POD reduced order modeling for evolution equations utilizing arbitrary finite element discretizations

Carmen Gräßle and Michael Hinze

Abstract

The main focus of the present work is the inclusion of spatial adaptivity for snapshot computation in the offline phase of model order reduction utilizing Proper Orthogonal Decomposition (POD-MOR). For each time level, the snapshots live in different finite element spaces, which means in a fully discrete setting that the snapshots are vectors of different length. In order to overcome this obstacle, we present a discretization independent POD reduced order model, which is motivated from a continuous perspective and is set up explicitly without interpolation of the snapshots. The analysis for the error between the resulting POD solution and the true solution is carried out. Finally, we present numerical examples to illustrate our approach.

1 Introduction

For many problem settings, model order reduction utilizing Proper Orthogonal Decomposition (POD) has proven to be a powerful tool in order to reduce large scale systems to surrogate models of low dimension while preserving a good approximation quality. The range of applications of POD model order reduction (POD-MOR) comprises a broad scope, including (amongst others) linear and nonlinear parabolic equations [26], optimal control of partial differential equations [17, 43, 25, 23, 3] and fluid dynamics [31, 20, 30]. A general introduction to POD and reduced order modeling can be found in [22, 36, 44], for example. The key idea of the POD technique is to apply a Galerkin ansatz, in which the ansatz functions, i.e. the POD basis functions, contain information about the underlying dynamical system. Following the approach of snapshot based POD in [41], the system information is retrieved from snapshots of the solution trajectory at several time instances, which are generated in a simulation.

In practice, many simulations require adaptive strategies for the spatial discretization in order to be implementable. For example, in the simulation of Cahn-Hilliard systems based on diffuse interface approaches, many degrees of freedom are required at the interfacial regions in order to well resemble the steep gradient, whereas in the pure phases only little numbers of degrees of freedom are needed, see [19] for example. Utilizing a uniform mesh for such problems would drastically enlarge the computational effort and storage problems would occur.

In [9] adaptive finite element methods and POD reduced order modeling are considered as two different techniques in order to reduce the complexity of the numerical solution of optimal control problems. The combination of both approaches contains a major challenge. The inclusion of spatial adaptivity in the context of model reduction means, in a discrete formulation, that the snapshots at each time instance may have different lengths due to their different spatial resolutions. For this reason, the snapshot matrix can not be set up directly and the usual POD procedure can not be carried out. In this paper, we consider the combination of adaptive finite element snapshots with POD model order reduction. Our perspective is based on the continuous setting, which allows us to derive a reduced order model which is independent of the underlying finite element discretization. The inclusion of spatial adaptivity in the POD framework is advantageous from two perspectives:

on the one hand, the use of adaptive finite elements for snapshot generation can remarkably reduce the offline computation time in comparison to the use of a uniform mesh with resolution of the finest level of the adaptive grids. On the other hand, we expect to speed up the computations when solving the POD surrogate model in contrast to utilizing adaptive finite elements, since we solve POD reduced systems of low order.

In order to overcome the difficulties arising from combining POD with adaptive finite elements, different concepts are proposed. In [15], the use of dynamically adaptive meshes is combined with POD-MOR for an unstructured ocean model. A fixed reference mesh is utilized, onto which the spatial adaptive snapshots are interpolated. This allows snapshots of the same lengths at each time level and the usual POD procedure can then be performed on this fixed reference mesh. In [28, Ch. 2.4.3] an interpolation approach is outlined. The idea is to interpolate given snapshots of arbitrary spatial discretizations by (piecewise) polynomials. For the fully discrete POD setting, the spatial discretization points are chosen appropriately for the numerical integration of the polynomials. In the context of reduced basis methods, adaptive wavelet discretizations are used in [5] in the offline snapshot computation phase. In [47], a reduced basis method is developed which is certified by a residual bound relative to the infinite-dimensional weak solution. Different adaptive strategies are considered for both the finite element and the reduced basis level. Furthermore, in [21] three numerical concepts to treat the moving free boundary for the calculation of the snapshots are compared: first a Landau-type transformation, second a control volume approach and third a moving mesh approach. In contrary to the first two concepts, the number of grid points in the moving mesh approach (r-adaptivity) is kept fixed, but they are moved according to the evolution of the free boundary. POD is applicable with only minor modifications. Recently and in parallel to our work, the combination of POD model reduction with adaptive finite element snapshots is realized in [42] by constructing common finite element spaces. Two options are considered: either all snapshots are expressed in terms of a common finite element basis or pairs of snapshots are expressed in terms of a common finite element basis of these pairs. Moreover, error statements for a parametrized elliptic boundary value problem are proved. In the numerical examples, h-adaptive finite elements with fixed polynomial degree are utilized.

The aim of this work is to derive a POD reduced order model for a general semilinear evolution equation which can be set up and solved for arbitrary finite element discretizations without using interpolation with respect to the spatial variable. This approach is motivated from a continuous perspective, where snapshots from different finite element spaces belong to a same Hilbert space. The assembly of the snapshot matrix can be avoided by directly setting up an eigenvalue problem for which we only need the calculation of the inner product of the finite element ansatz functions. This can be computed for arbitrary finite element discretizations and suffices to set up a POD-ROM. Moreover, the treatment of the nonlinearity is discussed.

The paper is organized as follows: the general problem setting of an abstract semilinear parabolic evolution problem is described in Section 2. We recall the POD method in real Hilbert spaces in Section 3 and set up a POD eigenvalue problem which can be assembled for any finite element discretization. The POD reduced order model for arbitrary finite element discretizations is discussed in Section 4. In order to validate the quality of the POD solution, we investigate the error between the POD solution and the true solution in Section 5. Numerical examples are presented in Section 6 to illustrate our approach.

2 Abstract semilinear parabolic evolution problem

2.1 Problem setting

Let us specify the abstract semilinear parabolic evolution problem which we consider in the following. Let $(V, \langle \cdot, \cdot \rangle_V)$ and $(H, \langle \cdot, \cdot \rangle_H)$ be real separable Hilbert spaces such that there exists a dense and continuous embedding $V \hookrightarrow H$. The dual space H' can be identified with H by the Riesz representation theorem and the Gelfand triple (V, H, V') is formed by $V \hookrightarrow H = H' \hookrightarrow V'$. Since V is continuously embedded in H, there exists a constant $c_v > 0$ such that

$$\|v\|_H^2 \leq c_v \|v\|_V^2 \quad \text{for all } v \in V.$$

For a given symmetric, V-elliptic bilinear form $a : V \times V \to \mathbb{R}$, we assume boundedness, i.e.

$$\exists \beta > 0 : \quad |a(u,v)| \leq \beta \|u\|_V \|v\|_V \quad \text{for all } u, v \in V$$

and coercivity, i.e.

$$\exists \kappa > 0 : \quad a(u,u) \geq \kappa \|u\|_V^2 \quad \text{for all } u \in V.$$

Let $\mathcal{A} : V \to V'$ be the bounded linear operator associated with the bilinear form a, i.e. $\mathcal{A} \in \mathcal{L}(V, V')$ and

$$a(u,v) = \langle \mathcal{A}u, v \rangle_{V',V} = \langle u, \mathcal{A}^\star v \rangle_{V,V'} \quad \text{for all } u, v \in V,$$

where $\langle \cdot, \cdot \rangle_{V',V}$ denotes the dual pairing of V' and V. Moreover, we denote by $\mathcal{N} : V \to V'$ a nonlinear operator. We are concerned with the following Cauchy problem for a semilinear evolution problem. Let $T > 0$ be a fixed end time. For a given initial function $y_0 \in H$ and external force $f \in L^2(0,T;V')$ we consider the problem: find $y \in W(0,T) := \{v \in L^2(0,T;V), v_t \in L^2(0,T;V')\}$ with

$$\begin{cases} \frac{d}{dt} \langle y(t), v \rangle_H + a(y(t), v) + \langle \mathcal{N}(y(t)), v \rangle_{V',V} & = & \langle f(t), v \rangle_{V',V}, \\ \langle y(0), v \rangle_H & = & \langle y_0, v \rangle_H, \end{cases} \tag{1}$$

for all $v \in V$ and for almost all $t \in (0,T]$. Note that it holds $\frac{d}{dt} \langle y(t), v \rangle_H = \langle y'(t), v \rangle_{V',V}$ for all $y \in L^2(0,T;V)$ with $y' \in L^2(0,T;V')$ and all $v \in V$ in the sense of distributions in $(0,T)$.

Assumption 2.1. For every $f \in L^2(0,T;V')$ and $y_0 \in H$ there exists a unique weak solution of (1) with

$$y \in L^2(0,T;V) \cap C([0,T];H) \cap H^1(0,T;V').$$

Remark 2.2. Under monotonicity, boundedness and Lipschitz continuity assumptions on the nonlinear operator $\mathcal{N}$, existence and uniqueness results for a general abstract evolution equation of type (1) are proved in [46, Th. 4.1] or [33, Ch. 6], for example.

Example 2.3 (Semilinear heat equation). Let $\Omega \subset \mathbb{R}^k, k \in \{1, 2, 3\}$ be a bounded open domain with Lipschitz continuous boundary $\partial \Omega$ and let $T > 0$ be a fixed end time. We set $Q := (0,T) \times \Omega$ and $\Sigma := (0,T) \times \partial \Omega$ and $c \geq 0$. For a given forcing term $f \in L^2(Q)$ and initial condition $y_0 \in L^2(\Omega)$, we consider the semilinear heat equation with homogeneous Dirichlet boundary condition:

$$\begin{cases} y_t(t,x) - \Delta y(t,x) + cy^3(t,x) & = & f(t,x) & \text{in } Q, \\ y(t,x) & = & 0 & \text{on } \Sigma, \\ y(0,x) & = & y_0(x) & \text{in } \Omega. \end{cases} \tag{2}$$

The existence of a unique solution to (2) is proved in [37], for example. We can write (2) as an abstract evolution problem of type (1) by deriving a variational formulation for (2) with $V = H_0^1(\Omega)$ as the space of test functions, $H = L^2(\Omega)$ and integrating over the space Ω. The bilinearform $a : V \times V \to \mathbb{R}$ is introduced by

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v$$

and the operator $\mathcal{N} : V \to V'$ is defined as $\mathcal{N}(y) = y^3$. For $c \equiv 0$, the heat equation (2) is linear.

Example 2.4 (Cahn-Hilliard equations). Let Ω, T, Q and Σ be defined as in Example 2.3. The Cahn-Hilliard system was proposed in [11] as a model for phase separation in binary alloys. Introducing the chemical potential w, the Cahn-Hilliard equations can be formulated in the common setting as a coupled system for the phase field c and the chemical potential w:

$$\begin{cases} c_t(t,x) + y \cdot \nabla c(t,x) & = & m\Delta w(t,x) & \text{in } Q, \\ w(t,x) & = & -\sigma \varepsilon \Delta c(t,x) + \frac{\sigma}{\varepsilon} W'(c(t,x)) & \text{in } Q, \\ \nabla c(t,x) \cdot \nu_\Omega & = & \nabla w(t,x) \cdot \nu_\Omega = 0 & \text{on } \Sigma, \\ c(0,x) & = & c_0(x) & \text{in } \Omega. \end{cases} \tag{3}$$

By ν_Ω we denote the outward normal on $\partial \Omega$, $m \geq 0$ is a constant mobility, $\sigma > 0$ denotes the surface tension and $0 < \varepsilon \ll 1$ represents the interface parameter. Note that the convective

term $y \cdot \nabla c$ describes the transport with (constant) velocity y. The transport term represents the coupling to the Navier-Stokes equations in the context of multiphase flow, see e.g. [18] and [2]. The phase field function c describes the phase of a binary material with components A and B. It is $c \equiv -1$ in the pure A-phase and $c \equiv +1$ in the pure B-phase. The interfacial region is described by $c \in (-1, 1)$ and its thickness is finite and of order $\mathcal{O}(\varepsilon)$. The function $W(c)$ is a double well potential. A typical choice for W is the polynomial free energy function

$$W^p(c) = (1 - c^2)^2/4, \tag{4}$$

which has exactly two minimal points at $c = \pm 1$, i.e. at the energetically favorable state. Another choice for W is the relaxed double-obstacle free energy

$$W_s^{\text{rel}}(c) = \frac{1}{2}(1 - c^2) + \frac{s}{2}(\max(c - 1, 0)^2 + \min(c + 1, 0)^2), \tag{5}$$

with relaxation parameter $s \gg 0$, which is introduced in [19] as the Moreau-Yosida relaxation of the double obstacle free energy

$$W^\infty(c) = \begin{cases} \frac{1}{2}(1 - c^2), & \text{if } c \in [-1, 1], \\ +\infty, & \text{else.} \end{cases}$$

For more details on the choices for W we refer to [1] and [7], for example. Concerning existence, uniqueness and regularity of a solution to (3), we refer to [7].

In order to derive a variational form of type (1), we write (3) as a single fourth order parabolic equation for c by

$$\begin{cases} c_t(t, x) + y \cdot \nabla c &= m\Delta(-\sigma\varepsilon\Delta c(t, x) + \frac{\sigma}{\varepsilon}W'(c(t, x))) & \text{in } Q, \\ \nabla c(t, x) \cdot \nu_\Omega &= \nabla(-\sigma\varepsilon\Delta c(t, x) + \frac{\sigma}{\varepsilon}W'(c(t, x))) \cdot \nu_\Omega = 0 & \text{on } \Sigma, \\ c(0, x) &= c_0(x) & \text{in } \Omega. \end{cases} \tag{6}$$

We choose $V = H^2 \cap H_0^1(\Omega)$ and $H = L^2(\Omega)$ and introduce the bilinear form $a : V \times V \to \mathbb{R}$ by

$$a(u, v) = \int_\Omega m\sigma\varepsilon\Delta u \cdot \Delta v + y \cdot \nabla c \cdot v$$

and define the nonlinear operator $\mathcal{N}$ by

$$\mathcal{N}(c) = -\frac{\sigma}{\varepsilon}\Delta W'(c).$$

2.2   Temporal and spatial discretization

In order to solve (1) numerically, we apply the implicit Euler method for temporal discretization. Of course, other time integration schemes are possible. For a given $n \in \mathbb{N}$ let

$$0 = t_0 < \ldots < t_n = T \tag{7}$$

denote an arbitrary grid in the time interval $[0, T]$ with time step sizes

$$\Delta t_j := t_j - t_{j-1}$$

for $j = 1, \ldots, n$. We set $I_j = [t_{j-1}, t_j]$ for each time interval $j = 1, \ldots, n$. The resulting time discrete system consists in finding a sequence $\{\bar{y}^j\}_{j=0}^n \subset V$ satisfying the following equation system

$$\begin{cases} \langle \frac{\bar{y}^j - \bar{y}^{j-1}}{\Delta t_j}, v \rangle_H + a(\bar{y}^j, v) + \langle \mathcal{N}(\bar{y}^j), v \rangle_{V', V} &= \frac{1}{\Delta t_j} \int_{t_{j-1}}^{t_j} \langle f_j, v \rangle_{V', V} dt, & \text{for } j = 1, \ldots, n \\ \langle \bar{y}^0, v \rangle_H &= \langle y_0, v \rangle_H, \end{cases} \tag{8}$$

4

for all $v \in V$ and f_j denotes $f_j = f(t_j) \in V'$ for $j = 1, \cdots, n$.

For the spatial discretization we utilize adaptive finite elements. At each time point t_j, $j = 0, \ldots, n$, we introduce a regular triangulation $\mathcal{T}_j$ of $\bar{\Omega}$ and define an N_j-dimensional conformal subspace V^j of V by

$$V^j := \mathrm{span}\{v_1^j, \ldots, v_{N_j}^j\} \subset V$$

with nodal basis $\{v_i^j\}_{i=1}^{N_j}$, i.e. $v_i^j(P_k^j) = \delta_{ik}$ for $i, k = 1, \ldots, N_j$ with the nodes $\{P_k^j\}_{k=1}^{N_j}$ of the underlying triangulation $\mathcal{T}_j$. Therefore, at each time level $j = 0, \ldots, n$, the utilized finite element spaces $\{V^j\}_{j=0}^n$ can differ both in the underlying triangulation of the domain $\bar{\Omega}$ and in the polynomial degree. This means that the solutions can be computed utilizing $h-$, $p-$ and $r-$adaptivity, where $h-$adaptivity denotes local refinement and coarsening of the triangulation according to certain error indicators, $p-$adaptivity means increasing and decreasing the polynomial degree according to the smoothness of the solution and $r-$adaptivity, or moving mesh methods, relocates the mesh points to concentrate them in specific regions. We apply a Galerkin scheme for (8). Thus we look for a sequence $\{y^j\}_{j=0}^n$ with $y^j \in V^j$ which fulfills

$$\begin{cases} \langle \dfrac{y^j - \mathbb{I}^j y^{j-1}}{\Delta t_j}, v \rangle_H + a(y^j, v) + \langle \mathcal{N}(y^j), v \rangle_{V',V} &= \langle f_j, v \rangle_{V',V}, \quad \text{for } j = 1, \ldots, n \\ \langle y^0, v \rangle_H &= \langle y_0, v \rangle_H, \end{cases} \qquad (9)$$

for all $v \in V^j$, where $\mathbb{I}^j : C(\bar{\Omega}) \to V^j$ denotes the Lagrange interpolation. Since $y^j \in V^j$ holds, we make the Galerkin ansatz

$$y^j = \sum_{i=1}^{N^j} \mathsf{y}_i^j v_i^j \in V^j \subset V \qquad (10)$$

for $j = 0, \ldots, n$ with appropriate mode coefficients $\{\mathsf{y}_i^j\}_{i=1}^{N^j}$.

3 POD method utilizing snapshots with arbitrary finite element discretizations

3.1 POD method in real Hilbert spaces

The aim of this work is to propose a POD-ROM which does not depend on the underlying finite element discretization. For this reason, the POD method is explained from an infinite-dimensional perspective in this section, where we use a finite number of snapshots which lie in a common Hilbert space. The POD method in Hilbert spaces is explained in [27] and [43], for example. Here, we recall the main aspects.

Assume we are given snapshots

$$y^0 \in V^0, \ldots, y^n \in V^n$$

of (1), which can be finite element samples of the solution trajectory $\mathcal{V} = \mathrm{span}\{y(t) \mid t \in [0, T]\}$ for (1) on the given timegrid $\{t_j\}_{j=0}^n$ introduced in (7). For each time level $j = 0, \ldots, n$ the snapshots belong to different subspaces $V^0, \ldots, V^n \subset V$. Note that by construction we have $\mathcal{V} := \mathrm{span}\{y^j\}_{j=0}^n \subset V \subset X$, where X denotes either of the spaces V or H.

The idea of the POD method is to describe the space $\mathcal{V}$ by means of few orthonormal functions $\{\psi_i\}_{i=1}^\ell \subset X$, with $\ell \leq d := \dim \mathcal{V}$, such that error between the snapshots $\{y^j\}_{j=0}^n$ and the projection of the snapshots onto the subspace $X^\ell = \mathrm{span}\{\psi_1, ..., \psi_\ell\} \subset X$ is minimized in the following sense:

$$\min_{\psi_1, \ldots, \psi_\ell \in X} \sum_{j=0}^n \alpha_j \left\| y^j - \sum_{i=1}^\ell \langle y^j, \psi_i \rangle_X \, \psi_i \right\|_X^2 \quad \text{s.t. } \langle \psi_i, \psi_j \rangle_X = \delta_{ij} \quad \text{for } 1 \leq i, j \leq \ell \qquad (11)$$

with e.g. nonnegative trapezoidal weights $\{\alpha_j\}_{j=0}^n$:

$$\alpha_0 = \frac{\Delta t_1}{2}, \ \alpha_j = \frac{\Delta t_j + \Delta t_{j+1}}{2} \text{ for } j = 1, \dots, n-1 \text{ and } \alpha_n = \frac{\Delta t_n}{2}. \tag{12}$$

A solution to (11) is called a rank-ℓ POD basis. For this equality constrained minimization problem (11), first order necessary (and by convexity sufficient) optimality conditions can be derived. For this purpose, we introduce the bounded linear operator $\mathcal{Y} : \mathbb{R}^{n+1} \to X$ by

$$\mathcal{Y}\phi = \sum_{j=0}^n \sqrt{\alpha_j}\phi_j y^j \quad \text{for } \phi = (\phi_0, \dots, \phi_n) \in \mathbb{R}^{n+1}.$$

Since the image $\mathcal{Y}(X) = \operatorname{span}\{y^0, \dots, y^n\}$ has finite dimension, the operator $\mathcal{Y}$ is compact. Its Hilbert space adjoint $\mathcal{Y}^\star : X \to \mathbb{R}^{n+1}$ satisfies $\langle \mathcal{Y}\phi, \psi \rangle_X = \langle \phi, \mathcal{Y}^\star\psi \rangle_{\mathbb{R}^{n+1}}$ for $\phi \in \mathbb{R}^{n+1}$ and $\psi \in X$ and is given by

$$\mathcal{Y}^\star\psi = \begin{pmatrix} \langle \psi, \sqrt{\alpha_0}y^0 \rangle_X \\ \vdots \\ \langle \psi, \sqrt{\alpha_n}y^n \rangle_X \end{pmatrix} \quad \text{for } \psi \in X.$$

Then, the action of $\mathcal{K} := \mathcal{Y}^\star\mathcal{Y} : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ is given by

$$\mathcal{K}\phi = \begin{pmatrix} \sum_{j=0}^n \sqrt{\alpha_j}\langle \sqrt{\alpha_0}y^0, y^j \rangle_X \phi_j \\ \vdots \\ \sum_{j=0}^n \sqrt{\alpha_j}\langle \sqrt{\alpha_n}y^n, y^j \rangle_X \phi_j \end{pmatrix} \quad \text{for } \phi \in \mathbb{R}^{n+1}.$$

$\mathcal{K}$ can be represented as the symmetric matrix

$$\mathcal{K} = \begin{pmatrix} \sqrt{\alpha_0}\sqrt{\alpha_0}\langle y^0, y^0 \rangle_X & \cdots & \sqrt{\alpha_0}\sqrt{\alpha_n}\langle y^0, y^n \rangle_X \\ \vdots & & \vdots \\ \sqrt{\alpha_n}\sqrt{\alpha_0}\langle y^n, y^0 \rangle_X & \cdots & \sqrt{\alpha_n}\sqrt{\alpha_n}\langle y^n, y^n \rangle_X \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}. \tag{13}$$

We introduce the operator $\mathcal{R} := \mathcal{Y}\mathcal{Y}^\star : X \to X$, whose action is given by

$$\mathcal{R}\psi = \sum_{j=0}^n \alpha_j \langle \psi, y^j \rangle_X y^j \quad \text{for } \psi \in X.$$

It can be shown that the operator $\mathcal{R}$ is bounded, nonnegative and self-adjoint. Since the image $\mathcal{R}(X) = \operatorname{span}\{y^0, \dots, y^n\}$ has finite dimension, the operator $\mathcal{R}$ is compact. Therefore the Hilbert-Schmidt theorem (cf. [38, Th. VI.16], for instance) can be applied which ensures the existence of a complete orthonormal basis $\{\psi_i\}_{i=1}^\infty$ for X and a sequence of corresponding nonnegative eigenvalues $\{\lambda_i\}_{i=1}^\infty$ with

$$\mathcal{R}\psi_i = \lambda_i\psi_i \quad \text{with } \lambda_1 \geq \dots \geq \lambda_d > 0 \text{ and } \lambda_i = 0 \text{ for all } i > d.$$

Likewise, one can compute the eigenvalues $\{\lambda_i\}_{i=1}^d$ of $\mathcal{K}$, which coincide with the eigenvalues for $\mathcal{R}$ except for possibly zero. The corresponding orthonormal eigenvectors $\{\phi_i\}_{i=1}^d \subset \mathbb{R}^{n+1}$ of $\mathcal{K}$ are

$$\phi_i = \frac{1}{\sqrt{\lambda_i}}(\mathcal{Y}^\star\psi_i) = \frac{1}{\sqrt{\lambda_i}} \begin{pmatrix} \langle \psi_i, \sqrt{\alpha_0}y^0 \rangle_X \\ \vdots \\ \langle \psi_i, \sqrt{\alpha_n}y^n \rangle_X \end{pmatrix} \in \mathbb{R}^{n+1} \quad \text{for } i = 1, \dots, d.$$

Thus, the functions $\{\psi_i\}_{i=1}^d$ can be determined via

$$\psi_i = \frac{1}{\sqrt{\lambda_i}}\mathcal{Y}\phi_i = \frac{1}{\sqrt{\lambda_i}}\sum_{j=0}^n \sqrt{\alpha_j}(\phi_i)_j y^j \in X \quad \text{for } i = 1, \dots, d, \tag{14}$$

where $(\phi_i)_j$ denotes the j-th component of $\phi_i \in \mathbb{R}^{n+1}$ for $i = 1, \ldots, d$.

The following theorem states the necessary (and by convexity sufficient) optimality conditions for problem (11) and presents the POD projection error.

Theorem 3.1. Let $\{\lambda_i\}_{i=1}^d$ denote the positive eigenvalues of $\mathcal{R}$, and let $\{\psi_i\}_{i=1}^d \subset X$ denote the corresponding eigenfunctions of $\mathcal{R}$. For every $\ell \in \mathbb{N}$ with $\ell \leq d$, a solution to (11) is given by the eigenfunctions $\{\psi_i\}_{i=1}^\ell$ corresponding to the ℓ largest eigenvalues $\{\lambda_i\}_{i=1}^\ell$. Moreover the projection error is

$$\sum_{j=0}^n \alpha_j \left\| y^j - \sum_{i=1}^\ell \langle y^j, \psi_i \rangle_X \, \psi_i \right\|_X^2 = \sum_{i=\ell+1}^d \lambda_i. \tag{15}$$

Proof. Since $\mathrm{span}\{y^j\}_{j=0}^n \subset X$, the proof runs analogously to the proof in [43, Th. 3]. $\qquad\square$

The basis $\{\psi_i\}_{i=1}^\ell$ can alternatively be computed via singular value decomposition (SVD). The SVD of the operator $\mathcal{Y}$ is given by

$$\mathcal{Y} = \sum_{i=1}^d \sigma_i \sqrt{\alpha_i} \langle \cdot, \phi_i \rangle_{\mathbb{R}^{n+1}} \psi_i,$$

where $\sigma_1 \geq \ldots \geq \sigma_d > 0$ is the ordered sequence of singular values of $\mathcal{Y}$ with $\sigma_i = \sqrt{\lambda_i}$ for $i = 1, \ldots, d$. For more details we refer to [38, Th. VI.17], for instance.

3.2 Numerical realization of the POD method utilizing snapshots from arbitrary finite element spaces

Let us now turn our perspective to the numerical realization of computing a POD basis for snapshots which live in arbitrary finite element spaces. For each time level $j = 0, \ldots, n$, the snapshots $\{y^j\}_{j=0}^n$ shall be taken from different finite element spaces $\{V^j\}_{j=0}^n$ which lie in a common Hilbert space V. In the fully discrete fomulation of the POD method we are given the evaluation of the snapshots on their corresponding grids, i.e. we are given the vectors

$$\mathsf{y}^0 \in \mathbb{R}^{N_0}, \ldots, \mathsf{y}^n \in \mathbb{R}^{N_n}$$

of different lengths with $\mathsf{y}^j = (\mathsf{y}_1^j, \ldots, \mathsf{y}_{N_j}^j)^T \in \mathbb{R}^{N_j}$, for $j = 0, \ldots, n$. This is why we are not able to set up the discrete counterpart to the operator $\mathcal{R}$, which is an $N \times N$ matrix for uniform spatial discretization with N nodes. Moreover, the representation of the POD basis as a linear combination of the snapshots is no longer possible.

To overcome this obstacle, our aim is to set up a reduced order model which can be formulated for arbitrary finite element discretizations. For this reason, we turn our attention to the matrix $\mathcal{K} \in \mathbb{R}^{(n+1)\times(n+1)}$. This matrix dimension only depends on the number of snapshots and can be computed for any underlying finite element discretization: the ij−th component $\mathcal{K}_{ij}$, for $i, j = 0 \ldots, n$, is given by

$$\sqrt{\alpha_i}\sqrt{\alpha_j}\langle y^i, y^j \rangle_X = \sqrt{\alpha_i}\sqrt{\alpha_j}\langle \sum_{k=1}^{N_i} \mathsf{y}_k^i v_k^i, \sum_{l=1}^{N_j} \mathsf{y}_l^j v_l^j \rangle_X = \sqrt{\alpha_i}\sqrt{\alpha_j} \sum_{k=1}^{N_i}\sum_{l=1}^{N_j} \mathsf{y}_k^i \mathsf{y}_l^j \langle v_k^i, v_l^j \rangle_X.$$

Thus for any $i, j = 0, \ldots, n$, we are able to compute the inner product $\langle v_k^i, v_l^j \rangle_X$ for $k = 1, \ldots, N_i$ and $l = 1, \ldots, N_j$, compare Figure 1. We discover, then, that the calculation of the matrix $\mathcal{K}$ as well as the determination of its eigenvectors can be done for any underlying finite element discretization. Thus, the eigenvectors $\{\phi_i\}_{i=1}^d$ of $\mathcal{K}$ are the right singular vectors of $\mathcal{Y}$ and contain the space independent time information. This fact will be used in the following to build up the reduced order model. We note that the calculation of the matrix $\mathcal{K}$ can be done for arbitrary finite element grids, i.e. all kinds of adaptivity (h-, p- and r-adaptivity) can be considered and we do not need a common reference mesh. The complexity of this methods lies in the computation of
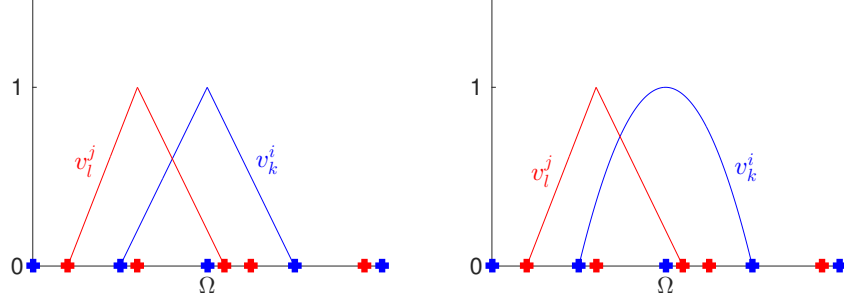
Figure 1: *1D finite element basis functions v_l^j and v_k^i on their corresponding grids. Left: both piecewise linear finite element ansatz functions. Right: piecewise linear and cubic finite element ansatz functions.*

the inner products of the intersections of the finite elements, whereas in [42] the challenge lies in lifting the snapshots from different finite element spaces up to a common finite element space.

Moreover, the calculation of the matrix $\mathcal{K} \in \mathbb{R}^{(n+1)\times(n+1)}$ (method of snapshots) is favorable, since we assume the temporal dimension n to be far smaller than the spatial dimension(s). Due to the symmetry of $\mathcal{K}$ it suffices to compute the entries on and upon the diagonal of the matrix. The computations of the matrix entries of $\mathcal{K}$ can be done fully in parallel. Hence, if sufficiently good hardware is available, setting up the matrix $\mathcal{K}$ can be done very fast.

Example 3.2. We choose $V = H^1(\Omega), H = L^2(\Omega)$ and set $X = H = L^2(\Omega)$. The triangulations of $\bar{\Omega}$ for each time level $j = 0, \ldots, n$ are denoted by $\{\mathcal{T}_j\}_{j=0}^n$ and the finite element spaces are defined by

$$V^j = V(\mathcal{T}_j) = \{v \in C^0(\bar{\Omega}) : v|_T \in \mathbb{P}_r(T), \forall T \in \mathcal{T}_j\} \subset X, \quad j = 0, \ldots, n,$$

where $\mathbb{P}_r$ denotes the space of polynomials of degree $r \in \mathbb{N}$. The computation of the ij-th entry $\mathcal{K}_{ij} = \sqrt{\alpha_i}\sqrt{\alpha_j}\langle y^i, y^j\rangle_{L^2(\Omega)}$ of the matrix $\mathcal{K}$ is calculated by

$$
\begin{aligned}
\sqrt{\alpha_i}\sqrt{\alpha_j}\langle y^i, y^j\rangle_{L^2(\Omega)} &= \sqrt{\alpha_i}\sqrt{\alpha_j}\int_\Omega y^i y^j \, dx \\
&= \sqrt{\alpha_i}\sqrt{\alpha_j}\sum_{k=1}^{N^i}\sum_{l=1}^{N^j} \mathsf{y}_k^i \mathsf{y}_l^j \int_\Omega v_k^i v_l^j \, dx \\
&= \sqrt{\alpha_i}\sqrt{\alpha_j}\sum_{k=1}^{N^i}\sum_{l=1}^{N^j} \mathsf{y}_k^i \mathsf{y}_l^j \left(\sum_{T \in \mathcal{T}_i}\sum_{\bar{T} \in \mathcal{T}_j} \int_{T \cap \bar{T}} v_k^i v_l^j \, dx \right).
\end{aligned}
$$

Computations become simpler when using nested grids. In this case, the intersection of two arbitrary n-dimensional simplices coincides either with the smaller simplex, or is a common edge simplex, or has no overlap.

4 POD reduced order modeling

4.1 POD reduced order modeling for arbitrary finite element discretizations

In this section, we stay in the infinite dimensional setting of the POD method and set up the POD reduced order model utilizing snapshots with arbitrary finite element discretizations. This perspective allows us to determine the mode coefficients of the POD Galerkin ansatz for arbitrary underlying finite element discretizations. Suppose for given snapshots $y^0 \in V_0, \ldots, y^n \in V^n$ we have computed the matrix $\mathcal{K}$, (13), with $\mathcal{K}_{ij} = \sqrt{\alpha_i}\sqrt{\alpha_j}\langle y^i, y^j\rangle_X$, for $i, j = 0, \ldots, n$ as well as its ℓ largest eigenvalues $\{\lambda_i\}_{i=1}^\ell$ and corresponding eigenvectors $\{\phi_i\}_{i=1}^\ell \subset \mathbb{R}^{n+1}$ of low rank ℓ with

$\ell \le n + 1$, according to the strategy presented in Section 3. The POD basis $\{\psi_i\}_{i=1}^{\ell}$ is then given by (14), i.e

$$\psi_i = \frac{1}{\sqrt{\lambda_i}} \mathcal{Y} \phi_i \quad \text{for } i = 1, \ldots, \ell.$$

This POD basis is utilized in order to compute a reduced order model for (1). For this reason, we make the POD Galerkin ansatz

$$y^{\ell}(t) = \sum_{i=1}^{\ell} \eta_i(t) \psi_i = \sum_{i=1}^{\ell} \eta_i(t) \frac{1}{\sqrt{\lambda_i}} \mathcal{Y} \phi_i \quad \text{for all } t \in [0, T], \tag{16}$$

as an approximation for $y(t)$, with the Fourier coefficients

$$\eta_i(t) = \langle y^{\ell}(t), \psi_i \rangle_X = \langle y^{\ell}(t), \frac{1}{\sqrt{\lambda_i}} \mathcal{Y} \phi_i \rangle_X$$

for $i = 1, \ldots, \ell$. Inserting y^{ℓ} into (1) and choosing $X^{\ell} = \text{span}\{\psi_1, \ldots, \psi_{\ell}\} \subset X$ as the test space leads to the system

$$\begin{cases} \dfrac{d}{dt} \langle y^{\ell}(t), \psi \rangle_H + a(y^{\ell}(t), \psi) + \langle \mathcal{N}(y^{\ell}(t)), \psi \rangle_{V', V} &= \langle f(t), \psi \rangle_{V', V} \\ \langle y^{\ell}(0), \psi \rangle_H &= \langle y_0, \psi \rangle_H \end{cases} \tag{17}$$

for all $\psi \in X^{\ell}$ and for almost all $t \in (0, T]$. Utilizing the ansatz (16), we can write (17) as an ℓ-dimensional ordinary differential equation system for the POD mode coefficients $\{\eta_i(t)\}_{i=1}^{\ell} \subset \mathbb{R}$:

$$\begin{cases} \displaystyle\sum_{j=1}^{\ell} \dot{\eta}_j(t) \langle \psi_i, \psi_j \rangle_H + \sum_{j=1}^{\ell} \eta_j(t) \langle \mathcal{A} \psi_j, \psi_i \rangle_{V', V} + \langle \mathcal{N}(y^{\ell}(t)), \psi_i \rangle_{V', V} &= \langle f(t), \psi_i \rangle_{V', V} \\ &\qquad\qquad \text{for } t \in (0, T], \\ \displaystyle\sum_{j=1}^{\ell} \eta_j(0) \langle \psi_i, \psi_j \rangle_H &= \langle y_0, \psi_i \rangle_H, \end{cases} \tag{18}$$

for $i = 1, \ldots, \ell$. Note that $\langle \psi_i, \psi_j \rangle_H = \delta_{ij}$ if we choose $X = H$ in the context of Section 3.

Since we want to construct a reduced order model which can be built and solved for arbitrary finite element discretizations, we rewrite system (18) utilizing the identity (14). Then, the system (18) can be written as

$$\begin{cases} \displaystyle\sum_{j=1}^{\ell} \dot{\eta}_j(t) \frac{1}{\sqrt{\lambda_i}} \frac{1}{\sqrt{\lambda_j}} \langle \mathcal{Y} \phi_i, \mathcal{Y} \phi_j \rangle_H + \sum_{j=1}^{\ell} \eta_j(t) \frac{1}{\sqrt{\lambda_i}} \frac{1}{\sqrt{\lambda_j}} \langle \mathcal{A} \mathcal{Y} \phi_j, \mathcal{Y} \phi_i \rangle_{V', V} \\ \qquad\qquad\qquad + \dfrac{1}{\sqrt{\lambda_i}} \langle \mathcal{N}(y^{\ell}(t)), \mathcal{Y} \phi_i \rangle_{V', V} = \dfrac{1}{\sqrt{\lambda_i}} \langle f(t), \mathcal{Y} \phi_i \rangle_{V', V} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for } t \in (0, T], \\ \displaystyle\sum_{j=1}^{\ell} \eta_j(0) \frac{1}{\sqrt{\lambda_i}} \frac{1}{\sqrt{\lambda_j}} \langle \mathcal{Y} \phi_i, \mathcal{Y} \phi_j \rangle_H = \dfrac{1}{\sqrt{\lambda_i}} \langle y_0, \mathcal{Y} \phi_i \rangle_H. \end{cases} \tag{19}$$

In order to write (19) in a compact matrix-vector form, let us introduce the diagonal matrix

$$D \in \mathbb{R}^{\ell \times \ell} \text{ with } D = \text{diag}\left(\frac{1}{\sqrt{\lambda_1}}, \ldots, \frac{1}{\sqrt{\lambda_{\ell}}} \right).$$

From the first ℓ eigenvectors $\{\phi_i\}_{i=1}^{\ell}$ of $\mathcal{K}$ we build the matrix

$$\Phi \in \mathbb{R}^{(n+1) \times \ell} \text{ by } \Phi = [\phi_1 \mid \ldots \mid \phi_{\ell}].$$

Then, the system (19) can be written as the system

$$\begin{cases} D\Phi^T \mathcal{K} \Phi D \, \dot{\eta}(t) + D\Phi^T \mathcal{Y}^{\star} \mathcal{A} \mathcal{Y} \Phi D \, \eta(t) + DN(\eta(t)) &= DF(t) \quad \text{for } t \in (0, T], \\ D\Phi^T \mathcal{K} \Phi D \, \eta(0) &= D\bar{\eta}_0, \end{cases} \tag{20}$$

for the vector-valued mapping $\eta(t) = (\eta_1(t), \ldots, \eta_\ell(t))^T : [0, T] \to \mathbb{R}^\ell$. Note that the right hand side $F(t)$ and the initial condition $\bar{\eta}_0$ are given by

$$(F(t))_i = \langle f(t), \mathcal{Y}\phi_i \rangle_{V', V} = \langle \mathcal{Y}^\star f(t), \phi_i \rangle_{\mathbb{R}^{n+1}}$$

and

$$(\bar{\eta}_0)_i = \langle y_0, \mathcal{Y}\phi_i \rangle_H = \langle \mathcal{Y}^\star y_0, \phi_i \rangle_{\mathbb{R}^{n+1}},$$

for $i = 1, \ldots, \ell$, respectively. Their calculation can be done explicitly for any arbitrary finite element discretization. For a given function $w \in V$ (for example $w = f(t)$ or $w = y_0$) with finite element discretization $w = \sum_{i=1}^{N_w} \mathsf{w}_i \varphi_i$, nodal basis $\{\varphi_i\}_{i=1}^{N_w} \subset V$ and appropriate mode coefficients $\{\mathsf{w}_i\}_{i=1}^{N_w}$ we can compute

$$(\mathcal{Y}^\star w)_j = \langle w, y^j \rangle_X = \langle \sum_{i=1}^{N_w} \mathsf{w}_i \varphi_i, \sum_{k=1}^{N_j} \mathsf{y}_k^j v_k^j \rangle_X = \sum_{i=1}^{N_w} \sum_{k=1}^{N_j} \mathsf{w}_i \mathsf{y}_k^j \langle \varphi_i, v_k^j \rangle_X, \quad \text{for } j = 0, \ldots, n$$

where $y^j \in V^j$ denotes the j-th snapshot. Again, for any $i = 1, \ldots, N_w$ and $k = 1, \ldots, N_j$, the computation of the inner product $\langle \varphi_i, v_k^j \rangle_X$ can be done explicitly.

Remark 4.1. Of course, the presented approach is also applicable to other classes of differential equations like elliptic PDEs, for example.

Obviously, for linear evolution equations the POD reduced order model (20) can be set up and solved utilizing snapshots with arbitrary finite element discretizations. The computation of the nonlinear component $N(\eta(t))$ needs particular attention. In the following Section 4.2 we discuss the options to treat the nonlinearity.

Time discrete reduced order model

In order to solve the reduced order system (17) numerically, we apply the implicit Euler method for time discretization and use for simplicity the same temporal grid $\{t_j\}_{j=0}^n$ (7) as for the snapshots. It is also possible to use a different time grid, cf. [27]. The time discrete reduced order model reads

$$\begin{cases} \left\langle \dfrac{y_j^\ell - y_{j-1}^\ell}{\Delta t_j}, \psi \right\rangle_H + a(y_j^\ell, \psi) + \langle \mathcal{N}(y_j^\ell), \psi \rangle_{V', V} &= \dfrac{1}{\Delta t_j} \int_{t_{j-1}}^{t_j} \langle f_j, \psi \rangle_{V', V} \quad \text{for all } \psi \in X^\ell, \\ \langle y_0^\ell, \psi \rangle_H &= \langle y_0, \psi \rangle_H, \end{cases} \tag{21}$$

for $j = 1, \ldots, n$ or equivalently

$$\begin{cases} D\Phi^T \mathcal{K}\Phi D \left(\dfrac{\eta^j - \eta^{j-1}}{\Delta t_j} \right) + D\Phi^T \mathcal{Y}^\star \mathcal{A} \mathcal{Y} \Phi D \, \eta^j + DN(\eta^j) &= DF_j \quad \text{for } j = 1, \ldots, n, \\ D\Phi^T \mathcal{K}\Phi D \, \eta^0 &= D\bar{\eta}_0. \end{cases} \tag{22}$$

4.2 Discussion of the nonlinear term $DN(\eta(t))$

Let us now consider the computation of the nonlinear term $DN(\eta(t)) \in \mathbb{R}^\ell$ of the POD-ROM (20). It holds true

$$\begin{aligned} (DN(\eta(t)))_k &= \langle \mathcal{N}(y^\ell(t)), \psi_k \rangle_{V', V} \\ &= \langle \mathcal{N}(\sum_{i=1}^\ell \eta_i(t)\psi_i), \psi_k \rangle_{V', V} \end{aligned}$$

for $k = 1, \cdots, \ell$. It is well-known that the evaluation of nonlinearities in the reduced order model context in computationally expensive. To make this clear, let us assume, we are given a uniform finite element discretization with N degrees of freedom. Then, in the fully discrete setting, the nonlinear term has the form

$$\Psi^T W \mathcal{N}(\Psi \eta(t)),$$

where $\Psi = [\psi_1 \mid \cdots \mid \psi_\ell] \in \mathbb{R}^{N \times \ell}$ is the matrix in which the POD modes are stored columnwise and $W \in \mathbb{R}^{N \times N}$ is a weighting matrix related to the utilized inner product. Hence, the treatment

10

of the nonlinearity requires the expansion of $y^\ell(t) = \Psi\eta(t) \in \mathbb{R}^N$ in the full space, then the nonlinearity can be evaluated and finally the result is projected back to the POD space. Obviously, this means that the reduced order model is not fully independent of the high order dimension and efficient simulation cannot be guaranteed. Therefore, it is convenient to seek for hyper reduction, i.e. for a treatment of the nonlinearity where the model evaluation cost is related to the low dimension ℓ. A possible remedy which is commonly used, is given by empirical interpolation methods, for which we refer to [8] for EIM, to [12] for DEIM and to [13] for Q-DEIM. Another option is given by [4] which investigates nonlinear model reduction via dynamic mode decomposition. Furthermore, in [45] nonlinear model reduction is realized by replacing the nonlinear term by its interpolation into the finite element space. The treatment of the nonlinearity is done in [6] by the missing point estimation method and in [32] by best the points interpolation method.

All these methods have in common, that a common reference mesh is needed. Since we want to circumvent the interpolation of the snapshots onto a common grid and we want to set up the reduced order model for arbitrary finite element discretizations, we have to go a different way.

Let us consider the linear reduced order system given by:

$$\begin{cases} \frac{d}{dt}\langle y^\ell(t), \psi\rangle_H + a(y^\ell(t), \psi) + \langle \mathcal{N}(\bar{y}(t)), \psi\rangle_{V',V} &= \langle f(t), v\rangle_{V',V}, \\ \langle y^\ell(0), \psi\rangle_H &= \langle y_0, v\rangle_H, \end{cases} \tag{23}$$

For a given state $\bar{y}(t) \in V$, this linear evolution problem (23) can be set up and solved explicitly without spatial interpolation. In the numerical examples in Section 6, we take the finite element solution as given state in each time step, i.e. $\bar{y}(t_j) = y^j$.

Furthermore, the linearization of the reduced order model (17) can be considered:

$$\begin{cases} \frac{d}{dt}\langle y^\ell(t), \psi\rangle_H + a(y^\ell(t), \psi) + \langle \mathcal{N}(\bar{y}(t)) + \mathcal{N}_y(\bar{y}(t))(y^\ell - \bar{y})(t), \psi\rangle_{V',V} &= \langle f(t), v\rangle_{V',V}, \\ \langle y^\ell(0), \psi\rangle_H &= \langle y_0, v\rangle_H, \end{cases} \tag{24}$$

where $\mathcal{N}_y$ denotes the Fréchet derivative. This linearized problem is of interest in the context of optimal control, where it occurs in each iteration level within the SQP method, see [24], for example. Choosing the finite element solution as given state in each time instance leads to:

$$\begin{aligned} \langle \mathcal{N}(y^j), \psi_i\rangle_{V',V} &= \frac{1}{\sqrt{\lambda_i}} \sum_{k=0}^n \sqrt{\alpha_k}(\phi_k)_k \langle \mathcal{N}(y^j), y^k\rangle_{V',V} \\ \langle \mathcal{N}_y(y^j)y^\ell(t_j), \psi_i\rangle_{V',V} &= \langle \mathcal{N}_y(y^j)(\sum_{k=1}^\ell \eta_k(t_j)\psi_k), \psi_i\rangle_{V',V} \\ &= \sum_{k=1}^\ell \eta_k(t_j) \frac{1}{\sqrt{\lambda_k}} \frac{1}{\sqrt{\lambda_i}} \sum_{p=0}^n \sum_{r=0}^n \sqrt{\alpha_p}\sqrt{\alpha_r}(\phi_k)_p(\phi_i)_r \langle \mathcal{N}_y(y^j)y^p, y^r\rangle_{V',V} \\ \langle \mathcal{N}_y(y^j)y^j, \psi_i\rangle_{V',V} &= \frac{1}{\sqrt{\lambda_i}} \sum_{k=0}^n \sqrt{\alpha_k}(\phi_i)_k \langle \mathcal{N}_y(y^j)y^j, y^k\rangle_{V',V} \end{aligned}$$

Finally, we approximate the nonlinearity $DN(\eta^j)$ in (22) by

$$(DN(\eta^j))_k \approx \langle \mathcal{N}(y^j) + \mathcal{N}_y(y^j)(y^\ell(t_j) - y^j), \psi_k\rangle_{V',V}$$

which can be written as

$$DN(\eta^j) \approx D\Phi^T \mathsf{N}^j + D\Phi^T \mathsf{N}_y^j \Phi D\eta^j - D\Phi^T \mathsf{N}_y^j$$

where

$$\mathsf{N}^j = \begin{pmatrix} \langle \mathcal{N}(y^j), \sqrt{\alpha_0}y^0\rangle_{V',V} \\ \vdots \\ \mathcal{N}(y^j), \sqrt{\alpha_n}y^n\rangle_{V',V} \end{pmatrix} \in \mathbb{R}^{n+1}, \mathsf{N}_y^j = \begin{pmatrix} \langle \mathcal{N}_y(y^j)y^j, \sqrt{\alpha_0}y^0\rangle_{V',V} \\ \vdots \\ \mathcal{N}_y(y^j)y^j, \sqrt{\alpha_n}y^n\rangle_{V',V} \end{pmatrix} \in \mathbb{R}^{n+1}$$

and

$$\mathbb{N}_y^j = \begin{pmatrix} \langle \mathcal{N}_y(y^j)\sqrt{\alpha_0}y^0, \sqrt{\alpha_0}y^0\rangle_{V',V} & \cdots & \langle \mathcal{N}_y(y^j)\sqrt{\alpha_n}y^n, \sqrt{\alpha_0}y^0\rangle_{V',V} \\ \vdots & & \vdots \\ \langle \mathcal{N}_y(y^j)\sqrt{\alpha_0}y^0, \sqrt{\alpha_n}y^n\rangle_{V',V} & \cdots & \langle \mathcal{N}_y(y^j)\sqrt{\alpha_n}y^n, \sqrt{\alpha_n}y^n\rangle_{V',V} \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}$$

For weakly nonlinear systems this approximation may be sufficient, depending on the problem and its goal. A great advantage of linearizing the semilinear partial differential equation is that only linear equations need to be solved which leads to a further speedup. However, if a more precise approximation is desired or necessary, we can think of approximations including higher order terms, like quadratic approximation, see e.g. [10] and [39], or Taylor expansions, see e.g. [34], [35] and [16]. Nevertheless, the efficiency of higher order approximations is limited due to growing memory and computational costs.

4.3 Expressing the POD solution in the full spatial domain

Having determined the solution $\eta(t)$ to (20), we can set up the reduced solution $y^\ell(t)$ in a continuous framework:

$$y^\ell(t) = \sum_{i=1}^{\ell} \eta_i(t) \left(\frac{1}{\sqrt{\lambda_i}} \sum_{j=0}^{n} \sqrt{\alpha_j} (\phi_i)_j y^j \right). \tag{25}$$

Now, let us turn to the fully discrete formulation of (25). For a time-discrete setting, we introduce for simplicity the same temporal grid $\{t_j\}_{j=0}^n$ as for the snapshots. Let us recall the spatial discretization of the snapshots (10) utilizing arbitrary finite elements

$$y^j = \sum_{k=1}^{N_j} \mathsf{y}_k^j v_k^j \quad \text{for } j = 0, \dots, n.$$

Let $\{Q_r^j\}_{r=1}^{M_j}$ denote an arbitrary set of grid points for the reduced system at time level t_j. The fully discrete POD solution can be computed by evaluation:

$$y^\ell(t_j, Q_r^j) = \sum_{i=1}^{\ell} \eta_i(t_j) \left(\frac{1}{\sqrt{\lambda_i}} \sum_{s=0}^{n} \sqrt{\alpha_s} (\phi_i)_s (\sum_{k=1}^{N_s} \mathsf{y}_k^s v_k^s(Q_r^j)) \right) \tag{26}$$

for $j = 0, \dots, n$ and $r = 1, \dots, M_j$. This allows us to use any grid in the POD simulation. For example we can use the same node points at time level j for the POD simulation as we have used for the snapshots, i.e. for $j = 0, \dots, n$ it holds $M_j = N_j$ and $Q_r^j = P_k^j$ for all $r, k = 1, \dots, N_j$. Another option can be to choose

$$\{Q_r^j\}_{r=1}^{M_j} = \bigcup_{j=0}^{n} \bigcup_{k=1}^{N_j} \{P_k^j\},$$

i.e. the common finest grid. Obviously, a special and probably the easiest case concerning the implementation is to choose snapshots which are expressed with respect to the same finite element basis functions and utilize the common finest grid for the simulation of the reduced order system, which is proposed by [42]. After expressing the adaptively sampled snapshots with respect to a common finite element space, the subsequent steps coincide with the common approach of taking snapshots which are generated without adaptivity. Then, expression (26) simplifies to

$$y^\ell(t_j, P_r^j) = \sum_{i=1}^{\ell} \eta_i(t_j) \left(\frac{1}{\sqrt{\lambda_i}} \sum_{s=0}^{n} \sqrt{\alpha_s} (\phi_i)_s \mathsf{y}^s \right). \tag{27}$$

5 Error analysis for the reduced order model

For the validation of the approximation quality of the POD reduced order model, we are interested in analyzing the error between the POD solution and the true solution. Our aim is to estimate the expression

$$\sum_{j=0}^{n} \alpha_j \|y(t_j) - y_j^\ell\|_H^2$$

where $\{y(t_j)\}_{j=0}^n \subset X$ denotes the true solution for (1) at time instances $\{t_j\}_{j=0}^n$ and $\{y_j^\ell\}_{j=0}^n$ is the solution to the time discrete reduced order model (21), i.e.

$$y_j^\ell = \sum_{i=1}^\ell \eta_i^j \psi_i, \tag{28}$$

with the POD basis $\{\psi_i\}_{i=1}^\ell$ computed from the snapshots $\{y^j\}_{j=0}^n$ from arbitrary finite element spaces, i.e. $y^j \in V^j$ for $j = 0, \ldots, n$, as explained in Section 3. The weights $\{\alpha_j\}_{j=0}^n$ are trapezoidal time weights, see (12). We choose $X = V$ in the context of Section 3. Let us introduce the orthogonal projection $\mathcal{P}^\ell : V \to V^\ell$ by

$$\mathcal{P}^\ell v = \sum_{i=1}^\ell \langle v, \psi_i \rangle_V \psi_i \quad \text{for all } v \in V.$$

It holds true $\|\mathcal{P}^\ell\|_{\mathcal{L}(V)} = 1$, where $\mathcal{L}(V)$ is the space of linear bounded operators from V to V. The subsequent calculations follow closely the proofs of [27, Thm. 4.7], [44, Thm. 3.2.5] and [26, Th. 7]. In our situation, we compute the POD basis corresponding to the fully discrete snapshots $\{y^j\}_{j=0}^n$ utilizing adaptive finite element spaces, whereas in [27], [44] and [26], the POD basis is computed from snapshots corresponding to the solution trajectory at the given time instances. For this reason, in the following estimation an additional term corresponding to the error for the spatial discretization will appear.

We make use of the decomposition

$$y(t_j) - y_j^\ell = y(t_j) - y^j + y^j - \mathcal{P}^\ell y^j + \mathcal{P}^\ell y^j - \mathcal{P}^\ell y(t_j) + \mathcal{P}^\ell y(t_j) - y_j^\ell = \eta_j + \varrho_j + \zeta_j + \vartheta_j \tag{29}$$

for $j = 0, \ldots, n$, where $\eta_j := y(t_j) - y^j$, $\varrho_j := y^j - \mathcal{P}^\ell y^j$, $\zeta_j = \mathcal{P}^\ell y^j - \mathcal{P}^\ell y(t_j)$ and $\vartheta_j := \mathcal{P}^\ell y(t_j) - y_j^\ell$. The term η_j is the discretization error. We utilize the decomposition

$$\eta_j = y(t_j) - y^j = y(t_j) - \bar{y}^j + \bar{y}^j - y^j = E_t^j + E_h^j$$

where $\bar{y}^j$ denotes the solution to the time discrete problem (8). By $E_t^j := y(t_j) - \bar{y}^j$ we denote the global time discretization error and $E_h^j := \bar{y}^j - y^j$ is the global spatial discretization error. It is

$$\|E_h^j\|_H \le \max_{j=0,\ldots,n} \|E_h^j\|_H =: \varepsilon_h$$

and

$$\|E_t^j\|_H \le \max_{j=0,\ldots,n} \|E_t^j\|_H =: \varepsilon_t.$$

Since we use the implicit Euler method for time integration, it is $\varepsilon_t = \mathcal{O}(\Delta t)$ with $\Delta t := \max_{j=0,\ldots,n} \Delta t_j$. Therefore, we can estimate

$$\sum_{j=0}^n \alpha_j \|y(t_j) - y^j\|_H^2 \le \sum_{j=0}^n \alpha_j \|E_t^j + E_h^j\|_H^2 \le 2\sum_{j=0}^n \alpha_j ((\Delta t)^2 + \varepsilon_h^2) \le 2T((\Delta t)^2 + \varepsilon_h^2). \tag{30}$$

Moreover, we have

$$\sum_{j=0}^n \alpha_j \|\zeta_j\|_H^2 = \sum_{j=0}^n \alpha_j \|\mathcal{P}^\ell y^j - \mathcal{P}^\ell y(t_j)\|_H^2 \le \|\mathcal{P}^\ell\|_{\mathcal{L}(H)}^2 \sum_{j=0}^n \alpha_j \|\eta_j\|_H^2. \tag{31}$$

The term ϱ_j is the projection error of the snapshot y^j projected onto the POD space V^ℓ. The weighted sum of all projection errors is given by the sum of the neglected eigenvalues (15), i.e.

$$\sum_{j=0}^n \alpha_j \|\varrho_j\|_H^2 = \sum_{j=0}^n \alpha_j \|y^j - \sum_{i=1}^\ell \langle y^j, \psi_i \rangle_V \psi_i\|_H^2 \le c_v \sum_{i=\ell+1}^d \lambda_i. \tag{32}$$

It remains to estimate the term ϑ_j which is the error between the projection of the true solution $y(t_j)$ at time instance t_j onto the POD space V^ℓ and the time discrete ROM solution y_j^ℓ to (28). With the use of the notation $\bar{\partial}\vartheta_j = (\vartheta_j - \vartheta_{j-1})/\Delta t_j$ for $j = 1, \ldots, n$, we get

$$
\begin{aligned}
\langle \bar{\partial}\vartheta_j, \psi \rangle_H &= \langle \mathcal{P}^\ell \left(\frac{y(t_j) - y(t_{j-1})}{\Delta t_j} \right) - \frac{y_j^\ell - y_{j-1}^\ell}{\Delta t_j}, \psi \rangle_H \\
&= \langle \mathcal{P}^\ell \left(\frac{y(t_j) - y(t_{j-1})}{\Delta t_j} \right) + \mathcal{N}(y_j^\ell) - f_j, \psi \rangle_H + a(y_j^\ell, \psi) \\
&= \langle \mathcal{P}^\ell \left(\frac{y(t_j) - y(t_{j-1})}{\Delta t_j} \right) - \frac{y(t_j) - y(t_{j-1})}{\Delta t_j} + \mathcal{N}(y_j^\ell) - \mathcal{N}(y(t_j)), \psi \rangle_H + a(y_j^\ell - y(t_j), \psi) \\
&= \langle z_j + \mathcal{N}(y_j^\ell) - \mathcal{N}(y(t_j)), \psi \rangle_H + a(y_j^\ell - y(t_j), \psi)
\end{aligned}
$$

for $\psi \in V^\ell$ with $z_j := \mathcal{P}\left(\frac{y(t_j) - y(t_{j-1})}{\Delta t_j} \right) - \frac{y(t_j) - y(t_{j-1})}{\Delta t_j}$. With the choice $\psi = \vartheta_j$ and the use of the identity

$$
2\langle u - v, u \rangle = \|u\|^2 - \|v\|^2 + \|u - v\|^2
$$

we obtain

$$
\|\vartheta_j\|_H^2 \leq \|\vartheta_{j-1}\|_H^2 + 2\Delta t_j \left(\beta \|y_j^\ell - y(t_j)\|_H + \|z_j\|_H + \|\mathcal{N}(y_j^\ell) - \mathcal{N}(y(t_j))\|_H \right) \|\vartheta_j\|_H.
$$

We assume that $\mathcal{N}$ is Lipschitz continuous, i.e. there exists $L > 0$ such that

$$
\|\mathcal{N}(y_j^\ell) - \mathcal{N}(y(t_j))\|_H \leq L \|y_j^\ell - y(t_j)\|_H \quad \text{for } j = 1, \ldots, n.
$$

Applying Young's inequality we find

$$
\|\vartheta_j\|_H^2 \leq \|\vartheta_{j-1}\|_H^2 + \Delta t_j (c_1 \|\varrho_j\|_H^2 + c_2 \|\vartheta_j\|_H^2 + \|z_j\|_H^2 + c_1 \|\eta_j\|_H^2 + c_1 \|\zeta_j\|_H^2)
$$

with the constants $c_1 := \beta + L$ and $c_2 := 5(\beta + L) + 1$. Under the assumption that Δt is sufficiently small, we conclude

$$
\|\vartheta_j\|_H^2 \leq e^{2c_2 j \Delta t} \left(\|\vartheta_1\|_H^2 + \sum_{k=1}^{j} \Delta t_k (\|z_k\|_H^2 + c_1 \|\varrho_k\|_H^2 + c_1 \|\eta_k\|_H^2 + c_1 \|\zeta_k\|_H^2) \right). \tag{33}
$$

For more details on this, we refer to [27] and [44]. We choose the initial condition for (21) such that $\vartheta_0 = \mathcal{P}^\ell y(t_0) - y_0^\ell = \mathcal{P}^\ell y_0 - y_0^\ell = 0$.

Next, we estimate the term involving z_k. It holds true

$$
\begin{aligned}
\|z_k\|_H^2 &= \|\mathcal{P}^\ell \left(\frac{y(t_k) - y(t_{k-1})}{\Delta t_k} \right) - \frac{y(t_k) - y(t_{k-1})}{\Delta t_k}\|_H^2 \\
&= \|\mathcal{P}^\ell \left(\frac{y(t_k) - y(t_{k-1})}{\Delta t_k} \right) - \mathcal{P}^\ell \dot{y}(t_k) + \mathcal{P}^\ell \dot{y}(t_k) - \dot{y}(t_k) + \dot{y}(t_k) - \frac{y(t_k) - y(t_{k-1})}{\Delta t_k}\|_H^2 \\
&\leq 2\|\mathcal{P}^\ell\|_{\mathcal{L}(H)}^2 \|\frac{y(t_k) - y(t_{k-1})}{\Delta t_k} - \dot{y}(t_k)\|_H^2 + 2\|\mathcal{P}^\ell \dot{y}(t_k) - \dot{y}(t_k)\|_H^2 + 2\|\dot{y}(t_k) - \frac{y(t_k) - y(t_{k-1})}{\Delta t_k}\|_H^2 \\
&\leq c_3 \|w_k\|_H^2 + 2\|\mathcal{P}^\ell \dot{y}(t_k) - \dot{y}(t_k)\|_H^2
\end{aligned}
$$

with $c_3 = 2 + 2\|\mathcal{P}^\ell\|_{\mathcal{L}(H)}^2$ and $w_k := \dot{y}(t_k) - \frac{y(t_k) - y(t_{k-1})}{\Delta t_k}$, which can be estimated as

$$
\sum_{k=1}^{j} \Delta t_k \|w_k\|_H^2 \leq \frac{(\Delta t)^2}{3} \|\ddot{y}\|_{L^2(0, t_j, H)}^2.
$$

For more details on this, we refer to [27] and [44].
Finally, we can summarize the estimation for the term involving ϑ_j by

$$
\|\vartheta_j\|_H^2 \leq c_4 \left(\sum_{k=1}^{n} \alpha_k (\|\mathcal{P}^\ell \dot{y}(t_k) - \dot{y}(t_k)\|_H^2 + \|\varrho_k\|_H^2 + \|\eta_k\|_H^2 + \|\zeta_k\|_H^2) + (\Delta t)^2 \|\ddot{y}\|_{L^2(0,T,H)}^2 \right)
$$

14

with $c_4 := e^{2c_2 T} \max\{\frac{c_3}{3}, 4, 2c_1\}$ and thus it is

$$
\begin{aligned}
\sum_{j=0}^{n} \alpha_j \|\vartheta_j\|_H^2 \;\leq\; & c_4 T (\sum_{j=0}^{n} \alpha_j \|\mathcal{P}^\ell \dot{y}(t_j) - \dot{y}(t_j)\|_H^2 + (\Delta t)^2 \|\ddot{y}\|_{L^2(0,T,H)}^2 + \sum_{i=\ell+1}^{d} \lambda_i \\
& + 2T(1 + \|\mathcal{P}^\ell\|_{\mathcal{L}(H)}^2)((\Delta t)^2 + \varepsilon_h^2))
\end{aligned}
\tag{34}
$$

Theorem 5.1. Let $\{y(t_j)\}_{j=0}^{n}$ denote the solution to problem (1) at the time grid $\{t_j\}_{j=0}^{n}$ and y_j^ℓ is the solution to (21). Let the nonlinear operator $\mathcal{N}$ be Lipschitz continuous and the maximal time step $\Delta t := \max_{j=0,\dots,n} \Delta t_j$ be sufficiently small. Furthermore, we assume $\ddot{y}(t)$ to be bounded on $[0, T]$. We choose the initial condition for (21) such that $\mathcal{P}^\ell y_0 = y_0^\ell$ is fulfilled. Then, there exists a constant $C > 0$ such that

$$
\sum_{j=0}^{n} \alpha_j \|y(t_j) - y_j^\ell\|_H^2 \leq C \left((\Delta t)^2 + \varepsilon_h^2 + \sum_{i=\ell+1}^{d} \lambda_i + \sum_{j=0}^{n} \alpha_j \|\mathcal{P}^\ell \dot{y}(t_j) - \dot{y}(t_j)\|_H^2 \right),
$$

where $\varepsilon_h := \max_{j=0,\dots,n} \|y^j - \bar{y}^j\|_H$ is the global spatial discretization error and $\bar{y}^j$ is the solution to (8) at time instance t_j.

Proof. Utilizing the decomposition (29) we infer

$$
\sum_{j=0}^{n} \alpha_j \|y(t_j) - y_j^\ell\|_H^2 \leq 2 \sum_{j=0}^{n} \alpha_j (\|\eta_j\|_H^2 + \|\varrho_j\|_H^2 + \|\zeta_j\|_H^2 + \|\vartheta_j\|_H^2).
$$

Together with (30), (31), (32) and (34) this leads to the claim. $\qquad\square$

Remark 5.2. If we choose $V = H^1(\Omega)$ and $H = L^2(\Omega)$ and utilize a static piecewise linear finite element discretization with h being the diameter of the triangles, then $\varepsilon_h = \mathcal{O}(h^2)$. In the case of adaptively refined spatial grids ε_h can be estimated by the prescribed error tolerance, if e.g. residual based a-posteriori error estimation is applied.

6 Numerical Realization and Examples

For all numerical examples, we choose $\Omega \subset \mathbb{R}^2$ as open and bounded domain and utilize conformal, piecewise linear finite elements for spatial discretization. For simplicity, we only utilize an h-adaptive concept, which means that at timestep $t_j, j = 0, \dots, n$ we refine or coarsen the spatial mesh according to the error indicator

$$
\eta_E = \sqrt{h_E} \| \left[\nabla y^j \right]_E \cdot \nu_E \|_E \quad \text{for all } E \in \mathcal{E},
\tag{35}
$$

where h_E is the length of the edge, $[\,.\,]_E$ denotes the jump of the function across the edge E, ν_E is the outward normal derivative on the edge E and $\mathcal{E}$ denotes the collection of all edges in the current triangulation. As refinement rule we use the marked edge bisection. We utilize structured, hierarchical and nested grids. The numerical test cases illustrate our approach to set up and solve a POD reduced order model utilizing snapshots with adaptive spatial discretization, which is explained in Section 3 and 4.

We compare our approach to the use of a uniform mesh, where the mesh size coincides with the fineness of the smallest triangle in the adaptive mesh. Furthermore, we compare our approach to the use of a finest mesh following [42]. For this, the practical numerical concept works as follows. In a full dimensional simulation, h-adaptive snapshots are generated at the time instances $\{t_j\}_{j=0}^{n}$. At the same time we carry along a reference grid with the simulation, which coincides with the computational grid at initial time and which is only refined in the same manner as the computational grid. In this way, the reference grid becomes the finest mesh (i.e. the overlay of all computational grids) at the end of the snapshot generation. Then, the snapshots are expressed with respect to the finite element basis functions corresponding to the finest mesh and the usual POD procedure is carried out.

All coding is done in C++ and we utilize FEniCS [29] for the solution of the differential equations and ALBERTA [40] for dealing with hierarchical meshes.

Example 6.1: Linear heat equation. We consider the Example 2.3 (2) of a heat equation with Dirichlet boundary condition and set $c \equiv 0$ such that the equation becomes linear. The spatial domain is chosen as $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$, the time interval is $[0,T] = [0,1.57]$. We construct an example in such a way that we know the analytical solution. It is given by

$$y(t,x) = r(t,x) \cdot (s_1(t,x) - s_2(t,x))$$

with

$$r(t,x) = \frac{50000 \cdot x_1 \cdot (1-x_1) \cdot (0.5+\cos(t)\cdot(x_1-0.5)-\sin(t)\cdot(x_2-0.5))^4 \cdot \frac{1}{t+1} \cdot (1-(0.5+\cos(t)\cdot(x_1-0.5)-\sin(t)\cdot(y-0.5)))^4)}{1+1000\cdot(\cos(t)\cdot(x_1-0.5)-\sin(t)\cdot(x_2-0.5))^2},$$

$$s_1(t,x) = \frac{10000 \cdot x_2 \cdot (1-x_2) \cdot (0.5+\sin(t)\cdot(x_1-0.5)+\cos(t)\cdot(x_2-0.5))^2 \cdot (0.5-\sin(t)\cdot(x_1-0.5)-\cos(t)\cdot(x_2-0.5))^2}{1+100*((0.5+\sin(t)\cdot(x_1-0.5)+\cos(t)\cdot(x_2-0.5))-0.25)^2},$$

$$s_2(t,x) = \frac{10000 \cdot x_2 \cdot (1-x_2) \cdot (0.5+\sin(t)\cdot(x_1-0.5)+\cos(t)\cdot(x_2-0.5))^2 \cdot (0.5-\sin(t)\cdot(x_1-0.5)-\cos(t)\cdot(x_2-0.5))^2}{(1+100*((0.5+\sin(t)\cdot(x_1-0.5)+\cos(t)\cdot(x_2-0.5))-0.75)^2}.$$

The forcing term f, the boundary condition g and the initial condition y_0 are chosen accordingly. For the temporal discretization we introduce the uniform time grid by

$$t_j = j\Delta t$$

for $j = 0, \ldots, 1570$ with $\Delta t = 0.001$. The analytical solution at three different time points is shown in Figure 2.
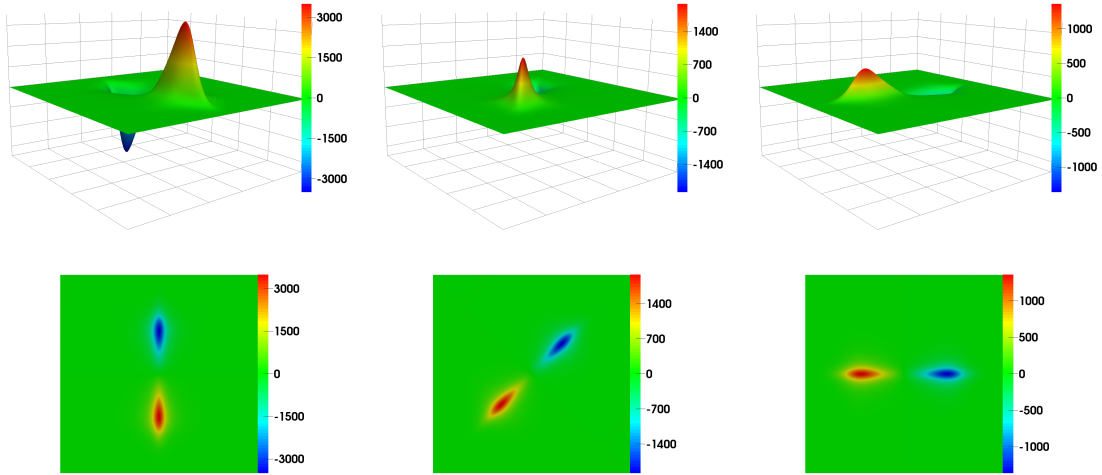

Figure 2: *Example 6.1: Surface plot (top) and view from above (bottom) of the analytical solution of (2) at $t = t_0$ (left), $t = T/2$ (middle) and $t = T$ (right)*

Due to the steep gradients in the neighbourhood of the minimum and maximum, respectively, the use of an adaptive finite element discretization is justified. The resulting computational meshes as well as the corresponding finest mesh (reference mesh at the end of the simulation) are shown in Figure 3. The number of node points of the adaptive meshes varies between 3637 and 7071 points. The finest mesh has 18628 node points. In contrary, a uniform mesh with the same discretization fineness as the finest triangle in the adaptive grids ($h_{\min} = 0.0047$) would have 93025 node points. This clearly reveals the benefit of using adaptive meshes for snapshot generation. Particularly, the comparison of the computational times emphasizes the benefit of adaptive snapshot sampling: the snapshot generation on the adaptive mesh takes 944 seconds, whereas utilizing the uniform mesh it takes 8808 seconds. Therefore, we gain a speedup factor of 9 (see Table 2).
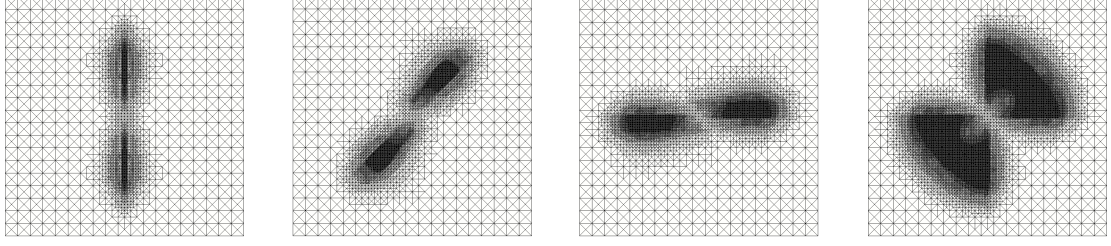
Figure 3: *Example 6.1: Adaptive finite element meshes at (i) $t = t_0$, (ii) $t = T/2$, (iii) $t = T$ and (iv) finest mesh*

In Figure 4, the resulting normalized eigenspectrum of the correlation matrix for uniform spatial discretization ("uniform FE mesh"), the normalized eigenspectrum of the matrix $\mathcal{K}$ (13) ("inf-POD") as well as the normalized eigenspectrum of the correlation matrix utilizing snapshots interpolated onto the finest mesh ("finest FE mesh") is shown. We observe that the eigenvalues for both adaptive approaches coincide. Moreover, we recognize that about the first 28 eigenvalues computed corresponding to the adaptive simulation coincide with the simulation on a uniform mesh. From index 29 on, the methods deliver different results: for the uniform discretizations, the normalized eigenvalues fall below machine precision at around index 100 and stagnate. In contrary, the normalized eigenvalues for both adaptive approaches flatten in the order around 10^{-10}. Concerning dynamical systems, the magnitude of the eigenvalue corresponds to the characteristic properties of the underlying dynamical system: the larger the eigenvalue, the more information is contained in the corresponding eigenfunction. Since all adaptive meshes are contained in the uniform mesh, the difference in the amplitude of the eigenvalues is due to the interpolation errors during refinement and coarsening. This is the price we have to pay in order to get a fast snapshot generation utilizing adaptive finite elements. Moreover, the investigation of the decay of the eigenvalues constitutes an analyzing tool for adaptivity in the following sense. Using an adaptive mesh technique means that some parts of the domain are resolved coarsely according to the utilized error estimation, i.e. information gets lost. In the sense of a singular value analysis, this implies that adaptivity neglects the noise which is indicated by the singular values on the uniform spatial mesh at those places which are not resolved with the adaptive grid. The overtones which get lost in the adaptive computations lie in the same space which is not considered by POD when using the adaptive finite element snapshots. This allows us to characterize the space which is not resolved by adaptivity. From this point of view, adaptivity can be interpreted as a smoother.
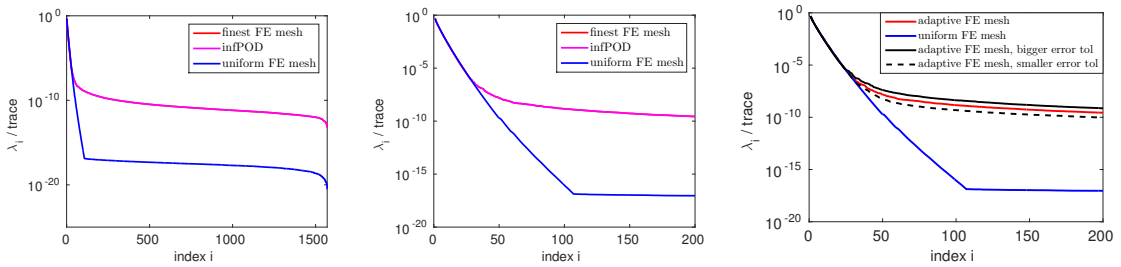

Figure 4: *Example 6.1: Comparison of the normalized eigenvalues utilizing an adaptive and a uniform spatial mesh, respectively. Left: all eigenvalues, middle: first 200 largest eigenvalues, right: first 200 largest eigenvalues with different error tolerances for the adaptivity (1.5 times bigger and smaller error tolerances, respectively)*

Since the first few POD basis functions are the most important ones regarding the captured information, we visualize ψ_1, ψ_2 and ψ_5 in Figure 5, which are computed corresponding to using an adaptive grid. The POD basis functions corresponding to the uniform spatial discretization have a similar appearance. Note that the POD bases are unique up to the sign. We can recognize the initial condition in the first POD basis function. Then, the index of the POD basis corresponds to the number of maxima and minima of the POD basis: ψ_2 has two minima and two maxima

etc. This behaviour is similar to the increasing oscillations in higher frequencies in trigonometric approximations. The increasing number of oscillations is necessary in order to approximate the transport of the steep gradients of the solution with increasing accuracy.
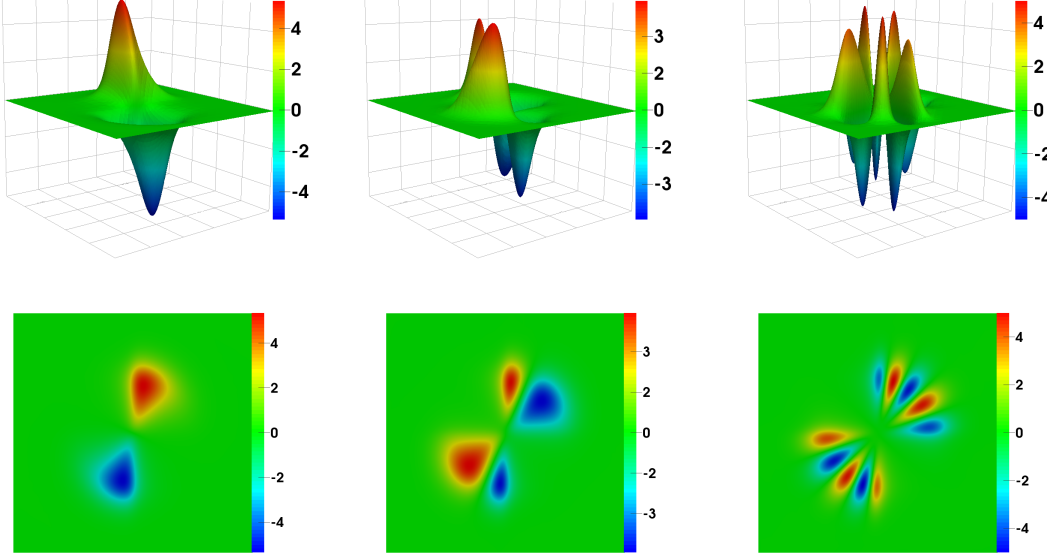

Figure 5: *Example 6.1: Surface plot (top) and view from above (bottom) of the POD basis functions ψ_1 (left), ψ_2 (middle) and ψ_5 (right)*

The POD solutions for $\ell = 10$ and $\ell = 50$ POD basis functions utilizing spatial adaptive snapshots which are interpolated onto the finest mesh are shown in Figure 6. The visual comparison makes clear which influence the increase of utilized POD basis functions has on the approximation quality. The more POD basis functions we use (until stagnation of the corresponding eigenvalues), the less oscillations appear in the POD solution and the better is the approximation.
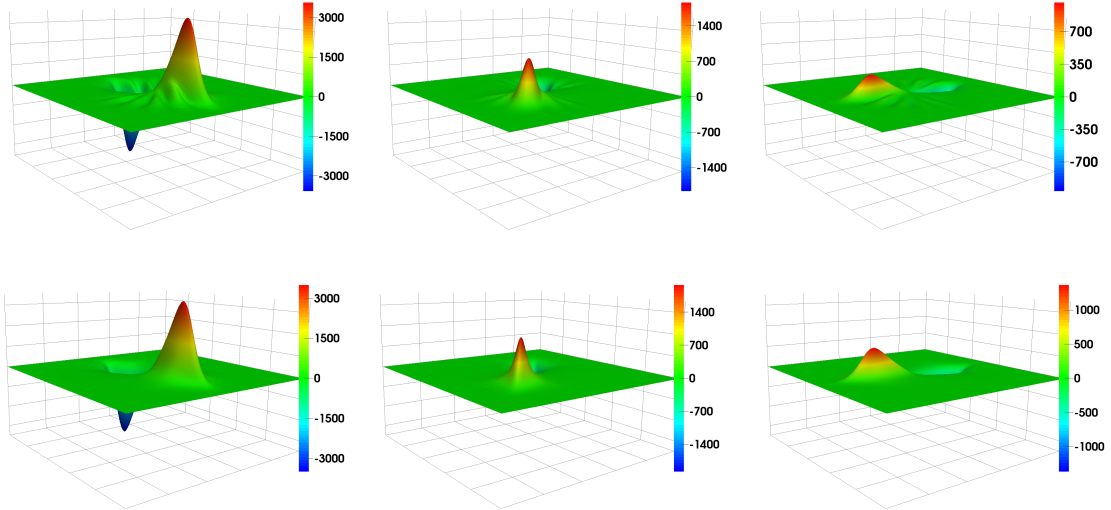

Figure 6: *Example 6.1: Surface plot of the POD solution utilizing $\ell = 10$ (top) and $\ell = 50$ (bottom) POD basis functions at $t = t_0$ (left), $t = T/2$ (middle) and $t = T$ (right)*

Table 1 compares the approximation quality of the POD solution utilizing adaptively generated

18

snapshots which are interpolated onto the finest mesh with snapshots of uniform spatial discretization depending on different POD basis lengths. Our approach from Sections 3 and 4 delivers very similar results as the use of adaptive finite element snapshots which are interpolated onto the finest mesh. For example, for $\ell = 20$ POD bases, we get the following errors: relative $L^2(Q)$-error between the POD solution and the finite element solution: $\varepsilon_{\text{FE}} = 3.07 \cdot 10^{-2}$, relative $L^2(Q)$-error between the POD solution and the true solution: $\varepsilon_{\text{true}} = 2.16 \cdot 10^{-2}$.

ℓ	$\varepsilon_{\text{FE}}^{\text{ad}}$	$\varepsilon_{\text{FE}}^{\text{uni}}$	$\varepsilon_{\text{true}}^{\text{ad}}$	$\varepsilon_{\text{true}}^{\text{uni}}$
1	$1.30 \cdot 10^0$	$1.30 \cdot 10^0$	$1.28 \cdot 10^0$	$1.30 \cdot 10^0$
3	$7.49 \cdot 10^{-1}$	$7.58 \cdot 10^{-1}$	$7.46 \cdot 10^{-1}$	$7.60 \cdot 10^{-1}$
5	$4.39 \cdot 10^{-1}$	$4.45 \cdot 10^{-1}$	$4.39 \cdot 10^{-1}$	$4.46 \cdot 10^{-1}$
10	$1.37 \cdot 10^{-1}$	$1.37 \cdot 10^{-1}$	$1.36 \cdot 10^{-1}$	$1.38 \cdot 10^{-1}$
20	$3.08 \cdot 10^{-2}$	$1.56 \cdot 10^{-2}$	$2.17 \cdot 10^{-2}$	$1.60 \cdot 10^{-2}$
30	$2.59 \cdot 10^{-2}$	$2.04 \cdot 10^{-3}$	$1.49 \cdot 10^{-2}$	$3.00 \cdot 10^{-3}$
50	$2.63 \cdot 10^{-2}$	$5.67 \cdot 10^{-5}$	$1.41 \cdot 10^{-2}$	$2.07 \cdot 10^{-3}$
100	$2.61 \cdot 10^{-2}$	$6.48 \cdot 10^{-8}$	$1.40 \cdot 10^{-2}$	$2.06 \cdot 10^{-3}$
150	$2.61 \cdot 10^{-2}$	$8.13 \cdot 10^{-7}$	$1.39 \cdot 10^{-2}$	$2.07 \cdot 10^{-3}$

Table 1: *Example 6.1: Relative $L^2(Q)$-error between the POD solution and the finite element solution (columns 2-3) and the true solution (columns 4-5), respectively, utilizing adaptive finite element snapshots which are interpolated onto the finest mesh and utilizing a uniform mesh*

We note that the error between POD solution and finite element solution utilizing a uniform mesh decays down to the order 10^{-8} ($\ell = 100$) and then stagnates. This behaviour is clear, since the more POD basis we include (up to stagnation of the corresponding eigenvalues), the better is the POD solution an approximation for the finite element solution. In contrary, the error between the POD solution and the true solution starts to stagnate from $\ell = 29$. This is due to the fact that at this point also the spatial discretization error has a notable influence and it is in accordance to the decay of the eigenvalues shown in Figure 4.

Finally, of particular interest is the computational efficiency of the POD reduced order modeling utilizing adaptive finite element discretizations. For this reason, the computational times for the full and the low order simulation utilizing uniform finite element discretizations and adaptive finite element snapshots, which are interpolated onto the finest mesh, respectively, are listed in Table 2.

	adaptive FE mesh	uniform FE mesh	speedup factor
FE simulation	944 sec	8808 sec	9.3
POD offline computations	264 sec	1300 sec	4.9
POD simulation	204 sec	1107 sec	5.4
speedup factor	4.6	7.9	–

Table 2: *Example 6.1: CPU times for FE and POD simulation utilizing uniform finite element meshes and adaptive finite element snapshots which are interpolated onto the finest mesh, respectively, and utilizing $\ell = 50$ POD modes*

Once the POD basis is computed in the offline phase, the POD simulation corresponding to adaptive snapshots is 5 times faster than the FE simulation utilizing adaptive finite element meshes. This speedup factor even gains greater importance, if we think of optimal control problems, where the repeated solving of several partial differential equations is necessary. We note that in the approach explained in Section 3 and 4, the computation of the matrix $\mathcal{K}$ (13) is expensive. Since $\mathcal{K}$ is symmetric, it suffices to calculate the entries on and above the diagonal, which are $\sum_{k=1}^{n+1} k = \frac{1}{2}((n+1)^2 + n + 1)$ entries. For each entry the calculation time is around 0.03 seconds, which leads to a computation time of around 36997 seconds for the matrix $\mathcal{K}$. Same effort is needed to build $\mathcal{Y}^\star \mathcal{A} \mathcal{Y}$. In this case, the offline phase is therefore around 88271 seconds. For this reason, the approach to interpolate the adaptive generated snapshots onto the finest mesh is computationally more favorable. But since the computation of $\mathcal{K}$ and $\mathcal{Y}^\star \mathcal{A} \mathcal{Y}$ can be parallelized,

the offline computation time can be reduced provided that the appropriate hardware is available.

Example 6.2: Cahn-Hilliard system. We consider Example 2.4, (3) of the Cahn-Hilliard equations given in the coupled formulation for the phase field c and the chemical potential w. The data is chosen as follows: we consider the rectangular domain $\Omega = (0, 1.5) \times (0, 0.75)$, the end time $T = 0.025$, constant mobility $m \equiv 0.00002$ and a constant surface tension $\sigma \equiv 24.5$. The interface parameter ε is set to $\varepsilon = 0.02$, which leads to an interface thickness of about $\pi \cdot \varepsilon \approx 0.0628$. We utilize the relaxed double-obstacle free energy W_s^{rel}, (5) with $s = 10^4$. As initial condition, we choose a circle with radius $r = 0.25$ and center $(0.375, 0.375)$. The initial condition is transported horizontally with constant velocity $v = (30, 0)^T$. As spaces for the Gelfand triple we consider $V = H_0^1(\Omega)$ and $H = L^2(\Omega)$. Let us define the uniform time discretization

$$t_j = j\Delta t$$

for $j = 0, \ldots, 1000$ with $\Delta t = 2.5 \cdot 10^{-5}$. We utilize a semi-implicit Euler scheme for temporal discretization. Let $c^{j-1} \in V$ and $c^j \in V$ denote the time discrete solution at t_{j-1} and t_j. The time discrete Cahn-Hilliard system reads as follows. For given c^{j-1}, find c^j with associated w^j solving

$$\begin{cases} \langle \dfrac{c^j - c^{j-1}}{\Delta t}, v_1 \rangle_H + \langle v \cdot \nabla c^{j-1}, v_1 \rangle_H + m \langle \nabla w^j, \nabla v_1 \rangle_H &= 0 \quad \forall v_1 \in V, \\ -\langle w^j, v_2 \rangle_H + \sigma \varepsilon \langle \nabla c^j, \nabla v_2 \rangle_H + \dfrac{\sigma}{\varepsilon} \langle W'_+(c^j) + W'_-(c^{j-1}), v_2 \rangle_H &= 0 \quad \forall v_2 \in V, \end{cases} \tag{36}$$

and $c^0 = c_0$. Note that the free energy function W is split into a convex part W_+ and a concave part W_-, such that $W = W_+ + W_-$ and W'_+ is treated implicitly with respect to time and W'_- is treated explicitly with respect to time. This leads to an unconditionally energy stable time marching scheme, compare [14]. The system (36) is discretized in space utiziling piecewise linear finite elements and solved using a semi-smooth Newton method.

Figures 7 shows the phase field (left) and the chemical potential (right) for the finite element simulation utilizing adaptive meshes. The initial condition c_0 is transported horizontally with constant velocity.
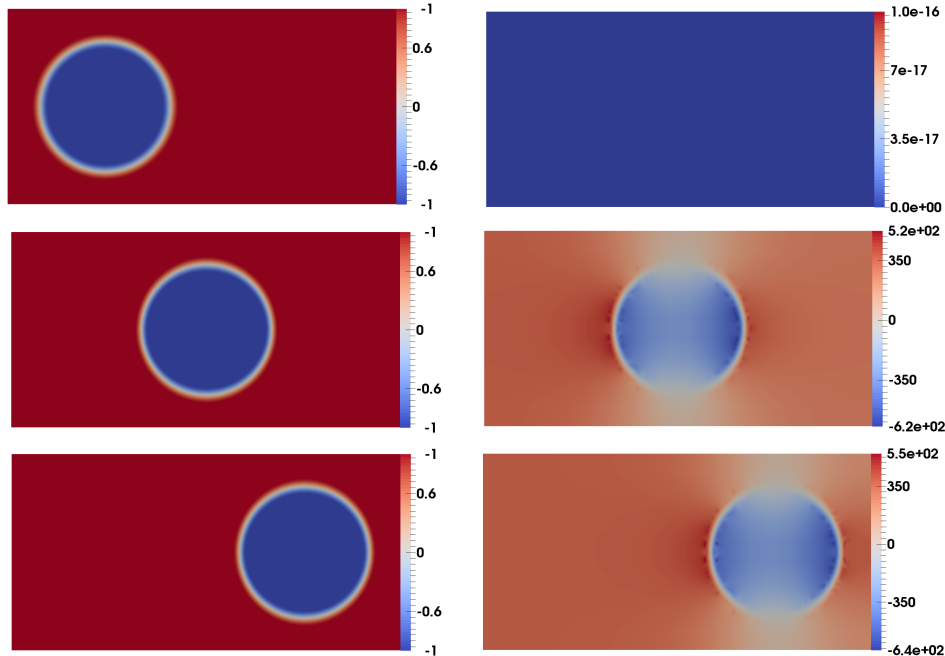

Figure 7: *Example 6.2: Phase field c (left) and chemical potential w (right) computed on adaptive finite element meshes at $t = t_0$ (top), $t = T/2$ (middle) and $t = T$ (bottom)*

The adaptive finite element meshes as well as the finest mesh which is generated during the adaptive finite element simulation are shown in Figure 8. The number of degrees of freedom in the adaptive

meshes varies between 6113 and 8795. The finest mesh (overlay of all adaptive meshes) has 54108 degrees of freedom, whereas a uniform mesh with discretization fineness as small as the smallest triangle in the adaptive meshes has 88450 degrees of freedom.
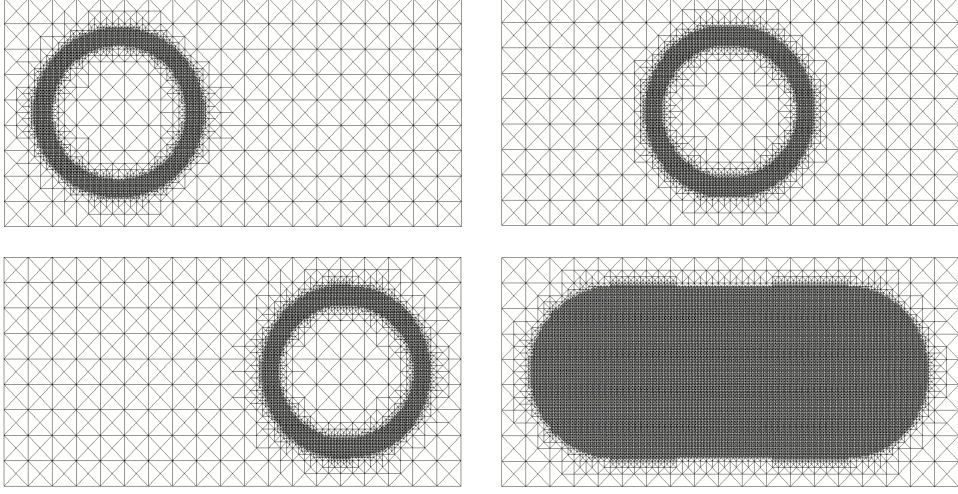

Figure 8: *Example 6.2: Adaptive finite element meshes and finest mesh*

In this example, we only compare the solution to the POD-ROM utilizing two kinds of snapshot discretizations: on the one hand we use adaptive finite elements and express these with respect to the finite element basis functions corresponing to the finest mesh. On the other hand we compute the solution to the POD-ROM with snapshots computed on a uniform finite element discretization, where the fineness is chosen to be of the same size as the smallest triangle in the adaptive meshes.

In Figure 9, a comparison is visualized concerning the normalized eigenspectrum for the phase field c and the chemical potential w utilizing uniform and adaptive finite element discretization. We note for the phase field c that about the first 180 eigenvalues computed corresponding to the adaptive simulation coincide with the eigenvalues of the simulation on the finest mesh. Then, the eigenvalues corresponding to the uniform simulation decay faster. Similar observations apply for the chemical potential w.
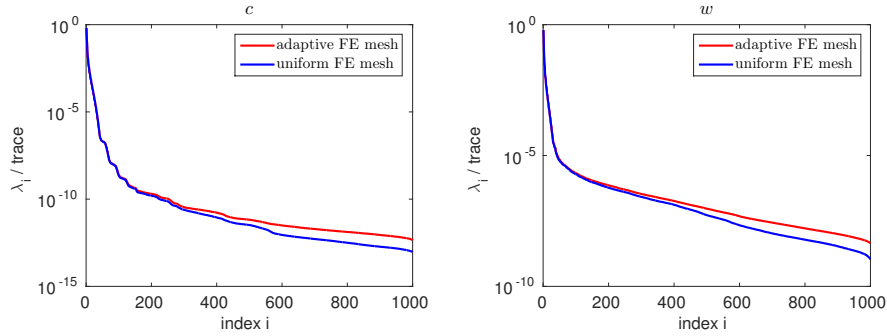

Figure 9: *Example 6.2: Comparison of the normalized eigenvalues for the phase field c (left) and the chemical potential w (right) utilizing an adaptive and a uniform spatial mesh, respectively*

The information content of a POD basis of rank ℓ relatively to the amount of the information content of all snapshots is given by the energy E, i.e. the ratio of modeled and total energy. It is defined by

$$E(\ell) := \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{d} \lambda_i}.$$

We will choose the numbers ℓ_c and ℓ_w, such that

$$\ell = \mathrm{argmin}\{E(\ell) : E(\ell) > 1 - p\}$$

21

for a given value p representing the loss of information. Table 3 and Figure 10 summarize how to choose ℓ_c and ℓ_w in order to achieve a desired accuracy.

p	ℓ_c^{ad}	ℓ_w^{ad}	ℓ_c^{uni}	ℓ_w^{uni}
10^{-01}	3	4	3	4
10^{-02}	10	13	10	13
10^{-03}	19	26	19	25
10^{-04}	29	211	28	160
10^{-05}	37	644	37	419
10^{-06}	65	-	51	562
10^{-10}	-	-	57	601

Table 3: *Example 6.2: Number of needed POD bases in order to achieve a loss of information below the tolerance p utilizing adaptive finite element meshes (columns 2-3) and uniform finite element discretization (columns 4-5).*
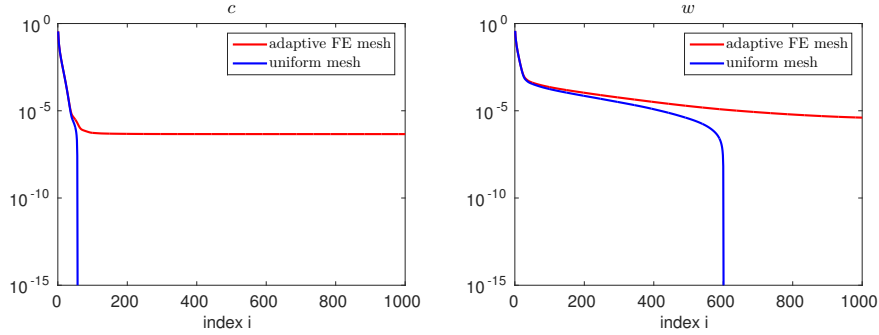

Figure 10: *Example 6.2: Number of needed POD bases in order to achieve a loss of information below a certain tolerance*

In the following, we run the numerical simulations for different combinations of numbers for ℓ_c and ℓ_w of Table 3.

Figure 11 visualizes the first, second and fifth POD modes for the phase field c and the chemical potential w. Analogue to Example 6.1, we observe a periodicity in the POD bases corresponding to its index number.

The approximation quality of the POD solution utilizing adaptive meshes is compared to the use of a uniform mesh in Table 4.

ℓ^c	ℓ^w	$c : \varepsilon_{\text{FE}}^{\text{ad}}$	$w : \varepsilon_{\text{FE}}^{\text{ad}}$	$c : \varepsilon_{\text{FE}}^{\text{uni}}$	$w : \varepsilon_{\text{FE}}^{\text{uni}}$
3	4	$8.44 \cdot 10^{-3}$	$3.00 \cdot 10^{0}$	$8.44 \cdot 10^{-3}$	$3.75 \cdot 10^{0}$
10	13	$3.30 \cdot 10^{-3}$	$3.77 \cdot 10^{-1}$	$3.30 \cdot 10^{-3}$	$4.32 \cdot 10^{-1}$
19	26	$1.57 \cdot 10^{-3}$	$2.12 \cdot 10^{-1}$	$1.57 \cdot 10^{-3}$	$2.39 \cdot 10^{-1}$
29	26	$7.34 \cdot 10^{-4}$	$1.09 \cdot 10^{-1}$	$7.32 \cdot 10^{-4}$	$1.16 \cdot 10^{-1}$
37	26	$3.57 \cdot 10^{-4}$	$4.82 \cdot 10^{-2}$	$3.55 \cdot 10^{-4}$	$5.04 \cdot 10^{-2}$
50	50	$1.88 \cdot 10^{-4}$	$2.17 \cdot 10^{-2}$	$1.86 \cdot 10^{-4}$	$2.33 \cdot 10^{-2}$
65	26	$9.74 \cdot 10^{-5}$	$1.11 \cdot 10^{-2}$	$9.56 \cdot 10^{-5}$	$1.15 \cdot 10^{-2}$
100	100	$3.37 \cdot 10^{-5}$	$3.56 \cdot 10^{-3}$	$3.22 \cdot 10^{-5}$	$3.42 \cdot 10^{-3}$

Table 4: *Example 6.2: Relative $L^2(Q)-$error between the POD solution and the finite element solution utilizing adaptive meshes (columns 3-4) and utilizing a uniform mesh (columns 5-6), respectively.*

Finally, the treatment of the nonlinearity shall be discussed. Utilizing the convex-concave splitting for W, we get for the Moreau-Yosida relaxed double obstacle free energy $W_-^{\text{rel}}(c) = \frac{1}{2}(1 -$
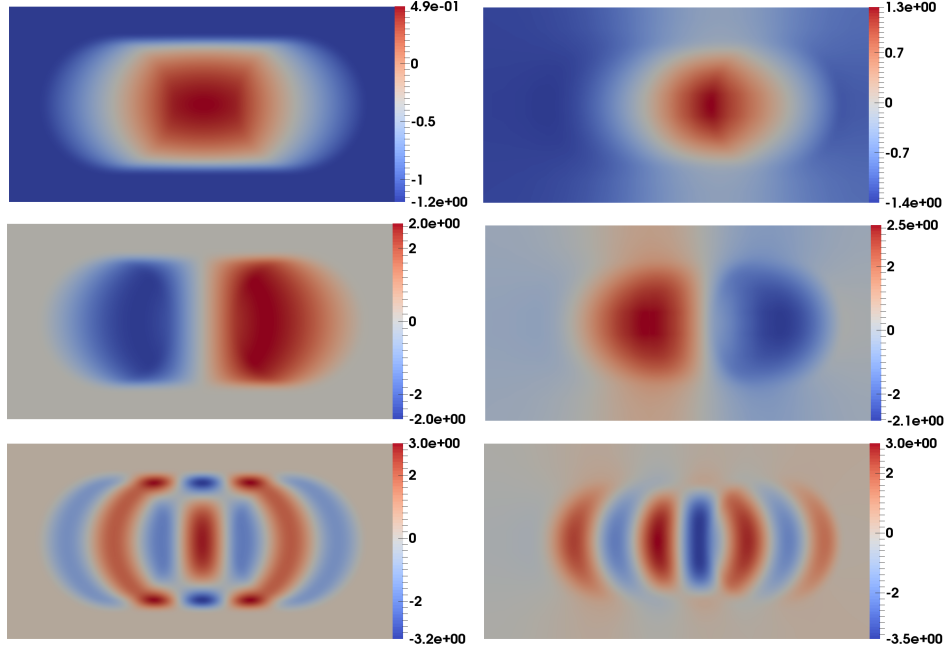
Figure 11: *Example 6.2: First, second and fifth POD bases for c (left) and w (right).*

c^2) for the concave part and $W_+^{\mathrm{rel}}(c) = \frac{s}{2}(\max(c-1,0)^2 + \min(c+1,0)^2)$ for the convex part. This means that the first derivative of the concave part is linear with respect to the phase field variable c. The challenging part is the convex term which first derivative is nonsmooth. For a comparison, we consider the smooth polynomial free energy which concave part is $W_-^p(c) = \frac{1}{4}(1 - 2c^2)$ and convex part is $W_+^p(c) = \frac{1}{4}c^4$.

Figure 12 shows the decay of the normalized eigenspectrum for the phase field c (left) and the first derivative of the convex part $W_+'(c)$ (right) for the polynomial and the relaxed double obstacle free energy. Obviously, in the nonsmooth case more POD modes are needed for a good approximation than in the smooth case. This behaviour is similiar to the decay of the Fourier coefficients in the context of trigonometric approximation, where the decay of the Fourier coefficients depends on the smoothness of the approximated object.
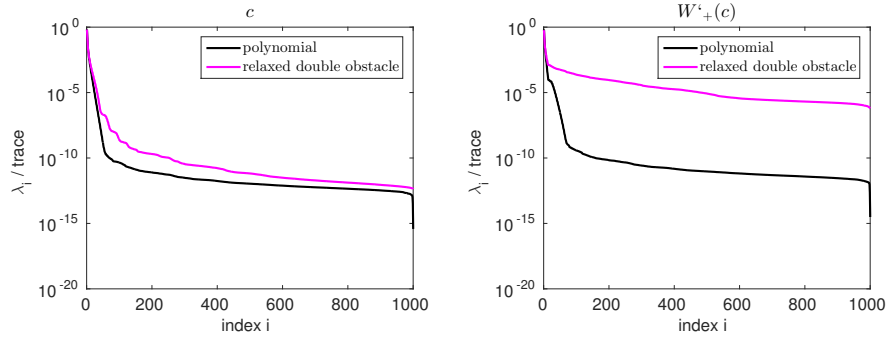

Figure 12: *Example 6.2: Comparison of the normalized eigenvalues for c (left) and the first derivative of the convex part W_+' of the free energy (right) utilizing polynomial and relaxed double obstacle energy, respectively*

Table 5 summarizes computational times for different finite element runs as well as reduced order simulations utilizing the polynomial and the relaxed double obstacle free energy, respectively. In addition, the approximation quality is compared. The computational times are rounded averages from various test runs. It turns out that the finite element simulation (row 1) using the smooth potential is around two times faster than using the nonsmooth potential. This is due to the fact

that in the smooth case, 2-3 Newton steps are needed for convergence in each time step, whereas in the nonsmooth case 6-8 iterations are needed in the semismooth Newton method.

Utilizing the smooth polynomial free energy, the reduced order simulation is 8-9 times faster than the finite element simulation, whereas utilizing the relaxed double obstacle free energy only delivers a very small speedup. The inclusion of DEIM (we use $\ell_{\text{deim}} = \ell_c$) in the reduced order model leads to immense speedup factors for both free energy functions (row 8). This is due to the fact that the evaluation of the nonlinearity in the reduced order model is still dependent on the full spatial dimension and hyper reduction methods are necessary for useful speedup factors. Note that the speedup factors are of particular interest in the context of optimal control problems. At the same time, the relative $L^2(Q)$-error between the finite element solution and the ROM-DEIM solution is close to the quality of the reduced order model solution (row 10-11).

However, in the case of the nonsmooth free energy function utilizing $\ell_c = 19$ POD modes for the phase field and $\ell_w = 26$ POD modes for the chemical potential, the inclusion of DEIM has the effect that the semismooth Newton method does not converge. For this reason, we treat the nonlinearity by applying the technique explained in Section 4.2., i.e. we project the finite element snapshots for $W'_+(c)$ (which are interpolated onto the finest mesh) onto the POD space. Since this leads to linear systems, the computational times are very small (row 6). The error between the finite element solution and the reduced order solution utilizing projection of the nonlinearity lies in the area $10^{-02}/10^{-03}$. Depending on the motivation, this approximation quality might be sufficient. Nevertheless, we note that that for large numbers of POD modes, utilizing the projection of the nonlinearity onto the POD space leads to a large increase of the error.

	W^p		W_s^{rel}	
	$\ell_c = 3, \ell_w = 4$	$\ell_c = 19, \ell_w = 26$	$\ell_c = 3, \ell_w = 4$	$\ell_c = 19, \ell_w = 26$
FE	1686 sec	1603 sec	3071 sec	3187 sec
POD offline	355 sec	355 sec	350 sec	349 sec
DEIM offline	8 sec	8 sec	9 sec	10 sec
ROM	183 sec	191 sec	2616 sec	3388 sec
ROM-DEIM	0.05 sec	0.1 sec	0.04 sec	no conv.
ROM-proj	0.008 sec	0.03 sec	0.01 sec	0.03 sec
speedup FE-ROM	9	8	1	none
speedup FE-ROM-DEIM	33720	16030	76775	–
speedup FE-ROM-proj	210750	53433	307100	106233
rel $L^2(Q)$ error ROM	$5.46 \cdot 10^{-03}$	$3.23 \cdot 10^{-04}$	$8.44 \cdot 10^{-03}$	$1.57 \cdot 10^{-03}$
rel $L^2(Q)$ error ROM-DEIM	$1.46 \cdot 10^{-02}$	$3.83 \cdot 10^{-04}$	$8.84 \cdot 10^{-03}$	–
rel $L^2(Q)$ error ROM-proj	$4.70 \cdot 10^{-02}$	$4.18 \cdot 10^{-02}$	$8.72 \cdot 10^{-03}$	$9.80 \cdot 10^{-03}$

Table 5: *Example 6.2: Computational times, speedup factors and approximation quality for different POD basis lengths and using different free energy potentials*

7 Conclusion

In this work, a POD reduced order model is proposed which can be set up and solved for snaphots which are discretized utilizing arbitrary finite elements. The method is applicable for h-, p- and r-adaptive finite elements. The approach is motivated from an infinite dimensional perspective. Using the method of snapshots we are able to set up the correlation matrix by explicitly evaluating the inner products of snapshots which live in different finite element spaces. For this reason, our approach avoids the interpolation onto a common reference mesh or the expression of the snapshots in terms of a common finite element basis. From a computational point of view, sufficient hardware should be available in order to compute the correlation matrix in parallel and make the offline computational time competitive. For semilinear evolution problems, the nonlinearity is treated by linearization. This is of interest in view of optimal control problems, in which a linearized state equation has to be solved in each SQP iteration level. In future work, we intend to study the combination of adaptive finite elements and POD reduced order modeling in the context of multi-phase flow and optimal control.

References

[1] **H. Abels.** Diffuse Interface Models for Two-Phase flows of Viscous Incompressible Fluids. *Max-Planck Institut für Mathematik in den Naturwissenschaften, Leipzig, Lecture Note, 36, 2007.*

[2] **H. Abels, H. Garcke, G. Grün.** Thermodynamically consistent, frame indifferent diffuse interface models for incompressible two-phase flows with different densities. *Mathematical Models and Methods in Applied Sciences, 22(3):40, 2012.*

[3] **K. Afanasiev, M. Hinze.** Adaptive control of a wake flow using proper orthogonal decomposition. *Shape optimization and optimal design (Cambridge, 1999), Lecture Notes in Pure and Appl. Math., 216:317-332. Dekker, New York, 2001.*

[4] **A. Alla, J. N. Kutz.** Nonlinear model order reduction via Dynamic Mode Decomposition. *to appear SIAM Journal on Scientific Computing, 2016.*

[5] **M. Ali, K. Steih, K. Urban.** Reduced basis methods based upon adaptive snapshot computations. *Advances in Computational Mathematics, 43(2):257-294, 2017.*

[6] **P. Astrid, S. Weiland, K. Willcox, T. Backx.** Missing Point Estimation in Models Described by Proper Orthogonal Decompostion. *IEE Trans. Automat. Control, 53(10), 2237-2251, 2008.*

[7] **J. F. Blowey, C. M. Elliott.** The Cahn-Hilliard gradient theory for phase separation with non-smooth free energy. Part I: Mathematical analysis. *European Journal of Applied Mathematics, 2:233-280, 1991.*

[8] **M. Barrault, Y. Maday, N. C. Nguyen, A. T. Patera.** An "empirical interpolation" method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris, Ser. I 339, 667-672, 2004.*

[9] **P. Benner, R. Rannacher.** Introduction to Part III Adaptivity and Model Reduction. *Trends in PDE Constrained Optimization, International Series of Numerical Mathematics, 165:249-250, 2014.*

[10] **Y. Chen.** Model order reduction for nonlinear systems. *Master's thesis, Massachusetts Institute of Technology, 1999.*

[11] **J. W. Cahn, J. E. Hilliard.** Free energy of a non-uniform system. I. Interfacial free energy. *J. Chem. Phys. 28:258-267, 1958.*

[12] **S. Chaturantabut, D. C. Sorensen.** Nonlinear model reduction via discrete empirical interpolation. *Siam J. Sci. Comput. 32(5), 2737-2764, 2010.*

[13] **Z. Drmac, S. Gugercin.** A New Selection Operator for the Discrete Empirical Interpolation Method – improved a priori error bound and extension. *SIAM J. Sci. Comput. 38(2):A631-A648, 2016.*

[14] **D. J. Eyre.** unconditionally Gradient Stable Time Marching the Cahn-Hilliard Equation. *MRS Proceedings, 529, 1998.*

[15] **F. Fang, C. C. Pain, I. M. Navon, M. D. Piggott, G. J. Gorman, P. A. Allison, A. J. Goddard.** Reduced-order modelling of an adaptive mesh ocean model. *Int. J. Numer. Meth. Fluids 59: 827-851, 2009.*

[16] **L. Feng, X. Zeng, C. Chiang, D. Zhou, Q. Fang.** Direct Nonlinear Order Reduction with Variational Analysis. *Proc. Design, Automation and Test in Europe, 1530-1591, 2004.*

[17] **M. Gubisch, S. Volkwein.** Proper Orthogonal Decomposition for Linear-Quadratic Optimal Control. *To appear in P. Benner, A. Cohen, M. Ohlberger, K. Willcox (eds.), Model Reduction and Approximation: Theory and Algorithms, SIAM, Philadelphia, PA, 2017.*

[18] **P. C. Hohenberg, B. I. Halperin.** Theory of dynamic critical phenomena. *Reviews of Modern Physics, 49(3):435-479, 1977.*

[19] **M. Hintermüller, M. Hinze, M. H. Tber.** An Adaptive Finite Element Moreau-Yosida-Based Solver for a Non-Smooth Cahn-Hilliard Problem. *Optim. Meth. Software 26:777-811, 2011.*

[20] **M. Hinze, K. Kunisch.** Three control methods for time-dependent fluid flow. *Flow, Turbulence and Combustion, 65:273-298, 2000.*

[21] **M. Hinze, J. Krenciszek, R. Pinnau.** Proper Orthogonal Decomposition for Free Boundary Value Problems. *Hamburger Beiträge zur Angewandten Mathematik, 2014.*

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

[23] **M. Hinze, S. Volkwein.** Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition. *Technical Report IMA02-05, KFU Graz, 2005.*

[24] **M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich.** Optimization with PDE Constraints. *Springer-Verlag, Berlin, 2009.*

[25] **K. Kunisch, S. Volkwein.** Control of Burgers' equation by a reduced order approach using proper orthogonal decomposition. *Journal on Optimization Theory and Apllications, 102:345-371, 1999.*

[26] **K. Kunisch, S. Volkwein.** Galerkin Proper Orthogonal Decomposition Methods for Parabolic Problems. *Numer. Math. 90:117-148, 2001.*

[27] **K. Kunisch, S. Volkwein.** Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. *SIAM J. Numer. Anal., 40(2):492-515, 2002.*

[28] **O. Lass.** Reduced order modeling and parameter identification for coupled nonlinear PDE systems. *PhD thesis, Universität Konstanz, 2014.*

[29] **A. Logg, K.-A. Mardal, G. Wells, eds.** Automated Solution of Differential Equations by the Finite Element Method. *The FEniCS Book, vol. 84 of Lecture Notes in Computational Science and Engineering, Springer, 2012.*

[30] **T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza.** Model order reduction in fluid dynamics: challenges and perspectives. *Reduced order methods for modeling and computational reduction, A. Quarteroni and G. Rozza, Eds., Springer MS&A Series, 9:235-274, 2014.*

[31] **J. L. Lumley.** The Structure of Inhomogeneous Turbulent Flows. *In A. M. Yaglom and V. I. Tatarski, editors, Atmospheric turbulence and radio propagation, pages 166-178. Nauka, Moscow, 1967.*

[32] **N. C. Nguyen, A. T. Patera, J. Peraire.** A "best point" interpolation method for efficient approximation of parametrized functions. *Int. J. Numer. Meth. Eng., 73(4) 521-543, 2008.*

[33] **A. Pazy.** Semigroups of Linear Operators and Applications to Partial Differential Equations. *Springer, New York, 1983.*

[34] **J. R. Phillips.** Automated extraction of nonlinear circuit macromodels. *in Proc. Custom Integrated Circuit Conf., 451-454, 2000.*

[35] **J. R. Phillips.** Projection-Based Approaches for Model Reduction of Weakly Nonlinear, Time-Varying Systems. *IEE Transactions on computer-aided design of integrated circuits and systems, 22(2), 2003.*

[36] **R. Pinnau.** Model Reduction via Proper Orthogonal Decomposition. *In W.H.A. Schilder, H. van der Vorst: Model Order Reduction: Theory, Research Aspects and Applications, Springer, 96-109, 2008.*

[37] **J. P. Raymond, H. Zidani.** Hamiltonian Pontryagin's Principles for control problems governed by semilinear parabolic equations. *Applied Math. and Optimization, 39:143-177, 1999.*

[38] **M. Reed, B. Simon.** Methods of modern mathematical physics. I: Functional analysis. *Academic Press 1980.*

[39] **M. Rewieński, J. White.** A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices. *IEE Transactions on computer-aided design of integrated circuits and systems, 22(2), 2003.*

[40] **A. Schmidt, K. G. Siebert.** Design of adaptive finite element software: The finite element toolbox ALBERTA. *Lecture Notes in Computational Science and Engineering, Springer, 42, 2005.*

[41] **L. Sirovich.** Turbulence and the dynamics of coherent structures. Parts I-II. *Quarterly of Applied Mathematics, XVL, 561-590, 1987.*

[42] **S. Ullmann, M. Rotkvic, J. Lang.** POD-Galerkin reduced-order modeling with adaptive finite element snapshots. *Journal of Computational Physics, 325:244-258, 2016.*

[43] **S. Volkwein.** Optimal Control of a Phase-Field Model Using Proper Orthogonal Decomposition. *Z. Angew. Math. Mech. 81(2):83-97, 2001.*

[44] **S. Volkwein.** Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling. *University of Konstanz, Lecture Notes, 2013.*

[45] **Z. Whang.** Nonlinear Model Reduction Based on the Finite Element Method With Interpolated Coefficients: Semilinear Parabolic Equations. *Numer. Meth. Partial. Diff. Eqs., 31(6):1713-1741, 2015.*

[46] **A. Yagi.** Abstract Parabolic Evolution Equations and their Applications. *Springer Monographs in Mathematics, 2010.*

[47] **M. Yano.** A minimum-residual mixed reduced basis method: Exact residual certification and simultaneous finite-element reduced-basis refinement. *ESAIM: M2AN 50(1), 163-185, 2016.*