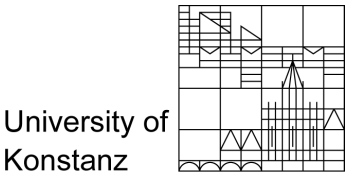
Diploma Thesis

Reduced-order methods for a parametrized model for erythropoiesis involving structured population equations with one structural variable

submitted by

Dennis Beermann

at the



Department of Mathematics and Statistics

in cooperation with the

Konstanz, 30.01.2015

Supervisor and 1st Reviewer: Prof. Dr. Stefan Volkwein, University of Konstanz

2nd Reviewer: Prof. Dr. Franz Kappel, University of Graz

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Diplomarbeit mit dem Thema:

Reduced-order methods for a parametrized model for erythropoiesis involving structured population equations with one structural variable

selbstständig verfasst und keine anderen Hilfsmittel als die angegebenen benutzt habe. Die Stellen, die anderen Werken dem Wortlaut oder dem Sinne nach entnommen sind, habe ich in jedem einzelnen Falle durch Angaben der Quelle, auch der benutzten Sekundärliteratur, als Entlehnung kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Konstanz, den 30.01.2015

(Dennis Beermann)

Acknowledgement

This diploma thesis is dedicated first and foremost to my family who has never ceased to show their full support during the years of my studies. It is a great comfort to know that they will always be there to fall back on in hard times, for which I would like to thank them.
Second, I am grateful for having had Prof. Dr. Stefan Volkwein as my supervisor during this thesis. His experience in application-oriented mathematics in general and in Proper Orthogonal Decomposition in particular has been immensely helpful on countless occasions. Furthermore, he always welcomed and encouraged cooperation within his team, thereby making it easy to find one's way in more complex and difficult subject areas.
I would also like to thank Dr. Doris Fürtinger from the Renal Research Institute for her forthcoming help with questions concerning biological and medical issues, as well as Prof. Dr. Franz Kappel from the University of Graz for agreeing to be the second reviewer of this thesis.
My last thanks go out to Laura Lippmann and Felicitas Binder who have been working on similar problems at the time and have helped to examine and confirm some of my numerical results.

This diploma thesis has been performed in cooperation with the Renal Research Institute.

ABSTRACT

The thesis investigates a one-dimensional, hyperbolic evolution equation containing one structural variable, with a particular focus on a model of erythropoiesis developed by Fürtinger et al. in 2012. Three different discretization techniques which all result in so-called high-fidility or detailed solutions are introduced and discussed. The methods used include Finite Differences and a polynomial representation of the structural variable. Viewed from the perspective of Optimal control, the model takes the form of a Parametrized Partial Differential Equation (P^2DE) where both the control and other data values are treated as parameters of the equation. This places the problem into a multi-query context, making model order reduction (MOR) techniques conceivable. Reduced basis (RB) strategies are employed to reduce the dimension of the utilized discretization spaces with a Galerkin projection. The reduced space is generated by applying a Greedy algorithm with methods including both the addition of single snapshots as well as Proper Orthogonal Decomposition (POD). In order to assess the error between the detailed and the reduced solution, two a-posteriori estimators are introduced and analyzed. Algorithmically, an offline/online decomposition scheme is used to enable efficient computations of both the reduced solutions and the estimators. Lastly, numerical experiments are presented to evaluate the feasibility of model order reduction techniques for the problem at hand.

Contents

1

Introduction & Outline

In recent years, Model Order Reduction (MOR) techniques have emerged as a powerful tool in the context of multi-query computations of Differential Equations. The model usually takes the form of a Parametrized Partial Differential Equation (P^2DE), which is a PDE depending on a parameter $\nu \in \mathscr{P}$, where $\mathscr{P} \subset \mathbb{R}^d$ is a set containing all admissible parameters. Many real world problems can be described by using a P^2DE, including parameter identification, design optimization or – as in our case – optimal control with PDE constraints. Likewise, the parameter ν can represent a variety of things, e.g. a material constant, geometric properties of the domain, a control value or a combination of the above. It is usually necessary to repeatedly solve the P^2DE numerically for many different values of ν, thereby creating a demand for efficient treatment in terms of computation time. For parabolic and hyperbolic equations, the numerical solutions can be described by a trajectory $\{y_N^k(\nu)\}_{k=0}^K \subset X_N$ where X_N is an N-dimensional Hilbert space and $k \in \{0, ..., K\}$ is the time variable corresponding to a time grid $t_0 < ... < t_K$. These solutions are called *detailed or high-fidelity solutions*. Typically, MOR is achieved using Reduced-Basis (RB) methods wherein X_N is replaced by a *reduced basis space* $X_H \subset X_N$ of significantly lower dimension H. X_H is chosen in such a way that it represents the detailed solutions under variations of the parameter ν. Using a Galerkin projection of the discretized P^2DE from X_N to X_H, a *reduced solution* $\{y_H^k(\nu)\}_{k=0}^K \subset X_H$ is computed along with an a-posteriori error estimator. In order to compute both of these in a efficient way,

an offline/online decomposition is usually employed, splitting the computations into *offline* values which are parameter-independent, and *online* values that are parameter-dependent. Whereas the former only need to be calculated once, the latter have to be updated for every variation of ν.
The following will outline and briefly summarize the remaining chapters of the thesis:

Chapter 2 will introduce the most important concepts that are used later on. The first section will present the most common components of constrained optimization theory, including the Lagrange function along with necessary conditions of first and second order. In the next section, we take a look at the Singular Value Decomposition (SVD) of a real matrix and its applications, mostly as far as operator norms are concerned. These are used in the next section to show how Proper Orthogonal Decomposition (POD) vectors are computed. POD is introduced as the problem of approximating several data vectors by only a few orthonormal vectors, and is formulated by two equivalent constrained optimization problems. The necessary conditions described in the first section are used to prove that the solution can, indeed, be obtained by a SVD of the data matrix. Furthermore, it is shown that the singular values of this matrix can be used to estimate the error of the approximation. In the fourth and last section, we establish that the Legendre polynomials are a family of orthogonal functions in $L^2(-1,1)$. Finally, a recursion formula is derived which will be used later on for the discretization of the upcoming P^2DE.

Chapter 3 will explain the phenomenon of erythropoiesis through the introduction of the P^2DE for this thesis and the demonstration of the underlying biological model. The P^2DE represents a population of CFU-E cells under the influence of external administration of the hormone Erythropoietin (EPO). By controlling the amount of injected EPO and formulating a desired state of a constant cell population, an optimal control problem is introduced which develops the context for the subsequent utilization of MOR.

Chapter 4 will focus on discretizing the P^2DE in three different ways, thus resulting in the detailed solutions identified above. For the first two ways, a polynomial space is introduced to perform a semidiscretization, turning the PDE into an Ordinary Differential Equation (ODE) in the very same way as it was done in [7]. Afterwards, two different single-step methods are used for the time discretization: A ϑ-method interpolating between an explicit and an implicit Euler scheme as well as the classical Runge-Kutta method (RK4). For the third discretization option, a Finite Difference (FD) scheme is employed using a Forward Euler method for

the structural variable and again a ϑ-method for the time discretization.

Chapter 5 will describe the generation of a reduced basis that spans a low-dimensional subspace of the discretized solution space. This is done by using an iterative method called a Greedy search, which looks for parameters of the P^2DE that are badly represented by the current basis and have to be better incorporated by additional basis vectors. Two major strategies are presented, namely the *Single-Time strategy* (which adds one single snapshot to the current basis) and the *POD strategy* (which compresses the information of an entire solution trajectory into a few vectors). Having found a suitable basis, it is shown how the recursion by which the detailed solutions are determined is projected onto the reduced space by a Galerkin projection. Furthermore, two different error estimators are derived that are used to assess the error between the reduced and detailed solution without actually having to compute the latter. Lastly, the offline/online decomposition is introduced for both the reduced solution and the error estimators.

In **Chapter 6**, experimental results are presented that analyze various aspects of the introduced methods. After the definition of general framework conditions for the problem at hand, a first analysis focuses on the performance of the three discretization techniques which lead to the detailed solutions. This is mainly done in order to identify good parameter choices, which is necessary to obtain suitable working conditions for the reduced basis algorithms. In the second section, MOR results are presented, including the qualitative and quantitative analysis of the error estimators as well as the generated reduced spaces. For the latter, comparisons are made regarding the impact of different RB strategies within the Greedy algorithm on the quality of the built space. Furthermore, the domain $\mathscr{P}$ of admissible parameters is examined as to which parameters were preferred more often than others during the search. Lastly, the computation times for the detailed solvers, the reduced solvers and the Greedy algorithm are investigated and compared, thereby assessing the question whether the application of MOR techniques is reasonable for the problem at hand.

Finally, **Chapter 7** summarizes the results of the thesis and presents an outlook to further possible studies for the subject matter.

2

Basics

2.1 Constrained Optimization

In this section, we consider the following equality-constrained optimization problem:

$$\min_{x \in \mathbb{R}^M} J(x) \qquad \text{s.t. } e(x) = 0 \tag{2.1}$$

where $J : \mathbb{R}^M \to \mathbb{R}$ is called the cost function and $e : \mathbb{R}^M \to \mathbb{R}^\ell$ the constraint function. ℓ is the number of constraints. We define the **Lagrange function**

$$L : \mathbb{R}^M \times \mathbb{R}^\ell \to \mathbb{R}, \quad L(x, \mu) := J(x) + \mu^T e(x)$$

and further introduce the **feasible set** $\mathcal{F} := \{x \in \mathbb{R}^M : e(x) = 0\}$.

Definition 2.1 (Solutions and regular points)
Let $x^* \in \mathcal{F}$ be a feasible point.

a) x^* is called a **global solution** of (2.1) if $J(x^*) \leq J(x)$ holds true for all $x \in \mathcal{F}$.
b) x^* is called a **local solution** of (2.1) if there is a neighbourhood $U \subset \mathbb{R}^M$ of x^* such that $J(x^*) \leq J(x)$ holds true for all $x \in \mathcal{F} \cap U$.
c) x^* is called a **regular point** of (2.1) if the gradients $\nabla e_1(x^*), ..., \nabla e_\ell(x^*) \in \mathbb{R}^M$ are linearly independent.

Theorem 2.2 (First order necessary condition)
Assume that $J \in C^1(\mathbb{R}^M)$ and $e \in C^1(\mathbb{R}^M, \mathbb{R}^\ell)$. Let $x^* \in \mathcal{F}$ be a local solution as

well as a regular point of (2.1). Then there exists a unique **Lagrange multiplier** $\mu^* \in \mathbb{R}^\ell$ satisfying

$$0 = \nabla_x L(x^*, \mu^*) = \nabla J(x^*) + \sum_{i=1}^{\ell} \mu_i^* \nabla e_i(x^*)$$

Proof. See for example the proof of Theorem 12.1 in [17]. □

Theorem 2.3 (Second order necessary condition)
Assume that $J \in C^2(\mathbb{R}^M)$ and $e \in C^2(\mathbb{R}^M, \mathbb{R}^\ell)$. Let $x^* \in \mathcal{F}$ be a local solution as well as a regular point of (2.1) with according Lagrange multiplier $\mu^* \in \mathbb{R}^\ell$. Then the matrix

$$\nabla_{xx} L(x^*, \mu^*) = \nabla^2 J(x^*) + \sum_{i=1}^{\ell} \mu_i^* \nabla^2 e_i(x^*)$$

is positive semidefinite on $\ker e'(x^*)$, meaning $v^T \nabla_{xx} L(x^*, \mu^*) v \geq 0$ holds true for all $v \in \ker e'(x^*)$. Here, $e'(x^*) \in \mathbb{R}^{\ell \times M}$ denotes the Jacobi matrix of e which is given by $(e'(x^*))_{im} = \partial_{x_m} e_i(x^*)$ for $m = 1, ..., M$, $i = 1, ..., \ell$.

Proof. See for example the proof of Theorem 12.5 in [17]. □

2.2 Singular Value Decomposition

Theorem 2.4 (Spectral Theorem)
Let $A \in \mathbb{R}^{N \times N}$ be a symmetric matrix with eigenvalues $\lambda_1, ..., \lambda_N \in \mathbb{R}$. Then an orthogonal matrix $U \in \mathbb{R}^{N \times N}$ exists such that

$$U^T A U \;=\; \operatorname{diag}(\lambda_1, ..., \lambda_N)$$

Proof. See for example [6, Section 5.6]. □

Apart from this spectral decomposition which exists for symmetric quadratic matrices, there is another decomposition which exists for any matrix and is called the Singular Value Decomposition (SVD):

Theorem 2.5 (Singular Value Decomposition)
Let $Y \in \mathbb{R}^{M \times N}$ be a an arbitrary real-valued matrix. Then there are orthogonal

matrices $V \in \mathbb{R}^{M\times M}$, $U \in \mathbb{R}^{N\times N}$ as well as a diagonal matrix $D = \text{diag}(\sigma_1, ..., \sigma_d) \in \mathbb{R}^{d\times d}$ such that

$$V^T Y U = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} =: \Sigma \in \mathbb{R}^{M\times N}$$

The values $\sigma_1 \geq ... \geq \sigma_d > 0$ are called the **singular values** of Y. If we write the matrices using column vectors, i.e. $V = [v^1, ..., v^M]$ as well as $U = [u^1, ..., u^N]$, then these vectors are eigenvectors to $YY^T \in \mathbb{R}^{M\times M}$ respectively $Y^TY \in \mathbb{R}^{N\times N}$. The corresponding eigenvalues are $\lambda_i = \sigma_i^2$ for $i = 1, ..., d$ and $\lambda_i = 0$ for $i > d$. Furthermore, we have

$$Yu^i = \sigma_i v^i, \quad Y^T v^i = \sigma_i u^i \qquad \text{for } i = 1, ..., d \tag{2.2}$$

Proof. It is obvious that $Y^TY \in \mathbb{R}^{N\times N}$ is a symmetrical matrix, meaning that by Theorem 2.4, there exists an orthogonal matrix $U \in \mathbb{R}^{N\times N}$ satisfying $U^TY^TYU = \text{diag}(\lambda_1, ..., \lambda_N)$ where $\lambda_1, ..., \lambda_N \in \mathbb{R}$ are the eigenvalues of Y^TY. Since Y^TY is in addition positive semidefinite, all eigenvalues are nonnegative and we can assume without loss of generality that $\lambda_1 \geq ... \geq \lambda_d > 0$ as well as $\lambda_{d+1} = ... = \lambda_N = 0$ where d is the rank of Y^TY.
We will now split the orthogonal matrix into $U = [U_1, U_2]$ with $U_1 \in \mathbb{R}^{N\times d}$, $U_2 \in \mathbb{R}^{N\times(N-d)}$. This means that the i-th column in U_1 is an eigenvector of Y^TY to the eigenvalue $\lambda_i > 0$ whereas each column in U_2 is an eigenvector to 0. Furthermore, let us define the matrix $D := \text{diag}(\sigma_1, ..., \sigma_d) \in \mathbb{R}^{d\times d}$ with $\sigma_i = \sqrt{\lambda_i}$. Inserting this into the spectral decomposition $U^TY^TYU = \text{diag}(\lambda_1, ..., \lambda_N)$ from above yields

$$\begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} Y^TY \, [U_1, U_2] = \begin{bmatrix} U_1^TY^TYU_1 & U_1^TY^TYU_2 \\ U_2^TY^TYU_1 & U_2^TY^TYU_2 \end{bmatrix} = \begin{bmatrix} D^2 & 0 \\ 0 & 0 \end{bmatrix}$$

Based on that, we introduce the matrix $V_1 := YU_1D^{-1} \in \mathbb{R}^{M\times d}$ and observe that

$$V_1^TV_1 = D^{-1}U_1^TY^TYU_1D^{-1} = D^{-1}D^2D^{-1} = \mathbb{1}_d$$

where $\mathbb{1}_d \in \mathbb{R}^{d\times d}$ denotes the d-dimensional unit matrix. This in turn means that V_1 consists of pairwise orthonormal columns, allowing it to be upgraded to an orthogonal matrix $V \in \mathbb{R}^{M\times M}$ which we write as $V = [V_1, V_2]$. Furthermore, we have by definition $V_1^TYU_1 = D$ which ultimately results in

$$V \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} U^T = [V_1, V_2] \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} = V_1DU_1^T = Y$$

This shows all claims in the theorem except (2.2). To prove this, we use the decompositions $Y = V\Sigma U^T$ and $Y^T = U\Sigma^T V^T$. These can be reshaped by utilizing the orthogonality of V and U to look like $YU = V\Sigma$ as well as $Y^T V = U\Sigma$. Looking only at the first d columns of these matrix equalities yields $Yu^i = \sigma_i v^i$ and $Y^T v^i = \sigma_i u^i$ for $i = 1, ..., d$ which had to be shown. □

Corollary 2.6
Let $Y \in \mathbb{R}^{N\times N}$ be a square matrix. Then its spectral norm $||Y||_2 = \max_{||x||_2=1} ||Yx||_2$ is given by its largest singular value.

Proof. It follows from Theorem 2.5 that a SVD of Y exists, meaning $V^T YU = \Sigma$ with orthogonal matrices $V, U \in \mathbb{R}^{N\times N}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{N\times N}$ containing the singular values on its diagonal. It was also shown in the proof of 2.5 that we can write $U^T Y^T YU = \Sigma^2$. For an arbitrary $x \in \mathbb{R}^N$ with $||x||_2 = 1$, this yields:

$$||Yx||_2^2 = x^T Y^T Yx = x^T U\Sigma^2 \underbrace{U^T x}_{=:y} = y^T \Sigma^2 y = \sum_{i=1}^{d} \sigma_i^2 y_i^2 \leq \sigma_1^2 ||y||_2^2 = \sigma_1^2$$

Note that, because of the orthogonality of U, we have $||y||_2 = ||x||_2 = 1$. This shows $||Y||_2 \leq \sigma_1$. We now choose $x := Ue_1$ where e_1 denotes the first unit vector in $\mathbb{R}^N$. This results in $||x||_2 = ||e_1|| = 1$ because of the orthogonality of U as well as $||Yx||_2^2 = \sigma_1^2$, meaning that in fact $||Y||_2 = \sigma_1$ holds true. □

Corollary 2.7
Let $W \in \mathbb{R}^{N\times N}$ be a symmetric, positive definite matrix which induces the weighted inner product $(x, y)_W := x^T Wy$ on $\mathbb{R}^N$.

a) There is a symmetric and positive definite matrix $W^{1/2} \in \mathbb{R}^{N\times N}$ satisfying $(W^{1/2})^2 = W$.
b) For any given matrix $Y \in \mathbb{R}^{N\times N}$, the operator norm with respect to the weighted inner product is given by the largest singular value of $W^{1/2} Y W^{-1/2}$, where $W^{-1/2} \in \mathbb{R}^{N\times N}$ denotes the inverse of $W^{1/2}$.

Proof. a) By Theorem 2.4, there exists a spectral decomposition $W = V^T \Sigma_W V$ with the eigenvalue matrix $\Sigma_W = \text{diag}(w_1, ..., w_N)$. Since W is positive definite, all eigenvalues are positive and the diagonal square root matrix $\Sigma_W^{1/2} = \text{diag}(w_1^{1/2}, ..., w_N^{1/2})$ exists. We define $W^{1/2} := V\Sigma_W^{1/2} V^T$ and immediately observe $(W^{1/2})^2 = W$ as well as the fact that $W^{1/2}$ is symmetric and positive definite.

b) Using the result of a), we observe for any $x \in \mathbb{R}^N$:

$$||Yx||_W^2 = x^T Y^T W Y x = x^T (W^{1/2}Y)^T (W^{1/2}Y)x = \left|\left|W^{1/2}Yx\right|\right|_2^2$$

As a result, we have for every $x \in \mathbb{R}^N$ with $||x||_W = 1$:

$$||Yx||_W = \left|\left|W^{1/2}YW^{-1/2}W^{1/2}x\right|\right|_2 \leq \left|\left|W^{1/2}YW^{-1/2}\right|\right|_2$$

Here we have made use of the fact that $||W^{1/2}x||_2 = ||x||_W = 1$. Also note that $W^{1/2}$ is regular because it is positive definite by a). The property above shows $||Y||_W \leq ||W^{1/2}YW^{-1/2}||_2$. To prove the other inequality, we choose $\widetilde{x} \in \mathbb{R}^N$ with $||\widetilde{x}||_2 = 1$ as well as

$$\left|\left|W^{1/2}YW^{-1/2}\widetilde{x}\right|\right|_2 = \max_{||z||_2=1} \left|\left|W^{1/2}YW^{-1/2}z\right|\right|_2 = \left|\left|W^{1/2}YW^{1/2}\right|\right|_2$$

This is possible because $\mathcal{S}^{N-1} := \{x \in \mathbb{R}^N : ||x||_2 = 1\}$ is a compact set and the mapping $z \mapsto ||W^{1/2}YW^{-1/2}z||$ is continuous from $\mathbb{R}^N$ to $\mathbb{R}$ as a composition of continuous functions. By setting $x := W^{-1/2}\widetilde{x}$, we obtain $||x||_W = 1$ as well as

$$||Yx||_W = \left|\left|W^{1/2}Yx\right|\right|_2 = \left|\left|W^{1/2}YW^{-1/2}\widetilde{x}\right|\right|_2 = \left|\left|W^{1/2}YW^{-1/2}\right|\right|_2$$

So in fact, $||Y||_W \geq ||W^{1/2}YW^{-1/2}||_2$ holds true as well. All in all, we have $||Y||_W = ||W^{1/2}YW^{-1/2}||_2$, which is identical to the largest singular value of $W^{1/2}YW^{-1/2}$ by Corollary 2.6.

□

2.3 Proper Orthogonal Decomposition

One important area of application for SVD is the so-called POD. Before dealing with this subject, we need to say some words about the notation in this section. Throughout the following pages, we work with variables that contain all the information of several vectors. For example, some vectors $x^1, ..., x^\ell \in \mathbb{R}^M$ will be pooled in a vector

$$x := \left(x_1^1, ..., x_M^1, \ldots, x_1^\ell, ..., x_M^\ell\right)^T =: (x^1; ...; x^\ell)^T \in \mathbb{R}^{M*\ell}$$

We will further work with matrices of according dimensions, for example a matrix $X \in \mathbb{R}^{(M*\ell)\times(N*p)}$ is to be understood as a block matrix of the following shape:

$$X = \begin{pmatrix} X^{1,1} & \dots & X^{1,p} \\ \vdots & \ddots & \vdots \\ X^{\ell,1} & \dots & X^{\ell,p} \end{pmatrix}, \qquad X^{i,j} \in \mathbb{R}^{M\times N} \quad (i = 1, ..., \ell, j = 1, ..., p)$$

Finally, we will be working with the euclidian inner product on $\mathbb{R}^M$, meaning that for $x, y \in \mathbb{R}^M$, we write $(x, y) := x^T y$.

2.3.1 Proper Orthogonal Decomposition

Suppose we are given a data matrix $Y = [y^1, ..., y^N] \in \mathbb{R}^{M\times N}$ whose column vectors are supposed to be approximated by a low-dimensional subspace $\Psi_\ell \subset \mathbb{R}^M$. If Ψ_ℓ is spanned by an orthonormal system $\{\psi^1, ..., \psi^\ell\} \subset \mathbb{R}^M$, then the projection of a data vector y^n onto Ψ^ℓ is given by $\sum_{i=1}^{\ell}(y^n, \psi^i)\psi^i$. An ideal subspace would then be given as a solution to the following constrained minimization problem:

$$\left\{ \begin{array}{c} \min\limits_{\psi^1,...,\psi^\ell\in\mathbb{R}^M} \sum\limits_{n=1}^{N} \left|\left| y^n - \sum\limits_{i=1}^{\ell}(y^n, \psi^i)\psi^i \right|\right|_2^2 \\ \text{s.t. } (\psi^i, \psi^j) = \delta_{ij} \quad \text{for } i, j = 1, ..., \ell \end{array} \right\} \tag{P^ℓ}$$

For orthonormality reasons, (P^ℓ) is equivalent to

$$\left\{ \begin{array}{c} \max\limits_{\psi^1,...,\psi^\ell\in\mathbb{R}^M} \sum\limits_{n=1}^{N}\sum\limits_{i=1}^{\ell}(y^n, \psi^i)^2 \\ \text{s.t. } (\psi^i, \psi^j) = \delta_{ij} \quad \text{for } i, j = 1, ..., \ell \end{array} \right\} \tag{$\widehat{P}^\ell$}$$

Formulation of the cost and constraint functions

If we formulate the problem $(\widehat{P}^\ell)$ as a minimization problem like in Section 2.1, we obtain the following cost function:

$$J : \mathbb{R}^{M^\ell} \to \mathbb{R}, \qquad J(\psi) := -\sum_{n=1}^{N}\sum_{i=1}^{\ell}(y^n, \psi^i)^2$$

There are a total of ℓ^2 constraints which we can model by a constraint function mapping to $\mathbb{R}^{\ell*\ell}$. This function can be given by

$$e : \mathbb{R}^{M*\ell} \to \mathbb{R}^{\ell*\ell}, \qquad e^i_j(\psi) := (\psi^i, \psi^j) - \delta_{ij} \quad (i, j = 1, ..., \ell)$$

Obviously, J as well as e are differentiable functions. The derivatives are of interest so the results of section 2.1 can be used.

Derivatives of the cost and constraint functions

For optimization purposes, we have to consider first and second derivatives of J and e. Starting with J, we get a gradient function $\nabla J : \mathbb{R}^{M*\ell} \to \mathbb{R}^{M^\ell}$ with the k-th block entry ($k = 1, ..., \ell$):

$$\begin{aligned}(\nabla J(\psi))^k = \nabla_{\psi^k} J(\psi) &= -2\sum_{n=1}^{N}(y^n, \psi^k)y^n \\ &= -2\sum_{n=1}^{N}\sum_{m=1}^{M} Y_{mn}\psi_m^k y^n = -2YY^T\psi^k\end{aligned}$$

A second derivation yields a Hessian block matrix which reads

$$\nabla^2 J(\psi) = \begin{pmatrix} -2YY^T & & \\ & \ddots & \\ & & -2YY^T \end{pmatrix} \in \mathbb{R}^{(M*\ell)\times(M*\ell)}$$

Next, we have to consider the gradients of e: For $i, j = 1, ..., \ell$, we get $\nabla e_j^i : \mathbb{R}^{M*\ell} \to \mathbb{R}^{M*\ell}$ with the k-th block entry

$$\begin{aligned}\left(\nabla e_j^i(\psi)\right)^k = \nabla_{\psi^k} e_j^i(\psi) &= \left\{ \begin{array}{ll} 2\psi^i & \text{if } k = i = j \\ \psi^i & \text{if } k \neq i, k = j \\ \psi^j & \text{if } k = i, k \neq j \\ 0 & \text{if } k \neq i, k \neq j \end{array} \right\} \\ &= \delta_{ik}\psi^j + \delta_{jk}\psi^i\end{aligned}$$

Again, a second derivation yields a Hessian Matrix $\nabla^2 e_j^i(\psi) \in \mathbb{R}^{(M*\ell)\times(M*\ell)}$ with

$$\left(\nabla^2 e_j^i(\psi)\right)^{k,r} = (\delta_{ik}\delta_{jr} + \delta_{jk}\delta_{ir})\mathbb{1}_M$$

where $\mathbb{1}_M \in \mathbb{R}^{M\times M}$ denotes the unit matrix.
Last of all, it is necessary to know the Jacobi matrix $e'(\psi) \in \mathbb{R}^{(\ell*\ell)\times(M*\ell)}$. By definition of this matrix, we have the block structure

$$e'(\psi) = \begin{pmatrix} \partial_{\psi^1} e^1(\psi) & \dots & \partial_{\psi^\ell} e^1(\psi) \\ \vdots & \ddots & \vdots \\ \partial_{\psi^1} e^\ell(\psi) & \vdots & \partial_{\psi^\ell} e^\ell(\psi) \end{pmatrix}$$

So the (i,j)-th block takes the shape

$$\left(e'(\psi)\right)^{i,j} = \partial_{\psi^j} e^i(\psi) = \begin{pmatrix} \partial_{\psi^j} e_1^i(\psi) \\ \vdots \\ \partial_{\psi^j} e_\ell^i(\psi) \end{pmatrix} = \begin{pmatrix} \left[\left(\nabla e_1^i(\psi)\right)^j\right]^T \\ \vdots \\ \left[\left(\nabla e_\ell^i(\psi)\right)^j\right]^T \end{pmatrix}$$
$$= \begin{pmatrix} \delta_{ij}\left[\psi^1\right]^T + \delta_{1j}\left[\psi^i\right]^T \\ \vdots \\ \delta_{ij}\left[\psi^\ell\right]^T + \delta_{\ell j}\left[\psi^i\right]^T \end{pmatrix} \in \mathbb{R}^{\ell\times M}$$

Lemma 2.8
For every $\psi \in \mathbb{R}^{M*\ell}$, we have the kernel representation:

$$\ker\left(e'(\psi)\right) = \left\{x \in \mathbb{R}^{M*\ell} : (x^i,\psi^j) + (x^j,\psi^i) = 0 \text{ for } i,j = 1,...,\ell\right\} \tag{2.3}$$

Proof. Let $x = (x^1;...;x^\ell)^T \in \mathbb{R}^{M*\ell}$ be given arbitrarily. Then the i-th block of $e'(\psi)x \in \mathbb{R}^{\ell*\ell}$ is

$$\left(e'(\psi)x\right)^i = \sum_{j=1}^{\ell}\left(e'(\psi)\right)^{ij} x^j = \sum_{j=1}^{\ell}\delta_{ij}\begin{pmatrix}\left[\psi^1\right]^T x^j \\ \vdots \\ \left[\psi^\ell\right]^T x^j\end{pmatrix} + \sum_{j=1}^{\ell}\begin{pmatrix}\delta_{1j}\left[\psi^i\right]^T x^j \\ \vdots \\ \delta_{\ell j}\left[\psi^i\right]^T x^j\end{pmatrix}$$
$$= \begin{pmatrix}\left[\psi^1\right]^T x^i + \left[\psi^i\right]^T x^1 \\ \vdots \\ \left[\psi^\ell\right]^T x^i + \left[\psi^i\right]^T x^\ell\end{pmatrix}$$

We immediately observe that the entire vector vanishes if and only if x satisfies the condition of the right-hand set in (2.3). □

First-order necessary condition

Now the time has come to consider the necessary condition of first order for $(\widehat{P}^\ell)$. By Theorem 2.2, we are looking for so-called critical points which are feasible vectors $\psi \in \mathscr{F}$ along with Lagrange multipliers $\mu \in \mathbb{R}^{\ell*\ell}$ satisfying

$$0 = \nabla J(\psi) + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\mu_j^i \nabla e_j^i(\psi)$$

Looking at the k-th block, this transforms into the condition

$$0 = -2YY^T\psi^k + \sum_{i=1}^{\ell}(\mu_k^i + \mu_i^k)\psi^i \tag{2.4}$$

Multiplying with $[\psi^k]^T$ from the left yields

$$[\psi^k]^T YY^T\psi^k = \mu_k^k \quad \text{for } k = 1, ..., \ell$$

which is an additional property that holds true if the first-order necessary condition holds true. It will be used later on.

Second-order necessary condition

For the necessary condition of second order which was introduced in Theorem 2.3, we take a critical point $\psi \in \mathcal{F}$ with Lagrange multiplier $\mu \in \mathbb{R}^{\ell * \ell}$ and a kernel element $x \in \ker(e'(\psi))$. This means that we have $(x^i, \psi^j) + (x^j, \psi^i) = 0$ for $i, j = 1, ..., \ell$. First of all, we compute $\nabla_{\psi\psi}L(\psi, \mu)x \in \mathbb{R}^{M*\ell}$: The k-th block is

$$\begin{aligned}
(\nabla_{\psi\psi}L(\psi,\mu)x)^k &= \left(\nabla^2 J(\psi)x\right)^k + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\mu_j^i\left(\nabla^2 e_j^i(\psi)x\right)^k \\
&= \sum_{r=1}^{\ell}\left(\nabla^2 J(\psi)\right)^{k,r}x^r + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\mu_j^i\sum_{r=1}^{\ell}\left(\nabla^2 e_j^i(\psi)\right)^{k,r}x^r \\
&= -2YY^Tx^k + \sum_{r=1}^{\ell}(\mu_r^k + \mu_k^r)x^r
\end{aligned}$$

So the second-order condition implies

$$x^T\nabla_{\psi\psi}L(\psi,\mu)x = -2\sum_{k=1}^{\ell}[x^k]^TYY^Tx^k + \sum_{k,r=1}^{\ell}(\mu_r^k + \mu_k^r)(x^k, x^r) \geq 0$$

or, equivalently

$$\sum_{k=1}^{\ell}[x^k]^TYY^Tx^k \leq \sum_{k,r=1}^{\ell}\frac{\mu_r^k + \mu_k^r}{2}(x^k, x^r) \tag{2.5}$$

In a next step, we will use the fact that the orthonormal family $\{\psi^1, ..., \psi^\ell\} \subset \mathbb{R}^M$ spans a subspace, meaning that vectors from $\mathbb{R}^M$ can be split into a component within this subspace as well as an orthogonal component. In particular, we write the k-th block of the kernel element as $x^k = \sum_{i=1}^{\ell}(x^k, \psi^i)\psi^i + z^k$ where $z^k \in$

$\{\psi^1, ..., \psi^\ell\}^\perp$. Inserting this into the inequality (2.5) yields the left-hand side

$$\sum_{k,i,j=1}^{\ell} (x^k, \psi^i)(x^k, \psi^j)[\psi^j]^T Y Y^T \psi^i$$
$$+2 \sum_{k,i=1}^{\ell} (x^k, \psi^i)[z^k]^T Y Y^T \psi^i + \sum_{k=1}^{\ell} [z^k]^T Y Y^T z^k$$

By using the first-order neccessary condition (2.4) for ψ^i, this term transforms into

$$\sum_{k,i,j,r=1}^{\ell} (x^k, \psi^i)(x^k, \psi^j) \frac{\mu_i^r + \mu_r^i}{2} [\psi^j]^T \psi^r$$
$$+2 \sum_{k,i,r=1}^{\ell} (x^k, \psi^i) \frac{\mu_i^r + \mu_r^i}{2} [z^k]^T \psi^r + \sum_{k=1}^{\ell} [z^k]^T Y Y^T z^k$$

Now, since z^k and ψ^r are orthogonal and ψ^j and ψ^r are orthonormal, the term simplifies to

$$\sum_{k,i,j=1}^{\ell} (x^k, \psi^i)(x^k, \psi^j) \frac{\mu_i^j + \mu_j^i}{2} + \sum_{k=1}^{\ell} [z^k]^T Y Y^T z^k$$

Finally, we use the kernel property (2.3) of x, meaning $(x^k, \psi^i) = -(x^i, \psi^k)$ as well as $(x^k, \psi^j) = -(x^j, \psi^k)$ which yields

$$\sum_{k,i,j=1}^{\ell} (x^i, \psi^k)(x^j, \psi^k) \frac{\mu_i^j + \mu_j^i}{2} + \sum_{k=1}^{\ell} [z^k]^T Y Y^T z^k$$

The right-hand side of (2.5) transforms to

$$\sum_{k,i,j,r=1}^{\ell} \frac{\mu_r^k + \mu_k^r}{2} (x^k, \psi^i)(x^r, \psi^j)(\psi^i, \psi^j) + \sum_{k,i,r=1}^{\ell} \frac{\mu_r^k + \mu_k^r}{2} (x^k, \psi^i)(\psi^i, z^r)$$
$$+ \sum_{k,i,r=1}^{\ell} \frac{\mu_r^k + \mu_k^r}{2} (x^r, \psi^i)(\psi^i, z^k) + \sum_{k,r=1}^{\ell} \frac{\mu_k^r + \mu_r^k}{2} (z^k, z^r)$$

Using again the orthonormality properties, this simplifies to

$$\sum_{k,i,r=1}^{\ell} \frac{\mu_r^k + \mu_k^r}{2} (x^k, \psi^i)(x^r, \psi^i) + \sum_{k,r=1}^{\ell} \frac{\mu_k^r + \mu_r^k}{2} (z^k, z^r)$$

Inserting these representations of the right- and left-hand side of (2.5) finally yields the equivalent second-order necessary condition

$$\sum_{k=1}^{\ell}[z^k]^T YY^T z^k \leq \sum_{k,r=1}^{\ell} \frac{\mu_k^r + \mu_r^k}{2}(z^k, z^r) \tag{2.6}$$

Let us now choose the kernel element more specifically: We fixate $k \in \{1, ..., \ell\}$ and choose $x^k = z$ with $z \in \{\psi^1, ..., \psi^\ell\}^\perp$, $||z|| = 1$ and $x^i = 0$ for $i \neq k$. We observe by the kernel representation (2.3) that x is indeed part of the kernel because of $(x^i, \psi^j) = 0$ for all $i, j = 1, ..., \ell$. The second-order condition (2.6) then takes the shape $z^T YY^T z \leq \mu_k^k$. Since k and z have been chosen arbitrarily and $\mu_k^k = [\psi^k]^T YY^T \psi^k$ by the first-order necessary condition, this means

$$z^T YY^T z \leq [\psi^k]^T YY^T \psi^k \quad \text{for any } k = 1, ..., \ell, z \in \{\psi^1, ..., \psi^\ell\}^\perp, ||z|| = 1$$

The only way that this can hold true is if $\psi^1, ..., \psi^\ell$ span the same subspace as the eigenvectors to the ℓ largest eigenvalues of YY^T.

Solutions and error considerations

Theorem 2.9 (Proper Orthogonal Decomposition 1)
Let $V^T YU = \Sigma$ be a SVD of Y as in Theorem 2.5. Then a global solution of (P^ℓ) and $(\widehat{P}^\ell)$ is given by $v^1, ..., v^\ell$, the first ℓ columns of V.

Proof. Since the feasible set is compact and the objective function is continuous, it is obvious that a global solution $\psi^1, ..., \psi^\ell \in \mathbb{R}^M$ exists. We have already established in the last subsections that any local solution of $(\widehat{P}^\ell)$ (and therefore also $\psi^1, ..., \psi^\ell$) has to span the same subspace as the eigenvectors to the ℓ largest eigenvalues of YY^T because of the first- and second-order necessary conditions. If we take a close look at (P^ℓ), it becomes clear that only the spanned space $\text{span}\{\psi^1, ..., \psi^\ell\}$ is relevant for the solution, meaning that if we take another orthonormal set $\widetilde{\psi}^1, ..., \widetilde{\psi}^\ell \in \mathbb{R}^M$ with $\text{span}\{\widetilde{\psi}^1, ..., \widetilde{\psi}^\ell\} = \text{span}\{\psi^1, ..., \psi^\ell\}$, the value of the cost function will be identical. This in turn means that we can directly choose $\psi^1, ..., \psi^\ell$ as orthonormal eigenvectors to the ℓ largest eigenvalues of YY^T. Recalling that these eigenvectors are given by the columns of V, we have almost found a solution.
The only problem remaining is that the largest eigenvalues of YY^T may not be unique. Therefore, let us denominate the eigenvalues of YY^T as

$$\lambda_1 \geq ... \geq \lambda_{q-1} > \lambda_q = ... = \lambda_\ell = ... = \lambda_r > \lambda_{r+1} \geq ... \geq \lambda_m \geq 0$$

We have shown that a global solution to (P^ℓ) is given by a certain combination $v^1, ..., v^{q-1}, v^{i_q}, ..., v^{i_\ell}$ where $i_q, ..., i_\ell \in \{q, ..., r\}$, because these are all the possible choices for ℓ orthonormal eigenvectors to the ℓ largest eigenvalues of YY^T. In particular, we choose $v^1, ..., v^\ell$ and observe that insertion into the goal function of $(\widehat{P}^\ell)$ yields

$$\begin{aligned}
\sum_{n=1}^{N}\sum_{i=1}^{\ell}(y^n, v^i)^2 &= \sum_{i=1}^{\ell}\sum_{n=1}^{N}\left((y^n, v^i)y^n, v^i\right) \\
&= \sum_{i=1}^{\ell}\sum_{n=1}^{N}\left(\left(\sum_{m=1}^{M} Y_{mn}v_m^i\right)y^n, v^i\right) \\
&= \sum_{i=1}^{\ell}\sum_{m=1}^{M}\left(\sum_{n=1}^{N} Y_{mn}\sum_{k=1}^{M} Y_{nk}v_k^i\right)v_m^i \\
&= \sum_{i=1}^{\ell}\sum_{m=1}^{M}(YY^Tv^i)_m v_m^i = \sum_{i=1}^{\ell}\left(YY^Tv^i, v^i\right) = \sum_{i=1}^{\ell}\lambda_i
\end{aligned}$$

This value would obviously be identical if any other choice of eigenvectors had been made above, meaning that all of these combinations present a global solution to $(\widehat{P}^\ell)$. In particular, $v^1, ..., v^\ell$ is indeed a global solution. □

Remark 2.10

Looking at the premises of Theorem 2.2 and Theorem 2.3, it has to be stated here that we did not check whether critical points $\psi \in \mathscr{F}$ are also regular points. In fact, one realises that this cannot be the case for $\ell > 1$ since the constraints $e_j^i(\psi)$ and $e_i^j(\psi)$ are identical for $i \neq j$. This obvious redundance in constraints leads to gradients $\{\nabla e_j^i(\psi)\}_{i,j=1}^{\ell}$ which will of course always be linear dependent, thus not allowing any regular points. It would be possible to rectify this by only admitting those constraint functions e_j^i where $i \geq j$, which would result in every feasible point $\psi \in \mathscr{F}$ automatically being regular. However, this would deteriorate the already complicated notation, leading to the replacement of the constraint space $\mathbb{R}^{\ell * \ell}$ by $\mathbb{R}^\ell \times \mathbb{R}^{\ell-1} \times ... \times \mathbb{R}^2 \times \mathbb{R}$. Therefore, we will forego these steps here and instead focus on further analysis of POD for more general cases.[1]

Corollary 2.11 (Error term)

Let again $V^TYU = \Sigma$ be a SVD of Y with and let $v^1, ..., v^\ell$ be the solution to (P^ℓ) consisting of the first ℓ columns of V. Then the insertion into the goal functions

[1] Further findings on POD can for example be found in [21, Chapter 2].

yields the following approximation error:

$$\varepsilon_\ell := \sum_{n=1}^{N} \left|\left| y^n - \sum_{i=1}^{\ell} (y^n, v^i) v^i \right|\right|^2 = \sum_{i=\ell+1}^{d} \lambda_i$$

where $\lambda_i = \sigma_i^2$ is the square of the i-th singular value of Y.

Proof. Since $v^1, ..., v^M$ form an orthonormal basis of $\mathbb{R}^M$, we immediately get

$$\varepsilon_\ell = \sum_{n=1}^{N} \sum_{i=\ell+1}^{M} (y^n, \psi^i)^2 = \sum_{i=\ell+1}^{M} \lambda_i = \sum_{i=1}^{d} \lambda_i$$

The last equality follows exactly like in the proof of Theorem 2.9. □

2.3.2 POD with a weighted inner product on $\mathbb{R}^M$

In addition to the matrix $Y = [y^1, ..., y^n] \in \mathbb{R}^{M \times N}$, let us now assume that we have a symmetrical, positive definite matrix $W \in \mathbb{R}^{M \times M}$ inducing a more general inner product $(\cdot, \cdot)_W$ on $\mathbb{R}^M$ by $(x, y)_W := x^T W y$. This inner product now replaces the previously euclidian product which will still be denoted by $(\cdot, \cdot)$ in this subsection. The corresponding problems to (P^ℓ) and $(\widehat{P}^\ell)$ are given by

$$\left\{ \begin{array}{c} \min\limits_{\psi^1,...,\psi^\ell \in \mathbb{R}^M} \sum\limits_{n=1}^{N} \left|\left| y^n - \sum\limits_{i=1}^{\ell} (y^n, \psi^i)_W \psi^i \right|\right|_W^2 \\ \text{s.t. } (\psi^i, \psi^j)_W = \delta_{ij} \quad \text{for } i, j = 1, ..., \ell \end{array} \right\} \tag{P_W^ℓ}$$

as well as

$$\left\{ \begin{array}{c} \max\limits_{\psi^1,...,\psi^\ell \in \mathbb{R}^M} \sum\limits_{n=1}^{N} \sum\limits_{i=1}^{\ell} (y^n, \psi^i)_W^2 \\ \text{s.t. } (\psi^i, \psi^j)_W = \delta_{ij} \quad \text{for } i, j = 1, ..., \ell \end{array} \right\} \tag{$\widehat{P}_W^\ell$}$$

Corollary 2.12 (Proper Orthogonal Decomposition 2)
If we consider the matrix $\bar{Y} := W^{1/2} Y = [\bar{y}^1, ..., \bar{y}^N] \in \mathbb{R}^{M \times N}$ with a SVD $\bar{V}^T \bar{Y} \bar{U} = \bar{\Sigma}$, the solution to (P_W^ℓ) and $(\widehat{P}_W^\ell)$ is given by $W^{-1/2}\bar{v}^1, ..., W^{-1/2}\bar{v}^\ell$ where $\bar{v}^1, ..., \bar{v}^\ell$ denote the first ℓ columns of $\bar{V}$. Inserting this solution into (P_W^ℓ) yields the approximation error

$$\varepsilon_\ell^W := \sum_{n=1}^{N} \left|\left| y^n - \sum_{i=1}^{\ell} (y^n, \psi^i)_W \psi^i \right|\right|_W^2 = \sum_{i=\ell+1}^{M} \bar{\lambda}_i$$

with $\bar{\lambda}_i = \bar{\sigma}_i^2$, where $\bar{\sigma}_1, ..., \bar{\sigma}_d$ are the descending singular values of $\bar{Y}$.

Proof. The equivalency of the two problems (P_W^ℓ) and $(\widehat{P}_W^\ell)$ can be shown the very same way as in Theorem 2.9. We can further observe that the condition $(\psi^i, \psi^j)_W = \delta_{ij}$ is identical to $(W^{1/2}\psi^i, W^{1/2}\psi^j) = \delta_{ij}$. Inserting an arbitrary feasible vector family $\psi^1, ..., \psi^\ell$ into the goal function of $(\widehat{P}_W^\ell)$ yields

$$\begin{aligned}\sum_{n=1}^{N}\sum_{i=1}^{\ell}(y^n, \psi^i)_W^2 &= \sum_{n=1}^{N}\sum_{i=1}^{\ell}(W^{1/2}y^n, W^{1/2}\psi^i)^2 = \sum_{n=1}^{N}\sum_{i=1}^{\ell}\left(\bar{y}^n, W^{1/2}\psi^i\right)^2 \\ &\leq \sum_{n=1}^{N}\sum_{i=1}^{\ell}\left(\bar{y}^n, \bar{v}^i\right)^2 = \sum_{n=1}^{N}\sum_{i=1}^{\ell}\left(y^n, W^{-1/2}\bar{v}^i\right)_W^2\end{aligned}$$

The inequality is exactly the claim of Theorem 2.9, applied to the matrix $\bar{Y}$. Since $\{\bar{v}^1, ..., \bar{v}^\ell\}$ is orthonormal with respect to $(\cdot,\cdot)$, the set $\{W^{-1/2}\bar{v}^1, ..., W^{-1/2}\bar{v}^\ell\}$ is orthonormal with respect to $(\cdot,\cdot)_W$ and therefore a global solution to $(\widehat{P}_W^\ell)$ and (P_W^ℓ).
Inserting this solution into the goal function of (P_W^ℓ) yields

$$\begin{aligned}\varepsilon_\ell^W &= \sum_{n=1}^{N}\left|\left|y^n - \sum_{i=1}^{\ell}(y^n, W^{-1/2}\bar{v}^i)_W W^{-1/2}\bar{v}^i\right|\right|_W^2 \\ &= \sum_{n=1}^{N}\left|\left|W^{-1/2}\left[\bar{y}^n - \sum_{i=1}^{\ell}\left(W^{-1/2}\bar{y}^n, W^{-1/2}\bar{v}^i\right)_W \bar{v}^i\right]\right|\right|_W^2 \\ &= \sum_{n=1}^{N}\left|\left|\bar{y}^n - \sum_{i=1}^{\ell}(\bar{y}^n, \bar{v}^i)\bar{v}^i\right|\right|^2 = \sum_{i=\ell+1}^{M}\bar{\lambda}_i\end{aligned}$$

The last equality follows from Corollary 2.11, applied to $\bar{Y}$. □

Lemma 2.13
The solution $\psi^1, ..., \psi^\ell \in \mathbb{R}^M$ to (P_W^ℓ) and $(\widehat{P}_W^\ell)$ can be obtained by either one of the two following ways:

a) Solve the symmetric $M \times M$ eigenvalue problem $W^{1/2}YY^TW^{1/2}\bar{v} = \bar{\lambda}\bar{v}$. For the ℓ highest eigenvalues $\bar{\lambda}_1, ..., \bar{\lambda}_\ell$ and corresponding orthonormal eigenvectors $\bar{v}^1, ..., \bar{v}^\ell \in \mathbb{R}^M$, set $\psi^i := W^{-1/2}\bar{v}^i$ $(i = 1, ..., \ell)$.
b) Solve the symmetric $N \times N$ eigenvalue problem $Y^TWY\bar{u} = \bar{\lambda}\bar{u}$. For the ℓ highest eigenvalues $\bar{\lambda}_1, ..., \bar{\lambda}_\ell$ and corresponding orthonormal eigenvectors $\bar{u}^1, ..., \bar{u}^\ell \in \mathbb{R}^N$, set $\psi^i := (\bar{\lambda}_i)^{-1/2}Y\bar{u}^i$ $(i = 1, ..., \ell)$.

Proof. If we consider the matrix $\bar{Y}$ from 2.12 and observe that $\bar{Y}\bar{Y}^T = W^{1/2}YY^TW^{1/2}$ as well as $\bar{Y}^T\bar{Y} = Y^TWY$, then a) is a direct result of computing the SVD of $\bar{Y}$. We can directly deduce b) from this if we use the fact $\bar{\sigma}_i\bar{v}^i = \bar{Y}\bar{u}^i$ and $\bar{\lambda}_i = \bar{\sigma}_i^2$

from Theorem 2.5:

$$\psi^i = W^{-1/2}\bar{v}^i = \bar{\lambda}_i^{-1/2} W^{-1/2}\bar{Y}\bar{u}^i = \bar{\lambda}_i^{-1/2} Y\bar{u}^i$$

□

2.3.3 POD with a weighted norm on $\mathbb{R}^N$

In addition to a data matrix $Y \in \mathbb{R}^{M\times N}$ and a symmetric, positive definite weight matrix $W \in \mathbb{R}^{M\times M}$, let $\alpha_1, ..., \alpha_N > 0$ be given weights which induce a norm $||\cdot||_\alpha$ on $\mathbb{R}^N$ by $||x||_\alpha = (\sum_{n=1}^N \alpha_n x_n^2)^{1/2}$. Then the problems corresponding to (P_W^ℓ) and $(\widehat{P}_W^\ell)$ are given by

$$\left\{ \begin{array}{c} \min\limits_{\psi^1,...,\psi^\ell \in \mathbb{R}^M} \sum\limits_{n=1}^N \alpha_n \left|\left| y^n - \sum\limits_{i=1}^\ell (y^n, \psi^i)_W \psi^i \right|\right|_W^2 \\ \text{s.t. } (\psi^i, \psi^j)_W = \delta_{ij} \quad \text{for } i,j = 1, ..., \ell \end{array} \right\} \qquad (P_{W,\alpha}^\ell)$$

as well as

$$\left\{ \begin{array}{c} \max\limits_{\psi^1,...,\psi^\ell \in \mathbb{R}^M} \sum\limits_{n=1}^N \sum\limits_{i=1}^\ell \alpha_n (y^n, \psi^i)_W^2 \\ \text{s.t. } (\psi^i, \psi^j)_W = \delta_{ij} \quad \text{for } i,j = 1, ..., \ell \end{array} \right\} \qquad (\widehat{P}_{W,\alpha}^\ell)$$

Corollary 2.14 (Proper Orthogonal Decomposition 3)
If we define $D := \text{diag}(\alpha_1, ..., \alpha_N) \in \mathbb{R}^{N\times N}$ and consider the matrix $\bar{Y} := W^{1/2} Y D^{1/2}$ $= [\bar{y}^1, ..., \bar{y}^N] \in \mathbb{R}^{M\times N}$ with a SVD $\bar{V}^T \bar{Y} \bar{U} = \bar{\Sigma}$, the solution to $(P_{W,\alpha}^\ell)$ and $(\widehat{P}_{W,\alpha}^\ell)$ is given by $\psi^i = W^{-1/2}\bar{v}^i$ $(i = 1, ..., \ell)$ where $\bar{v}^1, ..., \bar{v}^\ell$ denote the first ℓ columns of $\bar{V}$. Inserting this solution into $(P_{W,\alpha}^\ell)$ yields the approximation error

$$\varepsilon_\ell^{W,\alpha} := \sum_{n=1}^N \alpha_n \left|\left| y^n - \sum_{i=1}^\ell (y^n, \psi^i)_W \psi^i \right|\right|_W^2 = \sum_{i=\ell+1}^M \bar{\lambda}_i \qquad (2.7)$$

with $\bar{\lambda}_i = \bar{\sigma}_i^2$, where $\bar{\sigma}_1, ..., \bar{\sigma}_M$ are the descending singular values of $\bar{Y}$.

Proof. Again, the equivalency of the problems follows directly from orthonormality arguments. We consider the goal function of $(\widehat{P}_{W,\alpha}^\ell)$ and observe

$$\sum_{n=1}^N \sum_{i=1}^\ell \alpha_n (y^n, \psi^i)_W^2 = \sum_{n=1}^N \sum_{i=1}^\ell (\alpha_n^{1/2} y^n, \psi^i)_W^2$$

Furthermore, $\alpha_n^{1/2} y^n$ is the n-th column of $YD^{1/2}$ so the problem coincides with (P_W^ℓ) respectively $(\widehat{P}_W^\ell)$ if one replaces the matrix Y by $YD^{1/2}$. This immediately yields the claim. □

Lemma 2.15
The solution $\psi^1, ..., \psi^\ell \in \mathbb{R}^M$ to $(P_{W,\alpha}^\ell)$ and $(\widehat{P}_{W,\alpha}^\ell)$ can be obtained by either one of the two following ways:

a) Solve the symmetric $M \times M$ eigenvalue problem $W^{1/2} Y D Y^T W^{1/2} \bar{v} = \bar{\lambda} \bar{v}$. For the ℓ highest eigenvalues $\bar{\lambda}_1, ..., \bar{\lambda}_\ell$ and corresponding orthonormal eigenvectors $\bar{v}^1, ..., \bar{v}^\ell \in \mathbb{R}^M$, set $\psi^i := W^{-1/2} \bar{v}^i$ $(i = 1, ..., \ell)$.
b) Solve the symmetric $N \times N$ eigenvalue problem $D^{1/2} Y^T W Y D^{1/2} \bar{u} = \bar{\lambda} \bar{u}$. For the ℓ highest eigenvalues $\bar{\lambda}_1, ..., \bar{\lambda}_\ell$ and corresponding orthonormal eigenvectors $\bar{u}^1, ..., \bar{u}^\ell \in \mathbb{R}^N$, set $\psi^i := (\bar{\lambda}_i)^{-1/2} Y D^{1/2} \bar{u}^i$ $(i = 1, ..., \ell)$.

Proof. Compare Lemma 2.13 and replace the matrix Y with $YD^{1/2}$. □

2.4 Legendre polynomials

Throughout this thesis, we work with the Hilbert space

$$L^2(a,b) := \left\{ f : (a,b) \to \mathbb{R}, f \text{ is measurable with } \int_a^b |f|^2 dx < \infty \right\}$$

where $(a,b) \subset \mathbb{R}$ is a bounded interval. The inner product is given by $(f,g)_{L^2(a,b)} := \int_a^b fg dx$ for $f, g \in L^2(a,b)$. Furthermore, we consider for $N \in \mathbb{N}_0$ the polynomial space

$$\Pi_N(a,b) := \{ p : (a,b) \to \mathbb{R}, p \text{ is a polynomial with } \deg p \leq N \}$$

It is obvious that $\Pi_N(a,b)$ is a subspace of $L^2(a,b)$ with $\dim \Pi_N(a,b) = N+1$. Therefore, it can be represented by a basis $\{L_0, ..., L_N\} \subset \Pi_N(a,b)$. The choice of this basis is important and there are many choices preferable to the mononical basis $L_n(x) = x^n$. For example, since $\Pi_N(a,b)$ is a Hilbert space with the induced inner product $(\cdot,\cdot)_{L^2(a,b)}$, an orthogonal basis would be desirable for the purposes of simplicity and stability. We restrict ourselves to the case $(a,b) = (-1,1)$ here and introduce one of the most common orthogonal polynomial systems.

Theorem 2.16 (Legendre polynomials)

Let the **Legendre polynomials** $L_0, ..., L_N \in \Pi_N(-1,1)$ be recursively defined by

$$\begin{aligned} L_0(x) &= 1 && (x \in (-1,1)) \\ L_1(x) &= x && (x \in (-1,1)) \\ nL_n(x) &= (2n-1)xL_{n-1}(x) - (n-1)L_{n-2}(x) && (n \geq 2, x \in (-1,1)) \end{aligned} \tag{2.8}$$

Then $\{L_0, ..., L_N\}$ is a basis of $\Pi_N(-1,1)$ which is orthogonal with respect to the inner product $(\cdot,\cdot)_{L^2(-1,1)}$ and satisfies $||L_n||_{L^2(-1,1)} = (n+\frac{1}{2})^{-1/2}$.

Proof. See for example chapter 22 in [1]. □

Lemma 2.17 (Properties of Legendre Polynomials)
Let $L_0, ..., L_N \in \Pi_N(-1,1)$ be the first N Legendre polynomials.

a) It is $L_n(-1) = (-1)^n$ as well as $L_n(1) = 1$ for $n = 0, ..., N$.
b) For $n = 1, ..., N$, the following identity holds true:

$$(1-x^2)L_n'(x) = -nxL_n(x) + nL_{n-1}(x) \quad (x \in (-1,1)) \tag{2.9}$$

c) For $n = 1, ..., N$, we have

$$(2n+1)L_n(x) = \frac{d}{dx}\left[L_{n+1}(x) - L_{n-1}(x)\right] \quad (x \in (-1,1)) \tag{2.10}$$

d) The derivatives can be expressed by the polynomials in the following way for $n = 1, ..., N$:

$$\begin{aligned} L_n'(x) &= \left\{ \begin{array}{ll} \sum_{j=0}^{k-1} (4j+3)\, L_{2j+1}(x) & \text{if } n = 2k \\ \sum_{j=0}^{k} (4j+1) L_{2j}(x) & \text{if } n = 2k+1 \end{array} \right\} \\ &= (2n-1)L_{n-1}(x) + (2n-5)L_{n-3}(x) + ... \end{aligned} \tag{2.11}$$

Proof. a) See for example [1, Chapter 22].
b) Can also be found in [1, Chapter 22].
c) For $n = 1$, the equality can be directly computed:

$$\frac{d}{dx}\left[L_2(x) - L_0(x)\right] = \frac{d}{dx}\left[\frac{3}{2}x^2 - 1\right] = 3x = 3L_1(x)$$

For $n \geq 2$, we utilize (2.9) and observe:

$$\begin{aligned} &(1-x^2)\frac{d}{dx}\left[L_{n+1}(x) - L_{n-1}(x)\right] \\ =& -(n+1)xL_{n+1}(x) + (n+1)L_n(x) + (n-1)xL_{n-1}(x) - (n-1)L_{n-2}(x) \end{aligned}$$

From (2.8), we can further deduce that

$$\begin{aligned}
... =& -x\left[(2n+1)xL_n(x) - nL_{n-1}(x)\right] + (n+1)L_n(x) + (n-1)xL_{n-1}(x) \\
& - (n-1)L_{n-2}(x) \\
=& -x^2(2n+1)L_n(x) + (2n-1)xL_{n-1}(x) + (n+1)L_n(x) - (n-1)L_{n-2}(x) \\
=& -x^2(2n+1)L_n(x) + nL_n(x) + (n+1)L_n(x) \\
=& (1-x^2)(2n+1)L_n(x)
\end{aligned}$$

d) By repeatedly applying (2.10), we obtain

$$\begin{aligned}
L_n'(x) &= (2n-1)L_{n-1}(x) + L_{n-2}'(x) \\
&= (2n-1)L_{n-1}(x) + (2n-5)L_{n-3}(x) + L_{n-4}'(x) \\
&= ...
\end{aligned}$$

This process can be repeated until the remaining residual term is given by $L_0'(x)$ oder $L_1'(x)$, depending on whether n is even or odd. In the even case, i.e. $n = 2k$, we have $L_0'(x) = 0$ and what remains is a sum over the polynomials of odd degrees:

$$L_n'(x) = \sum_{j=0}^{k-1}(2(2j+1)+1)L_{2j+1}(x) = \sum_{j=0}^{k-1}(4j+3)L_{2j+1}(x)$$

In the odd case, i.e. $n = 2k+1$, the opposite is the case: A sum over the polynomials of even order and a residual term of $L_1'(x) = 1 = L_0(x)$:

$$L_n'(x) = \sum_{j=1}^{k}(2(2j)+1)L_{2j}(x) + L_0(x) = \sum_{j=0}^{k}(4j+1)L_{2j}(x)$$

□

3

Model & Equation

3.1 Model

Throughout this thesis, we will consider a population of CFU-E cells under the influence of the externally administered hormone EPO which affects cell death. CFU-E cells are a stage of the erythroid lineage as can be seen in Figure 3.1. The model is based on the one presented in [7] and is assuming a constant cell reproduction as well as a cell death dependent on the current EPO concentration. In contrast to [7] where all cell stages are simulated simultaneously by a coupled system of Partial Differential Equation (PDE)s, the CFU-E population here is considered independent from other cell stages.

The equation governing the CFU-E cell population is presented in the form of a linear, hyperbolic P^2DE of first order containing initial and boundary conditions:

$$\begin{array}{rll} y_t(t,\mathbf{x}) + y_{\mathbf{x}}(t,\mathbf{x}) & = \kappa(t,u,\mu)y(t,\mathbf{x}) & \text{for } (t,\mathbf{x}) \in (0,T)\times(\underline{\mathbf{x}},\overline{\mathbf{x}}) \\ y(t,\underline{\mathbf{x}}) & = g & \text{for } t \in (0,T) \\ y(0,\mathbf{x}) & = y_0(\mathbf{x}) & \text{for } \mathbf{x} \in (\underline{\mathbf{x}},\overline{\mathbf{x}}) \end{array} \tag{3.1}$$

First of all, the variables t and $\mathbf{x}$ denote time and the maturity attribute of the population which both vary within bounded intervals $(0,T)$ and $(\underline{\mathbf{x}},\overline{\mathbf{x}})$. Therefore, we may think of cells at $\mathbf{x} = \underline{\mathbf{x}}$ as ones that have have just entered the blood stream or have just been transformed from a previous cell stage. Furthermore, $y(t,\mathbf{x})$ is a density with respect to the attribute $\mathbf{x}$, so for any $t \in [0,T]$, the total cell population at that time is given by $\int_{\underline{\mathbf{x}}}^{\overline{\mathbf{x}}} y(t,\mathbf{x})d\mathbf{x}$.

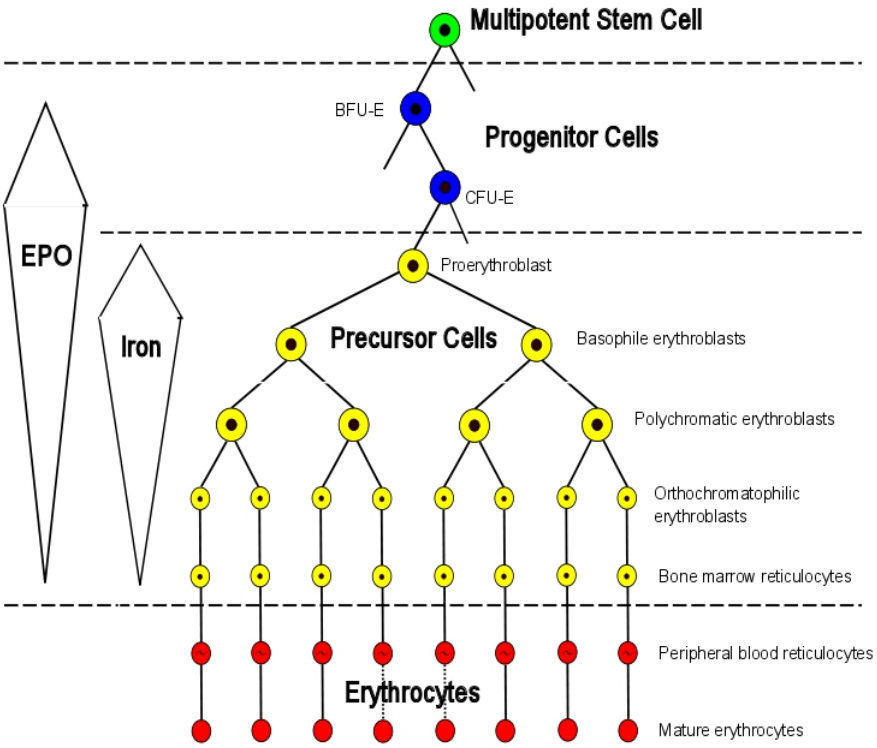


Figure 3.1: The erythroid lineage influenced by EPO concentration and iron availability.

Second, the variables u und μ are to be treated as parameters affecting the right-hand side κ of the equation, so of course the solution $y(t, \mathbf{x})$ will also depend on u and μ. However, this dependency will be omitted in the notation for simplicity purposes.
Third, the value $g \in \mathbb{R}$ represents a constant boundary condition for the cell population, meaning that a constant inlet of fresh cells is assumed. The function $y_0 : (\underline{\mathbf{x}}, \overline{\mathbf{x}}) \to \mathbb{R}$ represents an initial population at $t = 0$.

Parameter Range

We distinguish between two different kinds of parameters affecting (3.1): First of all, the parameter μ varies within a general boxed set $\mathcal{D} = [\underline{\mu}, \overline{\mu}] \subset \mathbb{R}^3$ where the interval definition is to be understood componentwise. This parameter allows for general variations in the exact form of the term κ on the right-hand side of the PDE.
Second, the parameter u is taken from another boxed set $U_{ad} = [0, 1]^m \subset \mathbb{R}^m$. It represents a control parameter which is varied in order to achieve a desired optimal

state of the solution. In practice, u_i stands for the relative amount of a hormone injected into the blood stream at a certain time t_i^* which in turn will affect the cell population.

Hormone and controls

The amount of the above hormone is described by $E(t) = \sum_{i=1}^{m} u_i \chi_i(t)/TBV$, where $\chi_1, ...\chi_m : [0,T] \to \mathbb{R}$ are bounded shape functions and $TBV > 0$ stands for the total blood volume of the subject. If $0 = t_1^* < ... < t_m^* \leq T$ are the specific injection times, the shape functions take the following form:

$$\chi_i(t) \;=\; \chi^{max} e^{-\lambda(t-t_i^*)} \mathbb{1}_{[t_i^*,T]}(t) \quad (t \in [0,T], i = 1, ..., m)$$

Here, $\mathbb{1}$ is the characteristic function, i.e. for any $A \subset [0,T]$, we have $\mathbb{1}_A(t) = 1$ if $t \in A$ and $\mathbb{1}_A(t) = 0$ if $t \notin A$. The constant $\lambda > 0$ is the degredation rate which depends on the half-life $T_{1/2} > 0$ of the hormone by $\lambda = \ln(2)/T_{1/2}$. Furthermore, $\chi^{max} > 0$ stands for the maximal dosis which can be injected, meaning that in fact $u_i \cdot \chi^{max}$ will be administered at the time t_i^*.

Proliferation and Apoptosis

The right-hand side of the PDE contains the function κ which may depend on $(t, \mathbf{x})$ and is affected by the parameters u and μ. Basically, κ can be viewed as a term controlling proliferation (i.e. cell reproduction) and apoptosis (i.e. cell death). In our case, it does not depend on $\mathbf{x}$ and takes the form $\kappa(t,u,\mu) \;=\; \beta - \alpha(t,u,\mu)$, where $\beta > 0$ denotes the constant proliferation rate and $\alpha(t,u,\mu) > 0$ the apoptosis rate. The second term is affected by time and the parameters in the following way:

$$\alpha : (0,T) \times U_{ad} \times \mathcal{D} \to \mathbb{R}$$
$$\alpha(t,u,\mu) = \frac{\mu_1}{1 + \exp(\mu_2 E(t) - \mu_3)} = \frac{\mu_1}{1 + \exp\left(\frac{\mu_2}{TBV} \sum_{i=1}^{m} u_i \chi_i(t) - \mu_3\right)}$$

3.2 Formulation as a Cauchy problem

Following the approach used in [7], we will formulate (3.1) as a Cauchy problem, since this will be useful for some of the upcoming discretization techniques. The

general Hilbert space is chosen as $L^2 := L^2(\underline{\mathbf{x}}, \overline{\mathbf{x}})$ along with an operator $\mathscr{A}(t,u,\mu) : L^2 \supset D(\mathscr{A}(t,u,\mu)) \to L^2$ for every $t \in (0,T)$, $u \in U_{ad}$, $\mu \in \mathscr{D}$:

$$\begin{aligned} D(\mathscr{A}(t,u,\mu))) = \Big\{ & v \in L^2 : v \text{ absolutely continuous on } [\underline{\mathbf{x}}, \overline{\mathbf{x}}], \\ & v(\underline{\mathbf{x}}) = 0, \kappa(t,u,\mu)v - v' \in L^2 \Big\} \\ \mathscr{A}(t)v = \kappa(t,u,\mu)v - v' & \end{aligned}$$

The system (3.1) cannot be formulated as a Cauchy problem directly due to the boundary condition at $\mathbf{x} = \underline{\mathbf{x}}$. Instead, a series of Cauchy problems is introduced whose solutions are meant to approximate the desired population density function. For this purpose, a sequence $(\delta_n)_{n\in\mathbb{N}} \subset L^2$ of functions approximating the δ-distribution at $\mathbf{x} = \underline{\mathbf{x}}$ is required, which is chosen as

$$\delta_n(\mathbf{x}) = \begin{cases} -2n^2\left(\mathbf{x} - \underline{\mathbf{x}} - \frac{1}{n}\right), & \text{for } \mathbf{x} \in \left[\underline{\mathbf{x}}, \underline{\mathbf{x}} + \frac{1}{n}\right] \\ 0, & \text{otherwise} \end{cases}$$

Using these functions, the n-th approximating Cauchy problem is introduced as

$$\boxed{\begin{aligned} \dot{y}_n(t) &= \mathscr{A}(t,u,\mu)y_n(t) + g\delta_n \quad \text{for } t \in (0,T) \\ y_n(0) &= y_0 \end{aligned}} \tag{3.2}$$

For the simplified case of time-independent $\kappa = \kappa(u,\mu)$ and zero-boundary value $g = 0$, it was argued in [7] that the problem (3.2) has a unique mild solution $y_n : (0,T) \to L^2$. Furthermore, for every $t \in (0,T)$, the series $(y_n(t))_{n\in\mathbb{N}} \subset L^2$ is a convergent sequence in L^2. The limit function defined by

$$y : (0,T) \times (\underline{\mathbf{x}}, \overline{\mathbf{x}}) \to \mathbb{R}, \quad y(t,\cdot) := \lim_{n\to\infty} y_n(t)$$

is a weak solution to (3.1). In Chapter 4, a similar method will be employed for the discretization of (3.1). This is done by replacing L^2 with a polynomial subspace of finite dimension and substituting finite-dimensional operators $\mathscr{A}_N(t,u,\mu)$ to replace $\mathscr{A}(t,u,\mu)$. Using these operators, a Cauchy problem of finite dimension will take the place of (3.2) to approximate the population density function y.

3.3 Optimal control context

From a practical point of view, EPO is administered at the time instants $t_1^*, ..., t_m^*$ because this leads to an increase in cell population. By doing so, one wishes to approximate a desired time-dependent population $y_d : (0,T) \to \mathbb{R}$ that would be beneficial for the patient's health. Furthermore, each control value $u \in U_{ad}$

represents the relative amounts of the hormone being injected. This is formally mirrored by the fact that the variable u is contained in the right-hand side of the P^{2}DE (3.1) and thereby affects its solution y.

The problem of how much EPO has to be administered in order to achieve the desired population can be expressed by an Optimal control problem. Let us therefore assume that the system (3.1) admits a unique solution

$$y \in Y := H^1(0,T;L^2) \cap L^2(0,T,H^1)$$

with the Sobolev space $H^1 := H^1(\underline{\mathbf{x}}, \overline{\mathbf{x}})$ of functions $f \in L^2$ which have a general weak derivative $f' \in L^2$. Accordingly, the Sobolev space $H^1(0,T;L^2)$ is meant to be understood as the space of functions $\widetilde{y} : (0,T) \to L^2$ having a weak derivative $\widetilde{y}' : (0,T) \to L^2$ with both these mappings being quadratically integrable, e.g. $\int_0^T ||\widetilde{y}(t)||_{L^2}^2 \, dt < \infty$. For a closer analysis of these spaces, we refer to [3].
Under the assumption that every control $u \in U_{ad}$ leads to a unique solution $y \in Y$, we can define a control-to-state operator $S(\cdot, \mu) : U_{ad} \to Y$, where $S(u,\mu) \in Y$ is the solution of (3.1) using the control u and the parameter μ. The problem for a fixated $\mu \in \mathcal{D}$ is now to find $u \in U_{ad}$ such that $S(u,\mu)$ comes as close to the desired state y_d as possible. We assume $y_d \in L^2(0,T;L^2)$ and can formulate a suitable minimization problem by

$$\min \;\; J(\widetilde{y}, u) \quad \text{s.t. } u \in U_{ad},\, \widetilde{y} = S(u,\mu) \tag{$\mathbf{P}_\mu$}$$

where $J : Y \times U_{ad} \to [0,\infty)$ is a cost function. For example, it could be given by

$$J(\widetilde{y}, u) = \frac{1}{2} \int_0^T ||\widetilde{y}(t) - y_d(t)||_{L^2}^2 \, dt \;\; + \frac{1}{2} \sum_{i=1}^m \sigma_i |u_i - \mathring{u}_i|^2 \quad (\widetilde{y} \in Y, u \in U_{ad})$$

Here, $\mathring{u} \in U_{ad}$ is a nominal control and $\sigma_1, ..., \sigma_m \in [0,\infty)$ are regulization parameters. For the solution of ($\mathbf{P}_\mu$), iterative strategies like a projected gradient descent method or a projected BFGS method are employed which require the solution of the state equation (3.1) in every iteration. This multi-query demand may be met by MOR techniques such as those introduced in Chapter 5.

4

Discretization

Before one can look toward MOR techniques or the optimal control context of the P^{2}DE (3.1), discretization methods have to be dervied which are used to solve the equation for a single parameter pair $(u, \mu) \in U_{ad} \times \mathscr{D}$. The methods presented here all share a common outline: In a first step, a semidiscretization is done by replacing the space $L^2 = L^2(\underline{\mathbf{x}}, \overline{\mathbf{x}})$ with a finite-dimensional subspace X_N, turning the PDE into an ODE. Afterwards, single-step methods are employed on a time grid $0 = t_0^* < ... < t_K = T$ to solve this. The solutions then take the form of discrete trajectories $\{y_N^k(u, \mu)\}_{k=0}^K \subset X_N$ and will later be referred to as detailed or high-fidelity solutions.

4.1 Discretization using a polynomial subspace

4.1.1 Semidiscretization

The general idea of any spatial discretization is always to substitute the infinite-dimensional space L^2 containing the maturity variable $\mathbf{x}$ with a finite-dimensional subspace $X_N \subset L^2$. For the polynomial method, X_N will be chosen as

$$X_N := \Pi_N(\underline{\mathbf{x}}, \overline{\mathbf{x}}) = \left\{ \psi \in L^2 : \psi \text{ is a polynomial with } \deg \psi \leq N \right\}$$

Of course, this means $X_N \subset C^\infty(\underline{\mathbf{x}}, \overline{\mathbf{x}})$ as well as $\dim X_N = N + 1$. Trivially, X_N is a Hilbert space with the induced inner product $(\cdot, \cdot)_{L^2}$.
Following [7], every step which was done in Section 3.2 using the space L^2 will now have to be repeated for X_N accordingly. First of all, we define an element $\delta_N \in X_N$ approximating the δ distribution by the demand $(\delta_N, \psi)_{L^2} = \psi(0)$ for

all $\psi \in X_N$. Note that such a δ_N exists and is unique by the Riesz theorem since $\psi \mapsto \psi(0)$ is linear from X_N to $\mathbb{R}$. Now, approximating operators $\mathscr{A}_N(t,u,\mu) \in L(X_N)$ are defined for every $t \in (0,T)$, $u \in U_{ad}$ and $\mu \in \mathscr{D}$, approximating the original operators $\mathscr{A}(t,u,\mu)$ in L^2 which were defined in Section 3.2:

$$\mathscr{A}_N(t,u,\mu)\psi = \kappa(t,u,\mu)\psi - \psi' - \psi(0)\delta_N \quad (\psi \in X_N)$$

The last term in this definition occurs due to the fact that in general, $\psi(0) \neq 0$ for $\psi \in X_N$, as opposed to the elements of $D(\mathscr{A}(t,u,\mu))$. The Cauchy problem approximating (3.2) is then given by

$$\boxed{\begin{array}{ll} \dot{y}_N(t) & = \mathscr{A}_N(t,u,\mu)y_N(t) + g\delta_N \quad \text{for } t \in (0,T) \\ y_N(0) & = P_N y_0 \end{array}} \tag{4.1}$$

Here, $P_N : L^2 \to X_N$ denotes the orthogonal projection in L^2 onto X_N.
In the case of $g = 0$, it was shown in [16] that $(y_N(t))_{N\in\mathbb{N}}$ is convergent in L^2 uniformly for t in bounded intervals. The limit function $y : [0,T] \times [\underline{\mathbf{x}}, \overline{\mathbf{x}}] \to \mathbb{R}$ with $y(t,\cdot) = \lim_{N\to\infty} y_N(t)$ then solves (3.1).

4.1.2 Basis choice and representations

For implementation purposes, a basis $\{e_0, ..., e_N\}$ of X_N will have to be chosen. Since orthogonality is always preferable to achieve stability, the first thing that comes to mind are the Legendre polynomials $(L_n)_{n\in\mathbb{N}_0} \subset L^2(-1,1)$ which have been introduced in Section 2.4. Rescaling $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ to $[-1,1]$, we define

$$e_n(\mathbf{x}) = \omega^{-1/2} L_n\left(-1 + 2\omega^{-1}(\mathbf{x} - \underline{\mathbf{x}})\right) \quad (\mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}])$$

where $\omega := \overline{\mathbf{x}} - \underline{\mathbf{x}}$ is length of the maturity range $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$ for the variable $\mathbf{x}$. In Figure 4.1, plots of the basis functions can be seen for the particular interval $[\underline{\mathbf{x}}, \overline{\mathbf{x}}] = [0,5]$. The next lemma proves some properties of these functions, including their orthogonality with respect to the inner product in L^2:

Lemma 4.1 (Properties of e_n)
For every $N \in \mathbb{N}_0$, the series $(e_n)_{n=0}^N \subset X_N$ satisfies

a) $(e_n, e_m)_{L^2} = (2n+1)^{-1}\delta_{nm}$ for $n, m = 0, ..., N$.
b) $e_n(\underline{\mathbf{x}}) = \omega^{-1/2}(-1)^n$ and $e_n(\overline{\mathbf{x}}) = \omega^{-1/2}$ for $n = 0, ..., N$.
c) span $\{e_0, ..., e_N\} = X_N$

Proof. a) Let $n, m \in \{0, ..., N\}$. Using the fact that the Legendre polynomials

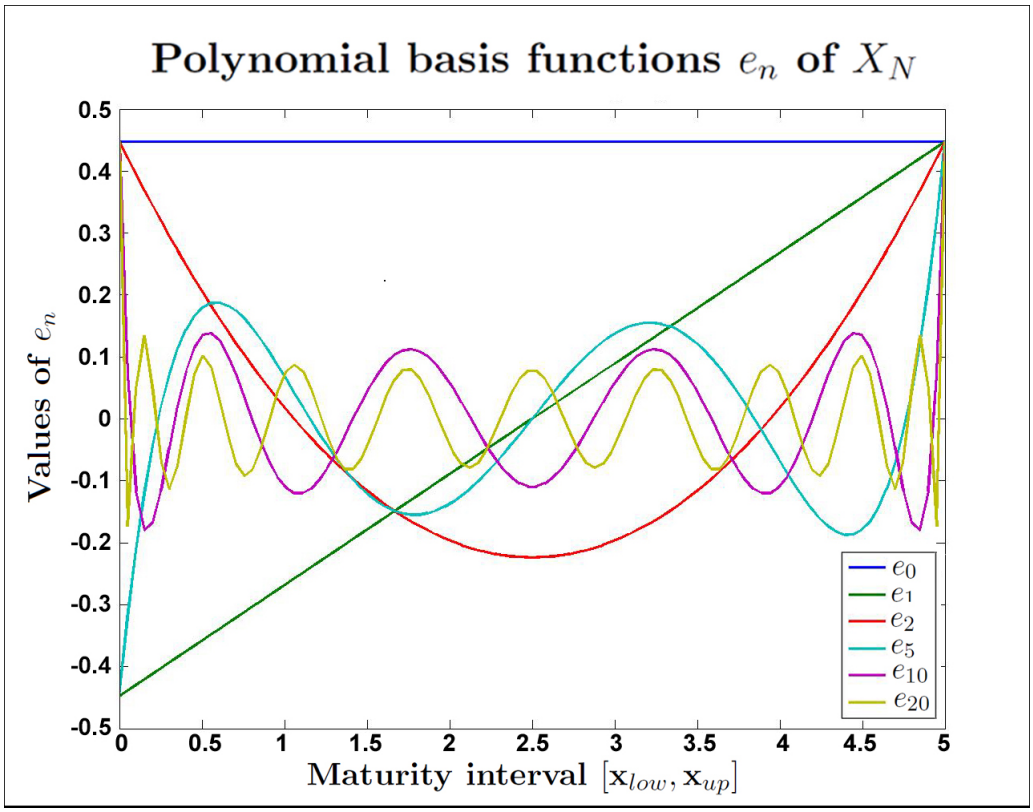


Figure 4.1: Plots of some basis polynomials $e_n : [0, 5] \to \mathbb{R}$

are pairwise orthogonal with $||L_n||_{L^2(-1,1)} = (n + \frac{1}{2})^{-1/2}$ by Theorem 2.16 a), one can see that

$$\begin{aligned}(e_n, e_m)_{L^2} &= \omega^{-1} \int_{\underline{\mathbf{x}}}^{\overline{\mathbf{x}}} L_n(-1 + 2\omega^{-1}(\mathbf{x} - \underline{\mathbf{x}}))L_m(-1 + 2\omega^{-1}(\mathbf{x} - \underline{\mathbf{x}}))d\mathbf{x} \\ &= \omega^{-1} \cdot \frac{\omega}{2} \int_0^1 L_n(\xi)L_m(\xi)d\xi = \frac{1}{2} \cdot (n + \frac{1}{2})^{-1}\delta_{nm} = (2n + 1)^{-1}\delta_{nm}\end{aligned}$$

b) It was shown in Lemma 2.17a) that the Legendre polynomials satisfy $L_n(-1) = (-1)^n$ and $L_n(1) = 1$ for $n = 0, ..., N$. Inserting this into the definition of e_n yields the proposition.

c) Follows directly from the fact that $\deg e_n = n$ for $n = 0, ..., N$.

□

Now that a basis is chosen, every vector from X_N and every operator from $L(X_N)$ that has been used until now will have to be represented by a coordinate vector or matrix with respect to this basis. Formally, this can be expressed by an

isomorphism $\Phi_N : \mathbb{R}^{N+1} \to X_N$ which in this case takes the form

$$\Phi_N z = \sum_{n=0}^{N} z_n e_n, \quad \Phi_N^{-1}\psi = \left(\frac{(\psi, e_n)_{L^2}}{\theta_n^2} \right)_{n=0}^{N} \qquad (z \in \mathbb{R}^{N+1}, \psi \in X_N)$$

where $\theta_n := ||e_n||_{L^2} = (2n+1)^{-1/2}$. If we endow $\mathbb{R}^{N+1}$ with the norm $||z||_\theta := (\sum_{n=0}^{N} \theta_n^2 z_n^2)^{1/2}$ and denote the resulting space with $\mathbb{R}_\theta^{N+1}$, Φ_N becomes additionally isometric. Using this isomorphism, many operations on X_N will prove easily computable by simple basis representations:

Lemma 4.2 (Inner Product and Derivation) a) We denote the weight matrix of the basis by $W := \operatorname{diag}(\theta_0^2, ...\theta_N^2) \in \mathbb{R}^{(N+1)\times(N+1)}$. Then the inner product takes the following representation for all $\psi, \widetilde{\psi} \in X_N$:

$$(\psi, \widetilde{\psi})_{L^2} = (\Phi_N^{-1}\psi)^T W^2 (\Phi_N^{-1}\widetilde{\psi}) = \sum_{n=0}^{N} (\Phi_N^{-1}\psi)_n (\Phi_N^{-1}\widetilde{\psi})_n \theta_n^2$$

b) Let $\partial_{\mathbf{x}} \in L(X_N)$ be the derivative operator. Then $\partial_{\mathbf{x}}$ allows for a basis representation $\partial_{\mathbf{x}} = \Phi_N B_N \Phi_N^{-1}$ with the matrix

$$B_N = \omega^{-3/2} \begin{pmatrix} 0 & 2 & 0 & 2 & \dots & \\ & 0 & 6 & 0 & \ddots & \vdots \\ & & 0 & 10 & \ddots & 4N-10 \\ & & & 0 & \ddots & 0 \\ & & & & \ddots & 4N-2 \\ & & & & & 0 \end{pmatrix} \in \mathbb{R}^{(N+1)\times(N+1)}$$

Proof. a) Direct Computation shows:

$$\begin{aligned} (\psi, \widetilde{\psi})_{L^2} &= \left(\sum_{n=0}^{N} (\Phi_N^{-1}\psi)_n e_n, \sum_{m=0}^{N} (\Phi_N^{-1}\widetilde{\psi})_m e_m \right)_{L^2} \\ &= \sum_{n,m=0}^{N} (\Phi_N^{-1}\psi)_n (\Phi_N^{-1}\widetilde{\psi})_m (e_n, e_m)_{L^2} \\ &= \sum_{n=0}^{N} (\Phi_N^{-1}\psi)_n (\Phi_N^{-1}\widetilde{\psi})_m \theta_n^2 = (\Phi_N^{-1}\psi)^T W (\Phi_N^{-1}\widetilde{\psi}) \end{aligned}$$

b) First of all, we have to show that $\partial_{\mathbf{x}} \in L(X_N)$ holds true: For every element $\psi \in X_N$, ψ is a polynomial with $\deg \psi \leq N$, meaning that ψ' is a polynomial with $\deg \psi' \leq N-1$, in particular $\psi' \in X_N$. So $\partial_{\mathbf{x}}$ is a linear operator

mapping from X_N to X_N. Since X_N is of finite dimension, this immediately ensures the continuity of $\partial_{\mathbf{x}}$, resulting in $\partial_{\mathbf{x}} \in L(X_N)$. Now we can define $B_N := \Phi_N^{-1}\partial_{\mathbf{x}}\Phi_N : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$. As a composition of linear operators, B_N itself is linear and can therefore be represented by a matrix which we also denote with $B_N \in \mathbb{R}^{(N+1)\times(N+1)}$. If we express B_N by its columns, it takes the form $B_N = (\Phi_N^{-1}e_0', ..., \Phi_N^{-1}e_N')$. In addition, the basis functions are defined as $e_n(\mathbf{x}) = \omega^{-1/2}L_n(-1 + 2\omega^{-1}(\mathbf{x} - \underline{\mathbf{x}}))$ and the derivative of L_n satisfies the recursion formula (2.11), which results in

$$e_n' = \omega^{-3/2}\left[(4(n-1)+2)e_{n-1} + (4(n-3)+2)e_{n-3} + ...\right] \quad (n = 1, ..., N)$$

Now the basis representations $\Phi_N^{-1}e_n'$ are clearly visible, resulting in the matrix B_N from the claim.

□

Now all pieces are assembled to obtain basis representations of $\mathscr{A}_N$ and δ_N which are the key components of the discretized problem (4.1). Let $A_N(t, u, \mu) \in \mathbb{R}^{(N+1)\times(N+1)}$ be the basis representation of $\mathscr{A}_N(t, u, \mu)$, i.e. $\mathscr{A}_N(t, u, \mu) = \Phi_N A_N(t, u, \mu)\Phi_N^{-1}$. This takes the form

$$A_N(t, u, \mu) = \kappa(t, u, \mu)\mathbb{1}_N - B_N - \Gamma_N \qquad (t \in (0, T), u \in U_{ad}, \mu \in \mathscr{D})$$

where $\mathbb{1}_N$ denotes the unity matrix in $\mathbb{R}^{N+1}$ and Γ_N is the basis representation for the mapping $\psi \mapsto \psi(0)\delta_N$. Similarly, let $\gamma_N \in \mathbb{R}^{N+1}$ be the basis representation of the vector δ_N, i.e. $\delta_N = \Phi_N\gamma_N$. If we can identify γ_N and Γ_N, then the solution y_N of (4.1) can be obtained by solving the following problem in $\mathbb{R}^{N+1}$:

$$\boxed{\begin{aligned} \dot{\varphi}_N(t) &= A_N(t, u, \mu)\varphi_N(t) + g\gamma_N \quad \text{for } t \in (0, T) \\ \varphi_N(0) &= \Phi_N^{-1}P_N y_0 \end{aligned}} \tag{4.2}$$

Setting $y_N(t) := \Phi_N\varphi_N(t)$ yields the solution of (4.1). Now all that is left to do is getting to γ_N and Γ_N:

Lemma 4.3 (γ_N and Γ_N)
γ_N and Γ_N take the following representations:

$$\gamma_N = \omega^{-1/2}\Big(1, -3, 5, ..., (-1)^N(2N+1)\Big)^T \in \mathbb{R}^{N+1}$$

$$\Gamma_N = \omega^{-1/2}\begin{pmatrix} 1 & -1 & \dots & 1\cdot(-1)^N \\ -3 & 3 & \dots & -3\cdot(-1)^N \\ 5 & -5 & \dots & 5\cdot(-1)^N \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \in \mathbb{R}^{(N+1)\times(N+1)}$$

Proof. We can write $\delta_N = \Phi_N\gamma_N$. For every $\psi \in X_N$, for example $\psi = \Phi_N\varphi$ with $\varphi \in \mathbb{R}^{N+1}$, δ_N satisfies $(\delta_N, \psi)_{L^2} = \psi(\underline{\mathbf{x}})$. Using Lemma 4.2a) along with Lemma 4.1b) this is equivalent to

$$\sum_{n=0}^{N}(\gamma_N)_n\psi_n\theta_n^2 = w^{-1/2}\left(\varphi_0 - \varphi_1 + \varphi_2 - \varphi_3 + ...\right)$$

Since this equality has to hold for every $\varphi \in \mathbb{R}^{N+1}$, we get $(\gamma_N)_n\theta_n^2 = (-1)^n w^{-1/2}$ and hence the proposed shape of γ_N.
For Γ_N, we first observe that as a basis representation, it takes the form

$$\begin{aligned}\Gamma_N &= \big(\Phi_N^{-1}(e_0(\underline{\mathbf{x}})\delta_N), ..., \Phi_N^{-1}(e_N(\underline{\mathbf{x}})\delta_N)\big) = (e_0(\underline{\mathbf{x}})\gamma_N, ..., e_N(\underline{\mathbf{x}})\gamma_N) \\ &= \omega^{-1/2}\left(\gamma_N, -\gamma_N, \gamma_N, ...\right)\end{aligned}$$

Along with the above shape of γ_N, this yields the proposition. □

At this point, the semidiscretization is complete. What remains to be done is the time integration of the resulting ODE (4.1) respectively (4.2).

4.1.3 Time Discretization

To solve the ODE (4.1) respectively (4.2), a time grid $0 = t_0 < ... < t_K = T$, is introduced which we assume to be equidistant for simplicity of notation, i.e. $t_k = k\Delta t$ where $\Delta t > 0$ is a time step size. We will examine two discretization methods in the following sections, both of which will result in a discrete system of the following kind:

$$\boxed{\begin{aligned}\mathcal{L}_I^k(u,\mu)y_N^{k+1} &= \mathcal{L}_E^k(u,\mu)y_N^k + \Delta t z_N^k(u,\mu) \quad \text{for } k = 0, ..., K-1 \\ y_N^0 &= P_N y_0\end{aligned}} \tag{4.3}$$

Here, $\{\mathscr{L}^k_{I/E}(u,\mu)\}_{k=0}^{K-1} \subset L(X_N)$ is a series of implicit and explicit operators and $\{z^k_N(u,\mu)\}_{k=0}^{K-1} \subset X_N$ are inhomogeneities. Let $\widehat{\mathscr{L}}^k_{I/E}(u,\mu) := \Phi_N^{-1}\mathscr{L}^k_{I/E}(u,\mu)\Phi_N \in \mathbb{R}^{(N+1)\times(N+1)}$ as well as $\widehat{z}^k_N(u,\mu) = \Phi_N^{-1} z^k_n(u,\mu) \in \mathbb{R}^{N+1}$ denote the basis representations in $\mathbb{R}^{N+1}$, then the basis representation form of (4.3) is given by

$$\boxed{\begin{aligned} \widehat{\mathscr{L}}^k_I(u,\mu)\varphi_N^{k+1} &= \widehat{\mathscr{L}}^k_E(u,\mu)\varphi_N^k + \Delta t \widehat{z}^k_N(u,\mu) \quad \text{for } k = 0,...,K-1 \\ \varphi_N^0 &= \Phi_N^{-1} P_N y_0 \end{aligned}} \tag{4.4}$$

This notation is strongly influenced by [10], where it is called a *parametrized evolution scheme* and it is assumed that $\mathscr{L}^k_I(u,\mu)$ respectively $\widehat{\mathscr{L}}^k_I(u,\mu)$ are positive definite operators. The latter is often the case for parabolic equations but does not hold true here.

A ϑ-method

We introduce a parameter $\vartheta \in [0,1]$ interpolating between a time-explicit ($\vartheta = 0$) and time-implicit ($\vartheta = 1$) method. Abbreviating $y^k_N \approx y_N(t_k)$ for $k = 0,...,K$, the ODE in (4.1) is replaced by

$$\frac{y_N^{k+1} - y_N^k}{\Delta t} = \vartheta \mathscr{A}^{k+1}_N(u,\mu) y_N^{k+1} + (1-\vartheta)\mathscr{A}^k_N(u,\mu) y_N^k + g\delta_N$$

Reforming this yields the system (4.3) with the operators

$$\mathscr{L}^k_I(u,\mu) = 1 - \vartheta \Delta t \mathscr{A}^{k+1}_N(u,\mu), \quad \mathscr{L}^k_E(u,\mu) = 1 + (1-\vartheta)\Delta t \mathscr{A}^k_N(u,\mu)$$

and the inhomogeneity $z^k_N(u,\mu) = g\delta_N$. Note that in this case, 1 is not be understood as a simple scalar but as the implied scalar multiplication in X_N, i.e. the identity mapping $1 : X_N \to X_N$, $\psi \mapsto 1 \cdot \psi = \psi$. The basis representation form (4.4) is then reached with the operators

$$\widehat{\mathscr{L}}^k_I(u,\mu) = 1 - \vartheta \Delta t A^{k+1}_N(u,\mu), \quad \widehat{\mathscr{L}}^k_E(u,\mu) = 1 + (1-\vartheta)\Delta t A^k_N(u,\mu)$$

as well as $\widehat{z}^k_N(u,\mu) = g\gamma_N$. Again, 1 denotes the scalar multiplication operator, this time in $\mathbb{R}^{N+1}$.

The classical Runge-Kutta method

The classical Runge-Kutta method is a one-step method of fourth order and has the following tableau:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

This means that the solution at the next time step is computed by

$$\frac{y_N^{k+1} - y_N^k}{\Delta t} = \frac{B_1^k}{6} + \frac{B_2^k}{3} + \frac{B_3^k}{3} + \frac{B_4^k}{6} \tag{4.5}$$

where $B_1^k, ..., B_4^k$ are the weights which depend on the right-hand side of (4.1). We will in the following omit the (u, μ)-argument for reasons of clarity and abbreviate $\mathscr{A}_N^{k+\frac{1}{2}} := \mathscr{A}_N\left(t_k + \frac{1}{2}\right)$. The weights then take the shape

$$\begin{aligned}
B_1^k &= \mathscr{A}_N^k y_N^k + g\delta_N \\
B_2^k &= \mathscr{A}_N^{k+\frac{1}{2}}\left(y_N^k + \frac{\Delta t}{2} B_1^k\right) + g\delta_N \\
&= \frac{\Delta t}{2}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^k y_N^k + \mathscr{A}_N^{k+\frac{1}{2}}\left(y_N^k + \frac{\Delta t}{2} g\delta_N\right) + g\delta_N \\
B_3^k &= \mathscr{A}_N^{k+\frac{1}{2}}\left(y_N^k + \frac{\Delta t}{2} B_2^k\right) + g\delta_N \\
&= \frac{\Delta t^2}{4}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^k y_N^k + \frac{\Delta t}{2}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}}\left(y_N^k + \frac{\Delta t}{2} g\delta_N\right) \\
&\quad + \mathscr{A}_N^{k+\frac{1}{2}}\left(y_N^k + \frac{\Delta t}{2} g\delta_N\right) + g\delta_N \\
B_4^k &= \mathscr{A}_N^{k+1}\left(y_N^k + \Delta t B_3^k\right) + g\delta_N \\
&= \frac{\Delta t^3}{4}\mathscr{A}_N^{k+1}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^k y_N^k + \frac{\Delta t^2}{2}\mathscr{A}_N^{k+1}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}}\left(y_N^k + \frac{\Delta t}{2} g\delta_N\right) \\
&\quad + \Delta t\mathscr{A}_N^{k+1}\mathscr{A}_N^{k+\frac{1}{2}}\left(y_N^k + \frac{\Delta t}{2} g\delta_N\right) + \mathscr{A}_N^{k+1}\left(y_N^k + \Delta t g\delta_N\right) + g\delta_N
\end{aligned}$$

Inserting these weights into (4.5) yields the form (4.3) where the operators are

given by $\mathscr{L}_I^k = 1$ and

$$\mathscr{L}_E^k = 1 + \frac{\Delta t}{6}\left(\mathscr{A}_N^k + 4\mathscr{A}_N^{k+\frac{1}{2}} + \mathscr{A}_N^{k+1}\right) + \frac{\Delta t^2}{6}\left(\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^k + \mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}} + \mathscr{A}_N^{k+1}\mathscr{A}_N^{k+\frac{1}{2}}\right)$$
$$+ \frac{\Delta t^3}{12}\left(\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^k + \mathscr{A}_N^{k+1}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}}\right) + \frac{\Delta t^4}{24}\mathscr{A}_N^{k+1}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^k$$

The inhomogeneities on the other hand are

$$z_N^k = \Big[1 + \frac{\Delta t}{6}\left(2\mathscr{A}_N^{k+\frac{1}{2}} + \mathscr{A}_N^{k+1}\right) + \frac{\Delta t^2}{12}\left(\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}} + \mathscr{A}_N^{k+1}\mathscr{A}_N^{k+\frac{1}{2}}\right)$$
$$+ \frac{\Delta t^3}{24}\mathscr{A}_N^{k+1}\mathscr{A}_N^{k+\frac{1}{2}}\mathscr{A}_N^{k+\frac{1}{2}}\Big] g\delta_N$$

The basis representations in (4.4) take the shape $\widehat{\mathscr{L}}_I^k = 1$ and

$$\widehat{\mathscr{L}}_E^k = 1 + \frac{\Delta t}{6}\left(A_N^k + 4A_N^{k+\frac{1}{2}} + A_N^{k+1}\right) + \frac{\Delta t^2}{6}\left(A_N^{k+\frac{1}{2}}A_N^k + A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}} + A_N^{k+1}A_N^{k+\frac{1}{2}}\right)$$
$$+ \frac{\Delta t^3}{12}\left(A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}}A_N^k + A_N^{k+1}A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}}\right) + \frac{\Delta t^4}{24}A_N^{k+1}A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}}A_N^k$$

Last of all, we have

$$\widehat{z}_N^k = \Big[1 + \frac{\Delta t}{6}\left(2A_N^{k+\frac{1}{2}} + A_N^{k+1}\right) + \frac{\Delta t^2}{6}\left(A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}} + A_N^{k+1}A_N^{k+\frac{1}{2}}\right)$$
$$+ \frac{\Delta t^3}{12}A_N^{k+1}A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}}\Big] g\gamma_N$$

4.2 Discretization using Finite Differences (FD)

As is typical for a FD method, a spatial grid $\underline{\mathbf{x}} = \mathbf{x}_0 < ... < \mathbf{x}_{N_x} = \overline{\mathbf{x}}$ is introduced to turn the PDE (3.1) into an ODE. Assuming that an equidistant grid is used, i.e. $\mathbf{x}_i = \mathbf{x}_0 + i\Delta\mathbf{x}$ with a step size $\Delta\mathbf{x} > 0$, the spatial derivatives $y_{\mathbf{x}}(\mathbf{x})$ at the grid point $\mathbf{x} = \mathbf{x}_i$ can be approximated by a forward Euler discretization as follows:

$$y_{\mathbf{x}}(t, \mathbf{x}_i) \approx \frac{y(t, \mathbf{x}_{i+1}) - y(t, \mathbf{x}_i)}{\Delta\mathbf{x}} \qquad (t \in (0, T), i = 0, ..., N_x - 1)$$

Utilizing this representation, the PDE (3.1) is replaced by the following ODE system:

$$\begin{aligned} y_t(t,\mathbf{x}_i) + \tfrac{y(t,\mathbf{x}_{i+1})-y(t,\mathbf{x}_i)}{\Delta \mathbf{x}} &= \kappa(t,u,\mu)y(t,\mathbf{x}_i) && \text{for } t \in (0,T), \\ & && \text{and } i = 0,...,N_x - 1 \\ y(t,\mathbf{x}_0) &= g && \text{for } t \in (0,T) \\ y(0,\mathbf{x}_i) &= y_0(\mathbf{x}_i) && \text{for } i = 0,...,N_x \end{aligned} \tag{4.6}$$

For the time discretization, another ϑ-method is introduced like in Section 4.1.3: Let $\vartheta \in [0,1]$ again be the parameter interpolating between the fully explicit ($\vartheta = 0$) and fully implicit ($\vartheta = 1$) time discretization. Furthermore, we abbreviate $y_i^k \approx y(t_k, \mathbf{x}_i)$ for $i = 0,...,N_x$ and $k = 0,...,K$. Applying the θ-method results in the following discrete system for $i = 1,...,N_x$ and $k = 0,...,K-1$:

$$\begin{aligned} \frac{y_{i-1}^{k+1} - y_{i-1}^k}{\Delta t} = \vartheta &\left(\kappa^{k+1}(u,\mu) y_{i-1}^{k+1} - \frac{y_i^{k+1} - y_{i-1}^{k+1}}{\Delta x} \right) \\ + (1-\vartheta) &\left(\kappa^{k}(u,\mu) y_{i-1}^{k} - \frac{y_i^{k} - y_{i-1}^{k}}{\Delta x} \right) \end{aligned} \tag{4.7}$$

Note that in comparison to the system (4.6), the index i has been shifted to $i-1$. We introduce for $k = 0,...,K$ the vector $y^k \in \mathbb{R}^{N_x}$ with $y^k = (y_1^k,...,y_{N_x}^k)^T$. Furthermore, let $B_{N_x} \in \mathbb{R}^{N_x \times N_x}$ be given by

$$B_{N_x} = \begin{pmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix}$$

We then observe that we can write $(y_0^k,...,y_{N_x-1}^k)^T = B_{N_x} y^k + g e_1$ where $e_1 \in \mathbb{R}^{N_x}$ denotes the first unit vector. This is due to the fact that $y_0^k = g$ always holds true because of the boundary condition at $\mathbf{x} = \underline{\mathbf{x}}$. Using this, (4.7) can be vectorized as follows:

$$\begin{aligned} &\left(1 - \Delta t \vartheta \kappa^{k+1}(u,\mu)\right)(B_{N_x} y^{k+1} + g e_1) + \vartheta \frac{\Delta t}{\Delta x}(y^{k+1} - B_{N_x} y^{k+1} - g e_1) \\ &= (1 + \Delta t(1-\vartheta)\kappa^k(u,\mu))(B_{N_x} y^k + g e_1) - (1-\vartheta)\frac{\Delta t}{\Delta x}\left(y^k - B_{N_x} y^k - g e_1\right) \end{aligned}$$

This can be written in the following form which is of the same shape as (4.3) or (4.4):

$$\boxed{\begin{aligned} \mathscr{L}_I^k(u,\mu)y^{k+1} &= \mathscr{L}_E^k(u,\mu)y^k + \Delta t z^k(u,\mu) \quad \text{for } k = 0,...,K-1 \\ y^0 &= \eta_0 \end{aligned}} \tag{4.8}$$

Here, $\eta_0 \in \mathbb{R}^{N_x}$ is the discretized initial condition, meaning $\eta_{0,i} = y_0(\mathbf{x}_i)$ for $i = 1,...,N_x$. The implicit and explicit operators are given by

$$\begin{aligned} \mathscr{L}_I^k(u,\mu) &= \left[1 - \Delta t\vartheta\left(\kappa^{k+1}(u,\mu) + \frac{1}{\Delta x}\right)\right] B_{N_x} + \vartheta\frac{\Delta t}{\Delta x} \\ \mathscr{L}_E^k(u,\mu) &= \left[1 + \Delta t(1-\vartheta)\left(\kappa^{k}(u,\mu) + \frac{1}{\Delta x}\right)\right] B_{N_x} - (1-\vartheta)\frac{\Delta t}{\Delta x} \end{aligned}$$

and the inhomogeneities look like this:

$$z^k(u,\mu) = \left(\frac{1}{\Delta x} + (1-\vartheta)\kappa^k(u,\mu) + \vartheta\kappa^{k+1}(u,\mu)\right) g e_1$$

We can finally give some thought to the space in which (4.8) is to be understood. Of course, the vectors all come from $\mathbb{R}^{N_x}$ yet it is not obvious what the inner product should look like. First of all, a vector $y = (y_1,...,y_{N_x}) \in \mathbb{R}^{N_x}$ can be understood as a discretization of a function $\widetilde{y} : [\underline{\mathbf{x}}, \overline{\mathbf{x}}] \to \mathbb{R}$. If we postulate $\widetilde{y} \in L^2(\underline{\mathbf{x}}, \overline{\mathbf{x}})$, then the norm $||y||$ should somehow approximate the norm $||\widetilde{y}||_{L^2(\underline{\mathbf{x}},\overline{\mathbf{x}})}$. Seeing as y only contains information at the grid points $\mathbf{x}_i$ for $i = 1,...,N_x$, the natural way to approximate the integral is using the trapezoidal rule:

$$||\widetilde{y}||^2_{L^2(\underline{\mathbf{x}},\overline{\mathbf{x}})} = \int_{\underline{\mathbf{x}}}^{\overline{\mathbf{x}}} |\widetilde{y}|^2 d\mathbf{x} \approx \frac{\Delta x}{2}y_1^2 + \Delta x \sum_{i=2}^{N_x-1} y_i^2 + \frac{\Delta x}{2}y_{N_x}^2$$

Introducing the symmetric, positive definite weight matrix

$$W := \text{diag}\left(\frac{\Delta x}{2}, \Delta x, ..., \Delta x, \frac{\Delta x}{2}\right) \in \mathbb{R}^{N_x \times N_x}$$

now allows us to formulate an inner product on $\mathbb{R}^{N_x}$ which can be written as $(y,z)_W := y^T W z$.

5

The RB-Method

The results of the discretization methods from Chapter 4 all share common characteristics: Working from a finite-dimensional Hilbert space X_N with inner product $(\cdot,\cdot)$, a sequence $\{y_N^k(u,\mu)\}_{k=0}^K \subset X_N$ is created where $y_N^k(u,\mu)$ represents in some way the discrete solution at the time t_k. Notice that in this chapter, the (u,μ)-dependency of the solution will always be expressed by writing $y_N^k(u,\mu)$. This series will from now on be referred to as the **detailed solution**. As was previously demonstrated, it allows for a recursive representation that can be expressed as

$$\begin{aligned} \mathscr{L}_I^k(u,\mu)y_N^{k+1}(u,\mu) &= \mathscr{L}_E^k(u,\mu)y_N^k(u,\mu) + \Delta t z^k(u,\mu), \quad k=0,...,K-1 \\ y_N^0(u,\mu) &= \eta^0 \end{aligned} \tag{5.1}$$

If N and K are large numbers, computing the detailed solution for different parameters $(u,\mu) \in U_{ad} \times \mathscr{D}$ can be very expensive. It is for such reasons that RB methods have been established that all share the same basic idea: A low-dimensional subspace $X_H \subset X_N$ of dimension $H \ll N$ called the *RB-space* is created after which the recursion (5.1) is projected from X_N to X_H, resulting in a so-called **reduced solution** $\{y_H^k(u,\mu)\}_{k=0}^K \subset X_H$. Because of $H \ll N$, these computations will be much less expensive. The problem that now has to be considered is the generation of this suitable space X_H, which has to fulfill two general requirements:

1. The loss of information between the detailed and the reduced solution has to be acceptable.

2. The time invested in the generation of the RB-space should be much less than what is saved through its utility.

This chapter focusses on two questions: First, for a given RB-space $X_H \subset X_N$, how exactly is the N-dimensional system (5.1) projected onto X_H? And second, how can a suitable RB-space be generated in the first place?
Throughout the following pages, we follow the approach laid out by Haasdonk and Ohlberger in [10], which adopts what they call a *parametrized evolution scheme* based primarily on a representation for the detailed solution of the type (5.1).

5.1 Model Order Reduction

In this section we assume that a RB-space $X_H \subset X_N$ has been generated. We can further assume that an orthonormal basis $\Psi_H = \{\psi_1, ..., \psi_H\}$ is available which spans X_H. Given this situation, we have to establish how to project the system (5.1) onto X_H as well as how the arising error can be estimated without having to compute the detailed solution. Lastly, it would be desirable to establish efficient computation strategies of both the reduced solution and the error estimator.

5.1.1 A Galerkin Ansatz

Since X_N is a Hilbert space with inner product $(\cdot, \cdot)$, projecting (5.1) onto X_H is a very straightforward Galerkin ansatz: The reduced solution $\{y_H^k(u,\mu)\}_{k=0}^K \subset X_H$ has to fulfill the following recursion:

$$\begin{aligned} &\Big(\mathscr{L}_I^k(u,\mu)y_H^{k+1}(u,\mu), \psi_h\Big) = \\ &\Big(\mathscr{L}_E^k(u,\mu)y_H^k(u,\mu) + \Delta t z^k(u,\mu), \psi_h\Big) \qquad (k = 0, ..., K, h = 1, ...H) \\ &\qquad \big(y_H^0(u,\mu), \psi_h\big) = \big(\eta^0, \psi_h\big) \qquad (h = 1, ..., H) \end{aligned}$$

Furthermore, $y_H^k(u,\mu)$ can be expressed by the basis Ψ_H, for example

$$y_H^k(u,\mu) = \sum_{h=1}^{H} a_h^k(u,\mu)\psi_h, \quad k = 0, ..., K-1, (u,\mu) \in U_{ad} \times \mathscr{D}$$

with the coordinate trajectory $\{a^k(u,\mu)\}_{k=0}^K \subset \mathbb{R}^H$. Inserting this into the system aboves yields the following reduced system in $\mathbb{R}^H$:

$$\boxed{\begin{aligned} L_I^k(u,\mu)a^{k+1}(u,\mu) &= L_E^k(u,\mu)a^k(u,\mu) + \Delta t b^k(u,\mu), \quad k = 0, ..., K-1 \\ a^0(u,\mu) &= \zeta^0 \end{aligned}} \tag{5.2}$$

where the abbreviations are as follows:

$$\begin{aligned}\left[L^k_{I/E}(u,\mu)\right]_{h\ell} &= \left(\mathscr{L}^k_{I/E}(u,\mu)\psi_\ell,\psi_h\right), && h,\ell = 1,...,H\\ b^k_h(u,\mu) &= \left(z^k(u,\mu),\psi_h\right), && h = 1,...,H\\ \zeta^0_h &= (\eta^0,\psi_h), && h = 1,...,H\end{aligned}$$

5.1.2 Error estimate

After having computed a reduced solution $\{y^k_H(u,\mu)\}^K_{k=0} \subset X_H$ whose RB representation is given by (5.2), the next objective is of course to assess the error to the detailed solution $\{y^k_N(u,\mu)\}^K_{k=0} \subset X_N$ given by (5.1). For efficiency purposes, this generally has to be done without actually computing the detailed solution. After all, the very reason why a RB method is used is in order to mostly prevent these costly computations.
Following again Haasdonk and Ohlberger in [10], we define for $k = 0,...,K$ and a fixed pair $(u,\mu) \in U_{ad} \times \mathscr{D}$ the error difference between the two solutions as $e^k_H(u,\mu) := y^k_H(u,\mu) - y^k_N(u,\mu) \in X_N$. Our objective is to obtain an error estimator $\Delta^k_H(u,\mu) > 0$ which is computable without the detailed solution and is a rigorous error bound, meaning that we have $\left|\left|e^k_H(u,\mu)\right|\right|_{X_N} \leq \Delta^k_H(u,\mu)$ for all $k = 0,...,K$.

Theorem 5.1
Let $(u,\mu) \in U_{ad} \times \mathscr{D}$ be a given control-parameter pair. Furthermore, let the operators in (5.1) be uniformly bounded in time, meaning that there are constants $C_I(u,\mu), C_E(u,\mu) > 0$ such that

$$\left|\left|\left(\mathscr{L}^k_I(u,\mu)\right)^{-1}\right|\right|_{L(X_N)} \leq C_I(u,\mu),\ \left|\left|\mathscr{L}^k_E(u,\mu)\right|\right|_{L(X_N)} \leq C_E(u,\mu) \quad (k = 0,...,K)$$

We define the residual $R^k_H(u,\mu) \in X_N$ for $k = 0,...,K-1$ by

$$R^{k+1}_H(u,\mu) := \frac{\mathscr{L}^k_I(u,\mu)y^{k+1}_H(u,\mu) - \mathscr{L}^k_E(u,\mu)y^k_H(u,\mu)}{\Delta t} - z^k(u,\mu)$$

If the RB space satisfies the inital condition $\eta^0 \in X_H$, then a rigorous error estimate as mentioned above is given by the following *canonical estimator*

$$\boxed{\Delta^k_H(u,\mu) = \Delta t \sum_{j=0}^{k-1} C_E(u,\mu)^{k-1-j} C_I^{k-j}(u,\mu) \left|\left|R^{j+1}_H(u,\mu)\right|\right|_{X_N}} \tag{5.3}$$

Proof. See Proposition 4.2 in [10]. □

Two things have to mentioned here. First of all, the error estimator $\Delta_H^k(u,\mu)$ depends on the operator bounds $C_I(u,\mu)$ and $C_E(u,\mu)$ as well as on the residual norms $||R_H^j(u,\mu)||$. The latter depend only on the reduced solution $\{y_H^k(u,\mu)\}_{k=0}^K$, so the computation of the detailed solution is in fact not necessary. Second, in the case $C_E(u,\mu), C_I(u,\mu) \leq 1$, the error estimator can be altered to the *simplified error estimator*

$$\widetilde{\Delta}_H^k(u,\mu) = \Delta t \sum_{j=0}^{k-1} \left|\left| R_H^{j+1}(u,\mu) \right|\right|_{X_N} \tag{5.4}$$

In Section 6.1.4, there is a brief overview concerning the occurring operator norms. However, we will already mention here that the general case will be $C_E = 1$ and $C_I > 1$ but $C_I \approx 1$. This will still allow for the uage of the estimator in (5.4). It has to be stated, however, that we cannot ensure that $\left|\left| e_H^k(u,\mu) \right|\right|_{X_N} \leq \widetilde{\Delta}_H^k(u,\mu)$ really holds true in all cases. In order to do so, we will have to use the estimator $\Delta_H^k(u,\mu)$ given by (5.3) with fitting values for C_E, C_I, although this may well lead to the estimates getting out of hand, especially for finer time grids because of the exponential dependence on k. A comparison between the Δ_H^k and $\widetilde{\Delta}_H^k$ is presented in Section 6.2.1.

5.1.3 Affine Parameter Dependency

By observing all possible shapes of the operators $\mathscr{L}_{I/E}^k(u,\mu) \in L(X_N)$ along with the vectors $z^k(u,\mu) \in X_N$, we observe that the variables k, u and μ only occur in scalar corollary functions. This means that there are time-, control- and parameter-independent operators respectively vectors such that $\mathscr{L}_{I/E}^k(u,\mu)$ and $z^k(u,\mu)$ are always contained in their linear span. In mathematical notation, we have operators $\mathscr{P}_{I/E}^1, ..., \mathscr{P}_{I/E}^{Q_{I/E}} \in L(X_N)$ and vectors $\varrho^1, ..., \varrho^{Q_z} \in X_N$ such that for all $k = 0, ..., K-1$, $u \in U_{ad}$ and $\mu \in \mathscr{D}$, we get

$$\mathscr{L}_{I/E}^k(u,\mu) = \sum_{q=1}^{Q_{I/E}} \sigma_{I/E}^q(t^k,u,\mu)\mathscr{P}_{I/E}^q, \qquad z^k(u,\mu) = \sum_{q=1}^{Q_z} \sigma_z^q(t^k,u,\mu)\varrho^q \tag{5.5}$$

We have made use of coefficient functions $\sigma_{I/E/z}^q : [0,T] \times U_{ad} \times \mathscr{D} \to \mathbb{R}$ which contain all time-, control- and parameter-dependencies.

Efficient computation of the reduced solution

Using the knowledge of these shapes renders a similar parameter-dependence for the matrices and vectors which were introduced in Section 5.1.1 for the computation of the reduced solution:

$$L^k_{I/E}(u,\mu) = \sum_{q=1}^{Q_{I/E}} \sigma^q_{I/E}(t^k,u,\mu) P^q_{I/E}, \qquad b^k_h(u,\mu) = \sum_{q=1}^{Q_z} \sigma^q_z(t^k,u,\mu) p^q$$

where we have set

$$\left[P^q_{I/E}\right]_{h\ell} = \left(\mathscr{P}^q_{I/E}\psi_\ell, \psi_h\right), \quad p^q_h = (\varrho^q, \psi_h) \qquad (h,\ell = 1,...,H)$$

For implementation purposes, this can be further simplified. First of all, we can assume without loss of generality that the Hilbert space is given by $\mathbb{R}^{N_x}$ with $N_x \in \mathbb{N}_0$ and that the inner product is given by $(\varphi, \widetilde{\varphi}) = \varphi^T W \widetilde{\varphi}$ where $W \in \mathbb{R}^{N_x \times N_x}$ is a symmetric, positive definite matrix. For the FD-method from Section 4.2, this is already the case with $W = \Delta x \ \mathrm{diag}(1,2,...,2,1)/2$. In case of the polynomial method introduced in Section 4.1, we can work with the basis representation formulation (4.4) which uses the space $\mathbb{R}^{N_x}$ with $N_x := N+1$ and the matrix $W = \Gamma^2$, as was shown in Lemma 4.2a).
We define the RB matrix $\Psi_H = [\psi_1, ..., \psi_H] \in \mathbb{R}^{N_x \times H}$ and can conclude

$$P^q_{I/E} = \Psi^T W \mathscr{P}^q_{I/E} \Psi \qquad p^q = \Psi^T W \varrho^q$$

Efficient computation of the error estimator

In order to make feasible use of the estimator Δ_H and $\widetilde{\Delta}_H$ from (5.3) respectively (5.4), the residual norms $||R^j_H(u,\mu)||$ have to be computed in an efficient way. Following again Haasdonk and Ohlberger in [10], we can transform the norm utilizing

the inner product $(\cdot,\cdot)$ in X_N:

$$\begin{aligned}\left|\left|R_H^{k+1}(u,\mu)\right|\right|_{X_N}^2 &= \left(R_H^{k+1}(u,\mu), R_H^{k+1}(u,\mu)\right)\\ &= \frac{1}{\Delta t^2}\left(\mathscr{L}_I^k(u,\mu)y_H^{k+1}(u,\mu), \mathscr{L}_I^k(u,\mu)y_H^{k+1}(u,\mu)\right)\\ &+ \frac{1}{\Delta t^2}\left(\mathscr{L}_E^k(u,\mu)y_H^{k}(u,\mu), \mathscr{L}_E^k(u,\mu)y_H^{k}(u,\mu)\right)\\ &- \frac{2}{\Delta t^2}\left(\mathscr{L}_I^k(u,\mu)y_H^{k+1}(u,\mu), \mathscr{L}_E^k y_H^k\right)\\ &- \frac{1}{\Delta t}\left(\mathscr{L}_I^k(u,\mu)y_H^{k+1}(u,\mu), z^k(u,\mu)\right)\\ &+ \frac{1}{\Delta t}\left(\mathscr{L}_E^k(u,\mu)y_H^{k}(u,\mu), z^k(u,\mu)\right) + \left(z^k(u,\mu), z^k(u,\mu)\right)\end{aligned} \tag{5.6}$$

As it was already done in the section above for the computation of the reduced solution, we are now interested in containing the parameter-control-dependency in scalar coefficient functions. Using the decompositions in (5.5), we observe for the first term above:

$$\begin{aligned}&\left(\mathscr{L}_I^k(u,\mu)y_H^{k+1}(u,\mu), \mathscr{L}_I^k(u,\mu)y_H^{k+1}(u,\mu)\right)\\ =& \sum_{q,q'=1}^{Q_I} \sigma_I^q(t^k,u,\mu)\sigma_I^{q'}(t^k,u,\mu)\left(\mathscr{P}_I^q y_H^{k+1}(u,\mu), \mathscr{P}_I^{q'} y_H^{k+1}(u,\mu)\right)\\ =& \sum_{q,q'=1}^{Q_I} \sigma_I^q(t^k,u,\mu)\sigma_I^{q'}(t^k,u,\mu)\sum_{h,\ell=1}^{H} a_h^{k+1}(u,\mu)a_\ell^{k+1}(u,\mu)\left(\mathscr{P}_I^q\psi_h, \mathscr{P}_I^{q'}\psi_\ell\right)\end{aligned}$$

For every pair $(q,q') \in \{1,...,Q_I\}^2$, we now define the matrix

$$K_{II}^{qq'} \in \mathbb{R}^{H\times H}, \quad \left[K_{II}^{qq'}\right]_{h\ell} = \left(\mathscr{P}_I^q\psi_h, \mathscr{P}_I^{q'}\psi_\ell\right)$$

so that the equality above becomes

$$\begin{aligned}&\left(\mathscr{L}_I^k(u,\mu)y_H^{k+1}(u,\mu), \mathscr{L}_I^k(u,\mu)y_H^{k+1}(u,\mu)\right)\\ =& \sum_{q,q'=1}^{Q_I} \sigma_I^q(t^k,u,\mu)\sigma_I^{q'}(t^k,u,\mu)\left[a^{k+1}(u,\mu)\right]^T K_{II}^{qq'}\left[a^{k+1}(u,\mu)\right]\end{aligned}$$

In the same way, the following matrices are defined:

$$\begin{aligned}
&K_{EE}^{qq'} \in \mathbb{R}^{H\times H}, && \left[K_{EE}^{qq'}\right]_{h\ell} = \left(\mathscr{P}_E^q\psi_h, \mathscr{P}_E^{q'}\psi_\ell\right), && q,q' = 1,...,Q_E\\
&K_{IE}^{qq'} \in \mathbb{R}^{H\times H}, && \left[K_{IE}^{qq'}\right]_{h\ell} = \left(\mathscr{P}_I^q\psi_h, \mathscr{P}_E^{q'}\psi_\ell\right), && q = 1,...,Q_I, q' = 1,...,Q_E\\
&K_{I}^{qq'} \in \mathbb{R}^{H}, && \left[K_{I}^{qq'}\right]_{h} = \left(\mathscr{P}_I^q\psi_h, \varrho^{q'}\right), && q = 1,...,Q_I, q' = 1,...,Q_E\\
&K_{E}^{qq'} \in \mathbb{R}^{H}, && \left[K_{E}^{qq'}\right]_{h} = \left(\mathscr{P}_E^q\psi_h, \varrho^{q'}\right), && q = 1,...,Q_E, q' = 1,...,Q_z\\
&K_{z}^{qq'} \in \mathbb{R}, && K_z^{qq'} = \left(\varrho^q, \varrho^{q'}\right), && q,q' = 1,...,Q_z
\end{aligned}$$

Let it again be noted that in case $X_N = \mathbb{R}^{N_x}$ and a weighted scalar product $(\varphi,\widetilde{\varphi}) = \varphi^T W\widetilde{\varphi}$, these matrices and vectors take the forms

$$K_{XY}^{qq'} = \Psi^T\left[\mathscr{P}_X^q\right]^T W\mathscr{P}_Y^{q'}\Psi, \quad K_X^{qq'} = \Psi^T\left[\mathscr{P}_I^q\right]^T W\varrho^{q'}, \quad K_z^{qq'} = [\varrho^q]^T W\varrho^{q'}$$

where $\Psi = [\psi_1,...,\psi_H] \in \mathbb{R}^{N_x\times H}$ is a matrix containing all RB vectors and $X,Y \in \{I,E\}$. Finally, the residual norm looks like this:

$$\begin{aligned}
\left|\left|R_H^{k+1}(u,\mu)\right|\right|_{X_N}^2 &= \frac{1}{\Delta t^2}\sum_{q,q'=1}^{Q_I}\sigma_I^q(t^k,u,\mu)\sigma_I^{q'}(t^k,u,\mu)\left[a^{k+1}(u,\mu)^T K_{II}^{qq'}a^{k+1}(u,\mu)\right]\\
&+ \frac{1}{\Delta t^2}\sum_{q,q'=1}^{Q_E}\sigma_E^q(t^k,u,\mu)\sigma_E^{q'}(t^k,u,\mu)\left[a^k(u,\mu)^T K_{EE}^{qq'}a^k(u,\mu)\right]\\
&- \frac{2}{\Delta t^2}\sum_{q=1}^{Q_I}\sum_{q'=1}^{Q_E}\sigma_I^q(t^k,u,\mu)\sigma_E^{q'}(t^k,u,\mu)\left[a^{k+1}(u,\mu)^T K_{IE}^{qq'}a^k(u,\mu)\right]\\
&- \frac{1}{\Delta t}\sum_{q=1}^{Q_I}\sum_{q'=1}^{Q_z}\sigma_I^q(t^k,u,\mu)\sigma_z^{q'}(t^k,u,\mu)\left[a^{k+1}(u,\mu)^T K_I^{qq'}\right]\\
&+ \frac{1}{\Delta t}\sum_{q=1}^{Q_E}\sum_{q'=1}^{Q_z}\sigma_E^q(t^k,u,\mu)\sigma_z^{q'}(t^k,u,\mu)\left[a^k(u,\mu)^T K_E^{qq'}\right]\\
&+ \sum_{q,q'=1}^{Q_z}\sigma_z^q(t^k,u,\mu)\sigma_z^{q'}(t^k,u,\mu)\left[K_z^{qq'}\right]
\end{aligned}$$

Now all pieces are assembled to perform the RB method and compute the error estimator $\widetilde{\Delta}_H^k$ and - if we know the boundary values C_I and C_E - also Δ_H^k. One only needs to know the exact values of $\sigma_{I/E/z}^q$, $\mathscr{P}_{I/E}^q$ and ϱ^q. Since these are fundamentally different for each discretization technique which has been introduced in chapter 4, we will look at each method individually and identify those variables:

The polynomial ϑ-method

For this method, we will work with the basis representations which were introduced in Section 4.1.3:

$$\begin{aligned}
\widehat{\mathscr{L}}_I^k(u,\mu) &= 1 - \vartheta\Delta t A_N^{k+1}(u,\mu) \\
&= \left(1 - \vartheta\Delta t\kappa(t^{k+1},u,\mu)\right)\mathbb{1}_N + \vartheta\Delta t\left(B_N + \Gamma_N\right) \\
\widehat{\mathscr{L}}_E^k(u,\mu) &= 1 + (1-\vartheta)\Delta t A_N^k(u,\mu) \\
&= \left(1 + (1-\vartheta)\Delta t\kappa(t^k,u,\mu)\right)\mathbb{1}_N - (1-\vartheta)\Delta t\left(B_N + \Gamma_N\right) \\
\widehat{z}_N^k(u,\mu) &= g\gamma_N
\end{aligned}$$

Inserting this into the definitions above yields $Q_I = 2$, $Q_E = 2$ and $Q_z = 1$. Furthermore, $\mathscr{P}_{I/E}^1 = \mathbb{1}_N$, $\mathscr{P}_{I/E}^2 = B_N + \Gamma_N$ and $\varrho^1 = g\gamma_N$. The scalar coefficient functions read

$$\begin{aligned}
\sigma_I^1(t^k,u,\mu) &= 1 - \vartheta\Delta t\kappa(t^{k+1},u,\mu) & \sigma_I^2(t^k,u,\mu) &= \vartheta\Delta t \\
\sigma_E^1(t^k,u,\mu) &= 1 + (1-\vartheta)\Delta t\kappa(t^k,u,\mu), & \sigma_E^2(t^k,u,\mu) &= -(1-\vartheta)\Delta t \\
\sigma_z^1(t^k,u,\mu) &= 1
\end{aligned}$$

The polynomial RK4 method

As it was done for the polynomial ϑ-method above, we will work with the basis representations which were identified in Section 4.1.3. For simplicity reasons, the dependency on (u,μ) will be omitted in the notation.

$$\begin{aligned}
\widehat{\mathscr{L}}_I^k &= 1 \\
\widehat{\mathscr{L}}_E^k &= 1 + \frac{\Delta t}{6}\left(A_N^k + 4A_N^{k+\frac{1}{2}} + A_N^{k+1}\right) + \frac{\Delta t^2}{6}\left(A_N^{k+\frac{1}{2}}A_N^k + A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}} + A_N^{k+1}A_N^{k+\frac{1}{2}}\right) \\
&\quad + \frac{\Delta t^3}{12}\left(A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}}A_N^k + A_N^{k+1}A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}}\right) + \frac{\Delta t^4}{24}A_N^{k+1}A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}}A_N^k \\
\widehat{z}_N^k &= \Big[1 + \frac{\Delta t}{6}\left(2A_N^{k+\frac{1}{2}} + A_N^{k+1}\right) + \frac{\Delta t^2}{6}\left(A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}} + A_N^{k+1}A_N^{k+\frac{1}{2}}\right) \\
&\quad + \frac{\Delta t^3}{12}A_N^{k+1}A_N^{k+\frac{1}{2}}A_N^{k+\frac{1}{2}}\Big]g\gamma_N
\end{aligned}$$

It has been shown that we can write $A_N(t,u,\mu) = \kappa(t,u,\mu) - B_N - \Gamma_N$. Inserting this into the representations above yields after lengthy, straightforward

computations:

$$\begin{aligned}
\widehat{\mathscr{L}}_I^k &= 1 \\
\widehat{\mathscr{L}}_E^k &= \sigma_E^1(t^k) + \sigma_E^2(t^k)(B_N + \Gamma_N) + ... + \sigma_E^5(t^k)(B_N + \Gamma_N)^4 \\
\widehat{z}_N^k &= \sigma_z^1(t^k) g\gamma_N + \sigma_z^2(t^k)(B_N + \Gamma_N) g\gamma_N + ... \sigma_z^4(t^k)(B_N + \Gamma_N)^3 g\gamma_N
\end{aligned}$$

The occuring coefficient functions are quite complicated and can be found in the appendix, Section 8.1. To summarize, we get $Q_I = 1$, $Q_E = 5$ and $Q_z = 4$. The constant operators take the shape

$$\begin{aligned}
\mathscr{P}_I^1 &= \mathbb{1}_N \\
\mathscr{P}_E^q &= (B_N + \Gamma_N)^{q-1} && \text{for } q = 1, .., 5 \\
\varrho^q &= (B_N + \Gamma_N)^{q-1} g\gamma_N && \text{for } q = 1, ..., 4
\end{aligned}$$

The FD method

The representations for the FD method have already been established in Section 4.2, we have $Q_I = 2$, $Q_E = 2$ and $Q_z = 1$. The operators take the shape $\mathscr{P}_{I/E}^1 = \mathbb{1}_N$, $\mathscr{P}_{I/E}^2 = B_N$ as well as $\rho^1 = ge_1$. Last of all, the coefficient functions look like this:

$$\begin{aligned}
\sigma_I^1(t^k, u, \mu) &= \vartheta \frac{\Delta t}{\Delta x} \\
\sigma_I^2(t^k, u, \mu) &= 1 - \Delta t \vartheta \left(\kappa(t^{k+1}, u, \mu) + \frac{1}{\Delta x} \right) \sigma_E^1(t^k, u, \mu) \quad = -(1 - \vartheta) \frac{\Delta t}{\Delta x} \\
\sigma_E^2(t^k, u, \mu) &= 1 + \Delta t (1 - \vartheta) \left(\kappa(t^{k+1}, u, \mu) + \frac{1}{\Delta x} \right)
\end{aligned}$$

5.2 Basis generation

Now that it has been established how a RB method is being performed and efficiently computed if a RB space is already known, the question remains how to generate said space. We will accomplish this by using a so-called Greedy algorithm. The idea is to start with an initial basis $\Psi_{H_0} = \{\psi_1, ..., \psi_{H_0}\} \subset X_N$ of length H_0. After this, the basis is iteratively enhanced by adding one or several new basis vectors in each iteration. The process is repeated until a maximal number of basis functions is reached or a given error tolerance is satisfied.

In each iteration, the quality of the current reduced solutions (which is governed

by the current reduced basis) has to be assessed with respect to the entire control-parameter domain $U_{ad} \times \mathcal{D}$. As is usually the case for Greedy algorithms, this is achieved by choosing a select and rather small subset $U_{train} \times \mathcal{D}_{train} \subset U_{ad} \times \mathcal{D}$, followed by computing the reduced solution along with error estimators for all these control-parameter pairs. The chosen elements are referred to as training parameters and they also play a major role in the generation of the next basis vectors: Among all possible combinations $(u, \mu) \in U_{train} \times \mathcal{D}_{train}$, the algorithm looks for the 'worst' pair (u^*, μ^*) which is the one where there is the highest estimated discrepancy between the detailed and the reduced solution. This is where the algorithm gets its name since it greedily picks the weakest member of the herd. The detailed solution to this control-parameter tuple is then computed and used to generate the next basis elements with the intention of improving the basis quality in this area of the control-parameter domain.

Algorithm 1: Greedy Search for the generation of an RB space

1: Compute an initial basis Ψ_{H_0} of length H_0.
2: Set $H := H_0$ as well as $\Psi_H := \Psi_{H_0}$.
3: Compute all offline values for Ψ_H.
4: **for all** $(u, \mu) \in U_{train} \times \mathcal{D}_{train}$ **do**
5: Compute all online values for Ψ_H at (u, μ).
6: Compute the reduced solution $\{y_H^k(u, \mu)\}_{k=0}^K$.
7: Compute the error estimator $\Delta_H^k(u, \mu)$ for $k = 0, ..., K$.
8: **end for**
9: Choose a pair $(u^*, \mu^*) \in U_{train} \times \mathcal{D}_{train}$ which 'maximizes' the error estimator trajectory $\{\Delta_H^k(u, \mu)\}_{k=0}^K$.
10: Compute the detailed solution $\{y_N^k(u^*, \mu^*)\}_{k=0}^K$ for this pair.
11: Use the the detailed solution above to generate new basis elements $\psi_{H+1}, ..., \psi_{H+\ell} \in X_N$.
12: Set $\widetilde{\Psi}_{H+\ell} := \Psi \cup \{\psi_{H+1}, ..., \psi_{H+\ell}\}$ and obtain the new basis by using the Gram-Schmidt process on $\widetilde{\Psi}_{H+\ell}$ to get an orthonormal system $\Psi_{H+\ell}$.
13: Set $H := H + \ell$ and repeat the steps 3-13 until a termination condition is satisfied.

The most basic outline of the Greedy search is depicted in Algorithm 1. We have purposefully used rather vague formulations in lines 9 and 11, which will be examined more thoroughly in the following sections. Section 5.2.1 will focus on line 9 in more detail, thereby explaining how exactly the 'worst' control-parameter pair can be identified. Section 5.2.2 will present various methods for the realization of line 11 by introducing strategies that can be alternatively utilized, depending on how exactly the detailed solution is used to generate new basis vectors. Furthermore, we will elaborate on numerical alternatives of Algorithm 1, which are used to rectify several shortcomings that have occurred during our simulations.

5.2.1 Line 9: The worst control-parameter pair

Suppose we are presented with either the canonical error estimator Δ_H^k or the simplified error estimator $\widetilde{\Delta}_H^k(u,\mu)$ given by (5.3) respectively (5.4). The question is how to identify a tuple $(u^*,\mu^*) \in U_{train} \times \mathcal{D}_{train}$ with the 'worst error' between detailed and reduced solution. Following [10], this is done by choosing $(u^*,\mu^*,k^*) \in U_{train} \times \mathcal{D}_{train} \times \{0,...,K\}$ with

$$\Delta_H^{k^*}(u^*,\mu^*) \;=\; \max_{(u,\mu)\in U_{train}\times\mathcal{D}_{train}} \max_{k=0,...,K} \Delta_H^k(u,\mu)$$

Furthermore, in our cases we are always presented with an estimator which is monotonically increasing over time, therefore the above discrete optimization problem can be simplified by setting $k^* = K$ and only having to maximize over $U_{train} \times \mathcal{D}_{train}$.

5.2.2 Line 11: Enhancement strategies

In line 11, we are given a detailed solution $\{y_N^k(u^*,\mu^*)\}_{k=0}^K \subset X_N$ and have to use it in order to create new basis vectors $\psi_{H+1},...,\psi_{H+\ell}$ which better represent this trajectory than the existing basis Ψ_H. There are two different strategies available to us here, each of which will be examined in the experiments.

The Single-Time strategy (ST)

This strategy focuses on the so-called 'worst-error snapshot', meaning that it looks for a single time index $k^* \in \{0,...,K\}$ so that the corresponding snapshot $y_N^{k^*}(u^*,\mu^*)$ is the one which is worst incorporated in the current basis Ψ_H. This vector is then simply added to the basis by defining $\psi_{H+1} := y_N^{k^*}(u^*,\mu^*)$. We follow the approach used in [8] as well as [10]:
We have to consider that all discretization techniques presented in chapter 4 are single-step strategies in time. This means that the error $e_H^k(u^*,\mu^*)$ at a given time instant k consists both of the single-step error and the propagated error. The single-step error stems from computing $y_H^k(u^*,\mu^*)$ from $y_H^{k-1}(u^*,\mu^*)$ whereas the propagated error is the sum of errors resulting from the computation of $y_H^{k-1}(u^*,\mu^*)$ from $y_H^0(u^*,\mu^*)$. On the one hand, this means that the error terms $e_H^k(u^*,\mu^*)$ will generally increase over time, as is the case for the error estimators $\Delta_H^k(u^*,\mu^*)$ and $\widetilde{\Delta}_H^k$. On the other hand, we are limited to a single snapshot in this strategy which means that we have to neglect the propagated error and focus on the single-step error. The latter is best assessed by considering the increments $\Delta_H^k(u^*,\mu^*) - \Delta_H^{k-1}(u^*,\mu^*)$. A large increment from $k-1$ to k is taken as evidence

that the snapshot $y_N^k(u^*, \mu^*)$ is badly represented by the basis Ψ_H. Turning to the strategy, this means that

$$k^* := \arg\max_{k=1,\dots,K} \left(\Delta_H^k(u^*, \mu^*) - \Delta_H^{k-1}(u^*, \mu^*) \right)$$

will be the time index for the worst-error snapshot.

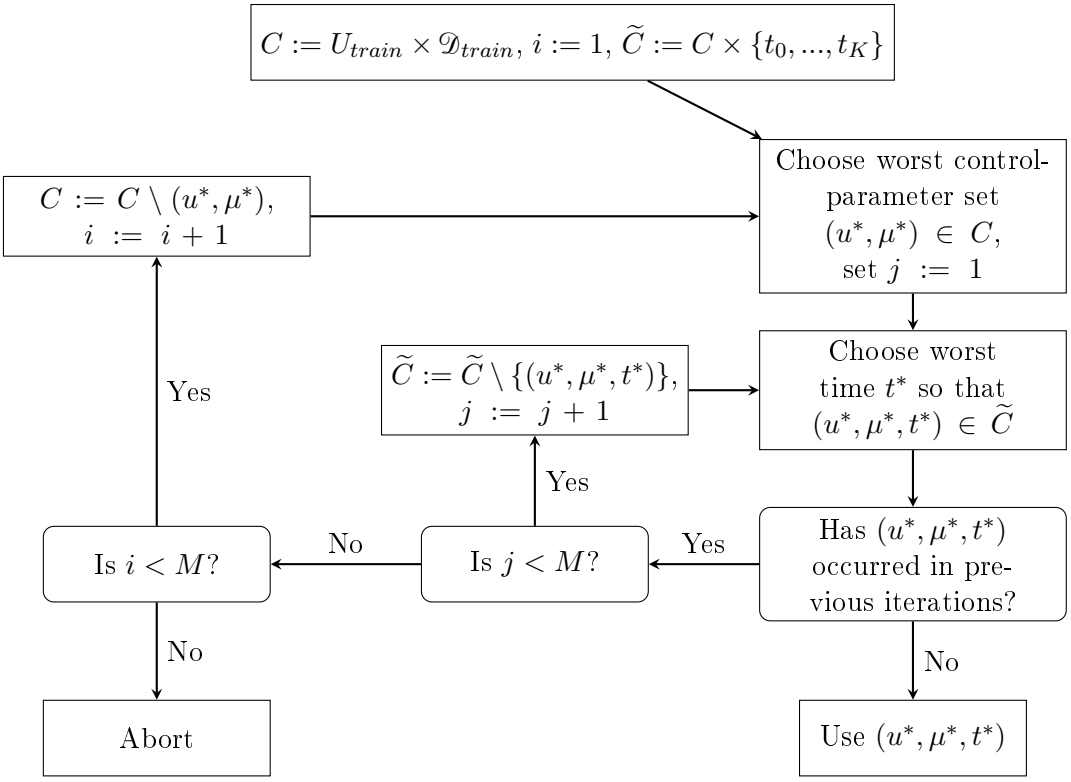
However, it can happen that a set (u^*, μ^*, k^*) ist chosen which has already occurred in previous iterations. This means that there is no new information added to the basis and can in some cases lead to the stagnation of the algorithm. A simple way to add some flexibility is to monitor these reocurrences and allow the Greedy search to fall back on the second or third worst parameter set in these cases, thereby ensuring that only new vectors are added to the basis.
In a more general formulation, the algorithm will look for the M worst control-parameter sets ($M \in \mathbb{N}$) respectively the M worst snapshots and use the worst one available that has not been used yet. If all of the M worst parameter sets have been used before, this will lead to an abort. The implementation for this variation ist depicted in detail in Algorithm 5.1. Whenever the flex$M$ variation is used, we shall refer to the strategy not only as ST, but as ST-flex$M$.

The POD Strategy (PODq)

In contrast to the ST-strategy above, whose main characteristic is that an entire solution trajectory $\{y_N^k(u^*, \mu^*)\}_{k=0}^K \subset X_N$ is computed and only a single snapshot $y_N^{k^*}(u^*, \mu^*)$ is used to generate the next basis element, the focus for the POD strategy is to compress the data contained in this trajectory into a few vectors as efficiently as possible. Therefore, it is first taken account for the fact that a part of this trajectory is already contained in the current reduced space X_H: Let $P_1 : X_N \to X_H$ be the orthogonal projection onto X_H and $P_2 : X_N \to X_H^\perp$ the projection onto its orthogonal space. Then every $z \in X_N$ allows for the decomposition $z = P_1 z + P_2 z$, see for example Theorem V.3.4 in [22]. This means that the orthogonal projection P_2 can be obtained by computing

$$P_2 z = z - P_1 z = z - \sum_{h=1}^{H} (z, \psi_h)\psi_h$$

If – like in our case – X_N is given by $\mathbb{R}^{N_x}$ and the scalar product can be expressed by the positive definite weight matrix $W \in \mathbb{R}^{N_x \times N_x}$, then this can be further simplified to $P_2 z = (1 - \Psi_H \Psi_H^T W)z$. As it was done before, the reduced basis has been assembled in a matrix $\Psi_H \in \mathbb{R}^{N_x \times H}$. We also summarize the solution in a



Algorithm 5.1: The flex$M$ variation for the ST strategy

similar way, meaning that we define

$$Y := \left[y_N^0(u^*,\mu^*),\ldots,y_N^K(u^*,\mu^*)\right] \in \mathbb{R}^{N_x\times(K+1)}$$

This allows for a computation of the orthogonal component matrix by

$$Y_\perp := \left[y_\perp^0,...,y_\perp^K\right] := (1-\Psi\Psi^T W)Y$$

The columns of $Y^\perp$, which are exactly the projections of the snapshots onto the orthogonal space $X_H^\perp$, now represent all the information contained in the solution which is right now not represented by the basis Ψ_H. The problem that poses itself is to find some - let us say ℓ - vectors that best represent this data. If we denote these vectors by $\varphi^1,...,\varphi^\ell$ and further demand that they have to be orthonormal, the error term between one snapshot and its approximation can for $k=0,...,K$ be expressed by $||y_\perp^k - \sum_{i=1}^\ell \left(y_\perp^k,\varphi^i\right)\varphi^i||$. Adding up these defects using a trapezoidal rule on the time grid results in an error term for the whole trajectory. By formulating the objective as a minimization problem, this leads to

$$\left\{\begin{array}{c}\min\limits_{\varphi^1,...,\varphi^\ell\in X_N}\sum\limits_{k=0}^K \alpha_k\left|\left|y_\perp^k-\sum\limits_{i=1}^\ell(y_\perp^k,\varphi^i)\varphi^i\right|\right|^2\\ \text{s.t. } (\varphi^i,\varphi^j)=\delta_{ij}\quad\text{for } i,j=1,...,\ell\end{array}\right\} \tag{5.7}$$

where we have made use of the notation $\alpha_0=\alpha_K=\frac{\Delta t}{2}$, $\alpha_k=\Delta t$ $(k=1,...,K-1)$. Since X_N is given by $\mathbb{R}^{N_x}$ and the inner product is described by a positive definite weight matrix $W\in\mathbb{R}^{N_x\times N_x}$, this is just a reformulation of the POD problem $(P^\ell_{W,\alpha})$ presented in Section 2.3.3. As it was shown in Lemma 2.15, the solution to this problem can be obtained by one of two eigenvalue decompositions: Introducing the matrix $D:=\text{diag}\,(\alpha_0,...,\alpha_K)\in\mathbb{R}^{(K+1)\times(K+1)}$, one has the possibilities:

a) Compute the orthonormalized eigenvectors $\bar v^1,...,\bar v^\ell\in\mathbb{R}^{N_x}$ to the ℓ largest eigenvalues of the matrix $W^{1/2}Y_\perp DY_\perp^T W^{1/2}\in\mathbb{R}^{N_x\times N_x}$ and set $\psi^i:=W^{-1/2}\bar v^i$ for $i=1,...,\ell$.
b) Compute the orthonormalized eigenvectors $\bar u^1,...,\bar u^\ell\in\mathbb{R}^{K+1}$ to the ℓ highest eigenvalues $\bar\lambda_1,...,\bar\lambda_\ell$ of the matrix $D^{1/2}Y_\perp^T WY_\perp D^{1/2}\in\mathbb{R}^{(K+1)\times(K+1)}$, then set $\psi^i:=\bar\lambda_i^{-\frac12}Y_\perp D^{1/2}\bar u^i$ for $i=1,...,\ell$.

The question which of the two possibilities should be used depends on the numbers N_x and K: For $K>N_x$, a) is obviously the better choice since only an $N_x\times N_x$ eigenvalue problem has to be solved. If $N_x>K$, b) is the better option. In any way, once $\varphi^1,...,\varphi^\ell$ have been computed, we have also found the next ℓ basis elements by defining $\psi_{H+i}:=\varphi^i$ for $i=1,...,\ell$.

The question remains how the number ℓ is chosen. Here we consider the approximation error

$$\varepsilon_\ell := \sum_{k=0}^{K} \alpha_k \left\| y_\perp^k - \sum_{i=1}^{\ell} (y_\perp^k, \varphi^i)\varphi^i \right\|^2 \overset{(2.7)}{=} \sum_{i=\ell+1}^{N_x} \sigma_i^2$$

where $\sigma_1, ..., \sigma_{N_x}$ are descending singular values of the matrix $W^{1/2} Y_\perp D^{1/2} \in \mathbb{R}^{N_x \times (K+1)}$. Obviously, the quality of approximation is tied to the magnitude of the remaining singular values. Motivated by this, the following quotient term is defined

$$\widehat{\varepsilon}_\ell := \left(\sum_{i=1}^{\ell} \sigma_i^2 \right) / \left(\sum_{i=1}^{N_x} \sigma_i^2 \right) \qquad \text{for } \ell = 1, ..., N_x$$

This quotient is monotonically increasing with ℓ and has an upper bound of 1 which is reached at least for $\ell = N_x$. We will therefore use $\widehat{\varepsilon}_\ell$ as an indicator for the approximation quality and accept ℓ if $\widehat{\varepsilon}_\ell \geq \frac{q}{100}$ holds true for a predetermined *control factor* $q \in (0, 100)$, meaning that $q\%$ of the information from the detailed solution is contained in the vectors $\varphi^1, ..., \varphi^\ell$.

6

Experiments

In the previous chapters, we have presented different approaches for solving and then dimensionally reducing the equation (3.1). More specifically, variations have occurred for the following points:

- Discretization techniques:
  - ▷ The polynomial method (compare Section 4.1) with a θ-method for time-discretization (PolTheta)
  - ▷ The polynomial method with the classical Runge-Kutta method for time-discretization (PolRK4)
  - ▷ The FD method (compare Section 4.2) (FD)
- Strategies for the enhancement of the RB space (compare Section 5.2.2):
  - ▷ The Single-Time strategy (ST)
  - ▷ The Single-Time strategy with enhanced flexibility for the choice of the worst-error parameters (ST-flex$M$)
  - ▷ The POD strategy with a control factor $\frac{q}{100} \in (0, 1)$ (PODq)

This abundance in options will make it necessary to name the different variations. In order to do so, labels have already been used to uniquely identify each option. These labels are now stringed together using a hyphen. For example, a method where the Polynomial theta method has been used for discretization and the RB space was generated by the ST strategy using a flexibility of $M = 3$ will be labeled PolTheta-ST-flex3.

We will present various experiments which are aimed at comparing the performances of different options. For this, we will use fixed values for the parameters occurring in Chapter 3:
First of all, the variable ranges are chosen as $(\underline{\mathbf{x}}, \overline{\mathbf{x}}) = (0, 5)$ as well as $(0, T) = (0, 3)$. The initial condition is set to the constant value of the boundary condition, i.e. $y_0(\mathbf{x}) = g$ for all $\mathbf{x} \in (\underline{\mathbf{x}}, \overline{\mathbf{x}})$. Furthermore, this boundary value is given by $g = 6 \cdot 10^6 e^{1.8}$. Concerning the parameters $\mu \in \mathcal{D} \subset \mathbb{R}^3$, we will vary only the parameter μ_2, more precisely $\mu_2 \in [2, 3]$ and leave the other parameters fixed at $\mu_1 = 1.5$, $\mu_3 = 0.3$. Technically, this would mean $\mathcal{D} = \{1.5\} \times [2, 3] \times \{0.3\}$. However, since μ_1 and μ_3 will always stay fixed, we simply define $\mathcal{D} := [2, 3]$, understand out parameter only as μ_2 and hope that we are forgiven this minor inconsistency in notation.
The injection times are on the first and third day, so we have $t_1^* = 0$ as well as $t_2^* = 2$. This also means that the admissible control set is given by $U_{ad} = [0, 1]^2$ and the total set of control and parameter values is three-dimensional, where each value can effectively be expressed by the choice of (u_1, u_2, μ_2). Lastly, the rest of the model-related values are given by $TBV = 5000$, $\chi_{max} = 2 \cdot 10^5$, $T_{1/2} = \frac{10}{24}$ and $\beta = 0.6$.

Using the above values, the behavior of the various strategies can be analyzed. In a first section, we will present some results on the detailed solutions. This is done mainly to establish good choices for the discretization parameters to be used in the second section, where various RB strategies are examined.

6.1 Behavior of the detailed solutions

For every discretization technique above, i.e. PolTheta, PolRK4 and FD, good values have to be determined for the number N_t of time grid points, the dimension N_x of the space X_N and the parameter ϑ in case of PolTheta and FD. As a model case, the parameter values are set to $(u_1, u_2, \mu_2) = (0.5, 0.5, 2.5)$.

6.1.1 The PolTheta method

For the polynomial discretizations, the space dimension is given by $N_x = N + 1$, where $N \in \mathbb{N}_0$ is the maximum degree of the polynomials. One has to bear in mind that a rescaling of the Legendre polynomials $L_0, ...L_N : [-1, 1] \to \mathbb{R}$ has been used as a basis of the polynomial space $\Pi_N(\underline{\mathbf{x}}, \overline{\mathbf{x}})$. With an increasing degree N, these functions tend to oscillate more and more between -1 and 1, a property that is transferred to the basis functions $e_0, ..., e_N : [\underline{\mathbf{x}}, \overline{\mathbf{x}}] \to \mathbb{R}$ of X_N as can be

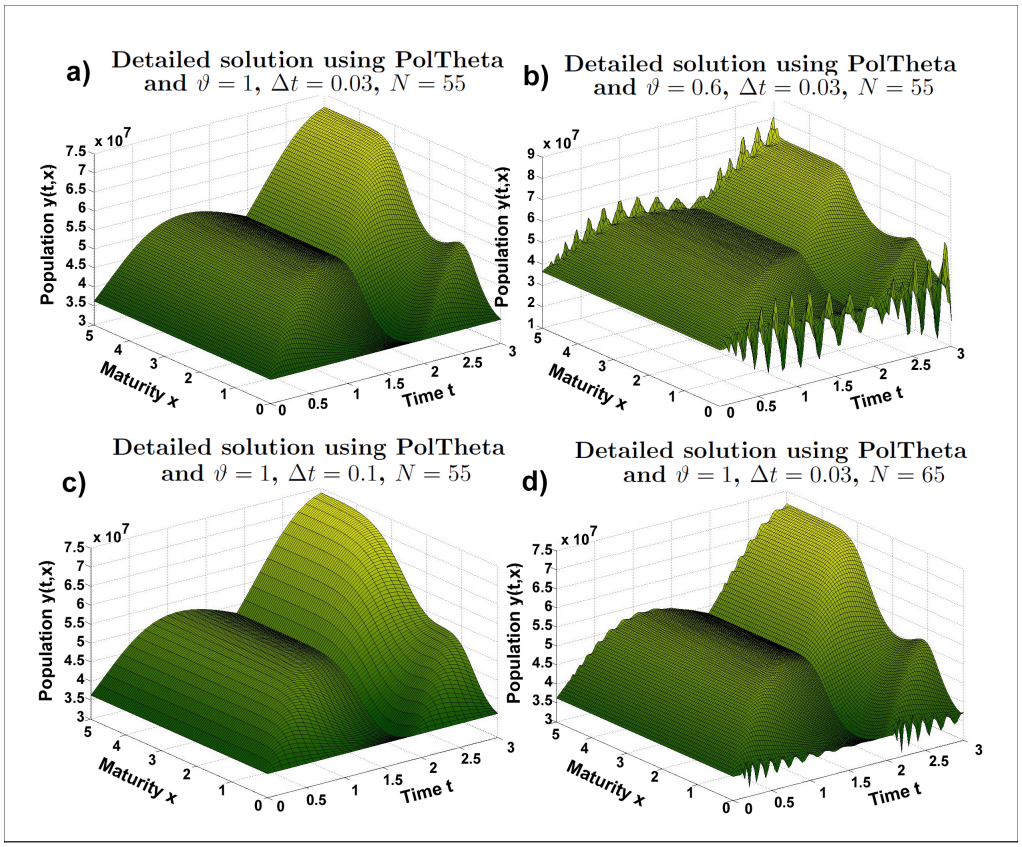


Figure 6.1: Detailed solutions using the PolTheta method and $(u_1, u_2, \mu_2) = (0.5, 0.5, 2.5)$

observed in Figure 4.1. As a result, it is only natural that instability effects occur if the number N is set too high. For example, we can observe in Figure 6.1a) a working set of discretizaton values given by $\vartheta = 1$, $\Delta t = 0.03$ and $N = 55$. By only increasing the polynomial degree to $N = 65$, we can see unwanted oscillations in Figure 6.1d), occurring at $\mathbf{x} = \underline{\mathbf{x}}$ and $\mathbf{x} = \overline{\mathbf{x}}$. This effect can be corrected by alternating Δt in a way that coarser time grids allow for higher polynomial degrees. If we limit ourselves to $\Delta t \leq 0.03$ - which in turn means a total number of at least 101 time grid points is used - oscillations occur for $N \geq 58$, meaning that the dimension of X_N is limited by $N_x \leq 58$. This will later on also limit the dimension H of the reduced space X_H for RB methods, seeing as reducing a space with $N_x = 58$ dimensions to $H = 30$ will be less effective than a reduction from $N_x = 1000$ to $H = 50$, as will be the case for the FD method.
In addition, we can see exemplarily in Figure 6.1b) that a reduction of ϑ causes further stability issues throughout the entire maturity interval $[\underline{\mathbf{x}}, \overline{\mathbf{x}}]$. This observation is quite common for hyperbolic equations in combination with a ϑ-method, see for example [19, Chapter 3]. In our case, we will simply set $\vartheta = 1$ which almost

guarantees stability as long as the above-mentioned maximal polynomial degree is not exceeded.

6.1.2 The PolRK4 method

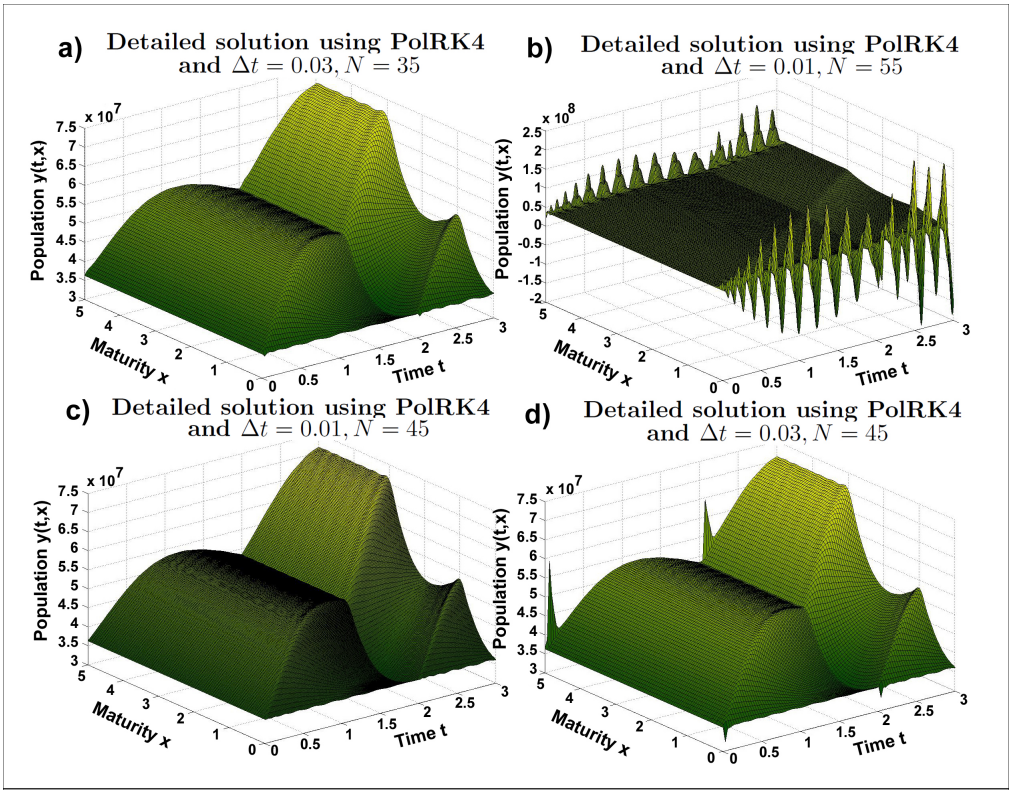


Figure 6.2: Detailed solutions using the PolRK4 method and $(u_1, u_2, \mu_2) = (0.5, 0.5, 2.5)$

First of all, we observe a very similar natural limit to the polynomial degrees as for the PolTheta method, which lies at $N \leq 40$ for $\Delta t = 0.03$. In this case however, instabilities can be remedied by increasing the number of time grid point as can be observed by comparing Figure 6.2 c) and d). However, we always observe oscillations occurring around $\mathbf{x} = \underline{\mathbf{x}}$, indepent of Δt or N. This can be explained by the fact that the time discretization in the PolRK4 method is done by the classical Runge-Kutta method. Being a purely explicit method, this technique offers poor stability but usually makes up for it by its high accuracy, which explains the fact that a finer time grid yields better results (as opposed to the PolTheta method). In this case however, it seems to be overstrained - a fact which is even fortified by observing Figure 6.2c): Even for the rather high step size $\Delta t = 0.01$, we can detect minor fluctuations in the solution for more advanced times $t \approx T$ and in

maturity regions close to $\mathbf{x} = \underline{\mathbf{x}}$. Figure 6.2b) finally shows that even for very fine time grids, a polynomial degree higher than $N = 50$ does not yield good results. In summary, we can conclude that the PolRK4 method is not suitable for the problem at hand. It will therefore not be tested for its performance with RB methods since the stability issues discussed above will make it impossible to distinguish between errors in the reduced solution and prior fluctuations in the detailed solution itself.

6.1.3 The FD method

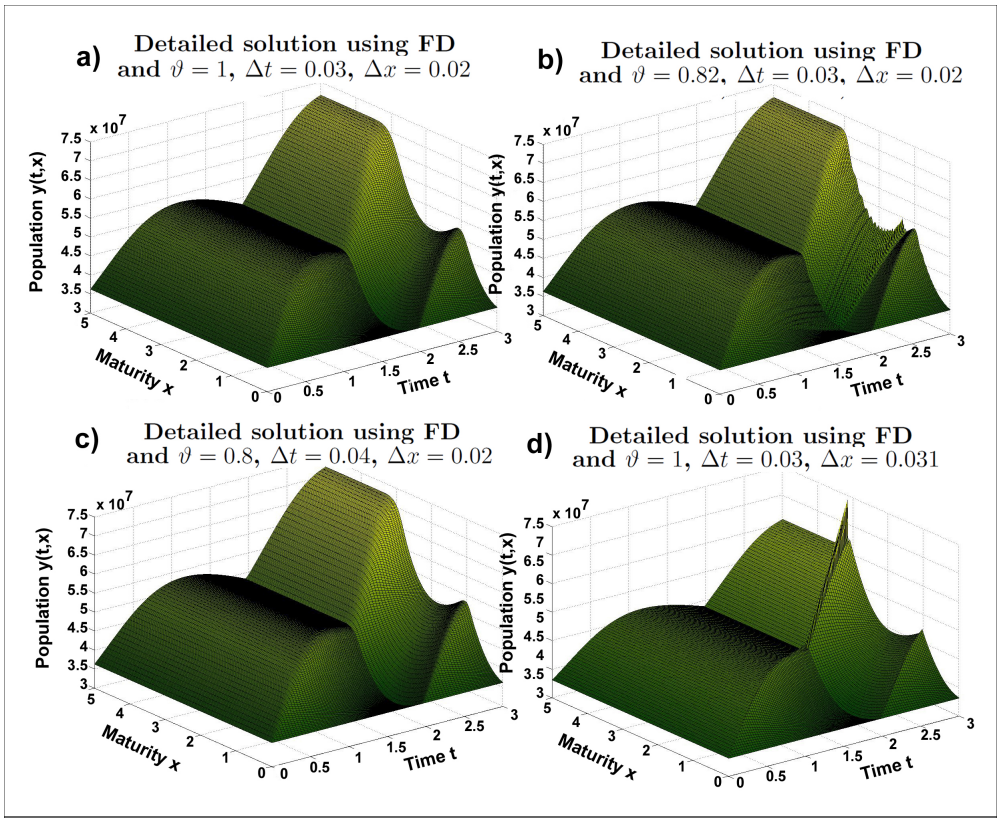


Figure 6.3: Detailed solutions using the FD method and $(u_1, u_2, \mu_2) = (0.5, 0.5, 2.5)$

In Figure 6.3 a) and c), we observe stable solutions using the FD method as opposed to b) and d), where fluctuations and spikes occur. Moreover, we can observe that the variable ϑ can not be responsible for stability by its own, since ϑ is lower in c) than in b), yet c) obviously presents a more stable solution. Also, $\vartheta = 1$ does not seem to guarantee stability as can be observed in d). As it turns out, another value besides ϑ has to be considered which is the quotient $\frac{\Delta x}{\Delta t}$. In order to test this hypothesis, simulations were run for 15 different settings of

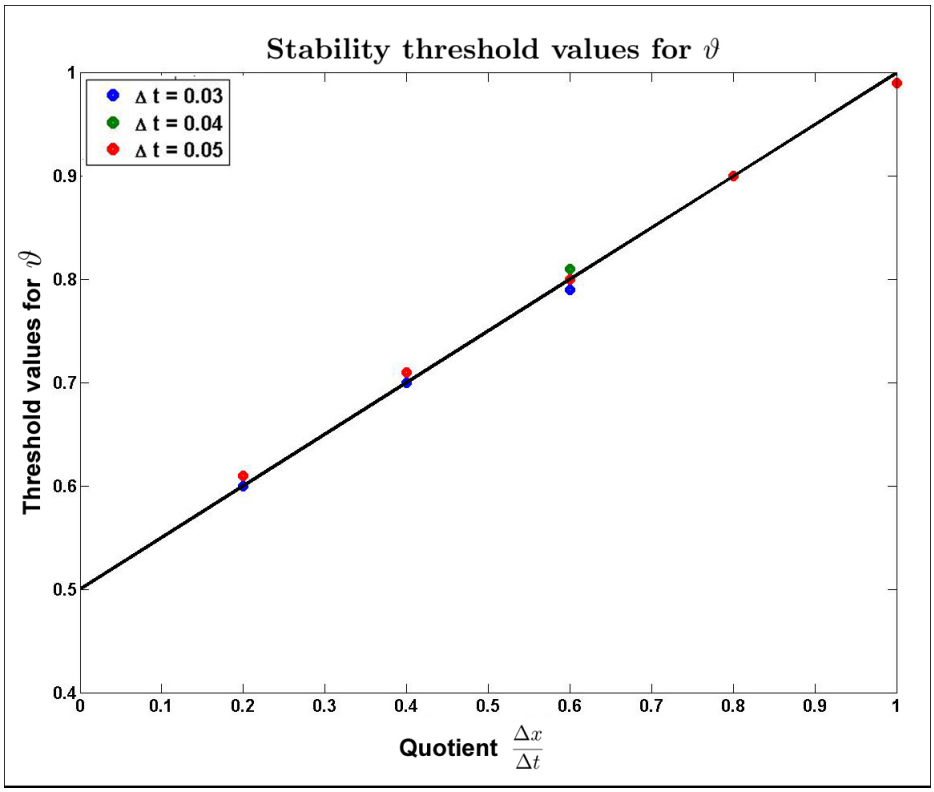


Figure 6.4: Threshold values $\vartheta(\Delta t, \Delta x)$ for $\Delta t = 0.03, 0.04, 0.05$ and correspondingly $\Delta x = \frac{\Delta t}{5}, \frac{2\Delta t}{5}, \frac{3\Delta t}{5}, \frac{4\Delta t}{5}, \Delta t$.

$(\Delta t, \Delta x)$. For each of these, there seems to be a threshold value $\vartheta(\Delta t, \Delta x)$ so that stability is reached exactly for $\vartheta > \vartheta(\Delta t, \Delta x)$. As can be seen in Figure 6.4, these values are all located along a straight with inclination 0.5 and y-axis intercept of 0.5. This means that a stability condition appears to given by

$$\boxed{\vartheta \geq \frac{1}{2} + \frac{\Delta x}{2\Delta t}}$$

In particular, this means that stability can only be achieved for $\Delta x \leq \Delta t$. A stability result such as this which depends on the quotient $\Delta x/\Delta t$ is typical for time-dependent problems, compare for example [5, Chapter 5.7.1] for a parabolical case.

6.1.4 Operator norms

Since all detailed solutions are computed by an iteration of the type (5.1) and the error estimators given by (5.3) and (5.4) depend on the operator norms of $\mathscr{L}_I^k(u, \mu)$ and $\mathscr{L}_E^k(u, \mu)$, we will analyze the magnitude of those norms. This will be done

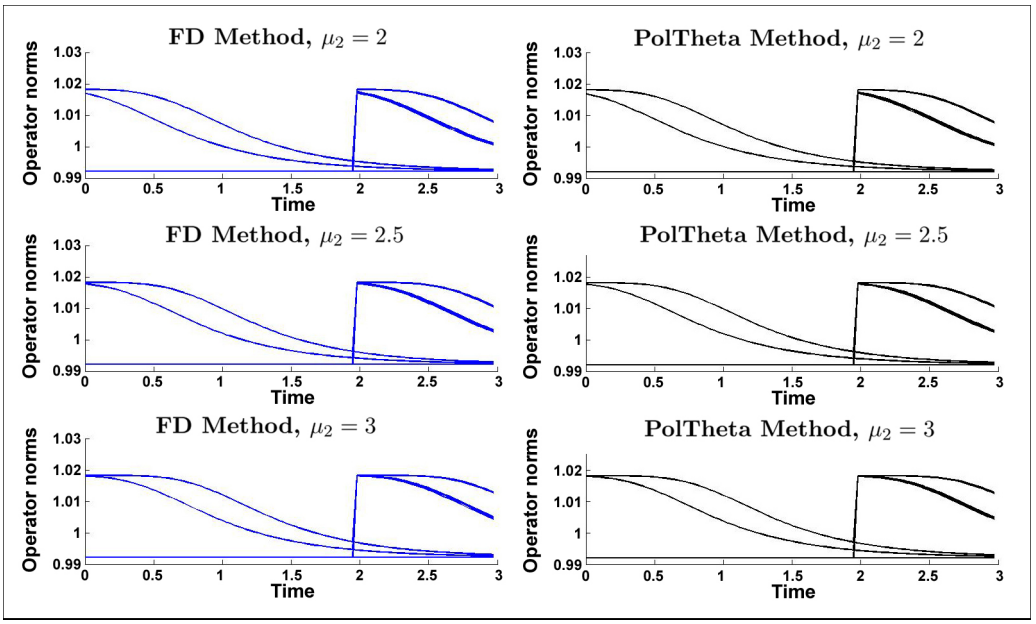


Figure 6.5: Operator norms $||(\mathscr{L}_I^k(u,\mu))^{-1}||$ over time for the FD method (left) and the PolTheta method (right), both for three different parameter. Each plot contains nine different operator norm trajectories for $u_1, u_2 \in \{0, 0.5, 1\}$.

for the PolTheta method and the FD method only, since PolRK4 has already shown to produce poor solutions due to stability issues. Also, as a consequence of the above observations, $\vartheta = 1$ will be chosen for the remaining discretizations, meaning that we will have fully implicit discretizations in time. This results in $\mathscr{L}_E^k(u,\mu) = 1$, so only the implicit operators are of interest. For these, the error estimator $\Delta_H^k(u,\mu)$ requires a boundary $C_I(u,\mu) > 0$ satisfying

$$\left|\left|\left(\mathscr{L}_I^k(u,\mu)\right)^{-1}\right|\right| \leq C_I(u,\mu) \qquad \text{for all } (u,\mu) \in U_{ad} \times \mathscr{D}$$

The operator norms can be computed following Corollary 2.7b), which is done for the parameter values $\mu_2 \in \{2, 2.5, 3\}$ and controls $u_1, u_2 \in \{0, 0.5, 1\}$, the results of which can be seen in Figure 6.5.

First of all, the dependency on μ_2 is almost neglectable, but the control u has an impact, especially at the injection times $t_1^* = 0$ and $t_2^* = 2$. However, a suitable upper bound C_I for all control-parameter sets still appears to be given by $C_I = 1.02$, with perhaps the exception of $(u_1, u_2) = (0, 0)$. This observations makes room for two possible strategies concerning the error estimator. First, one could use the values $C_I(u,\mu) = 1.02$ and $C_E(u,\mu) = 1$ for all $(u,\mu) \in U_{ad} \times \mathscr{D}$ along with the estimator Δ_H^k given by (5.3), thereby obtaining a rigorous error

estimator that is in fact an upper bound for the true errors. Alternatively, seeing as $1.02 \approx 1$, the simplified error estimator $\widetilde{\Delta}_H^k$ given by (5.4) could be used and checked for performance. However, let it again be noted that this version would not necessarily be rigorous, meaning that the true error could theoretically exceed the estimator. As can be seen in Figure 6.6, this was the case in some simulations.

6.2 Behavior of reduced solutions

In addition to the values defined at the beginning of this chapter, we will have to assign training sets $U_{train} \subset U_{ad}$ and $\mathcal{D}_{train} \subset \mathcal{D}$ whenever a reduced basis is generated. To keep things simple, these will always be chosen by an equidistant discretization of U_{ad} and $\mathcal{D}$ with 3 grid points along each dimension, leading to a training set $U_{train} \times \mathcal{D}_{train}$ containing a total of $3^3 = 27$ control-parameter values. For reviewing the impact of the RB method on the solutions, a second and finer equidistant discretization $U_{fine} \times \mathcal{D}_{fine} \subset U_{ad} \times \mathcal{D}$ is introduced with 7 grid points along each dimension, leading to a grid containing $7^3 = 343$ elements.
Furthermore, concerning discretization values, we carefully consider the findings from Section 6.1 and choose to perform RB strategies only for the PolTheta and the FD method, since we consider the PolRK4 method too instable to guarantee adequate results. For the PolTheta method, we choose a polynomial degree of $N = 58$, a time step size of $\Delta t = 0.03$ and $\vartheta = 1$, i.e. an implicit Euler method. For the FD method, the discretization values are set to $\Delta x = 0.02$, $\Delta t = 0.03$ and $\vartheta = 1$.

6.2.1 Performance of the error estimators I

At the end of Section 6.1.4, two possible strategies have emerged for the use of an error estimator concerning both the PolTheta and the FD discretization. One is to use the *simplified estimator* $\widetilde{\Delta}_H^k$ presented in (5.4), the other one was called the *canonical estimator* Δ_H^k and is given by (5.3) with the values $C_I(u, \mu) = 1.02$, $C_E(u, \mu) = 1$ for all $(u, \mu) \in U_{ad} \times \mathcal{D}$. Both these methods have to be compared to the true error. A first overview over the three trajectories over time can be seen in Figure 6.6 where a reduced basis of length $H = 10$ and three configurations for the values (u_1, u_2, μ_2) have been used to visualize the behavior over time. Seeing as the increments in the error estimators are used in the ST strategy to determine the next basis element, the time t_k has been plottet against the error increment $\Delta_H^k - \Delta_H^{k-1}$. As a first impression, both estimators appear to mimic the behavior of the true error trajectory, even though there are at times large differences.
Not much more can be said about which method performs better, so there is need

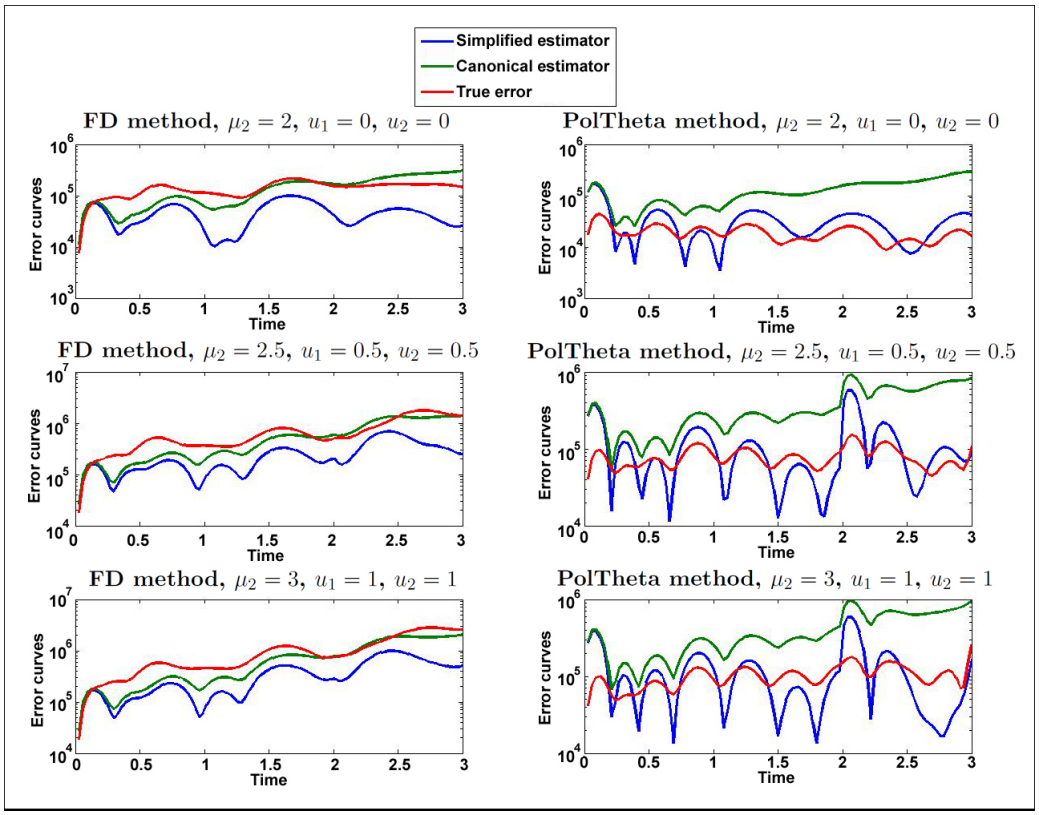


Figure 6.6: Trajectories of the simplified error estimator, the canonical error estimator and the true error for the FD method (left) and the PolTheta method (right), both for a reduced basis of length $H = 10$.

for further analysis: In order to do so, we will use the grid $U_{fine} \times \mathcal{D}_{fine} \subset U_{ad} \times \mathcal{D}$ introduced at the beginning of this section. For each pair $(u, \mu) \in U_{fine} \times \mathcal{D}_{fine}$, the medium accuracy over time of the estimators is rated by computing

$$\xi_H(u,\mu) := \left(\frac{\sum_{k=0}^{K} \alpha_k \left| \left(\Delta_H^k(u,\mu) - \Delta_H^{k-1}(u,\mu) \right) - \left| \left| y_N^k(u,\mu) - y_H^k(u,\mu) \right| \right| \right|^2}{\sum_{k=0}^{K} \alpha_k \left| \left| y_N^k(u,\mu) - y_H^k(u,\mu) \right| \right|^2} \right)^{1/2} \tag{6.1}$$

with the weights $\alpha_0, ..., \alpha_K$ of the trapezoidal time rule. Of course, a smaller value for $\xi_H(u, \mu)$ indicates a higher accuracy of the estimator. Taking the mean over all $(u, \mu) \in U_{fine} \times \mathcal{D}_{fine}$ results in an accuracy indicator $\xi_H > 0$ which only depends on the reduced space X_H. By iteratively generating such a space using Algorithm 1 and computing ξ_H in the process, we obtain a graphical representation that can be seen in Figure 6.7. This result illustrates that the simplified error estimator does

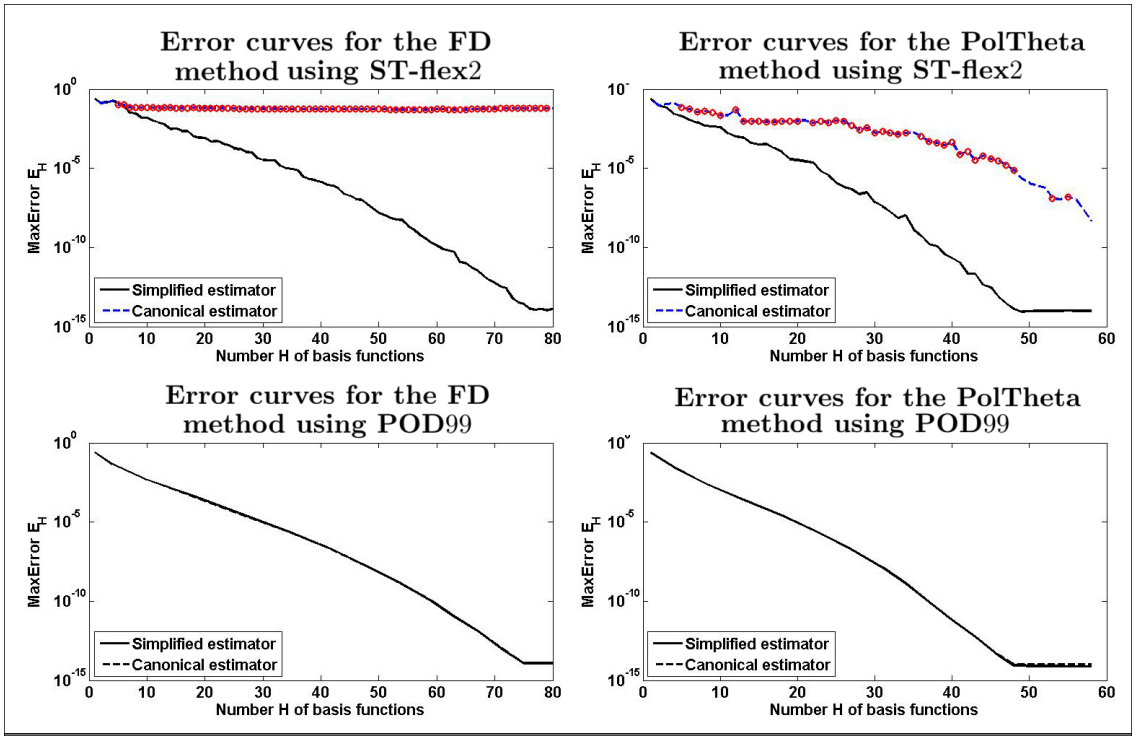


Figure 6.7: Accuracy ξ_H of the simplified and the canonical error estimator over the dimension H of the reduced space, both for the FD method (left) and the PolTheta method (right)

indeed perform more accurately than the canonical one, at least for the PolTheta method. For the FD method, we recognize a shift in accuracy depending on the number H of basis functions. However, the values of the canonical estimator for larger values of H still exceed those of the simplified estimator for lesser H-values, making the latter the better choice.
To underline this hypothesis, a performance test is done to measure the general quality of the respective RB spaces. This is done by computing a series of reduced spaces $X_1 \subset X_2 \subset \ldots \subset X_{H_{max}}$ following Algorithm 1. For each space X_H, the reduced solutions $\{y_H^k(u,\mu)\}_{k=0}^K$ are computed for all $(u,\mu) \in U_{fine} \times \mathcal{D}_{fine}$. Additionally, the detailed solution $\{y_N^k(u,\mu)\}_{k=0}^K$ is computed for $(u,\mu) \in U_{fine} \times \mathcal{D}_{fine}$. We can then calculate the true error $e_H^k(u,\mu) := ||y_N^k(u,\mu) - y_H^k(u,\mu)||$ for every time instant $k = 0, ..., K$. Similarily to the accuracy indicator $\xi_H(u,\mu)$ defined in (6.1), we obtain a relative error $e_H(u,\mu) > 0$ for the entire trajectory by

$$e_H(u,\mu) := \left(\frac{\sum_{k=0}^{K} \alpha_k \left|\left| y_H^k(u,\mu) - y_N^k(u,\mu) \right|\right|^2}{\sum_{k=0}^{K} \alpha_k \left|\left| y_N^k(u,\mu) \right|\right|^2} \right)^{1/2} \tag{6.2}$$

Again, $\alpha_0, ..., \alpha_K$ are the weights stemming from the trapezoidal rule on the time grid. Finally, taking $e_H := \max_{u \in U_{fine}} \max_{\mu \in \mathcal{D}_{fine}} e_H(u,\mu)$ gives us an error indicator depending only on the space X_H. Doing this utilizing both the simplified and the canonical estimator will present a feedback to their eligibilities.

The results can be seen in Figure 6.8. For the ST method, it is possible that a snapshot gets chosen which has already been used in a previous iteration (a so-called *double-pick*). These are indicated with red circles. We see a lot of these when the canonical estimator is used, which therefore performs a lot worse than the simplified estimator. The reason for this is that the canonical estimator largely favors late time instances for the choice of snapshots whereas the simplified version does not have this tendency, as can be seen in Figure 6.9. This can be explained by the exponential dependence on the time in definition (5.3), leading to larger values for higher values of k. As a result, the algorithm stagnates and fails to incorporate solution properties at earlier times into the reduced basis. For the POD method on the other hand, there does not seem to be any difference between the two estimators.

To sum up, the simplified estimators is clearly the overall better choice. Therefore, it will be used for the rest of the experiments.

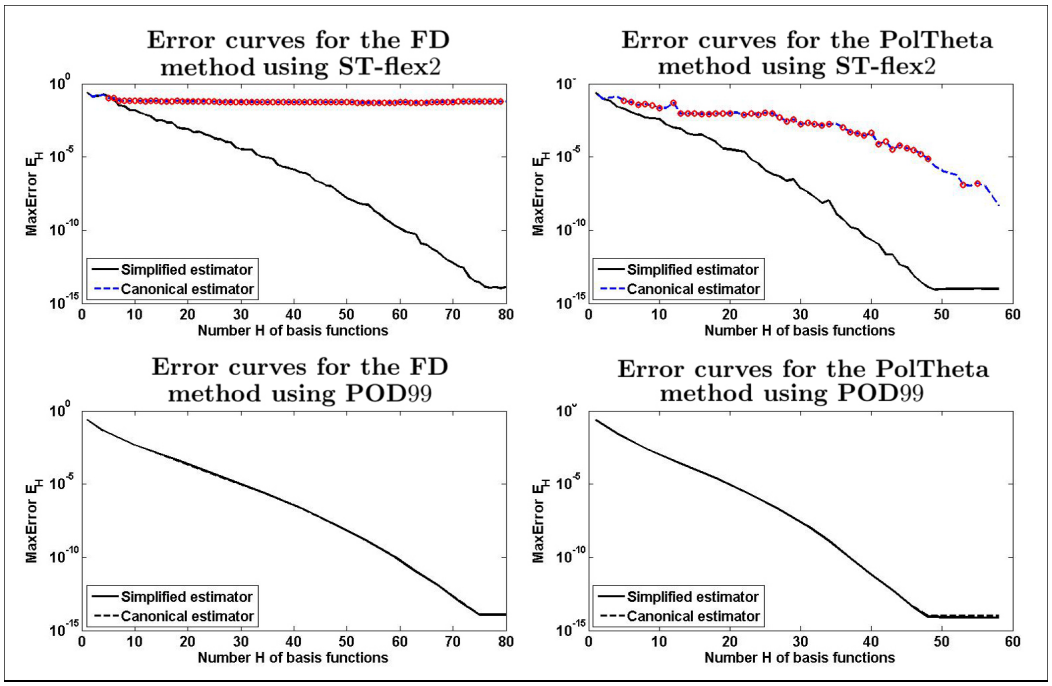


Figure 6.8: Comparison of error estimator performance for the FD method (left) and PolTheta method (right), both with the ST-flex2 strategy (top) and POD99 stratgey (bottom). Red circles indicate double-picks in the ST strategy

6.2.2 Choice of worst-error parameters during the Greedy search

During Algorithm 1, the set $(u^*, \mu_2^*) \in U_{train} \times \mathcal{D}_{train}$ with the highest error estimator is determined. It is essential to know which parameters are chosen more frequently than others because this knowledge leads the way to more efficient choices for the training grids U_{train} and $\mathcal{D}_{train}$. The current equidistant grid is three-dimensional and is contained in the boxed set $[0,1]^2 \times [2,3]$. A visual representation can be seen in Figures 6.10 and 6.11 where the centers of the circles represent the exact values of (u_1, u_2, μ_2) and the radii are scaled to the number of occurences this particular control-parameter set had during the Greedy algorithm. A circle with twice the radius indicates that the corresponding control-parameter set has been chosen twice as often.
We observe that corner values, especially $(u_1, u_2, \mu_2) = (1, 1, 3)$ are strongly favored by all strategies and for all discretization methods. Moreover, all of the chosen sets contain the value $u_1 = 1$ or $u_1 = 1/2$. For the FD discretization, we can additionally see a strong favoring of values with $\mu_2 = 3$. As a consequence, one can reflect on the question whether an equidistant threedimensional grid spanning the entire domain of training parameters is really the best way to go for generat-

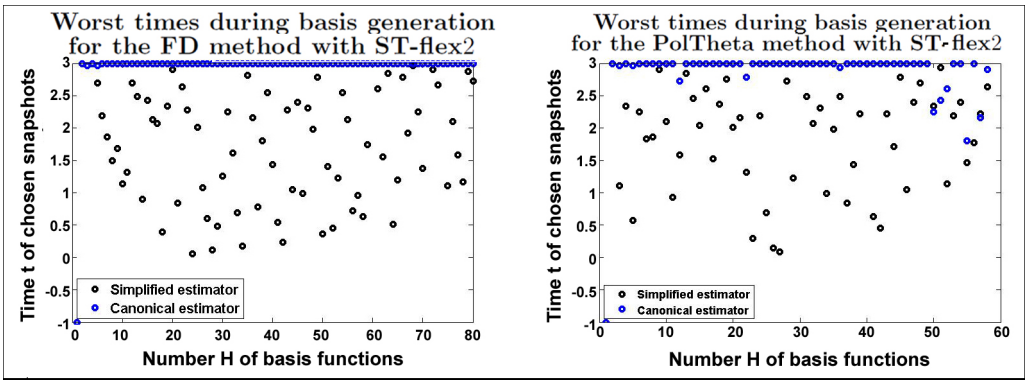


Figure 6.9: Times of chosen snapshots for the simplified and the canonical estimator, both for the FD method (left) and the PolTheta method (right) and the ST-flex2 strategy.

ing a reduced space. For example, one could imagine grids containing only values with $u_1 \in [1/2, 1]$. In addition, seeing as corner values appear to be preferable to the algorithms, grids do not have to be equidistant along each axis. They could rather be grouped closer towards corners and further apart towards the middle of the cube.
However, the fact that some parameters were chosen time and time again whereas others were not chosen at all does not necessarily mean that the quality of the computed space is poor. It might well be that the data information added from the corner points also provides sufficient representation for other parameter values. Further analysis would be required here to see if more carefully chosen training grids provided better and/or faster Greedy algorithms.

6.2.3 Analysis of the flex-M variation

In this section, the impact of the flex-M variation which adds flexibility to the choice of worst-error parameters for the ST method (compare Algorithm 5.1) is analysed. In order to do so, the performance of these strategies has to be measured in some way. We choose the maximum-error value ε_H depending on the dimension H of the reduced space which has been established in Section 6.2.1, more exactly in (6.2). Again, this value represents an indicator as to how effectively the error between detailed and reduced solution has decreased over the iterations of the Greedy algorithm with respect to a very fine grid on the control-parameter space.

In Figure 6.12, one can see these max-error curves in comparison for the PolTheta and the FD method, both with the standard and the flex2 variations. For the FD method, the added flexibility obviously offers a large improvement by avoiding a

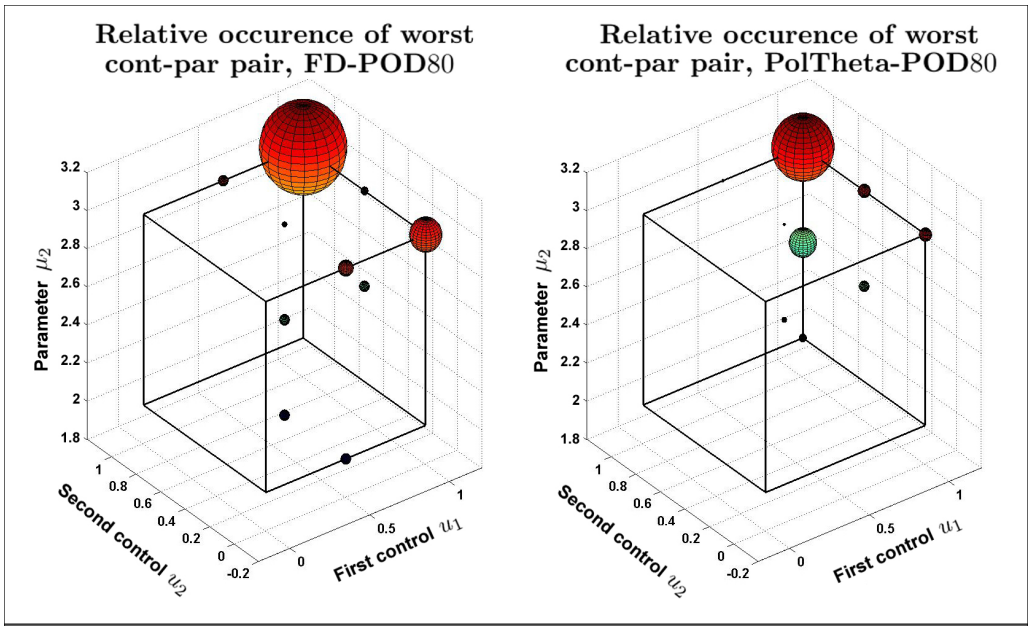


Figure 6.10: Illustration of the parameter choices for a POD80 method with FD discretization (left) and PolTheta discretization (right).

total stagnation after the third iteration, after which the standard variation insists on picking the same snapshot to the same parameters again and again. This is remedied by the flex2 option which contains no double-picks at all and produces a series of reduced solutions with an error curve that is consistently decreasing linearly on a logarithmic scale.
For the PolTheta method, the flex2 variation does perform slightly better than than the standard strategy by avoiding one double-pick at $H = 7$. In the whole however, both variations work quite well, so the choice of flex2 remains optional here, in contrast to the ST method.
To sum up, the flex$M$ variation can keep the algorithm from stagnating because of an endless choice of the same snapshot.

6.2.4 Analysis of the impact of q on the PODq method

As it was mentioned at the end of Section 5.2.2, the factor $q \in [0, 100]$ in the PODq strategy sets a lower limit to the percentage of information within the detailed trajectory $\{y_N^k(u^*, \mu^*)\}_{k=0}^K$ which is supposed to be contained in the reduced basis vectors $\psi_{H+1}, ..., \psi_{H+\ell}$. Since a larger percentage of information requires additional POD vectors, this means that an increase of q also means an increase in the number ℓ of basis functions which are added within each iteration. By choosing a very large q, this may lead to a certain control-parameter set $(u, \mu) \in U_{train} \times \mathcal{D}_{train}$

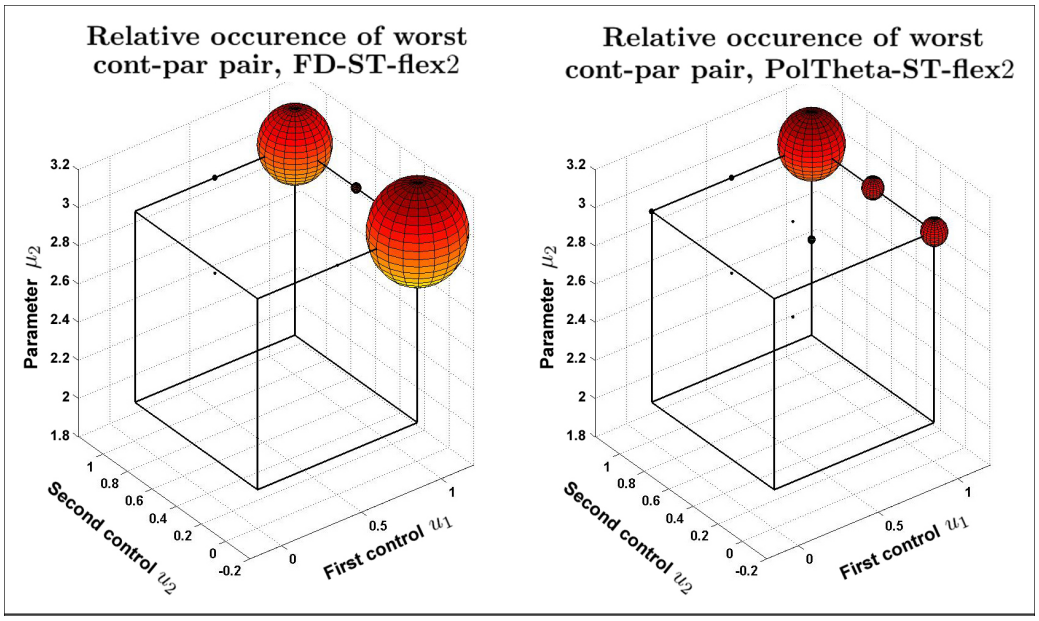


Figure 6.11: Illustration of the parameter choices for a ST-flex2 method with FD discretization (left) and PolTheta discretization (right).

being very well represented by the reduced basis Ψ_H while other sets are not accounted for at all. In simpler words, it is possible that the algorithm spends to much effort on the first chosen sets, thereby neglecting other ones. We will therefore analyse the effect the factor q has on the algortihm by choosing different values for it, namely $q = 80, 90, 99$.

The results can be seen in Figure 6.13 and are quite surprising at first: Neither in the FD nor in the PolTheta method, the number q seems to have any larger impact. This can be explained by contemplating some POD properties for a simplified example.

Let us therefore assume that we find ourselves in the first iteration of Algorithm 1 (with a PODq strategy for the basis enhancement phase, of course). This means that ℓ new basis vectors are generated as a solution of the minimization problem (5.7). Now suppose that we have two different settings for q: The larger one will produce two new orthonormal basis vectors $\varphi^1, \varphi^2 \in X_N$ whereas the smaller one will only generate one new vector $\widetilde{\varphi}^1 \in X_N$, after which it will find itself in a second iteration. Now, if the same control-parameter pair $(u^*, \mu^*) \in U_{train} \times \mathcal{D}_{train}$ is chosen for this iteration (which actually happens a lot for the problem at hand, compare Figures 6.10 and 6.11) and there is again only one new basis vector

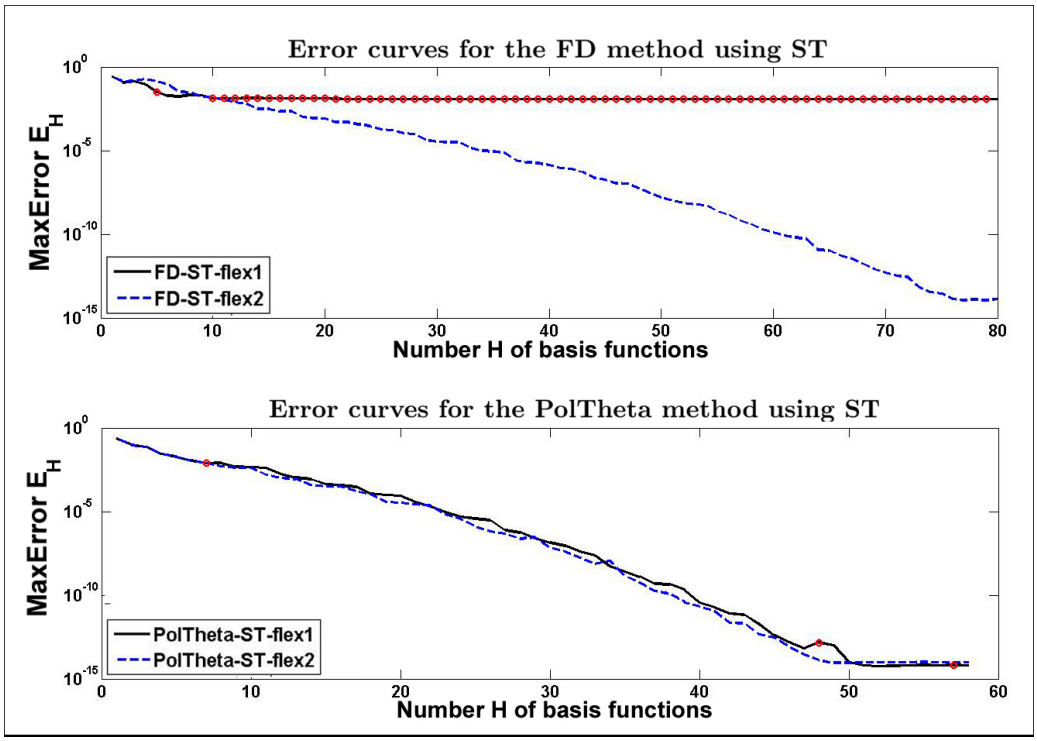


Figure 6.12: Max-Error curves E_H plot against the number H of basis functions for the ST method. Red circles indicate double-picks.

$\widetilde{\varphi}^2 \in X_N$ generated which is in turn a solution to the problem

$$\left\{ \begin{array}{c} \min\limits_{\widetilde{\varphi}^2 \in X_N} \sum\limits_{k=0}^{K} \alpha_k \left|\left| y_\perp^k - (y_\perp^k, \widetilde{\varphi}^2)\widetilde{\varphi}^2 \right|\right|^2 \\ \text{s.t.} \quad \left|\left| \widetilde{\varphi}^2 \right|\right| = 1 \end{array} \right\}$$

where $y_\perp^0, ..., y_\perp^K \in X_N$ are the projections of the solution vectors $y^0, ..., y^K$ onto $\{\widetilde{\varphi}^1\}^\perp$. Note that the above problem is equivalent to

$$\left\{ \begin{array}{c} \min\limits_{\widetilde{\varphi}^2 \in X_N} \sum\limits_{k=0}^{K} \alpha_k \left|\left| y^k - (y^k, \widetilde{\varphi}^2)\widetilde{\varphi}^2 \right|\right|^2 \\ \text{s.t.} \ (\widetilde{\varphi}^1, \widetilde{\varphi}^2) = 0 \text{ and } \left|\left| \widetilde{\varphi}^2 \right|\right| = 1 \end{array} \right\}$$

By comparing the generation processes of φ^1, φ^2 and $\widetilde{\varphi}^1, \widetilde{\varphi}^2$, it is perhaps not too surprising that $\varphi^1 = \widetilde{\varphi}^1$ and $\varphi^2 = \widetilde{\varphi}^2$ holds true. In fact, this was shown in [21]. In very simple words, φ^1 and φ^2 are chosen as the two vectors that together present the best two-dimensional approximation of the data[1]. Similarily, $\widetilde{\varphi}^1$ is chosen as the best one-dimensional approximation of the data and $\widetilde{\varphi}^2$ as the best one-

[1] 'data' is a short term for the solution trajectory $\{y_N^k(u^*, \mu^*)\}_{k=0}^K \subset X_N$ in this case.

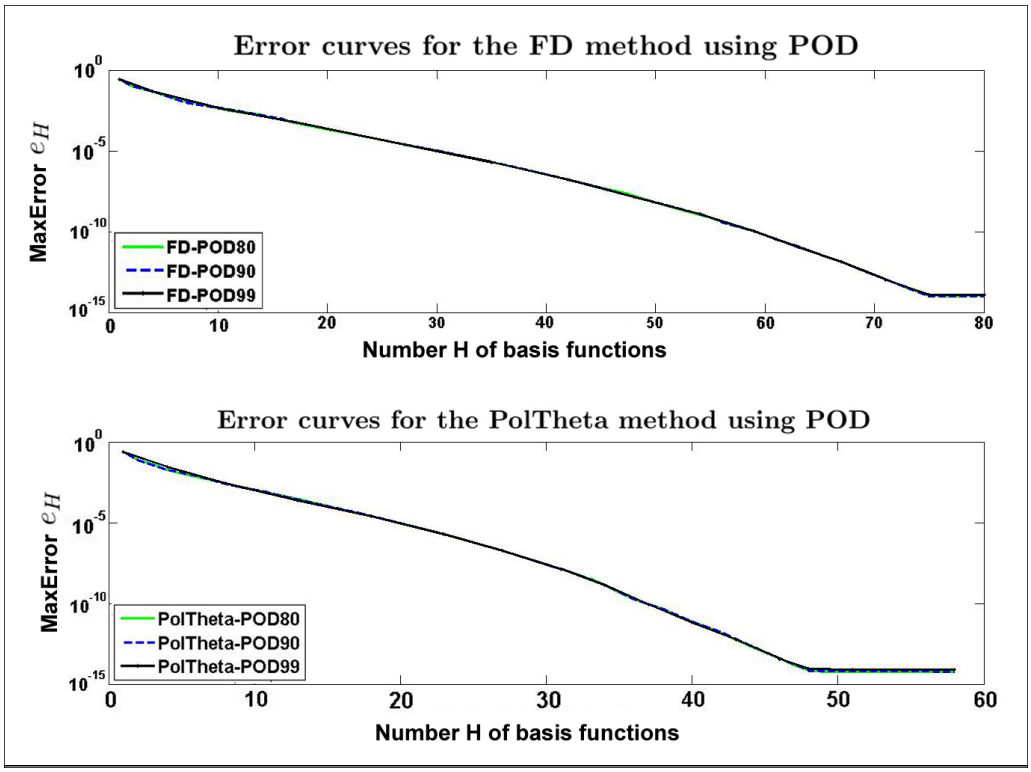


Figure 6.13: Max-Error curves E_H plot against the number H of basis functions for the ST method.

dimensional approximation of the 'remaining' data.
This in a nutshell is the reason why the error lines for various settings of q overlap so much: For lower values of q, a very similar basis is created than with larger values, only the generation is split up into more iterations. Seeing as every iteration requires some computations in Algorithm 1, it would seem that choosing $q = 99$ is the most feasible setting for our purposes at the time.

6.2.5 Performance of the ST vs. the PODq strategy

For each discretization technique, there are two basis generation strategies available: The ST method which adds one single snapshot in each iteration and the PODq method which adds POD approximation vectors until q% of the solution trajectory is represented. A comparison for these two strategies can bee seen in Figure 6.14, for which a flex2 variation was used for the ST strategy and $q = 99$ was chosen for the PODq method, both as a result of the analysis provided in Sections 6.2.3 and 6.2.4.

For the FD discretization, it is perceivable that the PODq method performs sig-

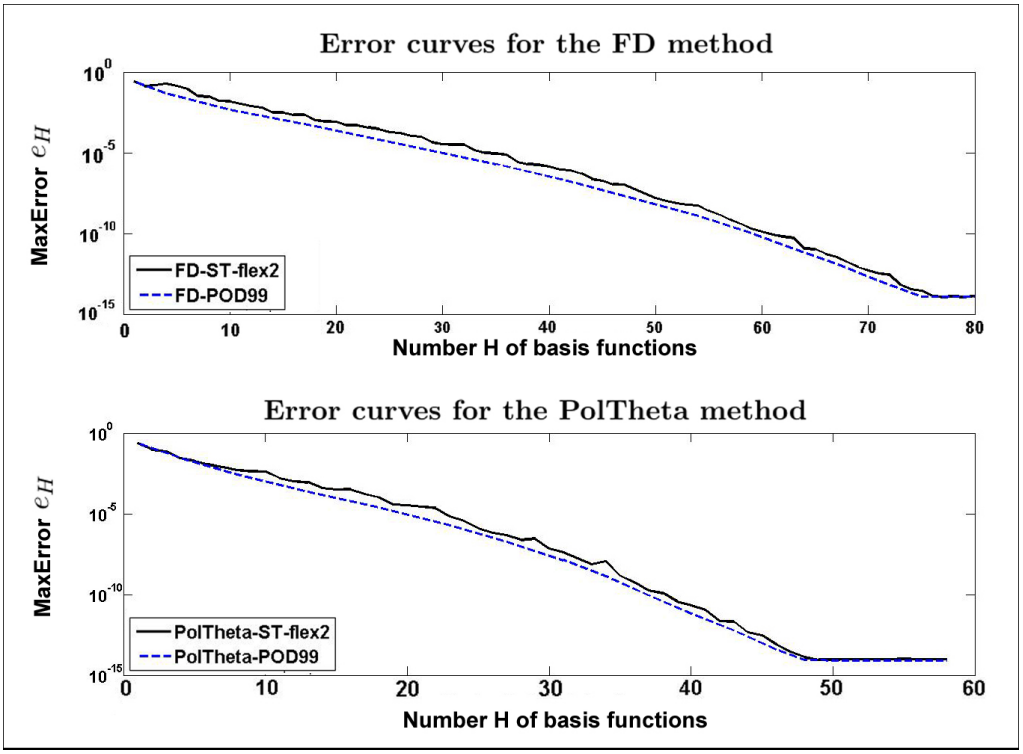


Figure 6.14: Worst-Error curves for the ST (top) and the PolTheta (bottom) strategy, both for the ST-flex2 and the POD99 method.

nifically better, seeing as its maximum error continually falls below the maximum error of the ST strategy. Also, the PODq line is monotonically decreasing whereas the ST line is sometimes momentarily increasing. This is due to the fact that by design, the PODq strategy for basis enhancement only adds vectors $\psi_{H+1}, ..., \psi_{H+\ell}$ that are still contained in the space orthogonal to the one spanned by the current basis Ψ_H, ensuring that the next basis $\Psi_{H+\ell}$ does indeed contain more information of the data. Furthermore, the added basis elements are chosen as vectors of maximum variance, thereby actively enhancing the current space X_H by most of the information that is not yet contained in it. The ST method on the other hand has a more empirical nature: It is not assured that the added information is not already contained in X_H. This does sometimes lead to the fact that the addition of a new snapshot slightly worsens the maximum error of the reduced solution.
A similar trend can be observed for the PolTheta method with the minor difference that the ST method does keep up much better during the first 10 basis functions. For $H \geq 10$, the PODq method gains further advantage, making it the altogether superior method as far as accuracy is concerned. The other important factor is feasibility which will be examined in Section 6.2.6.

6.2.6 Computation times I: Reduced vs detailed solution

If we reconsider the motivations for Reduced Basis methods introduced at the beginning of Chapter 5, the main objective was to reduce the computation time needed for the detailed solution while maintaining enough accuracy as possible. As far as accuracy is concerned, the last sections proved that both the ST and the PODq strategy deliver on accuracy if the number H of basis functions is set high enough, with the PODq method performing slightly better. In order to examine the feasibility issue, we can see in Figure 6.15 the computation times for both the detailed and the reduced solutions.

First of all, one can see that for the FD method, the computation time for the reduced solution exceeds the detailed computation time for $H \geq 20$. This appears to be rather strange at first, seeing as the lower-dimensional system takes in fact more time to calculate the solution than the high-fidelity system. This can be explained by examining the two recursions by which the solutions are computed, namely (4.8) for the detailed solution and (5.2) for the reduced solution. Looking at the occurring implicit operators $\mathscr{L}_I^k(u, \mu)$ for the high-dimensional system, we have a linear combination of the matrices $\mathbb{1}_{N_x}, B_{N_x} \in \mathbb{R}^{N_x \times N_x}$. These matrices are therefore quite sparse, seeing as only the main diagonal and the first lower diagonal are occupied. On the other hand, the reduced implicit operators $L_I^k(u, \mu)$ of the reduced-order system are generally full matrices. Given the speed advantage when solving a sparse linear equation system as opposed to a nonsparse one, this

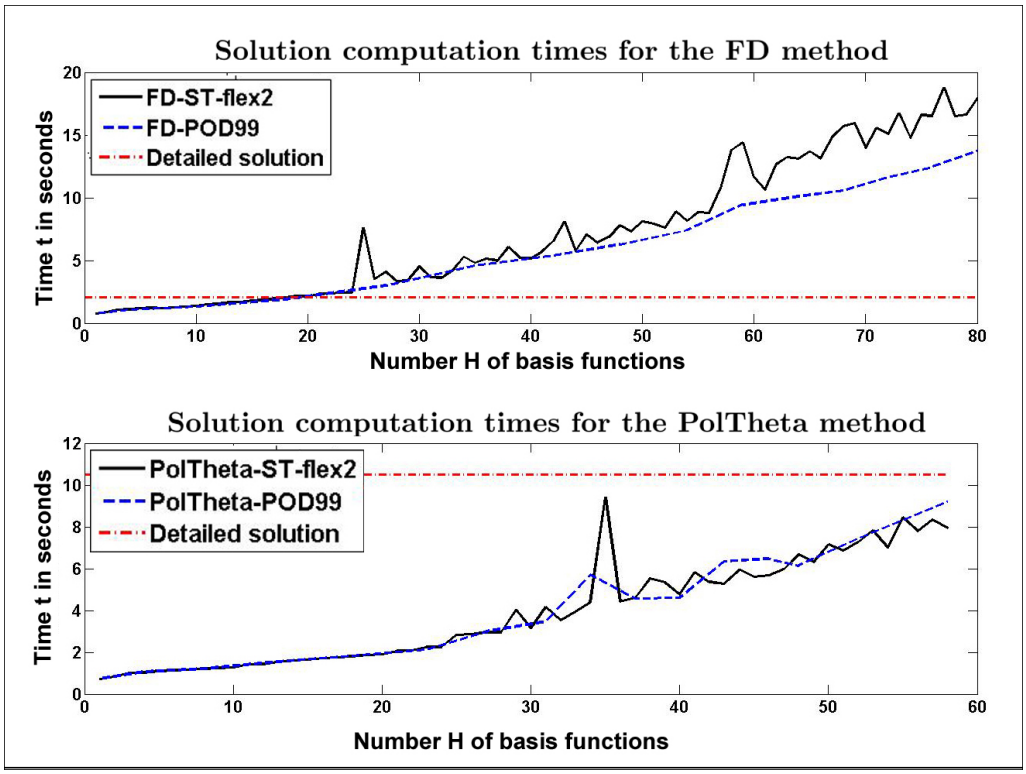


Figure 6.15: Computation times for the detailed and reduced solutions, both for the FD (top) and the PolTheta method (bottom).

explains the fact the MOR actually makes the problem more costly to solve: The FD discretization technique applied here is simply too efficient to begin with. This is mostly due to two reasons: First, the P^{2}DE (3.1) is a very simple one, seeing as it is one-dimensional, has only two input arguments and is linear as well as of first order. Second, the chosen discretization options within the FD method are both of first order (forward-Euler for the variable $\mathbf{x}$ and backward-Euler for the variable t) and therefore very fast. Turning to discretization techniques of higher order would necessarily require more computation time than is the case right now. Turning to the PolTheta method, we observe that there is actual speed gained for the solution computation when applying the MOR technique. In contrast to the FD method, the implicit operators $\widehat{\mathcal{L}}_I^k$ are full matrices, they are created by adding up multiples of the matrices $\mathbb{1}_N, B_N, \Gamma_N \in \mathbb{R}^{(N+1)\times(N+1)}$, the latter of which is a matrix with no zeros in it. It is therefore quite natural that solving the lower-dimensional model should take less time.

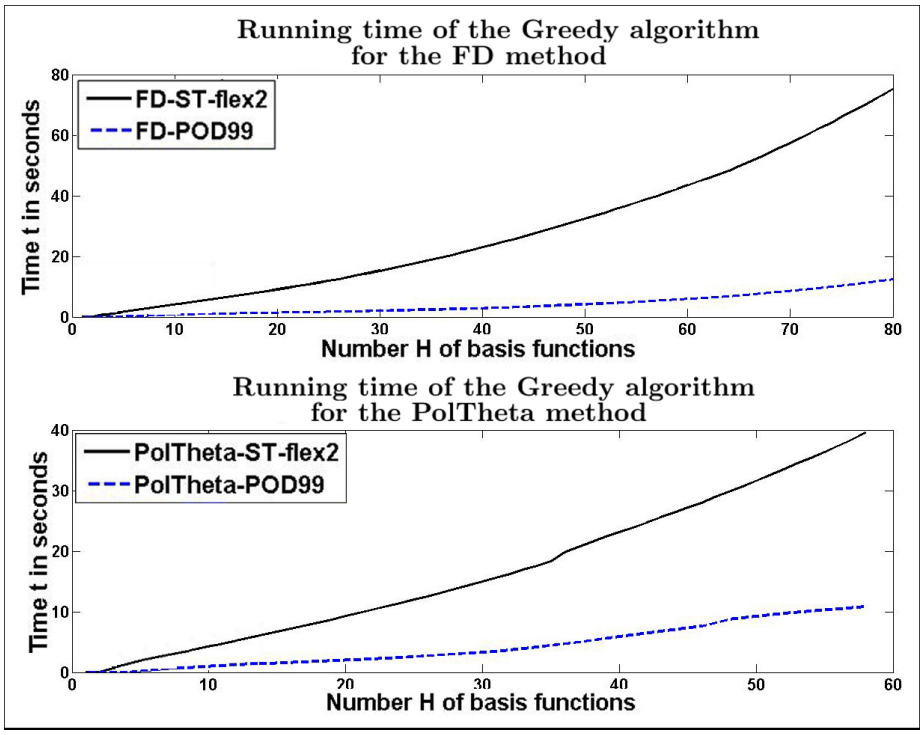


Figure 6.16: Computation times of the Greedy algorithm for the FD and the PolTheta discretization, both for the ST-flex2 and the POD99 strategy

6.2.7 Computation times II: Basis generation

One has to consider that by applying a RB strategy, the speed gained by a faster computation of solutions has to be put in contrast to the time it takes to actually construct the reduced space by using the Greedy search. In Figure 6.16, we can see these generation times depending on the dimension H of the reduced space. As could be expected, the POD99 strategy is much faster in both cases. This is due to the fact that it adds not one, but several new basis vectors in each iteration, thereby saving a lot of computation time. However, one has to consider that this gained speed comes at the prize of having to solve an eigenvalue problem of the size $N_t \times N_t$ respectively $N_x \times N_x$ in each iteration. Seeing as both values are rather low for the system at hand with $N_t = 100$ and $N_x = 250$, this did not present a problem here. Nevertheless, when considering the application of the methods used here to higher-dimensional cases, these eigenvalue computations will get more costly, thereby slowing down the PODq method.

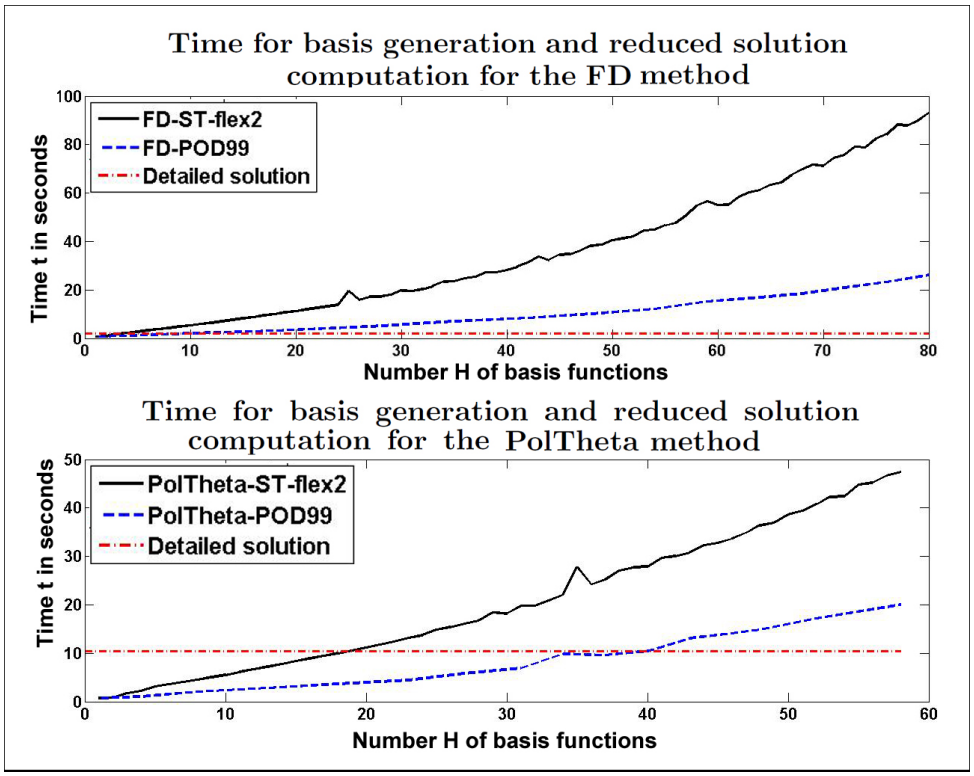


Figure 6.17: Computation time of the detailed solution versus the total running time of the reduced-order technique, both for the FD and the PolTheta discretization.

6.2.8 Computation times III: Total time

In Figure 6.17 is shown a final computation time comparison between the detailed solution and the reduced-order technique. The latter is composed of the running time of the Greedy search plus the computation time for the reduced solution. As was to be expected after Section 6.2.6, we can see that MOR isn't a sensible option for the FD discretization, neither with the ST nor with the POD strategy. Reasons for this have already been given in Section 6.2.6.
Moreover, we now observe that even for the PolTheta discretization, the speed which is gained by reduced-order modelling is nearly neglectable. Even with the faster POD99 strategy, choosing a reduced space of dimension $H = 20$ only saves about half the time it would have taken to solve the entire high-fidelity system. When considering that this little gained speed comes with a relative error of about $\varepsilon_H \approx 10^{-5}$ (compare Figure 6.13), we have to conclude that MOR techniques are not worthwhile for the problem at hand. Again, the reason for this lies in the simplicity of both the underlying P^2DE and the used discretization methods, which results in very fast detailed solvers that really do not require reduced-order

modeling to be effective.

7

Conclusion and Outlook

In this thesis, we have investigated the application of MOR techniques to a model for erythropoesis that simulates the population of CFU-E cells under the influence of external EPO administration. The model was provided by the Renal Research Institute in New York and modified to facilitate the implementation of the RB strategies. It consists of a linear, hyperbolic P^2DE of first order along with initial and boundary conditions.

In a first step, two discretization methods were introduced to obtain the detailed solution, including a FD method and a polynomial-based variant as it was applied in [7]. Thereupon, a reduced-order model was derived that is able to provide a low-dimensional projection of these and other discretizations. The developed methods include the generation of a reduced approximation space generated through the use of a Greedy algorithm, which iteratively builds a reduced basis. This can be done using a POD-based or a snapshot-based strategy. Additionally, an a-posteriori estimator was derived to assess the magnitude of the approximation error.

Both Greedy strategies were observed to generate qualitatively and quantitatively suitable reduced spaces that are able to approximate detailed solutions for the entire parameter domain. In a direct comparison, however, the POD strategy proved not only to be much faster, but it also led to the construction of a slightly better space than the snapshot-based strategy.

Concerning efficiency, the experiments showed that the application of reduced-order strategies even increased the simulation time of the FD method. For the polynomial-based discretization on the other hand, running time was decreased, but not significantly enough to justify the approximation error. All in all, it can be said that an application of MOR techniques for the present model is not practical. The reason for this lies in the simplicity of the equation, which makes high-fidelity solvers already sufficiently fast.

With the above observation in mind, however, applying the MOR methods introduced in this thesis to more complex cell population models is a very promising next step. One can easily consider variations of this kind of population equation that contain more detail, for example, when two or more types of cells are considered and a coupled system of P^2DEs arises (as is the case in [7]). Additionally, it is possible to consider cell attributes other than only the maturity $\mathbf{x}$, thus turning to three or more dimensions in the input arguments. All these complications would necessarily lead to discrete systems of much higher dimensions, thereby making numerical solvers slower than is the case right now. At this point, it is not hard to imagine that by using RB methods, computation speed might be reduced to a very small percentage of the detailed solvers while still maintaining an acceptable error. Furthermore, the qualitative results presented in this thesis suggest a high aptitude of the model for MOR techniques, which would likely still be the case for more complex variations. Additional work has to be done here, both for the application of the introduced RB methods to the higher-dimensional models and the realization of numerical experiments.

Moreover, the experiments have suggested, based on evidence from the analysis how often certain parameters were chosen during the algorithm, that training grids used to represent the parameter domain during the Greedy search could be constructed more efficiently. Various alternatives to the equidistant grids used right now quickly come to mind and would have to be tested for the quality of the resulting reduced spaces.

Lastly, it has to be mentioned that the RB strategies introduced in this thesis are based on the same reduced space over the entire time domain. There exist approaches, for example in [4], that partition this domain and use dedicated reduced spaces for each time subinterval. This is meant and has shown to speed up the online computation time, while at the same time ensuring a manageable approximation error. For the problem at hand, this approach might be particularly of interest because of the discrete EPO administration times affecting the popula-

tion in the present model, thereby leading to the detailed solutions showing a very different behavior before and after an injection time. It is not hard to imagine that dedicated reduced spaces for each time interval are better suited to represent this behavior individually than one global space having to incorporate both. As a result, we might see reduced spaces that approximate the detailed solutions better whilst using fewer dimensions, which would in turn lead to decreased online simulation times.

Summing up, the application of the MOR techniques introduced here to similar models with higher dimensions appears to be the logical next step. We have seen evidence that these methods could significantly increase solver efficiency, all the while maintaining an acceptable error tolerance.

8 Appendix

8.1 Coefficient functions in the RK4 method

With the abbreviations $\kappa^k := \kappa(t^k, u, \mu)$, $\kappa^{k+1/2} := \kappa(t^k + \frac{\Delta t}{2}, u, \mu)$ and $\kappa^{k+1} := \kappa(t^{k+1}, u, \mu)$, the coefficient functions read:

$$\begin{aligned}\sigma_E^1(t^k) = \; & 1 + \tfrac{\Delta t}{6}\left(\kappa^{k+1} + 4\kappa^{k+1/2} + \kappa^k\right) \\ + \; & \tfrac{\Delta t^2}{6}\left(\kappa^{k+1}\kappa^{k+1/2} + \kappa^{k+1/2}\kappa^{k+1/2} + \kappa^{k+1/2}\kappa^k\right) \\ + \; & \tfrac{\Delta t^3}{12}\left(\kappa^{k+1}\kappa^{k+1/2}\kappa^{k+1/2} + \kappa^{k+1/2}\kappa^{k+1/2}\kappa^k\right) \\ + \; & \tfrac{\Delta t^4}{24}\left(\kappa^{k+1}\kappa^{k+1/2}\kappa^{k+1/2}\kappa^k\right)\end{aligned}$$

$$\begin{aligned}\sigma_E^2(t^k) = \; & -\Delta t - \tfrac{\Delta t^2}{6}\left(\kappa^{k+1} + 4\kappa^{k+1/2} + \kappa^k\right) \\ - \; & \tfrac{\Delta t^3}{6}\left(\kappa^{k+1}\kappa^{k+1/2} + \kappa^{k+1/2}\kappa^{k+1/2} + \kappa^{k+1/2}\kappa^k\right) \\ - \; & \tfrac{\Delta t^4}{24}\left(\kappa^{k+1}\kappa^{k+1/2}\kappa^{k+1/2} + 2\kappa^{k+1}\kappa^{k+1/2}\kappa^k + \kappa^{k+1/2}\kappa^{k+1/2}\kappa^k\right)\end{aligned}$$

$$\begin{aligned}\sigma_E^3(t^k) = \; & \tfrac{\Delta t^2}{2} + \tfrac{\Delta t^3}{12}\left(\kappa^{k+1} + 4\kappa^{k+1/2} + \kappa^k\right) \\ + \; & \tfrac{\Delta t^4}{24}\left(2\kappa^{k+1}\kappa^{k+1/2} + \kappa^{k+1}\kappa^k + \kappa^{k+1/2}\kappa^{k+1/2} + 2\kappa^{k+1/2}\kappa^k\right)\end{aligned}$$

$$\sigma_E^4(t^k) = -\tfrac{\Delta t^3}{6} - \tfrac{\Delta t^4}{24}\left(\kappa^{k+1} + 2\kappa^{k+1/2} + \kappa^k\right)$$

$$\sigma_E^5(t^k) = \tfrac{\Delta t^4}{24}$$

$$\begin{aligned}\sigma_z^1(t^k) = \;& 1 + \tfrac{\Delta t}{6}\left(\kappa^{k+1} + 2\kappa^{k+1/2}\right) \\ + \;& \tfrac{\Delta t^2}{12}\left(\kappa^{k+1}\kappa^{k+1/2} + \kappa^{k+1/2}\kappa^{k+1/2}\right) + \tfrac{\Delta t^3}{24}\left(\kappa^{k+1}\kappa^{k+1/2}\kappa^{k+1/2}\right)\end{aligned}$$

$$\begin{aligned}\sigma_z^2(t^k) = \;& -\tfrac{\Delta t}{2} - \tfrac{\Delta t^2}{12}\left(\kappa^{k+1} + 3\kappa^{k+1/2}\right) \\ - \;& \tfrac{\Delta t^3}{24}\left(2\kappa^{k+1}\kappa^{k+1/2} + \kappa^{k+1/2}\kappa^{k+1/2}\right)\end{aligned}$$

$$\sigma_z^3(t^k) = \tfrac{\Delta t^2}{6} + \tfrac{\Delta t^3}{24}\left(\kappa^{k+1} + 2\kappa^{k+1/2}\right)$$

$$\sigma_z^4(t^k) = -\tfrac{\Delta t^3}{24}$$

8.2 Acronyms

EPO Erythropoietin 2

FD Finite Difference 2

MOR Model Order Reduction 1

ODE Ordinary Differential Equation 2

PDE Partial Differential Equation 23

P^{2}DE Parametrized Partial Differential Equation 1

POD Proper Orthogonal Decomposition 2

RB Reduced-Basis 1

SVD Singular Value Decomposition 2

Bibliography

[1] M. Abramowitz, *Handbook of mathematical functions with formulas, graphs and mathematical tables.* Dover Publ., 1972

[2] W. Arendt, K. Urban, *Partielle Differentialgleichungen.* Spektrum Akademischer Verlag, 2010

[3] R. Denk, *Fouriertransformationen und SobolevrÃ́ume.* Lecture Notes, University of Constance, 2012

[4] M. Dihlmann, M. Drohnmann, B. Haasdonk, *Model reduction of parametrized evolution problems using the reduced basis method with adaptive time partitioning.* ESAIM: Mathematical Modelling and Numerical Analysis, 2011

[5] G. Dzuik, *Theorie und Numerik Partieller Differentialgleichungen.* De Gruyter, 2010

[6] G. Fischer, *Lineare Algebra.* Vieweg+Teubner, 2008

[7] D. Fürtinger, F. Kappel, S. Thijssen, N. Levin & P. Kotanko, *A model of erythropoiesis in adults with sufficient iron availability.* Journal of Mathematical Biology, 66:1209-1240, 2012

[8] M.A. Grepl, A.T. Patera, *A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations.* ESAIM: Mathematical Modelling and Numerical Analysis 39:157-181, 2005

[9] M. Gubisch, S. Volkwein, *Proper Orthogonal Decomposition for Linear-Quadratic Optimal Control.* nbn-resolving.de/urn:nbn:de:bsz:352-250378, 2013

[10] B. Haasdonk, M. Ohlberger, *Reduced basis method for finite volume approximations of parametrized linear evolution equations.* ESAIM: Mathematical Modelling and Numerical Analysis 42:277-302 2008

[11] M. Hermann, *Numerische Mathematik.* Oldenbourg, 2011

[12] M. Hinze, S. Volkwein, *Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition.* Comput. Optim. Appl. 39:319-347, 2008

[13] P. Holmes, J.L. Lumley, G. Berkooz, C.W. Rowley, *Turbulence Coherent Structures, Dynamical Systems and Symmetry.* Cambridge University Press, 2012

[14] K. Ito, F. Kappel, *Evolution equations and approximations.* World Scientific, Singapore, 2002

[15] K. Ito, S.S. Ravindran, *A reduced basis method for control problems governed by PDEs*, in *Control and Estimation of Distributed Parameter Systems.* Proceedings of the Internation Conference in Vorau, 126:153-168, BirkhÃd'user-Verlag, 1996

[16] F. Kappel, K. Zhang, *Approximation of linear age-structured population models using Legendre polynomials.* J. Math. Anal. Appl. 180:518-549, 1993

[17] J. Nocedal, S. Wright, *Numerical Optimization.* Springer, 2006

[18] A.T. Patera, G. Rozza, *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations*, MIT, 2006.

[19] J. Schropp, *Numerik Partieller Differentialgleichungen II.* Lecture Notes, University of Constance, 2012

[20] S. Volkwein, *Numerische Verfahren der restringierten Optimierung.* Lecture Notes, University of Constance, 2009.

[21] S. Volkwein, *Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling.* Lecture Notes, University of Konstanz, 2012

[22] D. Werner, *Funktionalanalysis.* Springer, 2007