEPOD-OC we obtain a solution accuracy of 1.1×10^{-1} while with the HiFi model we have 2.5×10^{-2}. However, the error between the reference objective and the HiFi objective evaluated with

	Optimal control		Parameter estimation	
	HiFi	DEPOD	HiFi	DEPOD
λ_{TOL}/s_{TOL}	-	$2 \times 10^{-4}/2 \times 10^{-3}$	-	$2 \times 10^{-4}/2 \times 10^{-3}$
k/ℓ in last iter.	-	103/27	-	92/24
Major Iterations	-	2	-	3
Minor Iterations	28	29	10	16
Minor Iter. Time	22772s	7362s	4600s	1448s
Total Time	22772s	8939s	4600s	2459s
Error of solution	2.5×10^{-2}	1.1×10^{-1}	3.4×10^{-4}	4.6×10^{-2}

Table 9.3.: Results of comparison between HiFi optimization and DEPOD algorithms for the heterogeneous catalysis model.

the surrogate solution is smaller than 10^{-5} which is well within the application's practical accuracy requirement. In the parameter estimation case we have again the situation of fast local convergence of the Gauss–Newton method. Thus, the HiFi solution is significantly better than what we can expect from the termination tolerance $TOL_n = 10^{-2}$ of the DEPOD-PE algorithm for the same reasons as in the previous applications. Note that the accuracy requirements in both optimization examples are chosen relatively low in comparison to the other applications, however, from a practical point of view they are reasonable for the application. With POD we had no convergence at all in the parameter estimation case and we were still far from the solution after five major iterations in the optimal control case where then the runtime already exceeded the HiFi optimization costs. However, in this application the choice of the criteria λ_{TOL} and s_{TOL} and with this the number of basis functions is highly critical. The criteria should be chosen differently for POD which makes a reasonable comparison between POD and DEPOD impossible here.

In the computations only a relatively small number of basis functions per component is necessary. We needed between three and nine DEPOD basis functions in both optimization cases. This is essential for the model reduction approach to be efficient in this application as a larger amount of basis functions would destroy the savings gained with the surrogate optimization. Recall that we actually deal with a convection dominated problem, where usually a larger number of basis functions is necessary (compare the examples in §3.3.5). Moreover, we need to pass a transient phase in each of the 38 control intervals. However, as the process is run in a quasi steady-state most of the time with only the deactivation decreasing slowly the essential information of the dynamics can be captured by only a few modes.

We observe a general increase of runtime in this application, e.g., an optimal control objective and gradient computation takes about 750 seconds. The time integration of the problem is challenging as we need to resolve the transient phase before we can do larger time steps again in the quasi steady-state. Moreover, we compute the Jacobians of the right-hand sides in each time step. The time spent in the minor iterations, where only the surrogate model is used, has grown even faster than in the HiFi iterations. In the optimal control case it takes roughly 200 seconds for an evaluation of the objective and the gradient. The main reason for this is found in the way the right-hand side Jacobians are computed. As explained in chapter 6 we use seed matrices to evaluate the Jacobian for the HiFi problem. In the heterogeneous catalysis model 67 seed directions are necessary, which correspond to

67 directional derivatives of the right-hand side. As the surrogate model in general is dense we need as many directional derivatives as the system size k and in the examples we have $k \approx 100$. We observe that almost 90% of the time for the surrogate evaluation is spent in the Jacobian computation. In addition, in the heterogeneous catalysis model most of the time of the evaluation of the right-hand side is used in the nonlinear part. With DEIM we only reduce the number of nonlinear function evaluations by a factor of two, as we have already a relatively small spatial discretization in the HiFi case ($N = 65$). Thus, we end up with approximately the same effort for the Jacobian computation for HiFi and surrogate. The evaluation costs for the surrogate could be further reduced using seed matrices in the surrogate case as well. This would require the detection of a sparsity pattern from the dependencies between the system components.

The results show that the model reduction approach can also be efficiently applied to a large PDE system and we gain runtime savings with only minor loss of accuracy. With the DEPOD basis we are able to capture the necessary information with few basis functions per component. Moreover, we see that an efficient implementation of the model reduction techniques is of great importance for this application. In case of PDE systems a further reduction of costs could be achieved by exploiting the system sparsity pattern in the derivative computation.

List of Acronyms

AD	Automatic differentiation
ATO	Approximate-then-optimize
BDF	Backward Differentiation Formula
DTO	Discretize-then-optimize
DEIM	Discrete empirical interpolation method
DEDEIM	Derivative-extended discrete empirical interpolation method
DEPOD	Derivative-extended proper orthogonal decomposition
END	External numerical differentiation
FEM	Finite element method
IND	Internal numerical differentiation
IVP	Initial value problem
MOR	Model order reduction
NLP	Nonlinear programming problem
ODE	Ordinary differential equation
OTA	Optimize-then-approximate
OTD	Optimize-then-discretize
PDE	Partial differential equation
POD	Proper orthogonal decomposition
ROM	Reduced-order model
QP	Quadratic programming
SQP	Sequential quadratic programming
SVD	Singular value decomposition
VDE	Variational differential equation

Danksagung

Mit der Fertigstellung dieser Arbeit kann ich auf eine produktive und zugleich äußerst angenehme Promotionszeit an der Fakultät für Mathematik und Informatik zurückblicken. Dafür bedanke ich mich an dieser Stelle. Herausheben möchte ich meine Lehrer und Mentoren Hans Georg Bock, Stefan Körkel und Johannes Schlöder, auf deren Unterstützung ich zu jedem Zeitpunkt zählen konnte. Mein Dank geht auch an Andreas Potschka, mit dem ich viele grundlegende Fragen dieser Arbeit diskutiert habe.

Für Ideen und Unterstützung bei der Entstehung dieser Arbeit möchte ich mich des Weiteren bei verschiedenen Personen der Arbeitsgruppe Simulation und Optimierung, der BASF Junior Research Group Optimum Experimental Design, sowie anderer Universitäten und Freunden bedanken. In der Hoffnung niemanden zu vergessen haben sind dies: Jan Albersmeyer, Dörte Beigel, Jürgen Gutekunst, Christian Hoffmann, Dennis Janka, Christian Kirches, die Kaffeegruppe aus dem 4. Stock, Oliver Lass, Robert Lee, Felix Lenders, Mario Mommer, Sebastian Sager, Simeon Sauer, Pia Schmeken, Sebastian Walter, Christoph Weiler und Winnifried Wollner. Weiter möchte ich unserer langjährigen Sekräterin Margret Rothfuß sowie deren Nachfolgerin Anja Vogel und unserem Systemadministrator Thomas Kloepfer danken.

Erwähnt seien hier zudem Stefan Volkwein dem ich für die Zeit in Konstanz danke und der mich bei Fragen zur Modellreduktion unterstützt hat. Für die Ermöglichung meines Forschungsaufenthalts in Stanford danke ich Michael Saunders und der Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences.

Für die finanzielle Unterstützung geht mein Dank an die BASF SE im Rahmen der Junior Research Group Optimum Experimental Design , dem Bundesministerium für Bildung und Forschung im Rahmen des SBCancer Projekts, der Daimler AG im Rahmen des Projekts zur Untersuchung von Autoabgaskatalysatoren, und nochmals der Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences.

Heidelberg, den 3. Juni 2014, Andreas Schmidt

Bibliography

[1] K. Afanasiev and M. Hinze. Adaptive control of a wake flow using proper orthogonal decomposition. *Lecture Notes Pure Applied Mathematics*, 216:317–332, 2001.

[2] J. Albersmeyer. *Adjoint based algorithms and numerical methods for sensitivity generation and optimization of large scale dynamic systems.* PhD thesis, University of Heidelberg, 2010.

[3] J. Albersmeyer and H.G. Bock. Efficient sensitivity generation for large scale dynamic systems. Technical report, SPP 1253 Preprints, University of Erlangen, 2009.

[4] D. Amsallem and C. Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA journal*, 46(7):1803–1813, 2008.

[5] A.C. Antoulas. *Approximation of large-scale dynamical systems.* Society for Industrial and Applied Mathematics, 2005.

[6] E. Arian, M. Fahl, and E.W. Sachs. Trust-region proper orthogonal decomposition for flow control. Technical report, ICASE, 2000.

[7] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53(10):2237–2251, 2008.

[8] J.A. Atwell and B.B. King. Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Mathematical and computer modelling*, 33(1):1–19, 2001.

[9] N. Aubry. On the hidden beauty of the proper orthogonal decomposition. *Theoretical and Computational Fluid Dynamics*, 2(5-6):339–352, 1991.

[10] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43(1):9–44, 2002.

[11] W. Bangerth, T. Heister, G. Kanschat, et al. `deal.II` *Differential Equations Analysis Library, Technical Reference.* `http://www.dealii.org`.

[12] H.T. Banks, M.L. Joyner, B. Winchesky, and W.P. Winfree. Nondestructive evaluation using a reduced-order computational methodology. *Inverse Problems*, 16:1–17, 2000.

[13] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[14] I. Bauer. *Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsaufgaben in Chemie und Verfahrenstechnik.* PhD thesis, University of Heidelberg, 1999.

[15] R. Becker. *Adaptive Finite Elements for Optimal Control Problems.* PhD thesis, University of Heidelberg, 2001.

[16] D. Beigel. *Efficient goal-oriented global error estimation for BDF-type methods using discrete adjoints.* PhD thesis, University of Heidelberg, 2013.

[17] M. Bergmann, L. Cordier, and J.-P. Brancher. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of Fluids (1994-present)*, 17(9):097101–1–097101–21, 2005.

[18] G. Berkooz, P. Holmes, and J.L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25:539–575, 1993.

[19] H.G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In K.H. Ebert, P. Deuflhard, and W. Jäger, editors, *Modelling of Chemical Reaction Systems*, volume 18 of *Springer Series in Chemical Physics*, pages 102–125. Springer, Heidelberg, 1981.

[20] H.G. Bock. Recent advances in parameter identification techniques for ODE. In P. Deuflhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, pages 95–121. Birkhäuser, Boston, 1983.

[21] H.G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, volume 183 of *Bonner Mathematische Schriften.* Universität Bonn, Bonn, 1987.

[22] H.G. Bock, E.A. Kostina, and J.P. Schlöder. On the role of natural level functions to achieve global convergence for damped Newton methods. In M.J.D. Powell and S. Scholtes, editors, *System Modelling and Optimization. Methods, Theory and Applications*, pages 51–74. Kluwer, 2000.

[23] H.G. Bock and K.J. Plitt. A Multiple Shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pages 242–247, Budapest, 1984. Pergamon Press.

[24] D. Braess. *Finite elements.* Cambridge University Press, Cambridge, 2007. Theory, fast solvers, and applications in elasticity theory.

[25] A. E. Bryson and Y. Ho. *Applied optimal control.* Hemisphere, 1975.

[26] R. Buchholz, H. Engel, E. Kammann, and F. Tröltzsch. On the optimal control of the Schlögl-model. *Computational Optimization and Applications*, 56(1):153–185, 2013.

[27] T. Bui-Thanh, M. Damodaran, and K.E. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA journal*, 42(8):1505–1516, 2004.

[28] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224:880–896, 2007.

[29] J. Burkardt, M. Gunzburger, and H.-C. Lee. POD and CVT-based reduced-order modeling of Navier–Stokes flows. *Computer Methods in Applied Mechanics and Engineering*, 196(1):337–355, 2006.

[30] K. Carlberg and C. Farhat. A low-cost, goal-oriented 'compact proper orthogonal decomposition' basis for model reduction of static systems. *International Journal for Numerical Methods in Engineering*, 86(3):381–402, 2011.

[31] S. Chaturantabut and D.C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[32] S. Chaturantabut and D.C. Sorensen. A state space error estimate for POD-DEIM nonlinear model reduction. *SIAM Journal on Numerical Analysis*, 50(1):46–63, 2012.

[33] Y. Chen and J. White. A quadratic method for nonlinear model order reduction. In *Technical Proceedings of the 2000 International Conference on Modeling and Simulation of Microsystems*, 2000.

[34] S.S. Collis and M. Heinkenschloss. Analysis of the streamline upwind/Petrov–Galerkin method applied to the solution of optimal control problems. Technical report, Rice University, 2002.

[35] B. Dacorogna. *Direct methods in the calculus of variations*, volume 78. Springer, 2007.

[36] R. Dautray and J.-L. Lions. Evolution problems I. In A. Craig, editor, *Mathematical analysis and numerical methods for science and technology*, volume 5. Springer, 1992.

[37] P. Deuflhard. *Newton methods for nonlinear problems. Affine invariance and adaptive algorithms*, volume 35 of *Springer Series in Computational Mathematics.* Springer, 2006.

[38] P. Deuflhard and F. Bornemann. *Numerische Mathematik II.* de Gruyter Lehrbuch, 2002.

[39] F. Diwoky and S. Volkwein. Nonlinear boundary control for the heat equation utilizing proper orthogonal decomposition. In K.-H. Hoffmann, R.H.W. Hoppe, and V. Schulz, editors, *Fast Solution of Discretized Optimization Problems, International Series of Numerical Mathematics*, volume 138, pages 267–278. Birkhäuser, 2001.

[40] P. Eberhard and C. Bischof. Automatic Differentiation of numerical integration algorithms. *Mathematics of Computation*, 68(226):717–731, 1999.

[41] J.L. Eftang, M.A. Grepl, A.T. Patera, and E.M. Rønquist. Approximation of parametric derivatives by the empirical interpolation method. *Foundations of Computational Mathematics*, 13(5):763–787, 2013.

[42] M. Fahl. *Trust-region Methods for Flow Control based on Reduced Order Modelling.* PhD thesis, University of Trier, 2000.

[43] G.F. Froment and K.B. Bischoff. *Chemical reactor analysis and design.* Wiley, 2000.

[44] Keinosuke Fukunaga. *Introduction to statistical pattern recognition.* Access Online via Elsevier, 1990.

[45] A. V. Fursikov. *Optimal control of distributed systems: Theory and applications*, volume 187. American Mathematical Society, 1999.

[46] H. Gajewski, K. Gröger, and K. Zacharias. *Nichtlineare Operatorgleichungen und Operatordifferentialgleichungen.* Akademie-Verlag, Berlin, 1974.

[47] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International journal for numerical methods in engineering*, 81(12):1581–1608, 2010.

[48] Matthias Gerdts. *Optimal control of ODEs and DAEs.* Walter de Gruyter, 2012.

[49] P.E. Gill, W. Murray, and M.A. Saunders. User's guide for snopt version 7. Technical report, University of California, San Diego/ Stanford University, Stanford, 2008.

[50] S. Glavaski, J.E. Marsden, and Richard M Murray. Model reduction, centering, and the Karhunen-Loeve expansion. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 2, pages 2071–2076. IEEE, 1998.

[51] A. Griewank. *Evaluating derivatives, principles and techniques of algorithmic differentiation.* Number 19 in Frontiers in Applied Mathematics. SIAM, Philadelphia, 2000.

[52] A. Güthenke, D. Chatterjee, M. Weibel, B. Krutzsch, P. Kočí, M. Marek, I. Nova, and E. Tronconi. Current status of modeling lean exhaust gas aftertreatment catalysts. *Advances in Chemical Engineering*, 33:103–283, 2007.

[53] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving ordinary differential equations I*, volume 8 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin, 1993.

[54] E. Hairer and G. Wanner. *Solving ordinary differential equations II – Stiff and differential-algebraic problems.* Springer, Berlin, 1991.

[55] L. Hascoet and V. Pascual. The TAPENADE Automatic Differentiation tool: principles, model, and specification. *ACM Transactions on Mathematical Software (TOMS)*, 39(3):A1–A43, 2013.

[56] A. Hay, I. Akhtar, and J.T. Borggaard. On the use of sensitivity analysis in model reduction to predict flows for varying inflow conditions. *International Journal for Numerical Methods in Fluids*, 68(1):122–134, 2012.

[57] A. Hay, J. T. Borggaard, and Pelletier D. Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition. *Journal of Fluid Mechanics*, 629:41–72, 2009.

[58] J.S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*, volume 21 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, Cambridge, 2007.

[59] D. Hilberg, W. Lazik, and H.E. Fiedler. The application of classical POD and snapshot POD in a turbulent shear layer with periodic structures. In J.P. Bonnet and M.N. Glauser, editors, *Eddy Structure Identification in Free Turbulent Shear Flows*, pages 251–259. Springer, 1993.

[60] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE-constraints.* Springer, New York, 2009.

[61] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control. *Lecture Notes in Computational Science and Engineering*, 45:261–306, 2005.

[62] M. Hinze and S. Volkwein. Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition. *Computational Optimization and Applications*, 39(3):319–345, 2008.

[63] P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, coherent structures, dynamical systems and symmetry.* Cambridge University Press, 1998.

[64] D. Hömberg and S. Volkwein. Control of laser surface hardening by a reduced-order approach using proper orthogonal decomposition. *Mathematical and Computer Modelling*, 38(10):1003–1028, 2003.

[65] C. Homescu, L.R. Petzold, and R. Serban. Error estimation for reduced-order models of dynamical systems. *SIAM Journal on Numerical Analysis*, 43(4):1693–1714, 2005.

[66] C. Jörres, G. Vossen, and M. Herty. On an inexact gradient method using proper orthogonal decomposition for parabolic optimal control problems. *Computational Optimization and Applications*, 55(2):459–468, 2013.

[67] M. Kahlbacher and S. Volkwein. Model reduction by proper orthogonal decomposition for estimation of scalar parameters in elliptic PDEs. *Proceedings of ECCOMAS CFD, P. Wesseling, E. Onate, and J. Periaux (eds.), Egmont aan Zee*, 2006.

[68] M. Kahlbacher and S. Volkwein. Estimation of regularization parameters in elliptic optimal control problems by POD model reduction. In W. Mitkowski A. Korytowski, K. Malanowski and M. Szymkat, editors, *System Modelling and Optimization*, pages 307–318. Springer-Verlag, 2007.

[69] M. Kahlbacher and S. Volkwein. POD a-posteriori error based inexact SQP method for bilinear elliptic optimal control problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:491–511, 2012.

[70] I. Kalashnikova and M.F. Barone. Efficient non-linear proper orthogonal decomposition/galerkin reduced order models with stable penalty enforcement of boundary conditions. *International Journal for Numerical Methods in Engineering*, 90(11):1337–1362, 2012.

[71] E. Kammann, F. Troeltzsch, and S. Volkwein. A-posteriori error estimation for semilinear parabolic optimal control problems with application to model reduction by POD. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47:555–581, 2013.

[72] S. Körkel. *Numerische Methoden für Optimale Versuchsplanungsprobleme bei nichtlinearen DAE-Modellen.* PhD thesis, University of Heidelberg, 2002.

[73] B. Kragel. *Streamline diffusion POD models in optimization.* PhD thesis, University of Trier, 2005.

[74] K. Kunisch and S. Volkwein. Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. *Journal of Optimization Theory and Applications*, 102:345–371, 1999.

[75] K Kunisch and S Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90(1):117–148, 2001.

[76] K Kunisch and S Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40(2):492–515, 2002.

[77] K. Kunisch and S. Volkwein. Proper orthogonal decomposition for optimality systems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(1):1–23, 2008.

[78] K. Kunisch and S. Volkwein. Optimal snapshot location for computing POD basis functions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(3):509–529, 2010.

[79] K. Kunisch, S. Volkwein, and L. Xie. HJB-POD-based feedback design for the optimal control of evolution problems. *SIAM Journal on Applied Dynamical Systems*, 3(4):701–722, 2004.

[80] O. Lass and S. Volkwein. *Parameter identification for nonlinear elliptic-parabolic systems with application in Lithium-ion battery modeling.* Library of University of Konstanz, 2013.

[81] O. Lass and S. Volkwein. POD Galerkin schemes for nonlinear elliptic-parabolic systems. *SIAM Journal on Scientific Computing*, 35(3):A1271–A1298, 2013.

[82] R.J. LeVeque. *Finite volume methods for hyperbolic problems.* Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2002.

[83] M. Li and P.D. Christofides. Optimal control of diffusion-convection-reaction processes using reduced-order models. *Computers & chemical engineering*, 32(9):2123–2135, 2008.

[84] J.L. Lumley. Coherent structures in turbulence. In *Transition and turbulence*, volume 1, pages 215–242, 1981.

[85] M. Mangold and M. Krasnyk. Application of proper orthogonal decomposition to particulate processes. In *Mathematical Modelling*, volume 7, pages 728–733, 2012.

[86] M. Meyer and H.G. Matthies. Efficient model reduction in non-linear dynamics using the Karhunen–Loeve expansion and dual-weighted-residual methods. *Computational Mechanics*, 31(1-2):179–191, 2003.

[87] J. Nocedal and S.J. Wright. *Numerical optimization.* Springer Verlag, 2006.

[88] A. Noor and J. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA Journal*, 18:455–462, 1980.

[89] A.T. Patera and G. Rozza. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations. *MIT Pappalardo Graduate Monographs in Mechanical Engineering*, 2006.

[90] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2:559–572, 1901.

[91] J. Peterson. The reduced basis method for incompressible viscous flow calculations. *SIAM Journal on Scientific and Statistical Computing*, 10(4):777–786, 1989.

[92] T. Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation*, 45:487–496, 1985.

[93] A. Potschka. *A direct method for the numerical solution of optimization problems with time-periodic PDE constraints.* PhD thesis, University of Heidelberg, 2012.

[94] M. Rathinam and L.R. Petzold. Dynamic iteration using reduced order models: a method for simulation of large scale modular systems. *SIAM Journal on Numerical Analysis*, 40(4):1446–1474, 2002.

[95] M. Rathinam and L.R. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41:1893–1925, 2003.

[96] S. Ravindran. Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition. *SIAM Journal on Scientific Compting*, 23(6):1924–1942, 2002.

[97] A.M. Rehm, E.Y. Scribner, and H.M. Fathallah-Shaykh. Proper orthogonal decomposition for parameter estimation in oscillating biological networks. *Journal of Computational and Applied Mathematics*, 258:135–150, 2014.

[98] A. Rosenfeld and A.C. Kak. *Digital picture processing*, volume 1. Elsevier, 1982.

[99] E.W. Sachs and S. Volkwein. POD Galerkin approximations in PDE-constrained optimization. *GAMM-Mitteilungen*, 33:194–208, 2010.

[100] S. Sager. *Numerical methods for mixed–integer optimal control problems.* Der andere Verlag, Tönning, Lübeck, Marburg, 2005.

[101] A. Sandu. On the properties of Runge–Kutta discrete adjoints. In V. Alexandrov, G. van Albada, P. Sloot, and J. Dongarra, editors, *Computational Science – ICCS 2006*, volume 3994 of *Lecture Notes in Computer Science*, pages 550–557. Springer Berlin / Heidelberg, 2006.

[102] A. Sandu. Reverse Automatic Differentiation of linear multistep methods. In C.H. Bischof, H.M. Bücker, P. Hovland, U. Naumann, and J. Utke, editors, *Advances in Automatic Differentiation*, volume 64 of *Lecture Notes in Computational Science and Engineering*, pages 1–12. Springer-Verlag, Berlin, 2008.

[103] W. H. Schilders, H. A. Vorst, and J. Rommes. *Model order reduction: Theory, research aspects and applications.* Springer, 2008.

[104] A. Schmidt, A. Potschka, S. Körkel, and H.G. Bock. Derivative-extended reduced-order modeling for parameter estimation. *SIAM Journal on Scientific Compting*, 35(6):A2696–A2717, 2013.

[105] R. Serban, C. Homescu, and L.R. Petzold. The effect of problem perturbations on nonlinear dynamical systems and their reduced-order models. *SIAM Journal on Scientific Computing*, 29(6):2621–2643, 2007.

[106] S.Y. Shvartsman and I.G. Kevrekidis. Low-dimensional approximation and control of periodic solutions in spatially extended systems. *Physical Review E*, 58(1):361–368, 1998.

[107] L. Sirovich. Turbulence and the dynamics of coherent structures. 1. coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.

[108] G. Söderlind. The logarithmic norm. History and modern theory. *BIT Numerical Mathematics*, 46(3):631–652, 2006.

[109] D. B. Szyld. The many proofs of an identity on the norm of oblique projections. *Numerical Algorithms*, 42(3-4):309–323, 2006.

[110] A. Theodoropoulou, R.A. Adomaitis, and E. Zafiriou. Model reduction for optimization of rapid thermal chemical vapor deposition systems. *IEEE Transactions on Semiconductor Manufacturing*, 11(1):85–98, 1998.

[111] F. Tröltzsch. *Optimal control of partial differential equations*, volume 112. American Mathematical Society, 2010.

[112] F. Tröltzsch and S. Volkwein. POD a-posteriori error estimates for linear-quadratic optimal control problems. *Computational Optimization and Applications*, 44(1):83–115, 2009.

[113] A. Verhoeven. *Redundancy reduction of IC models by multirate time-integration and model order reduction*. PhD thesis, Eindhoven University of Technology, 2008.

[114] S. Volkwein. Optimal control of a phase-field model using proper orthogonal decomposition. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 81(2):83–97, 2001.

[115] S. Volkwein. Optimality system POD and a-posteriori error analysis for linear-quadratic problems. *Control & Cybernetics*, 40(4):1109–1124, 2011.

[116] S. Volkwein. Model reduction using proper orthogonal decomposition. `http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching`, 2013. lecture notes.

[117] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of operations research*, 37(1):357–373, 1992.

[118] A. Walther. Automatic Differentiation of explicit Runge–Kutta methods for optimal control. *Computational Optimization and Applications*, 36:83–108, 2007.

[119] A. Walther and A. Griewank. Getting started with ADOL-C. *Combinatorial Scientific Computing*, pages 181–202, 2012.

[120] J. Wang and N. Zabaras. Using bayesian statistics in the estimation of heat source in radiation. *International Journal of Heat and Mass Transfer*, 48(1):15–29, 2005.

[121] G. Weickum, M. S. Eldred, and K. Maute. A multi-point reduced-order modeling approach of transient structural dynamics with application to robust design optimization. *Structural and Multidisciplinary Optimization*, 38(6):599–611, 2009.

[122] C. Winton, J. Pettway, C. T. Kelley, S. Howington, and O. J. Eslinger. Application of proper orthogonal decomposition (POD) to inverse problems in saturated groundwater flow. *Advances in Water Resources*, 34(12):1519–1526, 2011.

[123] J. Wloka. *Partielle Differentialgleichungen: Sobolevräume u. Randwertaufgaben*. B.G. Teubner, Stuttgart, 1982.

[124] K. Zhou, J.C. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall New Jersey, 1996.

[125] R. Zimmermann. Gradient-enhanced surrogate modeling based on proper orthogonal decomposition. *Journal of Computational and Applied Mathematics*, 237(1):403–418, 2013.