Freie Universität Berlin

Numerical and Analytical Aspects of POD-Based Reduced-Order Modeling in Computational Fluid Dynamics

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

von

Swetlana Giere

Berlin
Juli 2016

1. Gutachter: Prof. Dr. Volker John, *Freie Universität Berlin und Weierstraß-Institut für Angewandte Analysis und Stochastik, Berlin*

2. Gutachter: Dr. Gabriel Barrenechea, *University of Strathclyde, Glasgow*

Datum der Disputation: 19. Oktober 2016

Abstract

This thesis studies projection-based reduced-order models (ROMs) in the context of computational fluid dynamics. Proper Orthogonal Decomposition (POD) is employed to compute the reduced-order basis from snapshots, which are assumed to represent the finite element solution of a partial differential equation. All investigations involve either convection-diffusion-reaction equation or the incompressible Navier–Stokes equations. The main contribution of the thesis can be divided into three parts.

Firstly, a Streamline-Upwind Petrov–Galerkin reduced-order model (SUPG-ROM) is investigated theoretically and numerically for convection-dominated convection-diffusion-reaction equations. Numerical analysis is utilized to propose the scaling of the stabilization parameter for the SUPG-ROM. Two approaches are used: One based on the underlying finite element discretization and the other one based on the POD truncation. The resulting SUPG-ROMs and the standard Galerkin ROM are studied numerically on several convection-dominated test problems aiming at answering several questions. One of the choices for the stabilization parameter is recommended.

Secondly, an alternative approach for the computation of the ROM initial condition is derived for problems, for which the standard approach, that is usually used in the literature, results in the ROM initial condition being polluted by spurious oscillations. The principal idea of the method consists in modifying the conventional ROM initial condition in a post-processing step by a filtering procedure. Numerical studies are performed in order to investigate the influence of the filtered initial condition on the ROM results. With respect to the minimum and maximum values of the ROM solution, which characterize the under- and overshoots, ROM results could be partly significantly improved compared to the results obtained with the standard ROM initial condition.

Thirdly, three velocity-pressure reduced-order models (vp-ROMs) for incompressible flows are investigated numerically. One method computes the ROM pressure solely based on the velocity POD modes, whereas the other two ROMs use pressure modes as well. One of the latter methods denoted by SM-ROM is developed within the framework of this dissertation. Moreover, the impact of the snapshot accuracy as well as of utilizing different linearization techniques on the ROM results is investigated numerically. Based on weakly divergence-free velocity snapshots, SM-ROM could reproduce the results of the finite element simulations in many cases better than the other vp-ROMs. Together with the fact that SM-ROM does not need any specification of additional pressure boundary conditions, which is required in the other methods, SM-ROM can be considered to be superior to other vp-ROMs for the computation of the ROM pressure.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Prof. Dr. Volker John, who gave me the chance to become a member of the research group ”Numerical Mathematics and Scientific Computing” of the Weierstrass Institute for Applied Analysis and Stochastics. I was preparing the diploma thesis when I first came in touch with the scientific publications of Prof. Dr. John without knowing him in person. It is always a big challenge to start working in a new area. Due to a clear and precise writing style of Prof. Dr. John, I truly enjoyed my first steps into the scientific world. After becoming a research associate, I had a privilege to be personally assisted by him when writing the scientific papers and this thesis. I am thankful for all the discussions we had, which helped me organize the thoughts and focus on the right questions, and for the support and understanding during difficult phases of working on this dissertation.

A special thank you goes to my colleague Dr. Alfonso Caiazzo, who helped me a lot with his expertise during the entire period of my stay in the research group. I consider myself lucky to have been able to collaborate with Prof. Traian Iliescu. It was very inspiring to work together during his stay in Berlin at the Weierstrass Institute and also during my visit of his research group at Virginia Tech. I am particularly grateful to David Wells, the former PhD student of Prof. Iliescu, for a warm welcome in Blacksburg and for the fruitful discussions during the preparation of the common paper.

I wish to acknowledge all my colleagues for the pleasant working atmosphere at the institute. Especially I would like to mention Ulrich Wilbrandt and Dr. Alfonso Caiazzo, with whom I shared the office and had a lot of interesting discussions, Dr. Nataliya Togobytska and Dr. Gonca Aki, who supported and encouraged me many times when it was needed. I would also like to extend my thanks to the secretary of the research group Marion Lawrenz for her valuable help in administrative and organizational matters.

I take this opportunity to express my deep appreciation to all my friends for sharing good and bad moments, which is of indispensable importance to me to keep the spiritual balance. I am deeply grateful to my beloved parents for the unconditional love and continuous support throughout my entire life. Last but not least, I would like to thank my adorable husband Lars and little son Max for the love, patience, support, and for the unforgettably wonderful and energizing moments we have together.

Contents

1. Introduction

1.1. Motivation

A great number of important dynamical processes in nature can be modeled by partial differential equations. However, the simulation of such equations by means of, e.g., finite element or finite volume methods can become computationally very expensive or for some practical problems in the applied sciences even not feasible. Especially, the numerical treatment of control and optimization problems in realistic engineering applications is often very challenging as repeated numerical simulations of large-scale dynamical systems are required.

One possible remedy is a simplification of the model from the physical point of view, which relies on the experience and intuition of the engineers or physicists. Another one is the so-called reduced-order modeling that is a mathematical approach serving to overcome high computational costs of the simulations. The underlying thesis is concerned with the latter technique. Its main idea is to approximate the large-scale problems by much smaller ones, which yield somewhat less accurate results but can be solved with a considerably less computational complexity. In the last decades a lot of effort has been made to develop various methods of model reduction. An overview of many currently existing approaches for the reduced-order modeling can be found, e.g, in [121].

For stationary linear state-space systems, some popular reduced-order approaches are the Moment Matching Approximation and the Balanced Truncation Method. The former method was proposed in [37, 43] and is based on projecting the dynamical system onto Krylov subspaces, which are computed by an Arnoldi- or Lanczos process. The latter approach, introduced in [148], is based on transforming the state-space system into a balanced form so that its controllability and observability Gramians, which are solutions to the two Lyapunov equations, become diagonal and equal.

For the sake of the model reduction of parametrized nonlinear problems, the most popular techniques are the Reduced Basis (RB) method and the Proper Orthogonal Decomposition (POD). Both approaches apply the Galerkin projection of the dynamical system onto a subspace, which is spanned by basis functions containing some relevant characteristics of the sought solution. The difference in both strategies consists in the computation of the so-called reduced or reduced-order basis of that subspace. The RB method was developed in [49, 110], see also [60] for an extensive overview of the progress and application fields. It is usually applied to build bases for stationary problems, in which the solution is sought for a large number of parameters. POD was first introduced in [103] in order to detect and analyze coherent structures in turbulent flows from experiments. It provides a low-dimensional basis computed from a known ensemble of experimental data or numerical solutions, the so-called snapshots, which optimally captures the dominant features of the data. POD is most commonly applied to build reduced-order bases for time-dependent problems. For the computation of the reduced-order basis of time-dependent parametrized problems, the so-called POD-greedy strategy was proposed for linear evolution equations in [57], the viscous nonlinear Burgers' equation in [106], and the Navier–Stokes equations in [145]. It is based on combining the POD method in time with the RB approach in parameter space. In this thesis, the main attention will be paid to the POD-based reduced-order models (POD-ROMs) for time-dependent problems.

POD models have been successfully applied in various areas. The probably most active fields within this context are optimization and control, see [141] for more details. Some examples for the utilization of the POD include design of practical, real-time feedback controllers for dynamical systems in [8], optimal control describing the laser surface hardening of steel in [64], inverse problems in [15], flow control in [20, 107], and calibration of models in option pricing in [120]. In optimization problems, expensive function evaluations lead to an enormous amount of computing time at each iteration step. Here, the reduced-order models (ROMs) can replace the dynamical system given by the partial differential equation in the objective function resulting in a suboptimal solution approach, which can substantially accelerate the optimization algorithm. At this point it is meaningful to clarify, which snapshots to employ for the computation of the reduced-order basis as it is by no means guaranteed that the reduced-order model based on the snapshots associated with a certain control function or a parameter value is able to approximate the probably somewhat different dynamical behavior of the system related to a different control function or parameter value. To overcome this difficulty, several approaches have been proposed in literature such as the Trust-Region and Optimality System POD methods in [6] and [90], respectively. Moreover, in [4] the POD-ROMs computed at different parameter values were interpolated to obtain a new POD-ROM that is valid also in the intermediate zone between the original parameter values. Other application areas of the reduced-order modeling based on POD are the parametrized fluid-structure interaction, e.g, see [14,31], and the uncertainty quantification for partial differential equations with parametrized random inputs, e.g., see [136].

1.2. Main Contributions

The underlying thesis is concerned with the POD-based reduced-order modeling in the context of computational fluid dynamics, for which the incompressible Navier–Stokes equations, governing the motion of numerous fluids, play the central role. A lot of research has been dedicated to this topic in the last decades, e.g., see [2,19,69,96,107,136, 144,145]. The main contribution of the dissertation consists of three parts. Firstly, a new reduced-order model for the computation of the pressure field is developed. Secondly, two versions of the stabilization parameter for a Streamline-Upwind Petrov–Galerkin reduced-order model are determined by means of the numerical analysis. Thirdly, an alternative approach for the computation of the ROM initial condition is derived for problems with polluted data. The motivation and description for these developments are presented in the following.

In many, probably even most published reports on reduced-order models based on the POD for incompressible flows, only velocity models are considered. Usually it is assumed that the velocity snapshots are divergence-free and due to the properties of the POD procedure also the velocity POD basis functions are divergence-free. It leads to the cancellation of the mixed velocity-pressure term and thus of the pressure contribution in the reduced-order model for the Navier–Stokes equations. However, from the practical point of view, the pressure is needed in many computational fluid dynamics applications, e.g., for the simulation of fluid-structure interaction problems and for the computation of relevant properties, such as drag and lift coefficients at solid bodies. In the literature, there exist several proposals for the computation of the pressure field in the framework of reduced-order modeling. One class of pressure ROMs consists in defining a ROM for the pressure that only uses the velocity POD modes [108, 138]. A second class of pressure ROMs employs pressure POD basis functions in addition to the velocity POD basis functions, e.g., see [2, 26, 136]. The pressure POD basis functions can be computed separately from the velocity POD basis functions (i.e., the decoupled approach) [2, 26, 108, 136, 138], or together with them (i.e., the coupled, monolithic approach) [19, 145]. In the former case, the computation of the reduced-order pressure can be considered as a post-processing step after having obtained the velocity solution. In this thesis, the decoupled approach will be considered. Three different pressure ROMs will be comprehensively numerically investigated. Two of the models proposed in [2, 108] utilize the pressure Poisson equation, whose derivation requires the pointwise divergence-free velocity field. However, this assumption is in general idealized. The third pressure ROM, referred later as SM-ROM, was first introduced in [26][1] and is part of the contribution of the underlying dissertation. Its derivation is based on a residual-based stabilization mechanism for the incompressible Navier–Stokes equations, which is a mathematically well understood method [21]. The advantage of SM-ROM over the two other studied pressure ROMs consists in the fact that its derivation requires the velocity snapshots to be only discretely divergence-free (but not pointwise), and it does not need any ad hoc treatment of external forces and pressure boundary conditions.

The application of most turbulent models for the simulation of turbulent flows with finite element methods requires a choice of some stabilization parameters, [75]. In the framework of POD-based reduced-order modeling, the stabilization parameters from the finite element method were used in the literature, like in [88], or an optimization problem for the determination of the parameters was solved, as in [19]. Certainly it is desirable to have some support for the choice of stabilization parameters coming from numerical analysis, as such parameters should be generally valid for a wide range of settings. To the best of the author's knowledge, the first contribution to this approach in the context of reduced-order modeling was provided within the framework of the thesis. The results were published in [44]. To avoid the effects of the velocity-pressure coupling and the nonlinearity present in the Navier–Stokes equations, a stabilized ROM for scalar convection-dominated convection-diffusion-reaction equation is investigated for this pur-

[1] Schyschlowa is the maiden name of the author, Swetlana Giere.

pose. The employed stabilization approach is the Streamline-Upwind Petrov–Galerkin (SUPG) method, which is one of the most popular stabilization schemes in the framework of finite element methods. As a result of the analytical considerations, two stabilization parameters are proposed. One of them is based on the finite element resolution and the other one is based on the POD spatial resolution. The ROMs combined with both versions of the stabilization parameter are extensively studied numerically on several test problems.

In the literature, the standard approach to determine the initial condition for a ROM consists in projecting the full-order initial condition, i.e., the finite element initial condition or interpolated initial condition of the continuous problem (if available), in the L^2 sense onto the POD basis. However, depending on the origin of the underlying snapshots, it can be observed that the ROM initial condition computed this way can be polluted by spurious oscillations. A good quality initial condition is essential for the numerical methods to produce accurate solutions. Therefore, it is desirable to be able to construct an initial condition that suppresses spurious oscillations as good as possible but still approximates well the full-order initial condition. For this purpose, an alternative approach for the computation of the ROM initial condition is proposed in this thesis. The principal idea consists in modifying the standard ROM initial condition in a post-processing step by a certain filtering procedure revived from the derivation of the Large Eddy Simulation [75], which is one of the most popular turbulence models. Numerical simulations of the convection-dominated convection-diffusion-reaction equation are performed to compare the effect of both versions of the ROM initial condition on the ROM results. Although it was motivated by the fluid dynamical applications, the proposed approach for the computation of the ROM initial condition can be extended to other types of problems with polluted data.

1.3. Outline

This thesis is organized as follows: Chapter 2 focuses on two partial differential equations, namely the convection-diffusion-reaction equation and the incompressible Navier–Stokes equations, which build a basis for the main investigations in terms of reduced-order modeling in the dissertation. In particular, the numerical methods are discussed, which are utilized to construct snapshots employed in the later chapters. Moreover, selected analytical results for the Streamline-Upwind Petrov–Galerkin method applied to the convection-diffusion-reaction equation are presented.

Chapter 3 is divided into two sections. In Section 3.1, the Proper Orthogonal Decomposition is motivated and derived for continuous and discrete settings. Furthermore, some practical aspects associated with the POD are addressed. Section 3.2 describes the Galerkin projection on the POD space resulting in a projection-based reduced-order model. The standard approach for the computation of the initial condition for the reduced-order model is supplemented by a new filtering procedure for the sake of suppressing possible oscillations for a certain type of problems. Moreover, the treatment of the boundary conditions and implementation issues are discussed.

Chapter 4 deals with the Streamline-Upwind Petrov–Galerkin reduced-order model for convection-dominated problems governed by the convection-diffusion-reaction equation. Numerical analysis is carried out in order to propose two different versions of the stabilization parameter. Subsequently, extensive numerical investigations are presented on three test problems, which aim to answer four different questions.

Chapter 5 is devoted to the POD-based reduced-order modeling of the incompressible Navier–Stokes equations. The reasoning for the cancellation of the pressure contribution from the standard Galerkin ROM under some typical assumptions is provided. The velocity ROM as well as three different reduced-order models for the computation of the pressure field are derived and discussed. Thereafter, comprehensive numerical studies of three different aspects are performed.

Finally, Chapter 6 summarizes the findings of this dissertation. Additionally, some open questions and proposals for further developments are outlined.

Definitions of the function spaces, norms, scalar products, and inequalities, which are most commonly utilized in this thesis, are presented in Appendix A.

2. Studied Partial Differential Equations

This chapter describes two partial differential equations, which will be employed for theoretical and numerical investigations within the framework of the thesis. In particular, some algorithms for the numerical solution of the equations with the Galerkin finite element method and with stabilized methods will be derived. Moreover, results of the error analysis necessary for the latter examinations will be presented.

The chapter consists of two sections and is structured as follows: Section 2.1 introduces the time-dependent scalar convection-diffusion-reaction equation and Section 2.2 concerns itself with the time-dependent incompressible Navier–Stokes equations.

2.1. Convection-Diffusion-Reaction Equation

Many processes in nature and industry such as transport of species or scalar physical quantities by a flow field, e.g., temperature or concentration, are modeled by the time-dependent scalar convection-diffusion-reaction equation which is given by

$$\begin{aligned} \partial_t u - \varepsilon\Delta u + \boldsymbol{b}\cdot\nabla u + cu &= f && \text{in} \quad (0,T]\times\Omega, \\ u &= 0 && \text{on} \quad [0,T]\times\Gamma, \\ u(0,\boldsymbol{x}) &= u^0(\boldsymbol{x}) && \text{in} \quad \Omega. \end{aligned} \tag{2.1}$$

Here, Ω is a bounded domain in $\mathbb{R}^d$, $d \in \{1,2,3\}$, with the boundary Γ, $\boldsymbol{b}(t,\boldsymbol{x})$ and $c(t,\boldsymbol{x})$ denote convection and reaction fields, respectively, $\varepsilon > 0$ is a constant diffusion coefficient, $u_0(\boldsymbol{x})$ is a given initial condition, and T is the length of the considered time interval.

A detailed discussion of the properties of the solution of (2.1) and of its numerical approximation can be found in [118], or in [131] for a shorter version.

2.1.1. Weak Formulation

Let $X = H_0^1(\Omega) = \big\{v \in H^1(\Omega) : \ v = 0 \text{ on } \Gamma\big\}$. The weak or also called variational formulation of the time-dependent convection-diffusion-reaction equation (2.1) is obtained by multiplying it with the test function $v \in X$ and integrating the equation over Ω. Finally, the weak form reads as follows: Find $u : [0,T] \to X$ such that

$$(\partial_t u, v) + a(u,v) = (f,v)\,, \quad \forall v \in X, \tag{2.2}$$

where $a(\cdot,\cdot)$ denotes a bilinear form in X defined by

$$a(u,v) = (\varepsilon\nabla u, \nabla v) + (\boldsymbol{b}\cdot\nabla u, v) + (cu, v)\,. \tag{2.3}$$

The first term on the right-hand side of (2.3) is obtained by using integration by parts and the Gaussian theorem as follows

$$-\int\limits_{\Omega} \varepsilon \Delta u v \, d\boldsymbol{x} = -\int\limits_{\Gamma} \varepsilon \nabla u \cdot \boldsymbol{n} v \, d\boldsymbol{s} + (\varepsilon \nabla u, \nabla v) \,. \tag{2.4}$$

The boundary term vanishes due to the prescribed homogeneous Dirichlet boundary condition in (2.1). To guarantee the coercivity of $a(\cdot,\cdot)$, an additional assumption on the reaction and convection fields has to be made, which is given in Lemma 2.1.

Lemma 2.1 (Coercivity of $a(\cdot,\cdot)$)**.** *Let $\boldsymbol{b}(t,\cdot)$, $\nabla \cdot \boldsymbol{b}(t,\cdot)$, $c(t,\cdot) \in L^\infty(\Omega)$ for almost all $t \in [0,T]$. If*

$$\left(c - \frac{1}{2}\nabla \cdot \boldsymbol{b}\right)(t,\boldsymbol{x}) \geq 0, \quad \forall (t,\boldsymbol{x}) \in [0,T] \times \Omega, \tag{2.5}$$

holds, then $a(\cdot,\cdot)$ is coercive, i.e., for all $v \in X$ one has

$$a(v,v) \geq \varepsilon \|\nabla v\|_0^2 = \varepsilon |v|_1^2.$$

Proof. Applying integration by parts and the product rule, one obtains

$$(\boldsymbol{b} \cdot \nabla v, v) = -\frac{1}{2}\left((\nabla \cdot \boldsymbol{b})\, v, v\right).$$

Inserting this relation into (2.3), setting $u = v$, and applying the assumption (2.5) yields

$$a(v,v) = (\varepsilon \nabla v, \nabla v) - \frac{1}{2}\left((\nabla \cdot \boldsymbol{b})\, v, v\right) + (cv, v) \geq \varepsilon \|\nabla v\|_0^2.$$

□

2.1.2. Galerkin Finite Element Method

Let $X_h \subset X$ denote a conforming N-dimensional finite element space spanned by piecewise polynomials of order $m \in \mathbb{N}$, i.e.,

$$X_h = \left\{ v_h \in X_h : v_h|_K \in P_m(K), \quad \forall K \in \mathcal{T}_h \right\},$$

and let $\{\varphi_{h,i}\}_{i=1}^N$ denote the finite element basis functions. Let the family of triangulations $\{\mathcal{T}_h\}$ of the domain Ω be shape-regular. Thus, the following local inverse inequality for finite element functions holds, e.g., see [29, Thm. 3.2.6],

$$\|v_h\|_{m,K} \leq \mu_{\text{inv}}\, h_K^{l-m} \|v_h\|_{l,K}\,, \quad \forall v_h \in X_h, \tag{2.6}$$

for $0 \leq l \leq m$, where h_K is the size (diameter) of the mesh cell $K \in \mathcal{T}_h$. Values of μ_{inv} for different situations can be found in [59]. They are usually of order one.

By replacing the space X in (2.2) by X_h, one obtains the time-continuous Galerkin finite element formulation of (2.2), which reads as follows: Find $u_h : (0,T] \to X_h$ such that

$$(\partial_t u_h, v_h) + a(u_h, v_h) = (f, v_h)\,, \quad \forall v_h \in X_h. \tag{2.7}$$

Equation (2.7) has to be equipped with an appropriate finite element approximation $u_h(0,\boldsymbol{x})$ of $u^0(\boldsymbol{x})$.

Equation (2.7) is still continuous in time. To discretize it in time, a temporal discretization scheme has to be applied. Here, a one-step θ-scheme will be employed.

Let Δt denote a fixed time step. Let u_h^n and f^n be the finite element solution and the right-hand side evaluated at $n\Delta t$, respectively. The fully discretized Galerkin finite element method reads as follows: For $n = 1, 2, \dots$ find $u_h^n \in X_h$ such that $\forall v_h \in X_h$

$$\begin{aligned}(u_h^n, v_h) + \Delta t\theta a(u_h^n, v_h) = {} & \left(u_h^{n-1}, v_h\right) - \Delta t(1-\theta)a(u_h^{n-1}, v_h) \\ & + \Delta t(1-\theta)\left(f^{n-1}, v_h\right) + \Delta t\theta\left(f^n, v_h\right).\end{aligned} \tag{2.8}$$

Parameter θ has to be chosen. In Table 2.1, some well-known one-step θ-schemes are listed.

Table 2.1.: Some one-step θ-schemes.

θ	Name of the scheme
0	Forward Euler scheme (FE)
1	Backward Euler scheme (BE)
$\frac{1}{2}$	Crank–Nicolson scheme (CN)

Problem (2.8) in matrix form is obtained by using the unique finite element representation of the solution

$$u_h^n = \sum_{i=1}^{N} u_{h,i}^n \varphi_{h,i}, \tag{2.9}$$

and by testing the equation with each finite element basis function separately. Hence, the system of linear equations has the form

$$\left(M_h + \Delta t\theta A_h^n\right)\underline{u}_h^n = \left(M_h - \Delta t(1-\theta)A_h^{n-1}\right)\underline{u}_h^{n-1} + \Delta t(1-\theta)\underline{f}_h^{n-1} + \Delta t\theta\underline{f}_h^n, \tag{2.10}$$

with $\underline{u}_h^n = \left(u_{h,1}^n, \dots, u_{h,N}^n\right)^T$ and

$$(M_h)_{ij} = \left(\varphi_{h,j}, \varphi_{h,i}\right), \quad i,j = 1,\dots,N, \tag{2.11}$$

$$(A_h^n)_{ij} = \left(\varepsilon\nabla\varphi_{h,j}, \nabla\varphi_{h,i}\right) + \left(\boldsymbol{b}^n \cdot \nabla\varphi_{h,j}, \varphi_{h,i}\right) + \left(c^n\varphi_{h,j}, \varphi_{h,i}\right), \quad i,j = 1,\dots,N, \tag{2.12}$$

$$f_{h,i}^n = \left(f^n, \varphi_{h,i}\right), \quad i = 1,\dots,N. \tag{2.13}$$

In many applications, the convection coefficient $\boldsymbol{b}$ has a much greater magnitude than the diffusion coefficient ε, i.e.,

$$\frac{|\boldsymbol{b}|}{\varepsilon} \gg 1.$$

Such problems are called convection-dominated problems. A characteristic feature of solutions of such problems is the presence of sharp layers. It is a well-known fact that

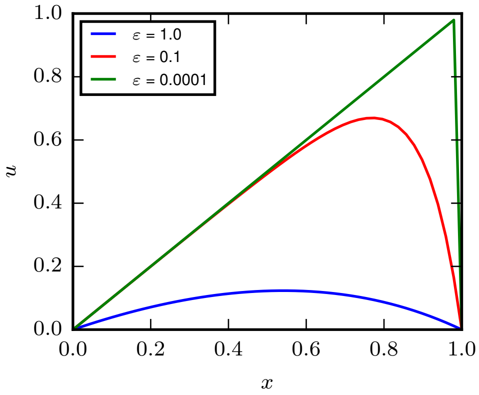

Figure 2.1.: Example 2.1: Solution of the continuous problem for different values of ε.

the Galerkin finite element method performs poorly for such problems as in general it is hard to resolve all the small scales which are important for the solution using practicable triangulations, particularly in higher dimensions. As a result, its solutions are strongly polluted by non-physical oscillations. To give a better understanding of the situation, it is enough to consider the steady-state version of (2.7), i.e., neglecting the first term on the left-hand side, in one dimension, see Example 2.1.

Example 2.1. Consider a one-dimensional boundary value problem given by

$$-\varepsilon u'' + u' = 1 \quad \text{on} \quad (0,1), \quad u(0) = u(1) = 0, \tag{2.14}$$

with the solution

$$u(x) = x - \frac{\exp\left(-\frac{1-x}{\varepsilon}\right) - \exp\left(-\frac{1}{\varepsilon}\right)}{1 - \exp\left(-\frac{1}{\varepsilon}\right)}.$$

The solution becomes steeper at the right boundary when choosing a smaller value of ε, see Figure 2.1.

In the steady-state case, it is known that with appropriate regularity assumptions and fulfillment of the conditions of Lemma 2.1, the application of the Lemma of Céa [23, p.55] yields the error estimate

$$\|u - u_h\|_X \leq C \frac{\max\{\|\boldsymbol{b}\|_{L^\infty(\Omega)}, \|c\|_{L^\infty(\Omega)}\}}{\varepsilon} \inf_{v_h \in X_h} \|u - v_h\|_X, \quad C \in \mathbb{R},$$

where u and u_h are the solutions of the corresponding steady-state versions of (2.1) and (2.7), respectively. In the convection-dominated case, the first factor of the estimate becomes very large. Therefore, it cannot be expected that the Galerkin finite element method gives a satisfactory numerical solution, unless the dimension of X_h is so high that the second factor in the estimate gets extremely small.

Indeed, such unsatisfactory behavior of the Galerkin method can be very well observed for problem (2.14) in the convection-dominated regime. Figure 2.2 shows the Galerkin

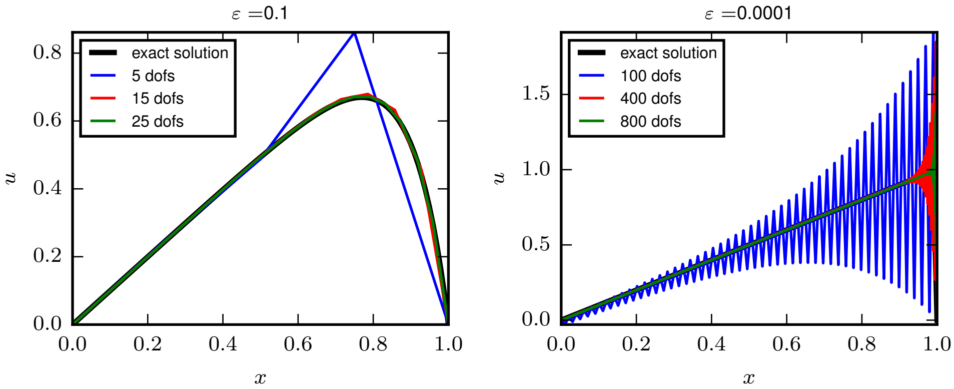

Figure 2.2.: Example 2.1: Galerkin finite element solution for $\varepsilon = 0.1$ (left) and $\varepsilon = 0.0001$ (right) using different numbers of degrees of freedom (dofs).

finite element solution for $\varepsilon = 0.1$ and $\varepsilon = 0.0001$. For the former value of ε it is sufficient to use around 15 degrees of freedom in order to obtain a good approximation of the solution, whereas for the latter value of ε (convection-dominated case), one can see strong spurious oscillations over the whole domain for 100 degrees of freedom, but even choosing 800 degrees of freedom does not yield a satisfactory approximation of the solution at the boundary $x = 1$. ◁

2.1.3. Streamline-Upwind Petrov–Galerkin Method

To overcome the poor performance of the Galerkin finite element method for convection-dominated problems, a so-called stabilized discretization is necessary. During the last few decades, a variety of such discretizations have been proposed, e.g., see [11, 76, 84] for reviews and numerical comparisons of many of these proposals. However, the question of finding a perfect discretization, i.e., a discretization which gives solutions with sharp layers and without spurious oscillations, is still open. One of the most popular stabilized finite element methods is the Streamline-Upwind Petrov–Galerkin (SUPG) scheme, also known as Streamline-Diffusion Finite Element Method (SDFEM), proposed in [24, 66]. Solutions computed with this method possess usually steep layers but also contain spurious oscillations in a vicinity of the layers.

The SUPG method adds weighted residuals, i.e., the terms

$$\sum_{K \in \mathcal{T}_h} \left(\mathcal{R}(u_h), \delta_{h,K} \boldsymbol{b} \cdot \nabla v_h\right)_K , \tag{2.15}$$

to the Galerkin finite element method (2.7). Here, $\mathcal{R}(u)$ denotes the residual of the first equation in (2.1), i.e.,

$$\mathcal{R}(u) = \partial_t u - \varepsilon \Delta u + \boldsymbol{b} \cdot \nabla u + cu - f \quad \text{in} \quad L^2(K) \quad \forall K \in \mathcal{T}_h.$$

The parameter $\delta_{h,K}$ for $K \in \mathcal{T}_h$ is a local stabilization parameter that has to be chosen. Note that the summation in (2.15) is necessary as in general $\Delta u_h \notin L^2(\Omega)$ but $\Delta u_h \in L^2(K)$ for each $K \in \mathcal{T}_h$.

Inserting (2.15) into (2.7) and rearranging the terms results in the time-continuous SUPG method which reads as follows: Find $u_h : (0,T] \to X_h$ such that

$$\begin{aligned}(\partial_t u_h, v_h) + a_{\mathrm{SUPG},h}(u_h, v_h) + \sum_{K\in\mathcal{T}_h} \delta_{h,K} \left(\partial_t u_h, \boldsymbol{b}\cdot\nabla v_h\right)_K \\ = (f, v_h) + \sum_{K\in\mathcal{T}_h} \delta_{h,K} \left(f, \boldsymbol{b}\cdot\nabla v_h\right)_K, \quad \forall v_h \in X_h,\end{aligned} \tag{2.16}$$

where the bilinear form $a_{\mathrm{SUPG},h}(\cdot,\cdot)$ is defined by

$$\begin{aligned}a_{\mathrm{SUPG},h}(u_h, v_h) = {} & (\varepsilon\nabla u_h, \nabla v_h) + (\boldsymbol{b}\cdot\nabla u_h, v_h) + (cu_h, v_h) \\ & + \sum_{K\in\mathcal{T}_h} \delta_{h,K} \left(-\varepsilon\Delta u_h + \boldsymbol{b}\cdot\nabla u_h + cu_h, \boldsymbol{b}\cdot\nabla v_h\right)_K,\end{aligned} \tag{2.17}$$

for all $u_h, v_h \in X_h$.

Remark 2.1. The name "SUPG" comes from the fact that the method can be considered as the Petrov–Galerkin method with the test space

$$\operatorname{span}\left\{ v + \sum_{K\in\mathcal{T}_h} \delta_{h,K} \boldsymbol{b}\cdot\nabla v \right\}.$$

The name "SDFEM" comes from the fact that the method introduces artificial diffusion only in the streamline direction $\boldsymbol{b}\cdot\nabla v$. ◁

Let Δt denote a fixed time step and let u_h^n be the finite element solution at $t_n = n\Delta t$. With the help of a one-step θ-scheme, the fully discretized SUPG method reads as follows: For $n = 1, 2, \ldots$ find $u_h^n \in X_h$ such that $\forall v_h \in X_h$

$$\begin{aligned}\left(u_h^n - u_h^{n-1}, v_h\right) + \Delta t\theta\, a_{\mathrm{SUPG},h}\left(u_h^n, v_h\right) = {} & - \sum_{K\in\mathcal{T}_h} \delta_{h,K} \left(u_h^n - u_h^{n-1}, \boldsymbol{b}^n\cdot\nabla v_h\right)_K \\ & + \Delta t(1-\theta) \left[\left(f^{n-1}, v_h\right) + \sum_{K\in\mathcal{T}_h} \delta_{h,K} \left(f^{n-1}, \boldsymbol{b}^n\cdot\nabla v_h\right)_K \right] \\ & + \Delta t\theta \left[\left(f^n, v_h\right) + \sum_{K\in\mathcal{T}_h} \delta_{h,K} \left(f^n, \boldsymbol{b}^n\cdot\nabla v_h\right)_K \right] \\ & - \Delta t(1-\theta)\, a_{\mathrm{SUPG},h}\left(u_h^{n-1}, v_h\right),\end{aligned} \tag{2.18}$$

where $u_h^0(\boldsymbol{x}) = u_h(0, \boldsymbol{x})$. In Table 2.1, some popular schemes are presented for different choices of the parameter θ.

The bilinear form $a_{\mathrm{SUPG},h}(\cdot,\cdot)$ is coercive under some more strict conditions than in Lemma 2.1 for the fields $\boldsymbol{b}$ and c, and under the condition that the stabilization parameters $\{\delta_{h,K}\}$ are appropriately bounded from above, e.g., see [131, Lemma 10.3].

Lemma 2.2 (Coercivity of $a_{\text{SUPG},h}(\cdot,\cdot)$). *Let the following condition be satisfied: There is a constant $\mu_0 > 0$ such that*

$$0 < \mu_0 \leq \mu(t,x) = \left(c - \frac{1}{2}\nabla\cdot\boldsymbol{b}\right)(t,\boldsymbol{x}), \quad \forall(t,\boldsymbol{x}) \in [0,T]\times\Omega. \tag{2.19}$$

If the SUPG stabilization parameters are chosen such that

$$\delta_{h,K} \leq \min\left\{\frac{\mu_0}{2\left\|c\right\|^2_{L^\infty(K)}}, \frac{h^2_{h,K}}{2\varepsilon\mu^2_{inv}}\right\}, \tag{2.20}$$

then the bilinear form $a_{\text{SUPG},h}(\cdot,\cdot)$ satisfies $\forall v_h \in X_h$

$$a_{\text{SUPG},h}(v_h,v_h) \geq \frac{1}{2}|||v_h|||^2_{\text{SUPG},h}, \tag{2.21}$$

with the so-called energy norm

$$|||v_h|||_{\text{SUPG},h} = \left(\varepsilon\left|v_h\right|^2_1 + \sum_{K\in\mathcal{T}_h}\delta_{h,K}\left\|\boldsymbol{b}\cdot\nabla v_h\right\|^2_{0,K} + \mu_0\left\|v_h\right\|^2_0\right)^{1/2}. \tag{2.22}$$

For piecewise linear finite elements, the second factor in the condition (2.20) can be omitted.

Proof. By using integration by parts and assumption (2.19), one obtains

$$\begin{aligned}
a_{\text{SUPG},h}(v_h,v_h) = {} & (\varepsilon\nabla v_h,\nabla v_h) + (\boldsymbol{b}\cdot\nabla v_h, v_h) + (cv_h,v_h) \\
& + \sum_{K\in\mathcal{T}_h}\delta_{h,K}\left(-\varepsilon\Delta v_h + \boldsymbol{b}\cdot\nabla v_h + cv_h, \boldsymbol{b}\cdot\nabla v_h\right)_K \\
\geq {} & \varepsilon|v_h|^2_1 + \mu_0\|v_h\|^2_0 + \sum_{K\in\mathcal{T}_h}\delta_{h,K}\|\boldsymbol{b}\cdot\nabla v_h\|^2_{0,K} \\
& + \sum_{K\in\mathcal{T}_h}\delta_{h,K}\left(-\varepsilon\Delta v_h + cv_h, \boldsymbol{b}\cdot\nabla v_h\right)_K \\
= {} & |||v_h|||^2_{\text{SUPG},h} + \sum_{K\in\mathcal{T}_h}\delta_{h,K}\left(-\varepsilon\Delta v_h + cv_h, \boldsymbol{b}\cdot\nabla v_h\right)_K.
\end{aligned} \tag{2.23}$$

The Cauchy–Schwarz inequality, Young's inequality, the inverse estimate (2.6), and assumption (2.20) yield the estimate

$$\begin{aligned}
&\left|\sum_{K\in\mathcal{T}_h}\delta_{h,K}\left(-\varepsilon\Delta v_h + cv_h, \boldsymbol{b}\cdot\nabla v_h\right)_K\right| \\
&\leq \left|\sum_{K\in\mathcal{T}_h}\delta_{h,K}\left(-\varepsilon\Delta v_h, \boldsymbol{b}\cdot\nabla v_h\right)_K\right| + \left|\sum_{K\in\mathcal{T}_h}\delta_{h,K}\left(cv_h, \boldsymbol{b}\cdot\nabla v_h\right)_K\right| \\
&\leq \sum_{K\in\mathcal{T}_h}\delta_{h,K}\varepsilon\left\|\Delta v_h\right\|_{0,K}\left\|\boldsymbol{b}\cdot\nabla v_h\right\|_{0,K} + \sum_{K\in\mathcal{T}_h}\delta_{h,K}\left\|cv_h\right\|_{0,K}\left\|\boldsymbol{b}\cdot\nabla v_h\right\|_{0,K} \\
&\leq \sum_{K\in\mathcal{T}_h}\delta_{h,K}\varepsilon^2\left\|\Delta v_h\right\|_{0,K}^2 + \sum_{K\in\mathcal{T}_h}\delta_{h,K}\left\|cv_h\right\|_{0,K}^2 + \frac{1}{2}\sum_{K\in\mathcal{T}_h}\delta_{h,K}\left\|\boldsymbol{b}\cdot\nabla v_h\right\|_{0,K}^2 \\
&\leq \sum_{K\in\mathcal{T}_h}\delta_{h,K}\varepsilon^2\mu_{\text{inv}}^2 h^{-2}\left|v_h\right|_{1,K}^2 + \sum_{K\in\mathcal{T}_h}\delta_{h,K}\left\|c\right\|_{L^\infty(K)}^2\left\|v_h\right\|_{0,K}^2 \\
&\quad + \frac{1}{2}\sum_{K\in\mathcal{T}_h}\delta_{h,K}\left\|\boldsymbol{b}\cdot\nabla v_h\right\|_{0,K}^2 \\
&\leq \frac{1}{2}\varepsilon\left|v_h\right|_1^2 + \frac{1}{2}\mu_0\left\|v_h\right\|_0^2 + \frac{1}{2}\sum_{K\in\mathcal{T}_h}\delta_{h,K}\left\|\boldsymbol{b}\cdot\nabla v_h\right\|_{0,K}^2 = \frac{1}{2}|||v_h|||_{\text{SUPG},h}^2.
\end{aligned}$$

Therefore, coercivity (2.21) of $a_{\text{SUPG},h}(\cdot,\cdot)$ holds. In the case of piecewise linear finite elements, the term $\left\|\Delta v_h\right\|_{0,K}^2$ vanishes. As a result, the second constraint in (2.20) can be omitted. □

The asymptotic value of the stabilization parameter $\delta_{h,K}$ for the convection-dominated problems is well known for steady-state problems from the finite element error analysis, e.g., see [118]. This stabilization parameter depends on the local mesh width $h_{h,K}$ and can be expressed as

$$\delta_{h,K} = C_0 h_K, \tag{2.24}$$

where C_0 is a constant to be chosen.

The situation is not completely clear for time-dependent problems. In [84], an advanced numerical study using various stabilization methods for convection-dominated problems such as SUPG method, spurious oscillations at layers diminishing (SOLD) methods, local projection stabilization (LPS) schemes, finite element methods - flux-corrected transport schemes (see Section 2.1.5) with a wide range of parameters is presented. The best results were obtained with the FEM-FCT schemes showing a good ratio of accuracy and efficiency. Several choices for the SUPG stabilization parameters proposed in the literature (e.g., see [30,38,86,102]) depending on the length of the time step were studied. The investigations showed that the SUPG method with this choice of stabilization parameters yielded poor results. Indeed, the stabilization parameters depending on the length of the time step do not reflect the fact that the reason for needing a stabilized discretization is the appearance of layers in the solution, which are spatial

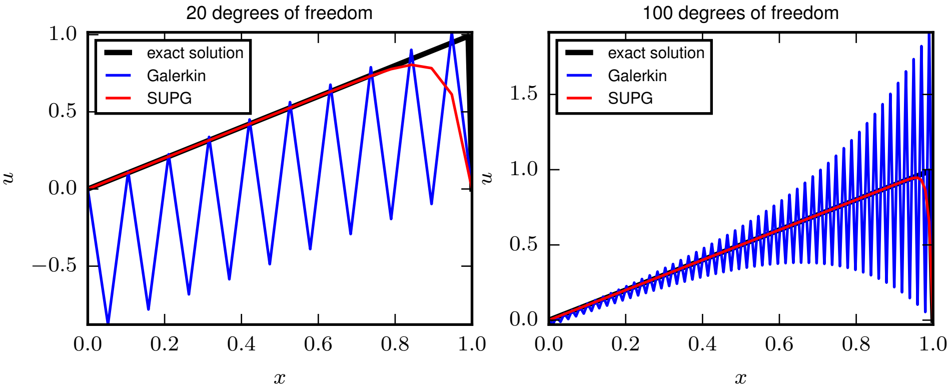

Figure 2.3.: Example 2.2: SUPG and Galerkin finite element solutions for 20 (left) and 100 (right) degrees of freedom.

features. Therefore, it is more meaningful to choose $\delta_{h,K}$ depending on the mesh width like in the steady-state case.

Finite element error analysis for the problem (2.18) can be found in [81] and will be discussed in Section 2.1.4. For the general case of time-dependent coefficients of the problem, an optimal error estimate could be proven for $\delta_h = \mathcal{O}(\Delta t)$. For the situation of steady-state convection and reaction coefficients, an optimal error estimate for $\delta_{h,K} = \mathcal{O}(h_K)$ could be derived in [81]. Numerical studies in [81] reveal that the choice of $\delta_{h,K} = \mathcal{O}(h_K)$ is more appropriate also in the general case.

Example 2.2. Consider the test case described in Example 2.1 in the convection-dominated regime with $\varepsilon = 0.0001$. In Figure 2.3, the exact, the Galerkin, and the SUPG solutions are shown for 20 and 100 degrees of freedom. While the Galerkin finite element method produces solutions globally polluted with spurious oscillations all over the domain even for large numbers of the degrees of freedom (see also Figure 2.2), with the SUPG method it is enough to use only 20 degrees of freedom to achieve an almost exact approximation of the solution away from the boundary layer at $x = 1$. ◁

2.1.4. Error Estimates for SUPG Method

In this section, the main analytical results for the SUPG finite element method of the convection-dominated convection-diffusion-reaction equation will be presented.

The numerical analysis for the steady-state problems is very well studied and understood, see [118] for a detailed discussion. In the convection-dominated case, the scaling analysis yields the stabilization parameter $\delta_{h,K} = C_0 h_K$ and the global error estimate has the form

$$\|u - u_h\|_0 + h^{1/2} \left(\sum_{K\in\mathcal{T}_h} \|\boldsymbol{b}\cdot\nabla(u-u_h)\|_{0,K}^2 \right)^{1/2} \le C h^{m+1/2} \left|u\right|_{m+1}, \tag{2.25}$$

where $h = \max_{K\in\mathcal{T}_h} h_K$. The finite element error in the L^2 norm is of order 1/2 less than optimal and the L^2 error of the derivative in the streamline direction is optimal.

Numerical analysis for the SUPG method applied to the time-dependent convection-dominated convection-diffusion-reaction equation (2.18) was carried out in [81]. Using the backward Euler scheme, analysis based on the standard energy arguments proposes SUPG parameters that depend on the length of the time step. The following result presents error estimates for this choice of the SUPG parameter.

Theorem 2.1 (Error estimate for $\delta_{h,K}$ depending on the length of the time step)**.** *Suppose $\boldsymbol{b} \in L^\infty(0,T;L^\infty(\Omega))$, $\nabla\cdot\boldsymbol{b}$, $c \in L^\infty(0,T;L^\infty(\Omega))$ for the coefficients in (2.2), and u, $\partial_t u \in L^\infty(0,T;H^{m+1}(\Omega))$, $\partial_t^2 u \in L^2(0,T;H^1(\Omega))$ for the solution of (2.2). Let the stabilization parameters fulfill (2.20), $\delta_{h,K} > 0$, and*

$$\delta_{h,K} < \frac{\Delta t}{4} \tag{2.26}$$

for all $K \in \mathcal{T}_h$. Denote $\delta_{\max} = \max_{K\in\mathcal{T}_h}\delta_{h,K}$ and $\delta_{\min} = \min_{K\in\mathcal{T}_h}\delta_{h,K}$. Then the following error estimates hold

$$\|u_h^n - u^n\|_0 \le C\left(h^{m+1} + \Delta t + h^{m-1}\delta_{\max}^{1/2}(h^2+h+\varepsilon) + \frac{h^{m+1}}{\delta_{\min}^{1/2}} + \left\|\pi_h u_0 - u_h^0\right\|_0\right)$$

and

$$\begin{aligned}\left(\Delta t\sum_{j=1}^{n}\left\|\left|u_h^j - u^j\right|\right\|_{\mathrm{SUPG},h}^2\right)^{1/2} \le{}& C\Bigg[h^m(\varepsilon^{1/2} + \delta_{\max}^{1/2} + h) + \Delta t \\ &+ h^{m-1}\delta_{\max}^{1/2}(h^2+h+\varepsilon) + \frac{h^{m+1}}{\delta_{\min}^{1/2}} + \left\|\pi_h u_0 - u_h^0\right\|_0\Bigg],\end{aligned}$$

where $u^n = u(t_n)$, the constants C depend on u, $\partial_t u$, $\partial_t^2 u$, $\boldsymbol{b}$, $\nabla\cdot\boldsymbol{b}$, c, and π_h is the elliptic projection from X into X_h defined by $(\nabla(u-\pi_h u), \nabla v_h) = 0$ for all $v_h \in X_h$.

A similar result as in Theorem 2.1 can also be obtained for the condition

$$\delta_{h,K} < \frac{\sigma(\Delta t)h_K}{\|\boldsymbol{b}\|_{L^\infty(K)}\mu_{\mathrm{inv}}}$$

with a function $\sigma(\Delta t)$ satisfying $0 < \sigma(\Delta t) \le \frac{1}{4}$ for all $K \in \mathcal{T}_h$ instead of (2.26). In this case, one obtains error estimates as in Theorem 2.1 but with an additional factor $(1+2\sigma^2(k))^n$.

As discussed in Section 2.1.4, the stabilization parameters depending on the length of the time step do not seem to be a correct choice as the difficulty of not being able to resolve the layers vanishes on sufficiently fine meshes but not for sufficiently small time steps. In [81], optimal error estimates for the backward Euler and Crank–Nicolson methods are also derived for stabilization parameters which do not depend on Δt and are proportional to the mesh width as follows:

$$\delta_{h,K} = \min\left\{\hat{\delta}_{h,K}, 1\right\}, \tag{2.27}$$

with

$$\hat{\delta}_{h,K} = \frac{h}{4\mu_{\mathrm{inv}} \|\boldsymbol{b}\|_{L^\infty(\Omega)}} \min \left\{ \frac{1}{2}, \frac{\mu_0^{1/2}}{4}, \frac{\mu_0}{4\|c\|_{L^\infty(\Omega)}}, \frac{\mu_0^{1/2}}{2\|c\|_{L^\infty(\Omega)}}, \frac{\mu_0^{1/2}}{\|c\|_{L^\infty(\Omega)}^{1/2}}, \|\boldsymbol{b}\|_{L^\infty(\Omega)}^{1/2} \right\}.$$

However, the estimates could be derived for a simplified setting, i.e., for $\boldsymbol{b}$, c, and μ being time-independent, for the uniform mesh with width h, and stabilization parameters equal for all mesh cells ($\delta_{h,K} = \delta_h$ for all $K \in \mathcal{T}_h$).

In the course of the analysis, a formally steady-state version of problem (2.1) is used. Let $\Pi_h u(t) \in X_h$ for $t \in [0,T]$ be $\forall v_h \in X_h$ the solution of

$$a_{\mathrm{SUPG},h}(\Pi_h u(t), v_h) = (f(t) - u_t(t), v_h) + \delta_h \left(f(t) - u_t(t), \boldsymbol{b} \cdot \nabla v_h\right),$$

where $u(t)$ is the solution of the corresponding continuous equation. The use of $\Pi_h u$ is necessary to introduce some useful estimates for the analysis. Next, error estimates from [81] for the stabilization parameter defined in (2.27) are presented.

Theorem 2.2 (Error estimate with $\delta_h = \mathcal{O}(h)$ for backward Euler method)**.** *Let* $t_n = T < \infty$, *let* u, $\partial_t u \in L^\infty(0,T; H^{m+1}(\Omega))$ *and* $\partial_t^2 \Pi_h u$, $\partial_t^3 u$, $\partial_t^3 \Pi_h u \in L^2(0,T; L^2(\Omega))$. *Let the stabilization parameter be defined as in* (2.27). *Then the following error estimate holds*

$$\|u^n - u_h^n\|_0^2 + \Delta t \sum_{j=1}^{n} \left|\left|\left| u^j - u_h^j \right|\right|\right|_{\mathrm{SUPG},h}^2 \leq C(h^{2m+1} + \Delta t^2), \tag{2.28}$$

where the constant C *depends on* $\boldsymbol{b}$, c, u, $\Pi_h u$, *and* μ_{inv}.

Theorem 2.3 (Error estimate with $\delta_h = \mathcal{O}(h)$ for Crank–Nicolson method)**.** *Let* $t_n = T < \infty$, *let* u, $\partial_t u \in L^\infty(0,T; H^{m+1}(\Omega))$, $\partial_t^2 u \in L^2(0,T; H^{m+1}(\Omega))$ *and* $\partial_t^3 \Pi_h u$, $\partial_t^4 u$, $\partial_t^4 \Pi_h u \in L^2(0,T; L^2(\Omega))$. *Let the stabilization parameter be defined as in* (2.27). *Then the following error estimate holds*

$$\|u^n - u_h^n\|_0^2 + \Delta t \sum_{j=1}^{n} \left|\left|\left| \frac{u^j - u^{j-1}}{2} - \frac{u_h^j - u_h^{j-1}}{2} \right|\right|\right|_{\mathrm{SUPG},h}^2 \leq C(h^{2m+1} + \Delta t^4), \tag{2.29}$$

where the constant C *depends on* $\boldsymbol{b}$, c, u, $\Pi_h u, \mu_{inv}$, *and linearly on* T.

Summarizing the results from Theorem 2.2 and Theorem 2.3, the error estimates can be formulated alternatively as follows

$$(\Delta t)^{1/2} \left(\sum_{n=1}^{N} \|u^n - u_h^n\|_0 + \sum_{n=1}^{N} |||u^n - u_h^n|||_{\mathrm{SUPG},h} \right) \leq C(h^{m+1/2} + (\Delta t)^k), \tag{2.30}$$

where $h = \max_{h_K \in \mathcal{T}_h} h_K$ and k is the order of the temporal discretization.

It is assumed that the space X_h satisfies the following local approximation property: For each $u \in X \cap H^{m+1}(\Omega)$ there exists a function $\hat{u}_h \in X_h$ such that

$$\|u - \hat{u}_h\|_{0,K} + h_K \|\nabla(u - \hat{u}_h)\|_{0,K} + h_T^2 \|\Delta(u - \hat{u}_h)\|_{0,K} \leq C h_K^{m+1} \|u\|_{m+1,K} \tag{2.31}$$

for all $K \in \mathcal{T}_h$. This property is given, for example, for Lagrange finite elements on mesh cells, which allow an affine mapping to a reference mesh cell.

In the course of the thesis, the Laplacian of the error for a simplified case of the uniform triangulations, i.e., $h_K = h$ for all $K \in \mathcal{T}_h$, will be needed. It can be obtained using the local approximation property (2.31), the inverse estimate (2.6), and the estimate (2.30) as follows

$$\begin{aligned}
&(\Delta t)^{1/2} \sum_{n=1}^{N} \sum_{K\in\mathcal{T}_h} \|\Delta(u^n - u_h^n)\|_{0,K} \\
&\le (\Delta t)^{1/2} \sum_{n=1}^{N} \left(\sum_{K\in\mathcal{T}_h} \|\Delta(u^n - \hat{u}_h^n)\|_{0,K} + \|\Delta(\hat{u}_h^n - u_h^n)\|_{0,K} \right) \\
&\le C(\Delta t)^{1/2} \sum_{n=1}^{N} \left(\sum_{K\in\mathcal{T}_h} h^{m-1} \|u^n\|_{m+1,K} + h^{-1} \|\nabla(\hat{u}_h^n - u_h^n)\|_{0,K} \right) \\
&\le C(\Delta t)^{1/2} \sum_{n=1}^{N} \sum_{K\in\mathcal{T}_h} \left(h^{m-1} + h^{-1} \|\nabla(u^n - \hat{u}_h^n)\|_{0,K} + h^{-1} \|\nabla(u^n - u_h^n)\|_{0,K} \right) \\
&\le C(\Delta t)^{1/2} \left(h^{m-1} + \varepsilon^{-1/2} h^{m-1/2} + \varepsilon^{-1/2} h^{-1} (\Delta t)^k \right).
\end{aligned} \tag{2.32}$$

2.1.5. Flux-Corrected Transport Schemes

In industrial applications, it is of great importance that the numerical solution follows the constraints dictated by physics, e.g., approximated densities, temperatures or concentrations should remain non-negative. Flux-corrected transport schemes represent a methodology that aims to obtain numerical solutions without non-physical oscillations.

The flux-corrected transport schemes have two basic goals:

1. Receive the solution in time by a low-order scheme that incorporates enough numerical diffusion to suppress under- and overshoots.

2. Modify the solution by using the so-called anti-diffusive fluxes limited in such a way that no new maxima or minima can arise and existing extrema cannot grow.

In the most stabilization approaches, some terms are added to the Galerkin finite element formulation (2.7). Finite element method flux-corrected transport (FEM-FCT) schemes, see [91–94, 105], work on the algebraic level, they modify the system matrix and the right-hand side of the Galerkin finite element formulation (2.10). Initially, the FEM-FCT schemes were developed for the transport equation, i.e., equations of type (2.1) with $\varepsilon = c = f = 0$. A detailed description of the schemes including the diffusion and reaction coefficients, and the right-hand side is given in [84], which is the basis of the underlying exposure.

If the maximum principle holds for the continuous equation, it is crucial not to loose the discrete counterpart of the property in the course of discretization. This way, undershoots and overshoots can be avoided. It can be achieved if the system matrix of the dicretized system is a so-called M-matrix.

Remark 2.2. A tridiagonal matrix B is a M-matrix if the following sufficient conditions hold [58, Theorem 84.3]:

- all diagonal entries of B are positive, i.e., $(B)_{ii} > 0,\ \forall i$,
- all off-diagonal entries of B are non-positive, i.e., $(B)_{ij} \leq 0,\ \forall j \neq i$,
- B is diagonally dominant, i.e., $|(B)_{ii}| \geq \sum\limits_j |(B)_{ij}|,\ \forall i$, and it holds $|(B)_{ii}| > \sum\limits_j |(B)_{ij}|$ for at least one i.

◁

The matrices in (2.10) do not satisfy the properties of the M-matrix and therefore have to be modified. The mass matrix M_h has to be approximated by the so-called lumped mass matrix M_L defined by

$$(M_L)_{ij} = \begin{cases} \sum\limits_{k=1}^{N} (M_h)_{ik}\,, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases} \tag{2.33}$$

where $i, j = 1, \ldots, N$ and N is the dimension of the finite element basis. Moreover, all positive off-diagonal entries of A_h^n in (2.10) have to be eliminated. It is accomplished by adding an artificial diffusion operator D^n defined by

$$D_{ij}^n = \begin{cases} \min\{0, -\left(A_h^n\right)_{ij}, -\left(A_h^n\right)_{ji}\}, & \text{if } i \neq j, \\ -\sum\limits_{k=1,k\neq i}^{N} D_{ik}^n, & \text{if } i = j, \end{cases} \tag{2.34}$$

to A_h^n and storing the result as L^n, i.e., $L^n = A_h^n + D^n$. By construction, the row sums of D^n are zero.

By replacing M_h and A_h^n by M_L and L^n, respectively, in (2.10), one obtains the algebraic representation of a stable low-order scheme

$$\left(M_L + \Delta t\theta L^n\right) \underline{u}_h^n = \left(M_L - \Delta t(1-\theta)L^{n-1}\right) \underline{u}_h^{n-1} + \Delta t(1-\theta)\underline{f}_h^{n-1} + \Delta t\theta \underline{f}_h^n, \tag{2.35}$$

which is the first goal of the FEM-FCT schemes. Its solution does not show spurious oscillations, however layers will be smeared because the operator on the left-hand side is too diffusive.

To achieve the second goal of the FEM-FCT schemes, i.e., to make the equation less diffusive while the spurious oscillations are still suppressed, the right-hand side of (2.35) has to be modified resulting in the system

$$\begin{aligned}\left(M_L + \Delta t\theta L^n\right) \underline{u}_h^n = {} & \left(M_L - \Delta t(1-\theta)L^{n-1}\right) \underline{u}_h^{n-1} + \Delta t(1-\theta)\underline{f}_h^{n-1} + \Delta t\theta \underline{f}_h^n \\ & + \underline{f}^*(\underline{u}_h^n, \underline{u}_h^{n-1}).\end{aligned} \tag{2.36}$$

In order to derive the representation of the correction term $\underline{f}^*(\underline{u}_h^n, \underline{u}_h^{n-1})$, consider the difference between the residuals of systems (2.35) and (2.10)

$$\begin{aligned}
\underline{r} &= (M_L + \Delta t\theta L^n - (M_h + \Delta t\theta A_h^n))\,\underline{u}_h^n \\
&\quad - \left(M_L - \Delta t(1-\theta)L^{n-1} - \left(M_h - \Delta t(1-\theta)A_h^{n-1}\right)\underline{u}_h^{n-1}\right) \\
&= (M_L - M_h)\left(\underline{u}_h^n - \underline{u}_h^{n-1}\right) + \Delta t\left(\theta D^n \underline{u}_h^n + (1-\theta)D^{n-1}\underline{u}_h^{n-1}\right).
\end{aligned}$$

Since $M_L - M_h$ and D are by construction symmetric matrices with zero row sums, the vector $\underline{r}$ can be decomposed into so-called skew-symmetric internodal fluxes $r_{ij},\ i,j \in \{1,\ldots,N\}$ as follows

$$\begin{aligned}
r_i = \sum_{j=1}^N r_{ij} = \sum_{j=1}^N \Big[&(M_h)_{ij}\left(u_{h,i}^n - u_{h,j}^n\right) - (M_h)_{ij}\left(u_{h,i}^{n-1} - u_{h,j}^{n-1}\right) \\
&- \Delta t\theta D_{ij}^n\left(u_{h,i}^n - u_{h,j}^n\right) - \Delta t(1-\theta)D_{ij}^{n-1}\left(u_{h,i}^{n-1} - u_{h,j}^{n-1}\right)\Big],
\end{aligned} \tag{2.37}$$

where r_i and $u_{h,i}^n$, $i = 1,\ldots,N$, are the components of vectors $\underline{r}$ and $\underline{u}_h^n$, respectively. For more details for the derivation of the fluxes r_{ij}, e.g., see [91].

The ansatz for the correction term in (2.36) has the form

$$\underline{f}^*(\underline{u}_h^n, \underline{u}_h^{n-1}) = \sum_{j=1}^N \alpha_{ij} r_{ij},\ i = 1,\ldots,N, \tag{2.38}$$

with the weights $\alpha_{ij} \in [0,1]$ to be computed.

Remark 2.3. By construction, the high-order discretization (2.10) is recovered if all weights α_{ij} are equal to 1 and the corresponding low-order system (2.35) is recovered when all weights are equal to 0. ◁

There are linear and nonlinear versions of FEM-FCT schemes. Here, only the nonlinear FEM-FCT scheme will be discussed as it will be used in the numerical studies. The description of the linear FEM-FCT scheme can be found in, e.g., [84,91].

The nonlinear FEM-FCT scheme computes an auxiliary explicit low-order solution $\tilde{\underline{u}}_h$ at the time $t_n - \theta\Delta t$ which is needed to guarantee the fulfillment of the maximum principle. Hence, in case of the backward Euler scheme $\tilde{\underline{u}}_h = \underline{u}_h^{n-1}$. In case the Crank-Nicolson scheme is employed, the auxiliary solution is obtained by solving (2.35) with the forward Euler scheme and the time step $\Delta t/2$, i.e.,

$$\tilde{\underline{u}}_h = \underline{u}_h^{n-1} - \frac{\Delta t}{2} M_L^{-1}\left(L^{n-1}\underline{u}_h^{n-1} - \underline{f}_h^{n-1}\right). \tag{2.39}$$

As $\tilde{\underline{u}}$ is computed with an explicit method, the size of the time step has to satisfy a CFL-like condition to ensure stability, e.g., see [91].

The low-order auxiliary solution $\tilde{\underline{u}}_h$ is used for the computation of the weights α_{ij}. It contributes to the decision making in which regions the additional diffusion in (2.36) can be eliminated resulting in the weights α_{ij} close to 1 and in which regions this diffusion is needed setting the weights α_{ij} close to 0.

Before computing the weights, it is recommended to perform the so-called pre-limiting (see [91, 94]) of the fluxes. The fluxes r_{ij} having the same sign as $\tilde{u}_j - \tilde{u}_i$ may cause problems as they flatten solution profiles instead of steepening them where needed. Therefore, such fluxes have to be canceled, i.e., set

$$r_{ij} = 0 \quad \text{if} \quad r_{ij}(\tilde{u}_j - \tilde{u}_j) > 0.$$

The weights α_{ij} in (2.38) are computed using Zalesk's algorithm [146], which is motivated and discussed in detail, e.g., in [91].

The algorithm proceeds as follows:

1. Compute the sums of positive and negative fluxes

$$P_i^+ = \sum_{j=1, j\neq i}^{N} \max\{0, r_{ij}\}, \quad P_i^- = \sum_{j=1, j\neq i}^{N} \min\{0, r_{ij}\},$$

2. Compute the distance to a local extremum of the auxiliary solution

$$Q_i^+ = \max\left\{0, \max_{j=1,\dots,N, j\neq i} (\tilde{u}_{h,j} - \tilde{u}_{h,i})\right\}, \quad Q_i^- = \min\left\{0, \max_{j=1,\dots,N, j\neq i} (\tilde{u}_{h,j} - \tilde{u}_{h,i})\right\},$$

3. Compute the nodal correction factors

$$R_i^+ = \max\left\{1, \frac{(M_L)_{ii}\, Q_i^+}{\Delta t P_i^+}\right\}, \quad R_i^- = \min\left\{1, \frac{(M_L)_{ii}\, Q_i^-}{\Delta t P_i^-}\right\},$$

4. Limit the fluxes r_{ij} and r_{ji} in a symmetric fashion

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\}, & \text{if } r_{ij} > 0, \\ \min\{R_i^-, R_j^+\}, & \text{otherwise,} \end{cases}$$

where $i, j = 1, \dots, N$.

2.2. Incompressible Navier–Stokes Equations

The motion of fluids such as water, oil, and air are governed by the general Navier–Stokes equations. However, numerous engineering applications involve fluids changing their density only to a lesser extent throughout space and time. Such fluids can be described by a simplified but still very important variation of the equations, the so-called incompressible Navier–Stokes equations.

Let $\Omega \subset \mathbb{R}^d$, $d \in \{2,3\}$, denote an open, bounded domain with the boundary Γ and T the length of the time interval. The time-dependent incompressible Navier–Stokes equations read in the dimensionless form as

$$\partial_t \boldsymbol{u} - \nu \Delta \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in} \quad (0,T] \times \Omega, \tag{2.40}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in} \quad [0,T] \times \Omega, \tag{2.41}$$

where $\boldsymbol{u}(t,\boldsymbol{x})$ and $p(t,\boldsymbol{x})$ denote the fluid velocity and pressure fields, respectively, $\boldsymbol{f}(t,\boldsymbol{x})$ is the given body force per unit mass, and $\nu = Re^{-1}$ denotes the dimensionless viscosity with Re being the Reynolds number. In what follows, bold symbols denote vector-valued quantities of dimension d.

Equation (2.40) is called the momentum equation. It follows from the conservation of linear momentum (Newton's second law of motion for fluids) stating that the rate of change of the linear momentum must be equal to the net force (combination of the internal and external forces) acting on a collection of particles. Equation (2.41) is called the continuity or mass equation. It is derived from the conservation of mass (general conservation law) stating that the rate of change of mass in an arbitrary volume $\mathcal{V} \subset \Omega$ must be equal to the flux of mass across the boundary of $\mathcal{V}$. Formulation (2.41) represents an incompressibility constraint for the velocity. For a detailed derivation of the Navier–Stokes equations, e.g., see [75, 95].

The Navier–Stokes equations (2.40) and (2.41) have to be equipped with appropriate boundary and initial conditions in order to become a well-posed problem. These issues will be discussed in Section 2.2.1.

Remark 2.4. Using the divergence constraint from (2.41), i.e., $\nabla \cdot \boldsymbol{u} = 0$, an equivalent formulation of the convective term in the momentum equation (2.40) can be obtained with the help of the equality

$$\nabla \cdot (\boldsymbol{u}\boldsymbol{v}^T) = (\nabla \cdot \boldsymbol{v})\boldsymbol{u} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{u}.$$

Setting $\boldsymbol{v} = \boldsymbol{u}$ yields the divergence form of the convective term

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nabla \cdot (\boldsymbol{u}\boldsymbol{u}^T). \tag{2.42}$$

Note that the above reformulations are in general no longer equivalent in the case of the discretized Navier–Stokes equations as the discrete velocity is generally not divergence-free. ◁

Remark 2.5. There are two main challenges when carrying out the mathematical analysis and the numerical simulations of the Navier–Stokes equations: First, the coupling of velocity and pressure, and second, the nonlinearity of the convective term. The first difficulty is characterized by the special way the velocity and the pressure fields are coupled, namely the absence of pressure in the continuity equation (2.41). Practically, the continuity equation can be considered as an incompressibility constraint for the velocity field. The pressure field in the momentum equation (2.40) is often referred to be the Lagrangian multiplier enforcing the incompressibility condition. This particular type of coupling is called saddle point problem. ◁

Remark 2.6. From the point of view of numerical simulations, one has to differ between the laminar and turbulent flows. There exists no exact definition of these notions. In practice, a flow is considered to be laminar if all its structures can be resolved on a reasonable grid. Otherwise, it is turbulent. The latter type of flows is much more challenging in the sense of simulations as so-called turbulence models are required to obtain a meaningful solution, see [73, 98, 111]. $\triangleleft$

Remark 2.7. In the special case when the body force, the velocity and pressure fields do not depend on time, the time derivative $\partial_t \boldsymbol{u}$ vanishes and one obtains the stationary Navier–Stokes equations. If, in addition, the fluid moves very slowly, i.e., the viscous transport dominates the convective transport, the convective term in the momentum equation can be neglected. As the result, one gets the so-called Stokes equations

$$\begin{aligned} -\nu \Delta \boldsymbol{u} + \nabla p &= \boldsymbol{f} \quad \text{in} \quad \Omega, \\ \nabla \cdot \boldsymbol{u} &= 0 \quad \text{in} \quad \Omega. \end{aligned} \tag{2.43}$$

$\triangleleft$

The focus of the presentation in this section lies on the numerical methods for the solution of the Navier–Stokes equations (2.40)-(2.41). For the results on existence and uniqueness of solutions, the reader is referred to [40, 45, 129]. Moreover, an extensive compilation of existing results for the Navier–Stokes equations including the uniqueness and existence of the solutions, error analysis and numerical methods can be found in [75]. The latter reference lays the foundation for the presentation in this section.

2.2.1. Initial and Boundary Conditions

The Navier–Stokes equations (2.40) and (2.41) are partial differential equations of first order in time and second order in space. Therefore, it is necessary to equip them with an initial condition at $t = 0$ and with boundary conditions on the boundary Γ of the domain Ω. It is important that the compatibility condition is fulfilled between the initial velocity condition and the limit of the boundary conditions for $t \to 0$, $t > 0$.

Unlike the time-dependent convection-diffusion-reaction equation discussed in Section 2.1, the initial condition for the Navier–Stokes equations cannot be chosen arbitrary just satisfying the required regularity. Namely, the initial velocity field $\boldsymbol{u}(0, \boldsymbol{x}) = \boldsymbol{u}^0(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$, has to be divergence-free in some sense on $\Omega \cup \Gamma$, e.g., see [50].

For incompressible flows, different kinds of boundary conditions can be prescribed. Here, only the boundary conditions used in the course of the thesis will be discussed. For a more detailed presentation, the reader is referred to, e.g., [50, 75, 115, 135].

Dirichlet Boundary Conditions

A widely used boundary condition for the incompressible flows is the so-called Dirichlet boundary condition, also called the essential boundary condition. It describes the velocity field on a part of the boundary denoted by Γ_{D} with $\Gamma_{\mathrm{D}} \subseteq \Gamma$ and reads as

$$\boldsymbol{u}(t, \boldsymbol{x}) = \boldsymbol{g}_{\mathrm{D}}(t, \boldsymbol{x}) \quad \text{in} \quad (0, T] \times \Gamma_{\mathrm{D}}. \tag{2.44}$$

This kind of boundary condition is usually used to prescribe the inflow into Ω or the outflow from Ω.

In the special case of $\boldsymbol{g}_{\mathrm{D}}(t,\boldsymbol{x}) = \boldsymbol{0}$, the boundary condition is called the no-slip or homogeneous boundary condition, and the corresponding boundary is denoted by Γ_0 with $\Gamma_0 \subseteq \Gamma_{\mathrm{D}}$. This condition states that the velocity field does not penetrate the wall, i.e., $\boldsymbol{u} \cdot \boldsymbol{n} = 0$, and does not slip along the wall, i.e., $\boldsymbol{u} \cdot \boldsymbol{t}_i = 0$, $i \in \{1,2\}$ for $\mathbb{R}^3$, where $\boldsymbol{n}$ and $\boldsymbol{t}_i$ are the unit normal and the tangential vectors, respectively, on the boundary Γ_0, which build an orthonormal system of vectors.

If the Dirichlet boundary condition is prescribed on the whole boundary of Ω, i.e., $\Gamma_{\mathrm{D}} = \Gamma$, then two issues have to be taken into account. The first one consists in the fact that the pressure field can only be determined up to an additive constant. To fix its value, an additional condition has to be introduced. Usually, the condition

$$\int\limits_{\Omega} p(t,\boldsymbol{x})\, d\boldsymbol{x} = 0, \quad t \in (0,T], \tag{2.45}$$

is used, which states that the integral mean value of the pressure should vanish. The second issue represents the compatibility condition for the prescribed function $\boldsymbol{g}_{\mathrm{D}}$ on the boundary Γ. With (2.41) and integration by parts, it follows that

$$0 = \int\limits_{\Omega} \nabla \cdot \boldsymbol{u}(t,\boldsymbol{x})\, d\boldsymbol{x} = \int\limits_{\Gamma} (\boldsymbol{u} \cdot \boldsymbol{n})(t,\boldsymbol{s})\, d\boldsymbol{s} = \int\limits_{\Gamma} (\boldsymbol{g}_{\mathrm{D}} \cdot \boldsymbol{n})(t,\boldsymbol{s})\, d\boldsymbol{s}, \quad \forall t \in (0,T].$$

Natural Boundary Conditions

This boundary condition is usually used to model the outflow from the bounded domain Ω in the numerical simulations and is proven to be well suited for (essentially) parallel domains, see [61, 135]. The part of the boundary, where it is prescribed, is denoted by Γ_{out} with $\Gamma_{\mathrm{out}} \subset \Gamma$ and the condition has the form

$$(\nu \nabla \boldsymbol{u} - p\mathbb{I})\, \boldsymbol{n} = \boldsymbol{0} \quad \text{on} \quad (0,T] \times \Gamma_{\mathrm{out}}, \tag{2.46}$$

where $\boldsymbol{n}$ denotes the unit normal vector on Γ_{out}. When building the weak formulation of the Navier–Stokes equations, the left-hand side of (2.46) appears in the integral on the boundary. With the condition (2.46), the corresponding integral vanishes on Γ_{out}. For that reason this condition is also called the natural (as it naturally appears in the variational formulation) or "do-nothing" (as it does not prescribe anything at Γ_{out}) boundary condition. The presence of the pressure in the natural boundary condition circumvents the problem of the non-unique pressure field, see [115], i.e., for $\Gamma_{\mathrm{out}} \neq \emptyset$ there is no need to employ the condition (2.45). Even if it is a widely used outflow boundary condition, it can cause some undesirable behavior of the numerical solution, e.g., for problems with more than one outlet (see [115, 135] for more details). Furthermore, the natural boundary condition (2.46) can cause numerical instabilities in the presence of a back flow at Γ_{out}. In [22], a modification of the boundary condition is proposed to prevent the instability.

2.2.2. Pressure Poisson Equation

In this section, the equation for the pressure, the so-called pressure Poisson equation, will be derived and some aspects of the correct boundary conditions will be presented. An extensive discussion of this topic can be found in [50, Section 3], [75]. The pressure Poisson equation is obtained by taking the divergence of the momentum equation (2.40) and by applying the continuity constraint (2.41) to it. It has to be assumed that the velocity $\boldsymbol{u}$ and the pressure p functions are sufficiently smooth up to the boundary.

By taking the divergence operator of (2.40), the equation becomes

$$\Delta p = \nabla \cdot [\nu \Delta \boldsymbol{u} + \boldsymbol{f} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \partial_t \boldsymbol{u}] \quad \text{in} \quad (0,T] \times \Omega. \tag{2.47}$$

Assuming sufficient regularity, it holds $\nabla \cdot \partial_t \boldsymbol{u} = \partial_t(\nabla \cdot \boldsymbol{u}) = 0$ in $(0,T]$. Moreover, one can invoke the rotational form of the Laplacian

$$\Delta \boldsymbol{u} = \nabla(\nabla \cdot \boldsymbol{u}) - \nabla \times \nabla \times \boldsymbol{u},$$

which yields

$$\nabla \cdot \Delta \boldsymbol{u} = \nabla \cdot \nabla(\nabla \cdot \boldsymbol{u}) - \nabla \cdot (\nabla \times \nabla \times \boldsymbol{u}) = 0.$$

Finally, the pressure Poisson equation has the form

$$\Delta p = \nabla \cdot [\boldsymbol{f} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}] \quad \text{in} \quad (0,T] \times \Omega. \tag{2.48}$$

To ensure the well-posedness of (2.48), the equation has to be equipped with appropriate boundary conditions. Consider the part of the boundary where the velocity in the normal direction is given, e.g., which is true for Dirichlet boundary conditions (2.44) on $\Gamma_{\mathrm{D}} \subseteq \Gamma$. To obtain the correct boundary condition for the pressure on Γ_{D}, the normal component of the momentum equation (2.40) restricted to Γ_{D} has to be taken. It results in the Neumann boundary condition for the pressure

$$\nabla p \cdot \boldsymbol{n} = (\nu \Delta \boldsymbol{u} + \boldsymbol{f} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \partial_t \boldsymbol{u}) \cdot \boldsymbol{n} \quad \text{on} \quad (0,T] \times \Gamma_{\mathrm{D}}, \tag{2.49}$$

where $\boldsymbol{n} \cdot \partial_t \boldsymbol{u} = \partial_t(\boldsymbol{n} \cdot \boldsymbol{g}_{\mathrm{D}})$ can be used for the last term. In the special case of the homogeneous boundary condition for the velocity, the boundary condition for the pressure reads

$$\nabla p \cdot \boldsymbol{n} = (\nu \Delta \boldsymbol{u} + \boldsymbol{f}) \cdot \boldsymbol{n} \quad \text{on} \quad (0,T] \times \Gamma_{\mathrm{D}}. \tag{2.50}$$

If the Dirichlet boundary condition for the velocity is prescribed on the whole boundary, i.e., $\Gamma_{\mathrm{D}} = \Gamma$, then the solution of the pressure Poisson equation with (2.49) is unique only up to an additive constant. To avoid it, one can prescribe the pressure at any point in $\bar{\Omega}$ or set $\int_\Omega p \, d\boldsymbol{x} = 0$. Moreover, in this case it can be shown that for $t \in (0,T]$ also the tangential components of the momentum equation (2.40) are automatically satisfied (see [50, Section 3.8.2] for further details).

If not the normal velocity but the normal traction is specified on the boundary Γ (or on the part of it), i.e.,

$$(\nu \nabla \boldsymbol{u} - p\mathbb{I})\boldsymbol{n} = \boldsymbol{w} \quad \text{on} \quad (0,T] \times \tilde{\Gamma},$$

where $\tilde{\Gamma} \subseteq \Gamma$ and $\boldsymbol{w}$ is the force applied by the boundary to the fluid, then it is inherited by the pressure Poisson equation but it becomes a Dirichlet boundary condition for the pressure. Consequently, one obtains the boundary condition

$$p = \nu \boldsymbol{n} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{n} - \boldsymbol{n} \cdot \boldsymbol{w} \quad \text{on} \quad (0,T] \times \tilde{\Gamma}.$$

However, it is often observed that the viscous part of (2.2.2) becomes small, see [50, Section 3.8.2], such that the following expression

$$p = -\boldsymbol{n} \cdot \boldsymbol{w} \quad \text{on} \quad (0,T] \times \tilde{\Gamma} \tag{2.51}$$

is usually used.

Remark 2.8. In the special case of the natural boundary condition for the velocity (2.46), the pressure boundary condition on the boundary Γ_{out} becomes the homogeneous Dirichlet boundary condition, i.e., $p = 0$ on Γ_{out}. ◁

Note that the pressure Poisson equation (2.48) is defined only for $t > 0$. For $t = 0$, it can be derived in a similar way as (2.48) but one has to be cautious when computing the pressure boundary condition as the velocity boundary condition is only prescribed for $t > 0$ and the initial velocity field $\boldsymbol{u}(0, \boldsymbol{x}) = \boldsymbol{u}^0(\boldsymbol{x})$ is divergence-free in some sense, see Section 2.2.1. Under certain conditions, one can ensure the well-posedness of the problem, see [50, 62], [75, Chapter 7].

2.2.3. Time Discretization - Saddle Point Problem

In order to perform numerical simulations of a flow, the Navier–Stokes equations need to be discretized in space and in time. First, the semi-discretization in time will be discussed. There is a number of time-stepping schemes available leading to coupled problems for velocity and pressure, the so-called saddle point problems, or to decoupled problems, where velocity and pressure are computed by separate equations. In this section, several popular time-stepping schemes known from the treatment of ordinary differential equations, which result in a saddle point problem, will be presented .

In what follows, let Δt_n be the length of the current time step from t_{n-1} to t_n, and let $\boldsymbol{u}^n$, p^n and $\boldsymbol{f}^n$ denote the evaluations of the functions at the time t_n.

One of the most popular time-stepping schemes for the incompressible flows is the one-step θ-scheme, which is a special case of the general fractional-step θ-schemes, e.g., see [75, Chapter 7]. For the Navier–Stokes equations (2.40)-(2.41), it reads: Given $\boldsymbol{u}^0(\boldsymbol{x}) = \boldsymbol{u}(0, \boldsymbol{x})$, find $(\boldsymbol{u}^n, p^n)$ such that

$$\begin{aligned}
\boldsymbol{u}^n &+ \theta \Delta t_n \left[-\nu \Delta \boldsymbol{u}^n + (\boldsymbol{u}^n \cdot \nabla) \boldsymbol{u}^n \right] + \Delta t_n \nabla p^n \\
&= \boldsymbol{u}^{n-1} - (1-\theta) \Delta t_n \left[-\nu \Delta \boldsymbol{u}^{n-1} + (\boldsymbol{u}^{n-1} \cdot \nabla) \boldsymbol{u}^{n-1} \right] \\
&\quad + (1-\theta) \Delta t_n \boldsymbol{f}^{n-1} + \theta \Delta t_n \boldsymbol{f}^n, \\
\nabla \cdot \boldsymbol{u}^n &= 0,
\end{aligned} \tag{2.52}$$

where the parameter θ has to be chosen, see Table 2.1 for some widely used choices resulting in the explicit or forward Euler, implicit or backward Euler and Crank–Nicolson methods.

Both the explicit and the implicit Euler methods are of first order of accuracy. The resulting ordinary equation is stiff with respect to time. Explicit time-stepping schemes, such as the forward Euler method, require very small time steps satisfying the CFL condition introduced in [33] to ensure stable simulations. It leads to very high computational costs and therefore the explicit Euler method is in general not recommended for the simulation of the Navier–Stokes equations. The backward Euler method is quite popular but the results are often too dissipative and therefore too inaccurate. The Crank–Nicolson method produces more accurate results as it is of second order of accuracy. However, it is only A-stable and sometimes it suffers from instabilities. For more details about the advantages and disadvantages of the one-step θ-methods applied to the simulations of the Navier–Stokes equations, e.g., see [75, 135].

Note that the scaling of the pressure term in the first equation of (2.52) is different to the one applied to the viscous and convective terms for $\theta \neq 1$. It is an often used formulation in the literature which does not require the initial condition for the pressure $p^0 = p(0, \boldsymbol{x})$. Its derivation can be conducted using the variational formulation in two steps. In the first step, the one-step θ-scheme is applied to the variational formulation only for the velocity in a weakly divergence-free space, a subspace of V. In the second step, the variational formulation is extended to $V \times Q$, where the pressure term $\Delta t_n(\nabla \cdot \boldsymbol{v}, p^n)$ appears as the Lagrange multiplier inforcing the incompressibility condition.

It is also possible to scale the pressure term in the same way as the viscous and convective terms yielding the following formulation: Given $(\boldsymbol{u}^0, p^0)$, find $(\boldsymbol{u}^n, p^n)$

$$\begin{aligned}
\boldsymbol{u}^n &+ \theta \Delta t_n \left[-\nu \Delta \boldsymbol{u}^n + (\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n + \nabla p^n\right] \\
&= \boldsymbol{u}^{n-1} - (1-\theta)\Delta t_n \left[-\nu \Delta \boldsymbol{u}^{n-1} + (\boldsymbol{u}^{n-1} \cdot \nabla)\boldsymbol{u}^{n-1} + \nabla p^{n-1}\right] \\
&\quad + (1-\theta)\Delta t_n \boldsymbol{f}^{n-1} + \theta \Delta t_n \boldsymbol{f}^n, \\
\nabla \cdot \boldsymbol{u}^n &= 0.
\end{aligned} \tag{2.53}$$

In this version of the method, besides the initial velocity also the initial pressure is needed. In practice, one can obtain it by storing the solution of the simulation of the developed flow, for which an arbitrary initial condition for the velocity (if none is given) and the pressure was used. Up to the best of the author's knowledge, there exits no extensive numerical comparison of both options (2.52) and (2.53) in the literature. For the numerical computations in this thesis, version (2.52) will be applied.

2.2.4. Time Discretization - Decoupled Problem

One of the difficulties mentioned in Remark 2.5 for the numerical simulation of incompressible flows is the special coupling of the velocity and the pressure resulting in a saddle point problem to be solved. Driven by the motivation to overcome this difficulty, the so-called projection methods or fractional step methods were first proposed in [28, 132]. The biggest advantage of the projection methods lies in the fact that they are very efficient as one only needs to solve decoupled elliptic equations for the pressure and the velocity in each time step of the simulation. Fractional step methods consist of three classes: the pressure-correction schemes, the velocity-correction schemes, and the consistent splitting methods. A detailed overview of the analysis and implementation of this type of schemes can be found in [53]. In what follows, the main focus will be on the pressure-correction schemes, see also [75, Chapter 7].

For simplicity of presentation, the Navier–Stokes equations (2.40)-(2.41) will be considered, which are equipped with homogeneous Dirichlet boundary conditions for the velocity field, i.e., $\boldsymbol{u} = \boldsymbol{0}$ on $(0, T] \times \Gamma_{\mathrm{D}} = \Gamma$. Let Δt_n denote the length of the time step with $\Delta t_n = t_n - t_{n-1}$ for $n \geq 1$.

The starting point is the approximation of the time derivative of the velocity field $\boldsymbol{u}$ at time t_n by the qth-order backward difference formula (BDFq)

$$\partial_t \boldsymbol{u}^n \approx \frac{1}{\Delta t_n} \left(\tau_q \boldsymbol{u}^n + \sum_{j=0}^{q-1} \tau_{q-1-j} \boldsymbol{u}^{n-j} \right), \quad \sum_{j=0}^{q} \tau_j = 0. \tag{2.54}$$

Especially, it holds

$$\partial_t \boldsymbol{u}^n \approx \begin{cases} \boldsymbol{u}^n - \boldsymbol{u}^{n-1}, & \text{if} \quad q = 1, \\ \frac{3}{2}\boldsymbol{u}^n - 2\boldsymbol{u}^{n-1} + \frac{1}{2}\boldsymbol{u}^{n-2}, & \text{if} \quad q = 2. \end{cases} \tag{2.55}$$

Let $\hat{p}^n \in L^2(\Omega)$ be given. In particular, assuming that the pressure field p is smooth, $\hat{p}^n$ is an extrapolation of p evaluated at t^n of order s with

$$\hat{p}^n = \begin{cases} 0, & \text{if} \quad s = 0, \\ p^{n-1}, & \text{if} \quad s = 1, \\ 2p^{n-1} - p^{n-2}, & \text{if} \quad s = 2. \end{cases} \tag{2.56}$$

Pressure-correction schemes consist of two steps. The first step is the equation for the intermediate velocity $\tilde{\boldsymbol{u}}$ which arises from the momentum equation (2.40). Here, the pressure is treated explicitly for $s > 0$ and ignored for $s = 0$. The first step reads

$$\begin{aligned} \frac{1}{\Delta t_n} \left(\tau_q \tilde{\boldsymbol{u}}^n + \sum_{j=0}^{q-1} \tau_{q-1-j} \boldsymbol{u}^{n-j} \right) - \nu \Delta \tilde{\boldsymbol{u}}^n + (\tilde{\boldsymbol{u}}^n \cdot \nabla)\, \tilde{\boldsymbol{u}}^n &= \boldsymbol{f}^n - \nabla \hat{p}^n \quad && \text{in} \quad \Omega, \\ \tilde{\boldsymbol{u}}^n &= \boldsymbol{0} && \text{on} \quad \Gamma, \end{aligned} \tag{2.57}$$

with the initial condition $\boldsymbol{u}(0, \boldsymbol{x}) = \tilde{\boldsymbol{u}}(0, \boldsymbol{x}) = \boldsymbol{u}^0$ in Ω. In general, the intermediate velocity $\tilde{\boldsymbol{u}}^n$ is not divergence-free. The second step can be considered as a pressure-correction step. It has the form

$$\begin{aligned} \frac{\tau_q}{\Delta t_n} (\boldsymbol{u}^n - \tilde{\boldsymbol{u}}^n) + \nabla (p^n - \hat{p}^n) &= \boldsymbol{0} \quad \text{in} \quad \Omega, \\ \nabla \cdot \boldsymbol{u}^n &= 0 \quad \text{in} \quad \Omega, \\ \boldsymbol{u}^n \cdot \boldsymbol{n} &= 0 \quad \text{on} \quad \Gamma. \end{aligned} \tag{2.58}$$

The problem (2.58) describes the L^2 projection of the intermediate velocity $\tilde{\boldsymbol{u}}^n$ into the space of weakly divergence-free functions

$$L^2_{\text{div}}(\Omega) = \left\{\boldsymbol{v} \in L^2(\Omega) : \ \nabla \cdot \boldsymbol{v} = 0, \ \boldsymbol{v} \cdot \boldsymbol{n}|_\Gamma = 0\right\}, \tag{2.59}$$

where the divergence and the boundary terms have to be understood in the sense of distributions and traces, respectively. Moreover, as notified in Appendix A, there is no difference in the notation between function spaces for scalar and vector-valued functions in the framework of this thesis. To prove that the projection (2.58) is well-defined, it is necessary to require the H^2-elliptic regularity of the domain Ω, see [53], [133, Remark 1.6]. System (2.57)-(2.58) represents the standard form of the fractional step methods, which is the only form that will be described here. In the literature, the so-called rotational form can also be found, which includes the gradient of $\nu\nabla \cdot \tilde{\boldsymbol{u}}^n$ on the left-hand side of the first equation of (2.58), e.g., see [53, 134] for more details.

Note that (2.58) is a partial differential equation of first order in space unlike the Navier–Stokes equations, which are of second order in space. Hence, it is not possible to enforce the same boundary condition for $\boldsymbol{u}$ in (2.58) as for the Navier–Stokes equations. As the result, the fractional step methods incorporate two solutions for the velocity field $\tilde{\boldsymbol{u}}^n$ and $\boldsymbol{u}^n$. The former one has the correct boundary condition but it is not divergence-free, and the latter one is divergence-free but does not satisfy the correct boundary condition. This property makes it difficult to decide which version of both velocities to consider as the correct one. This issue was intensively discussed in the literature, see [53] for the overview. Concerning the order of convergence, both velocities give the same results.

Next, two popular special cases of pressure-correction schemes will be presented.

Non-Incremental Pressure-Correction Scheme

The so-called non-incremental pressure-correction scheme can be reconstructed from the system (2.57)-(2.58) by choosing $q = 1$, i.e., the BDF1 scheme, which is the backward Euler method, and $s = 0$, i.e., $\hat{p}^n = 0$. Thus, the scheme has the form

$$\begin{aligned}
\frac{1}{\Delta t_n}\left(\tilde{\boldsymbol{u}}^n - \boldsymbol{u}^{n-1}\right) - \nu\Delta\tilde{\boldsymbol{u}}^n + \left(\tilde{\boldsymbol{u}}^n \cdot \nabla\right)\tilde{\boldsymbol{u}}^n &= \boldsymbol{f}^n && \text{in} \quad \Omega,\\
\frac{1}{\Delta t_n}\left(\boldsymbol{u}^n - \tilde{\boldsymbol{u}}^n\right) + \nabla p^n &= \boldsymbol{0} && \text{in} \quad \Omega,\\
\nabla \cdot \boldsymbol{u}^n &= 0 && \text{in} \quad \Omega,\\
\boldsymbol{u}^n \cdot \boldsymbol{n} &= 0 && \text{on} \quad \Gamma,\\
\tilde{\boldsymbol{u}}^n &= \boldsymbol{0} && \text{on} \quad \Gamma,\\
u(0, \boldsymbol{x}) = \tilde{\boldsymbol{u}}(0, \boldsymbol{x}) &= \tilde{\boldsymbol{u}}^0 && \text{in} \quad \Omega.
\end{aligned} \tag{2.60}$$

It is the first and the simplest pressure-correction scheme which was proposed by Chorin and Temam in [28, 132]. Sometimes it is called Chorin's method in the literature.

Restricting the second equation in (2.60) to the boundary Γ, one can see that a non-physical Neumann boundary condition

$$\nabla p^n \cdot \boldsymbol{n} = 0 \quad \text{on} \quad \Gamma \tag{2.61}$$

is enforced for the pressure. The irreducible error of this scheme is of order $\mathcal{O}(\Delta t)$, which makes it meaningless to use a higher-order time-stepping scheme. For further details concerning the convergence of (2.60) see [53, 114].

It is possible to reformulate system (2.60) such that the divergence-free velocity $\boldsymbol{u}$ is eliminated from the solving process. It can be obtained by substituting $\boldsymbol{u}^{n-1}$ in the first equation by

$$\boldsymbol{u}^{n-1} = \tilde{\boldsymbol{u}}^{n-1} - \Delta t_n \nabla p^{n-1}, \tag{2.62}$$

which is just the second equation at time t_{n-1}. The second equation is replaced by the equation for the pressure, which is obtained by taking the negative divergence of the original second equation. Altogether, one has to solve

$$\begin{aligned} \frac{1}{\Delta t_n}\left(\tilde{\boldsymbol{u}}^n - \tilde{\boldsymbol{u}}^{n-1}\right) - \nu\Delta\tilde{\boldsymbol{u}}^n + \left(\tilde{\boldsymbol{u}}^n \cdot \nabla\right)\tilde{\boldsymbol{u}}^n + \nabla p^{n-1} &= \boldsymbol{f}^n && \text{in} \quad \Omega, \\ \Delta p^n &= \frac{1}{\Delta t_n}\nabla \cdot \tilde{\boldsymbol{u}}^n && \text{in} \quad \Omega, \\ \tilde{\boldsymbol{u}}^n &= \boldsymbol{0} && \text{on} \quad \Gamma, \\ \nabla p^n \cdot \boldsymbol{n} &= 0 && \text{on} \quad \Gamma, \\ \boldsymbol{u}(0,\boldsymbol{x}) = \tilde{\boldsymbol{u}}(0,\boldsymbol{x}) &= \tilde{\boldsymbol{u}}^0 && \text{in} \quad \Omega. \end{aligned} \tag{2.63}$$

After having solved (2.63), the divergence-free velocity $\boldsymbol{u}^n$ can be reconstructed by (2.62).

Remark 2.9. In contrast to (2.60), the system (2.63) requires the initial condition not only for the velocity $\boldsymbol{u}^0$ but also for the pressure p^0. In practice, one can obtain it by solving the pressure Poisson equation, see Section 2.2.2, or by storing the numerical solution of the developed flow. ◁

Incremental Pressure-Correction Scheme

The non-incremental pressure-correction scheme (2.60) involves no gradient of the pressure in the first step. In [46], it was proposed to include the gradient of the known pressure field into the first equation of (2.60) leading to a better accuracy of the velocity solution in the correction step of the method. In [139], this idea was combined with the second-order time-stepping scheme resulting in the second-order incremental pressure-correction scheme. Due to the author's name, the method is often called van Kan scheme in the literature.

The second-order scheme scheme can be reconstructed from the system (2.57)-(2.58) using $q = 2$, which results in the BDF2 time-stepping scheme, and $s = 1$. Altogether, the method reads

$$\begin{aligned}
\frac{1}{2\Delta t_n}\left(3\tilde{\boldsymbol{u}}^n - 4\boldsymbol{u}^{n-1} + \boldsymbol{u}^{n-2}\right) - \nu\Delta\tilde{\boldsymbol{u}}^n + \left(\tilde{\boldsymbol{u}}^n\cdot\nabla\right)\tilde{\boldsymbol{u}}^n + \nabla p^{n-1} &= \boldsymbol{f}^n && \text{in} \quad \Omega,\\
\frac{3}{2\Delta t_n}\left(\boldsymbol{u}^n - \tilde{\boldsymbol{u}}^n\right) + \nabla\chi^n &= \boldsymbol{0} && \text{in} \quad \Omega,\\
\nabla\cdot\boldsymbol{u}^n &= 0 && \text{in} \quad \Omega,\\
\boldsymbol{u}^n\cdot\boldsymbol{n} &= 0 && \text{on} \quad \Gamma,\\
\tilde{\boldsymbol{u}}^n &= \boldsymbol{0} && \text{on} \quad \Gamma,
\end{aligned} \tag{2.64}$$

where

$$\chi^n = p^n - p^{n-1} \tag{2.65}$$

is called the pressure increment. Restricting the second equation in (2.64) to Γ and using the prescribed boundary conditions for $\tilde{\boldsymbol{u}}$ and $\boldsymbol{u}$, one obtains the following artificial boundary condition for the pressure

$$\nabla p^n\cdot\boldsymbol{n} = \nabla p^{n-1}\cdot\boldsymbol{n} = \ldots = \nabla p^0\cdot\boldsymbol{n} \quad \text{on} \quad \Gamma, \tag{2.66}$$

which is non-physical and thus leads to numerical boundary layers. Proofs of the error estimates that are the same for all second-order A-stable time-stepping schemes can be found in [36, 52, 124]. The irreducible splitting error of this scheme is of order $\mathcal{O}(\Delta t^2)$ and therefore the use of a higher-order time-stepping scheme would not improve the accuracy of the method.

Similarly to the non-incremental pressure-correction scheme, also here a system of equations without any occurrence of $\boldsymbol{u}$ can be obtained by substituting $\boldsymbol{u}^{n-1}$ in the first equation of (2.64) by

$$\boldsymbol{u}^{n-1} = \tilde{\boldsymbol{u}}^{n-1} - \frac{2\Delta t_n}{3}\nabla\chi^{n-1}, \tag{2.67}$$

which is basically the second equation of (2.64). The negative divergence of the second equation of (2.64) yields the pressure equation. Hence, one obtains the system

$$\begin{aligned}
\frac{1}{2\Delta t_n}\left(3\tilde{\boldsymbol{u}}^n - 4\tilde{\boldsymbol{u}}^{n-1} + \tilde{\boldsymbol{u}}^{n-2}\right) - \nu\Delta\tilde{\boldsymbol{u}}^n + \left(\tilde{\boldsymbol{u}}^n\cdot\nabla\right)\tilde{\boldsymbol{u}}^n + \nabla p^{n-1}&\\
+\frac{4}{3}\nabla\chi^{n-1} - \frac{1}{3}\nabla\chi^{n-2} &= \boldsymbol{f}^n && \text{in} \quad \Omega,\\
\Delta\chi^n &= \frac{3}{2\Delta t_n}\nabla\cdot\tilde{\boldsymbol{u}}^n && \text{in} \quad \Omega,\\
\tilde{\boldsymbol{u}}^n &= \boldsymbol{0} && \text{on} \quad \Gamma,\\
\nabla\chi^n\cdot\boldsymbol{n} &= 0 && \text{on} \quad \Gamma.
\end{aligned} \tag{2.68}$$

The divergence-free velocity $\boldsymbol{u}^n$ can be computed using equation (2.67).

Remark 2.10. Both versions of the incremental pressure-correction scheme (2.64) and (2.68) need to be equipped besides the initial condition for the velocity $\boldsymbol{u}^0$ also with p^0 and $(\boldsymbol{u}^1, p^1)$. Moreover, the system (2.68) even needs p^2. In practice, the value of

p^0 can be obtained as described in Remark 2.9. The other values can be computed by performing the required number of time steps of the non-incremental pressure-correction scheme or of the incremental pressure-correction scheme combined with a time-stepping scheme of first order. ◁

Remark 2.11. When the finite element method is applied to perform the simulations, which is the case in this thesis, it is of advantage to use the schemes (2.63) and (2.68), which do not incorporate the divergence-free velocity $\boldsymbol{u}$, see [53]. Otherwise, the divergence-free velocity has to be recovered by (2.62) or (2.67) after having solved the pressure equation (the second equation of (2.63) and (2.68), respectively) in the weak form. As both equations (2.62) and (2.67) require the computation of the pressure gradient, the resulting velocity occurs to be discontinuous. For this reason, in what follows only the schemes which do not incorporate the divergence-free velocity $\boldsymbol{u}$ will be considered. ◁

Remark 2.12. (Mixed boundary conditions). For the sake of simplicity, the above projection methods were presented for the Navier–Stokes equations (2.40)-(2.41) equipped with the homogeneous Dirichlet boundary condition for the velocity. However, the most applications require more complicated boundary conditions. For example, for the numerical investigations in Chapter 5 the boundary conditions

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{0} && \text{on} \quad (0,T] \times \Gamma_0, \\ \boldsymbol{u} &= \boldsymbol{g}_{\mathrm{D}} && \text{on} \quad (0,T] \times \Gamma_{\text{in}}, \\ (\nu \nabla \boldsymbol{u} - p\mathbb{I})\, \boldsymbol{n} &= \boldsymbol{0} && \text{on} \quad (0,T] \times \Gamma_{\text{out}} \end{aligned} \tag{2.69}$$

are prescribed, where Γ_0, Γ_{in}, and Γ_{out} are mutually disjoint parts of the boundary Γ with $\Gamma = \Gamma_0 \cup \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$ and $\Gamma_{\mathrm{D}} = \Gamma_0 \cup \Gamma_{\text{in}}$, see Section 2.2.1. For this setting, the velocity boundary conditions in the schemes (2.60) and (2.64) are

$$\begin{aligned} \tilde{\boldsymbol{u}}^n &= \boldsymbol{0}, & \boldsymbol{u}^n \cdot \boldsymbol{n} &= 0 && \text{on} \quad \Gamma_0, \\ \tilde{\boldsymbol{u}}^n &= \boldsymbol{g}_{\mathrm{D}}, & \boldsymbol{u}^n \cdot \boldsymbol{n} &= \boldsymbol{g}_{\mathrm{D}}^n \cdot \boldsymbol{n} && \text{on} \quad \Gamma_{\text{in}}. \end{aligned} \tag{2.70}$$

The second equations of (2.60) and (2.64), respectively, taken into the normal direction thus yield the following boundary conditions for the pressure and the pressure increment:

$$\nabla p^n \cdot \boldsymbol{n} = 0, \quad \nabla \chi^n \cdot \boldsymbol{n} = 0 \quad \text{on} \quad \Gamma_0 \cup \Gamma_{\text{in}}. \tag{2.71}$$

The choice of the correct boundary condition for the pressure and the pressure increment on Γ_{out} is not trivial. Based on the discussion in Section 2.2.2, the homogeneous Dirichlet boundary conditions can be prescribed for the pressure and the pressure increment on Γ_{out}, i.e.,

$$p^n = 0, \quad \chi^n = 0 \quad \text{on} \quad \Gamma_{\text{out}}. \tag{2.72}$$

For the intermediate velocity in (2.63) and (2.68), it holds

$$\nabla \tilde{\boldsymbol{u}}^n \boldsymbol{n} = \boldsymbol{0} \quad \text{on} \quad \Gamma_{\text{out}}. \tag{2.73}$$

◁

2.2.5. Linearization

In this section, the second difficulty of the Navier–Stokes equations mentioned in Remark 2.5 is discussed. Semi-discrete problems obtained in Section 2.2.3 and 2.2.4 are nonlinear and thus have to be solved iteratively in each time step. For the solution of the nonlinear saddle point problems, the reader is referred to [18] for more details.

In each nonlinear iteration at time t_n, the initial guess denoted by $\boldsymbol{u}_0^n$ is required. For its value, one can use the solution from the previous time step $\boldsymbol{u}^{n-1}$ or some extrapolation of the solution of more than one previous time step.

A popular method for the solution of the nonlinear problems is the fixed point iteration or also called the Picard iteration. It linearizes the convective term by the following approximation: Given a known velocity field $\boldsymbol{u}_{k-1}^n$, then

$$(\boldsymbol{u}_k^n \cdot \nabla)\boldsymbol{u}_k^n \approx (\boldsymbol{u}_{k-1}^n \cdot \nabla)\boldsymbol{u}_k^n. \tag{2.74}$$

The semi-discretized problem (2.52) combined with the Picard iteration reads as follows: For each $n = 1, 2, \dots$ and given $\boldsymbol{u}_{k-1}^n$ with $k = 1, 2, \dots$, find $(\boldsymbol{u}_k^n, p_k^n)$ such that

$$\begin{aligned} \boldsymbol{u}_k^n &+ \theta\Delta t_n \left[-\nu\Delta\boldsymbol{u}_k^n + (\boldsymbol{u}_{k-1}^n \cdot \nabla)\boldsymbol{u}_k^n\right] + \Delta t_n \nabla p_k^n \\ &= \boldsymbol{u}^{n-1} - (1-\theta)\Delta t_n \left[-\nu\Delta\boldsymbol{u}^{n-1} + (\boldsymbol{u}^{n-1} \cdot \nabla)\boldsymbol{u}^{n-1}\right] \\ &\quad + (1-\theta)\Delta t_n \boldsymbol{f}^{n-1} + \theta\Delta t_n \boldsymbol{f}^n, \\ \nabla \cdot \boldsymbol{u}_k^n &= 0. \end{aligned} \tag{2.75}$$

The right-hand side does not change during one time iteration. Equations of type (2.75) are called Oseen equations, see [75, Chapter 5].

The semi-implicit methods, also called the IMEX (implicit-explicit) schemes, avoid the solution of the nonlinear system. The convective term is approximated by

$$(\boldsymbol{u}_k^n \cdot \nabla)\boldsymbol{u}_k^n \approx (\boldsymbol{u}^{\text{old}} \cdot \nabla)\boldsymbol{u}_k^n, \tag{2.76}$$

where $\boldsymbol{u}^{\text{old}}$ can be obtained from the already computed solutions. The simplest choice consists in setting $\boldsymbol{u}^{\text{old}} = \boldsymbol{u}^{n-1}$, which can be interpreted as the Picard iteration with only one iteration step. This method will be simply denoted by IMEX. Another possibility is to use the linear extrapolation of the previous two time steps as follows

$$\boldsymbol{u}^{\text{old}} = \frac{\Delta t_n}{\Delta t_{n-1}}(\boldsymbol{u}^{n-1} - \boldsymbol{u}^{n-2}) + \boldsymbol{u}^{n-1}, \tag{2.77}$$

see [75] for more details. A similar scheme combined with the Crank–Nicolson method with a constant length of the time step Δt was proposed and investigated in [70]. It has the form

$$\Delta t \left[\left(\left(\boldsymbol{u}^{n-1} + 0.5\boldsymbol{u}^{n-2} - 0.5\boldsymbol{u}^{n-3}\right) \cdot \nabla\right) \boldsymbol{u}^{n+1/2}\right], \tag{2.78}$$

where $\boldsymbol{u}^{n+1/2} = \frac{1}{2}\left(\boldsymbol{u}^n + \boldsymbol{u}^{n-1}\right)$. In what follows, this method will be referred to IMEX-LE. More details on the relation between (2.77) and (2.78) can be found in [75].

In the framework of projection methods, the convective term is usually treated semi-implicitly or even explicitly to obtain a linear problem. Thus, instead of $(\tilde{\boldsymbol{u}}^n \cdot \nabla)\tilde{\boldsymbol{u}}^n$ in (2.60) and (2.64), one can use any version of the semi-implicit or explicit terms like

$$(\tilde{\boldsymbol{u}}^{n-1} \cdot \nabla)\tilde{\boldsymbol{u}}^{n},\ (\tilde{\boldsymbol{u}}^{n-1} \cdot \nabla)\tilde{\boldsymbol{u}}^{n-1},\ (\boldsymbol{u}^{n-1} \cdot \nabla)\tilde{\boldsymbol{u}}^{n},\ (\boldsymbol{u}^{n-1} \cdot \nabla)\boldsymbol{u}^{n-1}.$$

Another iterative approach to solve nonlinear problems is Newton's method, which uses the following approximation of the convective term: Given a known velocity field $\boldsymbol{u}_{k-1}^n$, then

$$(\boldsymbol{u}_k^n \cdot \nabla)\boldsymbol{u}_k^n \approx (\boldsymbol{u}_{k-1}^n \cdot \nabla)\boldsymbol{u}_k^n + (\boldsymbol{u}_k^n \cdot \nabla)\boldsymbol{u}_{k-1}^n - (\boldsymbol{u}_{k-1}^n \cdot \nabla)\boldsymbol{u}_{k-1}^n. \tag{2.79}$$

The semi-discretized problem (2.52) combined with Newton's method reads: For each $n = 1, 2, \ldots$ and given $\boldsymbol{u}_{k-1}^n$ with $k = 1, 2, \ldots$, find $(\boldsymbol{u}_k^n, p_k^n)$ such that

$$\begin{aligned}
&\boldsymbol{u}_k^n + \theta\Delta t_n \left[-\nu\Delta\boldsymbol{u}_k^n + (\boldsymbol{u}_{k-1}^n \cdot \nabla)\boldsymbol{u}_k^n + (\boldsymbol{u}_k^n \cdot \nabla)\boldsymbol{u}_{k-1}^n\right] + \Delta t_n \nabla p_k^n \\
&\quad = \boldsymbol{u}^{n-1} - (1-\theta)\Delta t_{n-1} \left[-\nu\Delta\boldsymbol{u}^{n-1} + (\boldsymbol{u}^{n-1} \cdot \nabla)\boldsymbol{u}^{n-1}\right] \\
&\qquad + \theta\Delta t_n (\boldsymbol{u}_{k-1}^n \cdot \nabla)\boldsymbol{u}_{k-1}^n + (1-\theta)\Delta t_n \boldsymbol{f}^{n-1} + \theta\Delta t_n \boldsymbol{f}^n, \\
&\nabla \cdot \boldsymbol{u}_k^n = 0.
\end{aligned} \tag{2.80}$$

In [74], a three-dimensional problem was studied to compare the efficiency of the Picard iteration and of the Newton's method. In that case, the Picard iteration was found to be much more efficient than the Newton's method. For a more detailed comparison of both methods in terms of accuracy, convergence and computational cost, the reader is referred to [75, Chapter 6].

As the stopping criterion for a nonlinear iteration, usually the Euclidean norm of the residual vector is evaluated. The iteration is proceeded until the norm is smaller than a certain tolerance number. If the tolerance threshold is too small, then one expects a very accurate solution but the computational time can grow unproportionally. If the tolerance threshold is set too large, the computation is more efficient but one possibly obtains a poor accuracy of the solution. To the best of the author's knowledge, there is no general approach to choose a good tolerance number. It depends on the example and the used numerical methods.

2.2.6. Weak Formulation

The semi-discrete problems presented in Sections 2.2.3 and 2.2.4 will be approximated by the Galerkin finite element method in Section 2.2.7, whose starting point is the variational or weak formulation of the problem. At this point, the first difficulty of the Navier–Stokes equations mentioned in Remark 2.5 has to be addressed. The velocity and the pressure fields are coupled in such a special way that the velocity and the pressure spaces cannot be chosen arbitrarily to obtain a well-posed problem. The stationary Stokes equations (2.43) possess the same coupling of the velocity and pressure as the time-dependent Navier–Stokes equations (2.40)-(2.41). As the Stokes equations are linear, the abstract theory of the linear saddle point problems can be applied to them to investigate the well-posedness of the problem, e.g., see [45]. According to the theory, the so-called inf-sup condition (2.81) has to be satisfied in order to ensure the uniqueness of the pressure solution.

Definition 2.1. *Let V and Q denote real Hilbert spaces with inner products $(\cdot,\cdot)_V$ and $(\cdot,\cdot)_Q$ and the induced norms $\|\cdot\|_V$ and $\|\cdot\|_Q$, respectively. The (Babuška-Brezzi) inf-sup condition reads as follows: There is a constant $\beta > 0$ such that*

$$\inf_{q\in Q,\, q\neq 0} \sup_{\boldsymbol{v}\in V,\, \boldsymbol{v}\neq \boldsymbol{0}} \frac{-\left(q, \nabla\cdot\boldsymbol{v}\right)}{\|\boldsymbol{v}\|_V \|q\|_Q} > \beta. \tag{2.81}$$

Consider the Navier–Stokes equations equipped with the boundary conditions

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{0} && \text{on} \quad (0,T]\times\Gamma_0, \\ \boldsymbol{u} &= \boldsymbol{g}_{\mathrm{D}} && \text{on} \quad (0,T]\times\Gamma_{\mathrm{in}}, \\ \left(\nu\nabla\boldsymbol{u} - p\mathbb{I}\right)\boldsymbol{n} &= \boldsymbol{0} && \text{on} \quad (0,T]\times\Gamma_{\mathrm{out}}, \end{aligned} \tag{2.82}$$

where Γ_0, Γ_{in}, and Γ_{out} are mutually disjoint parts of the boundary Γ with $\Gamma = \Gamma_0 \cup \Gamma_{\mathrm{in}} \cup \Gamma_{\mathrm{out}}$ and $\Gamma_{\mathrm{D}} = \Gamma_0 \cup \Gamma_{\mathrm{in}}$, see Section 2.2.1. It is the setting required for the numerical investigations in Chapter 5. The velocity space incorporates the prescribed essential boundary conditions. Let V and Q denote the velocity and pressure spaces, respectively, defined by

$$V = H^1_{\Gamma_{\mathrm{D}}}(\Omega) = \{\boldsymbol{v}\in H^1(\Omega) :\ \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_{\mathrm{D}}\}, \quad Q = L^2(\Omega), \tag{2.83}$$

where the value of $\boldsymbol{u}$ on the boundary Γ_{D} is to be understood in the sense of traces. It can be proven that V and Q satisfy the inf-sup condition (2.81), see [113, Proposition 5.3.2]. Furthermore, let $\boldsymbol{u}_{\boldsymbol{g}} \in H^1(\Omega)$ denote an extension of $\boldsymbol{g}_{\mathrm{D}}$ into Ω for $t\in(0,T]$.

Next, as an example, one variational formulation for each type of the problems from Sections 2.2.3 and 2.2.4, respectively, will be derived. The derivation of the variational formulation will be first presented for the semi-discrete equations (2.75) (saddle point problem) and then for the incremental pressure-correction scheme (2.68) (decoupled problem) with the convective term approximated by $(\tilde{\boldsymbol{u}}^{n-1}\cdot\nabla)\tilde{\boldsymbol{u}}^n$. For the other problems, the weak formulation can be built equivalently.

Saddle Point Problem

A weak formulation of the semi-discretized Navier–Stokes equations (2.75) is obtained by multiplying the momentum equation (the first equation) with a test function $\boldsymbol{v}\in V$ and the continuity equation (the second equation) with a test function $q\in Q$. Then, the equations are integrated over Ω.

For the continuity equation it holds

$$\int\limits_\Omega \left(\nabla\cdot\boldsymbol{u}^n\right) q \, d\boldsymbol{x} = \left(\nabla\cdot\boldsymbol{u}^n, q\right) = 0.$$

Using the integration by parts and the Gaussian theorem, the pressure term in the momentum equation (2.40) has the form

$$\int_\Omega \nabla p^n \ \cdot \boldsymbol{v} \, d\boldsymbol{x} = \int_\Gamma p^n \boldsymbol{v} \cdot \boldsymbol{n} \, d\boldsymbol{s} - (p^n, \nabla \cdot \boldsymbol{v}) \,. \tag{2.84}$$

The first term on the right-hand side is zero on $\Gamma_0 \cup \Gamma_{\text{in}}$ as $\boldsymbol{v} \in V$.

The variational formulation of the viscous term is obtained by integration by parts and the Gaussian theorem as follows

$$-\nu \int_\Omega \Delta \boldsymbol{u}^n \, \boldsymbol{v} \, d\boldsymbol{x} = -\nu \int_\Gamma \nabla \boldsymbol{u}^n \, \boldsymbol{v} \cdot \boldsymbol{n} \, d\boldsymbol{s} + \nu \, (\nabla \boldsymbol{u}^n, \nabla \boldsymbol{v}) \,, \tag{2.85}$$

where the first term on the right-hand side vanishes on $\Gamma_0 \cup \Gamma_{\text{D}}$ due to the definition of the space V. The L^2 inner product of tensors is defined component by component.

Altogether, the weak formulation of (2.75) reads: For $n = 1, 2, \ldots$ and given $\boldsymbol{u}_{k-1}^n \in H^1(\Omega)$ with $\boldsymbol{u}_{k-1}^n - \boldsymbol{u}_{\boldsymbol{g}}^n \in V$ and $k = 1, 2, \ldots$, find $(\boldsymbol{u}_k^n, p_k^n) \in H^1(\Omega) \times Q$ with $\boldsymbol{u}_k^n - \boldsymbol{u}_{\boldsymbol{g}}^n \in V$ such that

$$\begin{aligned} (\boldsymbol{u}_k^n, \boldsymbol{v}) &+ \theta \Delta t_n \left[(\nu \nabla \boldsymbol{u}_k^n, \nabla \boldsymbol{v}) + \big((\boldsymbol{u}_{k-1}^n \cdot \nabla) \boldsymbol{u}_k^n, \boldsymbol{v} \big) \right] - \Delta t_n \, (\nabla \cdot \boldsymbol{v}, p_k^n) \\ &= \big(\boldsymbol{u}^{n-1}, \boldsymbol{v} \big) - (1-\theta) \Delta t_n \left[\big(\nu \nabla \boldsymbol{u}^{n-1}, \nabla \boldsymbol{v} \big) + \big((\boldsymbol{u}^{n-1} \cdot \nabla) \boldsymbol{u}^{n-1}, \boldsymbol{v} \big) \right] \\ &\quad + (1-\theta) \Delta t_n \left(\boldsymbol{f}^{n-1}, \boldsymbol{v} \right) + \theta \Delta t_n \, (\boldsymbol{f}^n, \boldsymbol{v}) \,, \\ - (\nabla \cdot \boldsymbol{u}_k^n, q) &= 0, \end{aligned} \tag{2.86}$$

for all $(\boldsymbol{v}, q) \in V \times Q$ and parameter θ to be chosen, see Table 2.1. Note that the boundary terms on Γ_{out} from (2.84) and (2.85) cancel out due to the natural boundary condition imposed on Γ_{out}, see (2.82).

Decoupled Problem

According to Remark 2.12, one can employ the same velocity space V for the variational formulation of the decoupled problem (2.68) as for the saddle point problem because the intermediate velocity $\tilde{\boldsymbol{u}}^n$ satisfies the same Dirichlet boundary conditions as $\boldsymbol{u}$ in (2.86). However, one needs a different definition of the pressure space for the following reasons: Firstly, the second equation in (2.68) involves a higher-order derivative of the pressure (or more precisely the pressure increment) than the saddle point problem and, secondly, the decoupled problem is equipped with the homogeneous Dirichlet boundary condition for the pressure on Γ_{out}, which has to be incorporated by the definition of the corresponding space. Thus, the pressure space for the decoupled problem (2.68) is defined by

$$\tilde{Q} = H^1_{\Gamma_{\text{out}}}(\Omega) = \{ q \in H^1(\Omega) : \ q = 0 \text{ on } \Gamma_{\text{out}} \}. \tag{2.87}$$

The variational formulation of (2.68) with $(\tilde{\boldsymbol{u}}^{n-1} \cdot \nabla) \tilde{\boldsymbol{u}}^n$ as the approximation of the convective term and the boundary conditions from Remark 2.12 is obtained by multiplying the first equation with a test function $\boldsymbol{v} \in V$ and the second equation with a test function $q \in \tilde{Q}$. Then the resulting equations are integrated over Ω. For the terms including the gradient of the pressure and the Laplacian of the velocity, similar transformations with $\tilde{\boldsymbol{u}}$ instead of $\boldsymbol{u}$ can be undertaken as in (2.84) and (2.85), respectively. Due to the boundary conditions discussed in Remark 2.12, the respective boundary terms in (2.84) and (2.85) vanish on the entire boundary Γ. The weak form of the Laplacian of the pressure increment on the left-hand side of the second equation in (2.68) can be achieved by using integration by parts and the Gaussian theorem as follows

$$\int\limits_{\Omega} \Delta\chi^n q \, d\boldsymbol{x} = \int\limits_{\Gamma} \nabla\chi^n q \cdot \boldsymbol{n} \, d\boldsymbol{s} - (\nabla\chi^n, \nabla q) \, . \tag{2.88}$$

Also in this case the boundary term on the right-hand side vanishes on Γ following the specification of the boundary conditions for the pressure increment, see Remark 2.12.

Finally, the variational formulation reads: For $n = 2, 3, \ldots$ and given $\left(\tilde{\boldsymbol{u}}^{n-1}, \chi^{n-1}\right) \in H^1(\Omega) \times \tilde{Q}$ and $\left(\tilde{\boldsymbol{u}}^{n-2}, \chi^{n-2}\right) \in H^1(\Omega) \times \tilde{Q}$ with $\tilde{\boldsymbol{u}}^{n-1} - \boldsymbol{u}_{\boldsymbol{g}}^{n-1} \in V$ and $\tilde{\boldsymbol{u}}^{n-2} - \boldsymbol{u}_{\boldsymbol{g}}^{n-1} \in V$, find $(\tilde{\boldsymbol{u}}^n, \chi^n) \in H^1(\Omega) \times \tilde{Q}$ with $\tilde{\boldsymbol{u}}^n - \boldsymbol{u}_{\boldsymbol{g}}^n \in V$ such that

$$\begin{aligned}
\frac{3}{2}\left(\tilde{\boldsymbol{u}}^n, \boldsymbol{v}\right) + \Delta t_n \left[\nu\left(\nabla\tilde{\boldsymbol{u}}^n, \nabla\boldsymbol{v}\right) + \left(\left(\tilde{\boldsymbol{u}}^{n-1} \cdot \nabla\right)\tilde{\boldsymbol{u}}^n, \boldsymbol{v}\right)\right] &= 2\left(\tilde{\boldsymbol{u}}^{n-1}, \boldsymbol{v}\right) \\
&\quad - \frac{1}{2}\left(\tilde{\boldsymbol{u}}^{n-2}, \boldsymbol{v}\right) + \Delta t_n \Big[\left(p^{n-1}, \nabla\cdot\boldsymbol{v}\right) + \frac{4}{3}\left(\chi^{n-1}, \nabla\cdot\boldsymbol{v}\right) \\
&\quad - \frac{1}{3}\left(\chi^{n-2}, \nabla\cdot\boldsymbol{v}\right) + \left(\boldsymbol{f}^n, \boldsymbol{v}\right)\Big], \\
(\nabla\chi^n, \nabla q) &= -\frac{3}{2\Delta t_n}\left(\nabla\cdot\tilde{\boldsymbol{u}}^n, q\right),
\end{aligned} \tag{2.89}$$

for all $(\boldsymbol{v}, q) \in V \times \tilde{Q}$, and $\chi^n = p^n - p^{n-1}$.

2.2.7. Galerkin Finite Element Method

For the sake of clarity, Galerkin finite element method will be presented for a homogeneous Dirichlet boundary condition for the velocity field on the entire boundary Γ. Then, the corresponding velocity and pressure spaces have the form

$$\begin{aligned}
V &= H_0^1(\Omega) = \{\boldsymbol{v} \in H^1(\Omega) : \ \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma\}, \\
Q &= L_0^2(\Omega) = \{q \in L^2(\Omega) : \ \int_\Omega q \, d\boldsymbol{x} = 0\}, \quad \tilde{Q} = L_0^2(\Omega) \cap H^1(\Omega).
\end{aligned}$$

Implementation issues of the Galerkin finite element method for inhomogeneous Dirichlet conditions on $\Gamma_{\mathrm{D}} \subset \Gamma$ are discussed, e.g., in [3]. In what follows, the three-dimensional case will be considered. The two-dimensional representation is obtained equivalently.

The Galerkin finite element method is obtained by replacing the infinite-dimensional velocity space V and the pressure space Q (or $\tilde{Q}$) by the finite-dimensional spaces V_h and Q_h in the corresponding variational formulation. Here, only conforming finite element spaces will be considered, i.e., $V_h \subset V$ and $Q_h \subset Q$ (or $Q_h \subset \tilde{Q}$). The pair of the finite element spaces will be denoted by V_h/Q_h. Note that for the case of mixed boundary conditions (2.82), one needs to define a different finite element pressure space for the decoupled problem than that for the saddle point problem.

The velocity space V_h and the pressure space Q_h are equipped with a basis. For the vector-valued velocity space it holds

$$\begin{aligned} V_h &= \operatorname{span}\left\{\boldsymbol{\phi}_{h,i}\right\}_{i=1}^{3N_v} \\ &= \operatorname{span}\left\{\left\{\begin{pmatrix}\phi_{h,i}\\0\\0\end{pmatrix}\right\}_{i=1}^{N_v} \cup \left\{\begin{pmatrix}0\\\phi_{h,i}\\0\end{pmatrix}\right\}_{i=1}^{N_v} \cup \left\{\begin{pmatrix}0\\0\\\phi_{h,i}\end{pmatrix}\right\}_{i=1}^{N_v}\right\}, \end{aligned} \tag{2.90}$$

i.e., each basis function has non-zero values only in one of the components and N_v is the number of degrees of freedom in each component of the velocity. The pressure space is of the form

$$Q_h = \operatorname{span}\left\{\psi_{h,i}\right\}_{i=1}^{N_p}, \tag{2.91}$$

where N_p is the number of pressure degrees of freedom. Hence, the finite element velocity $\boldsymbol{u}_h$ and the pressure p_h have the unique representation

$$\boldsymbol{u}_h = \sum_{i=1}^{3N_v} \boldsymbol{u}_{h,i}\boldsymbol{\phi}_{h,i}, \quad p_h = \sum_{i=1}^{N_p} p_{h,i}\psi_{h,i}, \tag{2.92}$$

where $\underline{\boldsymbol{u}}_h = (\boldsymbol{u}_{h,i})_{i=1}^{3N_v}$ and $\underline{p}_h = (p_{h,i})_{i=1}^{N_p}$ are unknown real coefficients.

Saddle Point Problem

The fully discretized Galerkin finite element method for the Navier–Stokes equations combined with the one-step θ-scheme and the Picard iteration reads: For each $n = 1, 2, \ldots$ and given $\boldsymbol{u}_{h,k-1}^n \in V_h$ with $k = 1, 2, \ldots$, find $(\boldsymbol{u}_{h,k}^n, p_{h,k}^n) \in V_h \times Q_h$ such that

$$\begin{aligned} &\left(\boldsymbol{u}_{h,k}^n, \boldsymbol{v}_h\right) + \theta\Delta t_n \left[\left(\nu\nabla\boldsymbol{u}_{h,k}^n, \nabla\boldsymbol{v}_h\right) + \left((\boldsymbol{u}_{h,k-1}^n\cdot\nabla)\boldsymbol{u}_{h,k}^n, \boldsymbol{v}_h\right)\right] \\ &\qquad - \Delta t_n \left(\nabla\cdot\boldsymbol{v}_h, p_{h,k}^n\right) \\ &\qquad = \left(\boldsymbol{u}_h^{n-1}, \boldsymbol{v}_h\right) - (1-\theta)\Delta t_n \Big[\left(\nu\nabla\boldsymbol{u}_h^{n-1}, \boldsymbol{v}_h\right) \\ &\qquad\quad + \left((\boldsymbol{u}_h^{n-1}\cdot\nabla)\boldsymbol{u}_h^{n-1}, \boldsymbol{v}_h\right)\Big] + (1-\theta)\Delta t_n \left(\boldsymbol{f}^{n-1}, \boldsymbol{v}_h\right) + \theta\Delta t_n \left(\boldsymbol{f}^n, \boldsymbol{v}_h\right), \\ &-\left(\nabla\cdot\boldsymbol{u}_{h,k}^n, q_h\right) = 0, \end{aligned} \tag{2.93}$$

for all $(\boldsymbol{v}_h, q_h) \in V_h \times Q_h$. In practice, the representation (2.92) is inserted into (2.93) and tested with each basis function separately. As a result, one obtains the linear system of equations

$$\begin{pmatrix} M_h^{\mathrm{NS}} + \theta\Delta t_n \left[A_h^{\mathrm{NS}} + N_h^{\mathrm{NS}}(\underline{\boldsymbol{u}}_{h,k-1}^n)\right] & \left(B_h^{\mathrm{NS}}\right)^T \\ B_h^{\mathrm{NS}} & 0 \end{pmatrix} \begin{pmatrix} \underline{\boldsymbol{u}}_{h,k}^n \\ \underline{p}_{h,k}^n \end{pmatrix} = \begin{pmatrix} \underline{\boldsymbol{l}}_h^n \\ \underline{0} \end{pmatrix}, \tag{2.94}$$

where

$$\begin{aligned}
\left(M_h^{\mathrm{NS}}\right)_{ij} &= \left(\boldsymbol{\phi}_{h,j}, \boldsymbol{\phi}_{h,i}\right), \quad i,j = 1,\ldots,3N_v, && (2.95)\\
\left(A_h^{\mathrm{NS}}\right)_{ij} &= \left(\nu\nabla\boldsymbol{\phi}_{h,j}, \nabla\boldsymbol{\phi}_{h,i}\right), \quad i,j = 1,\ldots,3N_v, && (2.96)\\
\left(N_h^{\mathrm{NS}}(\underline{\boldsymbol{u}}_h)\right)_{ij} &= \left(\left(\underline{\boldsymbol{u}}_h \cdot \nabla\right)\boldsymbol{\phi}_{h,j}, \boldsymbol{\phi}_{h,i}\right), \quad i,j = 1,\ldots,3N_v, && (2.97)\\
\left(B_h^{\mathrm{NS}}\right)_{ij} &= -\left(\nabla\cdot\boldsymbol{\phi}_{h,j}, \psi_{h,i}\right), \quad i = 1,\ldots,N_p,\ j = 1,\ldots,3N_v, && (2.98)\\
\boldsymbol{l}_{h,i}^n &= M_h^{\mathrm{NS}}\underline{\boldsymbol{u}}_h^{n-1} - (1-\theta)\Delta t_n\left[A_h^{\mathrm{NS}} + N_h^{\mathrm{NS}}(\underline{\boldsymbol{u}}_h^{n-1})\right]\underline{\boldsymbol{u}}_h^{n-1} && (2.99)\\
&\quad + (1-\theta)\Delta t_n\left(\boldsymbol{f}^{n-1}, \boldsymbol{\phi}_{h,i}\right) + \theta\Delta t_n\left(\boldsymbol{f}^n, \boldsymbol{\phi}_{h,i}\right), \quad i = 1,\ldots,3N_v.
\end{aligned}$$

The matrices $M_h^{\mathrm{NS}}, A_h^{\mathrm{NS}}, N_h^{\mathrm{NS}}(\underline{\boldsymbol{u}}_h) \in \mathbb{R}^{3N_v\times 3N_v}$ consist of 3×3 blocks, e.g.,

$$M_h^{\mathrm{NS}} = \begin{pmatrix} M_{11} & 0 & 0\\ 0 & M_{11} & 0\\ 0 & 0 & M_{11}, \end{pmatrix}$$

where the expression (2.95) vanishes, if $\boldsymbol{\phi}_{h,i}$ and $\boldsymbol{\phi}_{h,j}$ represent different components, e.g., $\boldsymbol{\phi}_{h,i} = (\phi_{h,i}, 0, 0)^T$ and $\boldsymbol{\phi}_{h,j} = (0, \phi_{h,j}, 0)^T$. Non-zero matrices on the diagonal have all the same values as $\phi_{h,i}$ and $\phi_{h,j}$ are of the same form independent of the fact which component they describe. For more details on the practical implementation and solution of the linear saddle point systems of type (2.94), see [55, 75].

Decoupled Problem

The fully discretized Galerkin finite element method for the Navier–Stokes equations resulting from the incremental pressure-correction scheme (2.68) reads: For $n = 2, 3, \ldots$ and given $\left(\tilde{\boldsymbol{u}}_h^{n-1}, \chi_h^{n-1}\right) \in V_h \times Q_h$ and $\left(\tilde{\boldsymbol{u}}_h^{n-2}, \chi_h^{n-2}\right) \in V_h \times Q_h$, find $\left(\tilde{\boldsymbol{u}}_h^n, \chi_h^n\right) \in V_h \times Q_h$ such that

$$\begin{aligned}
\frac{3}{2}\left(\tilde{\boldsymbol{u}}_h^n, \boldsymbol{v}_h\right) + \Delta t_n\left[\nu\left(\nabla\tilde{\boldsymbol{u}}_h^n, \nabla\boldsymbol{v}_h\right) + \left(\left(\tilde{\boldsymbol{u}}_h^{n-1}\cdot\nabla\right)\tilde{\boldsymbol{u}}_h^n, \boldsymbol{v}_h\right)\right] &= 2\left(\tilde{\boldsymbol{u}}_h^{n-1}, \boldsymbol{v}_h\right)\\
-\frac{1}{2}\left(\tilde{\boldsymbol{u}}_h^{n-2}, \boldsymbol{v}_h\right) + \Delta t_n\Big[\left(p_h^{n-1}, \nabla\cdot\boldsymbol{v}_h\right) &+ \frac{4}{3}\left(\chi_h^{n-1}, \nabla\cdot\boldsymbol{v}_h\right)\\
-\frac{1}{3}\left(\chi_h^{n-2}, \nabla\cdot\boldsymbol{v}_h\right) &+ \left(\boldsymbol{f}^n, \boldsymbol{v}_h\right)\Big],\\
\left(\nabla\chi_h^n, \nabla q_h\right) &= -\frac{3}{2\Delta t_n}\left(\nabla\cdot\tilde{\boldsymbol{u}}_h^n, q_h\right),
\end{aligned} \tag{2.100}$$

for all $(\boldsymbol{v}_h, q_h) \in V_h \times Q_h$, and $\chi_h^n = p_h^n - p_h^{n-1}$. In contrast to (2.93), problem (2.100) is not a system of linear equations of the saddle point type but consists of two separate systems of linear equations for the intermediate velocity and the pressure increment, respectively, which have to be solved one after another.

As $\tilde{\boldsymbol{u}}_h^n \in V_h$ and $\chi_h^n \in Q_h$, the intermediate velocity and the pressure increment can be written in the same form as the velocity and the pressure in (2.92), namely as

$$\tilde{\boldsymbol{u}}_h = \sum_{i=1}^{3N_v}\tilde{\boldsymbol{u}}_{h,i}\boldsymbol{\phi}_{h,i}, \quad \chi_h = \sum_{i=1}^{N_p}\chi_{h,i}\psi_{h,i}, \tag{2.101}$$

where $\underline{\tilde{\boldsymbol{u}}}_h = (\tilde{\boldsymbol{u}}_{h,i})_{i=1}^{3N_v}$ and $\underline{\chi}_h = (\chi_{h,i})_{i=1}^{N_p}$ are unknown real coefficients. Inserting (2.101) into (2.100) and testing both equations with each basis function ϕ_i, $i = 1, \ldots, 3N_v$, and ψ_i, $i = 1, \ldots, N_p$, respectively, yields

$$\begin{aligned}\left(\frac{3}{2}M_h^{\mathrm{NS}} + \Delta t_n \left[A_h^{\mathrm{NS}} + N_h^{\mathrm{NS}}(\underline{\tilde{\boldsymbol{u}}}_h^{n-1})\right]\right)\underline{\tilde{\boldsymbol{u}}}_h^n &= M_h^{\mathrm{NS}}\left(2\underline{\tilde{\boldsymbol{u}}}_h^{n-1} - \frac{1}{2}\underline{\tilde{\boldsymbol{u}}}_h^{n-2}\right) \qquad (2.102)\\ +\Delta t_n \left(B_h^{\mathrm{NS}}\right)^T \left(\underline{p}_h^{n-1} + \frac{4}{3}\underline{\chi}_h^{n-1} - \frac{1}{3}\underline{\chi}_h^{n-2}\right) &+ \Delta t_n \underline{\boldsymbol{f}}_h^n,\\ P_h^{\mathrm{NS}}\underline{\chi}_h^n &= -\frac{2}{3\Delta t_n}B_h^{\mathrm{NS}}\underline{\tilde{\boldsymbol{u}}}_h^n, \qquad (2.103)\end{aligned}$$

where $\underline{p}_h^{n-1} = \underline{\chi}_h^{n-1} + \underline{p}_h^{n-2}$, the matrices M_h^{NS}, A_h^{NS}, $N_h^{\mathrm{NS}}(\cdot)$, B_h^{NS} are defined as in (2.95)-(2.98), and

$$\left(P_h^{\mathrm{NS}}\right)_{ij} = \left(\nabla\psi_{h,j}, \nabla\psi_{h,i}\right), \quad i,j = 1, \ldots, N_p, \qquad (2.104)$$

$$\boldsymbol{f}_{h,i}^n = \left(\boldsymbol{f}^n, \boldsymbol{\phi}_{h,i}\right), \quad i = 1, \ldots, 3N_v. \qquad (2.105)$$

On the Discrete Inf-Sup Condition

Finite element spaces V_h and Q_h are real Hilbert spaces. Therefore, the abstract theory of the linear saddle point problems, see [45], can be applied to the investigation of the existence and the uniqueness of a solution of the finite element problems of incompressible flow problems. In particular, the spaces V_h and Q_h have to satisfy the discrete inf-sup condition which reads: There is $\beta_h > 0$ such that

$$\inf_{q_h \in Q_h,\, q_h \neq 0}\ \sup_{\boldsymbol{v}_h \in V_h,\, \boldsymbol{v}_h \neq \boldsymbol{0}} \frac{-\left(q_h, \nabla\cdot\boldsymbol{v}_h\right)}{\|\boldsymbol{v}_h\|_{V_h}\|q_h\|_{Q_h}} > \beta_h. \qquad (2.106)$$

For optimal convergence, the constant β_h has to be independent of the mesh width h, see [75, Chapter 3]. It must be pointed out that even for conforming finite element spaces V_h and Q_h the fulfillment of the inf-sup condition (2.81) is not inherited from the spaces V and Q, see details in [75, Chapter 3]. One of the most popular inf-sup stable finite element spaces are the Taylor–Hood finite elements spaces, see [65]. They are given by P_k/P_{k-1}, $k \geq 2$, on triangular and tetrahedral grids and by Q_k/Q_{k-1}, $k \geq 2$, on quadrilateral and hexahedral grids. In particular, due to simplicity of the implementation, the Taylor–Hood finite elements with $k = 2$ are among the most popular finite elements for the simulation of incompressible flows, see [50]. Another popular pair of inf-sup stable finite element spaces is the so-called MINI element introduced in [7]. The basic idea is to start with piecewise linear finite element spaces P_1 for the velocity and the pressure fields, and then to enrich the velocity space such that the discrete inf-sup condition is satisfied. It is the lowest inf-sup stable pair of conforming finite element spaces. For more inf-sup stable finite element spaces and further details, the reader is referred to [50], [75, Chapter 3].

The fulfillment of the discrete inf-sup condition is essential for the linear system of equations (2.94), which is a saddle point problem, in order to ensure the uniqueness of the solution. In the incremental pressure-correction scheme, the projection step, i.e., the second equation of (2.100), serves as stabilization with respect to the discrete inf-sup condition, if the pressure increment is sufficiently large, see [53, 75]. In this case, one does not need necessarily inf-sup stable finite element spaces to perform flow simulations. However, if the stationary flow is reached then the stabilization vanishes. In such a situation spurious oscillations can occur, if V_h/Q_h do not satisfy the discrete inf-sup condition, see [54].

3. Reduced-Order Modeling Based on Proper Orthogonal Decomposition

The focus of this thesis is the investigation of the reduced-order models that are based on the Proper Orthogonal Decomposition. To build such a model, three major steps have to be performed. The first step involves obtaining the so-called snapshots. Their origin could be the result of a scientific experiment or the numerical solution of a parametrized partial differential equation. Within the framework of this thesis, the snapshots are assumed to be the finite element solution of a parabolic partial differential equation (PDE). In Chapter 2, finite element methods were presented for two parabolic PDEs that lay the foundation for the core investigations in this dissertation. In the second step, the reduced-order basis has to be computed by means of Proper Orthogonal Decomposition, which is described in Section 3.1. Finally, a Galerkin reduced-order model is built using that computed reduced-order basis. The presentation of this process takes place in Section 3.2.

3.1. Proper Orthogonal Decomposition

The main idea of the Proper Orthogonal Decomposition (POD) is to provide a low-dimensional basis of a known ensemble of data, e.g., of experimental data or solutions of numerical simulations, which captures the dominant features of the data. The POD was introduced in the context of turbulence in [103] as a method for detecting and analyzing coherent structures in experimental turbulent flows. In other scientific fields, the same procedure is also called Singular Value Decomposition (numerical mathematics), Karhuen–Loève Decomposition (stochastics), Principal Components Analysis (statistical analysis), Empirical Orthogonal Functions (meteorology), and Singular Spectrum Analysis (time series analysis). The interested reader is referred to [63,103,128,141,142] for a more detailed presentation of this method.

3.1.1. Continuous Setting

Let X denote a real separable Hilbert space with the inner product $(\cdot,\cdot)_X$, which induces the norm $\|\cdot\|_X$, and let $[0,T]$ be a finite time interval.

Hypothesis 3.1. *Suppose that for the Hilbert space X the equality*

$$\int_0^T (v,\phi)_X \, dt = \left(\int_0^T v \, dt, \phi \right)_X \tag{3.1}$$

holds for all $v \in L^2(0,T;X)$ *and* $\phi \in X$.

Lemma 3.1. *With Hypothesis 3.1, the identity*

$$\left(\int_0^T v(t,\cdot)\,dt, \int_0^T v(s,\cdot)\,ds\right)_X = \int_0^T\int_0^T (v(t,\cdot), v(s,\cdot))_X\,ds\,dt. \tag{3.2}$$

is true for all $u, v \in L^2(0,T;X)$.

Proof. Using the statement of Hypothesis 3.1, one obtains

$$\begin{aligned}\left(\int_0^T v(t,\cdot)\,dt, \int_0^T v(s,\cdot)\,ds\right)_X &= \int_0^T \left(u(t,\cdot), \int_0^T v(s,\cdot)\,ds\right)_X dt \\ &= \int_0^T \left(\int_0^T v(s,\cdot)\,ds, u(t,\cdot)\right)_X dt \\ &= \int_0^T\int_0^T (u(t,\cdot), v(s,\cdot))_X\,ds\,dt.\end{aligned}$$

□

Let $u(t,x) \in L^2(0,T;X)$ denote a given function. The goal of the POD approach consists in finding an orthonormal basis $\{\phi_i\}_{i=1}^r$ of a subspace of X with dimension $r \in \mathbb{N}$ which solves the problem

$$\operatorname*{arg\,min}_{\phi_1,\ldots,\phi_r} \int_0^T \left\| u(t,x) - \sum_{i=1}^r \xi_i(t)\phi_i(x) \right\|_X^2 dt \quad \text{s.t.} \quad (\phi_i,\phi_j)_X = \delta_{ij}, \tag{3.3}$$

where δ_{ij} denotes the Kronecker delta function and $\xi_i(t) \in \mathbb{R}$, $i = 1,\ldots,r$, $t \in [0,T]$, denote unknown coefficients. Hence, one seeks the best possible approximation of the given function u by an orthonormal basis of dimension r.

Definition 3.1. *The solution* $\{\phi_1,\ldots,\phi_r\}$ *of* (3.3) *is called the POD basis and its elements the POD basis functions or POD modes. The space spanned by the POD basis is called the POD space and is denoted by* $X_r = \operatorname{span}\{\phi_1,\ldots,\phi_r\}$.

By the general Hilbert space theory, it is known that the best approximation of u in X_r at a given time $t \in [0,T]$ can be computed by

$$\sum_{i=1}^r (u,\phi_i)_X\phi_i. \tag{3.4}$$

Note that this is a separated form since the first factor in each term depends only on t and the second one only on x. Thus, the optimization problem (3.3) can be reformulated as

$$\operatorname*{arg\,min}_{\phi_1,\dots,\phi_r} \int_0^T \left\| u(t,x) - \sum_{i=1}^{r} (u,\phi_i)_X \phi_i(x) \right\|_X^2 dt \quad \text{s.t.} \quad (\phi_i,\phi_j)_X = \delta_{ij}. \tag{3.5}$$

Using the orthonormality of the basis functions $\{\phi_i\}_{i=1}^r$, the properties of the inner product and the homogeneity of the integral, one obtains

$$\begin{aligned}
&\operatorname*{arg\,min}_{\phi_1,\dots,\phi_r} \int_0^T \left\| u - \sum_{i=1}^{r} (u,\phi_i)_X \phi_i \right\|_X^2 dt \\
&\quad = \operatorname*{arg\,min}_{\phi_1,\dots,\phi_r} \int_0^T \left(u - \sum_{i=1}^{r} (u,\phi_i)_X \phi_i , u - \sum_{j=1}^{r} (u,\phi_j)_X \phi_j \right)_X dt.
\end{aligned}$$

The first term is a constant for the optimization problem. Altogether, the minimization problem (3.5) is equivalent to the maximization problem

$$\operatorname*{arg\,max}_{\phi_1,\dots,\phi_r} \int_0^T \sum_{i=1}^{r} \left| (u,\phi_i)_X \right|^2 dt \quad \text{s.t.} \quad (\phi_i,\phi_j)_X = \delta_{ij}. \tag{3.6}$$

Problem (3.6) is an optimization problem with constraints. Its Lagrangian has the form

$$\mathcal{L}(\phi_1,\dots,\phi_r,\Lambda) = \int_0^T \sum_{i=1}^{r} \left| (u,\phi_i)_X \right|^2 dt - \sum_{i=1}^{r} \sum_{j=1}^{r} \lambda_{ij} \left[(\phi_i,\phi_j)_X - \delta_{ij} \right],$$

where $\Lambda \in \mathbb{R}^{r\times r}$ with $(\Lambda)_{ij} = \lambda_{ij}$.

First-order necessary optimality conditions are given by

$$\partial_{\phi_i} \mathcal{L} = 0, \quad i = 1,\dots,r, \tag{3.7}$$

$$\partial_{\lambda_{ij}} \mathcal{L} = 0, \quad i,j = 1,\dots,r. \tag{3.8}$$

For any $i = 1, \dots, r$ and $\psi \in X$, the Gateaux derivatives for $\mathcal{L}$ have the form

$$\begin{aligned}
\partial_{\phi_i}\mathcal{L} &= \lim_{\varepsilon\to 0} \frac{\int\limits_0^T \Big[(u,\phi_i+\varepsilon\psi)_X(\phi_i+\varepsilon\psi,u)_X - (u,\phi_i)_X(\phi_i,u)_X\Big]\,dt}{\varepsilon} \\
&\quad - \lim_{\varepsilon\to 0} \frac{\sum\limits_{j=1}^r \lambda_{ij}\Big[(\phi_i+\varepsilon\psi,\phi_j)_X - (\phi_i,\phi_j)_X\Big]}{\varepsilon} \\
&\quad - \lim_{\varepsilon\to 0} \frac{\sum\limits_{j=1}^r \lambda_{ji}\Big[(\phi_j,\phi_i+\varepsilon\psi)_X - (\phi_j,\phi_i)_X\Big]}{\varepsilon} \\
&= \lim_{\varepsilon\to 0} \frac{\int\limits_0^T \Big[(u,\varepsilon\psi)_X(\phi_i,u)_X + (u,\phi_i)_X(\varepsilon\psi,u)_X + (u,\varepsilon\psi)_X(u,\varepsilon\psi)_X\Big]\,dt}{\varepsilon} \\
&\quad - \lim_{\varepsilon\to 0} \frac{\sum\limits_{j=1}^r \Big[\lambda_{ij}(\varepsilon\psi,\phi_j)_X + \lambda_{ji}(\phi_j,\varepsilon\psi)_X\Big]}{\varepsilon} \\
&= 2\int_0^T (\phi_i,u)_X(u,\psi)_X\,dt - \sum_{j=1}^r(\lambda_{ji}+\lambda_{ij})(\phi_j,\psi)_X, \\
\partial_{\lambda_{ij}}\mathcal{L} &= (\phi_i,\phi_j)_X - \delta_{ij}.
\end{aligned} \tag{3.9}$$

With Hypothesis 3.1, the final expression of $\partial_{\phi_i}\mathcal{L}$ in (3.9) can be rewritten $\forall\psi \in X$ as

$$\begin{aligned}
&\left(2\int_0^T (\phi_i,u)_X\,u\,dt, \psi\right)_X - \sum_{j=1}^r(\lambda_{ji}+\lambda_{ij})(\phi_j,\psi)_X \\
&= \left(2\int_0^T (\phi_i,u)_X\,u\,dt - \sum_{j=1}^r(\lambda_{ji}+\lambda_{ij})\phi_j, \psi\right)_X, \quad i = 1,\dots,r,
\end{aligned}$$

where the commutativity of the integral and the inner product was used.

Hence, the first-order optimality condition (3.7) results in

$$\int_0^T (\phi_i,u)_X\; u\,dt = \frac{1}{2}\sum_{j=1}^r(\lambda_{ji}+\lambda_{ij})\phi_j, \quad i = 1,\dots,r, \tag{3.10}$$

with $(\phi_i,\phi_j)_X = \delta_{ij}$, $i,j = 1,\dots,r$.

Lemma 3.2. *Problem* (3.10) *is equivalent to the eigenvalue problem*

$$\int_0^T (\phi_i,u)_X\; u\,dt = \lambda_i\phi_i, \quad i = 1,\dots,r, \tag{3.11}$$

where $\lambda_i = \lambda_{ii}$.

Proof. The statement of this lemma will be proven by induction. For $r = 1$ it holds $i = 1$. Hence, expression (3.10) has the form

$$\int_0^T (\phi_1, u)_X \, u \, dt = \lambda_1 \phi_1.$$

Next, assume that for $r > 1$, the problem (3.10) is equivalent to

$$\int_0^T (\phi_i, u)_X \, u \, dt = \lambda_i \phi_i, \quad i = 1, \dots, r. \tag{3.12}$$

Now it has to be shown that the problem (3.10) for $r + 1$ is equivalent to

$$\int_0^T (\phi_i, u)_X \, u \, dt = \lambda_i \phi_i, \quad i = 1, \dots, r+1. \tag{3.13}$$

Due to assumption (3.12), it only has to be proven that

$$\int_0^T (\phi_{r+1}, u)_X \, u \, dt = \lambda_{r+1} \phi_{r+1}. \tag{3.14}$$

Using the orthonormality of the POD basis, assumption (3.12), expression (3.10), and Hypothesis 3.1 yields for any $i = 1, \dots, r$

$$\begin{aligned} 0 = \lambda_i (\phi_i, \phi_{r+1})_X &= \left(\int_0^T (\phi_i, u)_X \, u \, dt, \phi_{r+1} \right)_X = \int_0^T (u, \phi_i)_X \, (u, \phi_{r+1})_X \, dt \\ &= \left(\int_0^T (\phi_{r+1}, u)_X \, u \, dt, \phi_i \right)_X = \left(\frac{1}{2} \sum_{j=1}^{r+1} (\lambda_{j,r+1} + \lambda_{r+1,j}) \phi_j, \phi_i \right)_X = \frac{1}{2} (\lambda_{i,r+1} + \lambda_{r+1,i}). \end{aligned}$$

It follows that $\lambda_{i,r+1} = -\lambda_{r+1,i}$ for $i = 1, \dots, r$. Consequently, with (3.10) it holds

$$\begin{aligned} \int_0^T (\phi_{r+1}, u)_X \, u \, dt &= \frac{1}{2} \sum_{j=1}^{r} (\lambda_{j,r+1} + \lambda_{r+1,j}) \phi_j + \lambda_{r+1,r+1} \phi_{r+1} \\ &= \frac{1}{2} \sum_{j=1}^{r} (\lambda_{j,r+1} - \lambda_{j,r+1}) \phi_j + \lambda_{r+1,r+1} \phi_{r+1} \\ &= \lambda_{r+1,r+1} \phi_{r+1}. \end{aligned}$$

□

To study the solvability and the properties of (3.11), define an operator $\mathcal{R} : X \to X$ for a given $u \in L^2(0, T; X)$ by

$$\mathcal{R}\phi = \int_0^T (\phi, u)_X \; u \, dt. \tag{3.15}$$

With this operator, the eigenvalue problem in X defined by

$$\mathcal{R}\phi = \lambda\phi \tag{3.16}$$

has the same form as (3.11). Next, the spectral properties of $\mathcal{R}$ will be studied.

Lemma 3.3. *Let $\mathcal{R}$ be the operator given by* (3.15) *and $u \in L^2(0, T; X)$. Then $\mathcal{R}$ is linear and bounded.*

Proof. The linearity of $\mathcal{R}$ follows from the linearity of the inner product $(\cdot, \cdot)_X$ in the first argument. By using Lemma 3.1 and the Cauchy–Schwarz inequality, one obtains

$$\begin{aligned}
\|\mathcal{R}\phi\|_X^2 &= \int_0^T \int_0^T \left(\left((\phi, u)_X \, u \right)(t, \cdot), \left((\phi, u)_X \, u \right)(s, \cdot) \right)_X \, ds \, dt \\
&\leq \int_0^T \int_0^T \left\| \left((\phi, u)_X \, u \right)(t, \cdot) \right\|_X \left\| \left((\phi, u)_X \, u \right)(s, \cdot) \right\|_X \, ds \, dt \\
&= \int_0^T \int_0^T \left| \, (\phi, u(t, \cdot))_X \, \right| \|u(t, \cdot)\|_X \left| \, (\phi, u(s, \cdot))_X \, \right| \|u(s, \cdot)\|_X \, ds \, dt \\
&\leq \|\phi\|_X^2 \int_0^T \int_0^T \|u(t, \cdot)\|_X^2 \, \|u(s, \cdot)\|_X^2 \, ds \, dt \\
&= \|\phi\|_X^2 \int_0^T \|u(t, \cdot)\|_X^2 \, dt \int_0^T \|u(s, \cdot)\|_X^2 \, ds \\
&\leq \|\phi\|_X^2 \, \|u\|_{L^2(0,T;X)}^4,
\end{aligned}$$

yielding that $\mathcal{R}$ is bounded as $u \in L^2(0, T; X)$. □

Lemma 3.4. *Let $\mathcal{R}$ be the linear and bounded operator defined by* (3.15) *and $u \in L^2(0, T; X)$. Then $\mathcal{R}$ has the following properties:*

(i) non-negative, i.e., $(\mathcal{R}\phi, \phi)_X \geq 0, \quad \forall \phi \in X$,

(ii) self-adjoint, i.e, $(\mathcal{R}\phi, \psi)_X = (\phi, \mathcal{R}\psi)_X, \quad \forall \phi, \, \psi \in X$,

(iii) compact.

Proof. In Lemma 3.3 it was shown that $\mathcal{R}$ is linear and bounded. Statement (i) can be verified as follows

$$(\mathcal{R}\phi,\phi)_X = \left(\int_0^T (\phi,u)_X\; u\,dt, \phi\right)_X = \int_0^T (\phi,u)_X\,(u,\phi)_X\; dt = \int_0^T |\,(u,\phi)_X\,|^2\,dt \geq 0.$$

Statement (ii) results similarly from

$$\begin{aligned}(\mathcal{R}\phi,\psi)_X &= \left(\int_0^T (\phi,u)_X\; u\,dt,\psi\right)_X = \int_0^T (\phi,u)_X\,(u,\psi)_X\; dt\\ &= \int_0^T (\psi,u)_X\,(u,\phi)_X\; dt = \left(\int_0^T (\psi,u)_X\,u\,dt,\phi\right)_X\\ &= \left(\phi,\int_0^T (\psi,u)_X\,u\,dt\right)_X = (\phi,\mathcal{R}\psi)_X\,.\end{aligned}$$

(iii) In a separable Hilbert space, which is the case for X, an operator is called compact if it maps weakly convergent sequences into strongly convergent ones, see [87]. Let $\{\phi_n\}$ be an arbitrary weakly convergent sequence in X. By the definition of $\mathcal{R}$, one obtains

$$\lim_{n\to\infty}\mathcal{R}\phi_n = \lim_{n\to\infty}\int_0^T (\phi_n,u)_X\; u\,dt = \int_0^T \lim_{n\to\infty}(\phi_n,u)_X\; u\,dt = \int_0^T (\phi,u)_X\; u\,dt = \mathcal{R}\phi.$$

The second equality holds because of the continuity of the integral and the fact that the integrand is integrable for every element of X. □

Lemma 3.5. *The operator $\mathcal{R}$ defined by* (3.15) *has the following spectral properties:*

(i) *all eigenvalues of $\mathcal{R}$ are non-negative,*

(ii) *eigenfunctions of $\mathcal{R}$ corresponding to different eigenvalues are mutually orthogonal.*

Proof. (i) The non-negativity of the eigenvalues follows directly from the non-negativity of $\mathcal{R}$ (see Lemma 3.4).
(ii) Let λ and β with $\lambda \neq \beta$ be the eigenvalues of $\mathcal{R}$ corresponding to the eigenfunctions ϕ and ψ, respectively. The fact that $\mathcal{R}$ is self-adjoint yields

$$\lambda\,(\phi,\psi)_X = (\mathcal{R}\phi,\psi)_X = (\phi,\mathcal{R}\psi)_X = \beta\,(\phi,\psi)_X\,.$$

Since $\lambda \neq \beta$, it follows that $(\phi,\psi)_X = 0$. □

Now, a theorem will be formulated which is crucial for the solution properties of the eigenvalue problem (3.16) .

Theorem 3.1 (Hilbert–Schmidt). *Let X be a separable Hilbert space. Let $\mathcal{T} : X \to X$ define a bounded, self-adjoint, compact operator. Then there exists a sequence of non-zero real eigenvalues $\{\lambda_i\}_{i\in\mathbb{N}}$ with $\lambda_i \to 0$ as $i \to \infty$, and a complete orthonormal basis $\{\phi_i\}_{i\in\mathbb{N}}$ of X consisting of the corresponding eigenfunctions so that $\mathcal{T}\phi_i = \lambda_i\phi_i$.*

Proof. The reader is referred, e.g., to [117]. □

Corollary 3.1. *There exists a complete orthonormal basis $\{\phi_i\}_{i\in\mathbb{N}}$ of X and a sequence of real non-negative eigenvalues $\{\lambda_i\}_{i\in\mathbb{N}}$ so that*

$$\mathcal{R}\phi_i = \lambda_i\phi_i \quad \text{with} \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq 0, \tag{3.17}$$

where $\lambda_i \to 0$ as $i \to \infty$.

Proof. The statement follows directly from Lemma 3.3, Lemma 3.4, and Lemma 3.5. □

Remark 3.1. The number r which denotes the rank of the POD basis may not exceed the number of positive eigenvalues of $\mathcal{R}$ denoted by $\#\{\lambda_1, \lambda_2, \ldots : \lambda_i > 0\}$. ◁

Each element $v \in X$ can be expressed as

$$v = \sum_{i=1}^{\infty} (v, \phi_i)_X \phi_i.$$

Consequently, the operator $\mathcal{R}$ has the following spectral decomposition

$$\mathcal{R}v = \sum_{i=1}^{\infty} \lambda_i (v, \phi_i)_X \phi_i. \tag{3.18}$$

By now, only the first-order necessary condition (3.7) was considered. It was shown in [140] that there is no second-order sufficient condition for the solution of (3.7). As a result, it is still not clear if the eigenfunctions of the eigenvalue problem (3.11) or, equavalently, (3.17), really solve the optimization problem (3.6).

For $i = 1, 2, \ldots$, consider

$$\begin{aligned}\int_0^T |\,(u, \phi_i)_X\,|^2\, dt &= \int_0^T (\phi_i, u)_X\, (u, \phi_i)_X\; dt = \left(\int_0^T (\phi_i, u)_X\, u\, dt, \phi_i\right)_X \\ &= (\mathcal{R}\phi_i, \phi_i)_X = \lambda_i.\end{aligned}$$

If one uses the orthonormal basis $\{\phi_i\}_{i\in\mathbb{N}}$, the eigenfunctions $\phi_1, \ldots, \phi_r$ corresponding to the r largest eigenvalues yield the largest value of the functional in the optimization problem (3.6) as the eigenvalues are arranged in descending order. In the following theorem the optimality of the POD basis will be assured.

Theorem 3.2 (Optimality). *Let $\{\phi_i\}_{i\in\mathbb{N}}$ and $\{\lambda_i\}_{i\in\mathbb{N}}$ be the eigenfunctions and the corresponding eigenvalues of the eigenvalue problem (3.17) with $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$, and, $r \in \mathbb{N}$, $r \leq \#\{\lambda_1, \lambda_2, \ldots : \lambda_i > 0\}$. Then $\{\phi_i\}_{i=1}^r$ solves the optimization problem (3.6).*

Proof. The proof follows the lines of [140]. Let $\{\phi_i\}_{i=1}^r$ be the eigenfunctions of (3.17) corresponding to the first largest eigenvalues $\{\lambda_i\}_{i=1}^r$, and $\{\tilde{\phi}_i\}_{i=1}^r$ be any other orthonormal set in X of dimension r.

The goal is to show that

$$\int_0^T \sum_{i=1}^r \left|(u,\phi_i)_X\right|^2 dt \geq \int_0^T \sum_{j=1}^r \left|\left(u,\tilde{\phi}_j\right)_X\right|^2 dt. \tag{3.19}$$

The left-hand side can be rewritten as follows, see the proof of Lemma 3.4 (i),

$$\int_0^T \sum_{i=1}^r \left|(u,\phi_i)_X\right|^2 dt = \sum_{i=1}^r (\mathcal{R}\phi_i,\phi_i)_X = \sum_{i=1}^r \lambda_i. \tag{3.20}$$

Since $\{\phi_i\}_{i\in\mathbb{N}}$ is a complete orthonormal basis of X, the following holds

$$\tilde{\phi}_j = \sum_{i=1}^\infty \left(\tilde{\phi}_j,\phi_i\right)_X \phi_i, \quad 1 = \sum_{i=1}^\infty \left|\left(\phi_i,\tilde{\phi}_j\right)_X\right|^2$$

for every $j = 1,\ldots,r$. Accordingly, one obtains

$$\begin{aligned}
\int_0^T \left|\left(u,\tilde{\phi}_j\right)_X\right|^2 dt &= \left(\int_0^T \left(\tilde{\phi}_j,u\right)_X u,\tilde{\phi}_j\right)_X = \left(\mathcal{R}\tilde{\phi}_j,\tilde{\phi}_j\right)_X \\
&= \left(\sum_{i=1}^\infty \lambda_i \left(\tilde{\phi}_j,\phi_i\right)_X \phi_i,\tilde{\phi}_j\right)_X = \sum_{i=1}^\infty \lambda_i \left|\left(\phi_i,\tilde{\phi}_j\right)_X\right|^2 \\
&= \lambda_r \left(1 - \sum_{i=1}^\infty \left|\left(\phi_i,\tilde{\phi}_j\right)_X\right|^2\right) + \sum_{i=1}^\infty \lambda_i \left|\left(\phi_i,\tilde{\phi}_j\right)_X\right|^2 \\
&= \lambda_r + \sum_{i=1}^r (\lambda_i - \lambda_r) \left|\left(\phi_i,\tilde{\phi}_j\right)_X\right|^2 - \sum_{i=r+1}^\infty \underbrace{(\lambda_r - \lambda_i)}_{\geq 0} \left|\left(\phi_i,\tilde{\phi}_j\right)_X\right|^2 \\
&\leq \lambda_r + \sum_{i=1}^r (\lambda_i - \lambda_r) \left|\left(\phi_i,\tilde{\phi}_j\right)_X\right|^2, \quad j = 1,\ldots,r.
\end{aligned}$$

Summing up the last expression for $j = 1,\ldots,r$ yields

$$\int_0^T \sum_{j=1}^r \left| \left(u, \tilde{\phi}_j\right)_X \right|^2 dt \leq \sum_{j=1}^r \left[\lambda_r + \sum_{i=1}^r (\lambda_i - \lambda_r) \left| \left(\phi_i, \tilde{\phi}_j\right)_X \right|^2 \right] \tag{3.21}$$

$$= \sum_{i=1}^r \left[\lambda_r + \underbrace{(\lambda_i - \lambda_r)}_{\geq 0} \underbrace{\sum_{j=1}^r \left| \left(\phi_i, \tilde{\phi}_j\right)_X \right|^2}_{\leq 1} \right] \tag{3.22}$$

$$\leq \sum_{i=1}^r [\lambda_r + (\lambda_i - \lambda_r)] = \sum_{i=1}^r \lambda_i, \tag{3.23}$$

which, together with (3.20), verifies the assertion. □

Remark 3.2. (Role of the POD eigenvalues). With (3.11), one can represent the POD eigenvalues by

$$\lambda_i = (\lambda_i \phi_i, \phi_i)_X = \left(\int_0^T (\phi_i, u)_X \, u \, dt, \phi_i \right)_X = \int_0^T (\phi_i, u)_X \, (\phi_i, u)_X \, dt. \tag{3.24}$$

In the literature, the expression $(u, u)_X$ sometimes refers to the energy of the system. In the case that the function $u(t, x)$ represents the velocity field and $X = L^2(\Omega)$, then the value of $(u, u)_X$ is proportional to the kinetic energy. Let $\{\phi_i\}_{i=1}^{\infty}$ denote the orthonormal basis of X as an extension of the POD basis $\{\phi_i\}_{i=1}^{r}$. Then, by using (3.24) and $u(t, x) = \sum\limits_{i=1}^{\infty} (\phi_i, u)_X \, \phi_i$, one obtains

$$\begin{aligned} \int_0^T (u, u)_X \, dt &= \int_0^T \left(\sum_{i=1}^{\infty} (\phi_i, u)_X \, \phi_i, \sum_{j=1}^{\infty} (\phi_j, u)_X \, \phi_j \right)_X dt \\ &= \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \int_0^T (\phi_i, u)_X \, (\phi_j, u)_X \, (\phi_i, \phi_j)_X \, dt \\ &= \sum_{i=1}^{\infty} \int_0^T (\phi_i, u)_X \, (\phi_i, u)_X \, dt = \sum_{i=1}^{\infty} \lambda_i. \end{aligned} \tag{3.25}$$

Hence, every POD eigenvalue λ_i reflects the respective contribution to the energy of the system. ◁

3.1.2. Discrete Setting

In practice, one wishes to find the best low-dimensional approximation of an ensemble of data. The origin of the data can vary, e.g., representing known numerical solutions or experimental measurements. In case the data represents numerical solutions of a partial differential equation with, e.g., a finite element method, then each element of the given data is a finite element solution evaluated at a particular time instance for time-dependent problems or at a particular parameter value for parameterized problems. In this work, it will be assumed that the data is the finite element solution of a time-dependent partial differential equation.

In what follows, let X denote a Hilbert space with the properties given in Section 3.1.1 and $X_h \subset X$ a finite element space of dimension N which is spanned by a nodal finite element basis consisting of finite element basis functions $\{\varphi_{h,i}\}_{i=1}^{N}$, i.e.,

$$X_h = \operatorname{span}\{\varphi_{h,1}, \dots, \varphi_{h,N}\}.$$

Let $\{u_1, \dots, u_M\} \subset X_h$ denote the given data. Moreover, consider the time grid

$$0 \le t_1 < \ldots < t_M \le T \tag{3.26}$$

of the finite time interval $[0, T]$.

Definition 3.2. *The space X_R spanned by the given ensemble of data, i.e.,*

$$X_R = \operatorname{span}\{u_1, \dots, u_M\},$$

is called the snapshot space, and the elements of the data u_i, $i = 1, \dots, M$, are called snapshots. In general, the snapshots are not linearly independent. Therefore, it holds $R \le M$ with $R = \dim(X_R)$.

Each snapshot u_i can be interpreted as a finite element solution at the i-th time instance $t_i \in [0, T]$

$$u_i(x) = u(t_i, x), \quad i = 1, \dots, M$$

The goal of POD is to find an orthonormal basis $\{\varphi_{\mathrm{ro},1}, \dots, \varphi_{\mathrm{ro},r}\}$, called POD basis, of rank $r \le R$ with

$$\varphi_{\mathrm{ro},i} \in \operatorname{span}\{\varphi_{h,1}, \dots, \varphi_{h,N}\}, \quad i = 1, \dots, r,$$

that provides the best possible approximation of the given snapshots. It can be formulated as an optimization problem similarly to (3.5) in Section 3.1.1

$$\arg \min_{\varphi_{\mathrm{ro},1},\dots,\varphi_{\mathrm{ro},r}} \sum_{m=1}^{M} \omega_m \left\| u_m - \sum_{i=1}^{r} (u_m, \varphi_{\mathrm{ro},i})_X \varphi_{\mathrm{ro},i} \right\|_X^2 \quad \text{s.t.} \quad (\varphi_{\mathrm{ro},i}, \varphi_{\mathrm{ro},j})_X = \delta_{ij}, \tag{3.27}$$

where the integral over time interval is approximated by a quadrature formula with appropriate positive weights ω_m.

Example 3.1. Let the time grid (3.26) be uniform with $\Delta t = \frac{T}{M-1}$. In case of a trapezoidal rule, the weights in (3.27) are equal to

$$\omega_1 = \frac{1}{2}\Delta t, \quad \omega_m = \Delta t, \quad m = 2, \dots, M-1, \quad \omega_M = \frac{1}{2}\Delta t.$$

As the value of Δt is a constant for the optimization problem (3.27) and thus does not influence the solution, it can be set to 1 without loss of generality. ◁

Definition 3.3. *The space X_r spanned by the POD basis, i.e.,*

$$X_r = span\{\varphi_{\mathrm{ro},1}, \ldots, \varphi_{\mathrm{ro},r}\}$$

is called the POD space, and the elements of the POD basis are called POD basis functions or POD modes.

The POD space X_r is a subspace of X_h. Altogether, the following inclusions hold

$$X_r \subset X_R \subset X_h \subset X. \tag{3.28}$$

Usually one has the following dimensional relations

$$r \leq R \leq M \ll N. \tag{3.29}$$

The theoretical results presented in Section 3.1.1 for a continuous setting of POD can now be applied with only minor changes to the discrete case. Consequently, the minimization problem (3.27) is equivalent to the following maximization problem

$$\underset{\varphi_{\mathrm{ro},1},\ldots,\varphi_{\mathrm{ro},r}}{\arg\ \max} \sum_{m=1}^{M} \sum_{i=1}^{r} \omega_m \left|(u_m, \varphi_{\mathrm{ro},i})_X\right|^2 \quad \text{s.t.} \quad (\varphi_{\mathrm{ro},i}, \varphi_{\mathrm{ro},j})_X = \delta_{ij}, \tag{3.30}$$

and the POD modes can be determined by solving the eigenvalue problem

$$\sum_{m=1}^{M} \omega_m \left(\varphi_{\mathrm{ro},i}, u_m\right)_X u_m = \lambda_i \, \varphi_{\mathrm{ro},i}, \quad i = 1, \ldots, r, \tag{3.31}$$

with $\lambda_1 \geq \ldots \geq \lambda_r \geq 0$. Due to the inclusion relation (3.28), one can represent the snapshots and the POD modes with respect to the finite element basis as

$$u_i(x) = \sum_{j=1}^{N} u_{i,j}\varphi_{h,j}(x), \quad i = 1, \ldots, M, \tag{3.32}$$

$$\varphi_{\mathrm{ro},i}(x) = \sum_{j=1}^{N} \varphi_{\mathrm{ro},i,j}\varphi_{h,j}(x), \quad i = 1, \ldots, r, \tag{3.33}$$

with the finite element coefficients vectors $\underline{u}_i = (u_{i,j})_{j=1}^N$ and $\underline{\varphi}_{\mathrm{ro},i} = (\varphi_{\mathrm{ro},i,j})_{j=1}^N$, respectively. Hereinafter, the finite element coefficients vector of a function v will be denoted by $\underline{v}$.

In practice, the coefficients in (3.32) are summarized into a so-called snapshot matrix $U \in \mathbb{R}^{N\times M}$ as follows

$$U = \begin{pmatrix} u_{1,1} & \cdots & u_{M,1} \\ \vdots & \ddots & \vdots \\ u_{1,N} & \cdots & u_{M,N} \end{pmatrix} = \left(\underline{u}_1, \ldots, \underline{u}_M\right), \tag{3.34}$$

where the i-th column corresponds to the finite element coefficients of the discrete solution at the time instance t_i, $i = 1, \ldots, M$, from the time grid (3.26).

The rank of the snapshot matrix U is equal to the dimension of the snapshot space, i.e., $\operatorname{rank}(U) = R$.

Let $G \in \mathbb{R}^{N\times N}$ denote a symmetric, positive definite Gramian matrix which describes the inner product $(\cdot,\cdot)_X$, i.e., for u, $w \in X_h$ it holds

$$(u, w)_X = \underline{u}^T G \underline{w}. \tag{3.35}$$

Example 3.2. Set $X = L^2(\Omega)$, where $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, denotes a bounded domain. It holds then for u, $w \in X_h \subset X$

$$(u, w) = \left(\sum_{j=1}^{N} u_j\varphi_{h,j}, \sum_{i=1}^{N} w_i\varphi_{h,i}\right) = \sum_{j=1}^{N}\sum_{i=1}^{N} u_j w_i \left(\varphi_{h,j}, \varphi_{h,i}\right) = \underline{u}^T M_h \underline{w}, \tag{3.36}$$

where $M_h \in \mathbb{R}^{N\times N}$ is the mass matrix defined in (2.11). Therefore, it holds $G = M_h$.

Similarly, for $X = H_0^1(\Omega)$ one obtains

$$(\nabla u, \nabla w) = \underline{u}^T A_h \underline{w}, \tag{3.37}$$

where $A_h \in \mathbb{R}^{N\times N}$ denotes the stiffness matrix defined in (2.12). In this case, it holds $G = A_h$. ◁

Using notations (3.34) and (3.35), the eigenvalue problem (3.31) can be written in matrix form as an eigenvalue problem in $\mathbb{R}^N$ as

$$UWU^T G \underline{\varphi}_{\mathrm{ro},i} = \lambda_i \underline{\varphi}_{\mathrm{ro},i}, \quad i = 1, \ldots, r, \tag{3.38}$$

where $W = \operatorname{diag}\{\omega_1, \ldots, \omega_M\}$, i.e., W is a diagonal matrix with $(W)_{mm} = \omega_m$, $m = 1, \ldots, M$. Since all the weights ω_m are positive, it holds holds

$$W = W^{1/2}W^{1/2} = W^{1/2}W^{T/2},$$

where $W^{1/2} = \operatorname{diag}\{\sqrt{\omega_1}, \ldots, \sqrt{\omega_M}\}$. Let $\tilde{U}$ be defined by

$$\tilde{U} = UW^{1/2}, \tag{3.39}$$

with

$$\operatorname{rank}(\tilde{U}) = \operatorname{rank}(UW^{1/2}) = \operatorname{rank}(U) = R, \tag{3.40}$$

as $W^{1/2} \in \mathbb{R}^{M\times M}$ is a non-singular matrix. Then, the eigenvalue problem (3.38) can be rewritten as

$$\tilde{U}\tilde{U}^T G \underline{\varphi}_{\mathrm{ro},i} = \lambda_i \underline{\varphi}_{\mathrm{ro},i}, \quad i = 1, \ldots, r. \tag{3.41}$$

Example 3.3. (Trapezoidal rule). For the weights ω_i, $i = 1, \ldots, M$, originating from the trapezoidal rule (see Example (3.1)), one has

$$W^{1/2} = \operatorname{diag}\left\{\frac{1}{\sqrt{2}}, \underbrace{1, \ldots, 1}_{M-2\,\text{times}}, \frac{1}{\sqrt{2}}\right\}.$$

To obtain the matrix $\tilde{U}$, one just has to multiply the first and the last column of U with $\frac{1}{\sqrt{2}}$. ◁

Remark 3.3. Note that (3.41) is an eigenvalue problem in $\mathbb{R}^N$ with N being the dimension of the finite elment basis. Usually N is very large, and therefore the solution of the eigenvalue problem is very expensive. In Section 3.1.3, a more efficient way for the computation of the POD basis will be introduced. ◁

Remark 3.4. In practice, the POD basis is computed not from the snapshot matrix U which consists of the so-called raw snapshots but from a modified one, ensuring that its columns satisfy the homogeneous Dirichlet conditions of the problem. The detailed discussion of this issue is presented in Section 3.2.3. ◁

3.1.3. Method of Snapshots

The straightforward way to compute the POD modes is to solve the eigenvalue problem (3.41) in $\mathbb{R}^N$. However, in practice it can be very expensive since N, the dimension of the finite element basis, is usually very large. An alternative way to obtain the POD basis was introduced in [128], known as the method of snapshots. The basic idea is to transform the large eigenvalue problem (3.41) in $\mathbb{R}^N$ into a much smaller one in $\mathbb{R}^M$ (as $M \ll N$) by applying some algebraic manipulations. As a result, the POD basis functions can be obtained with much less computational effort.

Multiplication of (3.41) with $\tilde{U}^T G$ from the left-hand side yields an eigenvalue problem in $\mathbb{R}^M$

$$\tilde{U}^T G \tilde{U} \underline{\xi}_i = \lambda_i \underline{\xi}_i, \quad i = 1, \ldots, r, \tag{3.42}$$

with $\underline{\xi}_i := \tilde{U}^T G \underline{\varphi}_{\text{ro},i}$. The matrix $G \in \mathbb{R}^{N \times N}$ defined by (3.35) is symmetric and positive definite. Thus, it holds $G = G^{T/2} G^{1/2}$ and the matrix $\tilde{U}^T G \tilde{U}$ can be rewritten as

$$\tilde{U}^T G \tilde{U} = \left(G^{1/2}\tilde{U}\right)^T \left(G^{1/2}\tilde{U}\right). \tag{3.43}$$

Due to the equality (3.43), one can easily verify that $\tilde{U}^T G \tilde{U}$ is a symmetric, positive semi-definite matrix with $\operatorname{rank}(\tilde{U}^T G \tilde{U}) = R$. Therefore, the eigenvalue problem (3.42) is solvable and its solution consists of real, non-negative eigenvalues λ_i, $i = 1, \ldots, r$, with $\lambda_1 \geq \ldots \geq \lambda_r$, and corresponding orthogonal eigenvectors $\underline{\xi}_i$, $i = 1, \ldots, r$.

Now, multiplication of (3.42) with $\underline{\xi}_i^T$ from the left-hand side results in $\|\tilde{U}\underline{\xi}_i\|_X = \sqrt{\lambda_i(\underline{\xi}_i^T \underline{\xi}_i)}$. By setting

$$\underline{\varphi}_{\mathrm{ro},i} := \frac{\tilde{U}\underline{\xi}_i}{\|\tilde{U}\underline{\xi}_i\|_X} = \frac{1}{\sqrt{\lambda_i(\underline{\xi}_i^T \underline{\xi}_i)}}\tilde{U}\underline{\xi}_i = \frac{1}{\sqrt{\lambda_i}}\tilde{U}\frac{\underline{\xi}_i}{\sqrt{\underline{\xi}_i^T \underline{\xi}_i}}, \quad i = 1, \ldots, r, \tag{3.44}$$

one obtains the eigenvectors of (3.41). This can be verified by inserting (3.44) into (3.41) and using the definition of $\underline{\xi}_i$ as follows

$$\begin{aligned}
\tilde{U}\tilde{U}^T G\underline{\varphi}_{\mathrm{ro},i} &= \tilde{U}\tilde{U}^T G\left(\frac{1}{\sqrt{\lambda_i(\underline{\xi}_i^T \underline{\xi}_i)}}\tilde{U}\underline{\xi}_i\right)\\
&= \frac{1}{\sqrt{\lambda_i\left(\underline{\varphi}_{\mathrm{ro},i}^T G^T \tilde{U}\tilde{U}^T G\underline{\varphi}_{\mathrm{ro},i}\right)}}\tilde{U}\tilde{U}^T G\tilde{U}\tilde{U}^T G\underline{\varphi}_{\mathrm{ro},i}\\
&= \frac{\lambda_i^2\underline{\varphi}_{\mathrm{ro},i}}{\sqrt{\lambda_i^2\underbrace{\left(\underline{\varphi}_{\mathrm{ro},i}^T G\underline{\varphi}_{\mathrm{ro},i}\right)}_{=1}}} = \lambda_i\underline{\varphi}_{\mathrm{ro},i}.
\end{aligned}$$

Altogether, the method of snapshots employs the following algorithm:

1. Solve (3.42) and obtain $\underline{\xi}_i$, $i = 1, \ldots, r$.
2. Recover the POD basis coefficients $\underline{\varphi}_{\mathrm{ro},i}$ by (3.44) from $\underline{\xi}_i$, $i = 1, \ldots, r$.

3.1.4. Connection with Singular Value Decomposition

In this section the relationship between POD and Singular Value Decomposition (SVD) will be discussed.

Theorem 3.3 (Singular Value Decomposition). *Let $S \in \mathbb{R}^{n\times m}$ denote an arbitrary real matrix with* $\operatorname{rank}(S) = d \leq \min\{m, n\}$. *Then there exist orthogonal matrices $V_1 \in \mathbb{R}^{n\times n}$ and $V_2 \in \mathbb{R}^{m\times m}$ and real numbers $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_d > 0$ and such that*

$$V_1^T S V_2 = \begin{pmatrix} D & 0\\ 0 & 0\end{pmatrix} =: \Sigma \in \mathbb{R}^{n\times m}, \tag{3.45}$$

where $D = \operatorname{diag}\{\sigma_1, \ldots, \sigma_d\} \in \mathbb{R}^{d\times d}$ and zeros in (3.45) *denote zero matrices of appropriate dimensions.*
Moreover, the columns $\{\underline{v}_{1,i}\}_{i=1}^n$ of V_1, the so-called left-singular vectors of S, and the columns $\{\underline{v}_{2,i}\}_{i=1}^m$ of V_2, the so-called right-singular vectors of S, are the eigenvectors of SS^T and $S^T S$, respectively. The non-zero eigenvalues of SS^T and $S^T S$ are the squares of $\sigma_1, \ldots, \sigma_d$, which are the non-zero singular values of S.

Proof. E.g., see [123]. □

Multiplication of (3.41) with $G^{1/2}$ from the left-hand side and using the fast that $G = G^{T/2}G^{1/2}$ yields

$$G^{1/2}\tilde{U}\tilde{U}^T G^{T/2}G^{1/2}\underline{\varphi}_{\mathrm{ro},i} = \lambda_i G^{1/2}\underline{\varphi}_{\mathrm{ro},i}, \quad i = 1, \ldots, r, \tag{3.46}$$

or

$$\hat{U}\hat{U}^T\underline{\hat{\varphi}}_{\mathrm{ro},i} = \lambda_i\underline{\hat{\varphi}}_{\mathrm{ro},i}, \quad i = 1, \ldots, r, \tag{3.47}$$

where

$$\hat{U} = G^{1/2}\tilde{U} = G^{1/2}UW^{1/2} \quad \text{and} \quad \underline{\hat{\varphi}}_{\mathrm{ro},i} = G^{1/2}\underline{\varphi}_{\mathrm{ro},i}. \tag{3.48}$$

Let $\Phi \in \mathbb{R}^{N\times R}$ denote the so-called POD matrix columnwise consisting of the finite element coefficients vectors of the POD modes, i.e.,

$$\Phi = \left(\underline{\varphi}_{\mathrm{ro},1}, \ldots, \underline{\varphi}_{\mathrm{ro},r}\right), \tag{3.49}$$

and let $\hat{\Phi} \in \mathbb{R}^{N\times R}$ be the matrix with the vectors $\{\underline{\hat{\varphi}}_{\mathrm{ro},i}\}_{i=1}^r$ as columns defined in (3.48), i.e.,

$$\hat{\Phi} = \left(\underline{\hat{\varphi}}_{\mathrm{ro},1}, \ldots, \underline{\hat{\varphi}}_{\mathrm{ro},r}\right) = G^{1/2}\Phi. \tag{3.50}$$

According to Theorem 3.3, the eigenvectors $\{\underline{\hat{\varphi}}_{\mathrm{ro},i}\}_{i=1}^r$ of $\hat{U}\hat{U}^T$ in (3.47) are the left-singular vectors of $\hat{U}$. Thus, instead of solving the eigenvalue problem (3.41) or applying the method of snapshots ((3.42) together with (3.44)), one can compute the SVD of $\hat{U}$ to determine the POD basis.

Consequently, the POD basis $\{\varphi_{\mathrm{ro},i}\}_{i=1}^r$ can be computed by applying the following algorithm:

1. Compute SVD of $\hat{U}$, which results in $\hat{U} = \hat{\Phi}\Sigma\hat{\Xi}^T$.
2. Recover the POD basis coefficient vectors $\underline{\varphi}_{\mathrm{ro},i}$, $i = 1, \ldots, r$, from the first r columns of $\hat{\Phi}$ by computing

$$\underline{\varphi}_{\mathrm{ro},i} = G^{-1/2}\underline{\hat{\varphi}}_{\mathrm{ro},i}. \tag{3.51}$$

3.1.5. Practical Aspects

In practice, the POD basis is usually computed with the method of snapshots, see Section 3.1.3, by solving the eigenvalue problem (3.42) or by building the SVD of a slighly modified snapshot matrix, see Section 3.1.4. Numerous computer software packages are available to carry out these procedures. For instance, the linear algebra module numpy.linalg for Python and MATLAB provide several routines to solve eigenvalue problems and compute the SVD. Both are based on the routines of the LAPACK package. In this thesis, the POD modes are computed with the method of snapshots in the C/C++

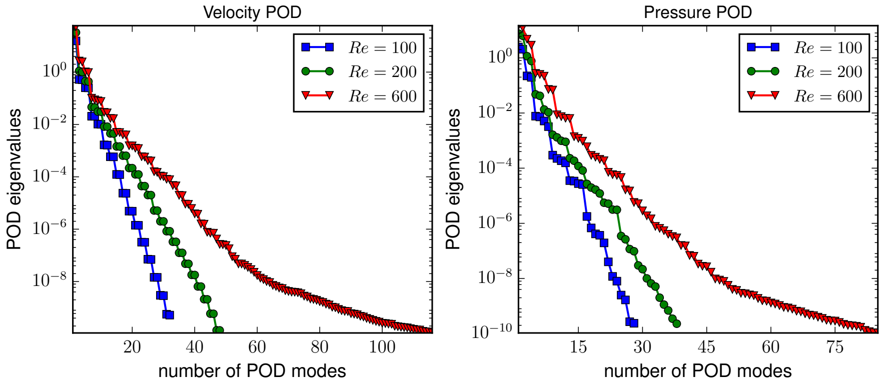

Figure 3.1.: Velocity (right) and pressure (left) POD eigenvalues for flow around a cylinder problem (Example 5.1) for different values of the Reynolds number.

code MooNMD [78] using the routine *dsyev_* from the LAPACK library. The lower threshold value for the eigenvalues is set to 10^{-10}.

In applications, the POD basis serves as a low-dimensional basis of the reduced-order model, see Section 3.2.1. One of the important parameters for the accuracy of the model is the dimension of the used POD basis. In the literature, there exists only a heuristic approach for the choice of the rank of the POD basis. It is based on the ratio of the modeled energy to the total energy of the system, the so-called energy ratio, denoted by $\mathcal{E}(r)$. In Remark 3.2 it was shown that for every i, the POD eigenvalues λ_i reflect the ith fraction of the system's energy. Therefore, the energy ratio $\mathcal{E}(r)$ is given by

$$\mathcal{E}(r) = \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{R} \lambda_i}, \tag{3.52}$$

where λ_i are the POD eigenvalues from (3.41). For example, if one wishes the POD basis to contain at least $99,9\%$ of the total energy, then r is chosen to be the smallest integer such that $\mathcal{E}(r) \geq 0.999$.

POD is most effective for problems for which the POD eigenvalues λ_i, $i = 1, \ldots, r$, decrease rapidly. In this case only a few POD modes are needed to approximate the elements from the snapshot space with good accuracy. Otherwise, a lot more POD modes are required.

In Figure 3.1, the distribution of the POD eigenvalues is depicted (semilogarithmic plot) for the flow around a cylinder problem, see Example 5.1, at different values of the Reynolds number ($Re = 100, 200, 600$). In all cases the flow is laminar. One can see that with an increasing value of Re, the POD eigenvalues decrease slower, i.e., the higher the values of the Reynolds number, the more POD modes are needed to capture a certain energy level of the system. For convection-dominated problems, the POD eigenvalues decrease slowly, e.g., see Example 4.2 or [137] for the decay of the POD eigenvalues in the case of a turbulent flow.

3.2. Galerkin Reduced-Order Modeling

This section presents how the POD basis (see Section 3.1) can be utilized to build a projection-based reduced-order model using the Galerkin projection. Moreover, such practical aspects as treatment of the initial and boundary conditions as well as efficient implementation are addressed.

3.2.1. Galerkin Projection

Galerkin projection, or Galerkin method, introduced by the Russian mathematician and engineer Boris Galerkin (1871-1945), is an approach that approximates an infinite-dimensional partial differential equation into a system of ordinary differential equations by projecting the equation into a finite-dimensional space. The Galerkin method is closely connected with the weak formulation of the partial differential equation. One of the popular applications of the approach is the Galerkin finite element method, see Sections 2.1.2 and 2.2.7 for its application to the time-dependent scalar convection-diffusion-reaction equation and the time-dependent incompressible Navier–Stokes equations, respectively. In a similar way, one can also derive a so-called projection-based or Galerkin reduced-order model (ROM) using the r-dimensional POD basis, see Section 3.1.

For the sake of simplicity, the derivation of the Galerkin ROM based on POD will be carried out for a prototypical example of a parabolic partial differential equation, the heat equation. It is the special case of the convection-diffusion-reaction equation (2.1) with zero convection and reaction fields, i.e., $\boldsymbol{b} = \boldsymbol{0}$ and $c = 0$, and $\varepsilon = 1$. The heat equation equipped with the non-homogeneous Dirichlet boundary condition and the initial condition is given by

$$\begin{aligned} \partial_t u - \Delta u &= f && \text{in} \quad (0,T] \times \Omega, \\ u(t,\boldsymbol{x}) &= g_{\mathrm{D}}(t,\boldsymbol{x}) && \text{on} \quad [0,T] \times \Gamma_{\mathrm{D}} = \Gamma, \\ u(0,\boldsymbol{x}) &= u^0(\boldsymbol{x}) && \text{in} \quad \Omega, \end{aligned} \tag{3.53}$$

where $\Omega \subset \mathbb{R}^d$, $d \in \{2,3\}$, is a bounded domain with Lipschitz boundary Γ.

Assume that $g_{\mathrm{D}}(t,\cdot) \in H^{1/2}(\Gamma_{\mathrm{D}})$ for all $t \in [0,T]$. By the trace theorem (e.g., see [1]), there exists $u_g(t,\cdot) \in H^1(\Omega)$, which is an extension of g_{D} into Ω. Furthermore, assume that $f(t,\cdot) \in L^2(\Omega)$ and $u^0 \in H^1(\Omega)$ with $u^0 - u_g(0,\cdot) \in H_0^1(\Omega)$ for all $t \in [0,T]$.

The time-continuous weak formulation of problem (3.53) has the following form: Find $u : (0,T] \to H^1(\Omega)$ with $u(t,\cdot) - u_g(t,\cdot) \in H_0^1(\Omega)$ for all $t \in [0,T]$, such that

$$(\partial_t u, v) + (\nabla u, \nabla v) = (f, v)\,, \quad \forall v \in H_0^1(\Omega). \tag{3.54}$$

The second term on the left-hand side is obtained from the Laplace term by applying integration by parts and the Gaussian theorem, see (2.4).

In the framework of the Galerkin method, instead of $H_0^1(\Omega)$, one considers in (3.54) a finite-dimensional subspace of $H_0^1(\Omega)$. By choosing this subspace to be a finite element space $X_h \subset H_0^1(\Omega)$, one recovers the Galerkin finite element formulation for (3.54).

Substitution of $H_0^1(\Omega)$ by the POD space $X_r \subset H_0^1(\Omega)$, spanned by the POD basis $\{\varphi_{\text{ro},i}\}_{i=1}^r$, see Section 3.1, results in the Galerkin reduced-order model (G-ROM).

Let $u_{\text{ro}}(t,\boldsymbol{x})$ denote the reduced-order approximation of the solution function $u(t,\boldsymbol{x})$ in X_r by the POD basis $\{\varphi_{\text{ro},i}\}_{i=1}^r$

$$u(t,\boldsymbol{x}) \approx u_{\text{ro}}(t,\boldsymbol{x}) = u_g(t,\boldsymbol{x}) + u_r(t,\boldsymbol{x}) = u_g(t,\boldsymbol{x}) + \sum_{i=1}^{r} \alpha_i(t)\varphi_{\text{ro},i}(\boldsymbol{x}), \tag{3.55}$$

where α_i, $i = 1,\dots,r$, are the unknown ROM coefficients that have to be determined, and $u_g(t,\boldsymbol{x})$ is the extension of the Dirichlet boundary condition into the domain Ω.

Finally, the reduced-order model based on POD associated with (3.54) reads as follows: Find $u_r = u_{\text{ro}} - u_g \colon (0,T] \to X_r$ such that

$$(\partial_t u_r, \varphi_{\text{ro},i}) + (\nabla u_r, \nabla\varphi_{\text{ro},i}) = (f, \varphi_{\text{ro},i}) - (\partial_t u_g, \varphi_{\text{ro},i}) - (\nabla u_g, \nabla\varphi_{\text{ro},i}), \tag{3.56}$$

with $i = 1,\dots,r$. Using the definition (3.55), the Galerkin ROM (3.56) can be written in matrix form as follows: Find $\underline{\alpha} = (\alpha_i)_{i=1}^r : (0,T] \to \mathbb{R}^r$ such that

$$M_{\text{ro}}\dot{\underline{\alpha}} + A_{\text{ro}}\underline{\alpha} = \underline{f}_{\text{ro}} + \underline{l}_{\text{ro}}, \tag{3.57}$$

where the dot over $\underline{\alpha}(t)$ denotes the time derivative, and

$$(M_{\text{ro}})_{ij} = (\varphi_{\text{ro},j}, \varphi_{\text{ro},i}), \quad i,j = 1,\dots,r, \tag{3.58}$$

$$(A_{\text{ro}})_{ij} = (\nabla\varphi_{\text{ro},j}, \nabla\varphi_{\text{ro},i}), \quad i,j = 1,\dots,r, \tag{3.59}$$

$$f_{\text{ro},i} = (f, \varphi_{\text{ro},i}), \quad i = 1,\dots,r, \tag{3.60}$$

$$l_{\text{ro},i} = -(\partial_t u_g, \varphi_{\text{ro},i}) - (\nabla u_g, \nabla\varphi_{\text{ro},i}), \quad i = 1,\dots,r, \tag{3.61}$$

with the initial condition obtained by one of the approaches presented in Section 3.2.2.

The ROM (3.57) is thus a system of r ordinary differential equations (ODEs). In practical applications, the dimension of the POD basis r should be chosen to be small, e.g., of order $\mathcal{O}(10)$, to obtain efficient ROM simulations by applying any ODE solver. The solution consists of the ROM coefficients $\underline{\alpha}(t)$, from which the reduced-order approximation $u_{\text{ro}}(t,x)$ in (3.55) can be constructed.

Note that for (3.57), the POD basis must satisfy homogeneous Dirichlet boundary conditions on the Dirichlet boundary as $X_r \subset H_0^1(\Omega)$. In Section 3.1, the presented algorithms for the computation of the POD basis were carried out for a general case disregarding any boundary conditions. A detailed discussion about the treatment of different Dirichlet boundary conditions within the framework of POD will be carried out in Section 3.2.3.

3.2.2. Initial Condition

In the literature, the intital condition $\underline{\alpha}^0$ for a projection-based reduced-order model based on the POD like (3.57) is usually obtained by projecting $u^0 - u_g^0$ in the L^2 sense onto the POD basis as follows

$$\alpha_i^0 = \left(u^0 - u_g^0, \varphi_{\text{ro},i}\right), \quad i = 1,\dots,r, \tag{3.62}$$

where $u_g^0 = (u_g(t,\cdot))$. Consequently, the reduced-order approximation of the initial condition u^0 has the form

$$u^0 \approx u_{\text{ro}}^0 = u_g^0 + \sum_{i=1}^{r} \alpha_i^0 \varphi_{\text{ro},i}. \tag{3.63}$$

It represents the best possible approximation of the given initial condition u^0 in the POD space X_r in the L^2 sense. Unless noted otherwise, any projection-based ROM in the thesis will be equipped with this initial condition.

However, there might be different goals than the optimality of the ROM approximation of the initial condition in the L^2 sense. Depending on the origin of the POD basis, the initial condition (3.62), although optimal in the L^2 sense, can be polluted by spurious oscillations, e.g., see the top row of Figure 4.28. A good quality initial condition is crucial for the numerical methods to produce accurate solutions. Therefore, it is desirable to be able to construct a ROM initial condition that suppresses spurious oscillations as good as possible but still approximates well the function u^0.

A possible way to achieve this goal originates from turbulence modeling. Turbulent flows contain tiny eddies that cannot be resolved by the realistic computational meshes. For that reason, the idea of the Large Eddy Simulation (LES) is to replace the solution $\boldsymbol{u}$ by some kind of the spatial average $\overline{\boldsymbol{u}}$, e.g., obtained by the convolution of the solution with the Gaussian filter function. Averaging the Navier–Stokes equations (2.40)-(2.41) defined in $\mathbb{R}^d$ yields the Navier–Stokes equations for the averaged solution with the additial term $\nabla \cdot \overline{\boldsymbol{u}\boldsymbol{u}^T}$, which has to be modeled. The approximation of the additional term is called the closure model. One of the approaches to model $\nabla \cdot \overline{\boldsymbol{u}\boldsymbol{u}^T}$ is based on the approximation of the Fourier transform of the Gaussian function using a rational approximation resulting in the rational LES, see [41, 75]. The closure model includes an operator that describes an elliptic, second order Helmholtz equation to be solved

$$-\mu^2 \Delta \overline{\boldsymbol{u}} + \overline{\boldsymbol{u}} = \boldsymbol{u}, \tag{3.64}$$

where μ is the filter width usually chosen to be $\mu = \mathcal{O}(h)$. In turbulence modeling, the problem (3.64) is called differential filter. It represents an approximation of the convolution operator with the Gaussian filter in $\mathbb{R}^d$. The differential filter is also utilized in further turbulence models such as Approximate Deconvolution Models and the Leray α-Model, see [75] for more details.

Thus, the ROM initial condition (3.63) can be filtered in a post-processing step by computing the Galerkin approximation of the Helmholtz equation (3.64) with respect to the POD basis $\{\varphi_{\text{ro},1}, \ldots, \varphi_{\text{ro},r}\}$. Finally, the following problem has to be solved: Find $\tilde{u}_{\text{ro}}^0$ with $\tilde{u}_{\text{ro}}^0 - u_g^0 = \sum_{i=1}^{r} \tilde{\alpha}_i^0 \varphi_{\text{ro},i} \in X_r$ such that

$$\mu^2 \left(\nabla \tilde{u}_{\text{ro}}^0, \nabla \varphi_{\text{ro},i}\right) + \left(\tilde{u}_{\text{ro}}^0, \varphi_{\text{ro},i}\right) = \left(u_{\text{ro}}^0, \varphi_{\text{ro},i}\right), \quad i = 1, \ldots, r, \tag{3.65}$$

where u_{ro}^0 is the ROM approximation (3.63).

Next, the convergence of the filtered ROM initial condition for the special case of a family of uniform triangulations will be investigated. Using the triangle inequality yields

$$\left\|u^0 - \tilde{u}^0_{\mathrm{ro}}\right\|_0 \leq \left\|u^0 - u^0_{\mathrm{ro}}\right\|_0 + \left\|u^0_{\mathrm{ro}} - \tilde{u}^0_{\mathrm{ro}}\right\|_0. \tag{3.66}$$

The first term on the right-hand side can be expected to be small by the construction of u^0_{ro}. In order to obtain an estimation of the second term on the right-hand side, the difference $u^0_{\mathrm{ro}} - \tilde{u}^0_{\mathrm{ro}}$ has to be employed as a test function in (3.65). By shifting the second term on the left-hand side to the right-hand side of the equation, and by using the Cauchy–Schwarz inequality as well as the global inverse estimate, cf. (4.13), one obtains

$$\begin{aligned}\left\|u^0_{\mathrm{ro}} - \tilde{u}^0_{\mathrm{ro}}\right\|_0^2 &\leq \mu^2 \left\|\nabla \tilde{u}^0_{\mathrm{ro}}\right\|_0 \left\|\nabla \left(u^0_{\mathrm{ro}} - \tilde{u}^0_{\mathrm{ro}}\right)\right\|_0 \\ &\leq C h^{-1} \mu^2 \left\|\nabla \tilde{u}^0_{\mathrm{ro}}\right\|_0 \left\|u^0_{\mathrm{ro}} - \tilde{u}^0_{\mathrm{ro}}\right\|_0.\end{aligned} \tag{3.67}$$

It must be noted that the inverse estimate is applicable as it is assumed that $X_r \subset X_h$. With $\mu = \mathcal{O}(h)$, it holds

$$\left\|u^0_{\mathrm{ro}} - \tilde{u}^0_{\mathrm{ro}}\right\|_0 = \mathcal{O}(h). \tag{3.68}$$

On the one hand, the filtering procedure (3.65) yields the solution that does not represent the best possible approximation of u^0 in the L^2 sense anymore but because of (3.68) the function $\tilde{u}^0_{\mathrm{ro}}$ is still a good approximation of u^0 with the convergence of at least first order in the L^2 sense. On the other hand, $\tilde{u}^0_{\mathrm{ro}}$ can lead to a better approximation of u^0 in the sense of other quantities of interest taking advantage of the suppression of spurious oscillations.

To the best of the author's knowledge, the utilization of the differential filter in the post-processing step to the ROM initial condition computed by (3.62) is new. In Figure 4.28, the plots in the bottom line show the results of the filtering procedure (3.65) applied to the initial conditions depicted in the top row. In Section 4.4, the filtered initial condition will be employed to investigate its impact on the results of the ROM simulations.

3.2.3. Boundary Conditions

In Section 3.2.1 it was described how a projection-based Galerkin ROM (3.57) based on POD can be obtained by applying the Galerkin projection to the weak formulation of the underlying PDE. Note that the POD basis functions that utilized as ansatz and test functions have to fulfill homogeneous Dirichlet boundary conditions.

In Section 3.1, the algorithms for the computation of the POD basis are based on snapshots that do not necessarily vanish on the Dirichlet boundary. In fact, every POD mode $\varphi_{\mathrm{ro},i}$ can be interpreted as a particular linear combination of the snapshots, see (3.44), yielding that the POD modes computed from the raw snapshots representing the finite element solution are in general non-zero on the Dirichlet boundary. Hence, some attention has to be paid to the treatment of the boundary conditions within the framework of the computation of the POD basis and building a ROM in order to obtain POD modes that are zero on the Dirichlet boundary.

There are several suggestions in the literature on how to deal with inhomogeneous Dirichlet boundary conditions. First, the treatment of steady Dirichlet boundary conditions will be depicted, and subsequently three approaches for the time-dependent case will be presented. As Neumann/outflow boundary conditions are usually homogeneous in applications, no attention is paid to this type of boundary conditions.

Steady Dirichlet Boundary Conditions

Let the snapshots $\{u_m\}_{m=1}^M$ represent the finite element solution of the problem with the steady Dirichlet boundary condition

$$u(t,\boldsymbol{x}) = g(\boldsymbol{x}) \quad \text{on} \quad [0,T] \times \Gamma_{\mathrm{D}} \subset \Gamma. \tag{3.69}$$

In this case, the POD is applied not to the raw snapshots $\{u_m\}_{m=1}^M$ like it is presented in (3.27), but to the modified snapshots $\{u_m - u_g\}_{m=1}^M$, where u_g is an extension of g into Ω which fulfills the boundary condition (3.69). Afterwards this function has to be added to the reduced-order approximation of u as it is done in (3.55). In the literature, the most popular choice for u_g is the time average of the snapshots, i.e.,

$$u_g := \bar{u}_h = \frac{1}{M}\sum_{i=1}^{M} u_i. \tag{3.70}$$

Thus, the POD modes are computed from the snapshots' fluctuations. This approach is often called the centered-trajectory method. For flow problems, using the average of the snaphsots for u_g has a big advantage as it preserves the linear properties of the solution such as divergence in the case of the velocity field.

The modified snapshots $\{u_m - u_g\}_{m=1}^M$ satisfy homogeneous Dirichlet boundary conditions. Due to (3.44), the POD is a linear procedure. Consequently, the POD basis functions also satisfy homogeneous Dirichlet boundary conditions and can be used to build a ROM.

Time-Dependent Dirichlet Boundary Conditions

In the literature one can find several suggestions on how to treat time-dependent Dirichlet conditions in the ROM applications, e.g., see [20, 47, 48, 56, 116]. In all cases it is assumed that the Dirichlet boundary conditions are given in the separated form, i.e., they can be expressed as a product of two functions depending only on space and on time, respectively. In this section, the following Dirichlet boundary condition will be considered

$$u(t,\boldsymbol{x}) = \gamma_k(t) g_k(\boldsymbol{x}) \quad \text{on} \quad (0,T] \times \Gamma_{\mathrm{D},k}, \quad k = 1,\ldots,K, \tag{3.71}$$

where $\Gamma_{\mathrm{D}} = \Gamma_{\mathrm{D},1} \cup \ldots \cup \Gamma_{\mathrm{D},K}$, $\Gamma_{\mathrm{D},i} \cap \Gamma_{\mathrm{D},j} = \emptyset$ for $i \neq j$, $i,j = 1,\ldots,K$. Note that the steady-state Dirichlet boundary condition is a special case of (3.71), i.e., γ_k is a constant for all $k = 1,\ldots,K$. Three methods for the treatment of the boundary condition of type (3.71) will be presented.

Method 1. The method presented here is often used in the literature for $K = 1$ in (3.71), e.g., see [20,25,116]. To the best of the author's knowledge, its generalization for $K \geq 1$ was initially introduced in [56] for the Navier–Stokes equations.

Similarly to the time-independent case presented in Section (3.2.3), the goal is to obtain a POD basis, which is zero on the Dirichlet boundary. For this purpose, the original snapshots $u_1, \dots, u_M$ have to be modified. The following algorithm is suggested in [56]:

1. Choose K vectors $\underline{\beta}_k \in \mathbb{R}^K$, $k = 1, \dots, K$. For the sake of simplicity, the vectors will be chosen here such that

$$\beta_{k,j} = \delta_{jk}, \quad j, k = 1, \dots, K,$$

where δ_{jk} denotes the Kronecker delta function, see [56] for a detailed description of the choice of $\underline{\beta}_k$, $k = 1, \dots, K$.

2. Generate K solutions $u_{S,k}$, $k = 1, \dots, K$, that satisfy the Dirichlet boundary condition

$$u_{S,k}(\boldsymbol{x}) = \beta_{k,j} g_k(\boldsymbol{x}) \quad \text{on} \quad [0,T] \times \Gamma_{\mathrm{D},k}, \quad j = 1, \dots, K. \tag{3.72}$$

Such solutions can be obtained, e.g., by solving the steady-state version of the PDE by the finite element method with the boundary condition (3.72), or by taking the solution of the time-dependent PDE at a particular time step such that (3.72) is fulfilled.

3. Modify the snapshots $u_1, \dots, u_M$ by

$$u_i(\boldsymbol{x}) - \sum_{k=1}^{K} \gamma_k(t_i) u_{S,k}(\boldsymbol{x}), \quad i = 1, \dots, M, \tag{3.73}$$

where the known functions γ_k are evaluated at the same time instances as the corresponding snapshots. The modified snapshots (3.73) fulfill homogeneous Dirichlet boundary conditions on the entire Dirichlet boundary Γ_{D}.

4. Compute POD basis functions $\{\varphi_{\mathrm{ro},i}\}_{i=1}^{r}$ as described in Section 3.1.3 or 3.1.4.

Finally, the computed POD modes can be used to build the ROM as described in Section 3.2.1. Moreover, the function u_g in the reduced-order approximation (3.55) is defined by

$$u_g(t, \boldsymbol{x}) = \sum_{k=1}^{K} \gamma_k(t) u_{S,k}(\boldsymbol{x}).$$

Remark 3.5. For many applications one has Dirichlet boundary conditions of type (3.71) with $K = 2$ such that

$$u(t, \boldsymbol{x}) = \begin{cases} \gamma_1(t) g_1(\boldsymbol{x}) & \text{on} \quad [0,T] \times \Gamma_{\mathrm{D},1}, \\ g_2(\boldsymbol{x}) & \text{on} \quad [0,T] \times \Gamma_{\mathrm{D},2}, \end{cases} \tag{3.74}$$

i.e., the boundary condition on $\Gamma_{\mathrm{D},2}$ is time-independent with $\gamma_2(t) = 1$. In this case, one can run the above algorithm for $K = 2$ generating additionally two solutions $u_{S,1}$ and $u_{S,2}$ by solving the fully discrete system as described in the algorithm. Alternatevely, one can apply a similar approach for the steady Dirichlet boundary condition on $\Gamma_{\mathrm{D},2}$ as presented in Section 3.2.3. For this sake, define $u_{S,2}$ by

$$u_{S,2}(\boldsymbol{x}) = \frac{1}{M}\sum_{i=1}^{M}\left(u_i(\boldsymbol{x}) - \gamma_1(t_i)u_{S,1}(\boldsymbol{x})\right). \tag{3.75}$$

Finally, the modified snapshots $\{u_i - \gamma_1(t_i)u_{S,1} - u_{S,2}\}_{i=1}^{M}$ are also zero on the entire Dirichlet boundary Γ_{D} but no additional problem has to be solved to generate $u_{S,2}$. ◁

Method 2. The approach for the treatment of time-dependent Dirichlet boundary conditions (3.71) that will be presented next was introduced in [56] in the context of the Navier–Stokes equations. In contrast to Method 1, the snapshots $\{u_1, \ldots, u_M\}$ generated using the boundary conditions (3.71) are not modified to fulfill homogeneous Dirichlet boundary conditions on Γ_{D} before the POD application. The algorithm reads as follows:

1. Compute POD basis functions $\{\varphi_{\mathrm{ro},i}\}_{i=1}^{r}$ from the raw snapshots $\{u_m\}_{m=1}^{M}$. Hence, the resulted POD basis functions are non-zero on Γ_{D}, see the discussion at the beginning of the Section.

2. Compute a new basis $\{\tilde{\varphi}_{\mathrm{ro},j}\}_{j=1}^{r-K}$ as a linear combination of the POD basis functions $\{\varphi_{\mathrm{ro},i}\}_{i=1}^{r}$ such that each $\tilde{\varphi}_{\mathrm{ro},j}$, $j = 1, \ldots, r-K$, satisfies homogeneous boundary conditions on Γ_{D}. The construction of such a basis will be explained below.

3. Set the ROM approximation of $u(t, \boldsymbol{x})$ as

$$u(t,\boldsymbol{x}) \approx u_{\mathrm{ro}}(t,\boldsymbol{x}) = \sum_{i=1}^{r}\alpha_i(t)\varphi_{\mathrm{ro},i}(\boldsymbol{x}) \tag{3.76}$$

   and build a projection-based ROM using $\{\tilde{\varphi}_{\mathrm{ro},j}\}_{j=1}^{r-K}$ as test functions. To enforce the correct boundary conditions, and obtain the same number of equations and unknowns, K more equations have to be added to the system:

$$u_{\mathrm{ro}}(t,\boldsymbol{x}_k) = \gamma_k(t)g_k(\boldsymbol{x}_k), \quad k = 1, \ldots, K,$$

   where $\boldsymbol{x}_k$ is any point on $\Gamma_{\mathrm{D},k}$ for which $g(\boldsymbol{x}_k) \neq 0$.

Altogether, the time-continuous projection-based ROM for the heat equation with boundary conditions (3.71) has the form: Find $u_{\mathrm{ro}}(t,\boldsymbol{x}) = \sum_{i=1}^{r}\alpha_i(t)\varphi_{\mathrm{ro},i}(\boldsymbol{x})$ such that

$$\begin{aligned} (\partial_t u_{\mathrm{ro}}, \tilde{\varphi}_{\mathrm{ro},i}) + (\nabla u_{\mathrm{ro}}, \nabla\tilde{\varphi}_{\mathrm{ro},i}) &= (f, \tilde{\varphi}_{\mathrm{ro},i}), \qquad && i = 1, \ldots, r-K, \\ u_{\mathrm{ro}}(t,\boldsymbol{x}_k) &= \gamma_k(t)g_k(\boldsymbol{x}_k), && k = 1, \ldots, K. \end{aligned} \tag{3.77}$$

Next, the construction of the basis $\{\tilde{\varphi}_{\mathrm{ro},j}\}_{j=1}^{r-K}$ in step 2 of the above algorithm will be presented. For this purpose, one has to choose an arbitrary point $\boldsymbol{x}_k$ with $g_k(\boldsymbol{x}_k) \neq 0$ on each boundary part $\Gamma_{\mathrm{D},k}$, $k = 1, \ldots, K$, and solve the problem

$$\tilde{\varphi}_{\mathrm{ro},j}(\boldsymbol{x}_k) = \sum_{i=1}^{r} \beta_{ij,i} \varphi_{\mathrm{ro},i}(\boldsymbol{x}_k) = 0, \quad k = 1, \ldots, K, \quad j = 1, \ldots, r - K.$$

The unknown coefficients $\beta_{i,j}$ can be computed as the solutions of the systems of equations

$$A\underline{\beta}_j = \underline{0}, \quad j = 1, \ldots, r - K, \tag{3.78}$$

where

$$A = \begin{pmatrix} \varphi_{\mathrm{ro},1}(x_1) & \cdots & \varphi_{\mathrm{ro},r}(x_1) \\ \vdots & \ddots & \vdots \\ \varphi_{\mathrm{ro},1}(x_K) & \cdots & \varphi_{\mathrm{ro},r}(x_K) \end{pmatrix} \in \mathbb{R}^{K \times r}, \quad \underline{\beta}_j = \begin{pmatrix} \alpha_{j,1} \\ \vdots \\ \alpha_{j,r} \end{pmatrix} \in \mathbb{R}^r.$$

To ensure that (3.78) is not over-determined, one has to choose r such that $r \geq K$. The solution of (3.78) is equivalent to determining $\ker(A)$. In [56], it is done by using the QR decomposition of A^T.

It is known that

$$\ker(A) \perp \mathrm{im}(A^T), \tag{3.79}$$

where $\mathrm{im}(A^T) = \{A^T \underline{b}\}_{\underline{b} \in \mathbb{R}^K}$. From the QR decomposition of A^T, i.e., $A^T = \tilde{Q}\tilde{R}$, it follows that

$$\tilde{Q}^T \mathrm{im}(A^T) = \tilde{R}\underline{b}, \quad \forall \underline{b} \in \mathbb{R}^K.$$

As $\tilde{R} \in \mathbb{R}^{r \times K}$ is an upper triangular matrix and $r \geq K$, the last $r-K$ rows of $\tilde{R}$ are zero. This means that the last $r - K$ rows of $\tilde{Q}^T$ are orthogonal to $\mathrm{im}(A^T)$, or, equivalently, that the last $r - K$ columns of $\tilde{Q}$ are orthogonal to $\mathrm{im}(A^T)$. Due to the relation (3.79), the last $r - K$ columns of $\tilde{Q}$ are elements of $\ker(A)$, i.e., they solve (3.78).

Therefore, the basis functions $\tilde{\varphi}_{\mathrm{ro},j}$, $j = 1, \ldots, r - K$, can be constructed by

$$\tilde{\varphi}_{\mathrm{ro},j} = \sum_{i=1}^{r} \tilde{Q}_{K+j,i} \varphi_{\mathrm{ro},i}. \tag{3.80}$$

Method 3. The third method consists in treating the time-dependent Dirichlet boundary conditions of a reduced-order model in the weak sense. This approach was proposed in [47, 48] in the context of reduced-order modeling for the Navier–Stokes equations and it is also known in the framework of finite element methods as the penalty method, see [39].

In [47, 48], the Dirichlet boundary conditions of type (3.71) for $K = 2$, see (3.74), is considered, i.e., on $\Gamma_{\mathrm{D},1}$ a time-dependent and on $\Gamma_{\mathrm{D},2}$ a steady boundary condition is imposed. For the sake of simplicity, the method from [47] will be presented for the heat equation, see Section 3.2.1. The algorithm reads as follows:

1. Modify the snapshots $\{u_m\}_{m=1}^M$ by
$$u_m - u_{g,2}, \quad m = 1, \ldots, M,$$
where $u_{g,2}$ is the function that satisfies the time-independent Dirichlet boundary condition on $\Gamma_{\mathrm{D},2}$ (see Section 3.2.3), e.g., the snapshots' average. Thus, the snapshots $\{u_m - u_{g,2}\}_{m=1}^M$ satisfy the homogeneous boundary condition on $\Gamma_{\mathrm{D},2}$.

2. Compute POD basis functions $\{\varphi_{\mathrm{ro},i}\}_{i=1}^r$ as described in Section 3.1.3 or 3.1.4 from the modified snapshots $\{u_m - u_{g,2}\}_{m=1}^M$. Note that the basis functions $\varphi_{\mathrm{ro},i}$, $i = 1, \ldots, r$, satisfy the homogeneous boundary condition on $\Gamma_{\mathrm{D},2}$ but not on $\Gamma_{\mathrm{D},1}$.

3. Build a ROM as it was described to build (3.56) using $\{\varphi_{\mathrm{ro},i}\}_{i=1}^r$ as the test functions and the reduced-order approximation of $u(t, \boldsymbol{x})$ of the form
$$u(t, \boldsymbol{x}) \approx u_{\mathrm{ro}}(t, \boldsymbol{x}) = u_{g,2}(t, \boldsymbol{x}) + u_r(t, \boldsymbol{x}) = u_{g,2}(t, \boldsymbol{x}) + \sum_{i=1}^r \alpha_i(t)\varphi_{\mathrm{ro},i}(\boldsymbol{x}).$$
In contrast to (3.56), here it holds $\operatorname{span}\{\varphi_{\mathrm{ro},1}, \ldots, \varphi_{\mathrm{ro},r}\} \not\subset H_0^1(\Omega)$. Consequently, the boundary term
$$-\int\limits_{\Gamma_{\mathrm{D},1}} \nabla u_{\mathrm{ro}}\varphi_i \cdot \boldsymbol{n}\, ds, \quad i = 1, \ldots, r,$$
arising from integration by parts of $(\Delta u_{\mathrm{ro}}, \varphi_{\mathrm{ro},i})$, see (2.4), does not cancel. The time-dependent boundary condition on $\Gamma_{\mathrm{D},1}$ is enforced in a weak sense by adding the following term to the ROM
$$-\int\limits_{\Gamma_{\mathrm{D},1}} \frac{u_{\mathrm{ro}} - \gamma_1 g_1}{\epsilon}\varphi_{\mathrm{ro},i}\, ds, \quad i = 1, \ldots, r,$$
where ϵ is a parameter to be chosen. It weights the importance of the fulfillment of the boundary condition versus the fulfillment of the underlying equation.

Altogether, the time-continuous projection-based Galerkin ROM for the heat equation reads as follows: Find $u_r = u_{\mathrm{ro}} - u_{g,2}$ such that
$$\begin{aligned}(\partial_t u_r, \varphi_{\mathrm{ro},i}) + (\nabla u_r, \nabla\varphi_{\mathrm{ro},i}) - \int\limits_{\Gamma_{\mathrm{D},1}} \nabla u_r\varphi_{\mathrm{ro},i} \cdot \boldsymbol{n}\, ds - \frac{1}{\epsilon}\int\limits_{\Gamma_{\mathrm{D},1}} u_r\varphi_{\mathrm{ro},i}\, ds \\ = (f, \varphi_{\mathrm{ro},i}) - (\partial_t u_{g,2}, \varphi_{\mathrm{ro},i}) - (\nabla u_{g,2}, \nabla\varphi_{\mathrm{ro},i}) \\ + \int\limits_{\Gamma_{\mathrm{D},1}} \nabla u_{g,2}\varphi_{\mathrm{ro},i} \cdot \boldsymbol{n}\, ds + \int\limits_{\Gamma_{\mathrm{D},1}} \frac{u_{g,2} - \gamma_1 g_1}{\epsilon}\varphi_{\mathrm{ro},i}\, ds.\end{aligned} \tag{3.81}$$

Remark 3.6. The algorithm above can be modified by, e.g., treating the steady boundary condition on $\Gamma_{\mathrm{D},2}$ the same way as the time-dependent one on $\Gamma_{\mathrm{D},1}$. In this case, step

1 can be skipped and one computes the POD basis $\{\varphi_{\mathrm{ro},i}\}_{i=1}^{r}$ from the raw snapshots $\{u_i\}_{m=1}^{M}$. In step 3, the ROM has to be modified by setting $u_{g,2} = 0$ and additionally inforcing the Dirichlet boundary condition on $\Gamma_{\mathrm{D},2}$, i.e., adding the boundary terms

$$-\int\limits_{\Gamma_{\mathrm{D},2}} \nabla u_{\mathrm{ro}}\varphi_{\mathrm{ro},i} \cdot \boldsymbol{n}\, ds - \int\limits_{\Gamma_{\mathrm{D},2}} \frac{u_{\mathrm{ro}} - g_2}{\epsilon}\varphi_{\mathrm{ro},i}\, ds$$

on the left-hand side of (3.81). The approach can be extended straightforwardly for the boundary condition of type (3.71) with $K \geq 2$. In this case, the reduced-order approximation has the form $u(t,x) \approx u_{\mathrm{ro}}(t,x) = \sum\limits_{i=1}^{r} \alpha_i(t)\varphi_{\mathrm{ro},i}(x)$. Moreover, one has to add

$$-\int\limits_{\Gamma_{\mathrm{D}}} \nabla u_{\mathrm{ro}}\varphi_{\mathrm{ro},i} \cdot \boldsymbol{n}\, ds - \sum_{k=1}^{K} \int\limits_{\Gamma_{\mathrm{D},k}} \frac{u_{\mathrm{ro}} - \gamma_k g_k}{\epsilon}\varphi_{\mathrm{ro},i}\, ds, \quad i = 1, \ldots, r, \tag{3.82}$$

to the ROM obtained in step 3 of the above algorithm. $\triangleleft$

3.2.4. Implementation

In practice, one has to disretize the problem (3.56) in time in order to solve it numerically. For the sake of simplicity, let Δt denote a fixed time step. The superscipt n, $n = 1, 2, ...,$ of a function denotes the evaluation of the function at time $t_n = n\Delta t$. The projection-based Galerkin reduced-order model of the heat equation (3.53) combined with the one-step θ-scheme reads: For $n = 1, 2, \ldots$ find $u_r^n = u_{\mathrm{ro}}^n - u_g^n \in X_r$ such that

$$\begin{aligned}(u_r^n, \varphi_{\mathrm{ro},i}) + \theta\Delta t\,(\nabla u_r^n, \nabla\varphi_{\mathrm{ro},i}) &= \left(u_r^{n-1}, \varphi_{\mathrm{ro},i}\right) - (1-\theta)\Delta t\left(\nabla u_r^{n-1}, \nabla\varphi_{\mathrm{ro},i}\right)\\ &\quad + (1-\theta)\Delta t\left(f^{n-1}, \varphi_{\mathrm{ro},i}\right) + \theta\Delta t\,(f^n, \varphi_{\mathrm{ro},i}) + \left(u_g^{n-1} - u_g^n, \varphi_{\mathrm{ro},i}\right)\\ &\quad - \Delta t\left(\theta\nabla u_g^n + (1-\theta)\nabla u_g^{n-1}, \nabla\varphi_{\mathrm{ro},i}\right), \quad i = 1, \ldots, r,\end{aligned} \tag{3.83}$$

where θ has to be chosen, e.g., see Table 2.1.

Its corresponding matrix-vector form reads: Find $\underline{\alpha}^n = (\alpha_i^n)_{i=1}^r \in \mathbb{R}^r$ such that

$$[M_{\mathrm{ro}} + \theta\Delta t A_{\mathrm{ro}}]\,\underline{\alpha}^n = [M_{\mathrm{ro}} - (1-\theta)\Delta t A_{\mathrm{ro}}]\,\underline{\alpha}^{n-1} + (1-\theta)\Delta t \underline{f}_{\mathrm{ro}}^{n-1} + \theta\Delta t \underline{f}_{\mathrm{ro}}^n + \underline{l}_{\mathrm{ro}}^n, \tag{3.84}$$

where M_{ro}, A_{ro} and $\underline{f}_{\mathrm{ro}}^n$ are defined by (3.58), (3.59), (3.60), respectively, and

$$l_{\mathrm{ro},i}^n = \left(u_g^{n-1} - u_g^n, \varphi_{\mathrm{ro},i}\right) - \Delta t\left(\theta\nabla u_g^n + (1-\theta)\nabla u_g^{n-1}, \nabla\varphi_{\mathrm{ro},i}\right), \tag{3.85}$$

$$\alpha_i^0 = \left(u^0 - u_g^0, \varphi_{\mathrm{ro},i}\right), \quad i = 1, \ldots, r. \tag{3.86}$$

The computational process for the reduced-order modeling based on POD can be integrated into the already existing finite element software used to compute snapshots. Thus, one does not need to provide any new assembling routines for the ROM setting. By applying some algebraic manipulations, one can switch between the finite element and the reduced-oder model presentations as it can be considered as a basis transformation.

One of the core goals of the reduced-order models is computational efficiency. The computational time of a ROM can be divided into offline and online stages. The offline stage includes the computations that have to be performed only once, before the time iteration loop. The offline stage usually comprises the computation of the POD basis functions and all parts of the system that do not depend on time. The online stage consists of computations that have to be repeated at each iteration inside the time loop. Therefore, any computations involving the complexity of the finite element dimension $N \gg r$ should be avoided in the online stage.

Let the Dirichlet boundary condition in (3.53) be of the form (3.71)

$$u(t,\boldsymbol{x}) = \gamma_k(t) g_k(\boldsymbol{x}) \quad \text{on} \quad (0,T] \times \Gamma_{\mathrm{D},k}, \quad k = 1,\ldots,K.$$

Next, the practical computation of the Galerkin ROM (3.84) will be presented applying Method 1 in Section 3.2.3 for the treatment of the time-dependent Dirichlet boundary conditions. The implementation of the other two methods can be carried out similarly and therefore will not be described here.

Let the POD basis functions $\{\phi_{\mathrm{ro},i}\}_{i=1}^{r}$ be constructed following the algorithm of Method 1 in Section 3.2.3, and let $\Phi \in \mathbb{R}^{N\times r}$ denote the matrix consisting of the finite element coefficients of the POD modes columnwise, see (3.49). Hence, the function u_g fulfilling the correct Dirichlet boundary conditions on Γ_{D} reads

$$u_g(t,\boldsymbol{x}) = \sum_{k=1}^{K} \gamma_k(t) u_{S,k}(\boldsymbol{x}), \tag{3.87}$$

where $u_{S,k} \in X_h$, $k = 1,\ldots,K$, are obtained as described in Step 2 of the algorithm. Hence, in what follows, u_g is to be understood as the finite element representation of the extension of the Dirichlet boundary condition into Ω.

With (3.33), the reduced mass and stiffness matrices in (3.84) can be represented by

$$M_{\mathrm{ro}} = \Phi^T M_h \Phi, \quad A_{\mathrm{ro}} = \Phi^T A_h \Phi, \tag{3.88}$$

where M_h and A_h are the finite element mass and stiffness matrices, defined by (2.11) and (2.12), respectively. Note that the reduced mass matrix M_{ro} is the identity matrix if the POD basis is computed with respect to the L^2 inner product. The computation of M_{ro} and A_{ro} can be conducted in the offline stage by assembling M_h and A_h, and subsequently by multiplying the matrices with Φ as shown in (3.88).

Expression $\underline{l}_{\mathrm{ro}}^n$ defined by (3.85) can be computed efficiently in the online stage due to the separated form of the function u_g. For this sake, r-dimensional vectors

$$\Phi^T M_h \underline{u}_{S,k}, \quad \Phi^T A_h \underline{u}_{S,k}, \quad k = 1,\ldots,K, \tag{3.89}$$

are constructed only once before the time loop. At each time iteration, one needs to compute

$$\underline{l}_{\mathrm{ro}}^n = \sum_{k=1}^{K} \Big[(\gamma_k^n - \gamma_k^{n-1}) \Phi^T M_h \underline{u}_{S,k} - \Delta t (\theta_1 \gamma_k^n + \theta_2 \gamma_k^{n-1}) \Phi^T A_h \underline{u}_{S,k} \Big]. \tag{3.90}$$

Similarly to $\underline{l}_{\mathrm{ro}}^n$, the source term $\underline{f}_{\mathrm{ro}}^n$ can be obtained with a low computational effort in the online stage if the source field f can be formulated in the separated form by $f(t, \boldsymbol{x}) = \gamma_f(t) g_f(\boldsymbol{x})$. In such a case, in the offline stage one has to assemble the finite element vector $\underline{g}_{f,h} = ((g_f, \varphi_{h,1}), \ldots, (g_f, \varphi_{h,N}))^T$. At the nth time step, one computes $\underline{f}_{\mathrm{ro}}^n$ by

$$\underline{f}_{\mathrm{ro}}^n = \gamma_f^n \Phi^T \underline{g}_{f,h}. \tag{3.91}$$

Let u_h^n denote the finite element solution at time t_n, $n \geq 0$. Similarly to (3.86), the ROM solution $\underline{\alpha}^n$ at time t_n can be reconstructed from u_h^n by

$$\underline{\alpha}^n = \Phi^T M_h (\underline{u}_h^n - \underline{u}_g^n). \tag{3.92}$$

Vice versa, the reduced-order approximation of the finite element solution is given by

$$\underline{u}_h^n \approx \underline{u}_g^n + \Phi \underline{\alpha}^n. \tag{3.93}$$

The reduced-order models of more complicated equations like the convection-diffusion-reaction equation (2.1) and the Navier–Stokes equations (2.40)-(2.41) include additional terms compared to (3.84). Their implementation will be discussed in Section 4.2 and Section 5.3, respectively.

4. SUPG Reduced-Order Models for Convection-Dominated Problems

This chapter presents a projection-based Streamline-Upwind Petrov–Galerkin reduced-order model (SUPG-ROM) for convection-dominated problems which are described by the convection-diffusion-reaction equation (2.1). The ROM is based on the Proper Orthogonal Decomposition, see Section 3.1, and investigated theoretically and numerically. A large part of the results presented in this chapter can be found in [44]. To the best of the author's knowledge, the SUPG-ROM was first studied in [88] and later on in [19], in both papers for the Navier–Stokes equations. Numerical analysis is utilized to propose the scaling of the stabilization parameter for the SUPG-ROM. In this respect two approaches are applied: One based on the underlying finite element discretization and the other one based on the POD truncation. For the numerical investigation of the determined stabilization parameters, the SUPG finite element method was used on realistic meshes for computing the snapshots, leading to some noise in the POD data. The resulting SUPG-ROMs and the standard Galerkin ROM (G-ROM) are studied numerically. Another objective of the numerical investigations consists in exploring the impact of the accuracy of the snapshots and of the utilized initial condition on the ROM results.

4.1. Introduction

Reduced-order models based on POD are already used for many complex systems. There are situations where G-ROMs are efficient and relatively accurate (see [63, 97, 107]). However, in other situations, a G-ROM might produce inaccurate results [10]. One of the main reasons for these inaccurate results is that the underlying G-ROM can be numerically unstable, e.g., the G-ROM solution can blow up in a nonphysical way, see [5, 16, 85] for the compressible Navier-Stokes equations and [13, 127] for the incompressible Navier-Stokes equations. Various stabilized ROMs have been proposed, see [5, 9, 12, 13, 19, 71, 88, 127, 144]; see also [34, 35, 109] for similar work in reduced basis methods. This chapter focuses on ROMs for convection-dominated convection-diffusion-reaction equations and these ROMs' potential numerical instability due to the unresolved layers. For the stabilization of the ROMs, the Streamline-Upwind Petrov–Galerkin is employed.

The finite element Streamline-Upwind Petrov–Galerkin method is one of the most popular stabilized finite element methods, see Section 2.1.3. The method contains a stabilization parameter to be chosen, whose asymptotic value for steady-state problems is well known from finite element error analysis (e.g., [118]); it depends on the local mesh width. However, the situation is not completely clear for time-dependent problems. For general problems, optimal estimates can only (to the best of the author's knowledge) be derived for parameters dependent on the length of the time step. For a simplified situation, estimates can also be proven for parameters dependent on the mesh width; see [81] and Section 2.1.3 for details. From the practical point of view, the latter choice seems to be more appropriate since the difficulty of not being able to resolve the layers vanishes on sufficiently fine meshes but not for sufficiently small time steps.

As in the finite element SUPG method, the question of appropriate stabilization parameters for SUPG-ROMs arises. SUPG-ROM parameters depending only on the spatial resolution are preferable for the same reasons as in the finite element method (see the discussion in Section 2.1.3). A ROM based on finite element data has two parts to its spatial resolution: The spatial resolution from the finite element space and the spatial resolution from the space of POD modes used in the ROM, which is a subspace of the finite element space. One can ask on which spatial resolution the stabilization parameter for the SUPG-ROM should depend. This is the main question studied here.

The question of appropriate stabilization parameters for the SUPG-ROM is addressed by means of a numerical analysis of this problem. To the best of the author's knowledge, the use of numerical analysis to propose the SUPG-ROM stabilization parameter is first introduced in [44]. In the literature so far, simply the stabilization parameter from the finite element method was used, like in [88], or an optimization problem for the determination of the parameter was solved, as in [19]. Motivations for these approaches with numerical analysis were not provided. Without any doubt it is desirable to have some support for the choice of stabilization parameters coming from numerical analysis, since parameters determined with considerations from numerical analysis should be valid for a wide range of settings (e.g., with respect to the diffusion coefficients and the convection vector). As a result of the analytical considerations, two stabilization parameters will be proposed. One of them is based on the finite element resolution and the other one is based on the POD spatial resolution. The resulting ROMs will be denoted as FE-SUPG-ROM and POD-SUPG-ROM, respectively.

Solutions of the convection-dominated convection-diffusion-reaction problems (2.1) describing such physical quantities as, e.g., concentration or temperature, can take values only in a certain interval. Especially for the simulation of strongly coupled systems, e.g., like in [80], it is of great importance to employ numerical methods producing solutions without or at least with small under- and overshoots. The reason is that non-physical solutions serving as an input to other equations could cause instabilities or completely incorrect model results. In the context of finite element methods, there exist schemes that aim to prevent the under- and overshoots of the solution, e.g., the FEM-FCT scheme presented in Section 2.1.5. One of the objectives of the numerical studies will be to provide an answer to the question if the employed ROMs based on the physically meaningful snapshots are able to yield solutions without or with very small under- and overshoots.

In some situations, the reduced-order approximation of the initial condition computed in the standard way by (3.62) can contain some spurious oscillations, e.g., see the upper row in Figure 4.28. In order to reduce those oscillations, a filtering post-processing procedure of the initial condition was proposed in Section 3.2.2. Within the scope of the numerical investigations in this chapter, the impact of the utilization of the filtered initial condition on the ROM results will be studied.

The chapter is organized as follows. Section 4.2 gives formulations of the G-ROM and the SUPG-ROM. The core of Section 4.3 is the proposal of two stabilization parameters for the SUPG-ROM, using a numerical analysis of the SUPG-ROM. The SUPG-ROMs and the G-ROM are studied numerically in Section 4.4 on three convection-dominated convection-diffusion-reaction problems.

4.2. ROM Setting

In this section, the SUPG reduced-order model for the convection-dominated problem (2.1) will be derived.

Let $X = H_0^1(\Omega)$, and $X_h \subset X$ denote a conforming finite element space spanned by piecewise polynomials of order m used to compute the snapshots. Let X_R be the R-dimensional space of the snapshots, and X_r the r-dimensional POD space spanned by the POD basis functions $\{\varphi_{\mathrm{ro},1}, \ldots, \varphi_{\mathrm{ro},r}\}$ with $r \leq R$. The POD modes are assumed to be computed with respect to the L^2 inner product, which is the most common choice found in the literature. To compute the POD basis functions, the centered-trajectory method is utilized, i.e., the POD modes are computed from the fluctuation of the snapshots $u_h^m - \bar{u}_h$, $m = 1, \ldots, M$, where $\bar{u}_h$ is the average of the snapshots, see Section 3.2.3.

The projection-based G-ROM for (2.1) can be built as described in Section 3.2 by projecting the continuous problem into the finite-dimensional POD space X_r.

Let the ROM approximation of the solution u be expressed by

$$u(t, \boldsymbol{x}) \approx u_{\mathrm{ro}}(t, \boldsymbol{x}) = \bar{u}_h(\boldsymbol{x}) + u_r(t, \boldsymbol{x}),$$

where $u_r(t, \boldsymbol{x}) = \sum_{i=1}^{r} \alpha_i(t) \varphi_{\mathrm{ro},i}(\boldsymbol{x})$ with the unknown coefficients $\{\alpha_i\}_{i=1}^{r}$. Then, the Galerkin reduced-order model for (2.1) reads: Find $u_r = u_{\mathrm{ro}} - \bar{u}_h \colon (0, T] \to X_r$ such that $\forall v_r \in X_r$

$$\begin{aligned} &(\partial_t u_r, v_r) + (\varepsilon \nabla u_r, \nabla v_r) + (\boldsymbol{b} \cdot \nabla u_r, v_r) + (c u_r, v_r) \\ &= (f, v_r) - (\varepsilon \nabla \bar{u}_h, \nabla v_r) - (\boldsymbol{b} \cdot \nabla \bar{u}_h, v_r) - (c \bar{u}_h, v_r)\,. \end{aligned} \tag{4.1}$$

In many engineering applications, one has to deal with the convection-dominated regime, see Section 2.1.2. An important question is how to compute a solution in the POD space X_r which is as accurate as possible compared with the solution of the continuous problem. To achieve this goal, it might be necessary in the convection-dominated regime to add some stabilization to the finite-dimensional problem (4.1) similarly to the finite element methods presented in Section 2.1. Here, the SUPG stabilization, see Section 2.1.3, of (4.1) will be considered.

Let the superscript n of a function denote the evaluation of the function at the time instance t_n and let Δt denote the fixed time step. The backward Euler/SUPG reduced-order model reads as follows: For $n = 1, 2, \ldots$ find $u_r^n = u_{\mathrm{ro}}^n - \bar{u}_h \in X_r$ such that

$\forall v_r \in X_r$

$$\begin{aligned}&\left(u_r^n - u_r^{n-1}, v_r\right) + \Delta t a_{\mathrm{SUPG},r}\left(u_r^n, v_r\right) = \Delta t\left(f^n, v_r\right)\\&+ \Delta t \sum_{K\in\mathcal{T}_h} \delta_{r,K}\left(f^n, \boldsymbol{b}^n\cdot\nabla v_r\right)_K - \sum_{K\in\mathcal{T}_h}\delta_{r,K}\left(u_r^n - u_r^{n-1}, \boldsymbol{b}^n\cdot\nabla v_r\right)_K\\&- \Delta t\, a_{\mathrm{SUPG},r}\left(\bar{u}_h, v_r\right),\end{aligned}\tag{4.2}$$

where

$$\begin{aligned}a_{\mathrm{SUPG},r}(u_r, v_r) = {}&(\varepsilon\nabla u_r, \nabla v_r) + (\boldsymbol{b}^n\cdot\nabla u_r, v_r) + (c^n u_r, v_r)\\&+ \sum_{K\in\mathcal{T}_h}\delta_{r,K}\left(-\varepsilon\Delta u_r + \boldsymbol{b}^n\cdot\nabla u_r + c^n u_r, \boldsymbol{b}^n\cdot\nabla v_r\right)_K\end{aligned}\tag{4.3}$$

for all $u_r, v_r \in X_r$.

Remark 4.1. The bilinear form $a_{\mathrm{SUPG},r}(\cdot,\cdot)$ consists of the same terms as the bilinear form (2.17) but with the SUPG-ROM parameter $\delta_{r,K}$ instead of $\delta_{h,K}$ on the underlying triangulation $\mathcal{T}_h$. ◁

Remark 4.2. Note that by setting $\delta_{r,K} = 0$ in (4.2), the Galerkin ROM (4.1) discretized with the backward Euler scheme in time is recovered. ◁

Similarly to (3.84), one can formulate the problem (4.2) in the matrix-vector form as follows: Find $\underline{\alpha}^n = (\alpha_i^n)_{i=1}^r \in \mathbb{R}^r$ such that

$$\left[M_{\mathrm{ro}}^* + \theta\Delta t S_{\mathrm{ro}}\right]\underline{\alpha}^n = M_{\mathrm{ro}}^*\underline{\alpha}^{n-1} + \theta\Delta t \underline{f}_{\mathrm{ro}}^{*,n} + \underline{l}_{\mathrm{ro}}^{*,n},\tag{4.4}$$

where

$$(M_{\mathrm{ro}}^*)_{ij} = (\varphi_{\mathrm{ro},j}, \varphi_{\mathrm{ro},i}) + \sum_{K\in\mathcal{T}_h}\int_K \delta_{r,K}\varphi_{\mathrm{ro},j}\boldsymbol{b}^n\cdot\nabla\varphi_{\mathrm{ro},i}\, d\boldsymbol{x},\quad i,j = 1,\ldots,r,\tag{4.5}$$

$$\begin{aligned}(S_{\mathrm{ro}})_{ij} = {}&(\varepsilon\nabla\varphi_{\mathrm{ro},j}, \nabla\varphi_{\mathrm{ro},i}) + (\boldsymbol{b}^n\cdot\nabla\varphi_{\mathrm{ro},j}, \nabla\varphi_{\mathrm{ro},i}) + (c^n\varphi_{\mathrm{ro},j}, \varphi_{\mathrm{ro},i})\\&+ \sum_{K\in\mathcal{T}_h}\int_K \delta_{r,K}\left(-\varepsilon\Delta\varphi_{\mathrm{ro},j} + \boldsymbol{b}^n\cdot\nabla\varphi_{\mathrm{ro},j} + c^n\varphi_{\mathrm{ro},j}\right)\boldsymbol{b}^n\cdot\nabla\varphi_{\mathrm{ro},i}\, d\boldsymbol{x},\\&i,j = 1,\ldots,r,\end{aligned}\tag{4.6}$$

$$f_{\mathrm{ro},i}^{*,n} = (f^n, \varphi_{\mathrm{ro},i}) + \sum_{K\in\mathcal{T}_h}\int_K \delta_{r,K} f^n\boldsymbol{b}^n\cdot\nabla\varphi_{\mathrm{ro},i}\, d\boldsymbol{x},\quad i = 1,\ldots,r,\tag{4.7}$$

$$\begin{aligned}l_{\mathrm{ro},i}^{*,n} = {}&(\varepsilon\nabla\bar{u}_h, \nabla\varphi_{\mathrm{ro},i}) + (\boldsymbol{b}^n\cdot\nabla\bar{u}_h, \nabla\varphi_{\mathrm{ro},i}) + (c^n\bar{u}_h, \varphi_{\mathrm{ro},i})\\&- \sum_{K\in\mathcal{T}_h}\int_K \delta_{r,K}\Delta\bar{u}_h\boldsymbol{b}^n\cdot\nabla\varphi_{\mathrm{ro},i}\, d\boldsymbol{x} + \sum_{K\in\mathcal{T}_h}\int_K \delta_{r,K}\boldsymbol{b}^n\cdot\nabla\bar{u}_h\boldsymbol{b}^n\cdot\nabla\varphi_{\mathrm{ro},i}\, d\boldsymbol{x}\\&+ \sum_{K\in\mathcal{T}_h}\int_K \delta_{r,K}c^n\bar{u}_h\boldsymbol{b}^n\cdot\nabla\varphi_{\mathrm{ro},i}\, d\boldsymbol{x},\quad i,j = 1,\ldots,r,\end{aligned}\tag{4.8}$$

$$\alpha_i^0 = \left(u^0 - \bar{u}_h, \varphi_{\mathrm{ro},i}\right),\quad i = 1,\ldots,r.\tag{4.9}$$

Remark 4.3. The fourth term of S_{ro} with the piecewise defined Laplacian is well-defined as the POD basis functions $\varphi_{\mathrm{ro},i}$, $i = 1, \ldots, r$, are, by construction, linear combinations of finite element basis functions, see Section 3.1.2. The term vanishes when less than second order finite elements are used. $\lhd$

The computational efficiency of the reduced-order models is of crucial importance. Therefore, within the time loop it is desired to avoid computations of the complexity of the finite element dimension. In Section 3.2.4, the implementation of the terms in (4.4) which do not include the reaction and the convection fields was discussed. All terms with $\boldsymbol{b}$ and c can be efficiently computed if the convection and the reaction fields can be written in the separated form as

$$\boldsymbol{b}(t, \boldsymbol{x}) = \gamma_{\boldsymbol{b}}(t)\boldsymbol{g}_{\boldsymbol{b}}(\boldsymbol{x}), \quad c(t, \boldsymbol{x}) = \gamma_c(t) g_c(\boldsymbol{x}).$$

For the sake of brevity, only the implementation of the third term of S_{ro} will be described. The computation of other terms can be carried out following the same idea. Let C_{ro} denote the reduced-order matrix with the entries

$$(C_{\mathrm{ro}})_{ij} = \left(c^n \varphi_{\mathrm{ro},j}, \varphi_{\mathrm{ro},i}\right), \quad i, j = 1, \ldots, r.$$

Its computation comprises two steps. In the first step, which is the offline stage, one has to assemble the finite element matrix C_h with the time-independent part of the reaction field once, i.e.,

$$(C_h)_{ij} = \left(g_c \varphi_{h,j}, \varphi_{h,i}\right), \quad i, j = 1, \ldots, N.$$

In the second step, which is the online stage, C_{ro} is obtained at every time iteration by

$$C_{\mathrm{ro}} = \gamma_c^n \Phi^T C_h \Phi, \quad n = 1, 2, \ldots.$$

4.3. Stabilization Parameter Based on Numerical Analysis

In this section, numerical analysis is used to propose scalings of the stabilization parameter of the SUPG-ROM (4.2). In what follows, C denotes a generic constant which does not depend on the mesh width h and on the size of the diffusion ε. The investigation is restricted to the case of uniform meshes. Therefore, $\delta_{r,K} = \delta_r$ for all $K \in \mathcal{T}_h$. Thus, the SUPG-ROM (4.2) can be reformulated as follows: For $n = 1, 2, \ldots$ find $u_r^n = u_{\mathrm{ro}}^n - \bar{u}_h \in X_r$ such that $\forall v_r \in X_r$

$$\begin{aligned} &\left(u_r^n - u_r^{n-1}, v_r\right) + \Delta t a_{\mathrm{SUPG},r}\left(u_r^n, v_r\right) = \Delta t \left(f^n, v_r\right) \\ &+ \Delta t \delta_r \left(f^n, \boldsymbol{b}^n \cdot \nabla v_r\right) - \delta_r \left(u_r^n - u_r^{n-1}, \boldsymbol{b}^n \cdot \nabla v_r\right) - \Delta t\, a_{\mathrm{SUPG},r}\left(\bar{u}_h, v_r\right). \end{aligned} \tag{4.10}$$

The analytical study for the proposal of the stabilization parameter δ_r in (4.10) is based on considerations of the error between the solution u of the continuous problem (2.2) and the solution u_{ro} of (4.10). Here, it is assumed that the SUPG finite element method (2.18) is applied to generate snapshots needed for the computation of the POD basis. As a first step, the error is split in the form

$$u - u_{\mathrm{ro}} = (u - u_h) + (u_h - P_r(u_h)) + (P_r(u_h) - u_{\mathrm{ro}}), \tag{4.11}$$

where u_h is the finite element solution and

$$P_r(u_h) := \bar{u}_h + \sum_{j=1}^{r} (u_h - \bar{u}_h, \varphi_{\mathrm{ro},j})\, \varphi_{\mathrm{ro},j} \tag{4.12}$$

is the L^2 projection of a finite element function into the space X_r.

The first term on the right-hand side of (4.11) is the finite element error discussed in Section 2.1.4. In the course of the numerical analysis, different types of the inverse estimates are discussed in Section 4.3.1. To estimate the second term in (4.11), estimates of the projection error for the projection from X_h to X_r are derived in Section 4.3.2. Next, conditions for the coercivity of the SUPG bilinear form in X_r are given in Section 4.3.3. Finally, in Section 4.3.4 two different versions of the SUPG-ROM parameters are proposed.

4.3.1. Inverse Estimates

Assuming that the family of triangulations is quasi-uniform, the local inverse inequality (2.6) for finite element functions with the inverse constant μ_{inv} holds. For families of uniform triangulations, as considered in this analytical study, one can derive from (2.6) global inverse estimates of the form

$$\left(\sum_{K \in \mathcal{T}_h} \|v_h\|_{m,K}^2 \right)^{1/2} \leq C h^{l-m} \left(\sum_{K \in \mathcal{T}_h} \|v_h\|_{l,K}^2 \right)^{1/2}, \quad \forall v_h \in X_h, \tag{4.13}$$

where μ_{inv} is included into the generic constant C.

Inverse estimates are also known in the context of POD. In [89], the following inverse estimate was proven:

$$\|\nabla v_r\|_0 \leq \sqrt{|||A_{\mathrm{ro}}|||_2\, |||M_{\mathrm{ro}}^{-1}|||_2}\, \|v_r\|_0, \quad \forall v_r \in X_r,\ r \in \{1, \ldots, R\}, \tag{4.14}$$

where $|||\cdot|||_2$ denotes the spectral norm of a matrix, M_{ro} and A_{ro} are the reduced mass and stiffness matrices defined by (3.58) and (3.59), respectively. When $L^2(\Omega)$ is the norm to generate the POD modes (as in this study), M_{ro} in (4.14) is the identity matrix. Estimate (4.14) was derived in [89] for the situation that the POD basis is computed from snapshots in an infinite-dimensional Hilbert space.

In the setting of [89], the POD basis functions are known to belong to the infinite-dimensional Hilbert space. In practice, however, the POD basis is usually computed from snapshots of some numerical approximation of the solution of the continuous problem. Here, the snapshots are computed with a finite element method and belong to X_h. Consequently, the POD basis functions $\{\varphi_{\mathrm{ro},1}, \ldots, \varphi_{\mathrm{ro},r}\}$, $r \leq R$, belong not only to X but also to X_h and X_R, see Section 3.1.2. Hence, two inverse estimates hold for functions in X_r: a POD estimate of form (4.14) and a finite element estimate of form (4.13).

The POD inverse estimate (4.14) can be extended to the piecewise defined Laplacian. Let the reduced Hessian matrix, denoted by H_{ro}, be given by

$$(H_{\text{ro}})_{ij} = \sum_{K \in \mathcal{T}_h} \int_K \Delta\varphi_{\text{ro},j} \Delta\varphi_{\text{ro},i} \, d\boldsymbol{x}, \quad i, j = 1, \ldots, r. \tag{4.15}$$

Lemma 4.1 (POD Inverse Estimate)**.** *For all $v_r \in X_r$, $1 \le r \le R$, the following estimate holds:*

$$\|\Delta v_r\|_0 \le \sqrt{|||H_r|||_2 \left|\left|\left|A_{\text{ro}}^{-1}\right|\right|\right|_2} \|\nabla v_r\|_0, \tag{4.16}$$

where $\|\Delta v_r\|_0$ is defined by a sum over the mesh cells as used in (4.15)*.*

Proof. The proof follows [89]. Let $v_r = \sum_{j=1}^{r} x_j \varphi_{\text{ro},j}$ and $\boldsymbol{x} = (x_1, \ldots, x_r)^T$. Then, one obtains from the definition of H_{ro}, a standard estimate of matrix-vector products, the fact that A_{ro} is symmetric and positive definite, and the definition of A_{ro}

$$\begin{aligned}
\|\Delta v_r\|_0^2 &= \boldsymbol{x}^T H_{\text{ro}} \boldsymbol{x} \le |||H_{\text{ro}}|||_2 \|\boldsymbol{x}\|_{\text{eucl}}^2 = |||H_{\text{ro}}|||_2 \boldsymbol{x}^T \boldsymbol{x} \\
&= |||H_{\text{ro}}|||_2 \boldsymbol{x}^T A_{\text{ro}}^{T/2} A_{\text{ro}}^{-1} A_{\text{ro}}^{1/2} \boldsymbol{x} \le |||H_{\text{ro}}|||_2 \left|\left|\left|A_{\text{ro}}^{-1}\right|\right|\right|_2 \left\|A_{\text{ro}}^{1/2} \boldsymbol{x}\right\|_{\text{eucl}}^2 \\
&= |||H_{\text{ro}}|||_2 \left|\left|\left|A_{\text{ro}}^{-1}\right|\right|\right|_2 \boldsymbol{x}^T A_{\text{ro}} \boldsymbol{x} = |||H_{\text{ro}}|||_2 \left|\left|\left|A_{\text{ro}}^{-1}\right|\right|\right|_2 \|\nabla v_r\|_0^2,
\end{aligned}$$

where $\|\cdot\|_{\text{eucl}}$ denotes the Euclidian vector norm. □

If in the generation of the POD basis, the $H^1(\Omega)$ seminorm was used (see, e.g., [68,89]) instead of the $L^2(\Omega)$ norm utilized in the underlying study, then A_{ro} in (4.16) is the identity matrix.

Remark 4.4. The asymptotic behavior of the first factor on the right-hand side of (4.14) and (4.16) will be discussed in a simplified situation. As in [68], it is assumed that the POD vectors are the Fourier basis in a single dimension with homogeneous Dirichlet boundary conditions, i.e., $\varphi_{\text{ro},j}(x) = \sin(\pi j x)$ on $[0, T]$. The single dimension case is relevant because the considered convection-dominated problems exhibit motion along a preferred direction. Therefore, the matrices A_{ro}, H_{ro}, M_{ro}^{-1}, and A_{ro}^{-1} are diagonal and have the entries

$$\begin{aligned}
(A_{\text{ro}})_{jj} &= \int_0^1 (\pi j)^2 \cos(\pi j x)^2 = \mathcal{O}(j^2), \quad j = 1, \ldots, r, \\
(H_{\text{ro}})_{jj} &= \int_0^1 (\pi^2 j^2)^2 \sin(\pi j x)^2 dx = \mathcal{O}(j^4), \quad j = 1, \ldots, r, \\
(M_{\text{ro}}^{-1})_{jj} &= \left(\int_0^1 \sin(\pi j x)^2 dx\right)^{-1} = \mathcal{O}(1), \quad j = 1, \ldots, r, \\
(A_{\text{ro}}^{-1})_{jj} &= \left(\int_0^1 (\pi j)^2 \cos(\pi j x)^2\right)^{-1} = \mathcal{O}(j^{-2}), \quad j = 1, \ldots, r.
\end{aligned}$$

Hence, $|||A_{\text{ro}}|||_2 = \mathcal{O}(r^2)$, $\left|\left|\left|M_{\text{ro}}^{-1}\right|\right|\right|_2 = \mathcal{O}(1)$, $|||H_{\text{ro}}|||_2 = \mathcal{O}(r^4)$ and $\left|\left|\left|A_{\text{ro}}^{-1}\right|\right|\right|_2 = \mathcal{O}(1)$. Altogether, the first factors on the right-hand sides of (4.14) and (4.16) scale like $\mathcal{O}(r)$ and $\mathcal{O}(r^2)$, respectively. ◁

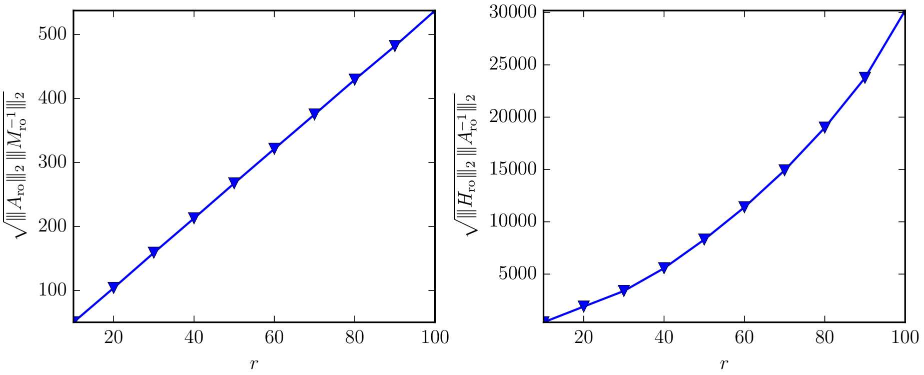

Figure 4.1.: Remark 4.5: Dependence of the constants from the inverse estimates (4.14) (left) and (4.16) (right) on the dimension of the POD basis.

The following numerical example will demonstrate that this scaling can be observed also in more general situations.

Remark 4.5. Consider a two-dimensional test example for the convection-diffusion-reaction equation (2.1) describing a traveling wave with $\varepsilon = 10^{-8}$. A detailed specification of the problem setting is given in Example 4.2 below. In Figure 4.1, the dependence of the constants from the inverse estimates (4.14) and (4.16) on the dimension of the POD basis r is shown. The asymptotic behavior discussed in Remark 4.4 can be observed also in this two-dimensional case. $\triangleleft$

4.3.2. Projection Error

This section presents an estimate for the error between the snapshots and their projection into the POD space X_r. This error is the second term on the right-hand side of the error decomposition (4.11).

Lemma 4.2. *Let $\bar{u}_h = 0$ and let $\langle \cdot, \cdot \rangle_s$, $s \in \{0, 1, 2\}$, be a semi-inner product, see [32], with induced seminorm $| \cdot |_s$ of $H^s(\Omega)$. Then the POD projection error in the s-seminorm satisfies*

$$\sum_{m=1}^{M} \left| u_h^m - \sum_{j=1}^{r} \left(u_h^m, \varphi_{\mathrm{ro},j}\right) \varphi_{\mathrm{ro},j} \right|_s^2 = \sum_{j=r+1}^{R} \lambda_j |\varphi_{\mathrm{ro},j}|_s^2.$$

For $s = 0$, $\langle \cdot, \cdot \rangle_s$ and $| \cdot |_s$ are the inner product $(\cdot, \cdot)$ and the norm $\|\cdot\|_0$, respectively. For $s = 2$, the definitions of $\langle \cdot, \cdot \rangle_s$ and $| \cdot |_s$ have to be understood as a sum over the mesh cells.

Proof. Taking the s-seminorm of the POD truncation error $\sum_{j=r+1}^{R} \left(u_h^m, \varphi_{\mathrm{ro},j}\right) \varphi_{\mathrm{ro},j}$, $m = 1, \ldots, M$, using the definition of the eigenvalues and eigenfunctions of the POD, e.g., see [26, Eq. (6)] or (3.38) for the matrix-vector representation, and applying the orthogonality of the POD basis functions yields

$$\begin{aligned}
\sum_{m=1}^{M}\left|u_h^m-\sum_{j=1}^{r}\left(u_h^m,\varphi_{\mathrm{ro},j}\right)\varphi_{\mathrm{ro},j}\right|_s^2 &= \sum_{m=1}^{M}\left|\sum_{j=r+1}^{R}\left(u_h^m,\varphi_{\mathrm{ro},j}\right)\varphi_{\mathrm{ro},j}\right|_s^2\\
&= \sum_{m=1}^{M}\left\langle\sum_{j=r+1}^{R}\left(u_h^m,\varphi_{\mathrm{ro},j}\right)\varphi_{\mathrm{ro},j},\sum_{k=r+1}^{R}\left(u_h^m,\varphi_{\mathrm{ro},k}\right)\varphi_{\mathrm{ro},k}\right\rangle_s\\
&= \sum_{m=1}^{M}\sum_{j=r+1}^{R}\sum_{k=r+1}^{R}\left(u_h^m,\varphi_{\mathrm{ro},j}\right)\left(u_h^m,\varphi_{\mathrm{ro},k}\right)\left\langle\varphi_{\mathrm{ro},j},\varphi_{\mathrm{ro},k}\right\rangle_s\\
&= \sum_{j=r+1}^{R}\sum_{k=r+1}^{R}\left(\sum_{m=1}^{M}\left(u_h^m,\varphi_{\mathrm{ro},j}\right)u_h^m,\varphi_{\mathrm{ro},k}\right)\left\langle\varphi_{\mathrm{ro},j},\varphi_{\mathrm{ro},k}\right\rangle_s\\
&= \sum_{j=r+1}^{R}\sum_{k=r+1}^{R}\left(\lambda_j\varphi_{\mathrm{ro},j},\varphi_{\mathrm{ro},k}\right)\left\langle\varphi_{\mathrm{ro},j}\varphi_{\mathrm{ro},k}\right\rangle_s\\
&= \sum_{j=r+1}^{R}\lambda_j|\varphi_{\mathrm{ro},j}|_s^2.
\end{aligned}$$

□

The result of Lemma 4.2 is similar to results obtained in [69, 126].

Corollary 4.1. *It holds that*

$$\sum_{m=1}^{M}|u_h^m-P_r(u_h^m)|_s^2=\sum_{j=r+1}^{R}\lambda_j|\varphi_{\mathrm{ro},j}|_s^2. \tag{4.17}$$

Proof. By the definition (4.12) of the projection, it follows that

$$\sum_{m=1}^{M}|u_h^m-P_r(u_h^m)|_s^2=\sum_{m=1}^{M}\left|(u_h^m-\bar{u}_h)-\sum_{j=1}^{r}\left(u_h^m-\bar{u}_h,\varphi_{\mathrm{ro},j}\right)\varphi_{\mathrm{ro},j}\right|_s^2.$$

Since the arithmetic average $\overline{u_h^m-\bar{u}_h}$ is zero, the application of Lemma 4.2 proves the statement. □

Corollary 4.2. *It holds that*

$$\sum_{m=1}^{M}|u_h^m-P_r(u_h^m))|_s^2\le Ch^{-2s}\sum_{j=r+1}^{R}\lambda_j. \tag{4.18}$$

Proof. Since $X_r \subset X_h$, one can apply the inverse estimate (4.13) to the right-hand side of (4.17)

$$|\varphi_{\mathrm{ro},j}|_s^2 \leq Ch^{-2s} \|\varphi_{\mathrm{ro},j}\|_0^2.$$

The statement of the corollary follows by utilizing the fact that the POD basis functions are normalized. □

Altogether, there are two ways to bound the projection error: with data from the POD only, see (4.17), or with data from the POD and the finite element method, see (4.18). Note that the sums on the right-hand side of (4.17) and (4.18) can be practically computed.

4.3.3. Coercivity of the SUPG-ROM Bilinear Form in X_r

The coercivity of the SUPG bilinear form (2.17) in X_r is essential for the well-posedness of the SUPG-ROM problem in X_r. This property gives first restrictions on the stabilization parameters.

Lemma 4.3. *Let either*

$$0 \leq \delta_r \leq \frac{1}{2} \min \left\{ \frac{\mu_0}{\|c\|_{L^\infty(\Omega)}^2}, \frac{h^2}{\varepsilon \mu_{\mathrm{inv}}^2} \right\}, \tag{4.19}$$

or

$$0 \leq \delta_r \leq \frac{1}{2} \min \left\{ \frac{\mu_0}{\|c\|_{L^\infty(\Omega)}^2}, \frac{1}{\varepsilon \left\| |H_{\mathrm{ro}}| \right\|_2 \left\| |A_{\mathrm{ro}}^{-1}| \right\|_2} \right\}. \tag{4.20}$$

Then

$$a_{\mathrm{SUPG},r}(v_r, v_r) \geq \frac{1}{2} |||v_r|||_{\mathrm{SUPG},r}^2, \quad \forall v_r \in X_r, \tag{4.21}$$

with

$$|||v_r|||_{\mathrm{SUPG},r} = \left(\varepsilon |v_r|_1^2 + \mu_0 \|v_r\|_0^2 + \delta_r \|\boldsymbol{b} \cdot \nabla v_r\|_0^2 \right)^{1/2}, \tag{4.22}$$

and μ_0 *from* (2.19).

Proof. The proof follows the standard lines as it can be found, e.g., in [118], see also the proof of Lemma 2.2. Using integration by parts, the assumption (2.19), and the definition (4.22) one obtains

$$\begin{aligned} a_{\mathrm{SUPG},r}(v_r, v_r) &\geq \varepsilon |v_r|_1^2 + \mu_0 \|v_r\|_0^2 + \delta_r \|\boldsymbol{b} \cdot \nabla v_r\|_0^2 - \delta_r \sum_{K \in \mathcal{T}_h} (-\varepsilon \Delta v_r + c v_r, \boldsymbol{b} \cdot \nabla v_r)_K \\ &= |||v_r|||_{\mathrm{SUPG},r}^2 - \delta_r \sum_{K \in \mathcal{T}_h} (-\varepsilon \Delta v_r + c v_r, \boldsymbol{b} \cdot \nabla v_r)_K . \end{aligned} \tag{4.23}$$

By the Cauchy–Schwarz inequality and Young's inequality, the last term on the right-hand side of (4.23) can be estimated by

$$
\begin{aligned}
&\left|\delta_r \sum_{K\in\mathcal{T}_h} \left(-\varepsilon\Delta v_r + c v_r, \boldsymbol{b}\cdot\nabla v_r\right)_K\right| \\
&\le \delta_r \left|\sum_{K\in\mathcal{T}_h} \left(-\varepsilon\Delta v_r, \boldsymbol{b}\cdot\nabla v_r\right)_K\right| + \delta_r \left|\left(c v_r, \boldsymbol{b}\cdot\nabla v_r\right)\right| \\
&\le \delta_r \left(\sum_{K\in\mathcal{T}_h} \varepsilon\|\Delta v_r\|_{0,K}\right) \|\boldsymbol{b}\cdot\nabla v_r\|_0 + \delta_r \|c v_r\|_0 \|\boldsymbol{b}\cdot\nabla v_r\|_0 \\
&\le \delta_r \sum_{K\in\mathcal{T}_h} \varepsilon^2\|\Delta v_r\|_{0,K}^2 + \delta_r\|c\|_{L^\infty(\Omega)}^2\|v_r\|_0^2 + \frac{\delta_r}{2}\|\boldsymbol{b}\cdot\nabla v_r\|_0^2.
\end{aligned}
$$

Inserting this estimate into (4.23) results in

$$
a_{\mathrm{SUPG},r}(v_r, v_r) \ge |||v_r|||_{\mathrm{SUPG},r}^2 - \delta_r \sum_{K\in\mathcal{T}_h} \varepsilon^2\|\Delta v_r\|_{0,K}^2 - \delta_r\|c\|_{L^\infty(\Omega)}^2\|v_r\|_0^2 - \frac{\delta_r}{2}\|\boldsymbol{b}\cdot\nabla v_r\|_0^2.
$$

For estimating the term with the Laplacian, one can either use the finite element inverse estimate (4.13) or the POD inverse estimate (4.16). Inserting in either case of the restrictions on the stabilization parameter proves the statement of the lemma. □

Note that the second restriction in (4.19) and (4.20) is only necessary if finite elements are used where the restriction of the Laplacian to a mesh cell does not vanish.

4.3.4. On Error Estimates Involving the SUPG-ROM Solution

The first approach considers the error between u^n, the solution of the continuous problem (2.2) evaluated at time step t_n, and the SUPG-ROM solution u_{ro}^n of (4.10) directly, using the splitting

$$
u^n - u_{\mathrm{ro}}^n = \left(u^n - P_r(u_h^n)\right) + \left(P_r(u_h^n) - u_{\mathrm{ro}}^n\right) = \eta^n - \zeta_r^n. \tag{4.24}
$$

Subtracting the SUPG-ROM problem (4.10) from the continuous equation (2.2) and using $\zeta_r^n \in X_r \subset X$ as a test function yields

$$
\begin{aligned}
&\left(\partial_t u^n - \frac{u_r^n - u_r^{n-1}}{\Delta t}, \zeta_r^n\right) + \left(\varepsilon\nabla u^n, \nabla\zeta_r^n\right) + \left(\boldsymbol{b}^n\cdot\nabla u^n, \zeta_r^n\right) + \left(c^n u^n, \zeta_r^n\right) \\
&= a_{\mathrm{SUPG},r}(u_{\mathrm{ro}}^n, \zeta_r^n) + \delta_r \left(\frac{u_r^n - u_r^{n-1}}{\Delta t}, \boldsymbol{b}^n\cdot\nabla\zeta_r^n\right) - \delta_r\left(f^n, \boldsymbol{b}^n\cdot\nabla\zeta_r^n\right).
\end{aligned} \tag{4.25}
$$

Adding the residual of (2.1) at time step t_n tested by $\boldsymbol{b}^n\cdot\nabla\zeta_r^n$ and multiplied by δ_r, i.e.,

$$
\delta_r\left(u_t^n - \varepsilon\Delta u^n + \boldsymbol{b}^n\cdot\nabla u^n + c^n u^n - f^n, \boldsymbol{b}^n\cdot\nabla\zeta_r^n\right),
$$

to (4.25), and using the fact that it vanishes almost everywhere, one obtains

$$\begin{aligned}a_{\mathrm{SUPG},r}(u_{\mathrm{ro}}^n,\zeta_r^n) =&a_{\mathrm{SUPG},r}(u^n,\zeta_r^n)+\left(\partial_t u^n-\frac{u_r^n-u_r^{n-1}}{\Delta t},\zeta_r^n\right)\\&+\delta_r\left(\partial_t u^n-\frac{u_r^n-u_r^{n-1}}{\Delta t},\boldsymbol{b}^n\cdot\nabla\zeta_r^n\right).\end{aligned}\tag{4.26}$$

A modification of (4.26) by adding

$$a_{\mathrm{SUPG},r}(P_r(u_h^n),\zeta_r^n)-a_{\mathrm{SUPG},r}(P_r(u_h^n),\zeta_r^n)=0$$

and using the definition of η^n and ζ_r^n from (4.24) results in the error equation

$$\begin{aligned}a_{\mathrm{SUPG},r}\left(\zeta_r^n,\zeta_r^n\right)=&\ a_{\mathrm{SUPG},r}\left(\eta^n,\zeta_r^n\right)+\left(\partial_t u^n-\frac{u_r^n-u_r^{n-1}}{\Delta t},\zeta_r^n\right)\\&+\delta_r\left(\partial_t u^n-\frac{u_r^n-u_r^{n-1}}{\Delta t},\boldsymbol{b}^n\cdot\nabla\zeta_r^n\right).\end{aligned}\tag{4.27}$$

The second approach considers the error between the SUPG finite element solution u_h^n of (2.18) with the backward Euler method and the SUPG-ROM solution u_{ro}^n of (4.10) and uses the decomposition

$$u_h^n-u_{\mathrm{ro}}^n=(u_h^n-P_r(u_h^n))+(P_r(u_h^n)-u_{\mathrm{ro}}^n)=\eta_h^n-\zeta_r^n.\tag{4.28}$$

Since $X_r\subset X_h$, POD functions can be used as test functions in the finite element problem (2.18). By subtracting the SUPG-ROM problem (4.10) from (2.18) using the backward Euler method, i.e., $\theta=1$, with a test function ζ_r^n, one obtains

$$\begin{aligned}&(u_h^n-u_{\mathrm{ro}}^n,\zeta_r^n)-(u_h^{n-1}-u_{\mathrm{ro}}^{n-1},\zeta_r^n)+\Delta t\, a_{\mathrm{SUPG},h}(u_h^n,\zeta_r^n)-\Delta t\, a_{\mathrm{SUPG},r}(u_{\mathrm{ro}}^n,\zeta_r^n)\\&+\delta_h(u_h^n-u_h^{n-1},\boldsymbol{b}^n\cdot\nabla\zeta_r^n)-\delta_r(u_{\mathrm{ro}}^n-u_{\mathrm{ro}}^{n-1},\boldsymbol{b}^n\cdot\nabla\zeta_r^n)=\Delta t(\delta_h-\delta_r)(f^n,\boldsymbol{b}^n\cdot\nabla\zeta_r^n).\end{aligned}\tag{4.29}$$

The third and the fourth term in (4.29) can be reformulated by adding a zero term

$$(\delta_r-\delta_r)\sum_{K\in\mathcal{T}_h}(-\varepsilon\Delta u_h^n+\boldsymbol{b}^n\cdot\nabla u_h^n+c^n u_h^n,\boldsymbol{b}^n\cdot\nabla\zeta_r^n)_K$$

as follows

$$\begin{aligned}&\Delta t\, a_{\mathrm{SUPG},h}(u_h^n,\zeta_r^n)-\Delta t\, a_{\mathrm{SUPG},r}(u_{\mathrm{ro}}^n,\zeta_r^n)=\Delta t\, a_{\mathrm{SUPG},r}(u_h^n-u_{\mathrm{ro}}^n,\zeta_r^n)\\&+\Delta t(\delta_h-\delta_r)\sum_{K\in\mathcal{T}_h}(-\varepsilon\Delta u_h^n+\boldsymbol{b}^n\cdot\nabla u_h^n+c^n u_h^n,\boldsymbol{b}^n\cdot\nabla\zeta_r^n)_K.\end{aligned}\tag{4.30}$$

Using the error definition (4.28) and inserting (4.30) in (4.29) gives

$$\begin{aligned}&(\eta_h^n,\zeta_r^n)-\|\zeta_r^n\|_0^2-(\eta_h^{n-1},\zeta_r^n)+(\zeta_h^{n-1},\zeta_r^n)+\Delta t\, a_{\mathrm{SUPG},r}(\eta_h^n-\zeta_r^n,\zeta_r^n)\\&+\Delta t(\delta_h-\delta_r)\sum_{K\in\mathcal{T}_h}(-\varepsilon\Delta u_h^n+\boldsymbol{b}^n\cdot\nabla u_h^n+c^n u_h^n,\boldsymbol{b}^n\cdot\nabla\zeta_r^n)_K\\&+\delta_h(u_h^n-u_h^{n-1},\boldsymbol{b}^n\cdot\nabla\zeta_r^n)-\delta_r(u_{\mathrm{ro}}^n-u_{\mathrm{ro}}^{n-1},\boldsymbol{b}^n\cdot\nabla\zeta_r^n)=\Delta t(\delta_h-\delta_r)(f^n,\boldsymbol{b}^n\cdot\nabla\zeta_r^n).\end{aligned}\tag{4.31}$$

The right-hand side of (4.31) can be rewritten by replacing f^n by the solution of the continuous problem (2.1) at time t^n, i.e.,

$$(f^n, \boldsymbol{b}^n \cdot \nabla \zeta_r^n) = \sum_{K\in\mathcal{T}_h} (\partial_t u^n - \varepsilon\Delta u^n + \boldsymbol{b}^n \cdot \nabla u^n + c^n u^n, \boldsymbol{b}^n \cdot \nabla \zeta_r^n)_K.$$

The first and the third term on the left-hand side of (4.31) vanish as $P_r(u_h^n)$ is a L^2 projection of u_h^n into X_r. Finally, the error equation at time t^n reads

$$\begin{aligned}&\|\zeta_r^n\|_0^2 + \Delta t\, a_{\mathrm{SUPG},r}(\zeta_r^n, \zeta_r^n) = \left(\zeta_r^{n-1}, \zeta_r^n\right) + \Delta t\, a_{\mathrm{SUPG},r}(\eta_h^n, \zeta_r^n)\\ &+ \Delta t(\delta_h - \delta_r) \sum_{K\in\mathcal{T}_h} (-\varepsilon\Delta(u_h^n - u^n) + \boldsymbol{b}^n \cdot \nabla(u_h^n - u^n) + c(u_h^n - u^n), \boldsymbol{b}^n \cdot \nabla \zeta_r^n)_K\\ &+ \delta_h\left(u_h^n - u_h^{n-1} - \Delta t \partial_t u^n, \boldsymbol{b}^n \cdot \nabla \zeta_r^n\right) - \delta_r\left(u_r^n - u_r^{n-1} - \Delta t \partial_t u^n, \boldsymbol{b}^n \cdot \nabla \zeta_r^n\right).\end{aligned} \tag{4.32}$$

Ideally, one would obtain optimal choices for δ_r by deriving an error estimate from (4.27) or (4.32). However, it is not known if such a derivation is possible. Even if it is possible, one has to expect that in the general case δ_r depends on the length of the time step like for the finite element error estimate in [81]. Numerical evidence for simulations of the finite element problem shows that the stabilization parameter should not depend on the length of the time step; see the discussion of this topic in Section 2.1.3. In [81], an error estimate with the stabilization parameter depending on the mesh width as in the steady-state case was proven in a simplified case. Here, also a simplified situation will be considered, which also ensures that the stabilization parameter does not depend on the length of the time step: The steady-state situation will be studied. To this end, all dependencies of the previous results on the length of the time step will be neglected in the following. Thus, the error equation (4.27) simplifies to

$$a_{\mathrm{SUPG},r}(\zeta_r^n, \zeta_r^n) = a_{\mathrm{SUPG},r}(\eta^n, \zeta_r^n) \tag{4.33}$$

and the error equation (4.32) to

$$\begin{aligned}&a_{\mathrm{SUPG},r}(\zeta_r^n, \zeta_r^n) = a_{\mathrm{SUPG},r}(\eta_h^n, \zeta_r^n)\\ &+ (\delta_h - \delta_r) \sum_{K\in\mathcal{T}_h} (-\varepsilon\Delta(u_h^n - u^n) + \boldsymbol{b}^n \cdot \nabla(u_h^n - u^n) + c^n(u_h^n - u^n), \boldsymbol{b}^n \cdot \nabla \zeta_r^n)_K.\end{aligned} \tag{4.34}$$

Stabilization Parameters Obtained with (4.33)

By the definition of the bilinear form $a_{\mathrm{SUPG},r}(\cdot,\cdot)$ in (4.3), the right-hand side of (4.33) reads as follows

$$\begin{aligned}&(\varepsilon\nabla\eta^n, \nabla\zeta_r^n) + (\boldsymbol{b}^n \cdot \nabla\eta^n, \zeta_r^n) + (c^n\eta^n, \zeta_r^n)\\ &+ \delta_r \sum_{K\in\mathcal{T}_h} (-\varepsilon\Delta\eta^n + \boldsymbol{b}^n \cdot \nabla\eta^n + c^n\eta^n, \boldsymbol{b}^n \cdot \nabla\zeta_r^n)_K,\end{aligned} \tag{4.35}$$

and its estimate is obtained in the same way as it can be found in the literature, e.g., see [118]. With the Cauchy–Schwarz inequality and (4.22), the first term in (4.35) can be estimated by

$$
\begin{aligned}
|\varepsilon\left(\nabla \eta^n, \nabla \zeta_r^n\right)| &\leq \varepsilon \|\nabla \eta^n\|_0 \|\nabla \zeta_r^n\|_0 \\
&\leq \varepsilon^{1/2} \|\nabla \eta^n\|_0 \left(\varepsilon \|\nabla \zeta_r^n\|_0^2\right)^{1/2} \\
&\leq \varepsilon^{1/2} \|\nabla \eta^n\|_0 |||\zeta_r^n|||_{\mathrm{SUPG},r}.
\end{aligned}
\tag{4.36}
$$

Using integration by parts for the convective term, the triangle inequality, the Cauchy–Schwarz inequality, and (4.22) gives the estimates of the second and the third term of (4.35) which read as

$$
\begin{aligned}
|(\boldsymbol{b}^n \cdot \nabla \eta^n, \zeta_r^n)| &\leq |((\nabla \cdot \boldsymbol{b}^n)\eta^n, \zeta_r^n)| + |(\eta^n, \boldsymbol{b}^n \cdot \nabla \zeta_r^n)| \\
&\leq \|\nabla \cdot \boldsymbol{b}^n\|_{L^\infty(\Omega)} \|\eta^n\|_0 \|\zeta_r^n\|_0 + \|\eta^n\|_0 \|\boldsymbol{b}^n \cdot \nabla \zeta_r^n\|_0 \\
&= \frac{\|\nabla \cdot \boldsymbol{b}^n\|_{L^\infty(\Omega)}}{c_0^{1/2}} \|\eta^n\|_0 \left(c_0 \|\zeta_r^n\|_0^2\right)^{1/2} + \delta_r^{-1/2} \|\eta^n\|_0 \left(\delta_r \|\boldsymbol{b}^n \cdot \nabla \zeta_r^n\|_0^2\right)^{1/2} \\
&\leq \left(\frac{\|\nabla \cdot \boldsymbol{b}^n\|_{L^\infty(\Omega)}}{c_0^{1/2}} + \delta_r^{-1/2}\right) \|\eta^n\|_0 |||\zeta_r^n|||_{\mathrm{SUPG},r}
\end{aligned}
\tag{4.37}
$$

and

$$
\begin{aligned}
|(c^n \eta^n, \zeta_r^n)| &\leq \|c^n\|_{L^\infty(\Omega)} \|\eta^n\|_0 \|\zeta_r^n\|_0 \\
&= \frac{\|c^n\|_{L^\infty(\Omega)}}{c_0^{1/2}} \|\eta^n\|_0 \left(c_0 \|\zeta_r^n\|_0^2\right)^{1/2} \\
&\leq \frac{\|c^n\|_{L^\infty(\Omega)}}{c_0^{1/2}} \|\eta^n\|_0 |||\zeta_r^n|||_{\mathrm{SUPG},r}.
\end{aligned}
\tag{4.38}
$$

The estimates of the other terms in (4.35) can be obtained in the similar way resulting in

$$
\begin{aligned}
\delta_r \sum_{K \in \mathcal{T}_h} \varepsilon\left(-\Delta \eta^n, \boldsymbol{b}^n \cdot \nabla \zeta_r^n\right)_K &\leq \delta_r \varepsilon \left(\sum_{K \in \mathcal{T}_h} \|\Delta \eta^n\|_{0,K}^2\right)^{1/2} \|\boldsymbol{b}^n \cdot \nabla \zeta_r^n\|_0 \\
&\leq \delta_r^{1/2} \varepsilon \left(\sum_{K \in \mathcal{T}_h} \|\Delta \eta^n\|_{0,K}^2\right)^{1/2} \left(\delta_r \|\boldsymbol{b}^n \cdot \nabla \zeta_r^n\|_0^2\right)^{1/2} \\
&\leq \delta_r^{1/2} \varepsilon \left(\sum_{K \in \mathcal{T}_h} \|\Delta \eta^n\|_{0,K}^2\right)^{1/2} |||\zeta_r^n|||_{\mathrm{SUPG},r},
\end{aligned}
\tag{4.39}
$$

$$
\begin{aligned}
\delta_r\left(\boldsymbol{b}^n \cdot \nabla \eta^n, \boldsymbol{b}^n \cdot \nabla \zeta_r^n\right) &\leq \delta_r^{1/2} \|\boldsymbol{b}^n \cdot \nabla \eta^n\|_0 \left(\delta_r \|\boldsymbol{b}^n \cdot \nabla \zeta_r^n\|_0\right)^{1/2} \\
&\leq \delta_r^{1/2} \|\boldsymbol{b}^n \cdot \nabla \eta^n\|_0 |||\zeta_r^n|||_{\mathrm{SUPG},r},
\end{aligned}
\tag{4.40}
$$

and

$$
\begin{aligned}
\delta_r\left(c^n \eta^n, \boldsymbol{b}^n \cdot \nabla \zeta_r^n\right) &\leq \delta_r \|c^n\|_{L^\infty(\Omega)} \|\eta^n\|_0 \|\boldsymbol{b}^n \cdot \nabla \zeta_r^n\|_0 \\
&\leq \delta_r^{1/2} \|c^n\|_{L^\infty(\Omega)} \|\eta^n\|_0 |||\zeta_r^n|||_{\mathrm{SUPG},r}.
\end{aligned}
\tag{4.41}
$$

Using the coercivity (4.21) of the SUPG-ROM bilinear form, inserting (4.36)–(4.41) into (4.33), and including all data from convection and reaction into the constant yields

$$\begin{aligned}\|\!|\zeta_r^n|\!\|_{\mathrm{SUPG},r} \le C\Bigg[&\left(1+\delta_r^{-1/2}+\delta_r^{1/2}\right)\|\eta^n\|_0+\varepsilon^{1/2}\|\nabla\eta^n\|_0\\ &+\delta_r^{1/2}\|\boldsymbol{b}^n\cdot\nabla\eta^n\|_0+\delta_r^{1/2}\varepsilon\left(\sum_{K\in\mathcal{T}_h}\|\Delta\eta^n\|_{0,K}^2\right)^{1/2}\Bigg].\end{aligned} \tag{4.42}$$

The second factor on the right-hand side of (4.42) shall be minimized, thereby providing information about an appropriate choice of the stabilization parameter.

The straightforward approach consists in decomposing

$$\eta^n = u^n - P_r(u_h^n) = (u^n - u_h^n) + (u_h^n - P_r(u_h^n)) \tag{4.43}$$

and using the error estimates (2.30) and (2.32) for the first part and the estimate (4.18) (the finite element option) or (4.17) (the POD option) for the second part.

The Finite Element Option. Here, in order to estimate the norms including η^n in (4.42), the error estimates (2.30) and (2.32) for the first part and the estimate (4.18) for the second part of the decomposition of η^n in (4.43) will be used. Thus, one gets

$$\begin{aligned}\|\eta^n\|_0 &\le C(h^{m+1/2}+\Lambda_0),\\ \|\nabla\eta^n\|_0 &\le C(\varepsilon^{-1/2}h^{m+1/2}+h^{-1}\Lambda_0),\\ \|\boldsymbol{b}^n\cdot\nabla\eta^n\|_0 &\le C(\delta_h^{-1/2}h^{m+1/2}+h^{-1}\Lambda_0),\\ \left(\sum_{K\in\mathcal{T}_h}\|\Delta\eta^n\|_{0,K}^2\right)^{1/2} &\le C(h^{m-1}+\varepsilon^{-1/2}h^{m-1/2}+h^{-2}\Lambda_0),\end{aligned}$$

where

$$\Lambda_0 = \left(\sum_{j=r+1}^{R}\lambda_j\right)^{1/2}. \tag{4.44}$$

The term to minimize becomes

$$\begin{aligned}g(\delta_r) = &\left(1+\delta_r^{-1/2}+\delta_r^{1/2}\right)\left(h^{m+1/2}+\Lambda_0\right)+h^{m+1/2}+\varepsilon^{1/2}h^{-1}\Lambda_0+\delta_r^{1/2}\delta_h^{-1/2}h^{m+1/2}\\ &+\delta_r^{1/2}h^{-1}\Lambda_0+\delta_r^{1/2}\varepsilon h^{m-1}+\delta_r^{1/2}\varepsilon^{1/2}h^{m-1/2}+\delta_r^{1/2}\varepsilon h^{-2}\Lambda_0.\end{aligned}$$

Calculating the first derivative of g with $\delta_h = h$ yields

$$\begin{aligned}g'(\delta_r) =&\frac{1}{2}\delta_r^{-3/2}\Big[-\left(h^{m+1/2}+\Lambda_0\right)+\delta_r\Big(h^{m-1}(h^{3/2}+h+\varepsilon+\varepsilon^{1/2}h^{1/2})\\ &+\Lambda_0(1+h^{-1}+\varepsilon h^{-2})\Big)\Big].\end{aligned}$$

By setting it to zero, one obtains the extremum

$$\delta_r^* = \frac{h^{m+1/2} + \Lambda_0}{h^{m-1}(h^{3/2} + h + \varepsilon + \varepsilon^{1/2}h^{1/2}) + \Lambda_0(1 + h^{-1} + \varepsilon h^{-2})}. \tag{4.45}$$

The second derivative of g has the following form:

$$\begin{aligned} g''(\delta_r) = &-\frac{3}{4}\delta_r^{-5/2}\Big[-\Big(h^{m+1/2} + \Lambda_0\Big) + \delta_r\Big(h^{m-1}(h^{3/2} + h + \varepsilon + \varepsilon^{1/2}h^{1/2}) \\ &+ \Lambda_0(1 + h^{-1} + \varepsilon h^{-2})\Big)\Big] + \frac{1}{2}\delta_r^{-3/2}\Big[h^{m-1}(h^{3/2} + h + \varepsilon + \varepsilon^{1/2}h^{1/2}) \\ &+ \Lambda_0(1 + h^{-1} + \varepsilon h^{-2})\Big] \end{aligned}$$

with

$$g''(\delta_r^*) = \frac{1}{2}\frac{\big(h^{m+1/2} + \Lambda_0\big)^{-3/2}}{\big(h^{m-1}(h^{3/2} + h + \varepsilon + \varepsilon^{1/2}h^{1/2}) + \Lambda_0(1 + h^{-1} + \varepsilon h^{-2})\big)^{-5/2}} > 0.$$

Consequently, δ_r^* is a minimum of g. Concentrating on the convection-dominated case $\varepsilon \ll h$ and for grid widths $h < 1$, δ_r^* in (4.45) can be reduced to

$$h\frac{h^{m+1/2} + \Lambda_0}{h^{m+1} + \Lambda_0}.$$

Numerical evidence, e.g., Remark 4.7 below, shows that Λ_0 dominates the finite element errors, which are theoretically of order $h^{m+1/2}$, see (2.30). Hence, in this case the stabilization parameter becomes

$$\delta_r^{\mathrm{FE}} = h. \tag{4.46}$$

The SUPG-ROM using δ_r^{FE} is denoted by FE-SUPG-ROM.

Remark 4.6. Some remarks on (4.46) are as follows:

- In the convection-dominated case, condition (4.19) for the coercivity of the SUPG-ROM bilinear form will be satisfied for δ_r^{FE}.

- There is an explicit impact of the setup for simulating the snapshots onto the stabilization used in the SUPG-ROM. It is not clear, if this situation is always desirable, e.g., if the snapshots were computed on a very fine mesh, there would be only a weak stabilization in the SUPG-ROM.

- For using δ_r^{FE}, one has to know the mesh width. If even the mesh itself is known, then it is possible to use also the local mesh width in assembling the terms for the stabilization, like usually done in the finite element method.

- There is no impact of the number of used snapshots or POD modes on δ_r^{FE}.

◁

The POD Option. Here, the norms of the second part of the decomposition (4.43) are estimated by applying (4.17) instead of (4.18).

Hypothesis 4.1. *Let the finite element simulation be sufficiently accurate or let sufficiently few POD basis functions be used in the ROM, such that the norms on the left-hand sides of* (2.30) *and* (2.32) *can be estimated by a constant multiplied by the respective terms of the right-hand side of* (4.17)*.*

This hypothesis implies that there is a constant C_Λ such that

$$\begin{aligned} \|\eta^n\|_0 &\le C_\Lambda \left(\sum_{j=r+1}^{R} \lambda_j \right)^{1/2} &&= C_\Lambda \Lambda_0, \\ \|\nabla \eta^n\|_0 &\le C_\Lambda \left(\sum_{j=r+1}^{R} \lambda_j |\varphi_{\mathrm{ro},j}|_1^2 \right)^{1/2} &&= C_\Lambda \Lambda_1, \\ \left(\sum_{K \in \mathcal{T}_h} \|\Delta \eta^n\|_{0,K}^2 \right)^{1/2} &\le C_\Lambda \left(\sum_{j=r+1}^{R} \lambda_j |\varphi_{\mathrm{ro},j}|_2^2 \right)^{1/2} &&= C_\Lambda \Lambda_2. \end{aligned} \tag{4.47}$$

Note that the constant C_Λ will cancel in further calculations such that its actual value does not influence the final result.

Remark 4.7. Considering the same problem as in Remark 4.5, Figure 4.2 shows the discrete versions of the $L^1(0,T;L^2(\Omega))$ and the $L^1(0,T;H_0^1(\Omega))$ errors of the finite element solution defined by

$$\frac{1}{N} \sum_{n=1}^{N} \|u^n - u_h^n\|_0 \quad \text{and} \quad \frac{1}{N} \sum_{n=1}^{N} \|\nabla(u^n - u_h^n)\|_0, \tag{4.48}$$

respectively. The errors are computed for the finite element solution on different meshes (levels 6, 7, and 8 are introduced in the beginning of Section 4.4) and for the corresponding values Λ_0 and Λ_1 for different dimensions of the ROM basis. One can observe that in these cases Hypothesis 4.1 is satisfied. ◁

Using the estimate $\|\boldsymbol{b}^n \cdot \nabla \eta^n\|_0 \le C \|\nabla \eta^n\|_0$, the factor to be minimized in (4.42) has the form

$$g(\delta_r) = \left(1 + \delta_r^{-1/2} + \delta_r^{1/2}\right) C_\Lambda \Lambda_0 + \left(\varepsilon^{1/2} + \delta_r^{1/2}\right) C_\Lambda \Lambda_1 + \delta_r^{1/2} \varepsilon C_\Lambda \Lambda_2.$$

Computing its first derivative yields

$$g'(\delta_r) = \frac{1}{2} \delta_r^{-3/2} \left(\delta_r \left(C_\Lambda \Lambda_0 + C_\Lambda \Lambda_1 + \varepsilon C_\Lambda \Lambda_2 \right) - C_\Lambda \Lambda_0 \right),$$

which leads to the extremum

$$\delta_r^* = \frac{\Lambda_0}{\Lambda_0 + \Lambda_1 + \varepsilon \Lambda_2}. \tag{4.49}$$

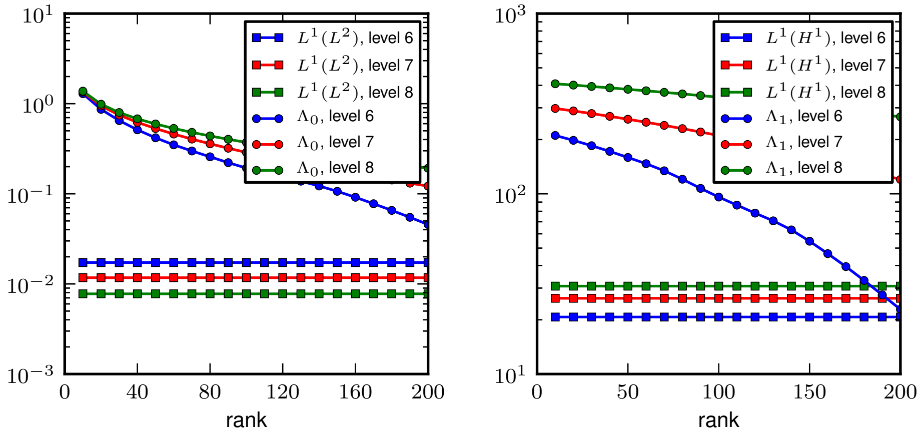

Figure 4.2.: Remark 4.7: Numerical verification of Hypothesis 4.1 with the help of the test example in Remark 4.5.

The second derivative of g reads

$$\begin{aligned} g''(\delta_r) &= -\frac{3}{4}\delta_r^{-5/2}\left(\delta_r\left(C_\Lambda\Lambda_0 + C_\Lambda\Lambda_1 + \varepsilon C_\Lambda\Lambda_2\right) - C_\Lambda\Lambda_0\right) \\ &\quad + \frac{1}{2}\delta_r^{-3/2}\left(C_\Lambda\Lambda_0 + C_\Lambda\Lambda_1 + \varepsilon C_\Lambda\Lambda_2\right) \\ &= -\frac{1}{4}\delta_r^{-3/2}\left(C_\Lambda\Lambda_0 + C_\Lambda\Lambda_1 + \varepsilon C_\Lambda\Lambda_2\right) + \frac{3}{4}\delta_r^{-5/2}C_\Lambda\Lambda_0, \end{aligned}$$

with

$$g''(\delta_r^*) = \frac{1}{2}\frac{\left(C_\Lambda\Lambda_0\right)^{-3/2}}{\left(C_\Lambda\Lambda_0 + C_\Lambda\Lambda_1 + \varepsilon C_\Lambda\Lambda_2\right)^{-5/2}} > 0.$$

Therefore, δ_r^* from (4.49) is in fact a minimum of the second factor in (4.42). In the convection-dominated regime, the last term in the denominator of δ_r^* is small. From estimate (4.18), it can be expected that $\Lambda_0 \ll \Lambda_1$, at least for small mesh widths. Thus, an appropriate choice of the stabilization parameter is

$$\delta_r^{\text{POD}} = \frac{\Lambda_0}{\Lambda_1}. \tag{4.50}$$

The SUPG-ROM using δ_r^{POD} is denoted by POD-SUPG-ROM.

Remark 4.8. For the problem described in Remark 4.5, the values in the denominator of (4.49) are presented in Figure 4.3. Hence, for this example, the assumptions made for reducing (4.49) to (4.50) are satisfied. The same behavior was observed also for other test cases. ◁

Remark 4.9. Some remarks on (4.50) are as follows:

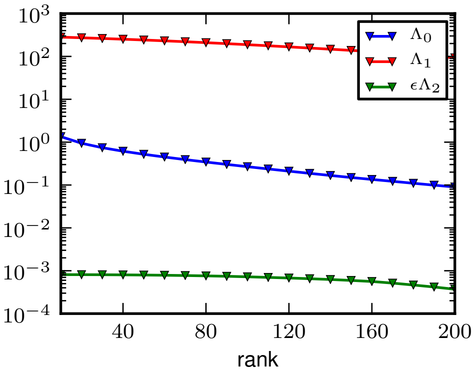

Figure 4.3.: Remark 4.8: Curves for Λ_0, Λ_1, and $\varepsilon\Lambda_2$ for the problem from Remark 4.5.

- If one of the conditions (4.19) or (4.20) for the coercivity of the SUPG-ROM bilinear form is satisfied by δ_r^{POD} is not clear a priori. In the numerical simulations carried out in the course of the underlying investigation, where the grids for computing the snapshots were not extremely fine, it was found that generally $\delta_r^{\mathrm{POD}} \leq \delta_r^{\mathrm{FE}}$, which implies the satisfaction of (4.19).
- The parameter δ_r^{POD} is influenced by the number r of used POD modes and also by the simulation for computing the snapshots, since this simulation determines the eigenvalues and eigenfunctions.
- For computing δ_r^{POD}, one has to consider all POD modes in the offline step of the ROM simulation , because they are necessary for the computation of Λ_1.
- There is no possibility to localize δ_r^{POD}.

◁

Stabilization Parameters Obtained with (4.34)

The goal of this section is to propose the scaling of the SUPG-ROM stabilization parameter based on the error equation (4.34) instead of (4.33) as it was done in the previous section. Most terms of the first term on the right-hand side of (4.34) are estimated in the same way as it was done in the previous section. Only, instead of (4.40),

$$\delta_r \left(\boldsymbol{b}^n \cdot \nabla \eta_h^n, \boldsymbol{b}^n \cdot \nabla \zeta_r^n\right) \leq C \delta_r^{1/2} \|\nabla \eta_h^n\|_0 \|\!|\zeta_r^n|\!\|_{\mathrm{SUPG},r}$$

is used, giving the upper bound

$$
\begin{aligned}
C\Bigg[&\left(1+\delta_r^{-1/2}+\delta_r^{1/2}\right)\|\eta_h^n\|_0 + \left(\varepsilon^{1/2}+\delta_r^{1/2}\right)\|\nabla\eta_h^n\|_0 \\
&+ \delta_r^{1/2}\varepsilon\left(\sum_{K\in\mathcal{T}_h}\|\Delta\eta_h^n\|_{0,K}^2\right)^{1/2}\Bigg]|||\zeta_r^n|||_{\mathrm{SUPG},r},
\end{aligned}
$$

where $\eta_h^n = u_h^n - P_r(u_h^n)$ is defined in (4.28).

For the estimation of the second term of (4.34), the finite element error estimates (2.30) and (2.32) are utilized, leading to the upper bound

$$
C|\delta_h-\delta_r|\delta_r^{-1/2}\left(\varepsilon h^{m-1}+\varepsilon^{1/2}h^{m-1/2}+\delta_h^{-1/2}h^{m+1/2}+h^{m+1/2}\right)|||\zeta_r^n|||_{\mathrm{SUPG},r}.
$$

Inserting these bounds into (4.34) and using the coercivity (4.21) of the bilinear form $a_{\mathrm{SUPG},r}(\cdot,\cdot)$ and $\delta_h = h$ results in the estimate

$$
\begin{aligned}
|||\zeta_r^n|||_{\mathrm{SUPG},r} \le C\Bigg[&\left(1+\delta_r^{-1/2}+\delta_r^{1/2}\right)\|\eta_h^n\|_0 + \left(\varepsilon^{1/2}+\delta_r^{1/2}\right)\|\nabla\eta_h^n\|_0 \\
&+ \delta_r^{1/2}\varepsilon\left(\sum_{K\in\mathcal{T}_h}\|\Delta\eta_h^n\|_{0,K}^2\right)^{1/2} + |h-\delta_r|\delta_r^{-1/2} \qquad (4.51)\\
&\times\left(\varepsilon h^{m-1}+\varepsilon^{1/2}h^{m-1/2}+h^m+h^{m+1/2}\right)\Bigg].
\end{aligned}
$$

The second factor on the right-hand side of (4.51) has to be minimized in order to determine the SUPG-ROM parameter.

The Finite Element Option. In order to estimate the norms including η_h^n in (4.51), the estimate (4.18) will be used here. Thus, one obtains

$$
\|\eta_h^n\|_0 \le C\Lambda_0,\quad \|\nabla\eta_h^n\|_0 \le Ch^{-1}\Lambda_0,\quad \left(\sum_{K\in\mathcal{T}_h}\|\Delta\eta_h^n\|_{0,K}^2\right)^{1/2} \le Ch^{-2}\Lambda_0,
$$

where

$$
\Lambda_0 = \left(\sum_{j=r+1}^{R}\lambda_j\right)^{1/2}.
$$

Therefore, one has to minimize the function

$$
\begin{aligned}
g(\delta_r) = &\left(1+\delta_r^{-1/2}+\delta_r^{1/2}\right)\Lambda_0 + \left(\varepsilon^{1/2}+\delta_r^{1/2}\right)h^{-1}\Lambda_0 + \delta_r^{1/2}\varepsilon h^{-2}\Lambda_0 \\
&+ |h-\delta_r|\delta_r^{-1/2}C_0,
\end{aligned}
$$

where $C_0 := \varepsilon h^{m-1} + \varepsilon^{1/2}h^{m-1/2} + h^m + h^{m+1/2}$.

The first derivative of g has the form

$$g'(\delta_r) = \frac{1}{2}\delta_r^{-3/2}\Big[\Lambda_0\left(-1+\delta_r(1+h^{-1}+\varepsilon h^{-2})\right) - \operatorname{sgn}(h-\delta_r)C_0\left(2\delta_r + |h-\delta_r|\right)\Big].$$

To determine the minimum of $g(\delta_r)$, the case-by-case analysis will be employed.

$\boxed{\text{Case } \delta_r < h.}$ In this case the first derivative of g is

$$g'(\delta_r) = \frac{1}{2}\delta_r^{-3/2}\Big[\Lambda_0\left(-1+\delta_r(1+h^{-1}+\varepsilon h^{-2})\right) - C_0\left(\delta_r + h\right)\Big].$$

Setting it to zero yields the extremum

$$\delta_r^* = \frac{\Lambda_0 + hC_0}{\Lambda_0\left(1+h^{-1}+\varepsilon h^{-2}\right) - C_0}.$$

The second derivative of g is

$$g''(\delta_r) = \frac{1}{2}\delta_r^{-5/2}\Big[\frac{3}{2}\left(\Lambda_0 + C_0 h\right) - \frac{1}{2}\left(\Lambda_0\left(1+h^{-1}+\varepsilon h^{-2}\right) - C_0\right)\delta_r\Big].$$

By the construction of g, the meaningful values of δ_r^* are positive real numbers. Hence, when evaluating g'' at δ_r^*, the last factor has to be investigated. It has the form

$$\frac{3}{2}\left(\Lambda_0 + C_0 h\right) - \frac{1}{2}(\Lambda_0 + hC_0) = \Lambda_0 + C_0 h > 0.$$

Hence, δ_r^* is the minimum of g.

For the convection-dominated setting when $\varepsilon \ll h$ and $h < 1$, one can pick the relevant terms in δ_r^* which gives

$$\delta_r^* = h\frac{\Lambda_0 + h^{m+1} + h^{m+3/2}}{\Lambda_0 - h^{m+1} - h^{m+3/2}}.$$

As already discussed in Section 4.3.4 for deriving δ_r^{FE}, Λ_0 dominates the finite element errors. Therefore, the stabilization parameter becomes

$$\delta_r^{FE} = h.$$

$\boxed{\text{Case } \delta_r > h.}$ In this case one obtains for the first derivative of g the expression

$$g'(\delta_r) = \frac{1}{2}\delta_r^{-3/2}\Big[\Lambda_0\left(-1+\delta_r(1+h^{-1}+\varepsilon h^{-2})\right) - C_0\left(3\delta_r - h\right)\Big].$$

The extremum obtained by setting the derivative to zero has the form

$$\delta_r^* = \frac{\Lambda_0 - C_0 h}{\Lambda_0\left(1+h^{-1}+\varepsilon h^{-2}\right) - 3C_0}.$$

The second derivative of g is

$$g''(\delta_r) = \frac{1}{2}\delta_r^{-5/2}\Big[\frac{3}{2}\left(\Lambda_0 - C_0 h\right) - \frac{1}{2}\left(\Lambda_0\left(1 + h^{-1} + \varepsilon h^{-2}\right) - 3C_0\right)\delta_r\Big].$$

Taking into account only positive values for δ_r^*, the last factor of g'', i.e., the expression in the square brackets, has to be evaluated at δ_r^* to investigate the behavior of the extremum. Due to the definition of δ_r^* it becomes $\Lambda_0 - C_0 h$. From Remark 4.7, it can be expected that for sufficiently small h and ε, which is the case in the simulations in the convection-dominated regime, the term is positive. Hence, δ_r^* is the minimum of g. In the convection-dominated regime, δ_r^* reduces to

$$\delta_r^* = h\frac{\Lambda_0 - C_0 h}{\Lambda_0 - 3C_0 h}.$$

As the second factor is of the same order, the stabilization parameter becomes

$$\delta_r^{FE} = h.$$

$\boxed{\text{Case } \delta_r = h.}$ In this case one obtains for the first derivative of g the expression

$$g'(\delta_r) = \frac{1}{2}\delta_r^{-3/2}\Lambda_0\left(-1 + \delta_r(1 + h^{-1} + \varepsilon h^{-2})\right).$$

Similar calculations as conducted for the other cases yield the extremum

$$\delta_r^* = \frac{1}{1 + h^{-1} + \varepsilon h^{-2}}.$$

The second derivative of g, which has the form

$$g''(\delta_r) = \frac{1}{2}\delta_r^{-5/2}\Big[-\frac{1}{2}\Lambda_0\left(1 + h^{-1} + \varepsilon h^{-2}\right)\delta_r + \frac{3}{2}\Lambda_0\Big],$$

has to be evaluated at δ_r^* resulting in

$$g''(\delta_r^*) = \frac{1}{2}\left(\delta_r^*\right)^{-5/2}\Lambda_0 > 0.$$

Hence, δ_r^* is the minimum of g. The value δ_r^* can be reduced in the convection-dominated regime to

$$\delta_r^{FE} = h.$$

Consequently, by using (4.34) the same stabilization parameter for the finite element option is obtained as in Section 4.3.4.

The POD Option. In this case, the norms including η_h^n in (4.51) will be estimated using (4.17). Finally, the function to minimize is

$$g(\delta_r) = \left(1 + \delta_r^{-1/2} + \delta_r^{1/2}\right) \Lambda_0 + \left(\varepsilon^{1/2} + \delta_r^{1/2}\right) \Lambda_1 + \delta_r^{1/2} \varepsilon \Lambda_2 + |h - \delta_r| \delta_r^{-1/2} C_0,$$

where $C_0 := \varepsilon h^{m-1} + \varepsilon^{1/2} h^{m-1/2} + h^m + h^{m+1/2}$, and Λ_i, $i = 1, 2, 3$, are defined in (4.47).

The first derivative of g has the form

$$g'(\delta_r) = \frac{1}{2} \delta_r^{-3/2} \Big[- \Lambda_0 + (\Lambda_0 + \Lambda_1 + \varepsilon \Lambda_2)\, \delta_r - \operatorname{sgn}(h - \delta_r) C_0 \,(2\delta_r + |h - \delta_r|) \Big].$$

Like for the finite element option, the case-by-case analysis will be used to calculate the minimum of g.

$\boxed{\text{Case } \delta_r < h.}$ In this case the first derivative of g is

$$g'(\delta_r) = \frac{1}{2} \delta_r^{-3/2} \Big[- \Lambda_0 + (\Lambda_0 + \Lambda_1 + \varepsilon \Lambda_2)\, \delta_r - C_0 \,(\delta_r + h) \Big]$$

with the extremum

$$\delta_r^* = \frac{\Lambda_0 + h C_0}{\Lambda_0 + \Lambda_1 + \varepsilon \Lambda_2 - C_0}.$$

In the convection-dominated regime, all the terms including ε can be omitted. From estimate (4.18), it can be asserted at least for small mesh widths that $\Lambda_0 \ll \Lambda_1$. Moreover, using the numerical evidence, see Remark 4.7, Λ_0 dominates the finite element error. Hence, the stabilization parameter becomes

$$\delta_r^{POD} = \frac{\Lambda_0}{\Lambda_1}.$$

$\boxed{\text{Case } \delta_r > h.}$ The first derivative of g has the form

$$g'(\delta_r) = \frac{1}{2} \delta_r^{-3/2} \Big[- \Lambda_0 + (\Lambda_0 + \Lambda_1 + \varepsilon \Lambda_2)\, \delta_r + C_0 \,(3\delta_r - h) \Big].$$

with the extremum

$$\delta_r^* = \frac{\Lambda_0 + h C_0}{\Lambda_0 + \Lambda_1 + \varepsilon \Lambda_2 + 3 C_0}.$$

Using the same argumentation for the convection-dominated regime as in the case when $\delta_r < h$, the stabilization parameter becomes

$$\delta_r^{POD} = \frac{\Lambda_0}{\Lambda_1}.$$

$\boxed{\text{Case } \delta_r = h.}$ Here, the first derivative of g has the from

$$g'(\delta_r) = \frac{1}{2} \delta_r^{-3/2} \Big[- \Lambda_0 + (\Lambda_0 + \Lambda_1 + \varepsilon \Lambda_2)\, \delta_r \Big].$$

One has the same situation when regarding the extremum as in the POD case in Section 4.3.4, which yields in the convection-dominated regime the stabilization parameter

$$\delta_r^{POD} = \frac{\Lambda_0}{\Lambda_1}.$$

Remark 4.10. Similarly to the evaluations of g'' for the finite element option, δ_r^* is the minimum of g also for the POD option for the considered three cases. ◁

4.4. Numerical Studies

The numerical investigations aim at answering the following questions:

1. Does the SUPG-ROM yield more accurate results than the G-ROM?
2. Which of the two stabilization parameters of the SUPG-ROM (derived in Section 4.3.4 and 4.3.4) yields more accurate results?
3. Does the filtering procedure (3.65) of the ROM initial condition improve the ROM results?
4. Is the accuracy of the ROM solution proportional to the accuracy of the underlying snapshots? In particular, are the ROMs based on the physically correct snapshots without under- and overshoots able to reproduce physically correct ROM solutions?

In total, the results of three test problems will be presented: A hump changing its height (Example 4.1), a traveling wave (Example 4.2), and a rotating body problem (Example 4.3). The former two tests are used to answer the questions 1 and 2. ROM simulations for a rotating body problem serve to investigate the questions 1, 3, and 4. In the numerical studies presented below, the analytical solutions of the continuous problems are known. To generate the snapshots, the Galerkin finite element method (2.7), the SUPG method (2.18) with P_k, $k = 1, 2$, finite elements, or the Lagrange interpolation of the analytical solution in the appropriate finite element space was used. Simulations with the SUPG finite element method presented in this section were performed with $\delta_h = h$ (see the discussion in Section 2.1.3). All test problems were defined in the unit square. For the coarsest grid (level 0), the unit square was divided by the diagonal from bottom left to top right into two triangles. Uniform grid refinement was applied for constructing the finer grids, resulting in 16641, 66049, 263169 degrees of freedom (including Dirichlet nodes) on levels 6, 7, and 8, respectively, with P_2 finite elements and on levels 7, 8, and 9, respectively with P_1 finite elements. In all cases, the POD modes were computed with respect to the $L^2(\Omega)$ inner product and with the centered-trajectory method, i.e., from the fluctuating parts of the snapshots.

Two types of ROMs were studied. The first one is the G-ROM, i.e., the ROM without any stabilization. It is known that solutions of convection-dominated problems can be approximated accurately with the Galerkin finite element method if the finite element space possesses sufficient information about the considered problem and its solution. This information can be used to construct suitable grids, e.g., layer-adapted grids as in [101], or to design appropriate finite element functions, e.g., exponentially fitted functions as in [125]. For the ROM, the POD modes already contain important features of the solution. The questions are if the information contained in these basis functions suffices to stabilize the G-ROM and if yes, how the accuracy of the computational results compares with the accuracy of the results obtained with the second type of the studied ROM, namely the ROM with SUPG stabilization (SUPG-ROM) given by (4.2). The same questions were posed in [109], where stabilized ROMs were used for convection-dominated convection-diffusion-reaction equations. In the “Offline-only” approach in [109], the SUPG method was used in the offline stage, but not in the online stage. Thus, the “Offline-only” approach is similar to the G-ROM considered in this thesis. In the “Offline-Online” approach in [109], the SUPG method was used both in the offline and online stages.

As already mentioned, Examples 4.1 and 4.2 deal, inter alia, with answering the second question posed at the beginning of the section. For the sake of clarity, the SUPG-ROMs using the stabilization parameters δ_r^{FE} from (4.46) and δ_r^{POD} from (4.50) will be denoted by FE-SUPG-ROM and POD-SUPG-ROM, respectively. From the practical point of view, the computation of the stabilization parameter δ_r^{FE} is much easier than the computation of δ_r^{POD}. The former parameter is equal to the mesh width h; no additional information is needed. The latter one requires the storage of all R POD modes and eigenvalues. Moreover, Λ_0 and Λ_1 have to be calculated, which can be time-consuming for problems with a high-dimensional snapshot space X_R. However, these values have to be computed only once in the offline step. In Example 4.3, the SUPG-ROM only with δ_r^{FE} will be studied. Therefore, its notation remains SUPG-ROM.

The fourth question investigated in Example 4.3 is mainly motivated by practical applications. In fact, solutions representing, e.g., concentrations are non-physical if they feature under- and overshoots. Such solutions are often not acceptable for practical purposes. In applications requiring simulations of strongly coupled systems, e.g., see [80], the non-physical solutions could cause some instabilities or completely incorrect model solutions if they serve as an input quantity to other equations. Therefore, it is of great importance to construct methods producing solutions without or at least with small under- and overshoots.

To evaluate the results of the performed simulations, several measures of interest will be monitored. Besides the so-called ”eye measure”, i.e., looking at the plots of the obtained solutions, the $L^2(\Omega)$ error at certain times and the discrete analog of the $L^1(0,T;L^2(\Omega))$ error, given, e.g., for the ROM solution by

$$\|u^n - u^n_{\mathrm{ro}}\|_0 \quad \text{and} \quad \frac{1}{N+1}\sum_{n=0}^{N}\|u^n - u^n_{\mathrm{ro}}\|_0, \tag{4.52}$$

respectively, will be considered, where u^n denotes the solution of the continuous problem at time t_n. In addition, the minimum and the maximum values of the solution for Example 4.3 will be computed in the vertices of the mesh cells evaluated at different times. The $L^2(\Omega)$ error gives some idea of the accuracy of the methods and the smearing in the numerical solutions. The minimum and the maximum values indicate the under- and overshoots of the numerical solution.

Remark 4.11. After computing the POD modes, the question is how to design a numerical method that gives a solution that is as accurate as possible in the space spanned by the first r POD modes compared with the solution of the continuous problem. Since the numerical method providing the most accurate solution in the r-dimensional space is not necessarily the method which was used for computing the snapshots, other ROM methods can be investigated as well. In the numerical studies below, the snapshots are computed in Examples 4.1 and 4.2 with the finite element SUPG method. In the ROM context, besides FE-SUPG-ROM (which corresponds to the finite element SUPG method but in the ROM context) two other methods, namely G-ROM and POD-SUPG-ROM, are used. ◁

Remark 4.12. When the solution of the continuous problem is not available, one could compare the ROM results with the solution of the simulation used to compute the snapshots. Based on own experience, one gets the smallest L^2 errors with respect to the finite element solution, when the computation of the snapshots and the ROM involve the same numerical methods. Note that the method is usually not the best approximation of the continuous solution in the finite element space. ◁

The code MooNMD [78] was utilized to perform the numerical experiments.

Example 4.1. *Hump changing its height.* This example is taken from [84]. It is defined in $\Omega = (0,1)^2$ and $(0,T) = (0,2)$. The coefficients of (2.1) were chosen to be $\varepsilon = 10^{-6}$, $\boldsymbol{b} = (2,3)^T$ and $c = 1$. There is a prescribed solution of the form

$$\begin{aligned} u(t,x,y) =& 16\ \sin(\pi t)\, x(1-x)y(1-y) \\ & \times \left[\frac{1}{2} + \frac{\arctan\left(2\varepsilon^{-1/2}\left(0.25^2 - (x-0.5)^2 - (y-0.5)^2\right)\right)}{\pi}\right]. \end{aligned} \tag{4.53}$$

The forcing term f, the initial condition u_0, and the boundary conditions were set such that (4.53) satisfies the boundary value problem. The solution (4.53) possesses an internal layer of size $\mathcal{O}\left(\sqrt{\varepsilon}\right)$.

The finite element problem for computing the snapshots was solved on level 7 with P_2 finite elements, such that $h = 1.1 \cdot 10^{-2}$, and the backward Euler scheme was applied with the time step $\Delta t = 10^{-3}$. Since the problem is convection-dominated and the solution has a layer, the use of a stabilized discretization is necessary, see Figure 4.4 for a comparison of snapshots from the Galerkin finite element method and the SUPG method (2.18). Whereas the solution of the Galerkin finite element method is globally polluted with spurious oscillations, there are only few oscillations, mainly in the right upper part of the domain, in the solution computed with the SUPG method.

For computing the POD basis, every fifth solution was stored such that 401 snapshots were used. If the finite element method accurately resolved all layers, then the POD

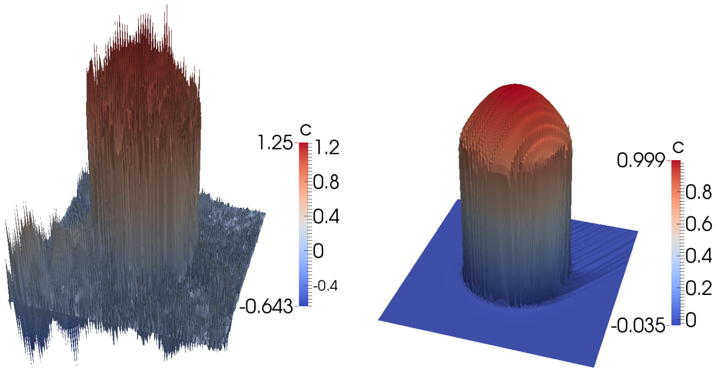

Figure 4.4.: Example 4.1: Galerkin finite element method (left), SUPG finite element method (right) at $t = 0.5$.

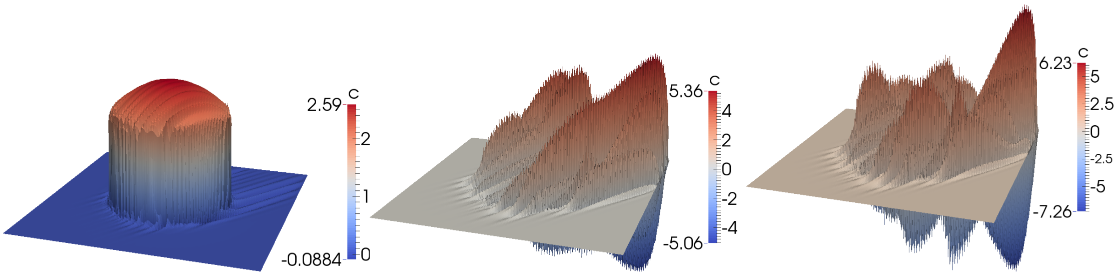

Figure 4.5.: Example 4.1: POD modes $\varphi_{\mathrm{ro},1}, \varphi_{\mathrm{ro},2}$, and $\varphi_{\mathrm{ro},3}$ (from left to right).

basis would consist just of one mode representing the time-independent part of (4.53). However, such a method is not known so far and in the simulations of convection-dominated problems one always has to expect numerical artifacts. With the spurious oscillations of the SUPG method, one obtains 14 POD modes, see Figure 4.5 for the first POD modes and Figure 4.6 for the corresponding eigenvalues, where the POD modes for $r > 1$ come from the spurious oscillations of the finite element solution.

Figures 4.7 and 4.8 present results for the three considered ROMs. In Figure 4.7, the temporal evolution of the error in the $L^2(\Omega)$ norm for $r = 9$ and the errors in the discrete $L^1(0, T; L^2(\Omega))$ norm are shown. Corresponding numerical solutions for $r = 9$ are depicted in Figure 4.8.

Before answering the first two questions posed in the beginning of this section, the following important observation should be made: The $L^1(0, T; L^2(\Omega))$ error in the right panel of Figure 4.7 shows that adding more POD modes that represent oscillations (i.e., POD modes $\varphi_{\mathrm{ro},2}, \varphi_{\mathrm{ro},3}, \ldots, \varphi_{\mathrm{ro},14}$) results in a continuous increase of the error. This behavior is in clear contrast with the standard POD-ROM experience, where adding more POD modes generally reduces the error. The reason for this behavior is that the POD uses noisy data, resulting in POD modes which contain mostly information on the numerical artifacts of the finite element solution. As already discussed above, with most of the finite element methods, the appearance of such modes is inevitable. Unlike the

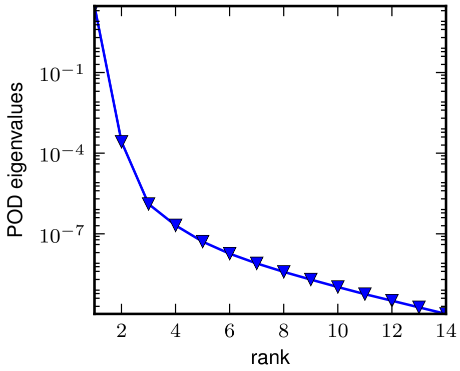

Figure 4.6.: Example 4.1: POD eigenvalues.

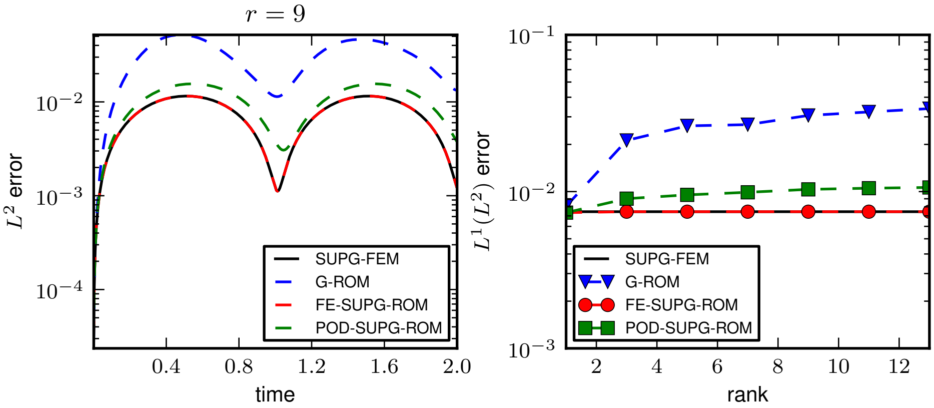

Figure 4.7.: Example 4.1: Errors for different ROMs, $L^2(\Omega)$ error (left), $L^1(0,T;L^2(\Omega))$ error (right).

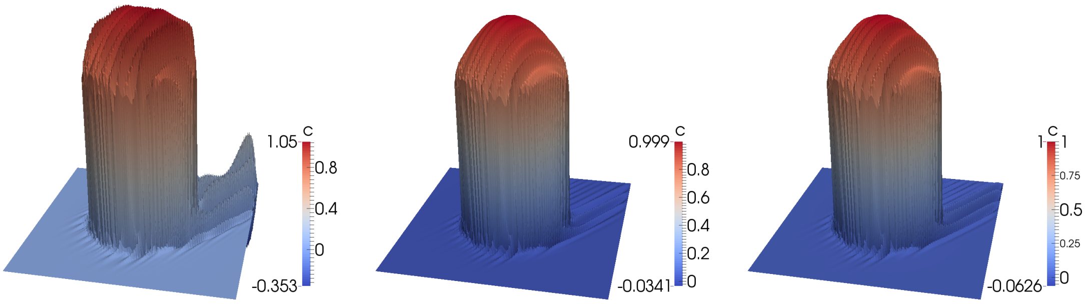

Figure 4.8.: Example 4.1: Solution at $t = 0.5$ for G-ROM, FE-SUPG-ROM, POD-SUPG-ROM (from left to right) for $r = 9$.

present example, in general it is not known which modes are strongly influenced by the noise. In adding more and more POD modes, it is important that the ROM can suppress

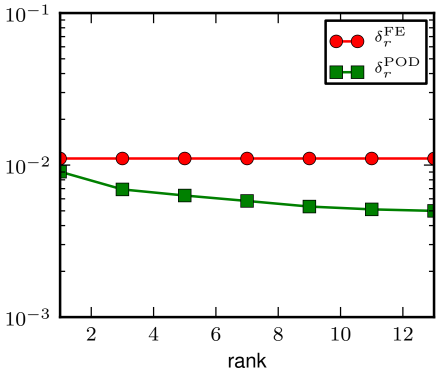

Figure 4.9.: Example 4.1: Stabilization parameters for the SUPG-ROMs.

the influence of such noisy modes.

The answer to the first question is given by Figures 4.7 and 4.8: Both the FE-SUPG-ROM and the POD-SUPG-ROM yield more accurate results than the G-ROM for $r \geq 2$. The stabilized ROMs can compute good solutions even if POD modes are used which are strongly influenced by spurious oscillations. Question 2 is answered in Figure 4.7: The FE-SUPG-ROM performs better than the POD-SUPG-ROM, which seems to be due to the larger stabilization parameters, see Figure 4.9. ◁

Example 4.2. *Traveling wave.* This example is similar to the one used in [51]. It is given in $\Omega = (0,1)^2$ and $(0,T) = (0,1)$ with the coefficients of (2.1) chosen as $\varepsilon \in \{10^{-4}, 10^{-8}\}$, $\boldsymbol{b} = (\cos(\pi/3), \sin(\pi/3))$, and $c = 1$. The analytical solution is defined by

$$u(t,x,y) = 0.5\,\sin(\pi x)\,\sin(\pi y)\left[\tanh\left(\frac{x+y-t-0.5}{\sqrt{\varepsilon}}\right)+1\right]. \tag{4.54}$$

The right-hand side f, the initial condition u_0, and the boundary condition were chosen such that (4.54) satisfies the boundary value problem. Solution (4.54) possesses a moving internal layer of width $\mathcal{O}\left(\sqrt{\varepsilon}\right)$.

First, the numerical results for the diffusion coefficient $\varepsilon = 10^{-4}$ will be investigated. The finite element problem using P_2 finite elements and the backward Euler scheme with a time step $\Delta t = 10^{-4}$ is solved on level 7 with the mesh width $h = 1.1 \cdot 10^{-2}$. Consequently, it holds $h \approx \mathcal{O}(\sqrt{\varepsilon})$, i.e., the internal layer is resolved and no stabilization is needed to obtain a satisfactory finite element solution. One can observe this behavior in Figure 4.10, where the Galerkin and SUPG finite element solutions at $t = 1.0$ are depicted. As the numerical analysis in Section 4.3 to derive the scaling of the SUPG-ROM parameter was carried out for snapshots computed with the SUPG finite element method, in addition to the Galerkin finite element snapshots also the SUPG snapshots will be used to build the ROMs.

For computing the POD basis, every tenth solution was stored such that 1001 snapshots were used. Note that Example 4.2 is more complex than Example 4.1 since the position of the layer depends on time. The complexity is reflected by the dimension of

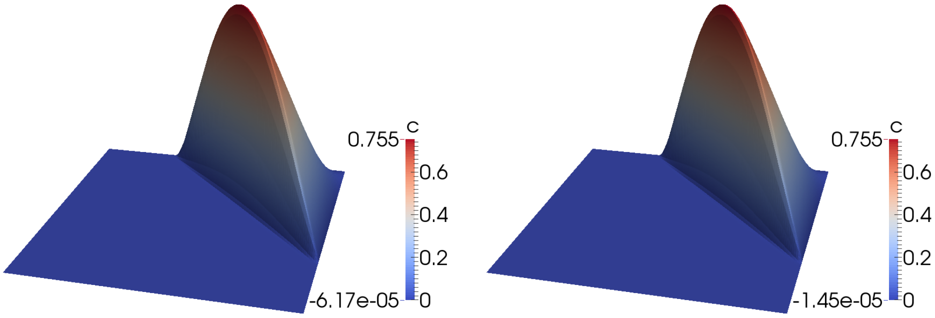

Figure 4.10.: Example 4.2 with $\varepsilon = 10^{-4}$: Galerkin (left) and SUPG (right) finite element solutions at $t = 1.0$.

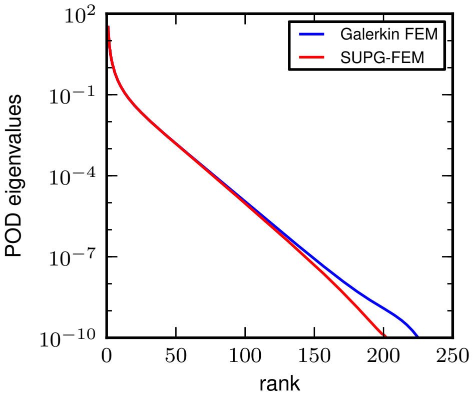

Figure 4.11.: Example 4.2 with $\varepsilon = 10^{-4}$: POD eigenvalues.

the snapshot space X_R, which is 14 for Example 4.1 and around 200 for Example 4.2 with $\varepsilon = 10^{-4}$ (see Figure 4.11 with the corresponding POD eigenvalues).

In Figure 4.12, the ROM errors in the discrete $L^1(0,T;L^2(\Omega))$ norm (left) and the SUPG-ROM parameters (right) are shown for the snapshots computed with the Galerkin finite element method. In Figure 4.13, the same quantities are depicted for the snapshots computed with the SUPG method. One can see that the difference between the both figures is very small. The answer to the first question from the beginning of the section is given by the right-hand side plots in Figures 4.12 and 4.13. Independently of the origin of the snapshots, the FE-SUPG-ROM and POD-SUPG-ROM yielded negligibly better results than G-ROM for smaller values of r. The G-ROM based on the snapshots computed with the Galerkin finite element method gives slightly more accurate results for larger r, starting at $r = 110$, than both SUPG-ROMs. All investigated ROMs built from the SUPG finite element snapshots yielded comparable errors in the discrete $L^1(0,T;L^2(\Omega))$ norm for larger values of r. Also the second question can be answered by Figures 4.12 and 4.13. Despite of different values of the SUPG-ROM parameters δ_r^{FE} and δ_r^{POD}, both SUPG-ROMs yielded almost the same results with negligibly small

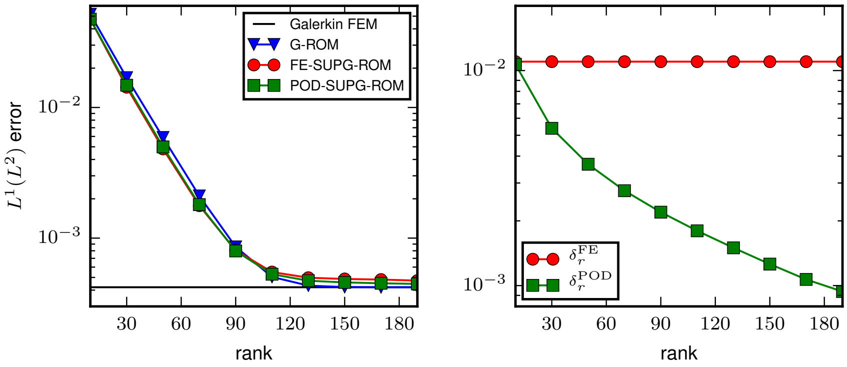

Figure 4.12.: Example 4.2 with $\varepsilon = 10^{-4}$: $L^1(0, T; L^2(\Omega))$ error for Galerkin finite element snapshots and for different ROMs.

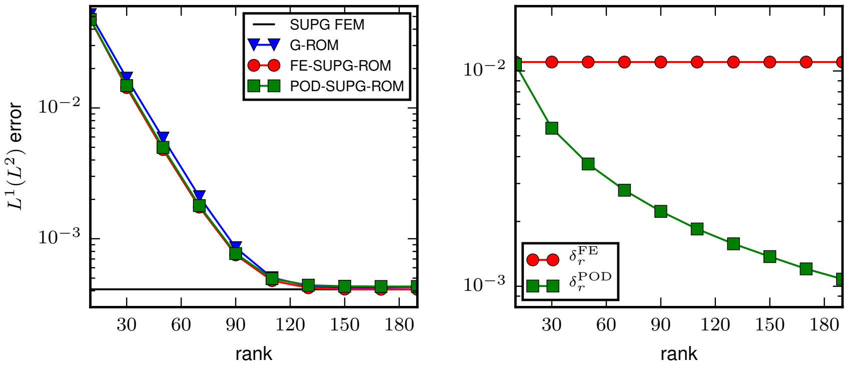

Figure 4.13.: Example 4.2 with $\varepsilon = 10^{-4}$: $L^1(0, T; L^2(\Omega))$ error for SUPG finite element snapshots and for different ROMs.

differences.

Now, the numerical results for the diffusion parameter $\varepsilon = 10^{-8}$ will be presented. It is a convection-dominated problem. By using realistic meshes with $h \in [10^{-3}, 10^{-2}]$, it is not possible to resolve the interval layer. Therefore, a stabilization method is needed to solve the finite element problem. To investigate the sensitivity of the numerical results with respect to the mesh width h, the finite element problem for computing the snapshots was solved on levels 6, 7, and 8. The backward Euler scheme was applied with a time step $\Delta t = 10^{-4}$. The SUPG stabilization was applied to the finite element method, just as in Example 4.1. In Figure 4.14, the Galerkin and SUPG finite element solutions evaluated at $t = 0.5$ and $t = 1.0$ are shown. One can see that, in contrast to the Galerkin method, the SUPG method is able to suppress the global oscillations. Every tenth solution was stored, resulting in 1001 snapshots, to compute the POD basis. Note that Example 4.2 with $\varepsilon = 10^{-8}$ is even more complex than Example 4.2 with $\varepsilon = 10^{-4}$ investigated before. Here, the snapshot space X_R has the dimension between 800 and 1000 depending on the

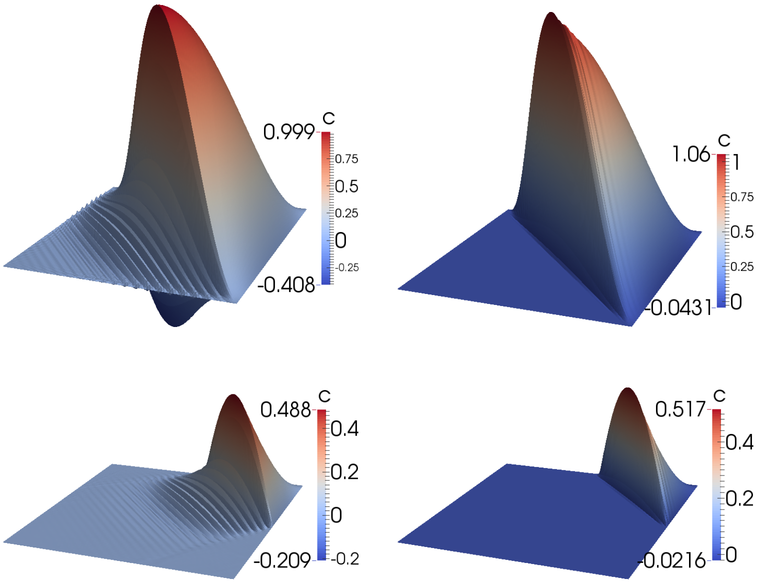

Figure 4.14.: Example 4.2 with $\varepsilon = 10^{-8}$ for level 7: Galerkin (left) and SUPG (right) finite element solutions at $t = 0.5$ (top) and $t = 1.0$ (bottom).

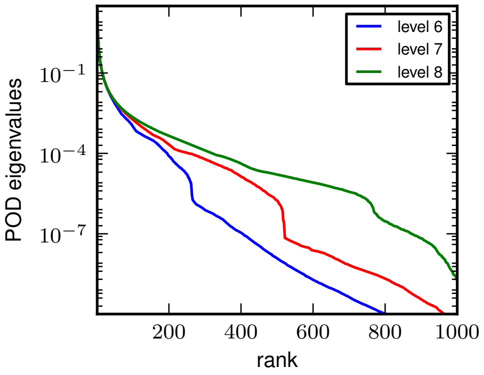

Figure 4.15.: Example 4.2 with $\varepsilon = 10^{-8}$: POD eigenvalues.

underlying spatial level (see Figure 4.15 with the corresponding POD eigenvalues). The plot in Figure 4.15 also shows that the snapshot space X_R changes with the mesh width h, its dimension increases with decreasing h. Like in Example 4.1, the studied ROMs use noisy POD data, which, as explained, is inevitable for convection-dominated problems on realistic meshes.

In Figures 4.16 – 4.18, the errors in the discrete $L^1(0, T; L^2(\Omega))$ norm are shown for the three considered ROMs and for spatial levels 6, 7, and 8. On every spatial level, a similar behavior as that in Example 4.1 is observed: Increasing the number of POD modes results eventually in an increase of the G-ROM error. This time, however, this

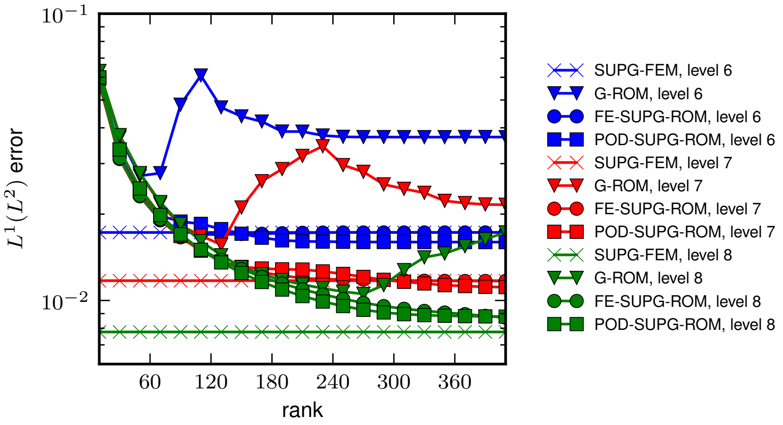

Figure 4.16.: Example 4.2 with $\varepsilon = 10^{-8}$: $L^1(0,T;L^2(\Omega))$ error for different ROMs and three different spatial levels.

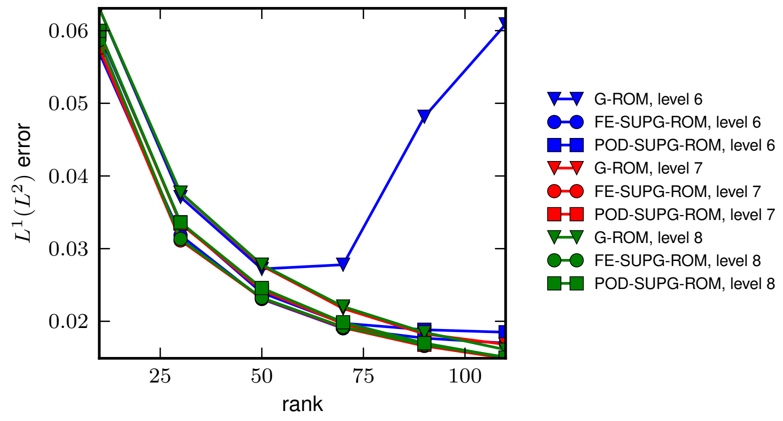

Figure 4.17.: Example 4.2 with $\varepsilon = 10^{-8}$: $L^1(0,T;L^2(\Omega))$ error for different ROMs, three different spatial levels, and small r.

error increase is observed later than in Example 4.1 (around $r = 70$ for level 6, $r = 130$ for level 7, and $r = 270$ for level 8). Like in Example 4.1, the G-ROM error grows due to the fact that the impact of noise becomes more and more dominant for modes with higher indices. Note that the finer the mesh, the higher the threshold rank is. Here, the threshold rank refers to the lowest POD dimension at which the G-ROM error starts to grow. Furthermore, below this threshold the discrete $L^1(0,T;L^2(\Omega))$ errors of the G-ROM and the SUPG-ROMs are similar.

The answer to the first question from the beginning of the section is given by Figures 4.16 – 4.19. For small numbers of POD modes, all ROMs have a similar error in the discrete $L^1(0,T;L^2(\Omega))$ norm and for large numbers of POD modes, both the FE-SUPG-ROM and POD-SUPG-ROM yield more accurate results than the G-ROM. With respect

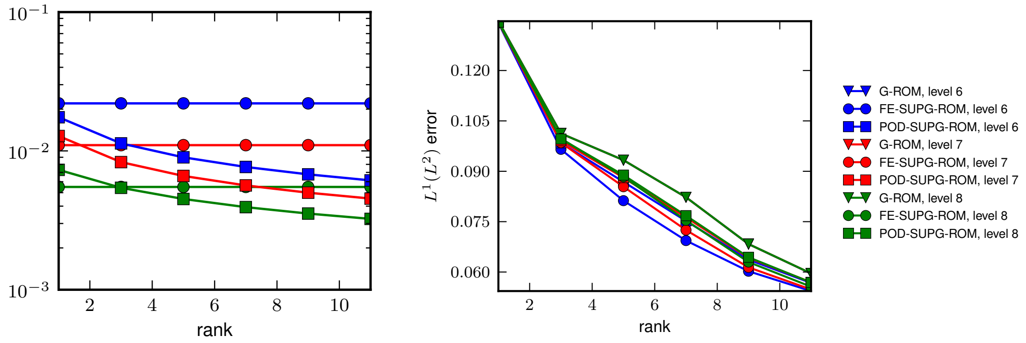

Figure 4.18.: Example 4.2 with $\varepsilon = 10^{-8}$: Different ROMs at three different spatial levels and for very small r: Stabilization parameters (left) and $L^1(0,T;L^2(\Omega))$ error (right). The G-ROM error curves for all spatial levels are the same.

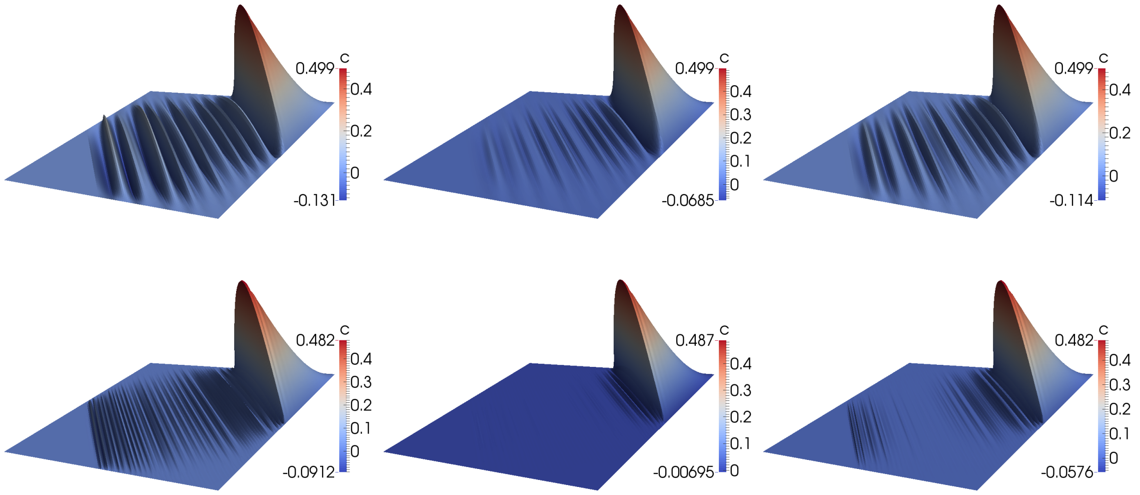

Figure 4.19.: Example 4.2 with $\varepsilon = 10^{-8}$ for level 7: Solution at $t = 1.0$ for G-ROM, FE-SUPG-ROM, POD-SUPG-ROM (from left to right) for $r = 30$, 130 (from top to bottom).

to the size of the spurious oscillations, the SUPG-ROMs are always better than the G-ROM. In Figures 4.18 and 4.20, the used stabilization parameters δ_r^{FE} and δ_r^{POD} for three spatial levels are shown. Even if the representation of δ_r^{POD} in (4.50) has no explicit dependence on the mesh width h, the stabilization parameter seems to depend implicitly on the underlying grid. This behavior is expected as Λ_0 and Λ_1 in (4.50) are computed using the POD modes and eigenvalues resulting from the finite element simulation. In particular, the value of δ_r^{POD} decreases for finer grids. The difference between the parameters δ_r^{FE} and δ_r^{POD} becomes more pronounced for large r. Question 2 can be studied with the results presented in Figures 4.16 – 4.19. Concerning the spurious oscillations, the results obtained with FE-SUPG-ROM are always better. With

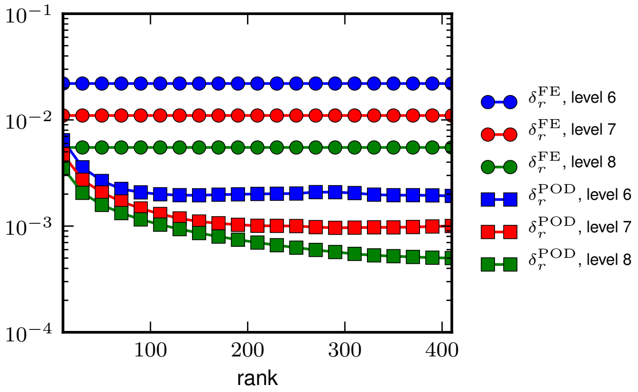

Figure 4.20.: Example 4.2 with $\varepsilon = 10^{-8}$: Stabilization parameters for the SUPG-ROMs.

respect to the error in the discrete $L^1(0,T;L^2(\Omega))$ norm, POD-SUPG-ROM performs slightly better than FE-SUPG-ROM for larger values of r. The explanation of both observations comes again from the different sizes of the stabilization parameters, see Figure 4.20. Since δ_r^{FE} is larger than δ_r^{POD}, the spurious oscillations in the FE-SUPG-ROM are damped more efficiently than in the POD-SUPG-ROM. To investigate why the POD-SUPG-ROM for larger r yielded slightly more accurate results than the SUPG finite element method in Figure 4.16, additional SUPG finite element simulations with $\delta_h = \delta_r^{\mathrm{POD}} = 0.2h$ were carried out. The resulting discrete $L^1(0,T;L^2(\Omega))$ error was comparable to that of the POD-SUPG-ROM for larger r in Figure 4.16. The plots in Figures 4.16 – 4.18 also show that as the mesh width h changes, the ROM results change as well, but the qualitative behavior of the ROMs (and in particular the answers to the first two questions from the beginning of the section) remains unchanged. The plots in Figure 4.18, however, show that for very low r as the mesh width decreases the difference between the stabilization parameters and the difference between the errors of the two SUPG-ROMs decrease as well. ◁

Example 4.3. *Rotating body problem.* This example describes one full counter-clock rotation of three disjoint bodies. It is an adaption of the transport problem from [99] and was extensively studied, e.g., in [91, 94] for transport problems. In [82, 84], the test case was used to study discretizations with a very small diffusion coefficient.

Consider (2.1) in $\Omega = (0,1)^2$ and $(0,T) = (0,6.28)$ with the coefficients $\varepsilon = 10^{-20}$, $\boldsymbol{b} = (0.5-y, x-0.5)^T$, and $c = f = 0$. The initial condition is represented by three disjoint bodies (slotted cylinder, conical body and hump) as shown in Figure 4.21. Each body has its center (x_0, y_0) and is placed within the radius $r_0 = 0.15$. The initial condition is zero outside the bodies. Let

$$r(x,y) = \frac{1}{r_0}\sqrt{(x-x_0)^2 + (y-y_0)^2}.$$

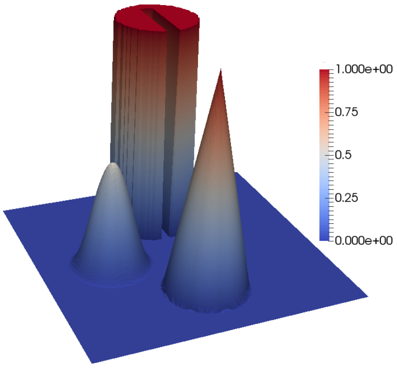

Figure 4.21.: Example 4.3: Initial condition.

The slotted cylinder is described by

$$u(0,x,y) = \begin{cases} 1, & \text{if } r(x,y) \leq 1,\ \|x - x_0\| \geq 0.0225 \text{ or } y \geq 0.85, \\ 0, & \text{elsewhere,} \end{cases}$$

with the center $(x_0, y_0) = (0.5, 0.75)$. The conical body is given by

$$u(0,x,y) = 1 - r(x,y),$$

and the center $(x_0, y_0) = (0.5, 0.25)$. Finally, the hump is described by $(x_0, y_0) = (0.25, 0.5)$ and

$$u(0,x,y) = \frac{1}{4}\left(1 + \cos(\pi \min\{r(x,y), 1\})\right).$$

The original version of the example from [99] is a pure transport problem with $\varepsilon = 0$. In this case, an ideal numerical method should yield the initial condition as the solution after each revolution of the three bodies. With such a small diffusion coefficient ε as used in this study, one can expect that an ideal numerical method nearly recovers the initial solution at $t = 6.28 \approx 2\pi$.

For the underlying numerical studies, three sets of snapshots were considered. The first set of snapshots, which will be denoted by SUPG-FEM, was obtained by approximating the solution of (2.1) with the SUPG finite element method, see Section 2.1.3, using the backward Euler scheme for the time discretization (as it was done in Examples 4.1 and 4.2). To create the second set of snapshots, which will be denoted by FEM-FCT, the nonlinear flux-corrected transport scheme presented in Section 2.1.5 was used with the Crank–Nicolson method as time integrator, e.g., see [84]. For both finite element methods, the piecewise linear finite elements P_1 and the length of the time step $\Delta t = 10^{-3}$ were used. The computations were carried out on the spacial level 7, such that $h = 1.1 \cdot 10^{-2}$. The third set of snapshots, which will be denoted by Reference, was obtained by interpolating the solution of the continuous problem (2.1) using the same finite element space as for other two snapshot sets. Note that the snapshot sets can

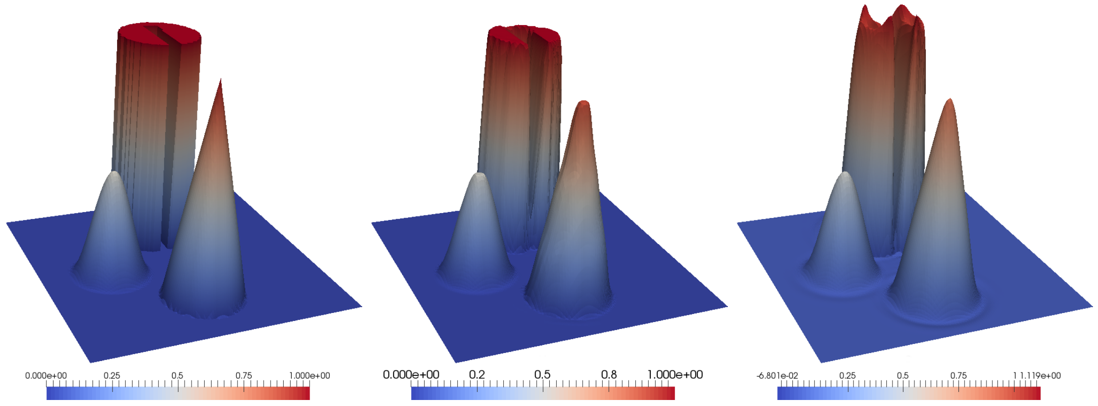

Figure 4.22.: Example 4.3: Solution at $t = 6.28$ for Reference (left), the FEM-FCT scheme (center), and the SUPG-FEM (right).

be divided into two types: Representing the solution of a discrete problem (FEM-FCT, SUPG-FEM) and the projection of the continuous solution onto the finite element space (Reference).

In Fig. 4.22, the solution for all three approaches is shown for the final time $t = 6.28$. One can see that the interpolated continuous solution (on the left-hand side) represents the optimal solution as it looks the same as the initial condition in Fig. 4.21. The FEM-FCT solution (in the center) looks less accurate due to the slightly smoothed corners but still better than the SUPG-FEM solution (on the right-hand side) which exhibits some oscillations at the slotted cylinder. In Fig. 4.23, the time evolution of the L^2 error, and the minimum and the maximum values for these three approaches are displayed. The interpolated continuous solution has the best performance in terms of the L^2 error and features no under- or overshoots, i.e., the minimum and the maximum values of the solution are equal to 0 and 1, respectively. The FEM-FCT method produces a greater L^2 error but due to the design of the numerical method, see Section 2.1.5, the solution has no under- or overshoots. Finally, the SUPG-FEM solution performs worse than the other two approaches in terms of all three measures of interest.

Each set of snapshots consists of 1257 elements, which represent every fifth solution of the finite element methods, i.e., the solution at times $t = 5n \cdot 10^{-3}$, $n = 0, \ldots, 1256$, and the evaluation of the interpolated solution at the same times. Three POD bases were computed from the fluctuating part of the snapshot sets SUPG-FEM, FEM-FCT, and Reference with respect to the L^2 inner product by the method of snapshots, see Section 3.1.3. The distribution of the POD eigenvalues and the missing energy ratio computed by $1 - \mathcal{E}(r)$, $r \leq R$, where $\mathcal{E}(r)$ is defined by (3.52), are shown in Fig. 4.24. From the plot, one immediately takes note of the fact that the POD eigenvalues based on the SUPG-FEM snapshot set fall much more rapidly than the eigenvalues based on Reference and FEM-FCT. By taking into account only the POD eigenvalues with the lower threshold of 10^{-10}, the dimension R of the snapshot space X_R for SUPG-FEM is only 119 and for the other two sets it is 1256. In the case of Reference and FEM-FCT,

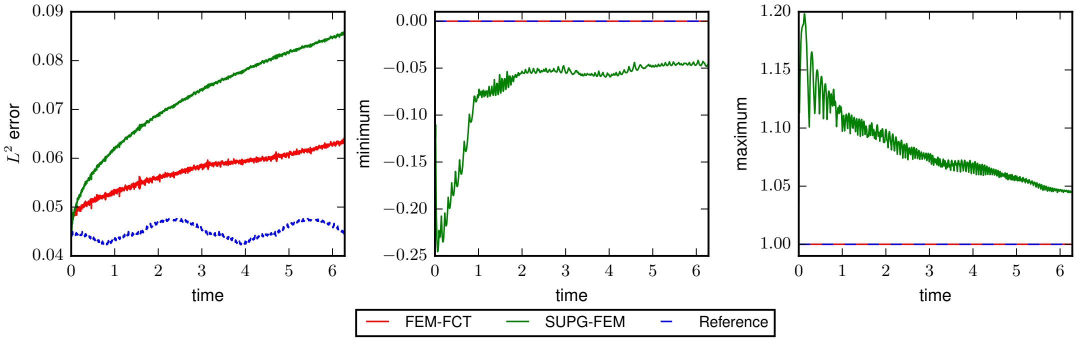

Figure 4.23.: Example 4.3: Time evolution of the measures of interest for the finite element methods used to compute the snapshots.

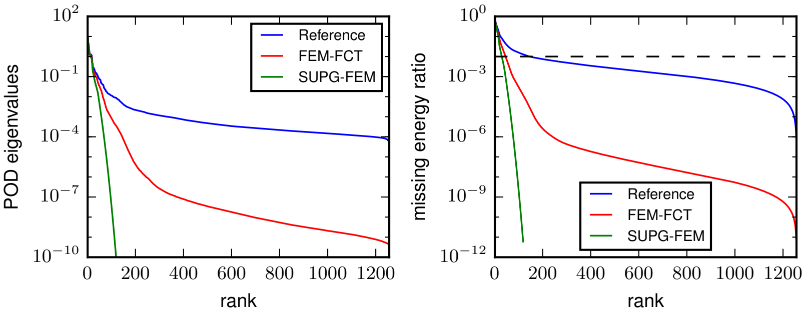

Figure 4.24.: Example 4.3: POD eigenvalues (left) and the missing energy ratio $1 - \mathcal{E}(r)$ (right), see (3.52), for the finite element methods used to compute snapshots. The dashed line on the right-hand side indicates the missing energy ratio of 0.01.

no single snapshot can be represented by a linear combination of the other snapshots because of the moving inner layers. The steeper descrease of the POD eigenvalues based on FEM-FCT can be explained by the smearing of the FEM-FCT solution at the inner layers. The possible reason for the low dimension of the snapshot space X_R arising from SUPG-FEM lies in the fact that the SUPG-FEM solution features some spurious oscillation around the inner layers. This not quite local effect might lead to the linear dependency of the snapshots. In Fig. 4.25, the snapshot mean $\bar{u}_h$ and the first three POD modes $\varphi_{\mathrm{ro},i}$, $i = 1, 2, 3$, are depicted based on all three snapshot sets. No essential differences besides of slightly different value ranges and minor shifts of the forms can be asserted.

In terms of reduced-order modeling, simulations with G-ROMs and SUPG-ROMs were run with the backward Euler scheme, see (4.2), using the length of the time step $\Delta t = 10^{-3}$. To distinguish the ROMs with respect to different snapshot sets, the notations G-ROM($*$) and SUPG-ROM($*$) will be utilized, where $*$ can be Ref (short version of Reference), FCT (short version of FEM-FCT), or SUPG (short version of SUPG-FEM).

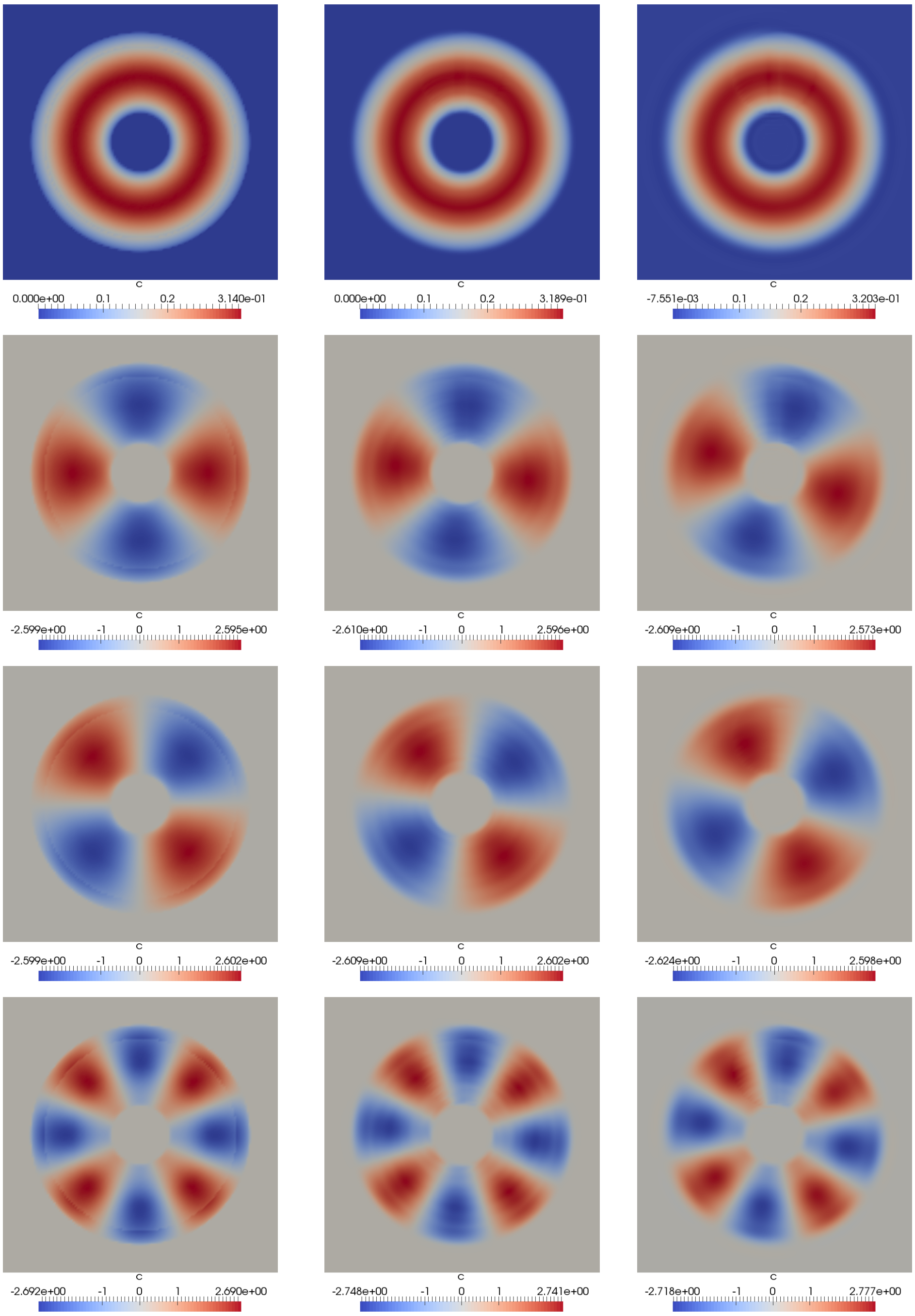

Figure 4.25.: Example 4.3: Snapshot mean $\bar{u}_h$ and POD modes $\varphi_{\mathrm{ro},1}$, $\varphi_{\mathrm{ro},2}$, $\varphi_{\mathrm{ro},3}$ (from top to bottom) based on the Reference, FEM-FCT, and SUPG-FEM snapshot sets (left to right).

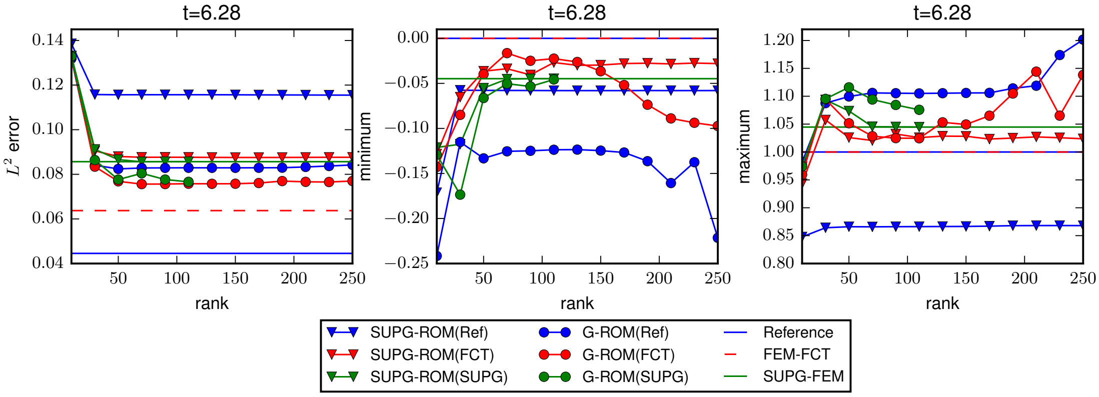

Figure 4.26.: Example 4.3: Measures of interest obtained with G-ROMs and SUPG-ROMs based on three different snapshot sets at the final time $t = 6.28$.

Unless otherwise stated, all ROMs will be equipped with the initial condition (3.62).

The L^2 errors, the minimum and the maximum values for the G-ROMs and SUPG-ROMs at the final time $t = 6.28$ for different dimensions of the POD space and the time evolution of all three measures of interest for $r \in \{50, 110\}$ are depicted in Figs. 4.26 and 4.27, respectively. These figures contain the answer to the first question from the beginning of the section. In terms of the L^2 error, the G-ROMs performed in general better than the corresponding SUPG-ROMs based on all three sets of snapshots. It is noticeable that the SUPG-ROM based on the most accurate snapshot set Reference produces a considerably larger L^2 error than the other ROMs. The situation is different with respect to the minimum and the maximum values. The SUPG-ROMs based on the snapshot sets representing the numerical solution of a discrete problem, i.e., FEM-FCT and SUPG-FEM, yielded in general better minimum and maximum values than the associated G-ROMs. At the first glance, the plots in Fig. 4.26 reveal that the G-ROMs based on SUPG-FEM and FEM-FCT snapshots result in similar or even more accurate minimum and maximum values for some POD dimensions r compared to the SUPG-ROMs. If one takes a closer look at the time evolution of these values in Fig. 4.27, one can see that the SUPG-ROMs produce more reliable results, as the minimum and maximum values of their solutions do not fluctuate over time as much compared to the G-ROMs. Both the G-ROM and the SUPG-ROM based on on the snapshot set Reference, which does not arise from the numerical solution of a discrete problem but from the projection of the continuous problem on the finite element space, did not result in satisfactory minimum and maximum values. In particular, the SUPG-ROM(Ref) features a strong decay of the maximum value in the course of time.

The reduced-order approximation of the initial condition $u_{\text{ro}}^0 = \bar{u}_h + \sum_{i=1}^r \alpha_i^0 \varphi_{\text{ro},i}$ with $\underline{\alpha}^0$ computed by (3.62), which was used for the above ROM computations, is shown in Fig. 4.28 in the top row. It is noteworthy that the ROM initial condition for all snapshot sets exhibits some oscillations not only at the bodies themselves but also in the smooth region along the trajectory of the rotation. The initial condition based on

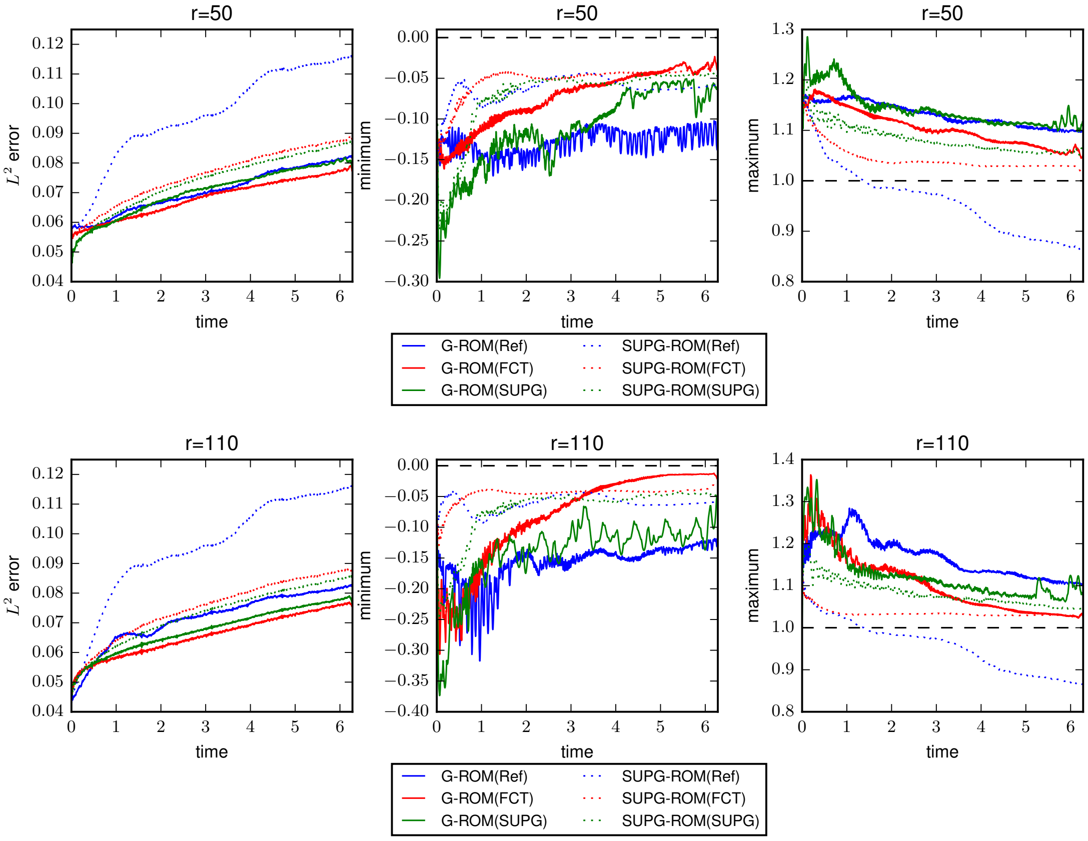

Figure 4.27.: Example 4.3: Time evolution of the measures of interest for G-ROMs and SUPG-ROMs based on three different snapshot sets for $r = 50$ (top) and $r = 110$ (bottom).

the interpolated exact solution is particularly strongly polluted in the smooth region. A good quality initial condition is of great importance for the numerical simulations employing implicit time discretization methods as it directly influences the accuracy of the solution at the subsequent time steps. In order to obtain a smoother initial condition, the post-processing filtering procedure (3.65) proposed in Section 3.2.2 was applied with the filter width $\mu = h$ to the initial condition given by (3.62). The plots in the bottom row of Fig. 4.28 present the filtered ROM initial condition. It is clearly visible that the filtered initial condition features much less spurious oscillations than the standard one.

Figures 4.29 and 4.30 show the L^2 errors, the minimum and the maximum values at the final time for different POD dimensions and their time evolution for $r \in \{50, 110\}$, respectively, for the G-ROMs and the SUPG-ROMs combined with the filtered initial condition. At this point, the third question from the beginning of this section will be addressed. According to the plots, the L^2 errors are slightly larger for all ROMs with the filtered initial condition than for the corresponding ROMs with the conventional initial

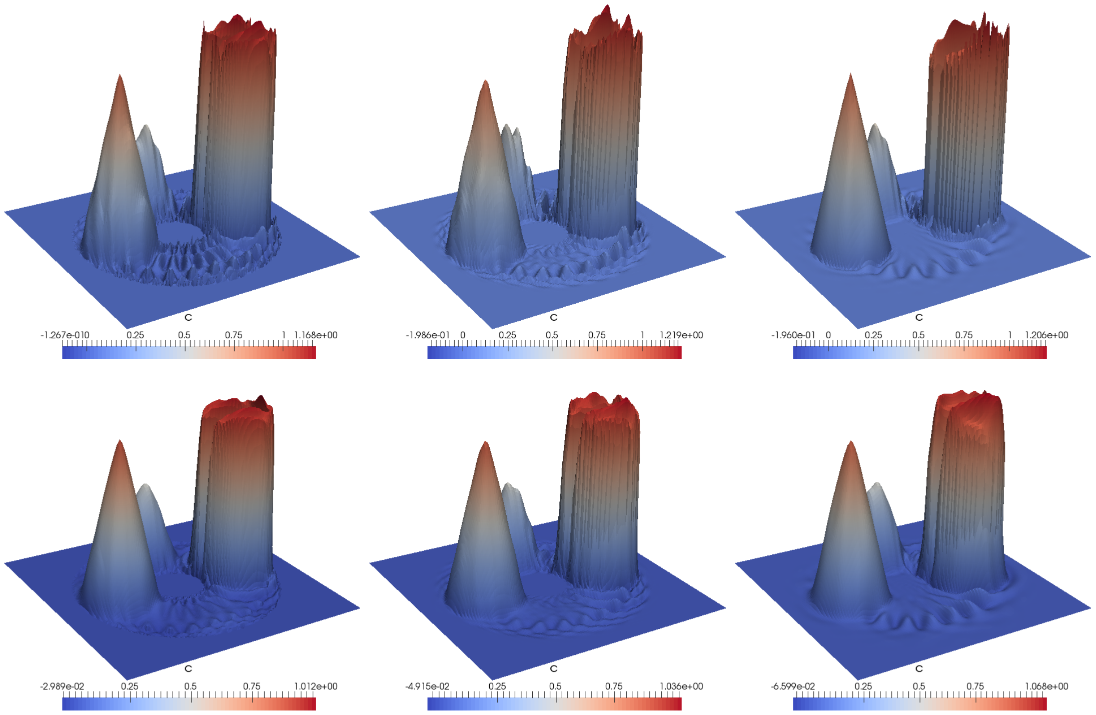

Figure 4.28.: Example 4.3: ROM initial condition given by (3.62) (top) and additionally filtered by (3.65) (bottom) based on the snapshots Reference (left), FEM-FCT (center), and SUPG-FEM (right) for $r = 50$.

condition. Despite of the smoother initial condition, the SUPG-ROM(Ref) resulted in by far the largest L^2 error as it was already observed in Figs. 4.26 and 4.27. Also the minimum and the maximum values of the SUPG-ROM(Ref) solution did not improve with the smoother initial condition. The maximum value became even worse. This observation supports the presumption that SUPG-ROM is not suitable for this kind of snapshots. In contrast, the filtering procedure of the initial condition has a positive effect on the minimum values of the SUPG-ROMs based on both the FEM-FCT and the SUPG-FEM snapshots. A close monitoring of the maximum values of the SUPG-ROM solutions in Figs. Figures 4.29 and 4.30 reveals a smaller distance to the optimal value for the SUPG-FEM snapshots and more or less the same distance for the FEM-FCT snapshots. However, a slightly dissipative behavioral pattern can be observed as the values are shifted below the optimum. Furthermore, a significant improvement of the minimum and the maximum values of the solution resulted from the G-ROMs based on the interpolated exact solution for $r \in [50, 170]$ and on the FEM-FCT snapshots for $r \in [50, 190]$. Regarding the first question, G-ROM(FCT) behaves in the lower part of the indicated range of the POD dimensions even somewhat better than the corresponding SUPG-ROM. The solution of G-ROM(Ref) is notably superior to SUPG-ROM(Ref) in the range $r \in [50, 170]$, where it features almost the optimal maximum

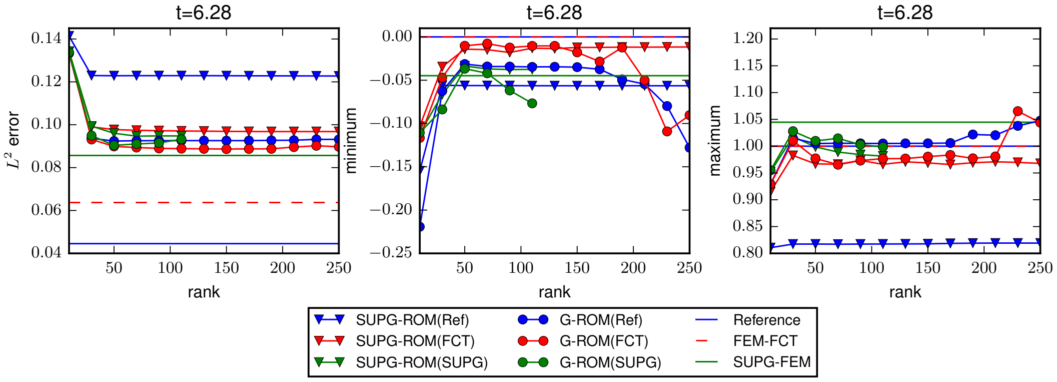

Figure 4.29.: Example 4.3: Measures of interest obtained with G-ROMs and SUPG-ROMs based on three different snapshot sets combined with the filtered initial condition.

value. The performance of the G-ROM based on the SUPG-FEM snapshots could also be improved by smoothing the initial condition. However, its solution still exhibits fluctuations of the minimum and the maximum values over time especially for higher POD dimensions. Thus, SUPG-ROM(SUPG) performed better than the corresponding G-ROM independently of the utilized initial condition.

Summing up all the presented findings, the fourth question posed at the beginning of this section will be responded. For the snapshot sets representing the numerical solution of a discrete problem, i.e., FEM-FCT and SUPG-FEM, SUPG-ROM combined with the standard initial condition (3.62) yields better results when it is based on the more accurate snapshots FEM-FCT. From Fig. 4.26, one can assert that SUPG-ROM based on the most inaccurate set of snapshots SUPG-FEM achieves the accuracy of the underlying finite element simulation for $r \geq 70$. SUPG-ROM based on the FEM-FCT snapshots never reaches the optimal minimum and maximum values 0 and 1, respectively, which were obtained by the corresponding finite element simulation. However, it yields closer values to the optimal ones compared to SUPG-ROM(SUPG). With the filtered initial condition, SUPG-ROM(FCT) does better than SUPG-ROM(SUPG) in terms of the minimum value, but the opposite behavior can be observed for the maximum value. SUPG-ROM seems to be an inappropriate model for the POD basis computed from the snapshots describing the interpolation of the continuous solution. The statements refer to both the unfiltered and the filtered initial condition. There is no obvious relation between the accuracy of the underlying snapshots and the accuracy of the G-ROM solutions.

For the most ROMs considered in the underlying investigation, it is sufficient to choose the POD dimension in the range of $[50, 70]$ to achieve the best possible results.

◁

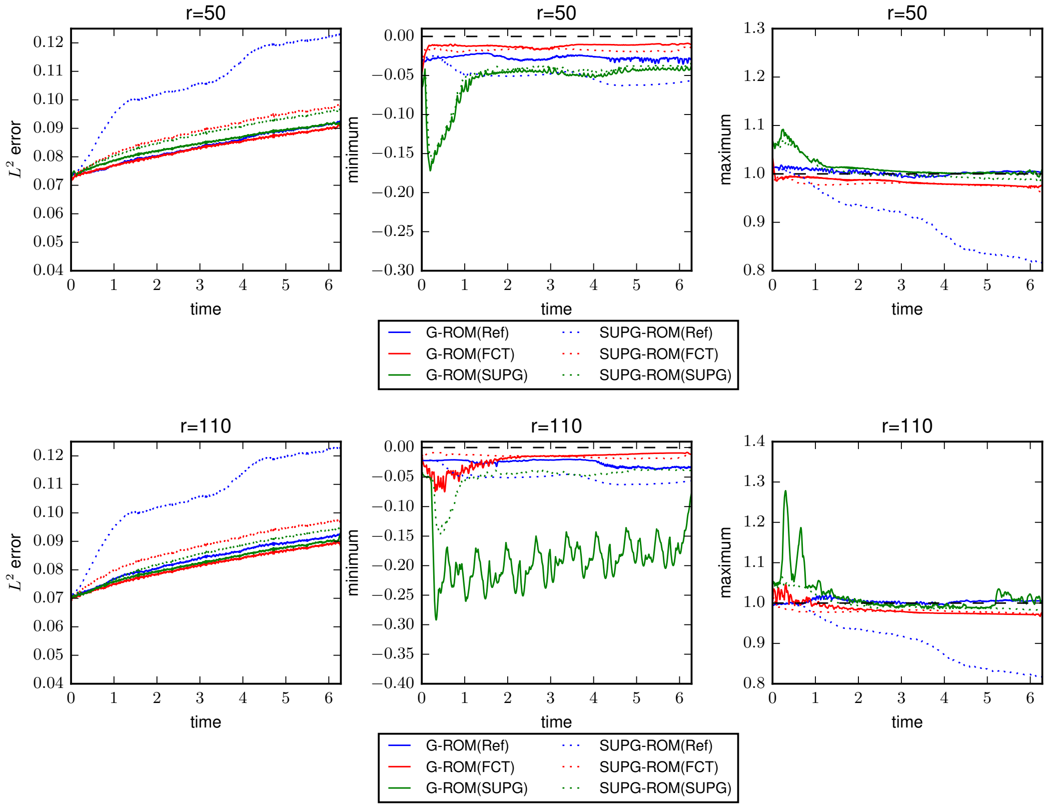

Figure 4.30.: Example 4.3: Time evolution of the measures of interest for G-ROMs and SUPG-ROMs based on three different snapshot sets for $r = 50$ (top) and $r = 110$ (bottom) combined with the by (3.65) filtered initial condition.

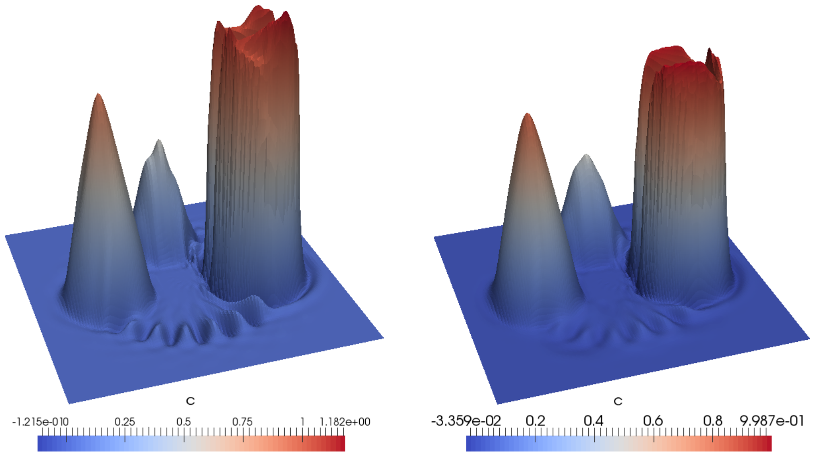

Figure 4.31.: Example 4.3: SUPG-ROM(SUPG) solution for $r = 50$ at $t = 6.28$ combined with the initial condition given by (3.62) (left) and with the initial condition additionally filtered by (3.65) (right).

5. Velocity-Pressure Reduced-Order Models for Incompressible Flows

This chapter has three main goals. First, it numerically investigates three different types of velocity-pressure reduced-order models (ROMs) for incompressible flows. The Proper Orthogonal Decomposition (POD) presented in Section 3.1 is used to generate the bases for the ROMs. One method computes the ROM pressure solely based on the velocity POD modes, whereas the other two ROMs use pressure POD modes as well. One of the velocity-pressure ROMs is new and was first introduced in [26]. Two further goals are to numerically investigate the impact of the snapshot accuracy and of the employed nonlinear iteration schemes on the results of the ROMs. A substantial part of the chapter can be found in [26].

5.1. Introduction

By means of the numerical methods for solving the Navier–Stokes equations (2.40)-(2.41) like, e.g., finite elements methods, one can compute more and more details of the flow field by increasing the dimension of the finite element spaces. However, increasing the number of basis functions yields large linear or nonlinear systems to be solved in the simulations. Consequently, the numerical simulation of the flow can be very time-consuming. It is especially the case, when similar problems need to be solved multiple times. In addition, the finite element basis is generally defined independently of the solution, and it only depends on the structure of the computational mesh. In the case that a priori information on the solution is available, one could transfer this knowledge to the finite element space by pre-adapting the triangulation of Ω.

Reduced-order models aim at reducing the computational cost of full finite element, finite difference or finite volume simulations by drastically reducing the dimension of the solution space. The key idea of ROMs consists in utilizing basis functions that already represent the most important features of the solution. In contrast to finite element bases, ROM bases are global bases. The focus in this chapter will be on the ROMs, in which the basis functions are obtained through a POD (see Section 3.1). Reduced-order modeling for incompressible flows based on the POD is meanwhile widely used and is an active field of research, see, e.g., [5, 12, 13, 27, 130, 144] for recent publications. Here, the snapshots will be obtained from detailed finite element simulations. It is worth noticing that generally the snapshots might even come from experimental data [9, 63].

Within the framework of this chapter, three main goals will be pursued. One of the goals consists in investigating three different types of ROMs that compute, besides the velocity, also the pressure, called here for shortness vp-ROMs. One of these ROMs was developed within the framework of this thesis and published in [26]. Furthermore, it will be studied how the choice of three nonlinear iteration schemes of different complexity affects the accuracy of the ROMs. Finally, this chapter investigates the impact of the accuracy of the simulations for computing the snapshots, shortly denoted by snapshot accuracy, on the vp-ROM results. The motivation and background for these numerical investigations are presented in the following.

To motivate the use of vp-ROMs, it must be noted that although most ROMs for incompressible flows do not include a pressure component, there are important settings in which vp-ROMs are appropriate. From the practical point of view, the pressure is needed in many computational fluid dynamics applications, e.g., for the simulation of fluid-structure interaction problems and for the computation of relevant quantities such as drag and lift coefficients on solid bodies, and for ROM simulations of shear flows [108]. Other reasons for including the pressure are connected to the definition of ROMs. Using only a velocity ROM leads to a comparatively simple model that can be simulated very efficiently. The rationale behind velocity ROMs, as it can be found in the literature, is that all snapshots are divergence-free, hence all basis functions are divergence-free and consequently the ROM velocity is divergence-free, such that the pressure (which acts as a Lagrange multiplier of the divergence-free constraint) is not needed. As it will be clarified in Section 5.3, the same rationale can be applied in the context of finite element methods and discretely divergence-free velocity fields. In this case, only the integrals of the product of the velocity divergence and all test functions from the discrete pressure space vanish. In fact, many numerical methods for computing the snapshots do not provide pointwise divergence-free flow fields. Even for finite element methods, the discretely divergence-free property does not hold for many popular discretizations of the Navier–Stokes equations. Such examples include the case of using the same finite element spaces for velocity and pressure, where a numerical stabilization becomes necessary, or pressure-correction schemes without reconstructing the discretely divergence-free solution. Experimental data will generally not be divergence-free as well. Altogether, the violation of the divergence-free constraint on the snapshots is another reason for incorporating the pressure into ROMs for incompressible flow simulations. Moreover, as already pointed out in [19], the availability of the pressure enables the computation of the residual of the strong form of the Navier–Stokes equations (2.40)-(2.41). Strong residuals are often needed in stabilized discretizations, e.g., for stabilization with respect to the violation of the inf-sup condition or with respect to dominating convection.

In the literature, one can find different proposals for an approximation of the pressure field or its inclusion into the ROM. One class of vp-ROMs consists in defining a ROM for the pressure that only uses the velocity POD modes [108, 138]. One of the methods is based on the continuous in space pressure Poisson equation (see Section 2.2.2) and the other one is based on the discretized version of the pressure Poisson equation (see [50] for a detailed discussion). The former pressure ROM, denoted by VMB-ROM, will be studied in the numerical studies in Section 5.5. A second class of vp-ROMs employs pressure POD basis functions in addition to the velocity POD basis functions, e.g., see [2, 26, 136]. The pressure POD basis functions can be computed separately from the velocity POD basis functions (i.e., the decoupled approach) [108], or together with them (i.e., the coupled, monolithic approach) [19,145]. In this study, the decoupled approach will be utilized.

Two vp-ROMs that employ a pressure POD basis will be investigated in this chapter. The first vp-ROM in this class, here denoted by PMB-ROM, is based on the approach proposed in [2]. The second vp-ROM, called SM-ROM, was introduced in [26]. SM-ROM uses a residual-based stabilization mechanism for the incompressible Navier–Stokes equations. It is based on a mathematically well understood method [21]. The advantage over the two other vp-ROMs consists in the fact that its derivation requires the snapshots to be only discretely divergence-free (but not pointwise), and it does not need any ad hoc treatment of external forces and pressure boundary conditions. Overall, three vp-ROMs will be considered in the numerical studies. VMB-ROM and PMB-ROM solve the same equation for the pressure but in different finite-dimensional spaces. PMB-ROM and SM-ROM work in the same space, but in these methods different equations for the pressure are solved. All vp-ROMs can be considered as a post-processing step to a velocity ROM.

In order to exploit their computational efficiency, ROMs are often used in combination with simple, and therefore potentially inaccurate, numerical methods. Indeed, ROMs generally avoid the solution of nonlinear systems and use, where possible, explicit time integration schemes, see, e.g., [130,143]. The second goal of the chapter is to investigate how strong the actual effect of different linearization techniques, known from the context of the finite element methods, is on the ROM results with respect to different quantities of interest.

The third main goal of this chapter is to investigate the impact of the accuracy of the snapshots, and therefore of the resulting POD basis, on the numerical results of the vp-ROMs. The generation of the snapshots might be time consuming. Considering, e.g., a turbulent flow, then one can perform a direct numerical simulation (DNS), if the Reynolds number is sufficiently small for this approach to be feasible, or one can apply more or less advanced turbulence models on more or less fine meshes for this purpose. All approaches (should) give reasonable approximations of the large and important flow structures. The main differences will be in the resolved details of the flow. However, the DNS has to be performed on a very fine mesh and its computing time is usually orders of magnitude higher than of a simulation with a turbulence model on a coarser grid. And even simulations with a simple turbulence model, such as the Smagorinsky model, might be much faster than simulations with an advanced model such as a variational multiscale method. Since ROMs aim to compute only the most important features of the solution, and since ROMs generally utilize computationally efficient numerical approaches, the following question naturally emerges: “How strong are the impacts of the snapshot accuracy, on the one hand, and of the (simple) numerical method used in the ROMs, on the other hand, onto the ROM results?” In this chapter, a first step in numerically investigating this question will be performed. To the best of the author’s knowledge, the first study of this topic was conducted in [26].

To construct snapshot data of different accuracies, two approaches can be considered. The first approach uses the same numerical method, but different discretization parameters, e.g., different mesh sizes and/or different time steps. The second approach uses the same discretization parameters, but different numerical methods. In this study, the second approach will be utilized.

The chapter is structured as follows: In Section 5.2, the derivation of the coupled Galerkin ROM for the Navier–Stokes equations will be carried out. The ROM for the computation of solely the velocity field, including its fully discretized formulation and the implementation aspects, will be presented in Section 5.3. Thereafter, the derivation of three different reduced-order models for the computation of the ROM pressure will be conducted in Section 5.4. Furthermore, the advantages and disadvantages of the respective methods will be discussed. Finally, detailed numerical studies on a two-dimensional test problem describing the flow around a cylinder will be presented.

5.2. Galerkin ROM for Navier–Stokes Equations

In this section the Galerkin reduced-order model for the Navier–Stokes equations (2.40)-(2.41) will be derived. It will be assumed that the Navier–Stokes equations are equipped with the boundary conditions required for the numerical studies in Section 5.5 of the form

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{0} && \text{on} \quad (0,T] \times \Gamma_0, \\ \boldsymbol{u} &= \boldsymbol{g}_{\mathrm{D}} && \text{on} \quad (0,T] \times \Gamma_{\mathrm{in}}, \\ (\nu \nabla \boldsymbol{u} - p\mathbb{I})\,\boldsymbol{n} &= \boldsymbol{0} && \text{on} \quad (0,T] \times \Gamma_{\mathrm{out}}, \end{aligned} \tag{5.1}$$

where Γ_0, Γ_{in}, and Γ_{out} are mutually disjoint parts of the boundary Γ with $\Gamma = \Gamma_0 \cup \Gamma_{\mathrm{in}} \cup \Gamma_{\mathrm{out}}$ and the Dirichlet boundary $\Gamma_{\mathrm{D}} = \Gamma_0 \cup \Gamma_{\mathrm{in}}$, see Section 2.2.1. Notice that, in general, $\boldsymbol{g}_{\mathrm{D}}$ might depend also on time.

Following the guidelines for the construction of the Galerkin reduced-order model presented in Section 3.2.1, the starting point is the weak formulation of the Navier–Stokes equations (see Section 2.2.6). Let the velocity space V and the pressure space Q be defined by (2.83). The time-continuous weak formulation of the Navier–Stokes equations reads: Find $\boldsymbol{u} : (0,T] \to H^1(\Omega)$, such that $\boldsymbol{u}(t,\cdot) - \boldsymbol{u}_{\boldsymbol{g}}(t,\cdot) \in V$ for all $t \in (0,T]$, and $p : (0,T] \to Q$ such that

$$\begin{array}{rcll} (\partial_t \boldsymbol{u}, \boldsymbol{v}) + (\nu \nabla \boldsymbol{u}, \nabla \boldsymbol{v}) + ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u}, \boldsymbol{v}) - (\nabla \cdot \boldsymbol{v}, p) & = & (\boldsymbol{f}, \boldsymbol{v}), & \forall\, \boldsymbol{v} \in V, \\ -(\nabla \cdot \boldsymbol{u}, q) & = & 0, & \forall\, q \in Q, \end{array} \tag{5.2}$$

where $\boldsymbol{u}_{\boldsymbol{g}}(t,\cdot) \in H^1(\Omega)$ is an extension of the Dirichlet boundary condition into Ω for $t \in (0,T]$.

Furthermore, the finite-dimensional POD bases for the velocity and the pressure have to be computed by one of the algorithms shown in Sections 3.1.3 and 3.1.4. Let

$$\{\boldsymbol{\phi}_{\mathrm{ro},1}, \dots \boldsymbol{\phi}_{\mathrm{ro},r_v}\} \tag{5.3}$$

denote the velocity POD basis of dimension r_v and

$$\{\psi_{\mathrm{ro},1}, \dots \psi_{\mathrm{ro},r_p}\} \tag{5.4}$$

the pressure POD basis of dimension r_p. Let $\{\boldsymbol{u}_m\}_{m=1}^{M}$ and $\{p_m\}_{m=1}^{M}$ denote sets of snapshots of the finite element velocity and pressure solutions, respectively.

As the velocity Dirichlet boundary condition on Γ_{D} is not homogeneous, one of the approaches from Section 3.2.3 has to be applied to treat the boundary condition in the ROM correctly. Let the velocity and the pressure POD bases be computed from the modified snapshots $\{\boldsymbol{u}_m - \boldsymbol{u}_{\boldsymbol{g}}(t_m,\cdot)\}_{m=1}^{M}$ and $\{p_m - \overline{p}_h\}_{m=1}^{M}$, respectively. For possible definitions of $\boldsymbol{u}_{\boldsymbol{g}}(t_m,\cdot)$ the reader is referred to Section 3.2.3; $\overline{p}_h$ stands for the average of the pressure snapshots p_m, $m=1,\ldots,M$. Note that theoretically the pressure snapshots do not need to be modified as for the pressure no Dirichlet boundary is imposed. However, in the literature the pressure snapshots are often modified in this way, e.g., see [2].

The reduced-order approximations of the velocity and the pressure fields read as follows:

$$\boldsymbol{u}(t,\boldsymbol{x}) \approx \boldsymbol{u}_{\mathrm{ro}}(t,\boldsymbol{x}) = \boldsymbol{u}_{\boldsymbol{g}}(t,\boldsymbol{x}) + \boldsymbol{u}_{r_v}(t,\boldsymbol{x}) = \boldsymbol{u}_{\boldsymbol{g}}(t,\boldsymbol{x}) + \sum_{i=1}^{r_v} \alpha_{v,i}(t)\boldsymbol{\phi}_{\mathrm{ro},i}(\boldsymbol{x}), \tag{5.5}$$

$$p(t,\boldsymbol{x}) \approx p_{\mathrm{ro}}(t,\boldsymbol{x}) = \overline{p}_h(\boldsymbol{x}) + p_{r_p}(t,\boldsymbol{x}) = \overline{p}_h(\boldsymbol{x}) + \sum_{j=1}^{r_p} \alpha_{p,j}(t)\psi_{\mathrm{ro},j}(\boldsymbol{x}). \tag{5.6}$$

Vectors $\underline{\alpha}_v = (\alpha_{v,1},\ldots,\alpha_{v,r_v})^T$ and $\underline{\alpha}_p = \big(\alpha_{p,1},\ldots,\alpha_{p,r_p}\big)^T$ represent the unknown ROM coefficients for the velocity and the pressure, respectively.

Altogether, the Galerkin ROM for the Navier–Stokes equations reads: Find $(\boldsymbol{u}_{\mathrm{ro}}, p_{\mathrm{ro}})$ with $\boldsymbol{u}_{\mathrm{ro}} - \boldsymbol{u}_{\boldsymbol{g}}\colon (0,T] \to \operatorname{span}\{\boldsymbol{\phi}_{\mathrm{ro},i}\}_{i=1}^{r_v}$ and $p_{\mathrm{ro}} - \overline{p}_h\colon (0,T] \to \operatorname{span}\{\psi_{\mathrm{ro},j}\}_{j=1}^{r_p}$ such that for $i=1,\ldots,r_v$, $j=1,\ldots,r_p$,

$$\begin{aligned}(\partial_t \boldsymbol{u}_{\mathrm{ro}}, \boldsymbol{\phi}_{\mathrm{ro},i}) + (\nu\nabla\boldsymbol{u}_{\mathrm{ro}}, \nabla\boldsymbol{\phi}_{\mathrm{ro},i}) + ((\boldsymbol{u}_{\mathrm{ro}}\cdot\nabla)\boldsymbol{u}_{\mathrm{ro}}, \boldsymbol{\phi}_{\mathrm{ro},i}) - (p_{\mathrm{ro}}, \nabla\cdot\boldsymbol{\phi}_{\mathrm{ro},i}) &= (\boldsymbol{f}, \boldsymbol{\phi}_{\mathrm{ro},i}),\\ -(\nabla\cdot\boldsymbol{u}_{\mathrm{ro}}, \psi_{\mathrm{ro},j}) &= 0,\end{aligned} \tag{5.7}$$

with the ROM coefficients of the initial velocity condition

$$\alpha_{v,i}(0) = \alpha_{v,i}^0 = \big(\boldsymbol{u}(0,\boldsymbol{x}) - \boldsymbol{u}_{\boldsymbol{g}}(0,\boldsymbol{x}), \boldsymbol{\phi}_{\mathrm{ro},i}(\boldsymbol{x})\big) = \big(\boldsymbol{u}^0 - \boldsymbol{u}_{\boldsymbol{g}}^0, \boldsymbol{\phi}_{\mathrm{ro},i}\big), \quad i=1,\ldots,r_v, \tag{5.8}$$

where $\boldsymbol{u}^0$ is the finite element approximation of the initial condition and $\boldsymbol{u}_{\boldsymbol{g}}^0$ the finite element representation of the function fulfilling the Dirichlet boundary values on the boundary part Γ_{D}.

The Galerkin ROM for the Navier–Stokes equations discretized in time by the one-step θ-scheme (superscript n) and linearized by the Picard iteration (subscript k) reads: For each $n=1,2,\ldots$ and given $(\boldsymbol{u}_{\mathrm{ro},k-1}^n, p_{\mathrm{ro},k-1}^n)$ with $k=1,2,\ldots$, find $(\boldsymbol{u}_{\mathrm{ro},k}^n, p_{\mathrm{ro},k}^n)$ with

$\boldsymbol{u}^n_{\mathrm{ro},k} - \boldsymbol{u}^n_{\boldsymbol{g}} \in \operatorname{span}\{\boldsymbol{\phi}_{\mathrm{ro},i}\}_{i=1}^{r_v}$ and $p^n_{\mathrm{ro},k} - \overline{p}_h \in \operatorname{span}\{\psi_{\mathrm{ro},j}\}_{j=1}^{r_p}$ such that

$$
\begin{aligned}
\left(\boldsymbol{u}^n_{\mathrm{ro},k}, \boldsymbol{\phi}_{\mathrm{ro},i}\right) + \theta\Delta t\left[\left(\nu\nabla\boldsymbol{u}^n_{\mathrm{ro},k}, \nabla\boldsymbol{\phi}_{\mathrm{ro},i}\right) + \left((\boldsymbol{u}^n_{\mathrm{ro},k-1}\cdot\nabla)\boldsymbol{u}^n_{\mathrm{ro},k}, \boldsymbol{\phi}_{\mathrm{ro},i}\right)\right]& \\
- \Delta t\left(\nabla\cdot\boldsymbol{\phi}_{\mathrm{ro},i}, p^n_{\mathrm{ro},k}\right)& \\
= \left(\boldsymbol{u}^{n-1}_{\mathrm{ro}}, \boldsymbol{\phi}_{\mathrm{ro},i}\right) - (1-\theta)\Delta t\Big[\left(\nu\nabla\boldsymbol{u}^{n-1}_{\mathrm{ro}}, \boldsymbol{\phi}_{\mathrm{ro},i}\right)& \qquad (5.9)\\
+ \left((\boldsymbol{u}^{n-1}_{\mathrm{ro}}\cdot\nabla)\boldsymbol{u}^{n-1}_{\mathrm{ro}}, \boldsymbol{\phi}_{\mathrm{ro},i}\right)\Big] + (1-\theta)\Delta t\left(\boldsymbol{f}^{n-1}, \boldsymbol{\phi}_{\mathrm{ro},i}\right)& \\
+ \theta\Delta t\left(\boldsymbol{f}^n, \boldsymbol{\phi}_{\mathrm{ro},i}\right),& \\
-\left(\nabla\cdot\boldsymbol{u}^n_{\mathrm{ro},k}, \psi_{\mathrm{ro},j}\right) = 0,&
\end{aligned}
$$

for all $i = 1, \dots, r_v$, $j = 1, \dots, r_p$.

5.3. Velocity ROM

In many, probably even most, published reports on reduced-order models based on the POD for incompressible flows, only velocity models are considered. This approach exploits the linearity of the divergence operator and of the POD procedure. If the raw velocity snapshots $\{\boldsymbol{u}_m\}_{m=1}^M$ are divergence-free, then also the modified ones $\{\boldsymbol{u}_m - \boldsymbol{u}_{\boldsymbol{g}}(t_m,\cdot)\}_{m=1}^M$ are divergence-free if $\boldsymbol{u}_{\boldsymbol{g}}$ is divergence-free. As every POD basis function $\boldsymbol{\phi}_{\mathrm{ro},i}$, $i = 1, \dots, r_v$, is a particular linear combination of the used snapshots, see (3.44), then the POD basis functions $\boldsymbol{\phi}_{\mathrm{ro},i}$, $i = 1, \dots, r_v$, are also divergence-free. In fact, if $\boldsymbol{u}_{\mathrm{ro}}$ and $\{\boldsymbol{\phi}_{\mathrm{ro},i}\}_{i=1}^{r_v}$ are divergence-free, the pressure term on the left-hand side of the momentum equation and the continuity equation of (5.7) drops out.

Finally, the projection-based velocity Galerkin ROM (v-ROM) has the following form: Find

$$\boldsymbol{u}_{\mathrm{ro}}(t,\boldsymbol{x}) = \boldsymbol{u}_{\boldsymbol{g}}(t,\boldsymbol{x}) + \boldsymbol{u}_{r_v}(t,\boldsymbol{x}) = \boldsymbol{u}_{\boldsymbol{g}}(t,\boldsymbol{x}) + \sum_{i=1}^{r_v}\alpha_{v,i}(t)\boldsymbol{\phi}_{\mathrm{ro},i}(\boldsymbol{x}),$$

such that, for $i = 1, \dots, r_v$,

$$(\partial_t\boldsymbol{u}_{\mathrm{ro}}, \boldsymbol{\phi}_{\mathrm{ro},i}) + (\nu\nabla\boldsymbol{u}_{\mathrm{ro}}, \nabla\boldsymbol{\phi}_{\mathrm{ro},i}) + ((\boldsymbol{u}_{\mathrm{ro}}\cdot\nabla)\boldsymbol{u}_{\mathrm{ro}}, \boldsymbol{\phi}_{\mathrm{ro},i}) = (\boldsymbol{f}, \boldsymbol{\phi}_{\mathrm{ro},i}). \qquad (5.10)$$

Note that (5.10) requires only the velocity POD basis. Due to the absence of the pressure in problem (5.10), the boundary conditions (5.1) have to be modified yielding

$$
\begin{aligned}
\boldsymbol{u} &= \boldsymbol{0} && \text{on} \quad (0,T]\times\Gamma_0,\\
\boldsymbol{u} &= \boldsymbol{g}_{\mathrm{D}} && \text{on} \quad (0,T]\times\Gamma_{\mathrm{in}}, \qquad (5.11)\\
\nabla\boldsymbol{u}\,\boldsymbol{n} &= \boldsymbol{0} && \text{on} \quad (0,T]\times\Gamma_{\mathrm{out}}.
\end{aligned}
$$

It must be emphasized that the assumption of divergence-free snapshots as the solution of a finite element method is idealized. For instance, in the context of inf-sup stable finite element discretizations there are only very few divergence-free pairs of spaces, like the Scott–Vogelius element on barycentric refined grids [147]. Most of the inf-sup stable pairs, in particular the most popular ones like the Taylor–Hood finite elements, see Section 2.2.7, are only discretely divergence-free. The magnitude of the divergence of the finite element solution can be even large [100]. Indeed, the standard finite element convergence theory shows that the $L^2(\Omega)$ norm of the divergence has the same order of convergence as the error in the $L^2(\Omega)$ norm of the velocity gradient [72].

The reduction from (5.7) to (5.10), however, can be achieved in certain situations by using the argument that the snapshots are only discretely divergence-free. This situation holds if the finite element continuity equation is not perturbed by any additional term. Moreover, the POD modes $\{\boldsymbol{\phi}_{\mathrm{ro},i}\}_{i=1}^{r_v}$ with the corresponding function $\boldsymbol{u_g}$ and $\{\psi_j\}_{j=1}^{r_p}$ with $\overline{p}_h$ should belong to the velocity and pressure finite element spaces, respectively. In this case, the pressure term in the ROM (5.7) drops out and (5.7) reduces to the velocity ROM (5.10). The above argument does not apply if the continuity equation is disturbed by additional terms, as in the case of finite element pairs that do not fulfill a discrete inf-sup condition, e.g., equal finite elements for velocity and pressure, which require additional stabilizations introducing a control on the pressure through a modification of the continuity equation (2.41) .

The velocity ROM (5.10) is continuous in time and nonlinear due to the convective term. Therefore, in order to solve it numerically it has to be discretized in time, see Section 2.2.3 and to be solved iteratively in each time step, see Section 2.2.5.

Let t^n denote the discrete times, the functions evaluated at those times with a corresponding superscript n, and the length of the equidistant time step by Δt. The k-th iteration of the nonlinear solver in each time step is denoted by the subscript k of the solution. The initial step for each nonlinear iteration at time t_n is the value of the solution evaluated at t_{n-1}, see [70] and references therein for other possible choices of the initial guess.

The velocity ROM (5.10) combined with the one-step θ-scheme and the Picard iteration reads in matrix form: For each $n = 1, 2, \ldots$ and given $\underline{\alpha}_{v,k-1}^n \in \mathbb{R}^{r_v}$ with $k = 1, 2, ...,$ find $\underline{\alpha}_{v,k}^n \in \mathbb{R}^{r_v}$ such that

$$\begin{aligned}\left[M_{\mathrm{ro}}^{\mathrm{NS}} + \theta\Delta t\left(A_{\mathrm{ro}}^{\mathrm{NS}} + N_{\mathrm{ro}}^{\mathrm{NS}}(\boldsymbol{u}_{\mathrm{ro},k-1}^n)\right)\right]\underline{\alpha}_{v,k}^n &= (1-\theta)\Delta t\underline{\boldsymbol{f}}_{\mathrm{ro}}^{n-1} + \theta\Delta t\underline{\boldsymbol{f}}_{\mathrm{ro}}^n \\ &+ \left[M_{\mathrm{ro}}^{\mathrm{NS}} - (1-\theta)\Delta t\left(A_{\mathrm{ro}}^{\mathrm{NS}} + N_{\mathrm{ro}}^{\mathrm{NS}}(\boldsymbol{u}_{\mathrm{ro}}^{n-1})\right)\right]\underline{\alpha}_v^{n-1} + \underline{\boldsymbol{l}}_{\mathrm{ro}}^n,\end{aligned} \tag{5.12}$$

where θ has to be chosen, see Table 2.1, and

$$\left(M_{\mathrm{ro}}^{\mathrm{NS}}\right)_{ij} = (\boldsymbol{\phi}_{\mathrm{ro},j}, \boldsymbol{\phi}_{\mathrm{ro},i}), \qquad i,j = 1,\ldots,r_v, \tag{5.13}$$

$$\left(A_{\mathrm{ro}}^{\mathrm{NS}}\right)_{ij} = (\nu\nabla\boldsymbol{\phi}_{\mathrm{ro},j}, \nabla\boldsymbol{\phi}_{\mathrm{ro},i}), \qquad i,j = 1,\ldots,r_v, \tag{5.14}$$

$$\left(N_{\mathrm{ro}}^{\mathrm{NS}}(\boldsymbol{u}_{\mathrm{ro}}^n)\right)_{ij} = \left((\boldsymbol{u}_{\mathrm{ro}}^n \cdot \nabla)\,\boldsymbol{\phi}_{\mathrm{ro},j}, \boldsymbol{\phi}_{\mathrm{ro},i}\right), \qquad i,j = 1,\ldots,r_v, \tag{5.15}$$

$$\boldsymbol{f}_{\mathrm{ro},i}^n = \left(\boldsymbol{f}^n, \boldsymbol{\phi}_{\mathrm{ro},i}\right), \qquad i = 1,\ldots,r_v, \tag{5.16}$$

and finally

$$\begin{aligned}\boldsymbol{l}_{\mathrm{ro},i}^n = &\left(\boldsymbol{u_g}^{n-1} - \boldsymbol{u_g}^n, \boldsymbol{\phi}_{\mathrm{ro},i}\right) - \Delta t\left(\theta\nu\nabla\boldsymbol{u_g}^n + (1-\theta)\nu\nabla\boldsymbol{u_g}^{n-1}, \nabla\boldsymbol{\phi}_{\mathrm{ro},i}\right) \\ &- \Delta t\left(\theta\left(\boldsymbol{u}_{\mathrm{ro},k-1}^n \cdot \nabla\right)\boldsymbol{u_g}^n + (1-\theta)\left(\boldsymbol{u}_{\mathrm{ro}}^{n-1} \cdot \nabla\right)\boldsymbol{u_g}^{n-1}, \boldsymbol{\phi}_{\mathrm{ro},i}\right), \quad i = 1,\ldots,r_v.\end{aligned} \tag{5.17}$$

The initial condition is given by (5.8).

5.3.1. Implementation

In this section, the implementation of the velocity ROM (5.12) equipped with the Dirichlet boundary conditions of type (3.71), i.e.,

$$\boldsymbol{u} = \gamma_k(t)\boldsymbol{g}_k(\boldsymbol{x}) \quad \text{on} \quad (0,T] \times \Gamma_{\mathrm{D},k}, \quad k = 1,\ldots,K, \tag{5.18}$$

with $\Gamma_{\mathrm{D}} = \bigcup_{k=1}^{K} \Gamma_{\mathrm{D},k}$, $\Gamma_{\mathrm{D},i} \cap \Gamma_{\mathrm{D},j} = \emptyset$ for $i \neq j$, $i,j = 1,\ldots,K$, will be discussed. Moreover, a zero source term, i.e., $\boldsymbol{f} = \boldsymbol{0}$, will be assumed which corresponds to the definition of the problems considered in Section 5.5.

Remark 5.1. The Dirichlet part of the boundary conditions (5.11) can be interpreted to be of type (5.18) for $K = 2$ with

$$\begin{aligned} \boldsymbol{u} &= \gamma_1(t)\boldsymbol{g}_1(\boldsymbol{x}) && \text{on} \quad (0,T] \times \Gamma_{\mathrm{in}} = \Gamma_{\mathrm{D},1}, \\ \boldsymbol{u} &= \gamma_2(t)\boldsymbol{g}_2(\boldsymbol{x}) && \text{on} \quad (0,T] \times \Gamma_0 = \Gamma_{\mathrm{D},2}, \end{aligned} \tag{5.19}$$

where $\boldsymbol{g}_{\mathrm{D}}(t,\boldsymbol{x}) = \gamma_1(t)\boldsymbol{g}_1(\boldsymbol{x})$, and $\gamma_2 = 1$, $\boldsymbol{g}_2 = \boldsymbol{0}$. ◁

As in Section 3.2.4, the velocity POD basis $\{\boldsymbol{\phi}_{\mathrm{ro},i}\}_{i=1}^{r_v}$ is computed from the modified snapshots following the algorithm of Method 1 introduced in Section 3.2.3. Hence, the function $\boldsymbol{u_g}$ satisfying the boundary conditions (5.18) reads

$$\boldsymbol{u_g}(t,\boldsymbol{x}) = \sum_{k=1}^{K} \gamma_k(t)\boldsymbol{u}_{S,k}(\boldsymbol{x}), \tag{5.20}$$

with the functions $\boldsymbol{u}_{S,k} \in V_h$ obtained as it is specified in the algorithm.

Let Φ_v denote the velocity POD matrix with the velocity POD modes as columns, i.e.,

$$\Phi_v = \left(\boldsymbol{\phi}_{\mathrm{ro},1},\ldots,\boldsymbol{\phi}_{\mathrm{ro},r_v}\right) \in \mathbb{R}^{N_v \times r_v}.$$

With the exception of the contributions from the convective term, i.e., the matrix $N_{\mathrm{ro}}^{\mathrm{NS}}(\boldsymbol{u}_{\mathrm{ro}}^{n})$ and the last term of $\underline{\boldsymbol{l}}_{\mathrm{ro}}^{n}$, all other parts of the system (5.12) can be built up efficiently following the lines in Section 3.2.4. Thereby, expressions M_{ro}, A_{ro}, M_h, A_h, Φ, $\underline{u}_{S,k}$ have to be substituted by $M_{\mathrm{ro}}^{\mathrm{NS}}$, $A_{\mathrm{ro}}^{\mathrm{NS}}$, M_h^{NS}, A_h^{NS}, Φ_v, $\underline{\boldsymbol{u}}_{S,k}$, respectively.

Next, the implementation of the contributions from the convective term will be discussed, i.e., expressions of the form

$$\left(N_{\mathrm{ro}}^{\mathrm{NS}}(\boldsymbol{u}_{\mathrm{ro}}^{n-1})\right)_{ij} = \left(\left(\boldsymbol{u}_{\boldsymbol{g}}^{n-1}\cdot\nabla\right)\boldsymbol{\phi}_{\mathrm{ro},j},\boldsymbol{\phi}_{\mathrm{ro},i}\right) + \left(\left(\boldsymbol{u}_{r_v}^{n-1}\cdot\nabla\right)\boldsymbol{\phi}_{\mathrm{ro},j},\boldsymbol{\phi}_{\mathrm{ro},i}\right) \tag{5.21}$$

and

$$\left(\left(\boldsymbol{u}_{\mathrm{ro}}^{n-1}\cdot\nabla\right)\boldsymbol{u}_{\boldsymbol{g}}^{n-1},\boldsymbol{\phi}_{\mathrm{ro},i}\right) = \left(\left(\boldsymbol{u}_{\boldsymbol{g}}^{n-1}\cdot\nabla\right)\boldsymbol{u}_{\boldsymbol{g}}^{n-1},\boldsymbol{\phi}_{\mathrm{ro},i}\right) + \left(\left(\boldsymbol{u}_{r_v}^{n-1}\cdot\nabla\right)\boldsymbol{u}_{\boldsymbol{g}}^{n-1},\boldsymbol{\phi}_{\mathrm{ro},i}\right). \tag{5.22}$$

Due to the separated form of $\boldsymbol{u_g}$, see (5.20), and the definition of $\boldsymbol{u}_{r_v}$, see (5.5), the computation of both terms on the right-hand side of (5.21) can be split up efficiently between the offline and online stages in three steps as follows:

1. Offline stage: Assemble finite element matrices
$$\{N_h^{\mathrm{NS}}(\boldsymbol{u}_{S,k})\}_{k=1}^{K} \quad \text{and} \quad \{N_h^{\mathrm{NS}}(\boldsymbol{\phi}_{\mathrm{ro},i})\}_{i=1}^{r_v}, \tag{5.23}$$
defined by (2.97).

2. Offline stage: Reduce the matrices by building
$$\{\Phi_v^T N_h^{\mathrm{NS}}(\boldsymbol{u}_{S,k})\Phi_v\}_{k=1}^{K} \quad \text{and} \quad \{\Phi_v^T N_h^{\mathrm{NS}}(\boldsymbol{\phi}_{\mathrm{ro},i})\Phi_v\}_{i=1}^{r_v}, \tag{5.24}$$
respectively. Each matrix is an element of the space $\mathbb{R}^{r_v \times r_v}$.

3. Online stage: Compute
$$\sum_{k=1}^{K} \gamma_k^{n-1} \Phi_v^T N_h^{\mathrm{NS}}(\boldsymbol{u}_{S,k})\Phi_v \quad \text{and} \quad \sum_{i=1}^{r_v} \alpha_{v,i}^{n-1} \Phi_v^T N_h^{\mathrm{NS}}(\boldsymbol{\phi}_{\mathrm{ro},i})\Phi_v$$
to obtain the first and the second terms of (5.21), respectively, at each iteration step.

Hence, in the online stage the computational complexity does not depend on the dimension of the finite element space V_h as it is desired in the framework of the reduced-order modeling.

Similarly, one can split the computation of both terms on the right-hand side of (5.22). In this case, the first step remains the same, i.e., the finite element matrices (5.23) need to be assembled. In the second step, instead of (5.24), one pre-computes
$$\begin{aligned} &\Phi_v^T N_h^{\mathrm{NS}}(\boldsymbol{u}_{S,k})\underline{\boldsymbol{u}}_{S,m}, && k, m = 1, \ldots, K, \\ &\Phi_v^T N_h^{\mathrm{NS}}(\boldsymbol{\phi}_{\mathrm{ro},i})\underline{\boldsymbol{u}}_{S,m}, && i = 1, \ldots, r_v,\ m = 1, \ldots, K, \end{aligned} \tag{5.25}$$
in the offline stage. Note that each expression in (5.25) is a vector in $\mathbb{R}^{r_v}$. Finally, at each iteration of the online stage, the first and the second summands of (5.22) can be obtained efficiently by computing
$$\sum_{k=1}^{K}\sum_{m=1}^{K} \gamma_k^{n-1}\gamma_m^{n-1} \Phi_v^T N_h^{\mathrm{NS}}(\boldsymbol{u}_{S,k})\underline{\boldsymbol{u}}_{S,m} \quad \text{and} \quad \sum_{i=1}^{r_v}\sum_{m=1}^{K} \alpha_{v,i}^{n-1}\gamma_m^{n-1} \Phi_v^T N_h^{\mathrm{NS}}(\boldsymbol{\phi}_{\mathrm{ro},i})\underline{\boldsymbol{u}}_{S,m},$$
respectively.

5.4. Pressure ROMs

To the best of the author's knowledge, the ROMs with a pressure component can be divided into two classes, depending on if they use pressure POD modes or not. If pressure modes are employed, there are again two principal approaches. In the decoupled approach, the velocity and pressure snapshots are considered separately. Choosing the velocity POD modes with the highest kinetic energy and the pressure POD modes with the largest $L^2(\Omega)$ norm, one obtains two separate bases. For this approach, it is straightforward to choose a different number of POD modes for velocity and pressure, based on the corresponding distribution of their eigenvalues. In the coupled approach, see [19], each snapshot, and thus, each POD mode, has both a velocity and the corresponding pressure component. This approach naturally yields the same number of velocity and pressure modes. In this thesis the decoupled approach will be considered.

5.4.1. Pressure ROM Based on Velocity Modes

After having solved the velocity ROM (5.10), the pressure field must be reconstructed a posteriori using the ROM velocity solution. In this section, the approach proposed in [108] will be considered. It utilizes the pressure Poisson equation

$$-\Delta p = \nabla \cdot ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) \quad \text{in} \quad \Omega, \tag{5.26}$$

which is obtained by taking the divergence of the momentum equation of the Navier–Stokes equations (2.40)-(2.41), see Section 2.2.2, assuming that the source term $\boldsymbol{f}$ is divergence-free. For the sake of simplicity, the pressure model will be derived for the stationary Dirichlet boundary condition for $\boldsymbol{u}$ on Γ_{in} such that the function $\boldsymbol{u}_{\text{g}}$ in (5.5) can be chosen to be the average of the snapshots $\overline{\boldsymbol{u}}_h$, see Section 3.2.3. Thus, the problem (5.26) is equipped with the Neumann boundary condition for the pressure

$$\nabla p \cdot \boldsymbol{n} = \nu \Delta \boldsymbol{u} \cdot \boldsymbol{n} - (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} \cdot \boldsymbol{n} \quad \text{on} \quad \Gamma \setminus \Gamma_{\text{out}}, \tag{5.27}$$

and the homogeneous Dirichlet boundary condition on Γ_{out}, see also Section 2.2.2. The main idea used in [108] consists in replacing $\boldsymbol{u}$ on the right-hand side of (5.26) by $\boldsymbol{u}_{\text{ro}}$ defined by (5.5), which can also be written in the form

$$\boldsymbol{u}_{\text{ro}}(t, \boldsymbol{x}) = \sum_{i=0}^{r_v} \alpha_{v,i}(t) \boldsymbol{\phi}_{\text{ro},i}(\boldsymbol{x}), \tag{5.28}$$

with $\alpha_{v,0}(t) = 1$ and $\boldsymbol{\phi}_{\text{ro},0}(\boldsymbol{x}) = \overline{\boldsymbol{u}}_h(\boldsymbol{x})$. Assuming that all velocity POD functions in (5.5) are divergence-free, one obtains in Ω

$$
\begin{aligned}
-\Delta p_{\mathrm{ro}} &= \sum_{i=0}^{r_v}\sum_{j=0}^{r_v} \alpha_{v,i}(t)\alpha_{v,j}(t)\,\nabla\cdot\left(\left(\boldsymbol{\phi}_{\mathrm{ro},i}\cdot\nabla\right)\boldsymbol{\phi}_{\mathrm{ro},j}\right)\\
&= \sum_{i=0}^{r_v}\sum_{j=0}^{r_v} \alpha_{v,i}(t)\alpha_{v,j}(t)\sum_{k=1}^{d}\partial_{x^k}\left(\left(\sum_{m=1}^{d}\boldsymbol{\phi}^m_{\mathrm{ro},i}\partial_{x^m}\right)\boldsymbol{\phi}^k_{\mathrm{ro},j}\right)\\
&= \sum_{i=0}^{r_v}\sum_{j=0}^{r_v} \alpha_{v,i}(t)\alpha_{v,j}(t)\left[\sum_{k=1}^{d}\sum_{m=1}^{d}\partial_{x^k}\boldsymbol{\phi}^m_{\mathrm{ro},i}\partial_{x^m}\boldsymbol{\phi}^k_{\mathrm{ro},j}\right.\\
&\quad\left.+\sum_{m=1}^{d}\boldsymbol{\phi}^m_{\mathrm{ro},i}\partial_{x^m}\underbrace{\left(\sum_{k=1}^{d}\partial_{x^k}\boldsymbol{\phi}^k_{\mathrm{ro},i}\right)}_{=0}\right]\\
&= \sum_{i=0}^{r_v}\sum_{j=0}^{r_v} \alpha_{v,i}(t)\alpha_{v,j}(t)\left(\sum_{k=1}^{d}\sum_{m=1}^{d}\partial_{x^k}\boldsymbol{\phi}^m_{\mathrm{ro},i}\partial_{x^m}\boldsymbol{\phi}^k_{\mathrm{ro},j}\right),
\end{aligned}
\tag{5.29}
$$

where $\boldsymbol{\phi}^k_{\mathrm{ro},i}$ denotes the kth component of the velocity POD mode $\boldsymbol{\phi}_{\mathrm{ro},i}$, ∂_{x^k} denotes the partial derivative with respect to the k-th component of $\boldsymbol{x}\in\Omega$, and d is the dimension of the domain Ω.

Problem (5.29) is an equation in space, in which the functions $\alpha_{v,i}(t),\alpha_{v,j}(t)$ act as constants. Hence, the solution of (5.29) has the quadratic form

$$
p_{\mathrm{ro}}(t,\boldsymbol{x}) = \sum_{i=0}^{r_v}\sum_{j=0}^{r_v}\alpha_{v,i}(t)\alpha_{v,j}(t)p_{ij}(\boldsymbol{x})\,, \tag{5.30}
$$

where the functions $p_{ij}(\boldsymbol{x})$ are obtained by solving

$$
-\Delta p_{ij} = \sum_{k=1}^{d}\sum_{m=1}^{d}\partial_{x^k}\boldsymbol{\phi}^k_{\mathrm{ro},i}\partial_{x^m}\boldsymbol{\phi}^m_{\mathrm{ro},j} \quad \text{in} \quad \Omega. \tag{5.31}
$$

In what follows, the velocity ROM (5.12) together with $p_{\mathrm{ro}}(\boldsymbol{x})$ given by (5.30), will be referred to as the VMB-ROM (velocity-modes-based ROM). Within the framework of the numerical studies in Section 5.5, the coefficients p_{ij} will be determined by applying the Galerkin projection to (5.31) on the pressure finite element space Q_h. Using integration by parts, the resulting left-hand side can be expressed for all $q_h\in Q_h$ by

$$
(-\Delta p_{ij},q_h) = (\nabla p_{ij},\nabla q_h) - \int\limits_{\Gamma\backslash\Gamma_{\mathrm{out}}}\nabla p_{ij}\cdot\boldsymbol{n}\,q_h\,d\boldsymbol{s}. \tag{5.32}
$$

Finally, the following finite element problem has to be solved: Find $p_{ij}\in Q_h$ such that

$$
(\nabla p_{ij},\nabla q_h) = \left(\sum_{k=1}^{d}\sum_{m=1}^{d}\partial_{x^k}\boldsymbol{\phi}^m_{\mathrm{ro},i}\partial_{x^m}\boldsymbol{\phi}^k_{\mathrm{ro},j},q_h\right) + \int\limits_{\Gamma\backslash\Gamma_{\mathrm{out}}}\nabla p_{ij}\cdot\boldsymbol{n}\,q_h\,d\boldsymbol{s},\quad \forall q_h\in Q_h. \tag{5.33}
$$

By applying the velocity ROM approximation (5.28) instead of $\boldsymbol{u}$, the Neumann boundary condition (5.27) on $\Gamma \setminus \Gamma_{\text{out}}$ can be represented by

$$\nabla p_{\text{ro}} \cdot \boldsymbol{n} = \sum_{i=0}^{r_v} \alpha_{v,i} \, \big(\nu \Delta \boldsymbol{\phi}_{\text{ro},i} - \left(\boldsymbol{\phi}_{\text{ro},i} \cdot \nabla\right) \overline{\boldsymbol{u}}_h\big) \cdot \boldsymbol{n}, \tag{5.34}$$

where it was used that the velocity POD modes $\{\boldsymbol{\phi}_{\text{ro},i}\}_{i=1}^{r_v}$ vanish on $\Gamma \setminus \Gamma_{\text{out}}$ by construction. In order to maintain the quadratic representation as in (5.30), the right-hand side can be rewritten leading to the form

$$\sum_{i=0}^{r_v} \alpha_{v,i} \, s_i = \begin{pmatrix} 1 & \alpha_{v,1} & \cdots & \alpha_{v,r_v} \end{pmatrix} \begin{pmatrix} s_0 & \frac{1}{2}s_1 & \cdots & \frac{1}{2}s_{r_v} \\ \frac{1}{2}s_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}s_{r_v} & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 1 \\ \alpha_{v,1} \\ \cdots \\ \alpha_{v,r_v} \end{pmatrix},$$

with $s_i = \big(\nu \Delta \boldsymbol{\phi}_{\text{ro},i} - \left(\boldsymbol{\phi}_{\text{ro},i} \cdot \nabla\right) \overline{\boldsymbol{u}}_h\big) \cdot \boldsymbol{n}$. Finally, the boundary term on the right-hand side of (5.33) can be computed by employing the following expressions

$$\nabla p_{ij} \cdot \boldsymbol{n} = \begin{cases} s_0, & \text{if} \quad i, j = 0, \\ \frac{1}{2}s_i, & \text{if} \quad i > 0, \, j = 0, \\ \frac{1}{2}s_j, & \text{if} \quad j > 0, \, i = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Note that $p_{ij}(\boldsymbol{x}) = p_{ji}(\boldsymbol{x})$ and, thus, system (5.31) has only $(r_v + 1)r_v/2$ unknowns. The functions $p_{ij}(\boldsymbol{x})$ can be computed in the offline stage. In this way, the ROM pressure $p_{\text{ro}}(\boldsymbol{x})$ can be efficiently reconstructed at each time step by using (5.30). It should be noted that this pre-processing approach does not work if the Navier–Stokes equations have a time-dependent body force which is not divergence-free.

In [108], the term $(\nabla p_{\text{ro}}, \boldsymbol{\phi}_{\text{ro},i})$ was even introduced into the momentum equation (5.10) to improve the results of ROMs for shear flows. If only the term $(\nabla p_{\text{ro}}, \boldsymbol{\phi}_{\text{ro},i})$ is of interest, while the explicit computation of p_{ro} is not required, it was proposed in [42,108] to approximate this term using linear models in $\alpha_{v,i}(t)$, resulting in additional minimization problems to be solved for determining the coefficients in the ansatz.

5.4.2. Pressure ROM Based on Pressure Modes

A second approach for defining a pressure ROM a posteriori consists in reducing the pressure equation (5.26) (see also Section 2.2.2) using a pressure POD basis computed from the pressure snapshots. This method was proposed in [2].

Assuming that the reduced velocity $\boldsymbol{u}_{\text{ro}}^n$ has already been computed by the velocity ROM (5.12), and assuming the velocity POD space to be divergence-free, one obtains the pressure Poisson equation

$$-\Delta p_{\text{ro}}^n = \nabla \cdot ((\boldsymbol{u}_{\text{ro}}^n \cdot \nabla)\boldsymbol{u}_{\text{ro}}^n) \quad \text{in} \quad \Omega, \tag{5.35}$$

with a homogeneous Dirichlet boundary condition on Γ_{out} and a Neumann boundary condition on $\Gamma \setminus \Gamma_{\text{out}}$ (5.27), using $\boldsymbol{u}_{\text{ro}}$ instead of $\boldsymbol{u}$.

In [2], it was suggested to compute the ROM pressure by applying the Galerkin projection to (5.35) on the pressure POD modes $\{\psi_{\text{ro},i}\}_{i=1}^{r_p}$. Employing integration by parts yields

$$\begin{aligned}(-\Delta p_{\text{ro}}^n, \psi_{\text{ro},i}) &= (\nabla p_{\text{ro}}^n, \nabla \psi_{\text{ro},i}) - \int\limits_{\Gamma\setminus\Gamma_{\text{out}}} \nabla p_{\text{ro}}^n \cdot \boldsymbol{n}\, \psi_{\text{ro},i}\, d\boldsymbol{s} \\ &= (\nabla p_{\text{ro}}^n, \nabla \psi_{\text{ro},i}) - \int\limits_{\Gamma\setminus\Gamma_{\text{out}}} \left(\nu \Delta \boldsymbol{u}_{\text{ro}}^n - (\boldsymbol{u}_{\text{ro}}^n \cdot \nabla)\, \boldsymbol{u}_{\boldsymbol{g}}^n\right) \cdot \boldsymbol{n}\, \psi_{\text{ro},i}\, d\boldsymbol{s},\end{aligned} \tag{5.36}$$

where it was used that $\boldsymbol{u}_{r_v}^n$ vanishes on $\Gamma \setminus \Gamma_{\text{out}}$ by the construction of the POD modes $\{\boldsymbol{\phi}_{\text{ro},i}\}_{i=1}^{r_v}$. Finally, one has to solve the following problem: Find

$$p_{\text{ro}}^n = \overline{p}_h + \sum_{i=1}^{r_p} \alpha_{p,i}^n \psi_{\text{ro},i}\,, \tag{5.37}$$

such that for $i = 1, \ldots, r_p$,

$$\begin{aligned}(\nabla p_{\text{ro}}^n, \nabla \psi_{\text{ro},i}) =& \left(\nabla \cdot ((\boldsymbol{u}_{\text{ro}}^n \cdot \nabla)\boldsymbol{u}_{\text{ro}}^n)\,, \psi_{\text{ro},i}\right) \\ &+ \int\limits_{\Gamma\setminus\Gamma_{\text{out}}} \left(\nu \Delta \boldsymbol{u}_{\text{ro}}^n - (\boldsymbol{u}_{\text{ro}}^n \cdot \nabla)\, \boldsymbol{u}_{\boldsymbol{g}}^n\right) \cdot \boldsymbol{n}\, \psi_{\text{ro},i}\, d\boldsymbol{s}.\end{aligned} \tag{5.38}$$

In the numerical studies presented in Section 5.5, the velocity ROM (5.12) together with (5.38), will be referred to as PMB-ROM (pressure-modes-based ROM).

Note that, although VMB-ROM and PMB-ROM are based on the same equation, the respective pressures are computed using different discrete spaces. In the VMB-ROM, the pressure is represented in terms of the functions computed from the derivatives of the velocity POD modes, see (5.31), whereas in the PMB-ROM the pressure is represented in terms of the pressure POD modes, cf. (5.37).

5.4.3. Pressure ROM Based on a Stabilization of the Coupled Problem

A ROM that is based on a coupled scheme for $(\boldsymbol{u}_{\text{ro}}, p_{\text{ro}})$, such as the ROM (5.7), raises the issue of the inf-sup condition for saddle point problems [45]. It seems to be hard to address this question for the general setting of the ROM, unlike for, e.g, finite element methods, where the approximation spaces are specified beforehand and the corresponding discrete inf-sup condition (2.106) can be investigated a priori. In the ROM framework based on POD, however, the approximation spaces are problem-dependent – they are known only after having performed the underlying finite element simulations, or even an actual physical experiment [9, 63]. Thus, checking beforehand whether the velocity and pressure POD spaces satisfy an inf-sup condition is generally not possible. In the context of finite element methods, the discrete inf-sup condition (2.106) states, loosely speaking, that the dimension of the discrete velocity space is sufficiently high compared with the dimension of the discrete pressure space. In the case of reduced basis method, several suggestions exist in the literature on how to enrich the velocity space to verify the inf-sup condition [96,112,119]. For ROMs based on POD, to the author's best knowledge, there are no results on that issue. In the framework of finite element methods, the coupled velocity-pressure problem (2.93) can be stabilized by including additional terms in the variational formulation, in order to overcome a possible violation of the inf-sup condition. This aspect motivates the new ROM for the pressure introduced in this section. The idea is to define an equation for the pressure based on a stabilization approach for the coupled velocity-pressure ROM (5.7). Among the stabilizations for incompressible flow problems [21], the class of residual-based approaches seems to be promising, since these methods immediately allow the stabilization of dominant convection. These approaches are also the basis of residual-based variational multiscale methods [17].

A popular residual-based stabilization is the SUPG/PSPG/grad-div method, see [21] and the references therein. In this approach, the residual of the momentum equation is tested with the streamline derivative of the velocity and the gradient of the pressure. Thus, the following stabilization term is added to the momentum equation in (2.93)

$$s_h = \sum_{K \in \mathcal{T}_h} \left(\partial_t \boldsymbol{u}_h - \nu \Delta \boldsymbol{u}_h + (\boldsymbol{u}_h \cdot \nabla)\boldsymbol{u}_h + \nabla p_h - \boldsymbol{f}, \delta_{K,\boldsymbol{u}_h} (\boldsymbol{u}_h \cdot \nabla)\boldsymbol{v}_h + \delta_{K,p} \nabla q_h \right)_K , \quad (5.39)$$

where K denotes a mesh cell of the considered triangulation $\mathcal{T}_h$ of Ω, and $\delta_{K,\boldsymbol{u}}$ and $\delta_{K,p}$ are the stabilization parameter functions. The so-called grad-div term is based on the residual of the continuity equation and it adds the following stabilization term to the momentum equation in (2.93)

$$\sum_{K \in \mathcal{T}_h} \left(\nabla \cdot \boldsymbol{u}_h, \mu_K \nabla \cdot \boldsymbol{v}_h \right)_K , \quad (5.40)$$

where μ_K denotes the stabilization parameter function. The SUPG term in (5.39) accounts for stabilizing dominating convection, the grad-div term (5.40) accounts for improving the discrete conservation of mass, and the PSPG term in (5.39) accounts for stabilizing a violated inf-sup condition.

Note that the SUPG/PSPG/grad-div method has already been used in [19,145] within a ROM framework. However, in [19,145] the ROM pressure was not computed by solving a separate pressure equation.

Although an explicit treatment of (5.39) and (5.40) might be advantageous in terms of computational efficiency, the stabilization of the inf-sup condition has to appear in the system matrix in order to become effective.

In the residual for the momentum balance (5.39), the viscous term is generally neglected, since it is of little importance in the interesting case of small viscosity. Denote by

$$\boldsymbol{res}_{\mathrm{ro}}^n = \frac{\boldsymbol{u}_{\mathrm{ro}}^n - \boldsymbol{u}_{\mathrm{ro}}^{n-1}}{\Delta t} + (\boldsymbol{u}_{\mathrm{ro}}^n \cdot \nabla)\boldsymbol{u}_{\mathrm{ro}}^n + \nabla p_{\mathrm{ro}}^n - \boldsymbol{f}^n$$

an reduced-order approximation of the residual at t_n. Then, the right-hand side of the momentum equation of the coupled system (5.9) at t_n contains the explicit stabilization terms

$$-\sum_{K\in\mathcal{T}_h}\delta_{K,\boldsymbol{u}}\big(\boldsymbol{res}_{\mathrm{ro}}^{n-1},(\boldsymbol{u}_{\mathrm{ro}}^{n-1}\cdot\nabla)\boldsymbol{\phi}_{\mathrm{ro},i}\big)_K,\quad -\sum_{K\in\mathcal{T}_h}\mu_K\big(\nabla\cdot\boldsymbol{u}_{\mathrm{ro}}^{n-1},\nabla\cdot\boldsymbol{\phi}_{\mathrm{ro},i}\big)_K, \tag{5.41}$$

$\forall i = 1\ldots,r_v$, where the stabilization parameters are now assumed to be piecewise constant. In the continuity equation in (5.9), the term

$$\sum_{K\in\mathcal{T}^h}\delta_{K,p}\big(\nabla p_{\mathrm{ro}}^n,\nabla\psi_{\mathrm{ro},j}\big)_K,\quad j=1,\ldots,r_p,$$

is included to the left-hand side, i.e., to the system matrix. Moving the velocity-pressure coupling of the stabilization

$$-\sum_{K\in\mathcal{T}_h}\delta_{K,p}\big(\boldsymbol{res}_{\mathrm{ro}}^n-\nabla p_{\mathrm{ro}}^n,\nabla\psi_{\mathrm{ro},j}\big)_K,\quad j=1,\ldots,r_p, \tag{5.42}$$

to the right-hand side of the continuity equation in (5.9), the matrix form of the coupled problem has the form

$$\begin{pmatrix} M_{\mathrm{ro}}^{\mathrm{NS}}+\theta\Delta t\left(A_{\mathrm{ro}}^{\mathrm{NS}}+N_{\mathrm{ro}}^{\mathrm{NS}}(\boldsymbol{u}_{\mathrm{ro},k-1}^n)\right) & (B_{\mathrm{ro}}^{\mathrm{NS}})^T\\ B_{\mathrm{ro}}^{\mathrm{NS}} & C_{\mathrm{ro}}^{\mathrm{NS}}\end{pmatrix}, \tag{5.43}$$

where $M_{\mathrm{ro}}^{\mathrm{NS}}$, $A_{\mathrm{ro}}^{\mathrm{NS}}$, $N_{\mathrm{ro}}^{\mathrm{NS}}(\boldsymbol{u}_{\mathrm{ro},k-1}^n)$ are defined by (5.13), (5.14), and (5.15), respectively, and

$$\begin{aligned}(B_{\mathrm{ro}}^{\mathrm{NS}})_{ij}&=(\nabla\cdot\boldsymbol{\phi}_{\mathrm{ro},j},\psi_{\mathrm{ro},i}), && i=1,\ldots,r_p,\ j=1,\ldots,r_v,\\ (C_{\mathrm{ro}}^{\mathrm{NS}})_{ij}&=\sum_{K\in\mathcal{T}_h}\delta_{K,p}(\nabla\psi_{\mathrm{ro},j},\nabla\psi_{\mathrm{ro},i})_K, && i,j=1,\ldots,r_p.\end{aligned}$$

Consider now the ROM matrix (5.43) for the case in which the velocity snapshots are discretely divergence-free, e.g., when they are computed with a Galerkin finite element method with inf-sup stable pairs of finite element spaces. In this case, the matrix $B_{\mathrm{ro}}^{\mathrm{NS}}$ vanishes. Hence, the system with matrix (5.43) results in two decoupled equations. After having computed the reduced velocity, the right-hand side (5.42) of the continuity equation can be evaluated. If the stabilizations of dominating convection and of violating the mass conservation (5.41) can be neglected, as for the flow problem considered in Section 5.5, the velocity equation corresponding to (5.43) is the same as that in the velocity ROM (5.12).

Altogether, we propose to combine the ROM velocity equation (5.12) with

$$\begin{aligned}&\sum_{K\in\mathcal{T}_h}\delta_{K,p}\big(\nabla p_{\mathrm{ro}}^n,\nabla\psi_{\mathrm{ro},j}\big)_K=\\ &=-\sum_{K\in\mathcal{T}_h}\delta_{K,p}\left(\frac{\boldsymbol{u}_{\mathrm{ro}}^n-\boldsymbol{u}_{\mathrm{ro}}^{n-1}}{\Delta t}+(\boldsymbol{u}_{\mathrm{ro}}^n\cdot\nabla)\boldsymbol{u}_{\mathrm{ro}}^n-\boldsymbol{f}^n,\nabla\psi_{\mathrm{ro},j}\right)_K,\quad j=1,\ldots,r_p.\end{aligned} \tag{5.44}$$

In what follows, the ROM (5.12) together with (5.44), will be referred to as SM-ROM (stabilization-motivated ROM). The SM-ROM (5.12), (5.44) was first proposed in [26].

The matrix for the pressure equation in (5.43) corresponds to the discretization of a scaled Laplacian. In (5.44), the stabilization parameters $\{\delta_{K,p}\}$ have to be chosen. Since there is no numerical analysis for this choice in the context of ROMs, the guidance provided by the standard finite element theory will be used. For this case, following the finite element theory, an optimal stabilized method is obtained with $\delta_{K,p} = C\, h_K$ in (5.44), where C is a generic constant and h_K is the diameter of the mesh cell K, [21]. Note that the value of the constant C has no effect on the SM-ROM, since it appears on both sides of (5.44). Thus, without loss of generality, $\delta_{K,p} = h_K$ can be used.

It is worth emphasizing that one of the advantages of SM-ROM is that its derivation requires the velocity snapshots to be only discretely divergence-free but not pointwise divergence-free, as needed for the derivation of the VMB-ROM and PMB-ROM. Furthermore, being based on a general formulation of the Navier–Stokes equations, the SM-ROM does not need any ad hoc treatment of external forces and any specification of additional pressure boundary conditions.

5.5. Numerical Studies

First, this section presents numerical results for the three vp-ROMs introduced in Section 5.4, which are summarized in Table 5.1. Second, the influence of nonlinear iteration schemes of different complexity and accuracy, employed in the velocity ROM, on the accuracy of the vp-ROM results with respect to various quantities of interest will be studied. Third, it investigates the impact of the snapshot accuracy on the results of the vp-ROMs. The effect of the dimension of the POD basis on the numerical results is also monitored.

Table 5.1.: Velocity-pressure ROMs presented in Section 5.4. VMB-ROM and PMB-ROM use the same equation for computing the ROM pressure, but the discrete spaces in these methods are different. PMB-ROM and SM-ROM apply the same discrete space, but different equations for the ROM pressure.

Acronym	Description	Equations
VMB-ROM	velocity-modes based	(5.12), (5.31)
PMB-ROM	pressure-modes based	(5.12), (5.38)
SM-ROM	stabilization motivated	(5.12), (5.44)

Example 5.1. *Laminar flow around a cylinder.* To allow a detailed discussion of the results, the numerical studies were carried out for the well understood example of a two-dimensional laminar flow around a circular cylinder defined in [122]. This problem is given in

$$\Omega = \{(0,2.2) \times (0,0.41)\} \setminus \{\boldsymbol{x} \ : \ (\boldsymbol{x} - (0.2,0.2))^2 \le 0.05^2\},$$

see Fig. 5.1. At the boundary $x = 0$ the steady-state inflow condition $\boldsymbol{u}(x,0) = (0.41^{-2}(6y(0.41-y)),0)^T$ is used, at the boundary $x = 2.2$ the outflow ("do nothing") condition $(\nu\nabla\boldsymbol{u} - p\mathbb{I})\boldsymbol{n} = \boldsymbol{0}$ is applied, while no-slip boundary conditions are prescribed elsewhere, see Section 3.2.3. The kinematic viscosity of the fluid is given by $\nu = 10^{-3}$ m^2/s. The initial condition is a fully developed flow field that has to be computed in a pre-processing step. Based on the mean inflow velocity $U = 1$ m/s, the diameter of the cylinder $L = 0.1$ m and the kinematic viscosity, the Reynolds number of the flow is $Re = 100$. In the fully developed periodic regime, a vortex shedding (von Kármán vortex street) can be observed behind the obstacle, see Fig. 5.2.

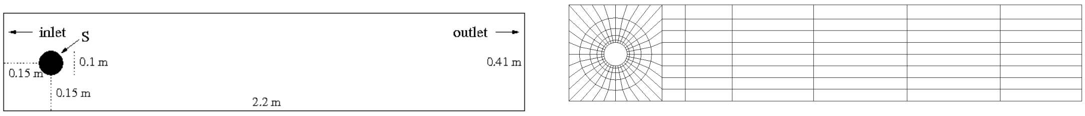

Figure 5.1.: Example 5.1: The flow domain (left) and the coarse grid (right).

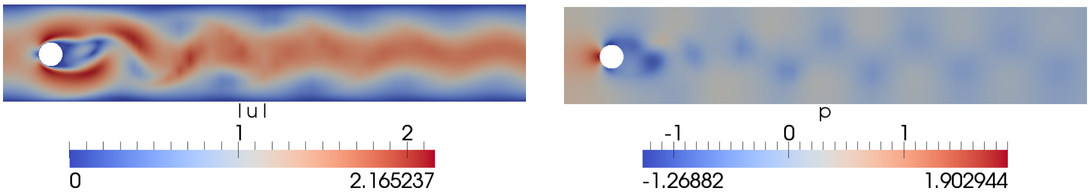

Figure 5.2.: Example 5.1: Snapshots of the finite element solution.

Quantities of interest are the drag and lift coefficients at the cylinder. In the presented numerical studies, these quantities were computed as volume integrals by the following expressions

$$c_{\mathrm{d}} = -\frac{2}{LU^2}\Big[(\partial_t\boldsymbol{u},\boldsymbol{v}_{\mathrm{d}}) + (\nu\nabla\boldsymbol{u},\nabla\boldsymbol{v}_{\mathrm{d}}) + ((\boldsymbol{u}\cdot\nabla)\boldsymbol{u},\boldsymbol{v}_{\mathrm{d}}) - (\nabla\cdot\boldsymbol{v}_{\mathrm{d}},p)\Big], \tag{5.45}$$

$$c_{\mathrm{l}} = -\frac{2}{LU^2}\Big[(\partial_t\boldsymbol{u},\boldsymbol{v}_{\mathrm{l}}) + (\nu\nabla\boldsymbol{u},\nabla\boldsymbol{v}_{\mathrm{l}}) + ((\boldsymbol{u}\cdot\nabla)\boldsymbol{u},\boldsymbol{v}_{\mathrm{l}}) - (\nabla\cdot\boldsymbol{v}_{\mathrm{l}},p)\Big], \tag{5.46}$$

for functions $\boldsymbol{v}_{\mathrm{d}}$, $\boldsymbol{v}_{\mathrm{l}} \in H^1(\Omega)$ such that $\boldsymbol{v}_{\mathrm{d}} = (1,0)^T$ and $\boldsymbol{v}_{\mathrm{l}} = (0,1)^T$ on the boundary of the cylinder and $\boldsymbol{v}_{\mathrm{d}} = \boldsymbol{v}_{\mathrm{l}} = (0,0)^T$ on all other boundaries. Since $\boldsymbol{v}_{\mathrm{d}}$ and $\boldsymbol{v}_{\mathrm{l}}$ are not discretely divergence-free, the last term in (5.45) and (5.46) does not vanish and the pressure is needed for computing the drag and the lift coefficients. One can compute these quantities by boundary integrals defined on the cylinder. The definitions, however, can be reformulated as integrals on the domain Ω, which is numerically advantageous, see [77] and the discussion therein. The latter approach, extended to the time-dependent Navier–Stokes equations, see [74], was used in the present numerical studies. In the periodic regime, another important quantity of interest is the Strouhal number

$$St = \frac{L\nu_f}{U},$$

which is correlated to the frequency of the vortex shedding ν_f. According to [122], the reference intervals for the functionals of interest are

$$\begin{aligned} &\text{Maximum drag coefficient:} \qquad && c_{\mathrm{d}}^{\max} \in [3.22, 3.24]\,, \\ &\text{Maximum lift coefficient:} && c_{\mathrm{l}}^{\max} \in [0.98, 1.02]\,, \\ &\text{Strouhal number:} && St \in [0.295, 0.305]\,. \end{aligned}$$

To the best of the author's knowledge, there is no known relation between the kinetic energy or the L^2 norm, which was the criterion used to compute the POD basis, and these quantities of interest.

All simulations were performed with the code MooNMD [78] on a grid obtained by three uniform red refinements of the coarse grid presented in Fig. 5.1, where the resolution of the cylinder was improved with each refinement. The Navier–Stokes equations (2.40)-(2.41) were discretized in space with the inf-sup stable Taylor–Hood Q_2/Q_1 finite elements, resulting in 107 712 velocity degrees of freedom and 13 616 pressure degrees of freedom.

Numerical Methods for Computing the Snapshots

One of the goals of this section is to numerically investigate the effect of the snapshot accuracy on the results obtained with the vp-ROMs. Different numerical methods on the same grids in time and space were employed for computing snapshots of different accuracies.

The most expensive numerical method, denoted by SP-NONLIN, requires the solution of a nonlinear saddle point problem at each discrete time. The nonlinear problem is solved by a fixed point iteration (Picard iteration), as described in Section 2.2.5. The second numerical method, denoted by SP-LIN, uses the IMEX version of the Crank–Nicolson scheme, see Section 2.2.5 for more details. Thus, the convective term is discretized explicitly in the convective component $((\boldsymbol{u}_h^{n-1} \cdot \nabla)\boldsymbol{u}_h^n, \boldsymbol{v}_h)$ and all other terms are handled implicitly. SP-LIN yields one linear saddle point problem at each time iteration. For both methods, the Crank–Nicolson time integration scheme with the time step $\Delta t = 0.005$ was employed, which showed, among simple time stepping schemes, a good balance between accuracy and computational efficiency [79, 83]. Finally, the third numerical method is the standard second order incremental pressure-correction scheme (2.100), denoted by PC, which employs the IMEX method and removes even the saddle point character of the problem. The method is similar to the van Kan scheme [53, 139], however it utilizes the BDF2 time stepping scheme. For the numerical simulations in this section the same time step $\Delta t = 0.005$ as for SP-NONLIN and SP-LIN will be used. At each discrete time, PC requires only the solution of one linear equation for the velocity, where the equations for the velocity components are decoupled, and one linear equation for the pressure. PC provides two approximations for the velocity. For the underlying numerical studies, the non-incompressible velocity approximation, which satisfies the correct boundary conditions, is used (see Remark 2.11 for the motivation of the choice).

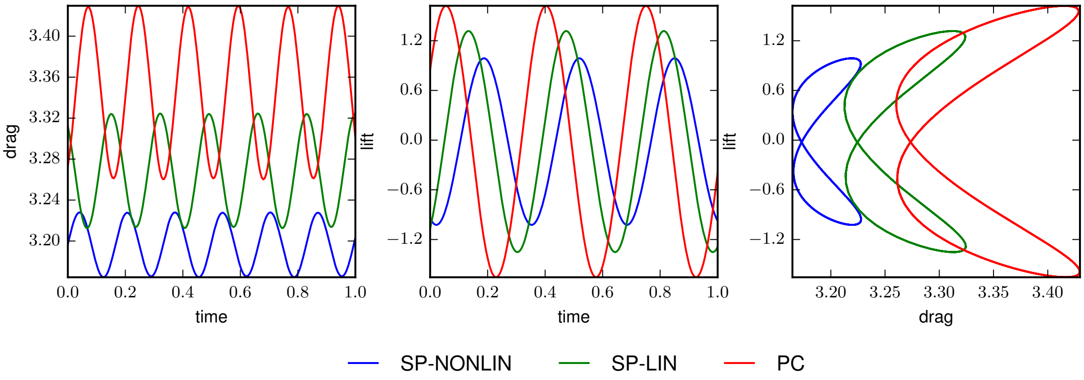

Figure 5.3.: Example 5.1: Drag and lift coefficients for the finite element simulations.

Clearly, the three different numerical methods possess different numerical costs. In the simulations for computing the snapshots, SP-NONLIN took about 2.6 times longer than SP-LIN, and SP-LIN took about 2.2 longer than PC. But it can be also expected that the three methods exhibit differences in accuracy. This expectation is met by the results presented in Fig. 5.3 and Table 5.2. One can observe that SP-NONLIN, the numerical method with the highest computational price, is also the most accurate one, as the results for all reference values are within the reference intervals given in Table 5.2. The accuracy deteriorates for SP-LIN and for PC, but one can see that the results of SP-LIN are still considerably more accurate than the results computed with PC. Accordingly, three sets of snapshots were obtained: of the highest accuracy, of intermediate accuracy, and of the lowest accuracy.

Table 5.2.: Example 5.1: Maximum drag coefficient, maximum lift coefficient, and Strouhal number for the finite element simulations.

	$c_{\mathrm{d}}^{\max}$	$c_{\mathrm{l}}^{\max}$	St
SP-NONLIN	3.23	0.99	0.301
SP-LIN	3.32	1.32	0.295
PC	3.43	1.62	0.287
reference results from [122]	$[3.22, 3.24]$	$[0.98, 1.02]$	$[0.295, 0.305]$

Impact of the Snapshot Accuracy on the POD Modes

Here, the influence of using numerical methods of different accuracies on the velocity and the pressure POD bases will be monitored.

From the simulations with SP-NONLIN, SP-LIN, and PC, after having collected snapshots over the time interval $[0, 2]$ for each discrete time, three different POD bases for the velocity and the pressure fields, respectively, were generated from the fluctuating part of the snapshots (see Section 3.2.3 for more details) by the method of snapshots described

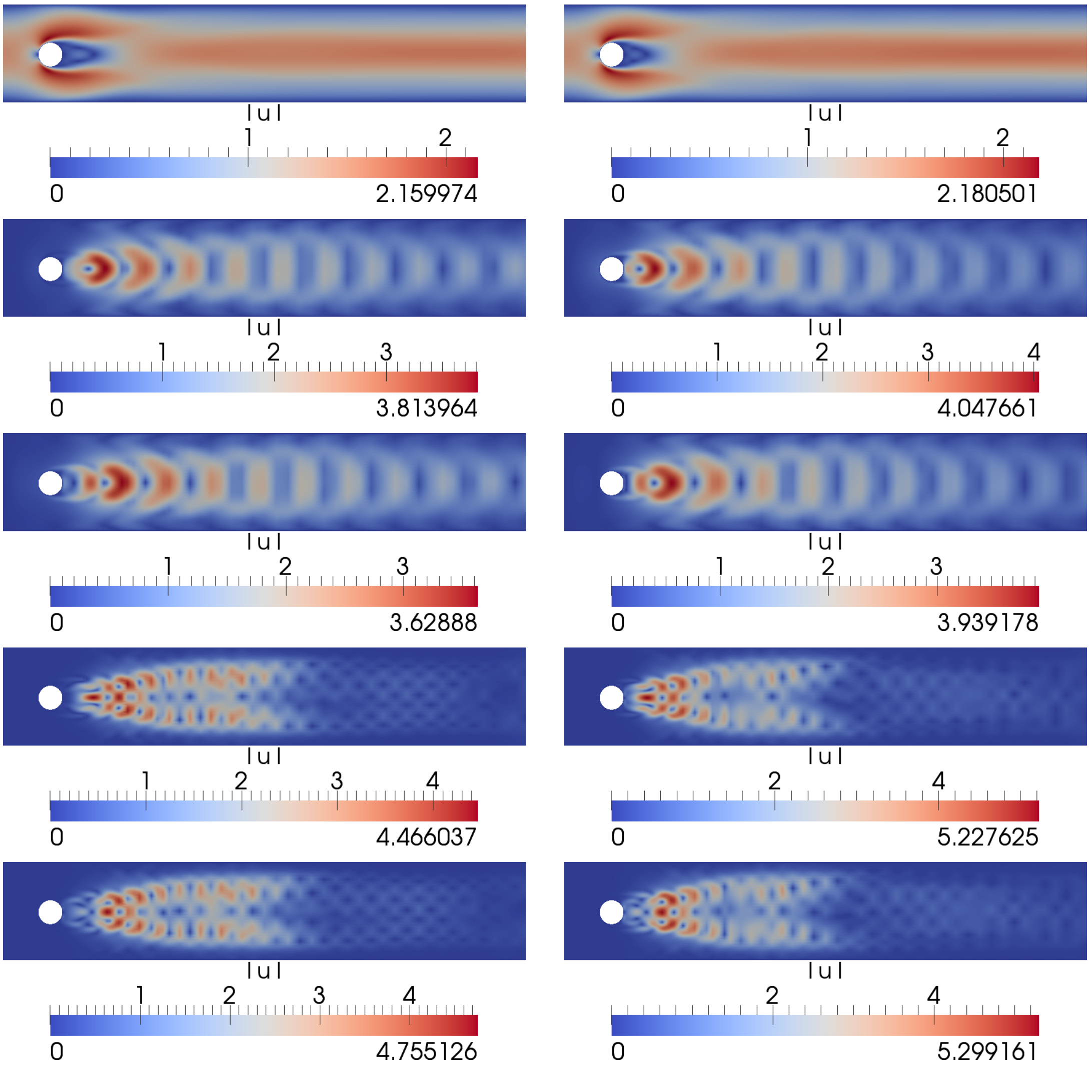

Figure 5.4.: Example 5.1: Norm of the velocity snapshots' mean (top) and the first velocity POD modes: POD basis computed from SP-NONLIN (left) and PC (right).

in Section 3.1.3. Moreover, the velocity and the pressure POD bases were computed with respect to the $L^2(\Omega)$ inner product, which is a popular choice in the literature. Figs. 5.4 and 5.5 display the norm of the snapshots' mean and of the first velocity and pressure POD modes, respectively. For clarity of presentation, only the most accurate (SP-NONLIN) and the lowest accurate (PC) numerical methods are considered.

Both Fig. 5.4 and Fig. 5.5 show that, although structurally similar, the maximum and minimum values are quite different for the two numerical methods.

Next, the POD bases are investigated in terms of the POD eigenvalues $\{\lambda_i\}$, defined in (3.41), and the missing energy ratio equal to $1 - \mathcal{E}(r)$ with $\mathcal{E}(r)$ defined by (3.52).

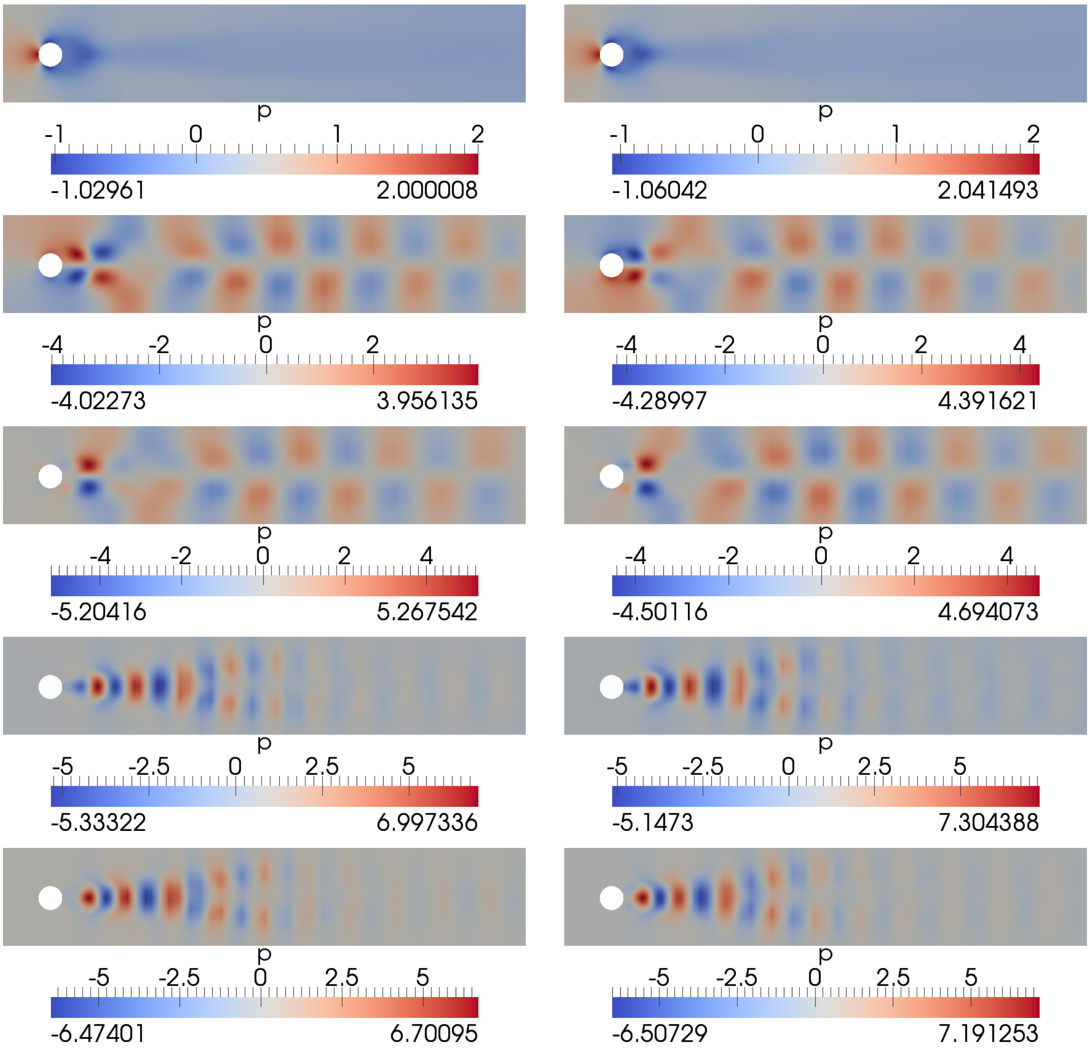

Figure 5.5.: Example 5.1: Pressure snapshots' mean (top) and the first pressure POD modes: POD basis computed from SP-NONLIN (left) and PC (right).

Figure 5.6 shows $\{\lambda_i\}$ and the missing energy ratio for the velocity and pressure POD bases for the three sets of snapshots. It can be observed that all sets of snapshots lead to a similar number of non-zero POD eigenvalues.

Figure 5.6 also shows that there are steep decreases in the eigenvalues of the velocity POD modes, e.g., after the second and the sixth mode. Similar jumps can be seen in the eigenvalues of the pressure POD modes after the second, fourth, and eighth mode. Correspondingly, there are strong decreases in the missing energy ratio. It is interesting to note that the velocity and pressure jumps in the eigenvalues and the missing energy ratio seem not to be correlated. This observation supports the point of view that using a different number of velocity and pressure POD modes might be advantageous. The

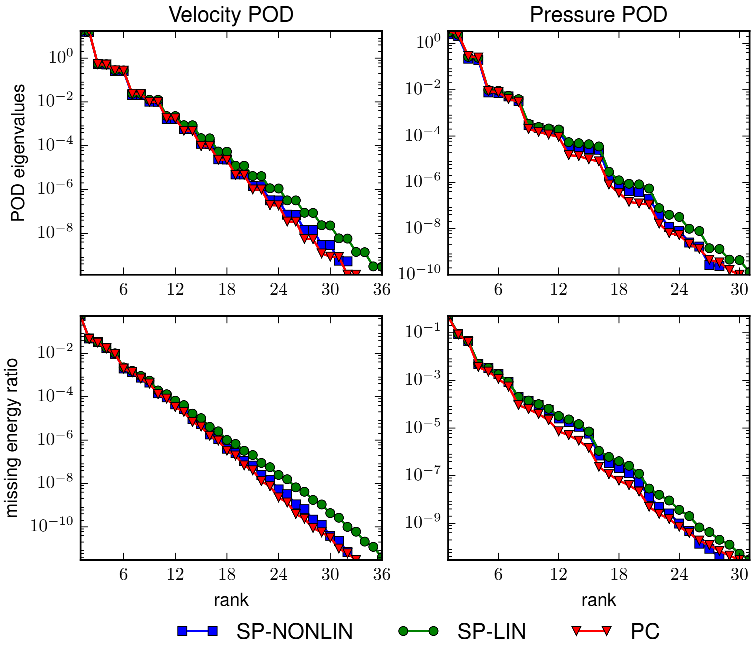

Figure 5.6.: Example 5.1: POD eigenvalues and missing energy ratio.

study of this issue, however, is outside the scope of the thesis and will not be further pursued herein.

Assessment of the vp-ROMs

This subsection presents an assessment of the effect of the snapshot accuracy on the three vp-ROMs introduced in Section 5.4, see Table 5.1. Within the framework of the present numerical studies, the same number of POD modes for the velocity and the pressure fileds is used, i.e., $r_v = r_p = r$.

Theoretical error estimates in [67], see also [89, 104], show that the total error in the numerical discretization of velocity-type ROMs consists of three parts: the spatial error due to the finite element discretization, the temporal error due to the time-stepping scheme, and the POD error due to the POD truncation. In the present numerical investigations of vp-ROMs, however, the spatial and temporal error components are constant, since the mesh size and the time step are fixed. Thus, assuming that the ROM estimates in [67, 89, 104] can be extended to vp-ROMs, for increasing values of r, one expects the POD error component of the vp-ROMs to initially decrease, but then to reach a plateau where the POD error component has the same or a lower magnitude than the spatial and temporal error components.

The v-ROM (5.10) of all three vp-ROMs investigated in this section does not include the pressure term $-(p_{\text{ro}}, \nabla \cdot \boldsymbol{\phi}_{\text{ro},i})$. When the POD modes were computed by solving a saddle point problem (SP-NONLIN, SP-LIN), the motivation for it was discussed in Section 5.3: since the velocity snapshots are discretely divergence-free, the term $-(p_{\text{ro}}, \nabla \cdot \boldsymbol{\phi}_{\text{ro},i})$ vanishes. In the case of PC, when the snapshots are obtained from a non divergence-free velocity field, this argument does not hold. The impact of adding the pressure term to the vp-ROMs was numerically tested but it did not yield any improvement of the results.

An essential motivation for developing ROMs is computational efficiency. For this reason, one usually prefers to avoid complex and time-consuming numerical methods in combination with the ROM, see, e.g., [130, 143]. For the numerical studies in this section, the Crank–Nicolson scheme ($\theta = \frac{1}{2}$, see Table 2.1) with the time step $\Delta t = 0.005$ is utilized for the time discretization of the velocity ROMs. In order to investigate the influence of the utilized nonlinear iteration scheme on the vp-ROM results, three different methods for the linearization of the velocity ROMs will be employed: the Picard method, the IMEX, and the IMEX-LE schemes (see Section 2.2.5). The Picard iteration is the most time-consuming method as it usually involves several nonlinear iterations, which requires solving a linear system in each iteration. The latter two methods are more efficient as they involve only one solution of a linear system at each time iteration. However, the IMEX-LE scheme performs in the framework of the finite element method significantly better than the IMEX scheme, see [75]. For the underlying test problem with the same time and space discretization as used for the computation of the snapshots SP-NONLIN and SP-LIN, the following measures of interest were obtained with the IMEX-LE scheme: $c_{\text{d}}^{\max} = 3.24$, $c_{\text{l}}^{\max} = 1.01$, $St = 0.30$. All three values lie within the reference intervals shown in Table 5.2. To distinguish the vp-ROMs with respect to the employed nonlinear iterations methods, the notations v-ROM($*$), vp-ROM($*$), PMB-ROM($*$), SM-ROM($*$), and VMB-ROM($*$) will be utilized, where $*$ stands for Picard, IMEX, and IMEX-LE.

To assess the behavior of the vp-ROM results, several quantities of interest will be evaluated for each set of snapshots. As there exists no analytical representation of the continuous equations (2.40)-(2.41) for the underlying problem, the ROM results will be compared with the corresponding finite element results. Firstly, the impact of the involved nonlinear iteration scheme on the accuracy of the ROM velocity will be investigated. For this sake, the mean kinetic energy error defined by

$$\overline{\mathcal{E}}_{\text{kin}} = \frac{1}{M} \sum_{m=1}^{M} \left| \frac{1}{2} \|\boldsymbol{u}_h^m\|_0^2 - \frac{1}{2} \|\boldsymbol{u}_{\text{ro}}^m\|_0^2 \right| \tag{5.47}$$

or the discretized version of the $L^1(0, T; L^2(\Omega))$ error

$$\overline{\mathcal{E}}_{\text{ROM}} = \frac{1}{M} \sum_{m=1}^{M} \|\boldsymbol{u}_h^m - \boldsymbol{u}_{\text{ro}}^m\|_0 \tag{5.48}$$

can be considered. As both measures yielded qualitatively similar results, only the former error will be discussed in the following part. For periodic flows, it is not meaningful to investigate the errors of type (5.47), which are based on the comparison of two solutions at each time step, if the frequencies with the largest amplitudes do not coincide. The verification of this fact can be achieved by performing, e.g., the Fast Fourier Transform (FFT) for one of the velocity components evaluated at a point behind the cylinder. For this purpose, the y-component of the velocity field at every time instance t_m, $m = 1, \ldots, M$, will be evaluated at (0.5, 0.2).

Secondly, the time evolution of the drag and lift coefficients computed by the vp-ROMs combined with the three nonlinear iteration schemes for different values of r will be monitored. The root mean square (rms) value of the local maxima of the drag and lift coefficients denoted by $c_{\mathrm{d,rms}}^{\max}$ and $c_{\mathrm{l,rms}}^{\max}$, respectively, will be computed in order to figure out how periodic the coefficients are in terms of the largest amplitudes. This investigation is of great importance if the maximum of the drag and lift coefficients $c_{\mathrm{d}}^{\max}$ and $c_{\mathrm{l}}^{\max}$ are the measures of interest for the vp-ROM results or if the long-time behavior of the flow is of interest. In both cases it is desired that $c_{\mathrm{d,rms}}^{\max}$ and $c_{\mathrm{l,rms}}^{\max}$ are as small as possible, e.g., all local maxima of the coefficients do not differ much.

Thirdly, the error in the Strouhal number, the errors in the mean values of the drag and lift coefficients $\overline{c}_{\mathrm{d}}$ and $\overline{c}_{\mathrm{l}}$, and the errors in the rms values of the drag and lift coefficients defined by

$$c_{\mathrm{d,rms}} = \left[\frac{1}{M}\sum_{m=1}^{M}(\overline{c}_{\mathrm{d}} - c_{\mathrm{d}}(t_m))^2\right]^{1/2}, \quad c_{\mathrm{l,rms}} = \left[\frac{1}{M}\sum_{m=1}^{M}(\overline{c}_{\mathrm{l}} - c_{\mathrm{l}}(t_m))^2\right]^{1/2},$$

will be investigated. Here, c_{d} (or c_{l}) denotes the drag (or lift) coefficient computed with a certain numerical method (finite element method or vp-ROM). The rms values provide information on the magnitude of the oscillations around the mean value. The errors are defined as the absolute values of the difference of the quantity of interest resulted from the vp-ROM simulation and the finite element simulation, which was used for computing the snapshots. The frequency of the vortex shedding, needed for the computation of the Strouhal number, was computed using the inverse of the average period of the lift coefficients.

All simulations were performed in the time interval $[0, 2]$ and the reference values were computed over five periods for the lift coefficient (5.46).

vp-ROMs using snapshots of the highest accuracy. The numerical results for the vp-ROMs using the snapshots from SP-NONLIN are presented in this part of the section. The top row of Fig. 5.7 shows the time evolution of the y-component of the velocity fields computed by SP-NONLIN and the v-ROMs combined with the Picard, IMEX, and IMEX-LE scheme evaluated at $(0.5, 0.2)$ for $r \in \{3, 6, 24\}$. One can assert almost no difference between the v-ROM results obtained using different nonlinear iteration methods. For $r = 3$, all three v-ROMs fail to reproduce the shape of the velocity component from the underlying finite element method SP-NONLIN. Already 6 POD modes are sufficient to obtain a rather good approximation of the y-component of the velocity solution obtained by SP-NONLIN. When using $r = 24$, all curves coincide. In the bottom row of Fig. 5.7 the corresponding frequency-based representation of the y-component of the velocity fields at $(0.5, 0.2)$ computed by the Fast Fourier Transform is depicted. From the plot, one can conclude that for $r = 3$ all v-ROMs do not manage

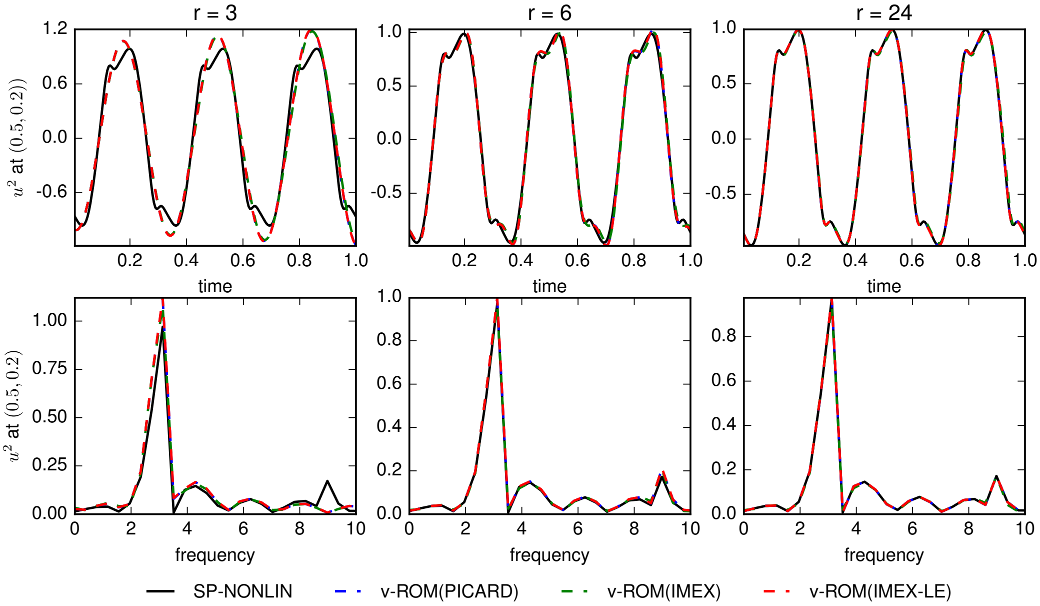

Figure 5.7.: Example 5.1, snapshots of the highest accuracy (SP-NONLIN): Time-based (top) and frequency-based (bottom) representation of the y-component of the velocity field evaluated at $(0.5, 0.2)$.

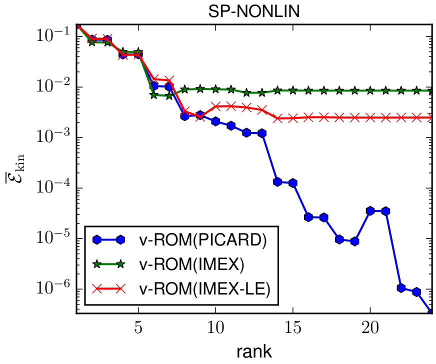

Figure 5.8.: Example 5.1, snapshots of the highest accuracy (SP-NONLIN): Mean kinetic energy error (5.47) for different POD dimensions r computed from the velocity field obtained by the v-ROMs combined with three different nonlinear iteration schemes.

to replicate the correct amplitude of the frequency value of 9. Beginning with $r = 6$, all frequencies with the largest amplitudes coincide. Due to this finding, the investigation of the mean kinetic energy $\overline{\mathcal{E}}_{\text{kin}}$ defined by (5.47) is meaningful. In Fig. 5.8, the error is shown for the v-ROMs combined with the three nonlinear iteration methods for the POD dimensions $r = 1, \ldots, 24$. In contrast to Fig. 5.7, one can clearly see that the three curves behave differently. If the Picard method is employed in the v-ROM, which corresponds to the same nonlinear iteration scheme used for the computation of the underlying snapshots, the graph falls as expected when the POD dimension increases (with the exception of $r = 20$ and $r = 21$). One can reach any order of precision if only the POD dimension is chosen large enough. In the case of other two linearization schemes, for small values of r the curves fall but then they reach a certain plateau at $r = 6$ for the IMEX and at $r = 8$ for the IMEX-LE methods. However, v-ROM(IMEX-LE) for larger POD dimensions produces a mean kinetic energy error which is almost one magnitude smaller than the one obtained with v-ROM(IMEX).

The time evolution of the drag and lift coefficients obtained by the vp-ROMs from Table 5.1 for three different POD dimensions together with the Picard, IMEX, and IMEX-LE linearization techniques is displayed in Fig. 5.9. There is no obviously visible difference between the behavior of the coefficients with respect to the utilized nonlinear iteration method. It can be observed that PMB-ROM and SM-ROM are able to reproduce the results of the underlying finite element simulation for the snapshots very well with already $r = 6$ POD modes. In contrast, the range of the drag coefficient computed with VMB-ROM is not correct, even for $r = 24$ POD modes. With respect to the lift coefficient, the results with the VMB-ROM are much better. In more detail, clear improvements in the quality of the reproduction can be seen for all three vp-ROMs when going from $r = 5$ to $r = 6$, which corresponds to a jump in the missing energy ratio of the velocity POD modes, see Fig. 5.6. For values $r \geq 6$ both PMB-ROM and SM-ROM yield drag coefficients within the reference intervals given in Table 5.2. For VMB-ROM, the size of the amplitude of the drag coefficient improved for $r = 6$ modes, but not the mean value of the drag. Even increasing the number of modes to $r = 24$, the mean value of the coefficient stays considerably below the reference. A closer look at the presentations of the drag in Fig. 5.9 reveals that also its time evolution is not fully periodic since the values of the peaks are changing visibly, which is another shortcoming of the method. Considering the lift coefficient, all three vp-ROMs perform well for $r \geq 6$ independently of the used linearization scheme.

To explain the inaccurate drag coefficient obtained with the VMB-ROM, note that the drag coefficient depends mainly on the pressure at the cylinder. In the simulations with VMB-ROM, the main contribution to the ROM pressure is $p_{00}(\boldsymbol{x})$, whereas the main part of the ROM pressure for PMB-ROM and SM-ROM is $\overline{p}_h(\boldsymbol{x})$. Both functions are depicted in Fig. 5.10. A closer look at the plots reveals that the pressure difference between the back and the front of the cylinder is somewhat smaller for $p_{00}(\boldsymbol{x})$ than for $\overline{p}_h(\boldsymbol{x})$, which results in inaccurate drag forces. The same behavior could also be observed for the other two snapshots sets SP-LIN and PC. Since the mean drag is often very important in applications, this result shows a considerable shortcoming of this method.

To better assess the periodic behavior of vp-ROMs, Fig. 5.11 displays the root mean square values of the local maxima of the drag and lift coefficients for all vp-ROMs. The less the rms value, the less differ the peaks of the coefficients resulting in a more periodic behavior. Immediately, one takes note of the fact that the smallest rms values are achieved with the combination of vp-ROMs with the Picard method, i.e., the same linearization scheme as employed for the computation of the underlying snapshots. For

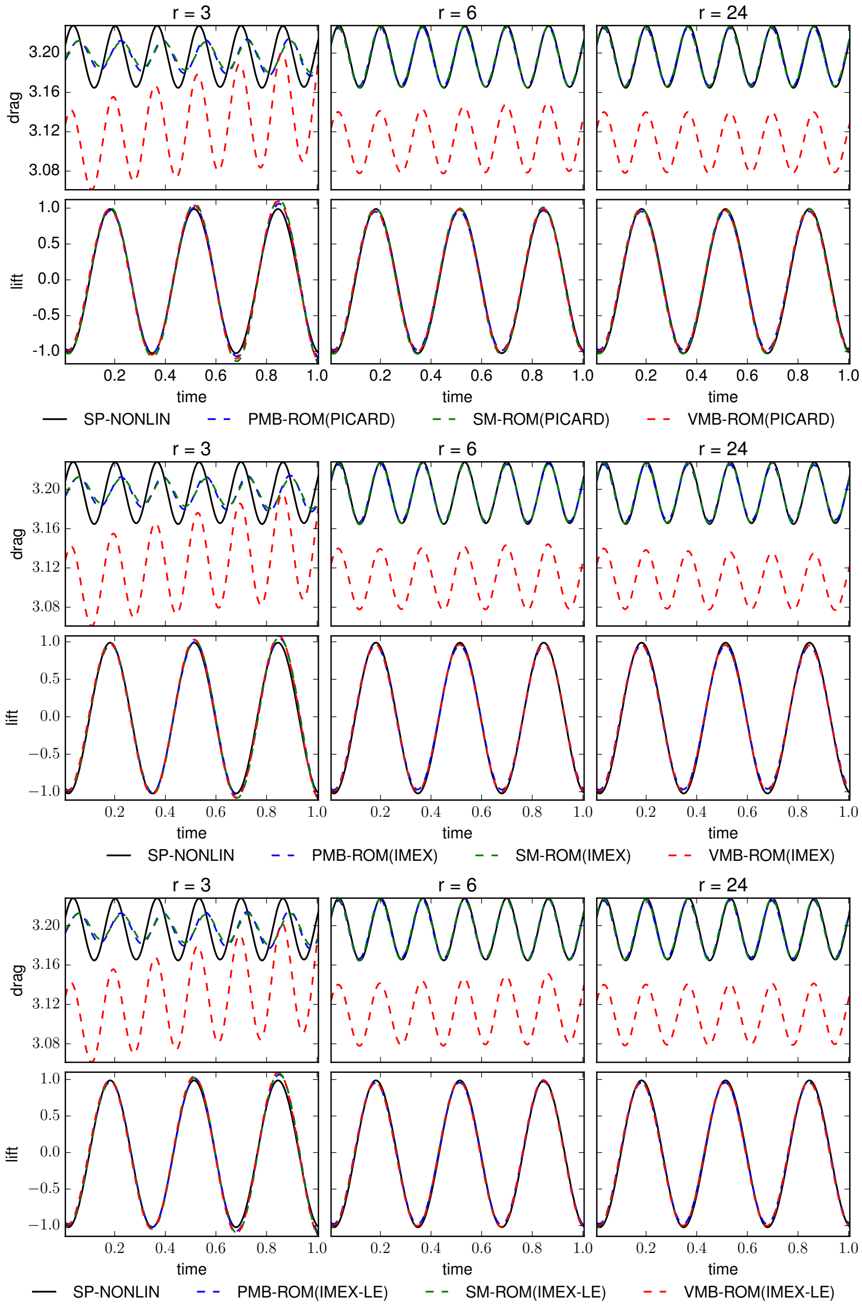

Figure 5.9.: Example 5.1, snapshots of the highest accuracy (SP-NONLIN): Time evolution of the drag and lift coefficients computed by the vp-ROMs combined with the Picard, IMEX and IMEX-LE nonlinear iteration schemes (top to bottom) for $r = 3, 6, 24$ (from left to right).

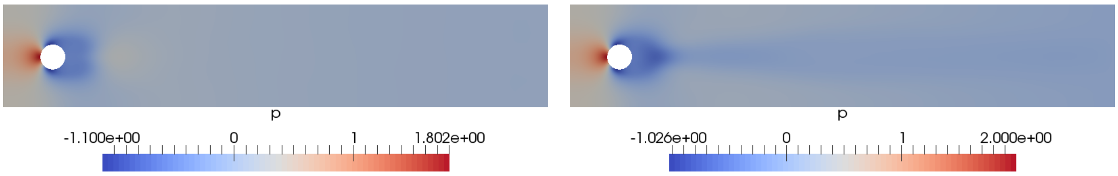

Figure 5.10.: Example 5.1, snapshots of the highest accuracy (SP-NONLIN): Pressure coefficient $p_{00}(\boldsymbol{x})$ computed with VMB-ROM (left), pressure snapshots' average $\overline{p}_h(\boldsymbol{x})$ for PMB-ROM and SM-ROM (right).

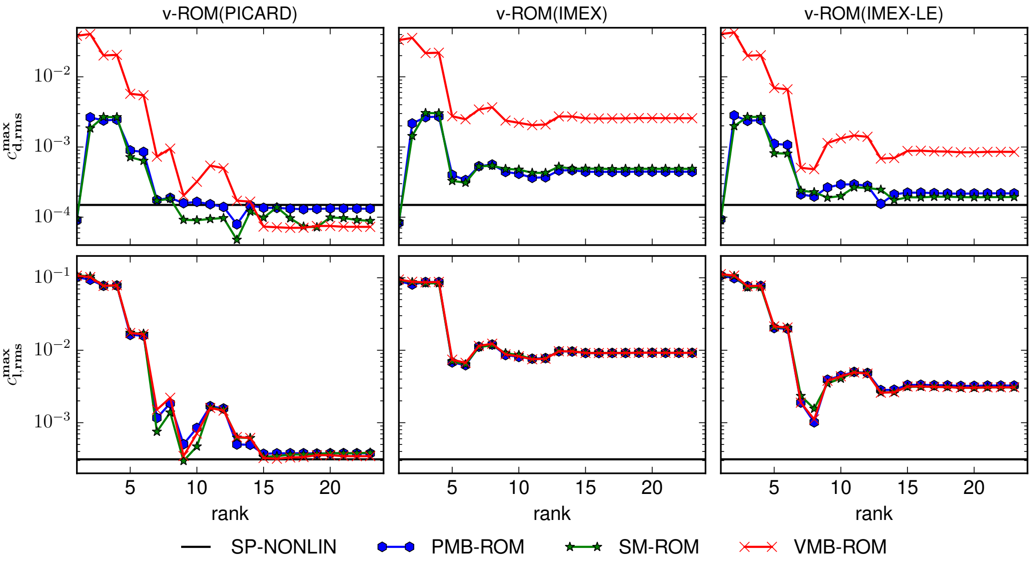

Figure 5.11.: Example 5.1, snapshots of the highest accuracy (SP-NONLIN): $c^{\max}_{\text{d,rms}}$ (top) and $c^{\max}_{\text{l,rms}}$ (bottom) for the finite element simulation SP-NONLIN, the three vp-ROMs combined with the Picard (left), IMEX (center), and IMEX-LE (right) nonlinear iteration schemes.

$r \geq 6$ with PMB-ROM and SM-ROM, and for $r \geq 13$ with VMB-ROM one gets the drag coefficient that is at least as periodic as the one obtained with SP-NONLIN. With respect to the lift coefficient, the three vp-ROMs achieve the accuracy of the finite element simulation for $r \geq 15$. By employing the IMEX scheme for the computation of the ROM velocity, all vp-ROMs yield coefficients, whose peaks vary most of all. With the IMEX-LE method, the PMB-ROM and SM-ROM almost achieve the rms value of the drag coefficient obtained with SP-NONLIN. The VMB-ROM performs worse than the other two vp-ROMs in terms of the drag coefficient, whereas all vp-ROMs perform comparably well for the lift coefficient.

Another way to investigate the quality of the vp-ROM results consists in the investigation of their errors with respect to the values of SP-NONLIN in the Strouhal number, in the mean drag, in the mean lift, in the drag rms, and in the lift rms values. These errors

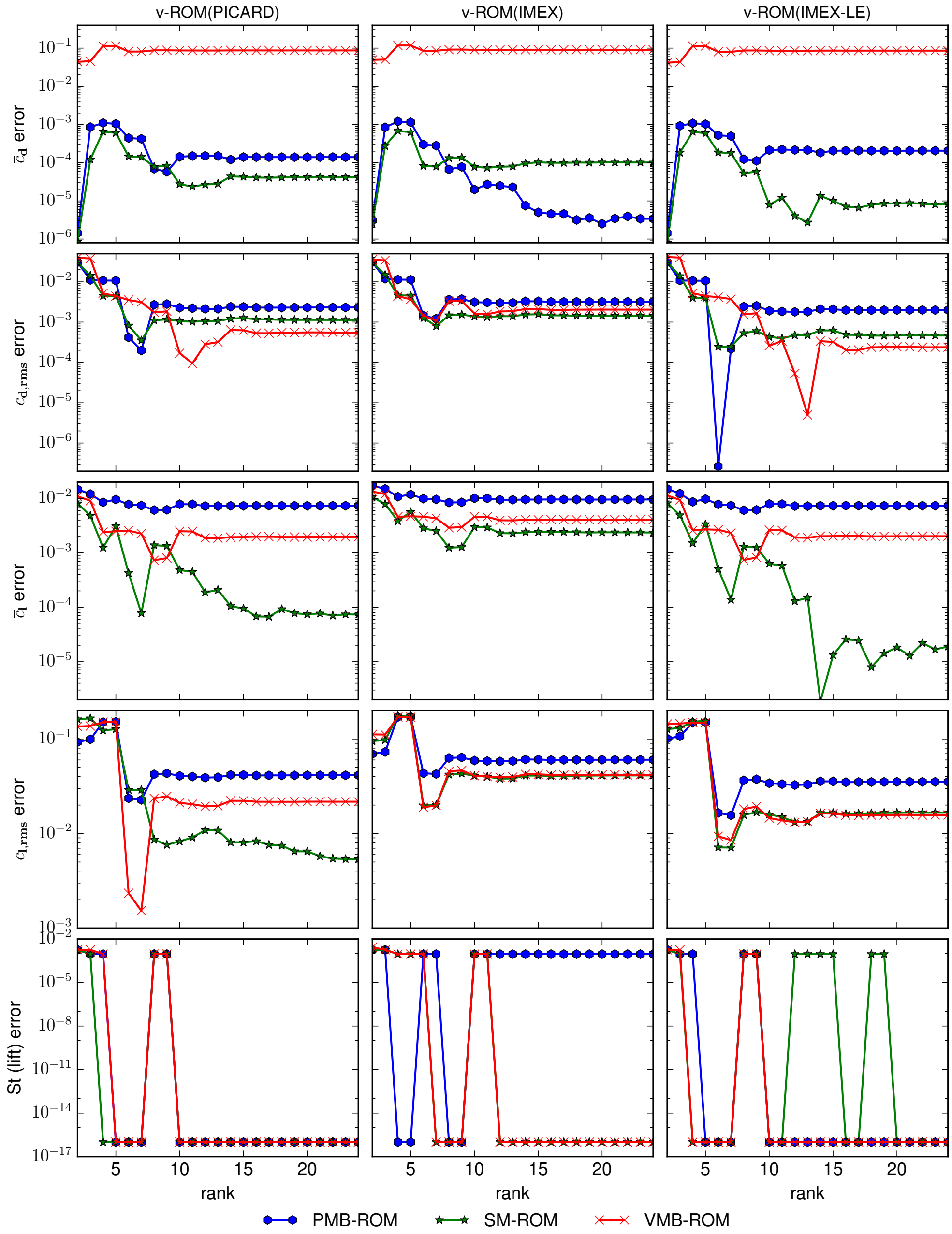

Figure 5.12.: Example 5.1, snapshots of the highest accuracy (SP-NONLIN): Drag mean, drag rms, lift mean, lift rms, and Strouhal number errors (from top to bottom) obtained by the vp-ROMs combined with the Picard, IMEX and IMEX-LE nonlinear iteration schemes (left to right).

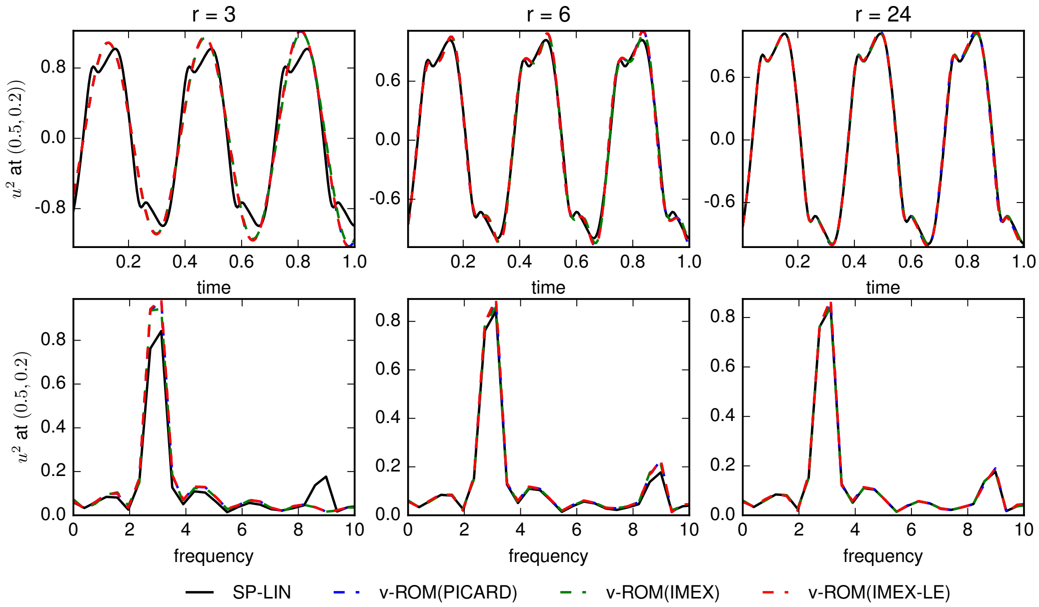

Figure 5.13.: Example 5.1, snapshots of intermediate accuracy (SP-LIN): Time-based (top) and frequency-based (bottom) representation of the y-component of the velocity field evaluated at $(0.5, 0.2)$.

are presented in Fig. 5.12. In contrast to Figs. 5.8 and 5.11, one cannot observe any obvious relation between the results with the linearization scheme employed in the finite element simulation for the computation of the snapshots and the vp-ROMs. As already seen in Fig. 5.9, VMB-ROM fails to reproduce the correct drag mean value. In terms of the other measures of interest, its performance is somewhat better than that of PMB-ROM and comparable or worse than that of SM-ROM. Moreover, it can be seen that SM-ROM yields often smaller errors than PMB-ROM. For $r \geq 10$, PMB-ROM(IMEX) fails to reproduce the Strouhal number as good as the other vp-ROMs do. The lengths of the period differ by about one time step Δt. The same behavior can be asserted for SM-ROM(IMEX-LE) for some values of r with $r \geq 12$.

vp-ROMs using snapshots of intermediate accuracy. After having studied the numerical results of the vp-ROMs based on the SP-NONLIN snapshots, the same measures of interest will be investigated here for the vp-ROMs built with respect to the SP-LIN snapshots.

Figure 5.13 displays the time evolution and the frequency-based representation of the y-component of the velocity field computed by SP-LIN serving as the reference and the v-ROM utilized with three different nonlinear iteration methods. Similarly to the corresponding plots for SP-NONLIN from Fig. 5.7, the finite element results can be reproduced quite well already with $r = 6$ POD modes: the frequencies with the largest amplitudes coincide for all tested ROMs and SP-LIN. Hence, the consideration of the

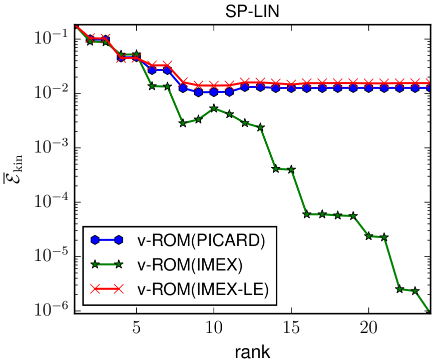

Figure 5.14.: Example 5.1, snapshots of intermediate accuracy (SP-LIN): Mean kinetic energy error (5.47) for different POD dimensions r computed from the velocity field obtained by the v-ROMs combined with three different nonlinear iteration schemes.

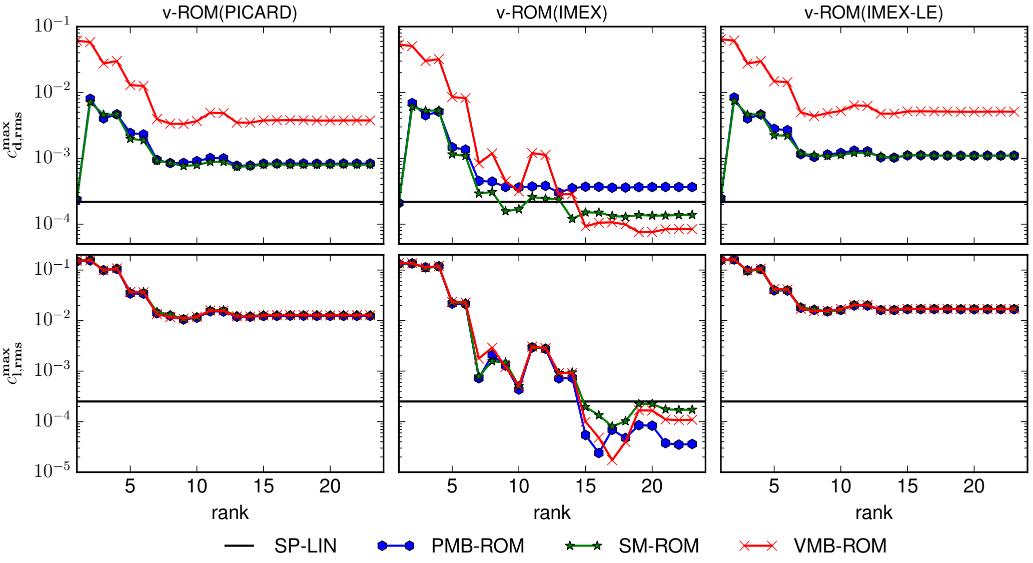

Figure 5.15.: Example 5.1, snapshots of intermediate accuracy (SP-LIN): $c_{\text{d,rms}}^{\max}$ (top) and $c_{\text{l,rms}}^{\max}$ (bottom) for the finite element simulation SP-LIN, the three vp-ROMs combined with the Picard (left), IMEX (center), and IMEX-LE (right) nonlinear iteration schemes.

mean kinetic energy error $\mathcal{E}_{\text{kin}}$, presented in Fig. 5.14, is meaningful. It can be asserted that the error for v-ROM(IMEX) falls with the increasing POD dimension, whereas the v-ROMs with the Picard and IMEX-LE schemes fall for small values of r, and for $r \geq 7$ both curves reach a plateau at $\overline{\mathcal{E}}_{\text{kin}} \approx 10^{-2}$. This fact supports the presumption that the mean kinetic energy error as well as the discretized version of the $L^1(0,T;L^2(\Omega))$ error

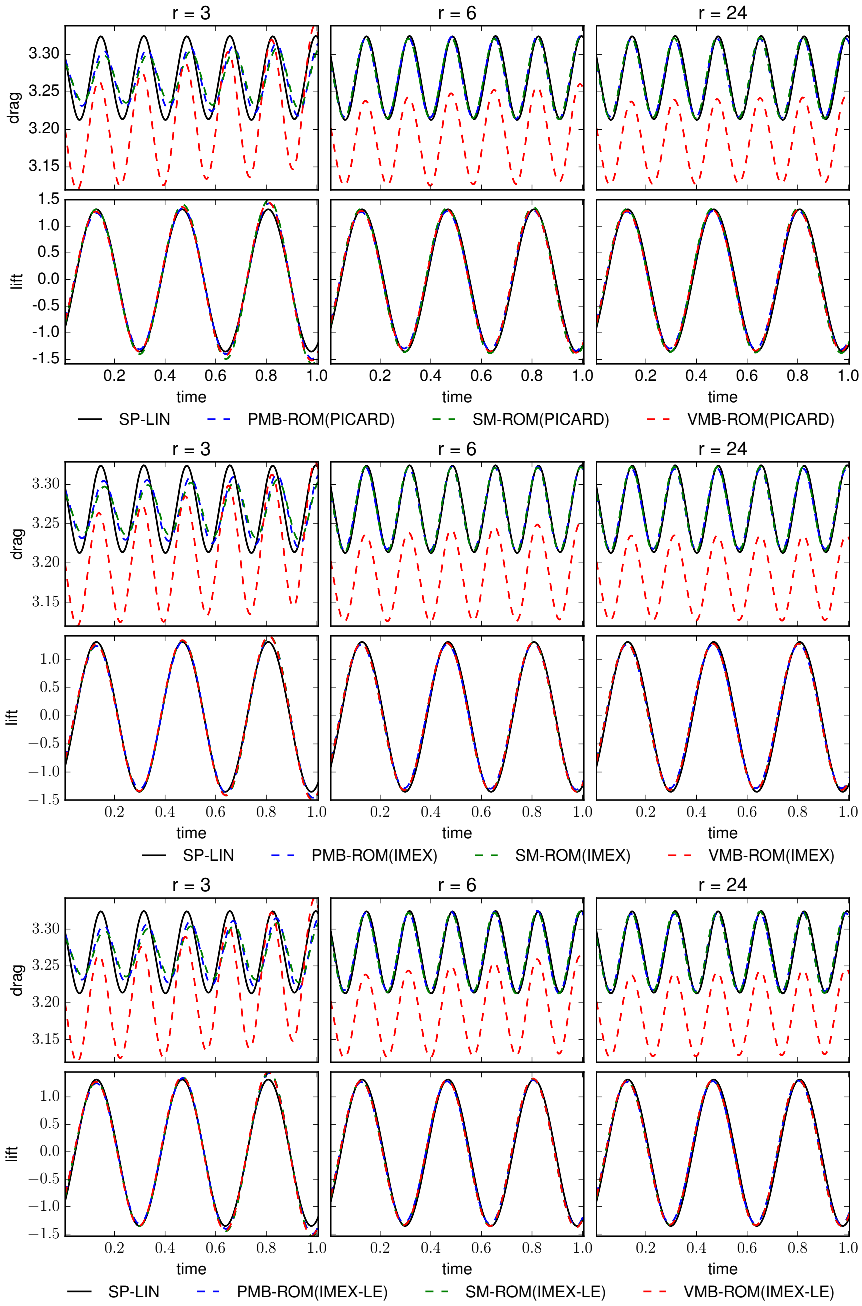

Figure 5.16.: Example 5.1, snapshots of intermediate accuracy (SP-LIN): Time evolution of the drag and lift coefficients computed by the vp-ROMs combined with the Picard, IMEX and IMEX-LE nonlinear iteration schemes (top to bottom) for $r = 3, 6, 24$ (from left to right).

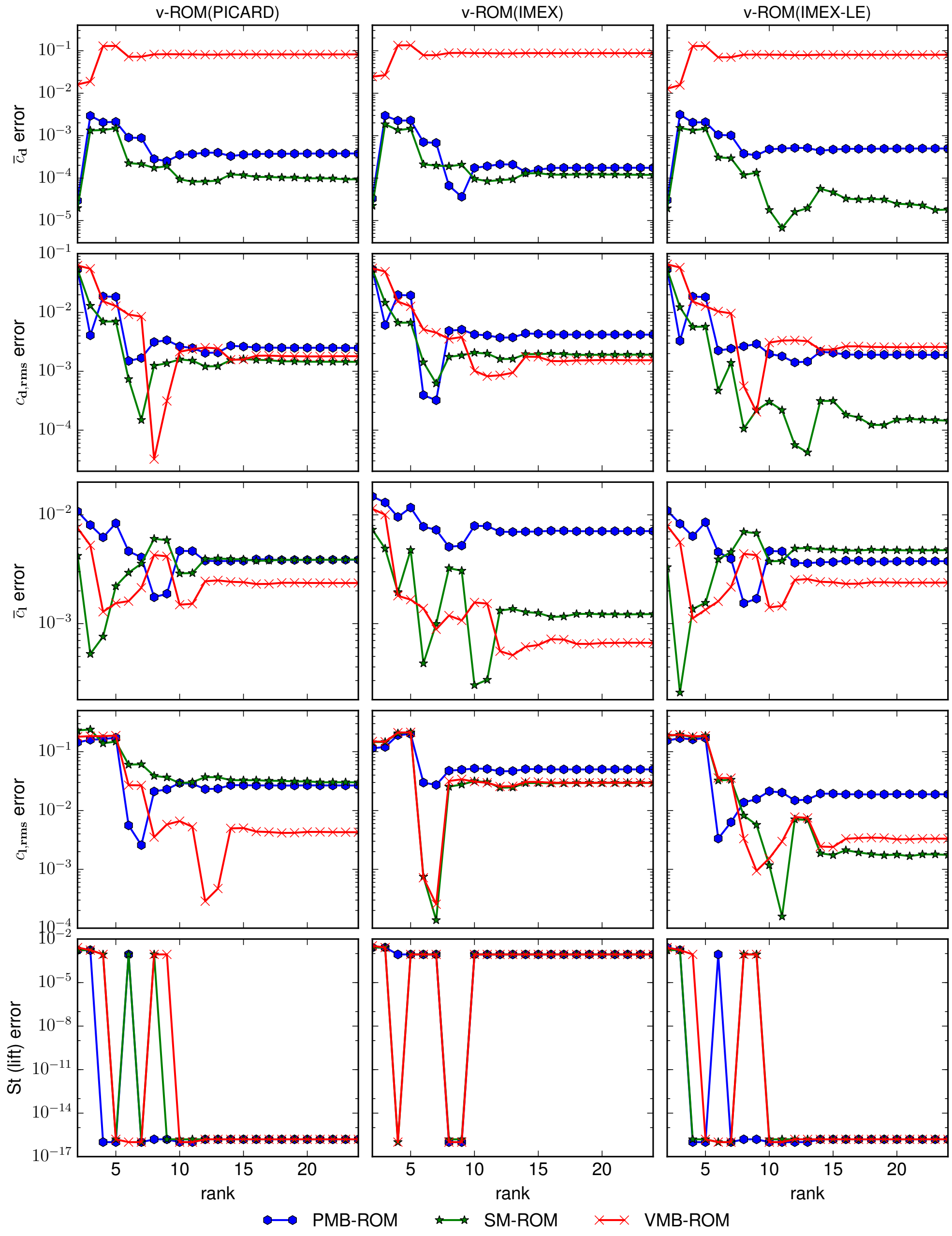

Figure 5.17.: Example 5.1, snapshots of intermediate accuracy (SP-LIN): Drag mean, drag rms, lift mean, lift rms, and Strouhal number errors (from top to bottom) obtained by the vp-ROMs combined with the Picard, IMEX and IMEX-LE nonlinear iteration schemes (left to right).

(5.48) can reach any magnitude of accuracy, if a large enough value of r is chosen, when the same linearization method is utilized both for the finite element simulation and the v-ROM.

In Fig. 5.16, the time evolution of the drag and lift coefficients for all studied vp-ROMs is shown. No obvious relation in the results can be observed between using any particular linearization method within the v-ROM and the IMEX scheme within SP-LIN. VBM-ROM does not manage to reproduce the range of the drag coefficient also for this set of snapshots. For $r \geq 6$, the drag and lift coefficients can be reproduced very well by both PMB-ROM and SM-ROM. The root mean square values of the peaks of the drag and lift coefficients are shown in Fig. 5.15. Similarly to the mean kinetic energy error in Fig. 5.14, the clearly best performance of the vp-ROMs is achieved when the same nonlinear iteration method is employed as in the finite element simulation, i.e., the IMEX scheme. In this case, all vp-ROMs, except PMB-ROM with respect to the drag coefficient, feature even somewhat better periodic behavior for both coefficients than SP-LIN for larger POD dimensions. All vp-ROMs combined with the two other linearization schemes reach for $r \geq 7$ a certain level of the rms value and never reach the accuracy of the underlying finite element simulation.

The representation of the errors with respect to the values obtained with SP-LIN in Fig. 5.17 allows a more detailed assessment of the results. In contrast to the rms values of the peaks of the coefficients in Fig. 5.17, there is no clear advantage to utilize the same linearization scheme in the vp-ROMs as in SP-LIN. SM-ROM performs often better than PMB-ROM, especially with the IMEX-LE scheme with the exception of the mean lift value. As expected, VMB-ROM features a large error in the very important for applications mean drag value, but otherwise it produces in many cases even smaller errors than the other linearizations. For $r \geq 10$, all vp-ROMs combined with the IMEX scheme fail to reproduce the Strouhal number as good as with the other linearization methods. The lengths of the period differ by about one time step Δt.

vp-ROMs using snapshots of the lowest accuracy. Finally, the numerical investigations of vp-ROMs based on the snapshots which were obtained with the finite element simulations PC will be presented. Similarly to the former studies for other snapshot sets, Fig. 5.18 shows the time evolution of the y-component of the velocity field and its corresponding frequency-based representation for PC and the v-ROM combined with the three different linearization schemes. One can assert that in general the frequencies with the largest amplitudes for the finite element simulation and the v-ROMs coincide, which is important for the mean kinetic energy error (5.47) to be a meaningful quantity of interest. However, the reproduction of the finite element results is visibly worse even for $r = 24$ than it was the case for SP-NONLIN and SP-LIN in Figs. 5.7 and 5.13. This trend can be also observed for the mean kinetic energy error shown in Fig. 5.19. In contrast to Figs. 5.8 and 5.14, the use of none of the linearization schemes yields an error of arbitrarily high accuracy, when the POD dimension is chosen to be large enough. According to previous findings, one would expect this behavior for v-ROM(IMEX) as PC involved the IMEX scheme for the sake of linearization. Although v-ROM(IMEX)

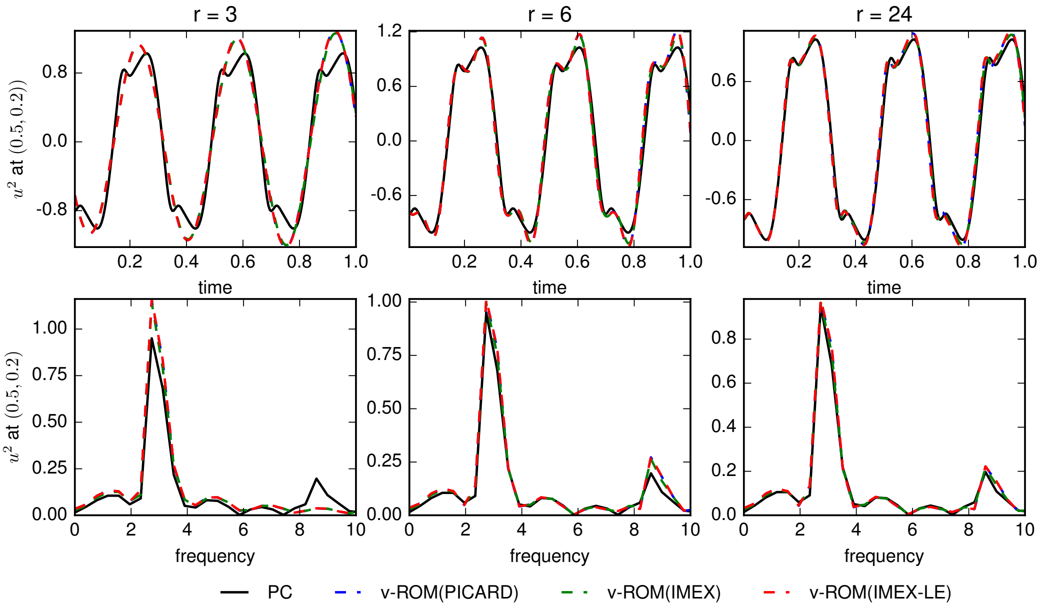

Figure 5.18.: Example 5.1, snapshots of the lowest accuracy (PC): Time-based (top) and frequency-based (bottom) representation of the y-component of the velocity field evaluated at $(0.5, 0.2)$.

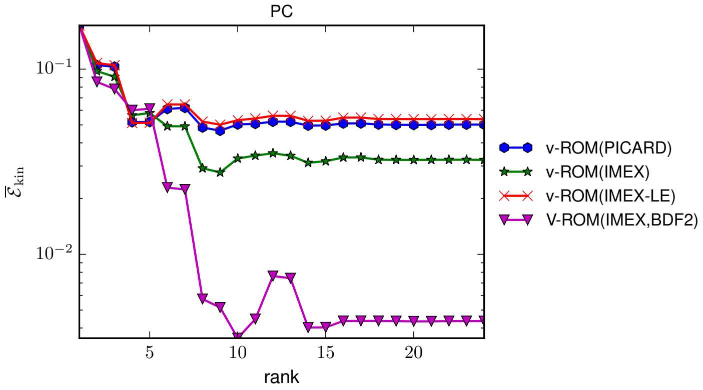

Figure 5.19.: Example 5.1, snapshots of the lowest accuracy (PC): Mean kinetic energy error (5.47) for different POD dimensions r computed from the velocity field obtained with the v-ROMs.

produces the smallest error compared to the v-ROMs combined with the Picard and IMEX-LE schemes, it stagnates for $r \geq 8$ at the level of $\overline{\mathcal{E}}_{\mathrm{kin}} \approx 2 \cdot 10^{-2}$. One possible explanation for this performance is the fact that the incremental pressure-correction scheme (2.100), which was employed to compute the underlying snapshots, includes the BDF2 time discretization scheme in contrast to the v-ROM which utilizes the Crank–Nicolson

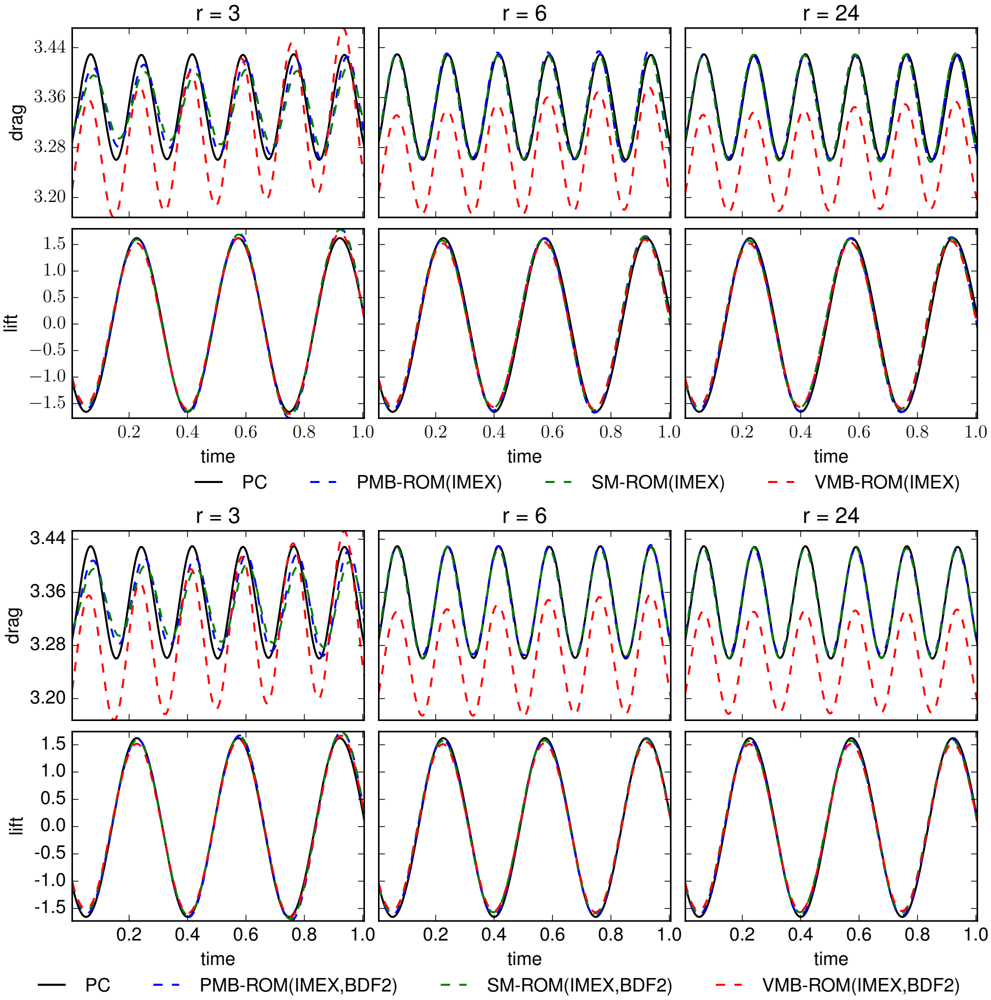

Figure 5.20.: Example 5.1, snapshots of the lowest accuracy (PC): Time evolution of the drag and lift coefficients computed by the vp-ROMs for $r = 3, 6, 24$ (from left to right).

method. To investigate if one can achieve any improvement of the results by changing the time discretization strategy, the v-ROM simulations together with the IMEX and BDF2 schemes, denoted by v-ROM(IMEX,BDF2), were carried out. The corresponding mean kinetic energy error is shown in Fig. 5.19. A clear improvement of the error of almost one order of magnitude, compared with vp-ROM(IMEX), can be asserted. However, also for this setting the error does not fall constantly for the increasing dimension of the POD space but reaches at some point a plateau. The remaining inaccuracy is most likely caused by the fact that the velocity snapshots are not discretely divergence-free, which was a necessary condition for the derivation of the velocity ROM in Section 5.3. Adding the pressure term $\left(p_{\mathrm{ro}}, \nabla \cdot \boldsymbol{\phi}_{\mathrm{ro},i}\right)$, $i = 1, \ldots, r$, was tested numerically but did

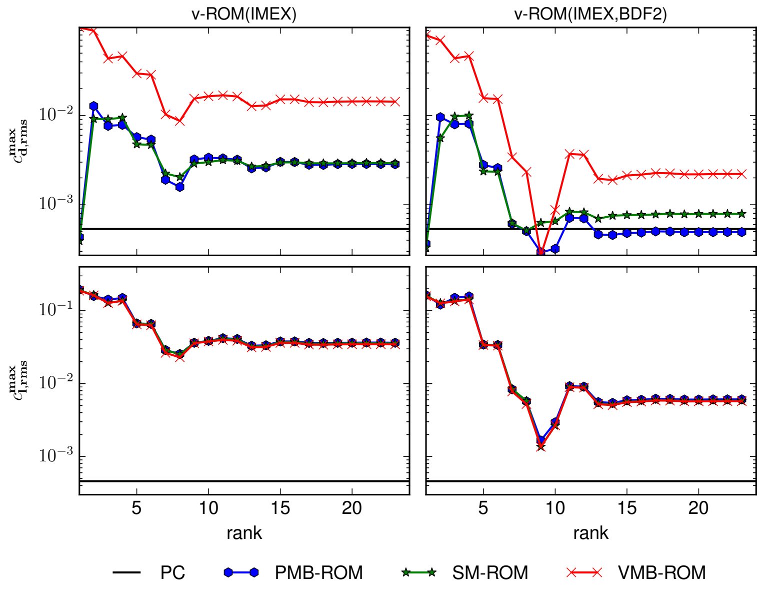

Figure 5.21.: Example 5.1, snapshots of the lowest accuracy (PC): $c_{\text{d,rms}}^{\max}$ (top) and $c_{\text{l,rms}}^{\max}$ (bottom) for the finite element simulation PC and the three vp-ROMs.

not yield any improvement of the results. A possible remedy could be, e.g., the application of the coupled Galerkin ROM for the Navier–Stokes equations (5.9). However, the investigation of this issue is out of scope of this thesis.

The rest of the measures of interest studied for the other two sets of snapshots were very similar for vp-ROM(Picard), vp-ROM(IMEX), and vp-ROM(IMEX-LE). For this reason, only the vp-ROM(IMEX) results will be presented. They will be compared with the results produced by the vp-ROMs combined with the IMEX scheme for linearization and the BDF2 method for time discretization of the momentum equation. These vp-ROMs will be correspondingly denoted by PMB-ROM(IMEX,BDF2), SM-ROM(IMEX,BDF2), and VMB-ROM(IMEX,BDF2).

Figure 5.20 displays the time evolution of the drag and lift coefficients for the three types of vp-ROM(IMEX) and vp-ROM(IMEX,BDF2). Again, PMB-ROM and SM-ROM in both settings were able to reproduce the results of the underlying snapshots quite well, whereas VMB-ROM failed for the drag coefficient. The deviation of the peaks from the mean value for both coefficients is depicted in Fig. 5.21. One can clearly see that $c_{\text{d,rms}}^{\max}$ for VMB-ROM(IMEX) and VMB-ROM(IMEX,BDF2) is larger than for the other two types of vp-ROMs. It is in accordance with the situation in Fig. 5.20, where the increase of the peaks of the drag coefficient for VMB-ROM(IMEX) can be observed even for $r = 24$. The other two types of vp-ROMs perform in terms of $c_{\text{d,rms}}^{\max}$ somewhat better if the BDF2 method is utilized instead of the Crank–Nicolson scheme. PMB-ROM(IMEX,BDF2) even reaches the accuracy of the underlying finite element simulation for $r \geq 13$. The rms value for the maximum lift coefficient is more or less the

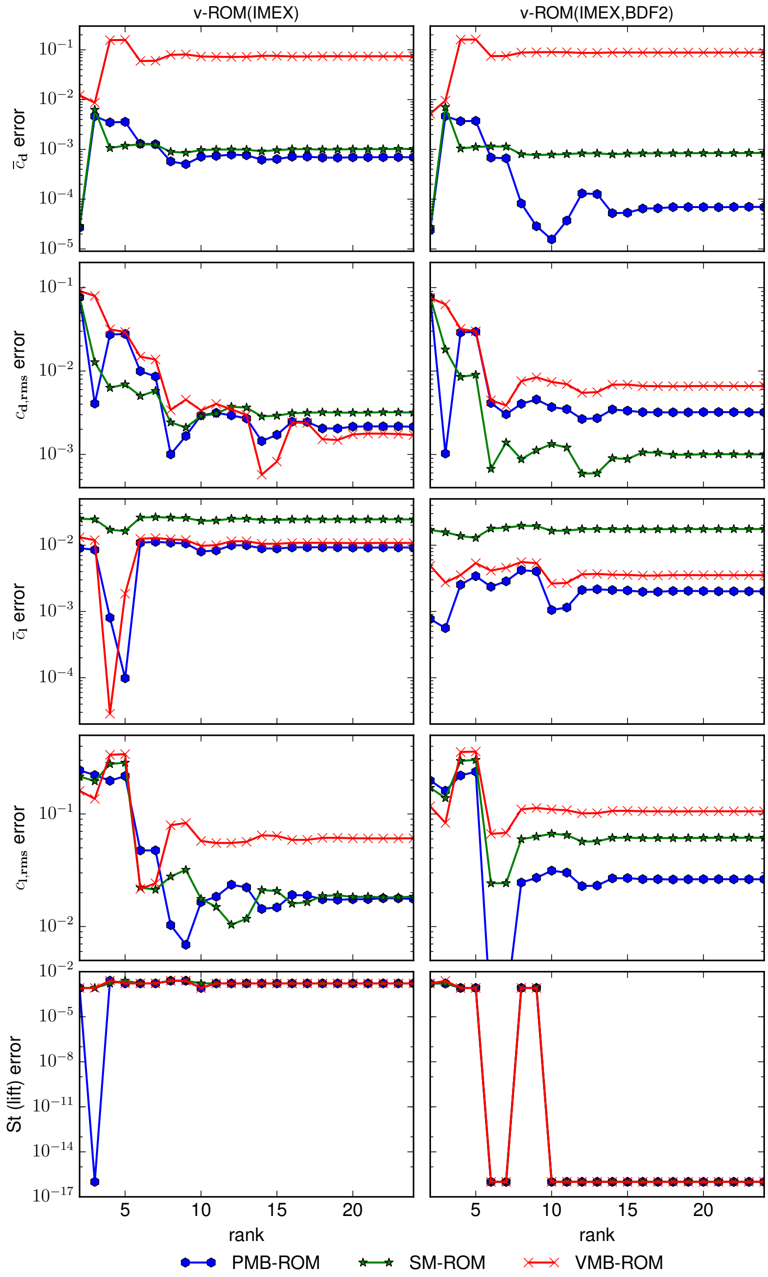

Figure 5.22.: Example 5.1, snapshots of the lowest accuracy (PC): Drag mean, drag rms, lift mean, lift rms, and Strouhal number errors (from top to bottom) obtained by the vp-ROMs.

same for all types of vp-ROMs. By employing the BDF2 and not the Crank–Nicolson method in the v-ROM, the values become however somewhat smaller but $c_{\mathrm{l,rms}}^{\mathrm{max}}$ for PC is not reached for any POD rank.

The errors in the mean and rms values for both coefficients as well as the error in the Strouhal number are presented in Fig. 5.22. One cannot assert any evident improvement in the results on the right-hand side of the plot. For some errors, e.g., in $\overline{c}_{\mathrm{d}}$ or St, the performance could be slightly improved by employing the BDF2 method. At the same time, other errors were comparable or became larger. In most cases, PMB-ROM produced somewhat smaller errors than SM-ROM.

Computational Cost

At each time step, the vp-ROMs that were investigated in this report were comprised of the computation of the ROM velocity solving the v-ROM of the form (5.12) followed by the computation of the ROM using one of the pressure ROMs presented in Sections 5.4.1-5.4.3.

Three different approaches for the linearization of the convective term were utilized in the v-ROM: the Picard, IMEX, ann IMEX-LE methods. All other computational steps, such as computation of the POD basis, the matrices and the right-hand side in the offline stage, are the same. The latter two methods have more or less the same computational costs, whereas the computation with the Picard method lasts from two to three times longer, depending on how many iterations are necessary to achieve the accuracy threshold of 10^{-10} in the nonlinear iteration loop.

Next, the computational complexity of the pressure ROMs will be discussed. The ROM pressure in PMB-ROM and SM-ROM requires the computation of the pressure POD modes, which represents the most time consuming part of their offline stage, and the assembling and factorization of the matrices in (5.38) and (5.44). These procedures are not necessary for VMB-ROM. However, the pressure coefficients $p_{ij}(\boldsymbol{x})$ in (5.30) have to be pre-computed by solving (5.31). In the presented numerical experiments, for $r = 25$, the computation of the $(r+1)r/2$ coefficients $p_{ij}(\boldsymbol{x})$ took about twice as long as the computation of the r pressure modes. Thus, in the offline stage, the computational costs of PMB-ROM and SM-ROM are lower than those of VMB-ROM.

In the online stage, the main difference is that VMB-ROM does not require the solution of a linear system for the pressure at each iteration, as the pressure is recovered as a linear combination of pre-computed functions $p_{ij}(\boldsymbol{x})$, see (5.30). Thus, it would seem that the computational cost of VMB-ROM is lower than the computational cost of the two other vp-ROMs. It could be observed, however, that this is not the case. In fact, the solution of the $r \times r$ linear system in PMB-ROM and SM-ROM requires only $\mathcal{O}(r^2)$ operations, yielding relatively low computational times. In addition, the cost of recovering the finite element approximation is $\mathcal{O}(rN_p)$, where N_p denotes the dimension of the pressure finite element space. On the other hand, for a given r, the computational complexity for VMB-ROM is $\mathcal{O}(r^2N_p)$, as the approximation of the finite element pressure solution is computed as a linear combination of the functions $p_{ij}(\boldsymbol{x})$, see (5.31), which are represented by N_p coefficients. Since $r \ll N_p$, the computational costs for VMB-ROM in the online stage is higher than for the other two vp-ROMs. In our numerical experiments, in the online stage, the computational times of PMB-ROM and SM-ROM combined with the IMEX or IMEX-LE schemes were about the same for moderate values of r ($r < 15$), representing between 0.01% and 0.06% of the computing time of SP-NONLIN. For the same range of values of r, VMB-ROM utilized with the same linearization schemes in the v-ROM was computationally more expensive, taking between 0.10% and 1.15% of the time of SP-NONLIN. $\triangleleft$

6. Summary and Outlook

6.1. Summary

This thesis studies projection-based reduced-order modeling in the context of computational fluid dynamics. Proper Orthogonal Decomposition (POD), introduced in Chapter 3, is utilized in order to obtain a reduced-order basis from the snapshots, which are assumed to represent the finite element solution of a partial differential equation. All investigations are based on the convection-diffusion-reaction equation and the incompressible Navier–Stokes equations. These equations and finite element methods employed for the computation of the snapshots are formulated in Chapter 2. The main results of this dissertation can be divided into three parts. Their summary is discussed in the following.

Firstly, Chapter 4 presented a theoretical and numerical investigation of a Streamline-Upwind Petrov–Galerkin reduced-order model, denoted by SUPG-ROM, for the convection-dominated convection-diffusion-reaction equation. At a theoretical level, numerical analysis was used to suggest the stabilization parameter applied in the SUPG-ROM. Two scalings for the stabilization parameter were proposed: One based on the underlying finite element discretization (which yields the FE-SUPG-ROM with the stabilization parameter δ_r^{FE}) and one based on the POD truncation (which yields the POD-SUPG-ROM with the stabilization parameter δ_r^{POD}). At a numerical level, the Galerkin reduced-order model (G-ROM), the FE-SUPG-ROM, and the POD-SUPG-ROM were tested on two convection-dominated convection-diffusion-reaction problems aiming at answering two questions: First, whether the SUPG-ROM yields more accurate results than the G-ROM, and, second, which of the two SUPG-ROMs produces more accurate results. The numerical investigations yielded that if the finite element discretization was fine enough to capture the internal layer (and thus no numerical stabilization was needed to compute the snapshots), then the standard G-ROM yielded accurate results. The SUPG-ROM gave only negligibly better results for smaller values of the POD dimension r. If the finite element discretization was not able to resolve the internal layer, which often occurs when dealing with convection-dominated problems, the following conclusions could be drawn:

- On relatively coarse meshes, which is the usual situation encountered in practice, a SUPG finite element discretization was used to generate the snapshots. This approach led to POD modes containing numerical artifacts (spurious oscillations). Thus, the considered ROMs used noisy POD data.
- The standard G-ROM yielded comparable results to the SUPG-ROMs with respect to the discrete $L^1(0,T;L^2(\Omega))$ error only if sufficiently few POD modes were used.

Once the number of POD modes was increased above a certain limit, the noise of these modes was reflected strongly in the results of the G-ROM.

- Both the FE-SUPG-ROM and the POD-SUPG-ROM yielded results that were significantly more accurate than those for the G-ROM for larger numbers of used POD modes. Both SUPG-ROMs suppressed the noise of the POD modes much better than the G-ROM.
- The exact meanings of "sufficiently few" and "large" numbers of POD modes in the previous points depend on the example and the mesh width. In practice, corresponding values for the number of POD modes will generally not be known.
- In the numerical studies, it was observed that δ_r^{POD} was in general smaller than δ_r^{FE}. Based on the author's experience so far, it is recommended that one uses δ_r^{FE}, i.e., FE-SUPG-ROM. This choice of the SUPG-ROM stabilization parameter suppressed the spurious oscillations in the ROM in many situations somewhat better than δ_r^{POD} and it was never observed that the results obtained with FE-SUPG-ROM were notably worse in any respect than the results of POD-SUPG-ROM. Moreover, the computation of δ_r^{FE} is easier than that of δ_r^{POD}.
- The sensitivity study with respect to the mesh width showed that, although the ROMs yield different results, their qualitative behavior remains unchanged.

Another objective of the numerical studies in Chapter 4 was to investigate, whether the accuracy of the ROM solution is related to the accuracy of the underlying snapshots. Of particular importance was the question, whether the ROMs based on physically correct snapshots, which exhibit no under- and overshoots, are able to reproduce physically correct ROM solutions. For this sake, three sets of snapshots were employed to compute the POD bases for the G-ROM and the SUPG-ROM. Two snapshot sets represented numerical solutions of a discrete problem by means of the SUPG finite element method and the flux-corrected transport scheme, denoted by SUPG-FEM and FEM-FCT, respectively. The third set of snapshots was obtained by interpolating the solution of the continuous problem using the same finite element space as for other two sets of snapshots. By construction, it was the most accurate one in terms of the L^2 error. The quantitative behavior of the under- and overshoots was measured by the minimum and maximum values of the solution. SUPG-FEM produced the largest L^2 error and was the only snapshot set, which exhibited under- and overshoots. The numerical simulations yielded that the accuracy of the minimum and maximum values of the SUPG-ROM solutions based on the snapshot sets, which represented the numerical solution of a discrete problem, i.e., SUPG-FEM and FEM-FCT, correlated with the accuracy of the underlying snapshots. The quality of the SUPG-ROM solutions in terms of the L^2 error was however similar for both sets of snapshots. Compared to the G-ROMs for all three snapshot sets, the SUPG-ROMs performed better in terms of the minimum and maximum values and slightly worse in terms of the L^2 error. It turned out that the SUPG-ROM is not a suitable ROM model for the POD basis obtained from the interpolated solution of the continuous problem. Its solution became too dissipative in the course of time. The under- and overshoots of the SUPG-ROM based on the FEM-FCT snapshots could not be completely overcome as it is the case for the corresponding finite element solution. However, the distance of the minima and maxima to the optimal values was smaller than for the SUPG-ROM based on the SUPG-FEM snapshots.

The second part of the results is associated with the computation of the ROM initial solution. Depending on the origin of the snapshots, the standard ROM initial condition, which is usually employed in the literature and represents the best approximation of the full-order initial condition in the L^2 sense, can be polluted by spurious oscillations. In Section 3.2.2, a new filtering procedure was proposed that aims at suppressing those spurious oscillations, which consists in utilizing the Galerkin approximation of the Helmholtz equation with respect to the POD basis in a post-processing step of the standard approach. The proposed ROM initial condition looses the property of the best possible approximation of the full-order solution in the L^2 sense. However, it could be shown for the special case of a family of uniform triangulations that the new ROM initial condition still approximates well the full-order initial condition in the L^2 sense with a convergence of at least first order. At the same time it can lead to a more appropriate approximation of the full-order initial condition in the sense of other quantities of interest, e.g., indicating the strength of under- and overshoots. In Section 4.4, the effect of the filtered ROM initial condition on the G-ROM and SUPG-ROM results with respect to the L^2 error as well as the minimum and maximum values of the solution was investigated. For this sake, three different sets of snapshots as described above were employed to compute the POD bases. It could be observed that the proposed filtering procedure was able to noticeably smoothen the standard ROM initial condition, which was plagued by spurious oscillations. The application of the new approach caused a slight increase of the L^2 error for all ROMs. With respect to the minimum and maximum values, the G-ROM and SUPG-ROM results could be for many settings significantly improved. In particular, for two sets of snapshots and moderate POD dimensions, one could achieve satisfactory results even by applying the standard G-ROM without any stabilization. The SUPG-ROM based on the SUPG-FEM snapshots resulted in a solution with smaller under- and overshoots than the corresponding finite element solution.

Finally, Chapter 5 deals with the velocity-pressure ROMs (vp-ROMs) for the incompressible Navier–Stokes equations, which comprises the third part of the results. The first goal of the chapter was to discuss and compare three different velocity-pressure ROMs. VMB-ROM uses only velocity POD modes, whereas PMB-ROM and SM-ROM use pressure POD modes as well. SM-ROM was developed within the framework of this dissertation and published in [26]. The second goal was to investigate the influence of employing different linearization techniques, known, e.g., from the context of finite element methods, on the accuracy of the ROMs in terms of several quantities of interest. The third goal was to perform the first step in answering the question about the impact of the snapshot accuracy, on the one hand, and of the potentially simple numerical methods used in the ROMs, on the other hand, onto the results of the ROMs. For studying these questions, three sets of snapshots with different accuracy were used. The numerical investigations showed that the snapshots had, irrespectively of the way they were computed and of their accuracy, a much stronger impact on the ROM results than the choice of the numerical methods used in the vp-ROMs. Generally, the results of the simulation for computing the snapshots were reproduced quite well with the velocity-pressure ROMs in terms of the drag and lift coefficients, and the Strouhal number. Altogether, this study clearly supports the approach of performing accurate (and probably time-consuming) simulations for computing the snapshots in order to obtain similar results in the ROM simulations.

Concerning the comparisons of the velocity-pressure ROMs, the main conclusion drawn from the numerical investigation is that the two ROMs that utilize pressure modes (PMB-ROM and SM-ROM) were superior, both in terms of reproducing the results of the simulations for computing the underlying snapshots and of efficiency, to the ROM that uses only velocity POD modes (VMB-ROM). The results obtained with VMB-ROM for an important quantity of interest, the mean drag coefficient, and in terms of the periodic behavior of the drag coefficient were not satisfactory. Based on weakly divergence-fee velocity snapshots, SM-ROM could reproduce the results of the finite element simulations in many cases better than PMB-ROM. Finally, together with the fact that SM-ROM does not need any specification of additional pressure boundary conditions, which is required in PMB-ROM, SM-ROM can be considered to be superior to other vp-ROMs for the computation of the ROM pressure.

The numerical studies associated with the second goal in Chapter 5 showed that the employed linearization techniques in the velocity ROM, namely the Picard, IMEX, and IMEX-LE methods discussed in Section 2.2.5, had different effects on the reproduction of the finite element simulations depending on the quantity of interest taken into account. In order to obtain the best possible mean kinetic energy error (5.47), the discrete version of the $L^1(0,T;L^2(\Omega))$ error (5.48), and the periodic behavior of the peaks of the drag and lift coefficients, it is of advantage to utilize the same nonlinear iteration scheme in the velocity ROM as in the underlying finite element simulation. However, for the most accurate set of snapshots, the IMEX-LE method performed better, especially in terms of the drag coefficient, than the IMEX method. With respect to the errors in the mean and the root mean squared values of the drag and lift coefficients, as well as the error in the Strouhal number, one could not observe any clear relation concerning the nonlinear iteration schemes between the finite element simulations for the computation of snapshots and the corresponding vp-ROMs. The vp-ROMs combined with the Picard and IMEX-LE methods yielded in many situations slightly smaller errors than the ones with the IMEX scheme. However, in contrast to finite element methods, for which the IMEX scheme is generally considerably less accurate than the other two methods, the difference in the accuracy of the results between all three schemes is very small in the context of the vp-ROMs.

6.2. Outlook

In the subsequent studies, it would be interesting to investigate whether the SUPG-ROM is generally not suitable for snapshots, which do not represent the solution of a discrete problem. This aspect is especially important for applications, in which the snapshots arise from physical experiments and not from numerical simulations. Moreover, it is desirable to develop a ROM able to suppress the non-physical under- and overshoots of its solution similarly to the FEM-FCT scheme in the context of finite element methods.

Several research directions could be pursued in the future concerning velocity-pressure ROMs for the incompressible Navier–Stokes equations. For instance, one can study whether the conclusions of the numerical studies in Chapter 5 carry over to the case of structure-dominated turbulent flows. In addition, the rigorous numerical analysis for discretizations of the new velocity-pressure ROM (SM-ROM) can be a topic of future research.

A. Function Spaces and Inequalities

In this chapter, the function spaces and inequalities, which are most commonly employed in the thesis, will be presented. Let Ω be an arbitrary domain in $\mathbb{R}^d$, where d is the dimension of the physical space, and let Γ denote the boundary of the domain. In the framework of this work, there will be no difference in the notation between function spaces for scalar and vector-valued functions. The reader is referred to [1] for a comprehensive presentation of the subject matter in this chapter.

Definition A.1 (Lebesgue spaces)**.** *For $1 \leq p < \infty$, the Lebesgue space denoted by $L^p(\Omega)$ consists of the all measurable functions f defined on Ω for which*

$$\int\limits_{\Omega} |f(\boldsymbol{x})|^p \, d\boldsymbol{x} < \infty.$$

The space $L^\infty(\Omega)$ is defined as a vector space of all functions f that are essentially bounded on Ω, i.e., f is measurable and there is a constant C such that $|f(\boldsymbol{x})| < C$ a.e. on Ω. The Lebesgue norm is defined by

$$|f\|_{L^p(\Omega)} = \begin{cases} \left(\int\limits_{\Omega} |f(\boldsymbol{x})|^p \, d\boldsymbol{x} \right)^{1/p}, & \textit{if} \quad 1 \leq p < \infty, \\ \operatorname{ess\,sup}_{\boldsymbol{x} \in \Omega} |f(\boldsymbol{x})|, & \textit{if} \quad p = \infty. \end{cases} \tag{A.1}$$

Remark A.1. An important special case is the Lebesgue space $L^2(\Omega)$ as it represents a Hilbert space. The associated inner product is defined by

$$(f, g) = \int\limits_{\Omega} f(\boldsymbol{x}) g(\boldsymbol{x}) \, d\boldsymbol{x}, \tag{A.2}$$

and the induced norm, which will be denoted by $\|\cdot\|_0$ instead of $\|\cdot\|_{L^2(\Omega)}$, is defined by $\|f\|_0 = (f, f)^{1/2}$. ◁

Definition A.2 (Sobolev spaces)**.** *For any positive integer m and $1 \leq p \leq \infty$, the Sobolev space denoted by $W^{m,p}(\Omega)$ is given by*

$$W^{m,p}(\Omega) = \left\{ f \in L^p(\Omega) : \ D^\alpha f \in L^p(\Omega) \text{ for } 0 \leq |\alpha| \leq m \right\}, \tag{A.3}$$

where $D^\alpha f$ denotes the weak partial derivative of f (e.g., see [1, Paragraph 1.62] for its definition).

The Sobolev norm is defined by

$$\|f\|_{W^{m,p}(\Omega)} = \begin{cases} \left(\sum\limits_{0\le|\alpha|\le m} \|D^\alpha f\|^p_{L^p(\Omega)} \right)^{1/p}, & \textit{if} \quad 0 \le p < \infty, \\ \max\limits_{0\le|\alpha|\le m} \|D^\alpha f\|_{L^\infty(\Omega)}, & \textit{if} \quad p = \infty. \end{cases} \tag{A.4}$$

Remark A.2. Sobolev spaces with $p = 2$ are Hilbert spaces and will be denoted by $W^{m,2}(\Omega) = H^m(\Omega)$. They are equipped with the inner product

$$(f, g)_m = \sum_{0\le|\alpha|\le m} (D^\alpha f, D^\alpha g) \tag{A.5}$$

and with the induced norm

$$\|f\|_m = \left(\sum_{0\le|\alpha|\le m} (D^\alpha f, D^\alpha f) \right)^{1/2}, \tag{A.6}$$

which is denoted for the sake of simplicity by $\|\cdot\|_m$ instead of $\|\cdot\|_{W^{m,2}(\Omega)}$. For $m = 0$, it holds $H^0(\Omega) = L^2(\Omega)$ with the corresponding notation $(f, g)_0 = (f, g)$. ◁

Remark A.3. Sobolev spaces denoted by $W_0^{m,p}(\Omega)$ are defined to be the closure of $C_0^\infty(\Omega)$ in the space $W^{m,p}(\Omega)$. Here, $C_0^\infty(\Omega)$ consists of all functions in $C^\infty(\Omega) = \bigcap_{m=0}^\infty C^m(\Omega)$ that have compact support in Ω. In this thesis, the case $p = 2$ is of importance, i.e., the spaces $W_0^{m,2}(\Omega) = H_0^m(\Omega)$.

For $1 \le m < \infty$ and for $1 \le j \le m$ the functionals $|\cdot|_j$ on $H^m(\Omega)$ defined by

$$|f|_j = \left(\sum_{|\alpha|=j} \left(\|D^\alpha f\|_0^2 \right) \right)^{1/2} \tag{A.7}$$

are called seminorms. If Ω is a bounded domain, then $|\cdot|_m$ is a norm on $H_0^m(\Omega)$ and, due to the Poincaré's inequality, it is equivalent to the norm $\|\cdot\|_m$ (see [1, Corollary 6.31]). Seminorm (A.7) can be induced from the semi-inner product

$$\langle f, g \rangle_m = \sum_{|\alpha|=m} (D^\alpha f, D^\alpha g). \tag{A.8}$$

◁

Remark A.4. In the context of finite element methods, often the norms (A.1) for $p = 2$ and $p = \infty$, (A.6), (A.7), and the inner product (A.2) are needed, which are not defined on the entire domain Ω but locally on the individual mesh cells $K \in \mathcal{T}_h$. Here, $\mathcal{T}_h$ denotes the triangulation of the domain Ω. These norms and the inner product will be denoted by $\|\cdot\|_{0,K}$, $\|\cdot\|_{L^\infty(K)}$, $\|\cdot\|_{m,K}$, $|\cdot|_{m,K}$, $(\cdot,\cdot)_K$, respectively, and are defined equivalently by replacing Ω by K in the according integral. ◁

Definition A.3 (Bochner spaces)**.** *Let Y be a Banach space with the norm $\|\cdot\|_Y$, and let (t_0, t_1) denote a time interval. For $1 \leq p \leq \infty$, the Bochner space denoted by $L^p(t_0, t_1; X)$ is defined by*

$$L^p(t_0, t_1; X) = \left\{ f(t, \boldsymbol{x}) : \ \|f\|_{L^p(t_0,t_1;X)} < \infty \right\},$$

where the norm reads

$$\|f\|_{L^p(t_0,t_1;X)} = \begin{cases} \left(\int_{t_0}^{t_1} \|f(t)\|_X^p \, dt \right)^{1/p}, & \text{if} \quad 1 \leq p < \infty, \\ \operatorname{ess\,sup}_{t_0 \leq t \leq t_1} \|f\|_X \, , & \text{if} \quad p = \infty. \end{cases} \tag{A.9}$$

Lemma A.1. *(Cauchy–Schwarz inequality). Let X be an inner product space with the inner product $(\cdot,\cdot)_X$, which induces the norm $\|\cdot\|_X$. Then the so-called Cauchy-Schwarz inequality holds*

$$|(f, g)_X| \leq \|f\|_X \|g\|_X \quad \forall f, g \in X. \tag{A.10}$$

Lemma A.2. *(Young's inequality for convolutions). Let $1 \leq p, q \leq \infty$ and $\frac{1}{r} = \frac{1}{p} + \frac{1}{q} - 1 \geq 0$. For $f \in L^p(\Omega)$ and $g \in L^q(\Omega)$, it holds $f * g \in L^r(\Omega)$, and the so-called Young's inequality holds*

$$\|f * g\|_{L^r(\Omega)} \leq \|f\|_{L^p(\Omega)} \|g\|_{L^q(\Omega)}. \tag{A.11}$$

Lemma A.3. *(Young's inequality for real numbers). Let $a, b \in \mathbb{R}$, then the following Young's inequality holds*

$$ab \leq \frac{t}{p} a^p + \frac{t^{-q/p}}{q} b^q, \quad \frac{1}{p} + \frac{1}{q} = 1, \quad 1 < p, q < \infty, \quad t > 0. \tag{A.12}$$

List of Principal Notations

$A_{\mathrm{ro}}^{\mathrm{NS}}$	Reduced stiffness matrix (velocity POD space)	120
A_{ro}	Reduced stiffness matrix	59
A_h^{NS}	Stiffness matrix (velocity finite element space)	37
B_h^{NS}	Mixed velocity-pressure matrix	37
G	Gramian matrix	53
$H^m(\Omega)$	Sobolev space $W^{m,2}(\Omega)$	160
H_{ro}	Reduced Hessian matrix	76
K	Number of boundary parts with Dirichlet boundary condition	62
$L^p(\Omega)$	Lebesgue space	159
$L^p(t_0, t_1; X)$	Bochner space	161
M	Number of snapshots	51
$M_{\mathrm{ro}}^{\mathrm{NS}}$	Reduced mass matrix (velocity POD space)	120
M_L	Lumped mass matrix	18
M_{ro}	Reduced mass matrix	59
M_h	Mass matrix	8
M_h^{NS}	Mass matrix (velocity finite element space)	37
N	Dimension of the finite element space X_h	7
$N_{\mathrm{ro}}^{\mathrm{NS}}(\cdot)$	Reduced convective matrix (velocity POD space)	120
$N_h^{\mathrm{NS}}(\cdot)$	Convective matrix (velocity finite element space)	37
N_p	Dimension of the pressure finite element space	37
N_v	Dimension of the velocity finite element space (one component)	37
P_h^{NS}	Stiffness matrix (pressure finite element space)	39
Q	Infinite-dimensional pressure space	34
Q_h	Pressure finite element space	37
R	Dimension of the snapshot space	51
Re	Reynolds number	21
St	Strouhal number	130
T	Length of considered time interval	6
U	Snapshot matrix	52
V	Infinite-dimensional velocity space	34
V_h	Velocity finite element space	37
W	Diagonal matrix with the weights of a quadrature formula	53
$W^{m,p}(\Omega)$	Sobolev space	159
X_R	Snapshot space	51
X_h	Finite element space	7
X_r	POD space	52
Γ	Boundary of the domain	159

Γ_0	Boundary with no-slip boundary condition	23
Γ_{in}	Inflow boundary with Dirichlet boundary condition	31
$\Gamma_{\text{D},k}$	kth boundary part with Dirichlet boundary condition	62
Γ_{D}	Boundary with Dirichlet boundary condition	22
Γ_{out}	Boundary with natural boundary condition	23
$\vert\vert\vert\cdot\vert\vert\vert_2$	Spectral matrix norm	75
$\Vert\cdot\Vert_{L^p(\Omega)}$	Norm on Lebesgue space $L^p(\Omega)$, $1 \leq p \leq \infty$	159
$\Vert\cdot\Vert_0$	Norm on Lebesgue space $L^2(\Omega)$	159
Ω	Arbitrary domain in $\mathbb{R}^d$	159
Φ	POD matrix	56
Φ_v	Velocity POD matrix	121
$\vert\vert\vert\cdot\vert\vert\vert_{\text{SUPG},h}$	SUPG norm in the finite element space	12
$\vert\vert\vert\cdot\vert\vert\vert_{\text{SUPG},r}$	SUPG norm in the POD space	79
$\boldsymbol{b}$	Convection field	6
$\boldsymbol{g}_{\text{D}}$	Velocity Dirichlet boundary condition	31
$\boldsymbol{\phi}_{\text{ro},i}$	Velocity POD basis function	117
$\boldsymbol{\phi}_{h,i}$	Velocity finite element basis function	37
$\boldsymbol{u}$	Velocity field	21
$\boldsymbol{u}_{\text{ro}}$	Reduced-order approximation of the velocity field	118
$\boldsymbol{u}_h$	Velocity finite element solution	37
$\boldsymbol{v}_h$	Finite element function in V_h	37
$\mathcal{E}(r)$	Energy ratio	57
$\mathcal{T}_h$	Triangulation of the domain Ω	7
χ^n	Pressure increment	30
δ_r	SUPG-ROM stabilization parameter	74
δ_r^{FE}	SUPG-ROM stabilization parameter (finite element option)	85
δ_r^{POD}	SUPG-ROM stabilization parameter (POD option)	87
$\delta_{h,K}$	Local SUPG finite element stabilization parameter	11
Δt	Length of the time step (constant)	8
ε	Diffusion coefficient	6
γ_k	Temporal component of the Diriclet boundary condition on $\Gamma_{\text{D},k}$	62
λ_i	POD eigenvalue	52
$(\cdot,\cdot)$	Inner product on Lebesgue space $L^2(\Omega)$	159
$\langle\cdot,\cdot\rangle_m$	Semi-inner product on Sobolev space $H^m(\Omega)$	160
μ_0	Constant for the condition on reaction and convection fields	12
μ_{inv}	Constant of the local inverse estimate	7
ν	Dimensionless viscosity	21
$\overline{\mathcal{E}}_{\text{kin}}$	Mean kinetic energy error	136
$\overline{c}_{\text{d}}$	Mean value of the drag coefficient	137
$\overline{c}_{\text{l}}$	Mean value of the lift coefficient	137
$\overline{p}_h$	Average of the pressure snapshots	118
$\overline{u}_h$	Average of the snapshots	62
$\psi_{\text{ro},i}$	Pressure POD basis function	117

$\psi_{h,i}$	Pressure finite element basis function	37
θ	Parameter for the one-step θ-scheme	8
$\tilde{\boldsymbol{u}}$	Intermediate velocity field	27
$\underline{\alpha}$	Reduced-order solution (array of length r)	59
$\underline{\alpha}_p$	Reduced-order pressure solution (array of length r_p)	126
$\underline{\alpha}_v$	Reduced-order velocity solution (array of length r_v)	120
$\varphi_{h,i}$	Finite element basis function	7
$\varphi_{\mathrm{ro},i}$	POD basis function	52
$a(\cdot,\cdot)$	Bilinear form	6
$a_{\mathrm{SUPG},h}(\cdot,\cdot)$	SUPG bilinear form in the finite element space	11
$a_{\mathrm{SUPG},r}(\cdot,\cdot)$	SUPG bilinear form in the POD space	73
c	Reaction field	6
c_{d}	Drag coefficient	130
c_{l}	Lift coefficient	130
$c_{\mathrm{d,rms}}^{\max}$	Root mean squared value of local maxima of c_{d}	137
$c_{\mathrm{l,rms}}^{\max}$	Root mean squared value of local maxima of c_{l}	137
$c_{\mathrm{d,rms}}$	Root squared mean value of the drag coefficient	137
$c_{\mathrm{d}}^{\max}$	Maximum drag coefficient	131
$c_{\mathrm{l,rms}}$	Root squared mean value of the lift coefficient	137
$c_{\mathrm{l}}^{\max}$	Maximum lift coefficient	131
d	Dimension of the physical space	159
g_{D}	Dirichlet boundary condition on Γ_{D}	58
g_k	Spacial component of the Ditrichlet boundary condition on $\Gamma_{\mathrm{D},k}$	62
h	Maximal size of the mesh cells in $\mathcal{T}_h$	15
h_K	Size of the mesh cell for $K \in \mathcal{T}_h$	7
p	Pressure field	21
p_{ro}	Reduced-order approximation of the pressure field	118
p_h	Pressure finite element solution	37
q_h	Finite element function in Q_h	37
r	Dimension of the POD basis	52
r_p	Dimension of the pressure POD basis	117
r_v	Dimension of the velocity POD basis	117
u_g	Extension of Dirichlet boundary condition on Γ_{D} into Ω	68
u_h	Finite element solution	7
u_{ro}	Reduced-order approximation of the full-order solution	59
v_h	Finite element function in X_h	7

Bibliography

[1] R.A. Adams and J.J.F. Fournier. *Sobolev Spaces*, volume 140. Academic Press, 2003.

[2] I. Akhtar, A.H. Nayfeh, and C.J. Ribbens. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theor. Comput. Fluid Dyn.*, 23:213–237, 2009.

[3] J. Alberty, C. Carstensen, and S.A Funken. Remarks around 50 lines of Matlab: short finite element implementation. *Numerical Algorithms*, 20(2-3):117–137, 1999.

[4] D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM Journal on Scientific Computing*, 33(5):2169–2198, 2011.

[5] D. Amsallem and C. Farhat. Stabilization of projection-based reduced-order models. *Internat. J. Numer. Methods Engrg.*, 91(4):358–377, 2012.

[6] E. Arian, M. Fahl, and E.W. Sachs. Trust-region proper orthogonal decomposition for flow control. Technical report, DTIC Document, 2000.

[7] D.N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21(4):337–344 (1985), 1984.

[8] J.A. Atwell, J.T. Borggaard, and B.B. King. Reduced order controllers for Burgers' equation with a nonlinear observer. *Applied Mathematics And Computer Science*, 11(6):1311–1330, 2001.

[9] N. Aubry, P. Holmes, J.L. Lumley, and E. Stone. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Mechanics*, 192(1):115–173, 1988.

[10] N. Aubry, W.Y. Lian, and E.S. Titi. Preserving symmetries in the proper orthogonal decomposition. *SIAM J. Sci. Comput.*, 14:483–505, 1993.

[11] M. Augustin, A. Caiazzo, A. Fiebach, J. Fuhrmann, V. John, A. Linke, and R. Umla. An assessment of discretizations for convection-dominated convection-diffusion equations. *Comput. Methods Appl. Mech. Engrg.*, 200(47-48):3395–3409, 2011.

[12] J. Baiges, R. Codina, and S. Idelsohn. Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 72(12):1219–1243, 2013.

[13] M. Balajewicz and E.H. Dowell. Stabilization of projection-based reduced order models of the Navier–Stokes. *Nonlinear Dynamics*, 70:1619–1632, 2012.

[14] F. Ballarin and G. Rozza. POD–Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 2016.

[15] H.T. Banks, M.L. Joyner, B. Wincheski, and W.P. Winfree. Nondestructive evaluation using a reduced-order computational methodology. *Inverse Problems*, 16(4):929, 2000.

[16] M.F. Barone, I. Kalashnikova, D.J. Segalman, and H.K. Thornquist. Stable Galerkin reduced order models for linearized compressible flow. *J. Comput. Phys.*, 228(6):1932–1946, 2009.

[17] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 197(1-4):173–201, 2007.

[18] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. *ACTA NUMERICA*, 14:1–137, 2005.

[19] M. Bergmann, C.-H. Bruneau, and A. Iollo. Enablers for robust POD models. *J. Comput. Phys.*, 228(2):516–538, 2009.

[20] M. Bergmann, L. Cordier, and J.-P. Brancher. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of Fluids*, 17:21, 2005.

[21] M. Braack, E. Burman, V. John, and G. Lube. Stabilized finite element methods for the generalized Oseen problem. *Comput. Methods Appl. Mech. Engrg.*, 196(4-6):853–866, 2007.

[22] M. Braack, P.B. Mucha, and W.M. Zajaczkowski. Directional do-nothing condition for the Navier–Stokes equations. *Journal of Computational Mathematics*, 32(5):507–521, 2014.

[23] D. Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics.* Cambridge University Press, 2007.

[24] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1-3):199–259, 1982.

[25] J. Burkardt, M. Gunzburger, and H.-C. Lee. POD and CVT-based reduced-order modeling of Navier–Stokes flows. *Comput. Methods Appl. Mech. Engrg.*, 196(1-3):337–355, 2006.

[26] A. Caiazzo, T. Iliescu, V. John, and S. Schyschlowa. A numerical investigation of velocity-pressure reduced order models for incompressible flows. *J. Comput. Phys.*, 259:598–616, 2014.

[27] S. Chaturantabut and D.C. Sorensen. A state space error estimate for POD-DEIM nonlinear model reduction. *SIAM J. Numer. Anal.*, 50(1):46–63, 2012.

[28] A.J. Chorin. Numerical solution of the Navier–Stokes equations. *Math. Comput.*, 22:745–762, 1968.

[29] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland Publishing Co., Amsterdam-New York-Oxford, 1978. Studies in Mathematics and its Applications, Vol. 4.

[30] R. Codina. On stabilized finite element methods for linear systems of convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Engrg.*, 188:61–82, 2000.

[31] C.M. Colciago. *Reduced order fluid-structure interaction models for haemodynamics applications.* Thesis, EPFL, 2014.

[32] J.B. Conway. *A Course in Functional Analysis.* Springer-Verlag, New York, 1990. Second Edition.

[33] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[34] W. Dahmen, C. Plesken, and G. Welper. Double greedy algorithms: reduced basis methods for transport dominated problems. *ESAIM: Math. Model. Numer. Anal.*, 48(3):623–663, 2014.

[35] L. Dede. Reduced basis method for parametrized advection-reaction problems. *J. Comput. Math*, 28(1):122–148, 2010.

[36] W. E and J.-G. Liu. Projection Method I: Convergence and Numerical Boundary Layers. *SIAM J. Numer. Anal.*, 32(4):1017–1057, 1995.

[37] P. Feldmann and R.W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 14(5):639–649, 1995.

[38] L.P. Franca and F. Valentin. On an improved unusual stabilized finite element method for the advective-reactive-diffusive equation. *Comput. Methods Appl. Mech. Engrg.*, 190(13-14):1785–1800, 2000.

[39] J. Freund and R. Stenberg. On weakly imposed boundary conditions for second order problems. In *Proceedings of the Ninth Int. Conf. Finite Elements in Fluids, Venice*, pages 327–336, 1995.

[40] G.P. Galdi. *An Introduction to the Mathematical Theory of the Navier–Stokes Equations: Steady-State Problems.* Springer Monographs in Mathematics. Springer New York, 2011.

[41] G.P. Galdi and W.J. Layton. Approximation of the larger eddies in fluid motions II: A model for space-filtered flow. *Mathematical Models and Methods in Applied Sciences*, 10(03):343–350, 2000.

[42] B. Galletti, C.-H. Bruneau, L. Zannetti, and A. Iollo. Low-order modelling of laminar flow regimes past a confined square cylinder. *J. Fluid Mech.*, 503:161–170, 2004.

[43] K. Gallivan, E. Grimme, and P. Van Dooren. Asymptotic waveform evaluation via a Lanczos method. *Applied Mathematics Letters*, 7(5):75–80, 1994.

[44] S. Giere, T. Iliescu, V. John, and D. Wells. SUPG reduced order models for convection-dominated convection-diffusion-reaction equations. *Comput. Methods Appl. Mech. Engrg.*, 289:454–474, 2015.

[45] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier–Stokes Equations*, volume 5 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin, 1986. Theory and algorithms.

[46] K. Goda. A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows. *J. Comput. Phys.*, 30:76–95, 1979.

[47] W.R. Graham, J. Peraire, and K.Y. Tang. Optimal control of vortex shedding using low-order models. Part I – open-loop model development. *Internat. J. Numer. Methods Engrg.*, 44(7):945–972, 1999.

[48] W.R. Graham, J. Peraire, and K.Y. Tang. Optimal control of vortex shedding using low-order models. Part II – model-based control. *Internat. J. Numer. Methods Engrg.*, 44(7):973–990, 1999.

[49] M.A. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 41(3):575–605, 2007.

[50] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method, Isothermal Laminar Flow.* John Wiley & Sons, 2000.

[51] J.-L. Guermond. Stabilization of Galerkin approximations of transport equations by subgrid modeling. *M2AN Math. Model. Numer. Anal.*, 33(6):1293–1316, 1999.

[52] J.-L. Guermond. Un résultat de convergence d'ordre deux en temps pour l'approximation des équations de Navier-Stokes par une technique de projection incrémentale. *M2AN Math. Model. Numer. Anal.*, 33(1):169–189, 1999.

[53] J.-L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 195(44-47):6011–6045, 2006.

[54] J.-L. Guermond and L. Quartapelle. On stability and convergence of projection methods based on pressure Poisson equation. *Internat. J. Numer. Methods Fluids*, 26(9):1039–1053, 1998.

[55] M.D. Gunzburger. *Finite Element Methods for Viscous Incompressible Flows: a Guide to Theory, Practice, and Algorithms.* Computer science and scientific computing. Academic Press, 1989.

[56] M.D. Gunzburger, J.S. Peterson, and J.N. Shadid. Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Comput. Methods Appl. Mech. Engrg.*, 196(4-6):1030–1047, 2007.

[57] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 42(2):277–302, 2008.

[58] M. Hanke-Bourgeois. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens.* Springer, 2009.

[59] I. Harari and T.J.R. Hughes. What are C and h?: Inequalities for the analysis and design of finite element methods. *Comput. Methods Appl. Mech. Engrg.*, 97(2):157–192, 1992.

[60] J.S. Hesthaven, G. Rozza, and B. Stamm. Certified reduced basis methods for parametrized partial differential equations. *SpringerBriefs in Mathematics*, 2015.

[61] J. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure conditions for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 22(5):325–352, January 1996.

[62] J.G. Heywood and R. Rannacher. Finite element approximation of the nonstationary Navier–Stokes problem. Part I: Regularity of solutions and second-order error estimates for spatial discretization. *SIAM J. Numer. Anal.*, 19(2):275–311, 1982.

[63] P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge University Press, Cambridge, UK, 1996.

[64] D. Hömberg and S. Volkwein. Control of laser surface hardening by a reduced-order approach using proper orthogonal decomposition. *Mathematical and computer modelling*, 38(10):1003–1028, 2003.

[65] P. Hood and C. Taylor. Navier–Stokes equations using mixed interpolation. In J.T. Oden, R.H. Gallagher, O.C. Zienkiewicz, and C. Taylor, editors, *Finite Element Methods in Flow Problems*, pages 121–132. University of Alabama in Huntsville Press, 1984.

[66] T.J.R. Hughes and A. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In *Finite Element Methods for Convection Dominated Flows*, volume 34 of *AMD*, pages 19–35. Amer. Soc. Mech. Engrs. (ASME), New York, 1979.

[67] T. Iliescu and Z. Wang. Variational multiscale proper orthogonal decomposition: Convection-dominated convection-diffusion-reaction equations. *Math. Comput.*, 82, 2013.

[68] T. Iliescu and Z. Wang. Are the snapshot difference quotients needed in the proper orthogonal decomposition? *SIAM J. Sci. Comput.*, 36(3):A1221A1250, 2014.

[69] T. Iliescu and Z. Wang. Variational multiscale proper orthogonal decomposition: Navier–Stokes equations. *Num. Meth. P.D.E.s*, 30(2):641–663, 2014.

[70] R. Ingram. A new linearly extrapolated Crank–Nicolson time-stepping scheme for the Navier–Stokes equations. *Math. Comp.*, 82(284):1953–1973, 2013.

[71] A. Iollo, A. Dervieux, J.-A. Désidéri, and S. Lanteri. Two stable POD-based approximations to the Navier–Stokes equations. *Comput. Vis. Sci.*, 3(1-2):61–66, 2000.

[72] E.W. Jenkins, V. John, A. Linke, and L.G. Rebholz. On the parameter choice in grad-div stabilization for the Stokes equations. *Advances in Computational Mathematics*, 40(2):491–516, 2014.

[73] V. John. *Large Eddy Simulation of Turbulent Incompressible Flows: Analytical and Numerical Results for a Class of LES Models.* Lecture Notes in Computational Science and Engineering. Springer Berlin Heidelberg, 2003.

[74] V. John. On the efficiency of linearization schemes and coupled multigrid methods in the simulation of a 3D flow around a cylinder. *Internat. J. Numer. Methods Fluids*, 50(7):845–862, 2006.

[75] V. John. *Finite Element Methods for Incompressible Flow Problems*, volume to be assigned of *Springer Series in Computational Mathematics.* Springer, submitted.

[76] V. John and P. Knobloch. On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations I. A review. *Comput. Methods Appl. Mech. Engrg.*, 196(17-20):2197–2215, 2007.

[77] V. John and G. Matthies. Higher-order finite element discretizations in a benchmark problem for incompressible flows. *Int. J. Numer. Methods Fluids*, 37(8):885–903, 2001.

[78] V. John and G. Matthies. MooNMD—a program package based on mapped finite element methods. *Comput. Vis. Sci.*, 6(2-3):163–169, 2004.

[79] V. John, G. Matthies, and J. Rang. A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 195(44-47):5995–6010, 2006.

[80] V. John, T. Mitkova, M. Roland, K. Sundmacher, L. Tobiska, and A. Voigt. Simulations of population balance systems with one internal coordinate using finite element methods. *Chemical Engineering Science*, 64(4):733–741, 2009.

[81] V. John and J. Novo. Error analysis of the SUPG finite element discretization of evolutionary convection-diffusion-reaction equations. *SIAM J. Numer. Anal.*, 49(3):1149–1176, 2011.

[82] V. John and J. Novo. On (essentially) non-oscillatory discretizations of evolutionary convection-diffusion equations. *J. Comput. Phys.*, 231(4):1570–1586, 2012.

[83] V. John and J. Rang. Adaptive time step control for the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 199(9-12):514–524, 2010.

[84] V. John and E. Schmeyer. Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion. *Comput. Methods Appl. Mech. Engrg.*, 198(3-4):475–494, 2008.

[85] I. Kalashnikova, S. van Bloemen Waanders, B. Arunajatesan, and M. Barone. Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment. *Comput. Methods Appl. Mech. Engrg.*, 272(15):251–270, 2014.

[86] T. Knopp, G. Lube, and G. Rapin. Stabilized finite element methods with shock capturing for advection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 191(27-28):2997–3013, 2002.

[87] A.N. Kolmogorov and S.V. Fomin. *Elements of the Theory of Functions and Functional Analysis: Measure. The Lebesgue integral. Hilbert space. Translated by H. Kamel and H. Komm.* Elements of the Theory of Functions and Functional Analysis. Graylock Press, 1965.

[88] B. Kragel. *Streamline diffusion POD models in optimization.* Dissertation, University of Trier, 2005.

[89] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numer. Math.*, 90(1):117–148, 2001.

[90] K. Kunisch and S. Volkwein. Proper orthogonal decomposition for optimality systems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(01):1–23, 2008.

[91] D. Kuzmin. Explicit and implicit FEM-FCT algorithms with flux linearization. *J. Comput. Phys.*, 228(7):2517–2534, 2009.

[92] D. Kuzmin and M. Möller. Algebraic flux correction I. Scalar conservation laws. In *Flux-Corrected Transport*, Sci. Comput., pages 155–206. Springer, Berlin, 2005.

[93] D. Kuzmin, M. Möller, and S. Turek. High-resolution FEM-FCT schemes for multidimensional conservation laws. *Comput. Methods Appl. Mech. Engrg.*, 193(45-47):4915–4946, 2004.

[94] D. Kuzmin and S. Turek. Flux correction tools for finite elements. *J. Comput. Phys.*, 175(2):525–558, 2002.

[95] L.D. Landau and E.M. Lifshits. *Fluid Mechanics.* Pergamon Press, Oxford, England; New York, 1987.

[96] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In *Reduced Order Methods for Modeling and Computational Reduction*, pages 235–273. Springer, 2014.

[97] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In A. Quarteroni and G. Rozza, editors, *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *Modeling, Simulation and Applications*, pages 235–274. Springer, 2014.

[98] W.J. Layton. *Introduction to the Numerical Analysis of Incompressible Viscous Flows.* Computational science and engineering series. Society for Industrial and Applied Mathematics, Philadelphia, 2008.

[99] R.J. Leveque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.*, 33(2):627–665, 1996.

[100] A. Linke. Collision in a cross-shaped domain—a steady 2D Navier–Stokes example demonstrating the importance of mass conservation in CFD. *Comput. Methods Appl. Mech. Engrg.*, 198(41-44):3278–3286, 2009.

[101] T. Linß and M. Stynes. Numerical methods on Shishkin meshes for linear convection-diffusion problems. *Computer methods in applied mechanics and engineering*, 190(28):3527–3542, 2001.

[102] G. Lube and G. Rapin. Residual-based stabilized higher-order FEM for a generalized Oseen problem. *Math. Models Methods Appl. Sci.*, 16(7):949–966, 2006.

[103] J.L. Lumley. The structure of inhomogeneous turbulence. *Athmospheric turbulence and wave propagation (ed. A. Yaglom and V. Tatarski)*, pages 166–178, 1967.

[104] Z. Luo, J. Chen, I.M. Navon, and X. Yang. Mixed finite element formulation and error estimates based on proper orthogonal decomposition for the nonstationary Navier–Stokes equations. *SIAM J. Numer. Anal.*, 47(1):1–19, 2008/09.

[105] K. Morgan, J. Peraire, and R. Löhner. Adaptive finite element flux corrected transport techniques for CFD. In *Finite elements*, ICASE/NASA LaRC Ser., pages 165–175. Springer, New York, 1988.

[106] N.-C. Nguyen, G. Rozza, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. *Calcolo*, 46(3):157–185, 2009.

[107] B.R. Noack, M. Morzynski, and G. Tadmor. *Reduced-Order Modelling for Flow Control*, volume 528. Springer Verlag, 2011.

[108] B.R. Noack, P. Papas, and P.A. Monkewitz. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *J. Fluid Mech.*, 523:339–365, 2005.

[109] P. Pacciarini and G. Rozza. Stabilized reduced basis method for parametrized advection-diffusion PDEs. *Comput. Meth. Appl. Mech. Eng.*, 274:1–18, 2014.

[110] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations.* MIT Pappalardo Graduate Monographs in Mechanical Engineering, 2006.

[111] S.B. Pope. *Turbulent Flows.* Cambridge University Press, Cambridge, 2000.

[112] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier–Stokes equations by reduced basis methods. *Numer. Methods Partial Differential Equations*, 23(4):923–948, 2007.

[113] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Numerical mathematics and scientific computation. Clarendon Press, 1999.

[114] R. Rannacher. On Chorin's projection method for the incompressible Navier–Stokes equations. In *The Navier-Stokes Equations II – Theory and Numerical Methods*, pages 167–183. Springer, 1992.

[115] R. Rannacher. Finite element methods for the incompressible Navier–Stokes equations. In *Fundamental Directions in Mathematical Fluid Mechanics*, Adv. Math. Fluid Mech., pages 191–293. Birkhäuser, Basel, 2000.

[116] S.S. Ravindran. Control of flow separation over a forward-facing step by model reduction. *Comput. Methods Appl. Mech. Engrg.*, 191(41-42):4599–4617, 2002.

[117] M. Reed and B. Simon. *Functional Analysis (Methods of Modern Mathematical Physics).* Academic Press, 1980.

[118] H.-G. Roos, M. Stynes, and L. Tobiska. *Numerical Methods for Singularly Perturbed Differential Equations: Convection-Diffusion-Reaction and Flow Problems.* Springer series in computational mathematics. Springer, 2008.

[119] G. Rozza and K. Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Engrg.*, 196(7):1244–1260, 2007.

[120] E.W. Sachs and M. Schu. Reduced order models (POD) for calibration problems in finance. In *Numerical Mathematics and Advanced Applications*, pages 735–742. Springer, 2008.

[121] E.W. Sachs and S. Volkwein. POD–Galerkin approximations in PDE-constrained optimization. *GAMM-Mitteilungen*, 33(2):194–208, 2010.

[122] M. Schäfer and S. Turek. The benchmark problem "Flow around a cylinder". In E.H. Hirschel, editor, *Flow Simulation with High-Performance Computers II*, volume 52 of *Notes on Numerical Fluid Mechanics*, pages 547–566. Vieweg, 1996.

[123] H.R. Schwarz and N. Köckler. *Numerische Mathematik.* Teuber, 2004.

[124] J. Shen. On error estimates of the projection methods for the Navier–Stokes equations: second-order schemes. *Math. Comp.*, 65(215):1039–1065, 1996.

[125] Y. Shih, J.-Y. Cheng, and K.-T. Chen. An exponential-fitting finite element method for convection-diffusion problems. *Appl. Math. Comput.*, 217(12):5798–5809, 2011.

[126] J.R. Singler. New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs. *SIAM J. Numer. Anal.*, 52(2):852–876, 2014.

[127] S. Sirisup and G.E. Karniadakis. A spectral viscosity method for correcting the long-term behavior of POD models. *J. Comput. Phys.*, 194(1):92–116, 2004.

[128] L. Sirovich. Turbulence and the dynamics of coherent structures. Parts I–III. *Quart. Appl. Math.*, 45(3):561–582, 1987.

[129] H. Sohr. *The Navier–Stokes Equations: An Elementary Functional Analytic Approach.* Springer Science & Business Media, 2012.

[130] R. Ştefănescu and I. M. Navon. POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model. *J. Comput. Phys.*, 2012.

[131] M. Stynes. Steady-state convection-diffusion problems. *Acta Numerica*, 14:445–508, 2005.

[132] R. Temam. Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires ii. *Arch. Ration. Mech. Anal.*, 33:673–688, 1969.

[133] R. Temam. *Navier–Stokes Equations: Theory and Numerical Analysis*, volume 2 of *Studies in mathematics and its applications.* North-Holland, 1985, Amsterdam, 1984.

[134] L.J.P. Timmermans, P.D. Minev, and F.N. Van De Vosse. An approximate projection scheme for incompressible flow using spectral elements. *International journal for numerical methods in fluids*, 22(7):673–688, 1996.

[135] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach.* Springer, January 1999. 3-540-65433-X.

[136] S. Ullmann. *POD–Galerkin Modeling for Incompressible Flows with Stochastic Boundary Conditions.* Verlag Dr. Hut, 2014.

[137] S. Ullmann and J. Lang. A POD–Galerkin reduced model with updated coefficients for Smagorinsky LES. In *Proceedings of the V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010*, 2010.

[138] S. Ullmann, S. Löbig, and J. Lang. Adaptive large eddy simulation and reduced-order modeling. In *Flow and Combustion in Advanced Gas Turbine Combustors*, pages 349–378. Springer, 2013.

[139] J. van Kan. A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM J. Sci. Statist. Comput.*, 7(3):870–891, 1986.

[140] S. Volkwein. Optimal control of a phase-field model using proper orthogonal decomposition. *ZAMM Z. Angew. Math. Mech.*, 81(2):83–97, 2001.

[141] S Volkwein. Proper orthogonal decomposition: Applications in optimization and control. Lecture Notes, CEA-EDF-INRIA Summer School, 2008.

[142] S. Volkwein. Model reduction using proper orthogonal decomposition. Lecture Notes, Faculty of Mathematics and Statistics, University of Konstanz, 2011.

[143] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Two-level discretizations of nonlinear closure models for proper orthogonal decomposition. *Journal of Computational Physics*, 230(1):126–146, 2011.

[144] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. *Comput. Methods Appl. Mech. Engrg.*, 237/240:10–26, 2012.

[145] J. Weller, E. Lombardi, M. Bergmann, and A. Iollo. Numerical methods for low-order modeling of fluid flows based on POD. *Internat. J. Numer. Methods Fluids*, 63(2):249–268, 2010.

[146] S.T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31(3):335–362, 1979.

[147] S. Zhang. A new family of stable mixed finite elements for the 3D Stokes equations. *Math. Comp.*, 74(250):543–554, 2005.

[148] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*, volume 40. Prentice Hall, 1996.

Kurzzusammenfassung

Diese Dissertation beschäftigt sich mit projektionsbasierten ordnungsreduzierten Modellen (ROMs) im Rahmen der numerischen Strömungsmechanik. Proper Orthogonal Decomposition (POD) wird zur Berechnung der ordnungsreduzierten Basis aus den sogenannten Schnappschüssen eingesetzt. Es wird angenommen, dass die Schnappschüsse die Finite-Elemente-Lösung einer partiellen Differentialgleichung darstellen. Der Beitrag der vorliegenden Dissertation besteht aus drei Teilen.

Erstens wird ein Streamline-Upwind Petrov–Galerkin ordnungsreduziertes Modell, bezeichnet als SUPG-ROM, für konvektionsdominante Konvektions-Diffusions-Reaktions-Gleichungen sowohl theoretisch als auch numerisch untersucht. Mittels numerischer Analyse wird die Skalierung der Stabilisierungsparameter für SUPG-ROMs vorgeschlagen. Dabei werden zwei Ansätze verwendet: Der eine basiert auf der zugrundeliegenden Finite-Elemente-Diskretisierung und der andere auf der räumlichen Auflösung im Zusammenhang mit POD. Die resultierenden SUPG-ROMs und das übliche Galerkin ROM werden mittels mehrerer konvektionsdominanter Testbeispiele untersucht.

Zweitens wird ein alternativer Ansatz für die Berechnung der ROM-Anfangsbedingung für Probleme entwickelt, bei welchen der Standard-Ansatz, der in der Regel in der Literatur verwendet wird, eine durch Störschwingungen verfälschte Anfangsbedingung erzeugt. Die Grundidee des Verfahrens besteht darin, die herkömmliche ordnungsreduzierte Anfangsbedingung in einem Nachbearbeitungsschritt durch ein Filterverfahren zu modifizieren. Der Einfluss der gefilterten ROM-Anfangsbedingung auf die ROM-Ergebnisse wird numerisch untersucht. In Bezug auf die Minimal- und Maximalwerte der ordnungsreduzierten Lösung, die als Maß für Unter- und Überschwingungen dienen, konnten die ROM-Ergebnisse im Vergleich zu denen mit üblichen ROM-Anfangsbedingungen zum Teil deutlich verbessert werden.

Drittens werden drei ordnungsreduzierte Modelle für Geschwindigkeit und Druck (vp-ROMs) für inkompressible Strömungen numerisch untersucht. Eines dieser Verfahren berechnet den ROM-Druck allein auf der Grundlage der POD-Moden für die Geschwindigkeit, während die beiden anderen vp-ROMs auch die POD-Basiselemente für den Druck verwenden. Eine der letztgenannten Methoden, bezeichnet als SM-ROM, wurde im Rahmen dieser Arbeit entwickelt. Des Weiteren wird die Auswirkung der Genauigkeit der Schnappschüsse sowie verschiedener Linearisierungsansätze auf die ROM-Ergebnisse numerisch untersucht. Für die im schwachen Sinne divergenzfreien Geschwindigkeits-Schnappschüsse konnte SM-ROM die Ergebnisse der Finite-Elemente-Simulationen in vielen Fällen am besten approximieren. Unter Berücksichtigung, dass SM-ROM im Gegensatz zu den anderen vp-ROMs keine Angabe zusätzlicher Druckrandbedingungen benötigt, lässt sich hieraus ableiten, dass SM-ROM im Vergleich zu den zwei anderen untersuchten vp-ROMs am besten zur Berechnung des POD-Drucks geeignet ist.

Selbstständigkeitserklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbstständig angefertigt habe und alle benutzten Hilfsmittel und Quellen vollständig angegeben habe. Eine Anmeldung der Promotionsabsicht habe ich an keiner anderen Fakultät oder Hochschule beantragt.

Berlin, den 21.07.2016

Swetlana Giere