MODEL ORDER REDUCTION BY
PROPER ORTHOGONAL DECOMPOSITION

CARMEN GRÄSSLE (UNIVERSITÄT HAMBURG),
MICHAEL HINZE (UNIVERSITÄT KOBLENZ-LANDAU),
STEFAN VOLKWEIN (UNIVERSITÄT KONSTANZ)

Abstract. We provide an introduction to POD-MOR with focus on (nonlinear) parametric PDEs and (nonlinear) time-dependent PDEs, and PDE constrained optimization with POD surrogate models as application. We cover the relation of POD and SVD, POD from the infinite-dimensional perspective, reduction of nonlinearities, certification with a priori and a posteriori error estimates, spatial and temporal adaptivity, input dependency of the POD surrogate model, POD basis update strategies in optimal control with surrogate models, and sketch related algorithmic frameworks. The perspective of the method is demonstrated with several numerical examples.

Key words. POD model order reduction, (discrete) empirical interpolation, adaptivity, parametric PDEs, evolutionary PDEs, certification with error analysis.

Mathematics Subject Classification. 35B30, 37M99, 41A05, 65K99, 93A15, 93C05

1. INTRODUCTION

Proper orthogonal decomposition (POD) is a method which comprises the essential information contained in data sets. Data sets may have their origin in various sources, like, e.g., (uncertain) measurements of geophysical processes, numerical simulations of (parameter-dependent) complex physical problems, or (dynamical) imaging. In order to illustrate the POD idea of information extraction let $\{y_1, \ldots, y_n\} \subset \mathbb{R}^m$ denote a vector cloud (which here serves as our data set), where we suppose at least one of the vectors y_j is nonzero. Let us collect the vectors y_j in the data matrix

$$Y = [y_1 \,|\, \ldots \,|\, y_n] \in \mathbb{R}^{m \times n}.$$

Then we have $\mathsf{r} = \operatorname{rank} Y \in \{1, \ldots, \min(m,n)\}$. Our aim now is to find a vector $\bar{\psi} \in \mathbb{R}^m$ with length one which carries as much information of this vector cloud as possible. Of course, we here have to specify what *information* in this context means. For this purpose we equip $\mathbb{R}^m$ with some inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$. We define the information content of vector y with respect to some unit vector ψ by the quantity $|\langle y, \psi \rangle|$. Then we determine the special vector $\bar{\psi} \in \mathbb{R}^m$ by solving the maximization problem

$$\bar{\psi} \in \arg\max\left\{ \sum_{j=1}^n \left|\langle y_j, \psi \rangle\right|^2 \,\Big|\, \psi \in \mathbb{R}^m \text{ with } \|\psi\| = 1 \right\}. \tag{1}$$

Notice that the solution to the maximization problem in (1) is not unique. If $\bar{\psi}$ is a vector, where the maximum is attained, then $-\bar{\psi}$ is an optimal solution, too. Let us label the vector $\bar{\psi}$ by ψ_1. We now iterate this procedure; suppose that for $2 \le \ell \le \mathsf{r}$ we have already computed such $\ell - 1$ orthonormal vectors $\{\psi_i\}_{i=1}^{\ell-1}$, then seek a unit vector $\psi_\ell \in \mathbb{R}^m$ which is perpendicular to the $(\ell-1)$-dimensional subspace

$$\mathcal{V}^{\ell-1} = \operatorname{span}\{\psi_1, \ldots, \psi_{\ell-1}\} \subset \mathbb{R}^m,$$

and which carries as much information of our vector cloud as possible, i.e., satisfies

$$\psi_\ell = \arg\max \left\{ \sum_{j=1}^n |\langle y_j, \psi \rangle|^2 \, \Big| \, \psi \in \mathbb{R}^m \text{ with } \|\psi\| = 1 \text{ and } \psi \perp \mathcal{V}^{\ell-1} \right\}.$$

It is now straightforward to see that the vectors $\{\psi_i\}_{i=1}^{\mathsf{r}}$ are given by

$$W^{1/2}\psi_i = \tilde{\psi}_i, \quad 1 \le i \le \mathsf{r}, \tag{2}$$

where the $\tilde{\psi}_i$'s solve the eigenvalue problem (cf. [42, 53])

$$\bar{Y}\bar{Y}^\top \tilde{\psi}_i = \lambda_i \tilde{\psi}_i, \quad i = 1, \dots, \mathsf{r} \text{ and } \lambda_1 \ge \dots \ge \lambda_{\mathsf{r}} > 0,$$

where $\bar{Y} = W^{1/2}Y \in \mathbb{R}^{m \times n}$ with the symmetric, positive definite (weighting) matrix

$$W = \big((\langle e_i, e_j \rangle) \big)_{1 \le i,j \le m}. \tag{3}$$

In (3) the vector e_i denotes the i-th unit vector in $\mathbb{R}^m$. The modes $\{\psi_i\}_{i=1}^\ell$ obtained in this way are called *POD Modes* or *Principal Components* of our data cloud. If now $m \gg n \ge \mathsf{r}$ it is advantageous to consider the eigenvalue problem

$$\bar{Y}^\top \bar{Y} \phi_i = \lambda_i \phi_i, \quad i = 1, \dots, \mathsf{r} \text{ and } \lambda_1 \ge \dots \ge \lambda_{\mathsf{r}} > 0,$$

which admits the same eigenvalues λ_i as before. The modes ψ_i and ϕ_i, $i = 1, \dots, \mathsf{r}$, are related by *singular value decomposition* (SVD):

$$\psi_i = \frac{1}{\sigma_i} \bar{Y} \phi_i, \quad i = 1, \dots, \mathsf{r},$$

and $\sigma_i = \sqrt{\lambda_i} > 0$ is the i-th singular value of the weighted data matrix $\bar{Y}$. Notice that in contrast to (2) the square root matrix $W^{1/2}$ is not required.

It is now clear that a vector cloud also could be replaced by a function cloud $\{y(\boldsymbol{\mu}_j) \, | \, j = 1, \dots, n\} \subset X$ in some Hilbert space $(X, \langle \cdot, \cdot \rangle_X)$, where $\{\boldsymbol{\mu}_j\}_{j=1}^n$ are parameters which may refer to, e.g., time instances of a dynamic process, or stochastic variables, and the concept of information extraction by the above maximization problems directly carries over to this situation. As it is shown in the next section we can even extend this concept to general Hilbert spaces. This will be formalized in Section 2.1 below. From the considerations above it also becomes clear that POD is closely related to SVD. This is outlined in Section 2.2. The POD method for abstract nonlinear evolution problems is explained in Section 2.3. The Hilbert space perspective also allows us to treat spatially discrete evolution equations, which include adaptive concepts for the spatial discretization. This is outlined in Section 2.4. The POD-Galerkin procedure is explained in Section 3, including a discussion of the treatment of nonlinearities. The certification of the POD method with a priori and a posteriori error bounds is outlined in Section 4. The POD approach heavily relies on the choice of the snapshots. Related approaches are discussed in Section 5. In Section 6 we briefly address the scope of the POD method in the context of optimal control of PDEs. Finally, in Section 7 we sketch further important research trends related to POD. Our analytical exposition is supported by several numerical experiments which give an impression of the power of the approach.

POD is one of the most successfully used model reduction techniques for nonlinear dynamical systems; see, e.g., [23, 42, 53, 75, 90] and the references therein. It is applied in a variety of fields including fluid dynamics, coherent structures [4, 9] and inverse problems [13]. Moreover in [11] POD is successfully applied to compute reduced-order controllers. The relationship between POD and balancing was considered in [61, 82, 101]. An error analysis for nonlinear dynamical systems in finite dimensions was carried out in [78] and a missing point estimation in models described by POD was studied in [10].

2. Proper Orthogonal Decomposition (POD)

In this section we introduce a discrete variant of the POD method, where we follow partially [42, Section 1.2.1]. For a continuous variant of the POD method and its relationship to the discrete one we refer the reader to [58] and [42, Sections 1.2.2 and 1.2.3].

2.1. **The POD method.** Suppose that $K, n_1, \ldots, n_K$ are fixed natural numbers. Let the so-called *snapshot ensembles* $\{y_j^k\}_{j=1}^{n_k} \subset X$ be given for $1 \le k \le K$, where X is a separable real Hilbert space. For POD in complex Hilbert spaces we refer the reader to [97]. We set $n = n_1 + \ldots + n_K$. To avoid a trivial case we suppose that at least one of the y_j^k's is nonzero. Then we introduce the finite dimensional, linear *snapshot space*

$$\mathcal{V} = \text{span} \left\{ y_j^k \,\middle|\, 1 \le j \le n_k \text{ and } 1 \le k \le K \right\} \subset X \tag{4}$$

with finite dimension $d \le n$. We distinguish two cases:

1) The separable Hilbert space X has finite dimension m: Then X is isomorphic to $\mathbb{R}^m$; see, e.g., [81, p. 47]. We define the finite index set $\mathbb{I} = \{1, \ldots, m\}$. Clearly, we have $1 \le r \le \min(n, m)$. Especially in case of $X = \mathbb{R}^m$, the snapshots $y_j^k = (y_{ij}^k)_{1 \le i \le m}$ are vectors in $\mathbb{R}^m$ for $k = 1, \ldots, K$.

2) X is infinite-dimensional: Since X is separable, each orthonormal basis of X has countably many elements. In this case X is isomorphic to the set ℓ_2 of sequences $\{x_i\}_{i \in \mathbb{N}}$ of complex numbers which satisfy $\sum_{i=1}^\infty |x_i|^2 < \infty$; see [81, p. 47], for instance. The index set $\mathbb{I}$ is now the countable, but infinite set $\mathbb{N}$.

The *POD method* consists in choosing a complete orthonormal basis $\{\psi_i\}_{i \in \mathbb{I}}$ in X such that for every $\ell \in \{1, \ldots, r\}$ the information content of the given snapshots y_j^k is maximized in the following sense:

$$\left. \begin{aligned} &\max \sum_{i=1}^\ell \sum_{k=1}^K \sum_{j=1}^{n_k} \alpha_j^k \left| \langle y_j^k, \psi_i \rangle_X \right|^2 \\ &\text{s.t. } \{\psi_i\}_{i=1}^\ell \subset X \text{ and } \langle \psi_i, \psi_j \rangle_X = \delta_{ij}, \ 1 \le i, j \le \ell \end{aligned} \right\} \tag{$\mathbf{P}^\ell$}$$

with positive weighting parameters α_j^k, $j = 1, \ldots, n_k$ and $k = 1, \ldots, K$. Here, the symbol δ_{ij} denotes the Kronecker symbol satisfying $\delta_{ii} = 1$ and $\delta_{ij} = 0$ for $i \ne j$.

An optimal solution $\{\Psi_i\}_{i=1}^\ell$ to $(\mathbf{P}^\ell)$ is called a *POD basis of rank* ℓ. It is proved in [42, Theorem 1.8] that for every $\ell \in \{1, \ldots, r\}$ a solution $\{\Psi_i\}_{i=1}^\ell$ to $(\mathbf{P}^\ell)$ is characterized by the eigenvalue problem

$$\mathcal{R}\Psi_i = \lambda_i \Psi_i \quad \text{for } 1 \le i \le \ell, \tag{5}$$

where $\lambda_1 \ge \ldots \ge \lambda_r > 0$ denote the largest eigenvalues of the linear, bounded, nonnegative and self-adjoint operator $\mathcal{R}: X \to X$ given as

$$\mathcal{R}\Psi = \sum_{k=1}^K \sum_{j=1}^{n_k} \alpha_j^k \langle \Psi, y_j^k \rangle_X y_j^k \quad \text{for } \Psi \in X. \tag{6}$$

Moreover, the operator $\mathcal{R}$ can be presented in the form

$$\mathcal{R} = \mathcal{Y}\mathcal{Y}^* \tag{7}$$

with the mapping

$$\mathcal{Y}: \mathbb{R}^n \to X, \quad \mathcal{Y}(\Phi) = \sum_{k=1}^K \sum_{j=1}^{n_k} \sqrt{\alpha_j^k} \phi_j^k y_j^k \quad \text{for } \Phi = \left(\phi_1^1, \ldots, \phi_{n_K}^K \right) \in \mathbb{R}^n,$$

where $\mathcal{Y}^*: X \to \mathbb{R}^n$ denotes the Hilbert space adjoint of $\mathcal{Y}$, whose action is given by

$$\mathcal{Y}^*(\Psi) = \left(\left\langle \Psi, \sqrt{\alpha_1^1} y_1^1 \right\rangle_X, \ldots, \left\langle \Psi, \sqrt{\alpha_{n_K}^K} y_{n_K}^K \right\rangle_X \right)^\top \quad \text{for } \Psi \in X.$$

The operator $\mathcal{K} : \mathbb{R}^n \to \mathbb{R}^n$, $\mathcal{K} := \mathcal{Y}^*\mathcal{Y}$ then admits the same nonzero eigenvalues $\lambda_1 \geq \ldots \geq \lambda_r > 0$ with corresponding eigenvectors $\Phi_1, \ldots, \Phi_r$, and its action is given by

$$\mathcal{K}\Phi = \sum_{k=1}^{K} \sum_{j=1}^{n_k} \left(\sqrt{\alpha_1^1 \alpha_j^k} \phi_j^k \langle y_j^k, y_1^1 \rangle_X, \ldots, \sqrt{\alpha_{n_K}^K \alpha_j^k} \phi_j^k \langle y_j^k, y_{n_K}^K \rangle_X \right)^\top \tag{8}$$

with the vector $\Phi = (\phi_1^1, \ldots, \phi_{n_K}^K) \in \mathbb{R}^n$. For the eigensystems of $\mathcal{R}$ and $\mathcal{K}$ there holds the relation

$$\Phi_i = \frac{1}{\sqrt{\lambda_i}} \mathcal{Y}^* \Psi_i, \text{ and } \Psi_i = \frac{1}{\sqrt{\lambda_i}} \mathcal{Y} \Phi_i, \quad \text{for } i = 1, \ldots, r. \tag{9}$$

Furthermore, we obtain

$$\sum_{i=1}^{\ell} \sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k \left| \langle y_j^k, \Psi_i \rangle_X \right|^2 = \sum_{i=1}^{\ell} \lambda_i,$$

and for the POD projection error we get

$$\sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k \left\| y_j^k - \sum_{i=1}^{\ell} \sum_{k=1}^{K} \sum_{j=1}^{n_k} \langle y_j^k, \Psi_i \rangle_X \Psi_i \right\|_X^2 = \sum_{i=\ell+1}^{r} \lambda_i. \tag{10}$$

Thus, the decay rate of the positive eigenvalues $\{\lambda_i\}_{i=1}^{r}$ plays an essential role for a successful application of the POD method. In general, one has to utilize a complete orthonormal basis $\{\Psi_i\}_{i \in \mathbb{I}} \subset X$ to represent elements in the snapshot space $\mathcal{V}$ by their Fourier sum. This leads to a high-dimensional or even infinite-dimensional approximation scheme. Nevertheless, if the term $\sum_{i=\ell+1}^{r} \lambda_i$ is sufficiently small for a not too large ℓ, elements in the subspace $\mathcal{V}$ can be approximated by a linear combination of the few basis elements $\{\Psi_i\}_{i=1}^{\ell}$. This offers the chance to reduce the number of terms in the Fourier series using the POD basis of rank ℓ, as shown in the following examples. For this reason it is useful to define information content of the basis $\{\Psi_i\}_{i=1}^{\ell}$ in $\mathcal{V}$ by the quantity

$$\mathcal{E}(\ell) = \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{r} \lambda_i} \in [0, 1]. \tag{11}$$

It can e.g. be utilized to determine a basis of length $\ell \in \{1, \ldots, r\}$ containing $\approx 99\%$ of the information contained in $\mathcal{V}$ by requiring $\mathcal{E}(\ell) \approx 99\%$. Now it is shown in [42, Section 1.2.1] that

$$\sum_{i=1}^{r} \lambda_i = \sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k \left\| y_j^k \right\|_X^2$$

holds true. This implies

$$\mathcal{E}(\ell) = \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k \left\| y_j^k \right\|_X^2} \in [0, 1],$$

so that the quantity $\mathcal{E}(\ell)$ can be computed without knowing the eigenvalues $\lambda_{\ell+1}, \ldots, \lambda_r$.

2.2. **Singular value decomposition and POD.** To investigate the relationship between singular value decomposition (SVD) and POD let us discuss the POD method for the specific case $X = \mathbb{R}^m$.

Then we define the matrices

$$
D^k = \begin{pmatrix} \alpha_1^k & & 0 \\ & \ddots & \\ 0 & & \alpha_{n_k}^k \end{pmatrix} \in \mathbb{R}^{n_k \times n_k} \quad \text{for } 1 \leq k \leq K,
$$

$$
D = \begin{pmatrix} D^1 & & 0 \\ & \ddots & \\ 0 & & D^K \end{pmatrix} \in \mathbb{R}^{n \times n},
$$

$$
Y^k = \left[y_1^k \,|\, \ldots \,|\, y_{n_k}^k \right] \in \mathbb{R}^{m \times n_k} \quad \text{for } 1 \leq k \leq K,
$$

$$
Y = \left[Y^1 \,|\, \ldots \,|\, Y^K \right] \in \mathbb{R}^{m \times n}, \quad \bar{Y} = W^{1/2} Y D^{1/2} \in \mathbb{R}^{m \times n},
$$

where we have introduced the weighting matrix $W \in \mathbb{R}^{m \times m}$ in (3).

Remark 1. Let us mention that $\bar{Y} = Y$ holds true provided all α_j^k are equal to one (i.e., D is the identity matrix) and the inner product in X is given by the Euclidean inner product (i.e., W is the identity matrix). $\diamond$

Now (5) is equivalent to the $m \times m$ eigenvalue problem

$$
\bar{Y} \bar{Y}^\top \bar{\Psi}_i = \lambda_i \bar{\Psi}_i \quad \text{for } 1 \leq i \leq \ell \tag{12}
$$

with $\Psi_i = W^{-1/2} \bar{\Psi}_i$ and the $n \times n$ eigenvalue problem

$$
\bar{Y}^\top \bar{Y} \bar{\Phi}_i = \lambda_i \bar{\Phi}_i \quad \text{for } 1 \leq i \leq \ell \tag{13}
$$

with $\Psi_i = Y D^{1/2} \bar{\Phi}_i / \sqrt{\lambda_i}$. If $m \ll n$ holds, we solve (12). However, we have to solve the linear system $W^{1/2} \Psi_i = \bar{\Psi}_i$ for any $i = 1, \ldots, \ell$ in order to get the POD basis $\{\Psi_i\}_{i=1}^{\ell}$. Thus, if $n \leq m$ holds, we will compute the solution $\{\bar{\Phi}_i\}_{i=1}^{\ell}$ to (13) and get the POD basis by the formula $\Psi_i = Y D^{1/2} \bar{\Phi}_i / \sqrt{\lambda_i}$. In that case we also have $\bar{Y}^\top \bar{Y} = Y^\top W Y$ so that we do not have to compute the square root matrix $W^{1/2}$. On the other hand, the diagonal matrix $D^{1/2}$ can be computed easily. The relationship between (12) and (13) is given by SVD: There exist real numbers $\sigma_1 \geq \ldots \geq \sigma_r > 0$ and orthogonal matrices $\Psi \in \mathbb{R}^{m \times m}$, $\Phi \in \mathbb{R}^{n \times n}$ with column vectors $\{\bar{\Psi}_i\}_{i=1}^{m}$, $\{\bar{\Phi}_i\}_{i=1}^{n}$, respectively, such that

$$
\Psi^\top \bar{Y} \Phi = \begin{pmatrix} \Sigma^{\mathsf{r}} & 0 \\ 0 & 0 \end{pmatrix} =: \Sigma \in \mathbb{R}^{m \times n}, \tag{14}
$$

where $\Sigma^{\mathsf{r}} = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\mathsf{r}}) \in \mathbb{R}^{\mathsf{r} \times \mathsf{r}}$ and the zeros in (14) denote matrices of appropriate dimensions. Moreover, the vectors $\{\bar{\Psi}_i\}_{i=1}^{\mathsf{r}}$ and $\{\bar{\Phi}_i\}_{i=1}^{\mathsf{r}}$ are eigenvectors of $\bar{Y} \bar{Y}^\top$ and $\bar{Y}^\top \bar{Y}$, respectively, with eigenvalues $\lambda_i = (\sigma_i)^2 > 0$ for $i = 1, \ldots, \mathsf{r}$. The vectors $\{\bar{\Psi}_i\}_{i=\mathsf{r}+1}^{m}$ and $\{\bar{\Phi}_i\}_{i=\mathsf{r}+1}^{n}$ (if $\mathsf{r} < m$ respectively $\mathsf{r} < n$) are eigenvectors of $\bar{Y} \bar{Y}^\top$ and $\bar{Y}^\top \bar{Y}$ with eigenvalue 0. We summarize the computation of the POD basis in the pseudo code **function** $[\Psi, \Lambda] = \texttt{POD}(Y, W, D, \ell, \texttt{flag})$.

---

function $[\Psi, \Lambda]$=$\texttt{POD}(Y, W, D, \ell, \texttt{flag})$

Require: Snapshots matrix $Y = [Y^1, \ldots, Y^K]$ with rank r, weighting matrices W, D, number ℓ of POD functions and $\texttt{flag}$ for the solver;

1: **if** $\texttt{flag} = 0$ **then**
2:     Set $\bar{Y} = W^{1/2} Y D^{1/2}$;
3:     Compute singular value decomposition $[\Psi, \Sigma, \Phi] = \texttt{svd}\,(\bar{Y})$;
4:     Define $\bar{\Psi}_i$ as the i-th column of Ψ and $\sigma_i = \Sigma_{ii}$ for $1 \le i \le \ell$;
5:     Set $\Psi_i = W^{-1/2} \bar{\Psi}_i$ and $\lambda_i = \sigma_i^2$ for $i = 1, \ldots, \ell$;
6: **else if** $\texttt{flag} = 1$ **then**
7:     Compute eigenvalue decomposition $[\Psi, \Lambda] = \texttt{eig}\,(\bar{Y}\bar{Y}^\top)$;
8:     Define $\bar{\Psi}_i$ as the i-th column of Ψ and $\lambda_i = \Lambda_{ii}$ for $1 \le i \le \ell$;
9:     Set $\Psi_i = W^{-1/2} \bar{\Psi}_i$ for $i = 1, \ldots, \ell$;
10: **else if** $\texttt{flag} = 2$ **then**
11:     Compute eigenvalue decomposition $[\Phi, \Lambda] = \texttt{eig}\,(\bar{Y}^\top \bar{Y})$;
12:     Define $\bar{\phi}_i$ as the i-th column of Φ and $\lambda_i = \Lambda_{ii}$ for $1 \le i \le \ell$;
13:     Set $\Psi_i = Y D^{1/2} \bar{\phi}_i / \sqrt{\lambda_i}$ for $i = 1, \ldots, \ell$;
14: **end if**
15: **return** $\Psi = [\Psi_1 \,|\, \ldots \,|\, \Psi_\ell]$ and $\Lambda = [\lambda_1 \,|\, \ldots \,|\, \lambda_\ell]$;

2.3. **The POD method for nonlinear evolution problems.** In this subsection we explain the POD method for abstract nonlinear evolution problems. We focus on the numerical realization. For detailed theoretical investigations we refer the reader to [42, 50, 51, 57, 58]; for instance.

2.3.1. *The nonlinear evolution problems.* Let us formulate the nonlinear evolution problem. For that purpose we suppose the following hypotheses.

Assumption 1. *Suppose that $T > 0$ holds, where $[0, T]$ is the considered finite time horizon.*

1) *V and H are real, separable Hilbert spaces and suppose that V is dense in H with compact embedding. By $\langle \cdot, \cdot \rangle_H$ and $\langle \cdot, \cdot \rangle_V$ we denote the inner products in H and V, respectively. We identify H with its dual (Hilbert) space H' by the Riesz isomorphism so that we have the Gelfand triple*

$$V \hookrightarrow H \simeq H' \hookrightarrow V',$$

where each embedding is continuous and dense. The last embedding is understood as follows: For every element $h \in H'$ and $v \in V$, we also have $v \in H$ by the embedding $V \hookrightarrow H$, so we can define $\langle h', v \rangle_{V', V} = \langle h', v \rangle_{H', H}$.

2) *For almost all $t \in [0, T]$ we define a time-dependent bilinear form $a(t; \cdot, \cdot) : V \times V \to \mathbb{R}$ satisfying*

$$\big| a(t; \varphi, \phi) \big| \le \gamma \, \|\varphi\|_V \|\phi\|_V \qquad \text{for all } \varphi, \phi \in V, \qquad t \in [0, T] \text{ a.e.,} \qquad (15a)$$

$$a(t; \varphi, \varphi) \ge \gamma_1 \, \|\varphi\|_V^2 - \gamma_2 \, \|\varphi\|_H^2 \qquad \text{for all } \varphi \in V, \qquad t \in [0, T] \text{ a.e.} \qquad (15b)$$

for time-independent constants $\gamma, \gamma_2 \ge 0$, $\gamma_1 > 0$ and where a.e. stands for almost everywhere.

3) *Assume that $y_\circ \in V$, $f \in L^2(0, T; H)$ holds. Here we refer to [27, pp. 469-472] for vector-valued function spaces.*

Recall the function space

$$W(0, T) = \big\{ \varphi \in L^2(0, T; V) \,\big|\, \varphi_t \in L^2(0, T; V') \big\}$$

which is a Hilbert space endowed with the standard inner product; cf. [27, pp. 472-479]. Furthermore, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} \, \langle \varphi(t), \phi \rangle_H = \langle \varphi_t(t), \phi \rangle_{V', V} \quad \text{for } \varphi \in W(0, T), \ \phi \in V$$

in the sense of distributions in $[0, T]$. Here, $\langle \cdot, \cdot \rangle_{V', V}$ stands for the dual pairing between V and its dual V'.

Now the evolution problem is given as follows: Find the state $y \in W(0, T) \cap C([0, T]; V)$ such that

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), \varphi \rangle_H + a(t; y(t), \varphi) + \langle \mathcal{N}(y(t)), \varphi \rangle_{V', V} = \langle f(t), \varphi \rangle_H$$
$$\forall \varphi \in V, t \in (0, T] \text{ a.e.,} \qquad (16)$$
$$\langle y(0), \varphi \rangle_H = \langle y_\circ, \varphi \rangle_H \quad \forall \varphi \in H.$$

Throughout we assume that (16) admits a unique solution $y \in W(0, T) \cap C([0, T]; V)$. Of course, this requires some properties for the nonlinear mapping $\mathcal{N}$ which we will not specify here.

Example 1 (Semilinear heat equation). Let $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$ 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 $y_\circ \in L^2(\Omega)$, we consider the semilinear heat equation with homogeneous Dirichlet boundary condition:

$$\left.\begin{array}{rl} y_t(t, \boldsymbol{x}) - \Delta y(t, \boldsymbol{x}) + cy^3(t, \boldsymbol{x}) = f(t, \boldsymbol{x}) & \text{in } Q, \\ y(t, \boldsymbol{x}) = 0 & \text{on } \Sigma, \\ y(0, \boldsymbol{x}) = y_\circ(\boldsymbol{x}) & \text{in } \Omega. \end{array}\right\} \qquad (17)$$

The existence of a unique solution to (17) is proved in [83], for example. We can write (17) as an abstract evolution problem of type (16) by deriving a variational formulation for (17) with $V = H_0^1(\Omega)$ as the space of test functions, $H = L^2(\Omega)$ and integrating over the space Ω. The bilinear form $a : V \times V \to \mathbb{R}$ is introduced by

$$a(\varphi, \phi) = \int_\Omega \nabla\varphi \cdot \nabla\phi \, \mathrm{d}\boldsymbol{x} \quad \text{for } \varphi, \phi \in V$$

and the operator $\mathcal{N} : V \to V'$ is defined as $\mathcal{N}(\varphi) = c\varphi^3$ for $\varphi \in V$. For $c \equiv 0$, the heat equation (17) is linear. $\Diamond$

Example 2 (Cahn-Hilliard equations). Let Ω, T, Q and Σ be defined as in Example 1. The Cahn-Hilliard system was proposed in [21] 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:

$$\left.\begin{array}{rl} c_t(t, \boldsymbol{x}) + y \cdot \nabla c(t, \boldsymbol{x}) = \mathsf{m}\Delta w(t, \boldsymbol{x}) & \text{in } Q, \\ w(t, \boldsymbol{x}) = -\sigma\varepsilon\Delta c(t, \boldsymbol{x}) + \dfrac{\sigma}{\varepsilon}W'\big(c(t, \boldsymbol{x})\big) & \text{in } Q, \\ \nabla c(t, \boldsymbol{x}) \cdot \nu_\Omega = \nabla w(t, \boldsymbol{x}) \cdot \nu_\Omega = 0 & \text{on } \Sigma, \\ c(0, x) = c_\circ(\boldsymbol{x}) & \text{in } \Omega. \end{array}\right\} \qquad (18)$$

By ν_Ω we denote the outward normal on $\partial\Omega$, $\mathsf{m} \geq 0$ 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. [52] and [2]. The phase field function c describes the phase of a binary material with components A and B and takes the values $c \equiv -1$ 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 admits a thickness of order $\mathcal{O}(\varepsilon)$, see e.g. Fig. 5, left column, where the binary phases are colored in blue and red, respectively, and the interfacial region is depicted in white. The function $W(c)$ represents the free energy and is of double well-type. A typical choice for W is the polynomial free energy function

$$W^p(c) = (1 - c^2)^2/4 \qquad (19)$$

with two minima at $c = \pm 1$, which describe the energetically favorable states. It is infinitely often differentiable. Another choice for W is the C^1 relaxed double obstacle free energy

$$W_s^{\mathrm{rel}}(c) = \frac{1}{2}(1 - c^2) + \frac{s}{2}(\max(c - 1, 0)^2 + \min(c + 1, 0)^2), \tag{20}$$

with relaxation parameter $s \gg 0$, which is introduced in [43] as the Moreau-Yosida relaxation of the double obstacle free energy

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

The energies $W^p(c)$ and $W_s^{\mathrm{rel}}(c)$ later will be used to compare the performance of POD on systems with smooth and less smooth nonlinearities. For more details on the choices for W we refer to [1] and [19], for example. Concerning existence, uniqueness and regularity of a solution to (18), we refer to [19]. In order to derive a variational form of type (16), we rewrite (18) as a single fourth-order parabolic equation for c by

$$\left. \begin{aligned} c_t(t, \boldsymbol{x}) + y \cdot \nabla c(t, \boldsymbol{x}) &= \mathsf{m}\Delta\left(-\sigma\varepsilon\Delta c(t, \boldsymbol{x}) + \frac{\sigma}{\varepsilon}W'(c(t, \boldsymbol{x}))\right) && \text{in } Q, \\ 0 = \nabla c(t, \boldsymbol{x}) \cdot \nu_\Omega &= \nabla\left(-\sigma\varepsilon\Delta c(t, \boldsymbol{x}) + \frac{\sigma}{\varepsilon}W'(c(t, \boldsymbol{x}))\right) \cdot \nu_\Omega && \text{on } \Sigma, \\ c(0, \boldsymbol{x}) &= c_\circ(\boldsymbol{x}) && \text{in } \Omega. \end{aligned} \right\} \tag{21}$$

We choose $V = \{v \in H^1(\Omega) : \frac{1}{|\Omega|}\int_\Omega v = 0\}$ equipped with the inner product $(u, v)_V := \int_\Omega \nabla u \nabla v$, so that the dual space of V is given by $V' = \{f \in (H^1(\Omega))' : \langle f, 1 \rangle = 0\}$ such that $V \hookrightarrow H = V'$ and $\langle ., . \rangle$ denotes the duality pairing. We note that $(V, (., .)_V)$ is a Hilbert space. We define the V'−inner product for $f, g \in V'$ as $(f, g)_{V'} := \int_\Omega \nabla(-\Delta)^{-1}f \cdot \nabla(-\Delta)^{-1}g$ where $(-\Delta)^{-1}$ denotes the inverse of the negative Laplacian with zero Neumann boundary data. Note that $(f, g)_{V'} = (f, (-\Delta)^{-1}g)_{L^2(\Omega)} = ((-\Delta)^{-1}f, g)_{L^2(\Omega)}$. We introduce the bilinear form $a : V \times V \to \mathbb{R}$ by

$$a(u, v) = \sigma\varepsilon(\nabla u, \nabla v)_{L^2(\Omega)} + \frac{1}{\mathsf{m}}(y \cdot \nabla u, v)_{V'}$$

and define the nonlinear operator $\mathcal{N}$ by $\mathcal{N}(c) = \frac{\sigma}{\varepsilon}W'(c)$. The evolution problem can be written in the form

$$\frac{1}{\mathsf{m}}(c_t(t), v)_{V'} + a(c(t), v) + \langle \mathcal{N}(c(t)), v \rangle = 0 \quad \forall v \in V \text{ and a.a. } t \in (0, T].^1 \tag{22}$$

We note that this fits our abstract setting formulated in (16) with the Gelfand triple $V \hookrightarrow H \equiv V' \hookrightarrow V'$. $\diamond$

2.3.2. *Temporal discretization and POD method.* Let $0 = t_1 < \ldots < t_{n_t} = T$ be a given time grid with step sizes $\Delta t_j = t_j - t_{j-1}$ for $j = 2, \ldots, n_t$. Suppose that for any $j \in \{1, \ldots, n_t\}$ the element $y_j \in V \subset H$ is an approximation of $y(t_j)$ computed by applying a temporal integration method (e.g., the implicit Euler method) to (16). Then we consider the snapshot ensemble

$$\mathcal{V} = \operatorname{span}\{y_j \,|\, 1 \le j \le n\} \subset V \subset H$$

with $n = n_t$ and $\mathsf{r} = \dim \mathcal{V} \le n$. In the context of $(\mathbf{P}^\ell)$ we choose $K = 1$ and $n = n_1 = n_t$. Moreover, X can be either V or H. For the weighting parameters we take the trapezoidal weights

$$\alpha_1 = \frac{\Delta t_1}{2}, \quad \alpha_j = \frac{\Delta t_j + \Delta t_{j-1}}{2} \text{ for } j = 2, \ldots, n_t - 1, \quad \alpha_{n_t} = \frac{\Delta t_{n_t}}{2}. \tag{23}$$

---

[1]We acknowledge a hint of Harald Garcke who pointed this form of the weak formulation of (21) to us.

Of course, other quadrature weights are also possible. Now, instead of $(\mathbf{P}^\ell)$ we consider the minimization problem

$$
\left.
\begin{aligned}
& \max \sum_{i=1}^{\ell} \sum_{j=1}^{n} \alpha_j \left| \langle y_j, \Psi_i \rangle_X \right|^2 \\
& \text{s.t. } \{\Psi_i\}_{i=1}^{\ell} \subset X \text{ and } \langle \Psi_i, \Psi_j \rangle_X = \delta_{ij}, \ 1 \le i, j \le \ell
\end{aligned}
\right\}
\tag{24}
$$

with either $X = V$ or $X = H$.

Remark 2. In [42, Sections 1.2.2 and 1.3.2] a continuous variant of the POD method is considered. In that case the trapezoidal approximation in (24) is replaced by integrals over the time interval $[0, T]$. More precisely, we consider

$$
\left.
\begin{aligned}
& \max \sum_{i=1}^{\ell} \int_0^T \left| \langle y(t), \Psi_i \rangle_X \right|^2 \mathrm{d}t \\
& \text{s.t. } \{\Psi_i\}_{i=1}^{\ell} \subset X \text{ and } \langle \Psi_i, \Psi_j \rangle_X = \delta_{ij}, \ 1 \le i, j \le \ell
\end{aligned}
\right\}
\tag{25}
$$

with either $X = V$ or $X = H$. For the relationship between solutions to (24) and (25) we refer to [58] and [42, Section 1.2.3]. $\diamond$

To compute the POD basis $\{\Psi_i\}_{i=1}^{\ell}$ of rank ℓ we have to evaluate the inner products $\langle y_j, \Psi_i \rangle_X$, where either $X = V$ or $X = H$ hold. In typical applications the space X is usually infinite dimensional. Therefore, a discretization of X is required in order to get a POD method that can be realized on a computer. This is the topic of the next subsection.

2.3.3. *Galerkin discretization.* We discretize the state equation by applying any spatial approximation scheme. Let us consider here a Galerkin scheme for (16). For this reason we are given linearly independent elements $\varphi_1, \dots, \varphi_m \in V$ and define the m-dimensional subspace

$$
V^h = \operatorname{span} \{ \varphi_1, \dots, \varphi_m \} \subset V
$$

endowed with the V topology. Then a Galerkin scheme for (16) is given as follows: find $y^h \in W(0, T) \cap C([0, T]; V^h)$ satisfying

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \langle y^h(t), \varphi^h \rangle_H &+ a(t; y^h(t), \varphi^h) + \langle \mathcal{N}(y^h(t)), \varphi \rangle_{V', V} \\
&= \langle f(t), \varphi^h \rangle_H \quad \forall \varphi^h \in V^h, \ t \in (0, T] \text{ a.e.,} \\
\langle y^h(0), \varphi^h \rangle_H &= \langle y_\circ, \varphi^h \rangle_H \quad \forall \varphi^h \in V^h.
\end{aligned}
\tag{26}
$$

Inserting the representation $y^h(t) = \sum_{i=1}^{m} \mathrm{y}_i^h(t) \varphi_i \in V^h$, $t \in [0, T]$, in (26) and choosing $\varphi^h = \varphi_i$ for $i = 1, \dots, m$ we derive the following m-dimensional initial value problem

$$
\begin{aligned}
\mathrm{M}^h \dot{\mathrm{y}}^h(t) + \mathrm{A}^h(t) \mathrm{y}^h(t) + \mathrm{N}^h(\mathrm{y}^h(t)) &= \mathrm{F}^h(t) \text{ for } t \in (0, T], \\
\mathrm{M}^h \mathrm{y}^h(0) &= \mathrm{y}_\circ^h,
\end{aligned}
\tag{27}
$$

where we have used the matrices and vectors

$$
\begin{aligned}
\mathrm{M}^h &= \left(\left(\langle \varphi_j, \varphi_i \rangle_H \right) \right)_{1 \le i,j \le m}, & \mathrm{y}^h(t) &= \left(\mathrm{y}_i^h \right)_{1 \le i \le m} \text{ for } t \in [0, T] \text{ a.e.,} \\
\mathrm{A}^h(t) &= \left(\left(a(t; \varphi_j, \varphi_i) \right) \right)_{1 \le i,j \le m}, & \mathrm{y}_\circ^h &= \left(\langle y_\circ, \varphi_i \rangle_H \right)_{1 \le i \le m}, \\
\mathrm{N}^h(\mathrm{v}) &= \left(\left\langle \mathcal{N} \left(\sum_{j=1}^{m} \mathrm{v}_j \varphi_j \right), \varphi_i \right\rangle_{V', V} \right)_{1 \le i \le m} & & \text{for } \mathrm{v} = \left(\mathrm{v}_j \right)_{1 \le j \le m}, \\
\mathrm{F}^h(t) &= \left(\langle f(t), \varphi_i \rangle_H \right)_{1 \le i \le m} \text{ for } t \in [0, T]. & &
\end{aligned}
$$

In the pseudo code **function** $[Y] = \texttt{StateSol}(\mathrm{y}_\circ^h)$ we present a solution method for (27) using the implicit Euler method.

---

function $[Y] = \texttt{StateSol}(\mathrm{y}_\circ^h)$

Require: Initial condition $\mathrm{y}_\circ^h$;
  1: Compute $\mathrm{y}_1^h \in \mathbb{R}^m$ solving $\mathrm{M}^h \mathrm{y}_1^h = \mathrm{y}_\circ^h$;
  2: **for** $j = 2$ **to** n_t **do**
  3:   Set $\mathrm{A}_j^h = \mathrm{A}^h(t_j) \in \mathbb{R}^{m \times m}$ and $\mathrm{F}_j^h = \mathrm{F}^h(t_j) \in \mathbb{R}^m$;
  4:   Solve (e.g., by applying Newton's method) for $\mathrm{y}_j^h \in \mathbb{R}^m$

$$\big(\mathrm{M}^h + \Delta t_j \mathrm{A}_j^h\big)\mathrm{y}_j^h + \Delta t_j \mathrm{N}^h(\mathrm{y}_j^h) = \mathrm{M}^h \mathrm{y}_{j-1}^h + \Delta t_j \mathrm{F}_j^h;$$

  5: **end for**
  6: **return** matrix $Y = [\mathrm{y}_1^h | \dots | \mathrm{y}_{n_t}^h] \in \mathbb{R}^{m \times n_t}$;

In the next subsection we discuss how a POD basis $\{\Psi_j\}_{j=1}^\ell$ of rank $\ell \leq \mathsf{r}$ can be computed from numerical approximations for the solution y^h to (27).

2.3.4. *POD method for the fully discretized nonlinear evolution problem.* Recall that we have introduced the temporal grid $\{t_j\}_{j=1}^{n_t} \subset [0, T]$ and set $n = n_t$. Let $y_1^h, \dots, y_n^h \in V^h$ be numerical approximations to the solution $y^h(t)$ to (27) at time instances $t = t_j$, $j = 1 \dots, n_t$. Then, a coefficient matrix $Y \in \mathbb{R}^{m \times n}$ is defined by the elements Y_{ij} given by

$$y_j^h = \sum_{i=1}^m Y_{ij} \varphi_i \in V^h \quad \text{for } 1 \leq j \leq n.$$

The j-th column of Y (denoted by $\mathrm{y}_j = Y_{\cdot,j}$) contains the Galerkin coefficients of the snapshot $y_j^h \in V^h$. We set $\mathsf{r} = \operatorname{rank} Y \leq \min(m, n)$ and

$$\mathcal{V}^h = \operatorname{span}\big\{y_j^h \,\big|\, 1 \leq j \leq n\big\} \subset V^h.$$

Due to $\mathcal{V}^h \subset V^h$ we have $\Psi_j \in V^h$ for $1 \leq j \leq \ell$. Therefore, there exists a coefficient matrix $\Psi \in \mathbb{R}^{m \times \ell}$ that is defined by the elements Ψ_{ij} satisfying

$$\Psi_j = \sum_{i=1}^m \Psi_{ij} \varphi_i \in V^h \quad \text{for } 1 \leq j \leq \ell,$$

where the j-th column $\Psi_{\cdot,j}$ of the matrix Ψ consists of the Galerkin coefficients of the element Ψ_j. Notice that

$$\langle v^h, w^h \rangle_H = (\mathrm{v}^h)^\top \mathrm{M}^h \mathrm{w}^h, \quad \langle v^h, w^h \rangle_H = (\mathrm{v}^h)^\top \mathrm{S}^h \mathrm{w}^h$$

hold for $v^h = \sum_{i=1}^m \mathrm{v}_i^h \varphi_i$, $w^h = \sum_{i=1}^m \mathrm{w}_i^h \varphi_i \in V^h$ and for the symmetric, positive definite stiffness matrix

$$\mathrm{S}^h = \big(\!\big(\langle \varphi_j, \varphi_i \rangle_V\big)\!\big)_{1 \leq i, j \leq m}.$$

Then, we have for $X = H$

$$\langle y_j^h, \Psi_i \rangle_X = \mathrm{y}_j^\top \mathrm{M}^h \Psi_{\cdot,i} = Y_{\cdot,j}^\top \mathrm{M}^h \Psi_{\cdot,i} \quad \text{for } 1 \leq j \leq n,\ 1 \leq i \leq \ell,$$

and for $X = V$

$$\langle y_j^h, \Psi_i \rangle_X = \mathrm{y}_j^\top \mathrm{S}^h \Psi_{\cdot,i} = Y_{\cdot,j}^\top \mathrm{S}^h \Psi_{\cdot,i} \quad \text{for } 1 \leq j \leq n,\ 1 \leq i \leq \ell.$$

Thus, we can apply the approach presented in Section 2.2 choosing $W = \mathrm{M}^h$ for $X = H$ and $W = \mathrm{S}^h$ for $X = V$. Moreover, we set $K = 1$, $n_1 = n_t = n$ and $\alpha_j^1 = \alpha_j$ defined in (23). Now a POD basis of rank ℓ for (27) can be computed by the pseudo code **function** $[Y, \Psi] = \texttt{PODState}(\mathrm{y}_\circ^h, W, D, \ell, \texttt{flag})$.

---

function $[Y,\ \Psi]$=PODState$\big(\mathrm{y}_\circ^h,\ W,\ D,\ \ell,\ \mathtt{flag}\big)$

Require: Initial condition $\mathrm{y}_\circ^h$, weighting matrices W, D, number
    ℓ of POD functions and $\mathtt{flag}$ for the solver;
1: Call $[Y]$=StateSol$\big(\mathrm{y}_\circ^h\big)$;
2: Call $[\Psi,\Lambda]$=POD$(Y,\ W,\ D,\ \ell,\ \mathtt{flag})$;
3: **return** $Y=[\mathrm{y}_1^h\,|\ldots|\,\mathrm{y}_{n_t}^h]$ and $\Psi=[\Psi_1\,|\ldots|\,\Psi_\ell]$;

In the next subsection we will discuss in detail how the POD method has to be applied in that case if we have – instead of V^h – different spaces V^{h_j} for each $j=1,\ldots,n$.

2.4. **The POD method with snapshots generated by spatially adaptive finite element methods.** In practical applications it often is desirable to provide POD models for time-dependent PDE systems, whose numerical treatment requires adaptive numerical techniques in space and/or time. Snapshots generated by those methods are not directly amenable to the POD procedure described in Section 2.3.4, since the application of spatial adaptivity means, that the snapshots at each time instance may have different lengths due to their different spatial resolutions. In fact, there is not one single discrete Galerkin space V^h for all snapshots generated by the fully discrete evolution, but at every time instance t_j the adaptive procedure generates a discrete Galerkin space $V^{h_j}\subset X$, so that in this case $y_j^h\equiv y_j^{h_j}\in V^{h_j}$. For this reason, no snapshot matrix Y can be formed with columns containing the basis coefficient vectors of the snapshots.

To obtain also a POD basis in this situation we inspect the operator $\mathcal{K}$ of (8) and observe that its action can be computed if the inner products $\langle y_j^k, y_i^l\rangle_X$ can be evaluated for all $1\le i\le n_l, 1\le j\le n_k$ and $1\le k,l\le K$.

Let us next demonstrate how to compute a POD basis for snapshots residing in arbitrary finite element (FE) spaces. To begin with we drop the superindex h and set $V_j:=V^{h_j}$. For each time instant $j=1,\ldots,n$ of our time discrete PDE system the snapshots $\{y_j\}_{j=1}^n$ are taken from different finite element spaces $V_j\subseteq X$ $(j=1,\ldots,n)$, where X denotes a common (real) Hilbert space. Let $V_j=\operatorname{span}\{\varphi_1^j,\ldots,\varphi_{m_j}^j\}$. Then we have the expansions

$$y_j=\sum_{i=1}^{m_j}\mathrm{y}_j^i\varphi_i^j\quad\text{for }j=1,\ldots,n\tag{28}$$

with coefficient vectors

$$\mathrm{y}_j=\big(\mathrm{y}_j^i\big)\in\mathbb{R}^{m_j}\quad\text{for }j=1,\ldots,n$$

containing the finite element coefficients. The inner product of the associated functions can thus be computed as

$$\langle y_i,y_j\rangle_X=\sum_{k=1}^{m_i}\sum_{l=1}^{m_j}\mathrm{y}_i^k\mathrm{y}_j^l\,\langle\varphi_k^i,\varphi_l^j\rangle_X\quad\text{for }i,j=1,\ldots,n,$$

so that the evaluation of the action $\mathcal{K}\Phi$ only relies on the evaluation of the inner products $\langle\varphi_k^i,\varphi_l^j\rangle_X$ $(1\le i,j\le n,\ 1\le k\le m_i,\ 1\le l\le m_j)$. In other words, once we are able to compute those inner products we are in the position to set up the eigensystem $\{(\lambda_i,\Phi_i)\}_{i=1}^{\mathsf{r}}$ of $\mathcal{K}$ from (8). The POD modes $\{\Psi_i\}_{i=1}^{\mathsf{r}}$ can then be computed according to (9) by

$$\Psi_i=\frac{1}{\sqrt{\lambda_i}}\mathcal{Y}\Phi_i\quad\text{for }i=1,\ldots,\mathsf{r}.$$

Details on this procedure can be found in [62, 39].

To illustrate how this procedure can be implemented we summarize Examples 6.1-6.3 from [39], which deal with nested and non-nested meshes. All coding was done in C++ with using FEniCS [8, 66] for the solution of the differential equations and ALBERTA [87] for dealing with hierarchical meshes. The numerical tests were run on a compute server with 512 GB RAM.

Run 1 ([39, Example 6.1]). We consider the Example 1 with homogeneous Dirichlet boundary condition and vanishing nonlinearity, i.e. we set $c \equiv 0$ so 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 $[0,T] = [0, 1.57]$. Furthermore, we choose $X = L^2(\Omega)$. For the temporal discretization we introduce the uniform time grid by

$$t_j = (j-1)\Delta t \quad \text{for } j = 1, \ldots, n_t = 1571$$

with $\Delta t = 0.001$. For the spatial discretization we use h-adapted piecewise linear, continuous finite elements on hierarchical and nested meshes. Snapshots of the analytical solution at three different time points are shown in Figure 1. Details on the construction of the analytical solution and the corresponding right hand side f are given in [39, Example 6.1].
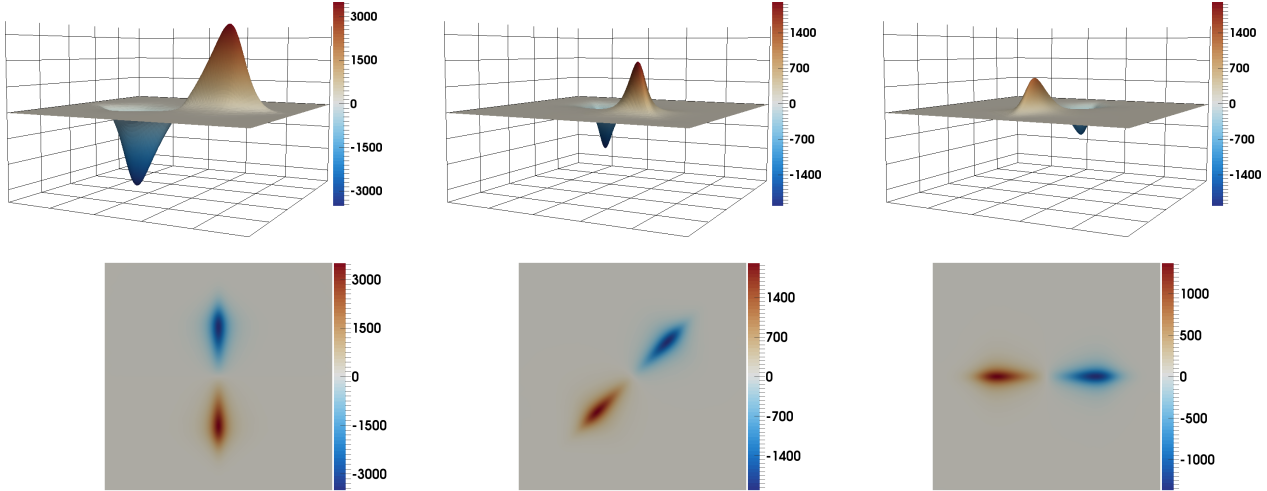
FIGURE 1. *Run 1. Surface plot (top) and view from above (bottom) of the analytical solution of* (17) *at* $t = t_1$ *(left),* $t = T/2$ *(middle) and* $t = T$ *(right).*

Due to the steep gradients in the neighborhood 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 which is the union of all adaptive meshes generated during the simulation) are shown in Figure 2.
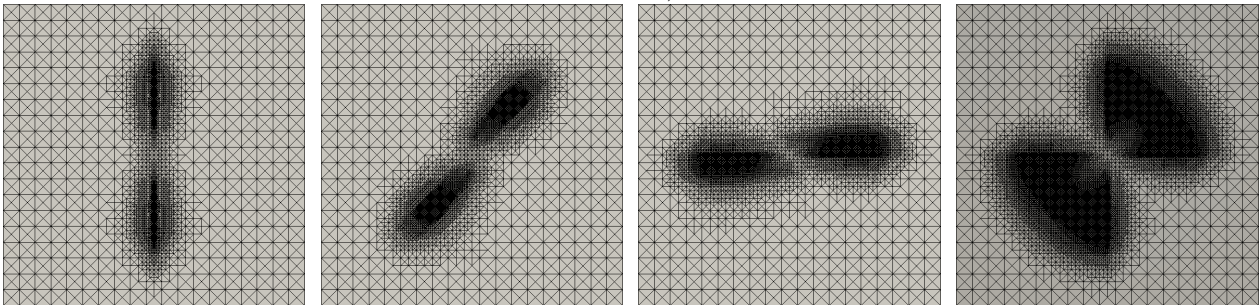

FIGURE 2. *Run 1. Adaptive finite element meshes at* $t = t_1$ *(left),* $t = T/2$ *(middle left),* $t = T$ *(middle right) and finest mesh (right).*

The number of nodes of the adaptive meshes varies between 3637 and 7071 points. The finest mesh has 18628 degrees of freedom. A uniform mesh with grid size of order of the diameter of the smallest triangles in the adaptive grids ($h_{\min} = 0.0047$) would have 93025 degrees of freedom. This clearly reveals the benefit of using adaptive meshes for snapshot generation which is also well reflected in the comparison of the computational times needed for the snapshot generation on the adaptive mesh taking 944 seconds compared to 8808 seconds on the uniform mesh, see Table 4) for the speedup

factors obtained by spatial adaptation. In Figure 3, the resulting normalized eigenspectrum of the correlation matrix K is shown for snapshots obtained by uniform spatial discretization ("uniform FE mesh"), for snapshots obtained by interpolation on the finest mesh ("adaptive FE mesh"), and for snapshots without interpolation ("infPOD"), where K is associated to the operator $\mathcal{K}$ from (8), see also (30).
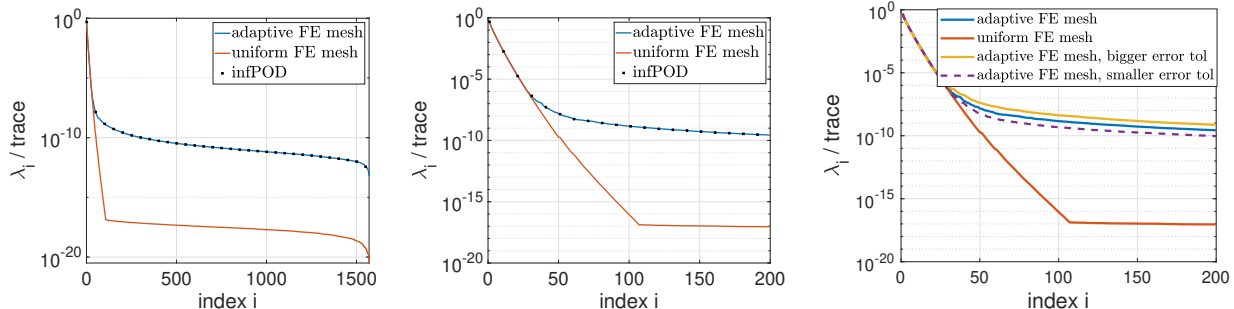

FIGURE 3. *Run 1. Comparison of the normalized eigenvalues using an adaptive and a uniform spatial mesh, respectively. Left: all eigenvalues, middle: first 200 largest eigenvalues, right: first 200 largest eigenvalues with different error tolerances for the adaptivity (1.5 times bigger and smaller error tolerances, respectively).*

We observe that the eigenvalues for both adaptive approaches coincide. This numerically validates what we expect from theory: the information content which is contained in the matrix $\mathcal{K}$ when we explicitly compute the entries without interpolation is the same as the information content contained within the eigenvalue problem which is formulated when using the finest mesh. No information is added or lost. 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}. If the error tolerance for the spatial discretization error is chosen larger (or smaller), the stagnation of the eigenvalues in the adaptive method takes place at a higher (or lower) order (see Figure 3, right). 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 for faster snapshot generation using adaptive methods. A further aspect gained from the decay behavior of the eigenvalues in the adaptive case is the following; the adaptive approach filters out the noise in the system which is related to the modes corresponding to the singular values that are not matched by the eigenvalues of the adaptive approach. This in the language of frequencies means that the overtones in the systems which get lost in the adaptive computations live in the space which is neglected by the POD method based on adaptive finite element snapshots. From this point of view, adaptivity can be interpreted as a smoother.

The first, second and fifth POD modes of Run 1 obtained by the adaptive approach are depicted in Figure 4. We observe the classical appearance of the basis functions. The initial condition is reflected by the first POD basis function. The next basis functions admit a number of minima and maxima corresponding to the index in the basis: Ψ_2 has two minima and two maxima etc. This behavior is similar to the increasing oscillations in higher frequencies in trigonometric approximations. The POD basis functions corresponding to the uniform spatial discretization have a similar appearance.◊

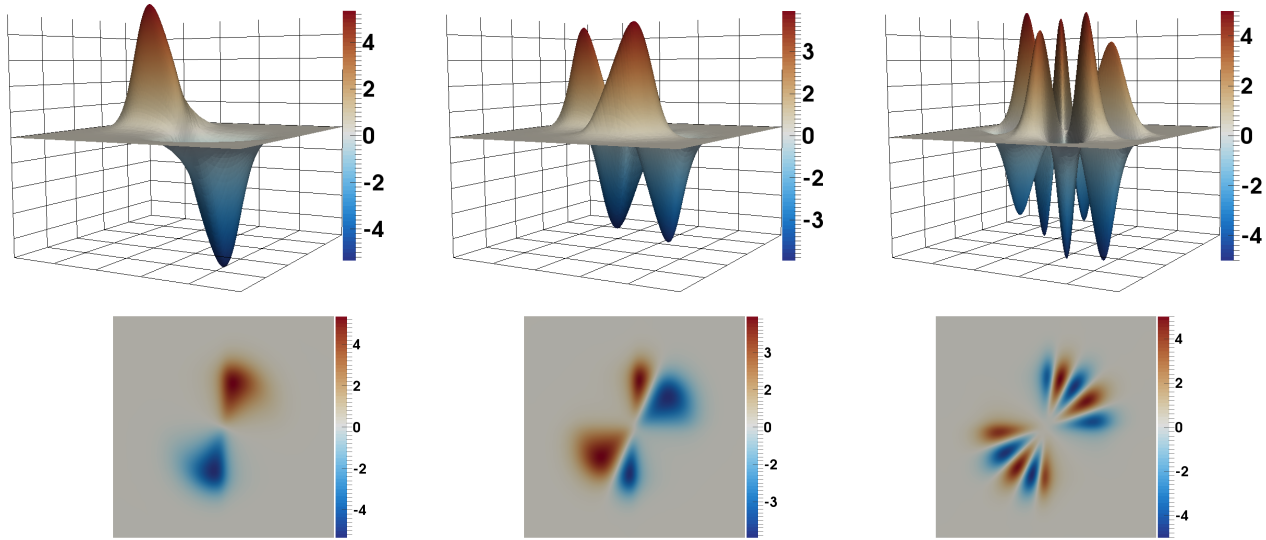FIGURE 4. *Run* 1. *Surface plot (top) and view from above (bottom) of the POD basis functions* Ψ_1 *(left),* Ψ_2 *(middle) and* Ψ_5 *(right).*

Run 2 ([39, Example 6.2]). *(Cahn-Hilliard system.)* We consider Example 2 in the form (18) with $\Omega = (0, 1.5) \times (0, 0.75)$, $T = 0.025$, constant mobility $\mathsf{m} \equiv 0.00002$, and constant surface tension $\sigma \equiv 24.5$. The interface parameter ε is set to $\varepsilon = 0.02$, with resulting interface thickness $\pi \cdot \varepsilon \approx 0.0628$. We use the relaxed double obstacle free energy W_s^{rel} from (20) 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 $v = (30, 0)^T$. We set

$$t_j = (j - 1)\Delta t \quad \text{for } j = 1, \ldots, n_t = 1001,$$

so that $\Delta t = 2.5 \cdot 10^{-5}$. The numerical computations are performed with the semi-implicit Euler scheme. For this purpose let $c^{j-1} \in V$ and $c^j \in V$ denote the time-discrete solution at t_{j-1} and t_j, respectively. Based on the variational formulation (22) we tackle the time discrete version of (18) in the form: given c^{j-1}, find c^j, w^j solving

$$\left.\begin{aligned}
\frac{1}{\Delta t} \langle c^j - c^{j-1}, \varphi_1 \rangle_{L^2} + \langle v \cdot \nabla c^{j-1}, \varphi_1 \rangle_{L^2} + \mathsf{m} \langle \nabla w^j, \nabla \varphi_1 \rangle_{L^2} = 0, \\
-\langle w^j, \varphi_2 \rangle_{L^2} + \sigma \varepsilon \langle \nabla c^j, \nabla \varphi_2 \rangle_{L^2} + \frac{\sigma}{\varepsilon} \langle W'_+(c^j) + W'_-(c^{j-1}), \varphi_2 \rangle_{L^2} = 0
\end{aligned}\right\} \quad (29)$$

for all $\varphi_1, \varphi_2 \in V$ and $j = 2, \ldots, n_t$ with $c^1 = c_\circ$. According to (22), here it is $V = \{v \in H^1(\Omega), \frac{1}{|\Omega|} \int_\Omega v \, dx = 0\}$. 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, whereas W'_+ is treated explicitly with respect to time. This leads to an unconditionally energy stable time marching scheme, compare [33]. The system (29) is discretized in space using piecewise linear and continuous finite elements. The resulting nonlinear equation systems are solved using a semi-smooth Newton method.

Figure 5 shows the phase field (left) and the chemical potential (right) for the finite element simulation using adaptive meshes. The initial condition $c_\circ$ is transported horizontally with constant velocity.
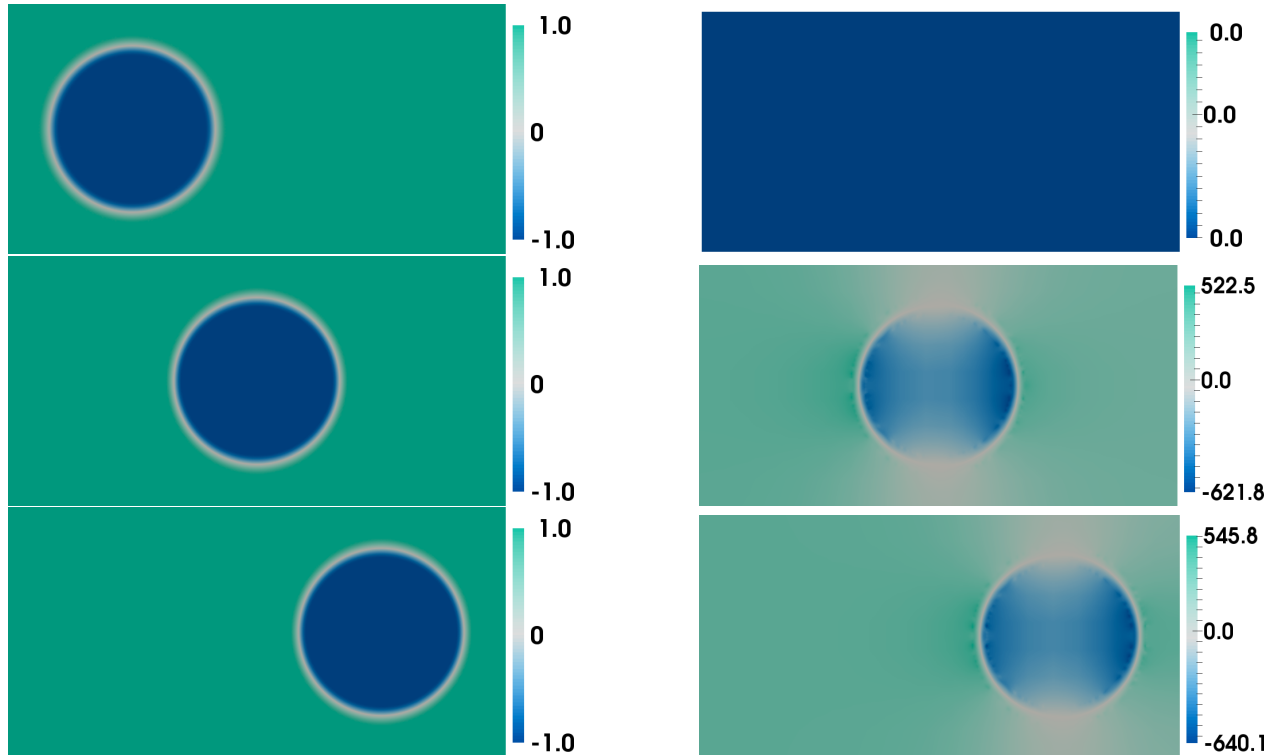
FIGURE 5. *Run 2. Phase field c (left) and chemical potential w (right) computed on adaptive finite element meshes at $t = t_1$ (top), $t = T/2$ (middle) and $t = T$ (bottom).*

The adaptive finite element meshes as well as the finest mesh which is generated during the adaptive finite element simulation are shown in Figure 6. 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.
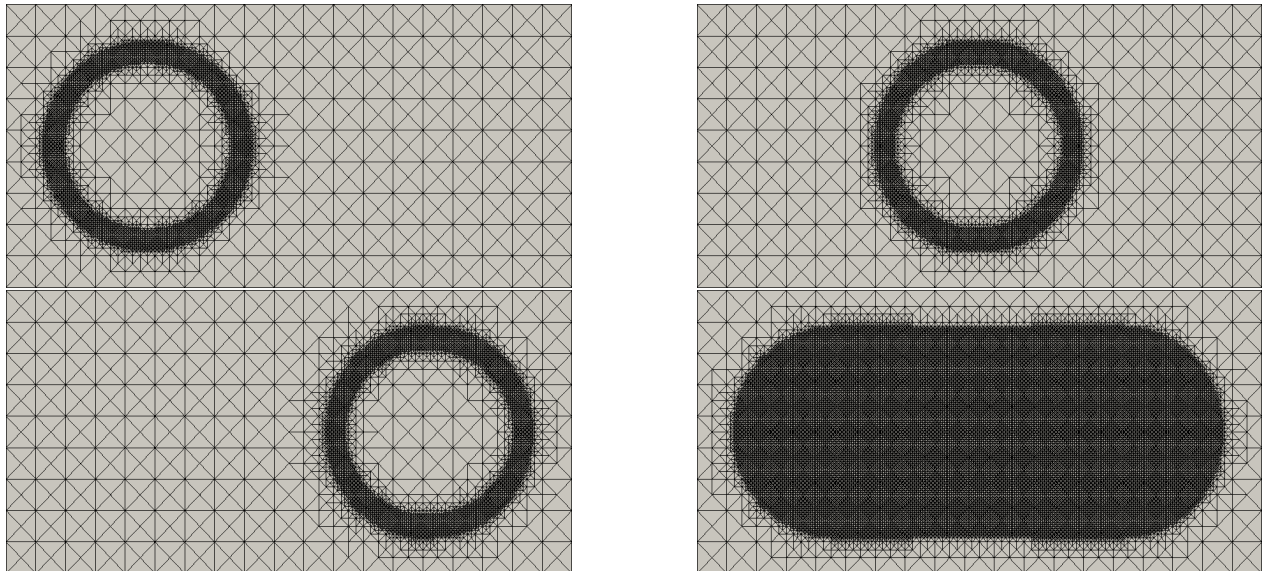

FIGURE 6. *Run 2. Adaptive finite element meshes at $t = t_1$ (top left), $t = T/2$ (top right) and $t = T$ (bottom left) together with the finest mesh (bottom right).*

Figure 7 shows the first, second and fifth POD mode for the phase field c and the chemical potential w. Analogously to Run 1, we observe a periodicity in the POD basis functions corresponding to their basis index numbers.
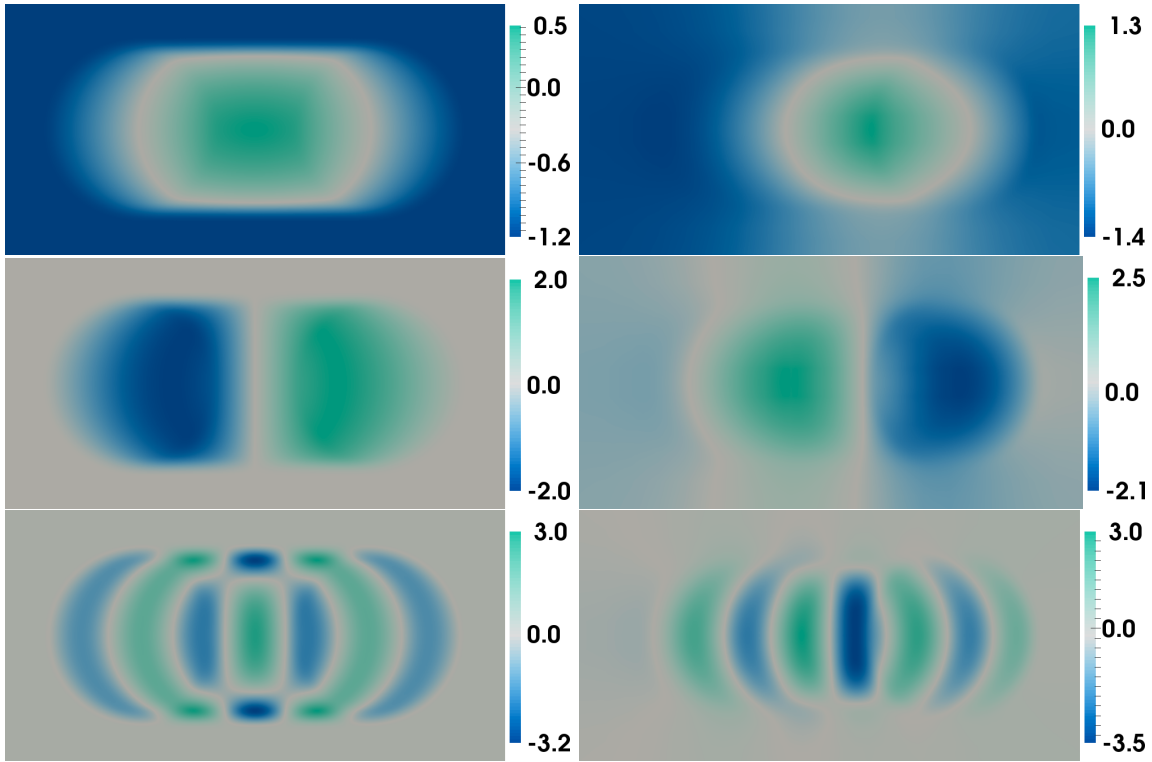

FIGURE 7. *Run 2. First, second and fifth POD modes for c (left) and w (right).*

In the present example we only compare the POD procedure for two kinds of snapshot discretizations, namely the adaptive approach with using a finest mesh, and the uniform mesh approach, where the gridsize is chosen to be of the same size as the smallest triangle in the adaptive meshes. We choose $X = L^2(\Omega)$ and compute a separate POD basis for each of the variables c and w.

In Figure 8, a comparison is visualized concerning the normalized eigenspectrum for the phase field c and the chemical potential w using 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.
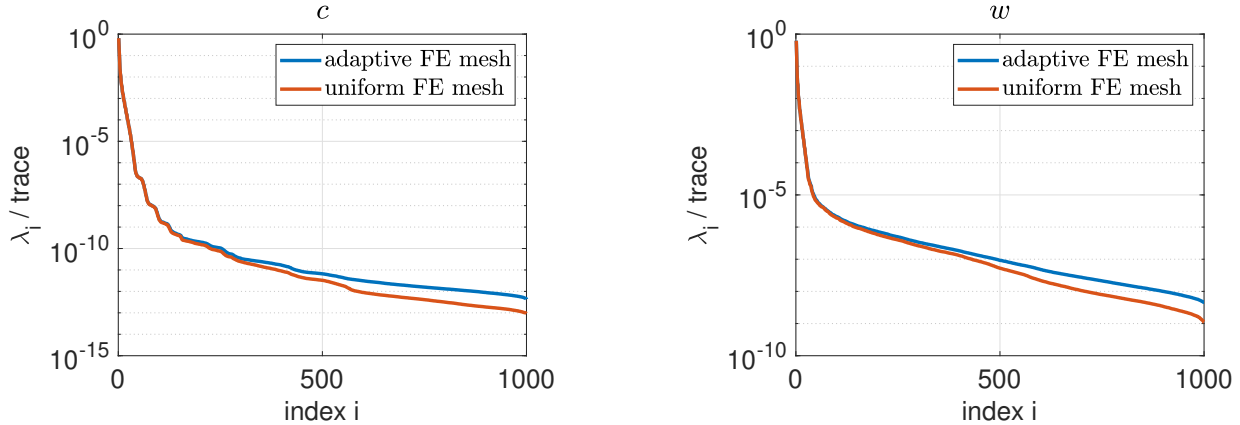
FIGURE 8. *Run 2. Comparison of the normalized eigenvalues for the phase field c (left) and the chemical potential w (right) using an adaptive and a uniform spatial mesh, respectively.*

We use the criterion (11) to determine the basis length ℓ which is required to represent a prescribed information content with the respective POD space. We will choose the POD basis length ℓ_c for the phase field c and the number of POD modes ℓ_w for the chemical potential, such that

$$\ell_{\min} = \arg\min\left\{\mathcal{E}(\ell) : \mathcal{E}(\ell) > 1 - p\right\}, \quad \text{with } \ell = \ell_c \text{ and } \ell_w, \text{ respectively,}$$

for a given value p representing the loss of information. Alternatively, the POD basis length could be chosen in alignment with the POD projection error (10) with the expected spatial and/or temporal discretization error, compare e.g. [39, Theorem 5.1]. Let us also refer, e.g., to the recent paper [12], where different adaptive POD basis extension techniques are discussed. Table 1 summarizes how to choose ℓ_c and ℓ_w in order to capture a desired amount of information. Moreover, it tabulates the POD projection error (10) depending on the POD basis length, where λ_i^c and λ_i^w denote the eigenvalues for the phase field c and the chemical potential w, respectively. The results in Table 1 agree with our expectations: the smaller the loss of information p is, the more POD modes are needed and the smaller is the POD projection error. $\diamond$

p	ℓ_c^{ad}	$\sum_{i>\ell}\lambda_i^c$	ℓ_w^{ad}	$\sum_{i>\ell}\lambda_i^w$	ℓ_c^{uni}	$\sum_{i>\ell}\lambda_i^c$	ℓ_w^{uni}	$\sum_{i>\ell}\lambda_i^w$
10^{-1}	3	$2.0 \cdot 10^{-3}$	4	$156.9 \cdot 10^0$	3	$2.0 \cdot 10^{-3}$	4	$157.6 \cdot 10^0$
10^{-2}	10	$2.1 \cdot 10^{-4}$	13	$15.8 \cdot 10^0$	10	$2.1 \cdot 10^{-4}$	13	$15.6 \cdot 10^0$
10^{-3}	19	$2.5 \cdot 10^{-5}$	26	$1.8 \cdot 10^0$	19	$2.5 \cdot 10^{-5}$	25	$1.8 \cdot 10^0$
10^{-4}	29	$2.0 \cdot 10^{-6}$	211	$1.8 \cdot 10^{-1}$	28	$2.6 \cdot 10^{-6}$	160	$1.9 \cdot 10^{-1}$
10^{-5}	37	$2.5 \cdot 10^{-7}$	644	$1.1 \cdot 10^{-2}$	37	$2.4 \cdot 10^{-7}$	419	$2.5 \cdot 10^{-2}$

TABLE 1. *Run 2. Number of needed POD bases in order to achieve a loss of information below the tolerance p using adaptive finite element meshes (columns 2-5) and uniform finite element discretization (columns 6-9) and POD projection error.*

Run 3 ([39, Example 6.3]). *(Linear heat equation revisited).* We again consider Example 1 with $c \equiv 0$. The purpose of this example is to confirm that our POD approach also is applicable in the case of non-nested meshes like it appears in the case of r-adaptivity, for example. We set up the matrix K for snapshots generated on sequences of non-nested spatial discretizations. This requires the integration over cut elements, see [39]. We choose $\Omega = (0,1) \times (0,1) \subset \mathbb{R}^2$, $[0,T] = [0,1]$, and we apply a uniform temporal discretization with time step size $\Delta t = 0.01$. The analytical solution in

the present example is given by

$$y(t, \boldsymbol{x}) = \sin(\pi x_0) \cdot \sin(\pi x_1) \cdot \cos(2\pi t x_0),$$

with $\boldsymbol{x} = (x_0, x_1)$, source term $f := y_t - \Delta y$ and the initial condition $g := y(0, \cdot)$. The initial condition is discretized using piecewise linear and continuous finite elements on a uniform spatial mesh which is shown in Figure 9 (left). Then, at each time step, the mesh is disturbed by relocating each mesh node according to the assignment

$$
\begin{aligned}
x_0 &\leftarrow x_0 + \theta \cdot x_0 \cdot (x_0 - 1) \cdot (\Delta t/10), \\
x_1 &\leftarrow x_1 + \theta \cdot 0.5 \cdot x_1 \cdot (x_1 - 1) \cdot (\Delta t/10),
\end{aligned}
$$

where $\theta \in \mathbb{R}_+$ is sufficiently small such that all coordinates of the interior nodes fulfill $0 < x_0 < 1$ and $0 < x_1 < 1$. After relocating the mesh nodes, the heat equation is solved on this mesh for the next time instance. We use Lagrange interpolation to transfer the finite element solution of the previous time step onto the new mesh. The disturbed meshes at $t = 0.5$ and $t = 1.0$ as well as an overlap of two meshes are shown in Figure 9. To compute the matrix K from (30) we have to evaluate the corresponding inner products of the snapshots, where we need to integrate over cut elements.
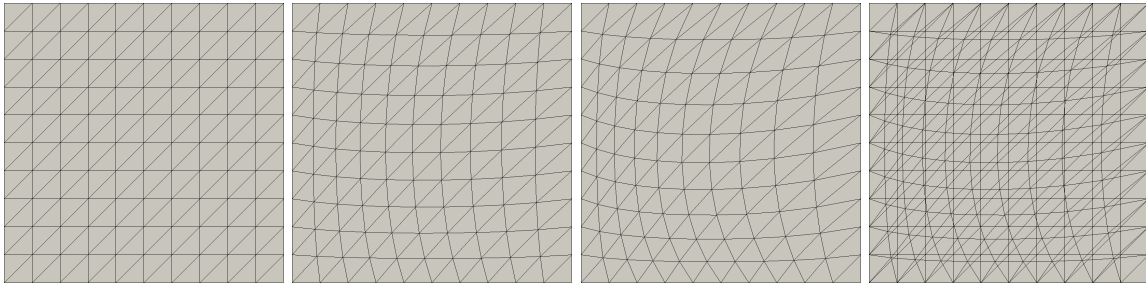

FIGURE 9. *Run* 3. *Uniform mesh (left), disturbed meshes at $t = 0.5$ and $t = 1.0$ (middle left, middle right), overlap of the mesh at $t = 0$ with the mesh at $t = 1.0$ (right). Here, we use $\theta = 10$.*

We compute the eigenvalue decomposition of the matrix representation K of the operator $\mathcal{K}$ (cf. (30)) for different values of θ and compare the results with a uniform mesh (i.e. $\theta = 0$) in Figure 10. We note that the eigenvalues of the disturbed mesh are converging to the eigenvalues of the uniform mesh for $\theta \to 0$. As expected, the eigenvalue spectrum depends only weakly on the underlying mesh given that the mesh size is sufficiently small. Concerning the computational complexity of POD
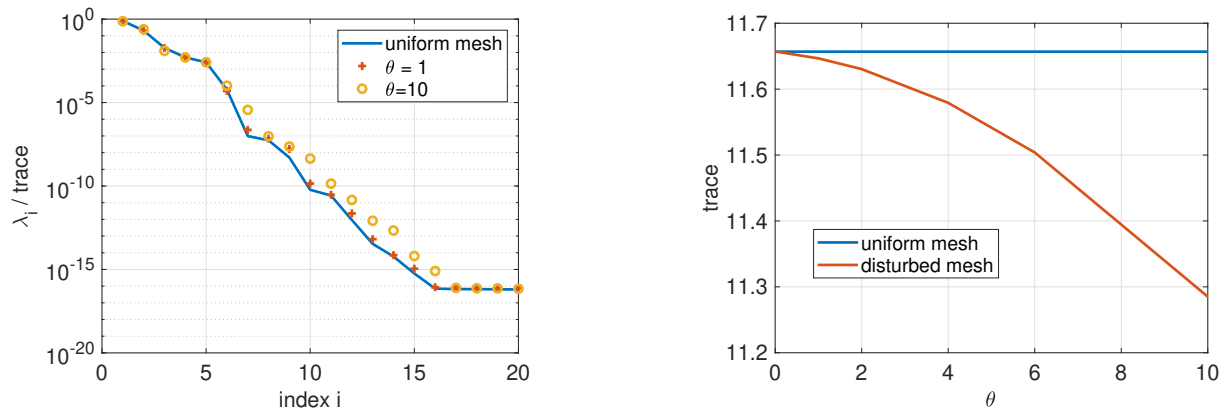

FIGURE 10. *Run* 3: *Decay of eigenvalues of matrix K with different meshes.*

with non-nested meshes let us note that solving the heat equation takes 2.1 seconds on the disturbed meshes and 1.8 seconds on the uniform mesh. The computational time needed to compute each entry of the matrix K is 0.022 seconds and computing the eigenvalue decomposition for K takes

0.0056 seconds. Note that the cut element integration problem for each matrix entry takes a fraction of time required to solve the finite element problem. ◇

3. The POD Galerkin procedure

Once the POD basis is generated it can be used to set up a POD-Galerkin approximation of the original dynamical system. This is discussed in the present section. In this context we recall that the space spanned by the POD basis is used with a Galerkin method to approximate the original system for e.g. other inputs and/or parameters than those used to generate the snapshots for constructing the POD basis. A typical application is given by PDE-constrained optimization, where the PDE system during the optimization is substituted by POD Galerkin surrogates, see Section 6 for more details.

3.1. **The POD Galerkin procedure.** Suppose that for given snapshots $y_j^h \in V^{h_j} \subset X$, $1 \leq j \leq n$, we have computed the symmetric matrix

$$\mathrm{K} = \left(\left(\sqrt{\alpha_i}\sqrt{\alpha_j}\,\langle y_i^h, y_j^h\rangle_X\right)\right)_{1 \leq i,j \leq n} \qquad \text{with rank}\,\mathrm{K} = \mathrm{r} \leq n \tag{30}$$

associated to the operator $\mathcal{K}$ from (8) together with its eigensystem. Its $\ell \in \{1, \dots, \mathrm{r}\}$ largest eigenvalues are $\{\lambda_i\}_{i=1}^{\ell}$ with corresponding eigenvectors $\{\Phi_i\}_{i=1}^{\ell} \subset \mathbb{R}^n$. The POD basis $\{\Psi_i\}_{i=1}^{\ell}$ is then given by (9), i.e.,

$$\Psi_i = \frac{1}{\sqrt{\lambda_i}}\,\mathcal{Y}\Phi_i \quad \text{for } i = 1, \dots, \ell.$$

This POD basis is utilized in order to compute a reduced-order model for (16) along the lines of Section 2.3.3, where the space V^h is replaced by the space $V^\ell = \mathrm{span}\,\{\Psi_1, \dots, \Psi_\ell\} \subset V$. More precisely we make the POD Galerkin ansatz

$$y^\ell(t) = \sum_{i=1}^{\ell} \eta_i(t)\Psi_i = \sum_{i=1}^{\ell} \eta_i(t)\frac{1}{\sqrt{\lambda_i}}\,\mathcal{Y}\Phi_i \quad \text{for all } t \in [0, T], \tag{31}$$

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

$$\eta_i(t) = \langle y^\ell(t), \Psi_i\rangle_X = \left\langle y^\ell(t), \frac{1}{\sqrt{\lambda_i}}\,\mathcal{Y}\Phi_i\right\rangle_X \quad \text{for } 1 \leq i \leq \ell.$$

Inserting y^ℓ into (16) and choosing $V^\ell \subset V$ as the test space leads to the system

$$\left.\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\,\langle y^\ell(t), \Psi\rangle_H + a(y^\ell(t), \Psi) + \langle\mathcal{N}(y^\ell(t)), \Psi\rangle_{V',V} &= \langle f(t), \Psi\rangle_{V',V} \\ \langle y^\ell(0), \Psi\rangle_H &= \langle y_\circ, \Psi\rangle_H\end{aligned}\right\} \tag{32}$$

for all $\Psi \in V^\ell$ and for almost all $t \in (0, T)$. The system (32) is called POD reduced-order model (POD-ROM). Using the ansatz (31), we can write (32) as an ℓ-dimensional ordinary differential equation system for the POD mode coefficients $\eta(t) = (\eta_i(t))_{1 \leq i \leq \ell}$, $t \in (0, T]$, as follows:

$$\left.\begin{aligned}\sum_{j=1}^{\ell} \dot\eta_j(t)\,\langle\Psi_i, \Psi_j\rangle_H + \sum_{j=1}^{\ell} \eta_j(t)\,a(\Psi_j, \Psi_i) &= \langle f(t) - \mathcal{N}(y^\ell(t)), \Psi_i\rangle_{V',V} \\ \sum_{j=1}^{\ell} \eta_j(0)\,\langle\Psi_i, \Psi_j\rangle_H &= \langle y_\circ, \Psi_i\rangle_H\end{aligned}\right\} \tag{33}$$

for $i = 1, \dots, \ell$. Note that $\langle \Psi_i, \Psi_j \rangle_H = \delta_{ij}$ if we choose $X = H$ in the context of Section 2.3. In a next step we rewrite this system using relation between Ψ_i and Φ_i given in (9). This leads to

$$\left. \begin{aligned} \sum_{j=1}^{\ell} \dot{\eta}_j(t) \frac{\langle \mathcal{Y}\Phi_i, \mathcal{Y}\Phi_j \rangle_H}{\sqrt{\lambda_i \lambda_j}} + \sum_{j=1}^{\ell} \eta_j(t) \frac{a(\mathcal{Y}\Phi_j, \mathcal{Y}\Phi_i)}{\sqrt{\lambda_i \lambda_j}} &= \frac{\langle f(t) - \mathcal{N}(y^\ell(t)), \mathcal{Y}\Phi_i \rangle_{V', V}}{\sqrt{\lambda_i}} \\ &\qquad \text{for } t \in (0, T], \\ \sum_{j=1}^{\ell} \eta_j(0) \frac{\langle \mathcal{Y}\Phi_i, \mathcal{Y}\Phi_j \rangle_H}{\sqrt{\lambda_i \lambda_j}} &= \frac{\langle y_\circ, \mathcal{Y}\Phi_i \rangle_H}{\sqrt{\lambda_i}} \end{aligned} \right\} \tag{34}$$

for $i = 1, \dots, \ell$. In order to write (34) in a compact matrix-vector form, let us introduce the diagonal matrix

$$\Lambda := \mathrm{diag}\left(\frac{1}{\sqrt{\lambda_1}}, \dots, \frac{1}{\sqrt{\lambda_\ell}} \right) \in \mathbb{R}^{\ell \times \ell}.$$

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

$$\Phi = \begin{bmatrix} \Phi_1 \,|\, \dots \,|\, \Phi_\ell \end{bmatrix} \in \mathbb{R}^{n \times \ell}.$$

Then, the system (34) can be written as the system

$$\left. \begin{aligned} \left(\Lambda \Phi^\top K \Phi \Lambda \right) \dot{\eta}(t) + \left(\Lambda A^\ell \Lambda \right) \eta(t) + \Lambda N^\ell(\eta(t)) &= \Lambda F^\ell(t) \quad \text{for } t \in (0, T], \\ \Lambda \Phi^\top K \Phi \Lambda \eta(0) &= \Lambda \eta_\circ \end{aligned} \right\} \tag{35}$$

for the vector-valued mapping $\eta = (\eta_1, \dots, \eta_\ell)^\top : [0, T] \to \mathbb{R}^\ell$, for the nonlinearity $N^\ell = (N_i^\ell(\cdot))_{1 \le i \le \ell} : \mathbb{R}^\ell \to \mathbb{R}^\ell$ with

$$N_i^\ell(v) = \left\langle \mathcal{N}\left(\sum_{j=1}^{\ell} v_j \Psi_j \right), \varphi_i \right\rangle_{V', V} = \left\langle \mathcal{N}\left(\sum_{j=1}^{\ell} v_j \mathcal{Y}\Phi_j / \sqrt{\lambda_j} \right), \varphi_i \right\rangle_{V', V}$$

and for the stiffness matrix $A^\ell = ((A_{ij}^\ell)) \in \mathbb{R}^{\ell \times \ell}$ given as

$$A_{ij}^\ell = a(\mathcal{Y}\Phi_j, \mathcal{Y}\Phi_i) \quad \text{for } 1 \le i, j \le \ell.$$

Note that the right hand side $F^\ell(t) = (F_i^\ell(t))_{1 \le i \le \ell}$ and the initial condition $\eta_\circ = (\eta_{\circ i})_{1 \le i \le \ell}$ are given by

$$F_i^\ell(t) = \langle f(t), \mathcal{Y}\Phi_i \rangle_{V', V} = \langle \mathcal{Y}^* f(t), \Phi_i \rangle_{\mathbb{R}^n}, \quad t \in [0, T] \text{ a.e.,}$$

and

$$\eta_{\circ i} = \langle y_\circ, \mathcal{Y}\Phi_i \rangle_H = \langle \mathcal{Y}^* y_\circ, \Phi_i \rangle_{\mathbb{R}^n},$$

for $i = 1, \dots, \ell$, 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 $w = y_\circ$) with finite element discretization $w = \sum_{i=1}^{m_w} w_i \chi_i$, nodal basis $\{\chi_i\}_{i=1}^{m_w} \subset V$ and appropriate mode coefficients $\{w_i\}_{i=1}^{m_w}$ we can compute

$$\left(\mathcal{Y}^* w \right)_j = \langle w, y_j \rangle_X = \left\langle \sum_{i=1}^{m_w} w_i \chi_i, \sum_{k=1}^{m_j} y_k^j \varphi_k^j \right\rangle_X = \sum_{i=1}^{m_w} \sum_{k=1}^{m_j} w_i y_k^j \langle \chi_i, \varphi_k^j \rangle_X$$

for $j = 1, \dots, n$ where $y_j^h = \sum_{k=1}^{m_j} y_k^j \varphi_k^j \in V^{h_j}$ denotes the j-th snapshot. Again, for any $i = 1, \dots, m_w$ and $k = 1, \dots, m_j$, the computation of the inner product $\langle \chi_i, \varphi_k^j \rangle_X$ can be done explicitly. Obviously, for linear evolution equations the POD reduced-order model (35) can be set up and solved using snapshots with arbitrary finite element discretizations. The computation of the nonlinear component $N^\ell(\eta(t))$ needs particular attention. In Section 3.3 we discuss the options to treat the nonlinearity.

3.2. **Time-discrete reduced-order model.** In order to solve the reduced-order system (32) numerically, we apply the implicit Euler method for time discretization and use for simplicity the same temporal grid $\{t_j\}_{j=1}^n$ as for the snapshots. It is also possible to use a different time grid, cf. [58]. The time-discrete reduced-order model reads

$$
\left.\begin{array}{c}
\dfrac{\langle y_j^\ell - y_{j-1}^\ell, \Psi\rangle_H}{\Delta t_j} + a(y_j^\ell, \Psi) + \langle \mathcal{N}(y_j^\ell), \Psi\rangle_{V',V} = \displaystyle\int_{t_{j-1}}^{t_j} \dfrac{\langle f(\tau), \Psi\rangle_{V',V}}{\Delta t_j}\, \mathrm{d}\tau, \\[12pt]
\langle y_1^\ell, \Psi\rangle_H = \langle y_\circ, \Psi\rangle_H
\end{array}\right\}
\tag{36}
$$

for all $\Psi \in V^\ell$ and $j = 2,\ldots,n$. Equivalently the following system holds for the coefficient vector $\eta(t) \in \mathbb{R}^\ell$ (cf. (35)):

$$
\left.\begin{array}{c}
\left(\Lambda\Phi^\top K\Phi\Lambda\right)\left(\dfrac{\eta^j - \eta^{j-1}}{\Delta t_j}\right) + \left(\Lambda A^\ell \Lambda\right)\eta^j + \Lambda N^\ell(\eta^j) = \Lambda F_j^\ell,\ j = 2,\ldots,n \\[12pt]
\Lambda\Phi^\top K\Phi\Lambda\eta^1 = \Lambda\eta_\circ
\end{array}\right\}
\tag{37}
$$

with the inhomogeneity $F_j^\ell = (F_{ji}^\ell)_{1\le i\le \ell}$, $j = 2,\ldots,n$, given as

$$
F_{ji}^\ell = \int_{t_{j-1}}^{t_j} \frac{\langle f(\tau), \mathcal{Y}\Phi_i\rangle_{V',V}}{\Delta t_j}\, \mathrm{d}\tau = \int_{t_{j-1}}^{t_j} \frac{\langle \mathcal{Y}^* f(\tau), \Phi_i\rangle_{\mathbb{R}^n}}{\Delta t_j}\, \mathrm{d}\tau.
$$

3.3. **Discussion of the computation of the nonlinear term.** Let us now consider the computation of the nonlinear term $\Lambda N^\ell(\eta^j) \in \mathbb{R}^\ell$ of the POD-ROM (35). It holds true

$$
\left(\Lambda N^\ell(\eta^j)\right)_k = \langle \mathcal{N}(y^\ell(t)), \Psi_k\rangle_{V',V} = \left\langle \mathcal{N}\left(\sum_{i=1}^\ell \eta_i(t)\Psi_i\right), \Psi_k\right\rangle_{V',V}
$$

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

$$
\Psi^\top W N^h(\Psi\eta(t)) \in \mathbb{R}^\ell, \quad t \in [0,T] \text{ a.e.,}
$$

where $\Psi = [\Psi_1 | \ldots | \Psi_\ell] \in \mathbb{R}^{m\times\ell}$ is the matrix in which the POD modes are stored columnwise and $W \in \mathbb{R}^{m\times m}$ is a weighting matrix related to the utilized inner product (cf. (3)). Hence, the treatment of the nonlinearity requires the expansion of $\Psi\eta(t) \in \mathbb{R}^m$ in the full space for $t \in [0,T]$ a.e. 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 m 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. Common choices are empirical interpolation methods like, e.g., EIM ([14]), DEIM ([24]), and QDEIM ([31]). Another option is dynamic mode decomposition for nonlinear model order reduction, see e.g. [7]. Furthermore, in [99] nonlinear model reduction is realized by replacing the nonlinear term by its interpolation in the finite element space. An alternative approach for the treatment of the nonlinearity is missing point estimation [10], or best points interpolation [70].

Most of these methods need a common reference mesh for the computations. To overcome this restriction we propose different paths which allow for more general discrete settings like r-adaptivity discussed in Run 3.

One option is to use EIM [14]. Alternatively, we can linearize and project the nonlinearity onto the POD space. For this approach, let us consider the linear reduced-order system for a fixed given state

$\bar{y}$, which takes the form

$$\left.\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \langle y^\ell(t), \Psi\rangle_H + a(y^\ell(t), \Psi) + \langle \mathcal{N}(\bar{y}(t)), \Psi\rangle_{V',V} &= \langle f(t), \Psi\rangle_{V',V}, \\
\langle y^\ell(0), \Psi\rangle_H &= \langle y_\circ, \Psi\rangle_H
\end{aligned}\right\} \tag{38}$$

for all $\Psi \in V^\ell$ and for almost all $t \in (0, T]$. The linear evolution problem (38) 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., $\bar{y}(t_j) = y_j$ for $j = 2, \ldots, n$. Furthermore, the linearization of the reduced-order model (32) can be considered:

$$\left.\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \langle y^\ell(t), \Psi\rangle_H + a(y^\ell(t), \Psi) + \langle \mathcal{N}'(\bar{y}(t)) y^\ell(t), \Psi\rangle_{V',V}, \\
= \langle f(t) - \mathcal{N}(\bar{y}(t)) + \mathcal{N}'(\bar{y}(t)))\bar{y}(t), \Psi\rangle_{V',V} \\
\langle y^\ell(0), \Psi\rangle_H = \langle y_\circ, \Psi\rangle_H,
\end{aligned}\right\} \tag{39}$$

for all $\Psi \in V^\ell$ and for almost all $t \in (0, T]$, where $\mathcal{N}'$ denotes the Fréchet derivative of the nonlinear operator $\mathcal{N}$. This linearized problem is of interest e.g. in the context of optimal control, where it occurs in each iteration level within sequential quadratic programming (SQP) methods; see [49], for example. Choosing the finite element solution as given state in each time instance and using (9) leads to

$$\langle \mathcal{N}(y_j), \Psi_i\rangle_{V',V} = \frac{1}{\sqrt{\lambda_i}} \sum_{k=1}^n \sqrt{\alpha_k} (\Phi_i)_k \langle \mathcal{N}(y_j), y_k\rangle_{V',V},$$

$$\langle \mathcal{N}'(y_j) y^\ell(t_j), \Psi_i\rangle_{V',V} = \left\langle \mathcal{N}'(y_j)\Big(\sum_{k=1}^\ell \eta_k(t_j)\Psi_k\Big), \Psi_i \right\rangle_{V',V},$$

$$= \sum_{k=1}^\ell \eta_k(t_j) \frac{1}{\sqrt{\lambda_k \lambda_i}} \sum_{\nu=1}^n \sum_{\mu=1}^n \sqrt{\alpha_\nu \alpha_\mu} (\Phi_k)_\nu (\Phi_i)_\mu \langle \mathcal{N}'(y_j) y_\nu, y_\mu\rangle_{V',V},$$

$$\langle \mathcal{N}'(y_j) y_j, \Psi_i\rangle_{V',V} = \frac{1}{\sqrt{\lambda_i}} \sum_{k=0}^n \sqrt{\alpha_k} (\Phi_i)_k \langle \mathcal{N}'(y_j) y_j, y_k\rangle_{V',V}$$

for $j = 2, \ldots, n$ and $i = 1, \ldots, \ell$. Finally, we approximate the nonlinearity $\Lambda \mathrm{N}^\ell(\eta^j) \in \mathbb{R}^\ell$ in (37) by

$$\left(\Lambda \mathrm{N}^\ell(\eta^j)\right)_i \approx \langle \mathcal{N}(y_j) + \mathcal{N}'(y_j)(y^\ell(t_j) - y_j), \Psi_i\rangle_{V',V}$$

for $j = 2, \ldots, n$ and $i = 1, \ldots, \ell$, which can be written as

$$\Lambda \mathrm{N}^\ell(\eta^j) \approx \Lambda \Phi^\top \mathrm{N}^j + \Lambda \Phi^\top \mathrm{N}_y^j \Phi \Lambda \eta^j - \Lambda \Phi^\top \mathrm{N}_y^j,$$

where

$$\mathrm{N}^j = \begin{pmatrix} \langle \mathcal{N}(y_j), \sqrt{\alpha_1} y_1\rangle_{V',V} \\ \vdots \\ \langle \mathcal{N}(y_j), \sqrt{\alpha_n} y_n\rangle_{V',V} \end{pmatrix} \in \mathbb{R}^n, \quad \mathrm{N}_y^j = \begin{pmatrix} \langle \mathcal{N}'(y_j) y_j, \sqrt{\alpha_1} y_1\rangle_{V',V} \\ \vdots \\ \langle \mathcal{N}'(y_j) y_j, \sqrt{\alpha_n} y_n\rangle_{V',V} \end{pmatrix} \in \mathbb{R}^n,$$

and with $\tilde{y}_j = \sqrt{\alpha_j} y_j$, $j = 1, \ldots, n$,

$$\mathbb{N}_y^j = \begin{pmatrix} \langle \mathcal{N}'(y_j)\tilde{y}_1, \tilde{y}_1\rangle_{V',V} & \cdots & \langle \mathcal{N}'(y_j)\tilde{y}_n, \tilde{y}_1\rangle_{V',V} \\ \vdots & & \vdots \\ \langle \mathcal{N}'(y_j)\tilde{y}_1, \tilde{y}_1\rangle_{V',V} & \cdots & \langle \mathcal{N}'(y_j)\tilde{y}_n, \tilde{y}_n\rangle_{V',V} \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

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, see Table 6. 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., [25] and [84], or Taylor expansions, see, e.g., [73, 74] and [35]. Nevertheless, the efficiency of higher order approximations is limited due to growing memory and computational costs.

3.4. **Expressing the POD solution in the full spatial domain.** Having determined the solution $\eta(t)$ to (35), we can set up the reduced solution $y^\ell(t)$ in a continuous framework:

$$y^\ell(t) = \sum_{i=1}^{\ell} \eta_i(t) \left(\frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{n} \sqrt{\alpha_j} (\Phi_i)_j y_j \right). \tag{40}$$

Now, let us turn to the fully discrete formulation of (40). For a time-discrete setting, we introduce for simplicity the same temporal grid $\{t_j\}_{j=1}^n$ as for the snapshots. The snapshots (28) admit the expansion

$$y_j = \sum_{i=1}^{m_j} \mathsf{y}_j^i \varphi_i^j \quad \text{for } j = 1, \dots, n.$$

Let $\{Q_r^j\}_{r=1}^{l_j}$ 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:

$$y^\ell(t_j, Q_r^j) = \sum_{i=1}^{\ell} \eta_i(t_j) \left(\frac{1}{\sqrt{\lambda_i}} \sum_{\nu=1}^{n} \sqrt{\alpha_\nu} (\Phi_i)_\nu \left(\sum_{k=1}^{m_\nu} \mathsf{y}_k^\nu \varphi_k^\nu (Q_r^j) \right) \right) \tag{41}$$

for $r = 1, \dots, l_j$ and $j = 1, \dots, n$. This allows us to use any grid for expressing the POD solution in the full spatial domain. For example, we can use the same nodes at time level j for the POD simulation as we have used for the snapshots, i.e., for $j = 1, \dots, n$ it holds $l_j = m_j$ and $Q_r^j = P_k^j$ for all $r, k = 1, \dots, m_j$. Another option can be to choose

$$\{Q_r^j\}_{r=1}^{l_j} = \bigcup_{j=1}^{n} \bigcup_{k=1}^{m_j} \{P_k^j\} \quad \text{for } j = 1, \dots, n,$$

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 [95]. 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 (41) simplifies to

$$y^\ell(t_j, P_r) = \sum_{i=1}^{\ell} \eta_i(t_j) \left(\frac{1}{\sqrt{\lambda_i}} \sum_{\nu=1}^{n} \sqrt{\alpha_\nu} (\Phi_i)_\nu \mathsf{y}_\nu \right) \quad \text{for } j = 1, \dots, n, \tag{42}$$

where $\{P_r\}_{r=1}^m$ are the nodes of the common finite element space.

Run 4 ([39, Example 6.1]). Let us revisit Run 1 and consider its POD Galerkin solutions. The POD solutions for $\ell = 10$ and $\ell = 50$ POD basis functions using spatial adaptive snapshots which are interpolated onto the finest mesh are shown in Figure 11. As expected, the more POD basis functions we use (until stagnation of the corresponding eigenvalues), the less oscillations appear in the POD solution and the better the approximation is.
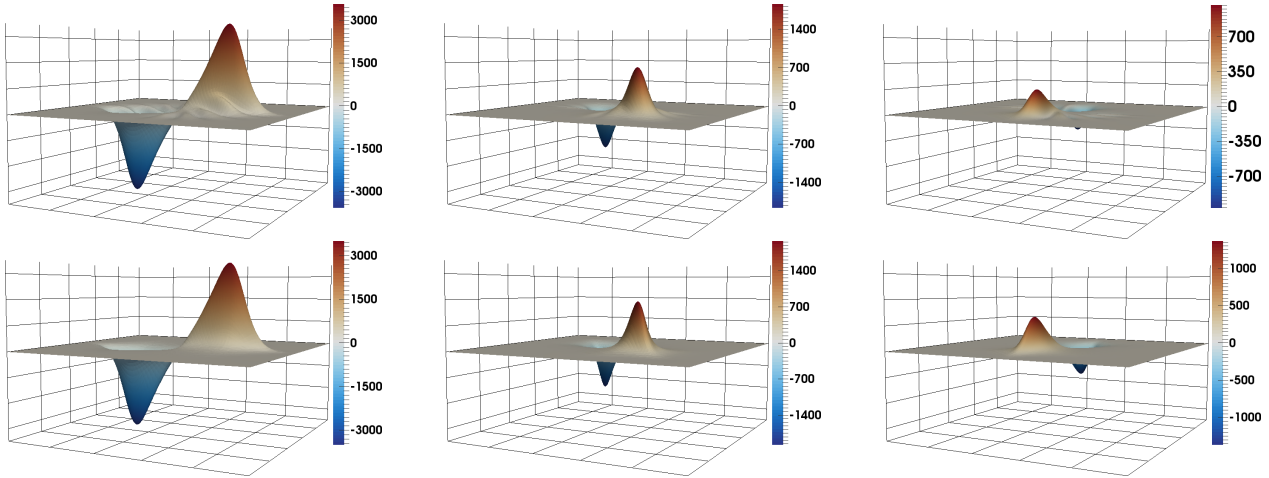
FIGURE 11. *Run 4. Surface plot of the POD solution using $\ell = 10$ (top) and $\ell = 50$ (bottom) POD basis functions at $t = t_1$ (left), $t = T/2$ (middle) and $t = T$ (right).*

Table 2 compares the approximation quality in the relative $L^2(0, T; L^2(\Omega))$-norm of the POD solution using adaptively generated snapshots which are interpolated onto the finest mesh with snapshots of uniform spatial discretization depending on different POD basis lengths. Then, for $\ell = 20$ we obtain a relative $L^2(0, T; L^2(\Omega))$-error between the POD solution and the finite element solution of size $\varepsilon_{\text{FE}}^{\text{ad}} = 3.08 \cdot 10^{-2}$, and a relative $L^2(0, T; L^2(\Omega))$-error between the POD solution and the true solution of size $\varepsilon_{\text{true}}^{\text{ad}} = 2.17 \cdot 10^{-2}$.

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 2. *Run 4. Relative $L^2(0, T; L^2(\Omega))$-error between the POD solution and the finite element solution (columns 2-3) and the true solution (columns 4-5), respectively, using adaptive finite element snapshots which are interpolated onto the finest mesh and using a uniform mesh.*

We note that $\varepsilon_{\text{FE}}^{\text{uni}}$ decays down to 10^{-8} ($\ell = 100$) and then stagnates if using a uniform mesh. This behavior is clear, since the more POD basis elements we include (up to stagnation of the corresponding eigenvalues), the better is the POD solution an approximation for the finite element solution. On the other hand, both $\varepsilon_{\text{true}}^{\text{uni}}$ and $\varepsilon_{\text{true}}^{\text{ad}}$ start to stagnate after $\ell = 30$ in Table 2, columns 4 and 5. This is due to the fact that at this point the spatial (and temporal) discretization error dominates the modal error. This is in accordance with the decay of the eigenvalues shown in Figure 3 and is accounted for e.g. in the error estimation presented in [39, Theorem 5.1]. Similar observations hold true for the relative $L^2(0, T; H^1(\Omega))$-error listed in Table 3 with the difference that the $L^2(0, T; H^1(\Omega))$-error is larger than the respective $L^2(0, T; L^2(\Omega))$-error.

ℓ	$\varepsilon_{\mathrm{FE}}^{\mathrm{ad}}$	$\varepsilon_{\mathrm{FE}}^{\mathrm{uni}}$	$\varepsilon_{\mathrm{true}}^{\mathrm{ad}}$	$\varepsilon_{\mathrm{true}}^{\mathrm{uni}}$
1	$1.46 \cdot 10^{0}$	$1.46 \cdot 10^{0}$	$1.46 \cdot 10^{0}$	$1.47 \cdot 10^{0}$
3	$1.21 \cdot 10^{0}$	$1.22 \cdot 10^{0}$	$1.22 \cdot 10^{0}$	$1.22 \cdot 10^{0}$
5	$9.39 \cdot 10^{-1}$	$9.45 \cdot 10^{-1}$	$9.47 \cdot 10^{-1}$	$9.51 \cdot 10^{-1}$
10	$4.22 \cdot 10^{-1}$	$4.25 \cdot 10^{-1}$	$4.33 \cdot 10^{-1}$	$4.31 \cdot 10^{-1}$
20	$7.76 \cdot 10^{-2}$	$7.27 \cdot 10^{-2}$	$1.02 \cdot 10^{-1}$	$8.19 \cdot 10^{-2}$
30	$2.92 \cdot 10^{-2}$	$1.22 \cdot 10^{-2}$	$7.26 \cdot 10^{-2}$	$3.52 \cdot 10^{-2}$
50	$2.61 \cdot 10^{-2}$	$4.74 \cdot 10^{-4}$	$7.05 \cdot 10^{-2}$	$3.27 \cdot 10^{-2}$
100	$2.79 \cdot 10^{-2}$	$4.78 \cdot 10^{-7}$	$6.94 \cdot 10^{-2}$	$3.27 \cdot 10^{-2}$
150	$2.93 \cdot 10^{-2}$	$2.84 \cdot 10^{-7}$	$6.87 \cdot 10^{-2}$	$3.27 \cdot 10^{-2}$

TABLE 3. *Run 4. Relative $L^2(0,T;H^1(\Omega))$-error between the POD solution and the finite element solution (columns 2-3) and the true solution (columns 4-5), respectively, using adaptive finite element snapshots which are interpolated onto the finest mesh and using a uniform mesh.*

The computational times for the full and the low order simulation using uniform finite element discretizations and adaptive finite element snapshots, which are interpolated onto the finest mesh, respectively, are listed in Table 4.

	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	0.07 sec		$-$
speedup factor	13485	125828	$-$

TABLE 4. *Run 4. CPU times for FE and POD simulation using uniform finite element meshes and adaptive finite element snapshots which are interpolated onto the finest mesh, respectively, and using $\ell = 50$ POD modes.*

Once the POD basis is computed in the offline phase, the POD simulation corresponding to adaptive snapshots is 13485 times faster than the FE simulation using adaptive finite element meshes. This speedup factor is important when one considers e.g. optimal control problems with time-dependent PDEs, where the POD-ROM can be used as surrogate model in repeated solution of the underlying PDE model. In the POD offline phase, the most expensive task is to express the snapshots with respect to the common finite element space, which takes 226 seconds. Since K (30) is symmetric, it suffices to calculate the entries on and above the diagonal, which are $\sum_{k=1}^{n} k = (n^2 + n)/2$ entries. Thus, the computation of each entry in the correlation matrix K using a common finite element space takes around 0.00018 seconds. We note that in the approach explained in Sections 2.4 and 3, the computation of the matrix K is expensive. For each entry the calculation time is around 0.03 seconds, which leads to a computation time of around 36997 seconds for the matrix K. The same effort is needed to build $A^\ell = a(\mathcal{Y}\Phi_j, \mathcal{Y}\Phi_i)$. In this case, the offline phase takes therefore around 88271 seconds. For this reason, the approach to interpolate the adaptively generated snapshots onto the finest mesh is computationally more favorable. But since the computation of K can be parallelized, the offline computation time can be reduced provided that the appropriate hardware is available. $\diamondsuit$

Run 5 (Cahn-Hilliard equations)**.** Now let us revisit Run 2, where we in the following run the numerical simulations for different combinations of numbers for ℓ_c and ℓ_w of Table 1. The approximation quality of the POD solution using adaptive meshes is compared to the use of a uniform

mesh in Table 5. As expected, Table 5 shows that the error between the POD surrogate solution and the high-fidelity solution gets smaller for an increasing number of utilized POD basis functions. Moreover, a larger number of POD modes is needed for the chemical potential w than for the phase field c in order to get an error in the same order which is in accordance to the fact that the decay of the eigenvalues for w is slower than for c as seen in Figure 8.

ℓ^c	ℓ^w	$c : \varepsilon_{\mathrm{FE}}^{\mathrm{ad}}$	$w : \varepsilon_{\mathrm{FE}}^{\mathrm{ad}}$	$c : \varepsilon_{\mathrm{FE}}^{\mathrm{uni}}$	$w : \varepsilon_{\mathrm{FE}}^{\mathrm{uni}}$
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 5. *Run 5. Relative $L^2(0,T; L^2(\Omega))$-error between the POD solution and the finite element solution using adaptive meshes (columns 3-4) and using a uniform mesh (columns 5-6), respectively.*

We now discuss the treatment of the nonlinearity and also investigate the influence of non-smoothness of the model equations to the POD procedure. Using the convex-concave splitting for W, we obtain for the Moreau-Yosida relaxed double obstacle free energy the concave part $W_-^{\mathrm{rel}}(c) = \frac{1}{2}(1 - c^2)$ and the convex part $W_+^{\mathrm{rel}}(c) = \frac{s}{2}(\max(c - 1, 0)^2 + \min(c + 1, 0)^2)$. 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 with non-smooth first derivative. For a comparison, we consider the smooth polynomial free energy with concave part $W_-^{p}(c) = \frac{1}{4}(1 - 2c^2)$ and convex part $W_+^{p}(c) = \frac{1}{4}c^4$.

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 non-smooth case more POD modes are needed for a good approximation than in the smooth case. This behavior is similar 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.
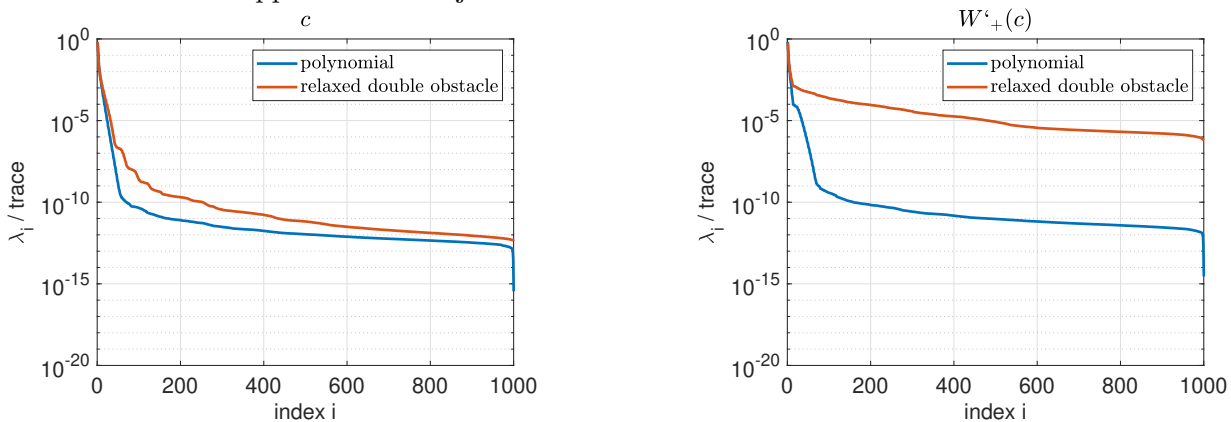

FIGURE 12. *Run 5. Comparison of the normalized eigenvalues for c (left) and the first derivative of the convex part W_+' of the free energy (right) using polynomial and relaxed double obstacle energy, respectively.*

Table 6 summarizes computational times for different finite element runs as well as reduced-order simulations using 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 non-smooth potential. This is due to the fact that in the smooth case, two to three Newton steps are needed for convergence in each time step, whereas in the non-smooth case six to eight iterations are needed in the semismooth Newton method.

	W^p		W_s^{rel}	
FE	1644 s		3129 s	
	$\ell_c = 3$ $\ell_w = 4$	$\ell_c = 19$ $\ell_w = 26$	$\ell_c = 3$ $\ell_w = 4$	$\ell_c = 19$ $\ell_w = 26$
POD offline	355 s	355 s	350 s	349 s
DEIM offline	8 s	8 s	9 s	10 s
ROM	183 s	191 sec	2616 s	3388 s
ROM-DEIM	0.05 s	0.1 s	0.04 s	no conv.
ROM-proj	0.008 s	0.03 s	0.01 s	0.03 s
speedup FE-ROM	8.9	8.6	1.1	none
speedup FE-ROM-DEIM	32880	16440	78225	–
speedup FE-ROM-proj	205500	54800	312900	104300
rel $L^2(Q)$ error ROM	$5.46 \cdot 10^{-3}$	$3.23 \cdot 10^{-4}$	$8.44 \cdot 10^{-3}$	$1.57 \cdot 10^{-3}$
rel $L^2(Q)$ error ROM-DEIM	$1.46 \cdot 10^{-2}$	$3.83 \cdot 10^{-4}$	$8.84 \cdot 10^{-3}$	–
rel $L^2(Q)$ error ROM-proj	$4.70 \cdot 10^{-2}$	$4.18 \cdot 10^{-2}$	$8.72 \cdot 10^{-3}$	$9.80 \cdot 10^{-3}$

TABLE 6. *Run 5. Computational times, speedup factors and approximation quality for different POD basis lengths and using different free energy potentials.*

Using the smooth polynomial free energy, the reduced-order simulation is 8-9 times faster than the finite element simulation, whereas using the relaxed double obstacle free energy only delivers a very small speedup. The inclusion of DEIM (we use $\ell_{\mathrm{deim}} = \ell_c$) 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 $L^2(0, T; L^2(\Omega))$-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 non-smooth free energy function using $\ell_c = 19$ 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 3.1, 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 using projection of the nonlinearity is of the magnitude $10^{-02}/10^{-03}$. Depending on the motivation, this approximation quality might be sufficient. Nevertheless, we note that for large numbers of POD modes, using the projection of the nonlinearity onto the POD space leads to a large increase of the error. ◇

To summarize, a POD reduced-order model construction approach is proposed which can be set up and solved for snapshots originating from arbitrary FE (and also other) spaces. The method is applicable for h-, p- and r-adaptive finite elements. It is motivated from an infinite-dimensional

perspective. Using the method of snapshots we are able to set up the correlation matrix K from (30) by evaluating the inner products of snapshots which live in different FE spaces. For non-nested meshes, this requires the detection of cell collision and integration over cut finite elements. A numerical strategy how to implement this practically is elaborated and numerically tested. Using the eigenvalues and eigenvectors of this correlation matrix, we are able to set up and solve a POD surrogate model that does not need the expression of the snapshots with respect to the basis of a common FE space or the interpolation onto a common reference mesh. Moreover, an error bound for the error between the true solution and the solution to the POD-ROM using spatially adapted snapshots is available in [39, Theorem 5.1]. The numerical tests show that the POD projection error decreases if the number of utilized POD basis functions is increased. However, the error between the POD solution and the true solution stagnates when the spatial discretization error dominates. Moreover, the numerics show that using the correlation matrix calculated explicitly without interpolation in order to build a POD-ROM gives the same results as the approach where the snapshots are interpolated onto the finest mesh. 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. An appropriate treatment of the nonlinearity in our applications gains significant speedup of the ROM in computational times when compared to the full simulations. This makes POD-MOR with adaptive finite elements an ideal approach for the construction of surrogate models in e.g. optimal control with nonlinear PDE systems as they arise e.g. in the context of multi-phase flow control problems.

4. Certification with a priori and a posteriori error estimates

As we have seen in Section 3 POD provides a method for deriving low order models of dynamical systems. It can be thought of as a Galerkin approximation in the spatial variable, built from functions corresponding to the solution of the physical system at prespecified time instances. After carrying out a singular value decomposition the leading ℓ generalized eigenfunctions are chosen as the POD basis $\{\Psi_j\}_{j=1}^{\ell}$ of rank ℓ. As soon as one uses POD, questions concerning the quality of the approximation properties, convergence, and rate of convergence become relevant. Let us refer, e.g., to the literature [22, 42, 56, 58, 57, 85, 88, 89, 80] for a priori error analysis for POD Galerkin approximations. It turns out that the error depends on the decay of the sum $\sum_{i>\ell} \lambda_i$, the error Δt^{β} (with an appropriate $\beta \geq 1$) due to the used time integration method, the used Galerkin spaces $\{V^{h_j}\}_{j=1}^{n}$ and the choice $X = H$ or $X = V$. In particular, best approximation properties hold provided the time differences $\dot{y}^h(t_j)$ (or the finite difference discretizations) are included in the snapshot ensembles; cf. [56, 58, 89].

Let us recall numerical test examples from [42, Section 1.5]. The programs are written in Matlab using the Partial Differential Equation Toolbox for the computation of the piecewise linear FE discretization. For the temporal integration the implicit Euler method is applied based on the equidistant time grid $t_j = (j-1)\Delta t$, $j = 1, \ldots, n$ and $\Delta t = T/(n-1)$.

Run 6 (POD for the heat equation; cf. [42, Run 1]). We choose the final time $T = 3$, the spatial domain $\Omega = (0, 2) \subset \mathbb{R}$, the Hilbert spaces $H = L^2(\Omega)$, $V = H_0^1(\Omega)$, the source term $f(t, \boldsymbol{x}) = t^3 - \boldsymbol{x}^2$ for $(t, \boldsymbol{x}) \in Q = (0, T) \times \Omega$ and the discontinuous initial value $y_\circ(\boldsymbol{x}) = \chi_{(0.5, 1.0)} - \chi_{(1, 1.5)}$ for $\boldsymbol{x} \in \Omega$, where, e.g., $\chi_{(0.5, 1)}$ denotes the characteristic function on the subdomain $(0.5, 1) \subset \Omega$, $\chi_{(0.5, 1)}(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in (0.5, 1)$ and $\chi_{(0.5, 1)}(\boldsymbol{x}) = 0$ otherwise. We consider a discretization of the linear heat equation
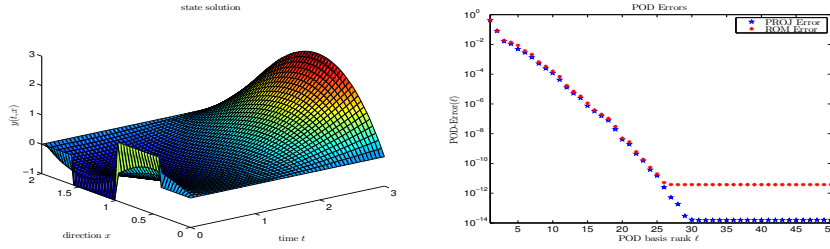
FIGURE 13. Run 6 (cf. [42, Figure 1.1]). The FE solution y^h (left) and the residuals corresponding to the POD basis rank ℓ (right).

(compare (17) with $c \equiv 0$)

$$
\begin{aligned}
y_t(t, \boldsymbol{x}) - \Delta y(t, \boldsymbol{x}) &= f(t, \boldsymbol{x}) && \text{for } (t, \boldsymbol{x}) \in Q, \\
y(t, \boldsymbol{x}) &= 0 && \text{for } (t, \boldsymbol{x}) \in \Sigma = (0, T) \times \partial\Omega, \\
y(0, \boldsymbol{x}) &= y_\circ(\boldsymbol{x}) && \text{for } \boldsymbol{x} \in \Omega.
\end{aligned}
\tag{43}
$$

To obtain an accurate approximation of the exact solution we choose $n = 4000$ so that $\Delta t \approx 7.5 \cdot 10^{-4}$ holds. For the FE discretization we choose $m = 500$ spatial grid points and the equidistant mesh size $h = 2/(m+1) \approx 4 \cdot 10^{-3}$. Thus, the FE error – measured in the H-norm – is of the order 10^{-4}. In the left graphic of Figure 13, the FE solution y^h to the state equation (43) is visualized. To compute a POD basis $\{\Psi_i\}_{i=1}^\ell$ of rank ℓ we utilize the multiple discrete snapshots $y_j^1 = y^h(t_j)$ for $1 \le j \le n_t$ as well $y_1^2 = 0$ and $y_j^2 = (y^h(t_j) - y^h(t_{j-1})/\Delta t$, $j = 2, \ldots, n_t$, i.e., we include the temporal difference quotients in the snapshot ensemble and $K = 2$, $n_1 = n_2 = n_t$. We choose $X = H$ and utilize the (stable) SVD to determine the POD basis of rank ℓ; compare Section 2.2. We address this issue in a more detail in Run 9. Since the snapshots are FE functions, the POD basis elements are also FE functions. In the right plot of Figure 13, the projection and reduced-order error given by

$$
\text{PROJ Error}(\ell) = \left(\sum_{j=1}^{n_t} \alpha_j \left\| y^h(t_j) - \sum_{i=1}^\ell \langle y^h(t_j), \psi_i \rangle_H \, \psi_i \right\|_H^2 \right)^{1/2},
$$

$$
\text{ROM Error}(\ell) = \left(\sum_{j=1}^{n_t} \alpha_j \left\| y^h(t_j) - y^\ell(t_j) \right\|_H^2 \right)^{1/2}
$$

are plotted for different POD basis ranks ℓ. The chosen trapezoidal weights α_j have been introduced in (23). We observe that both errors decay rapidly and coincide until the accuracy 10^{-12}, which is already significant smaller than the FE discretization error. These numerical results reflect the a priori error estimates presented in [42, Theorem 1.29]. $\diamondsuit$

Run 7 (POD for a convection dominated heat equation; cf. [42, Run 2]). Now we consider a more challenging example. We study a convection-reaction-diffusion equation with a source term which is close to being singular: Let T, Ω, $y_\circ$, H and V be given as in Run 6. The parabolic problem reads as follows

$$
\begin{aligned}
y_t(t, \boldsymbol{x}) - c y_{\boldsymbol{x}\boldsymbol{x}}(t, \boldsymbol{x}) + \beta y_x(t, \boldsymbol{x}) + a y(t, \boldsymbol{x}) &= f(t, \boldsymbol{x}) && \text{for } (t, \boldsymbol{x}) \in Q, \\
y(t, \boldsymbol{x}) &= 0 && \text{for } (t, \boldsymbol{x}) \in \Sigma, \\
y(0, \boldsymbol{x}) &= y_\circ(\boldsymbol{x}) && \text{for } \boldsymbol{x} \in \Omega.
\end{aligned}
$$

We choose the diffusivity $c = 0.025$, the velocity $\beta = 1.0$ that determines the speed in which the initial profile $y_\circ$ is shifted to the boundary and the reaction rate $a = -0.001$. Finally, $f(t, \boldsymbol{x}) = \mathbb{P}(\frac{1}{1-t}) \cos(\pi \boldsymbol{x})$ for $(t, \boldsymbol{x}) \in Q$, where $(\mathbb{P}z)(t) = \min(+l, \max(-l, z(t)))$ restricts the image of z on a bounded interval. In this situation, the state solution y develops a jump at $t = 1$ for $l \to \infty$; see the left plot of Figure 14. The right plot of Figure 14 demonstrates that in this case, the decay of the

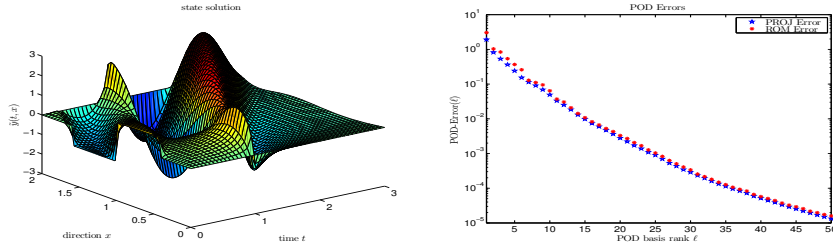FIGURE 14. Run 7 (cf. [42, Figure 1.2]. The FE solution y^h (left) and the residuals corresponding to the POD basis rank ℓ (right).

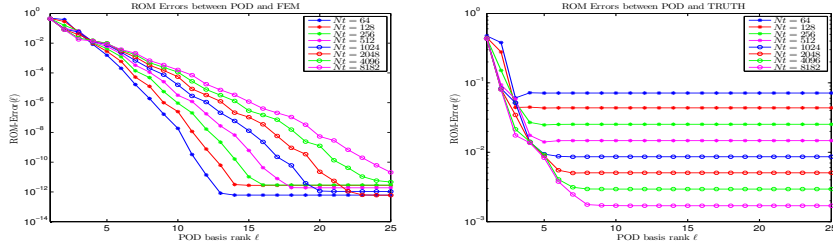
FIGURE 15. Run 8 (cf. [42, Figure 1.3]. The ROM errors with respect to the true solution (left) and the exact one (right).

reconstruction residuals and the decay of the errors are much slower as in the right plot of Figure 13. The manifold dynamics of the state solution require an inconvenient large number of POD basis elements. Since the supports of these ansatz functions in general cover the whole domain Ω, the corresponding system matrices of the reduced model are not sparse. This is different for the matrices arising in the FE Galerkin framework. Model order reduction is not effective for this example if a good accuracy of the solution function y^ℓ is required. Strategies to improve the accuracy and robustness of the POD-ROM in those situations are discussed in e.g. [18, 100]   ◇

Run 8 (True and exact approximation error; cf. [42, Run 3])**.** We consider the setting introduced in Run 6 again. The exact solution to (43) does not possess a representation by elementary functions. Hence, the presented reconstruction and reduction errors actually are the residuals with respect to a high-order FE solution y^h. To compute an approximation y of the exact solution y_{ex} we apply a Crank-Nicolson method (with Rannacher smoothing [77]) ensuring $\|y - y_{\mathsf{ex}}\|_{L^2(0,T;H)} = \mathcal{O}(\Delta t^2 + h^2) \approx 10^{-5}$. In the context of model reduction, such a state is sometimes called the "true" solution. To compute the FE state y^h we apply the Euler method. In the left plot of Figure 15 we compare the true solution y_{ex} with the associated POD approximation for different values $n_t \in \{64, 128, 256, ..., 8192\}$ of the time integration and for the spatial mesh size $h = 4 \cdot 10^{-3}$. For the norm we apply a discrete $L^2(0,T;H)$-norm as in Run 6. Let us mention that we compute for every n_t a corresponding FE solution y^h. We observe that the residuals ignore the errors arising by the application of time and space discretization schemes for the full-order model. The errors decay below the discretization error 10^{-5}. If these discretization errors are taken into account, the residuals stagnate at the level of the full-order model accuracy instead of decaying to zero; cf. right plot of Figure 15. Due to the implicit Euler method we have $\|y^h - y_{\mathsf{ex}}\|_{L^2(0,T;H)} = \mathcal{O}(\Delta t + h^2)$ with the mesh-size $h = 4 \cdot 10^{-3}$. In particular, from $n_t \in \{64, 128, 256, ..., 8192\}$ it follows that $\Delta t > 3 \cdot 10^{-4} > h^2 = 1.6 \cdot 10^{-5}$. Therefore, the spatial error is dominated by the time error for all values of n_t. We can observe that the exact residuals do not decay below a limit of the order Δt. One can observe that for fixed POD basis rank ℓ, the residuals with respect to the true solution increase if the high-order accuracy is improved by enlarging n_t, since the reduced-order model has to approximate a more complex system in this case,
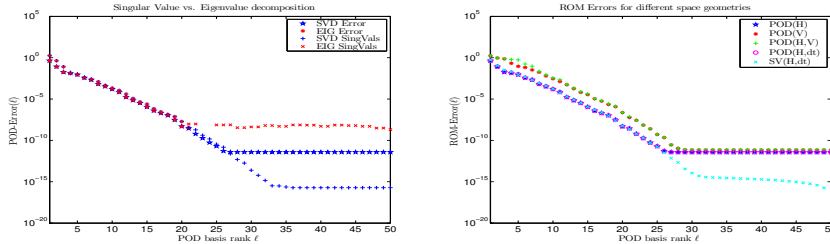
FIGURE 16. Run 9 (cf. [42, Figure 1.4]. Singular values σ_i using the SVD (SVD Vals) or the eigenvalue decomposition (EIG Vals) and the associated ROM errors (SVD error and EIG Error, respectively) (left); ROM errors for different the choices for X, the error norm and the snapshot ensembles (right).

where the residuals with respect to the exact solution decrease due to the lower limit of stagnation $\Delta t = 3/(n_t - 1)$. $\diamond$

Run 9 (Different strategies for a POD basis computation; cf. [42, Run 4]). As we have explained in Section 2.2, let $Y \in \mathbb{R}^{m \times n}$ denote the matrix of snapshots with rank r, $W \in \mathbb{R}^{m \times m}$ be the (sparse) spatial weighting matrix consisting of the elements $\langle \varphi_j, \varphi_i \rangle_X$ (introduced Section 2.3.3) and $D \in \mathbb{R}^{n \times n}$ be the diagonal matrix containing the nonnegative weighting parameters α_j^k. As we have explained in Section 2.2, the POD basis $\{\Psi_i\}_{i=1}^{\ell}$ of rank $\ell \leq$ r can be determined by providing an eigenvalue decomposition of the matrix $\bar{Y}\bar{Y}^\top = W^{1/2}YDY^\top W^{1/2} \in \mathbb{R}^{m \times m}$, one of $\bar{Y}^\top \bar{Y} = D^{1/2}Y^\top WYD^{1/2} \in \mathbb{R}^{n \times n}$, or a singular value decomposition of $\bar{Y} = W^{1/2}YD^{1/2} \in \mathbb{R}^{m \times n}$. Since $n \gg m$ in Runs 6-8, the first variant is the cheapest one from a computational point of view. In case of multiple space dimensions or if a second-order time integration scheme such as some Crank-Nicolson technique is applied, the situation is converse. On the other hand, a singular value decomposition is more accurate and stable than an eigenvalue decomposition if the POD elements corresponding to eigenvalues/singular values which are close to zero are taken into account: Since $\lambda_i = \sigma_i^2$ holds for all eigenvalues λ_i and singular values σ_i, the singular values are able to decay to machine precision, where the eigenvalues stagnate significantly above. This is illustrated in the left graphic of Figure 16. Indeed, for $\ell > 20$ the EIG-ROM system matrices become singular due to the numerical errors in the eigenfunctions and the reduced-order system is ill-posed in this case while the SVD-ROM model remains stable. In the right plot of Figure 16 POD elements are constructed with respect to different scalar products and the resulting ROM errors are compared: $\| \cdot \|_H$-residuals for $X = H$ (denoted by POD(H)), $\| \cdot \|_V$-residuals for $X = V$ (denoted by POD(V)), $\| \cdot \|_V$-residuals for $X = H$ (denoted by POD(H,V)), which also works quite well, the consideration of time derivatives in the snapshot sample (denoted by POD(H,dt)) which allows to apply the a priori error estimate given in [42, Theorem 1.29-2)] and the corresponding sums of singular values (denoted by SV(H,dt)) corresponding to the unused eigenfunctions in the latter case which indeed nearly coincide with the ROM errors. $\diamond$

Notice that in many applications, the quality of the reduced-order model does not vary significantly if the weights matrix W refers to the space $X = H$ or $X = V$ and if time derivatives of the used snapshots are taken into account or not. Especially, the ROM residual decays with the same order as the sum over the remaining singular values, independent of the chosen geometrical framework.

5. Optimal snapshot location for computing POD basis functions

The construction of reduced-order models for nonlinear dynamical systems using proper orthogonal decomposition (POD) is based on the information carried of the so-called snapshots. These provide the spatial distribution of the nonlinear system at discrete time instances. Thus, we are interested

in optimizing the choice of these time instances in such a manner that the error between the POD-solution and the trajectory of the dynamical system is minimized. This approach was suggested in [59] and was extended in [64] to parametrized elliptic problems. Let us briefly mention some related issues of interest. In [26, 32] the situation of missing snapshot data is investigated and gappy POD is introduced for their reconstruction. An important alternative to POD model reduction is given by reduced basis approximations; we refer to [72] and references given there. In [37] a reduced model is constructed for a parameter dependent family of large scale problems by an iterative procedure that adds new basis variables on the basis of a greedy algorithm. In the Ph.D thesis [20] a model reduction is sought of a class for a family of models corresponding to different operating stages. Suppose that we are given the n_t snapshots $\{y(t_j)\}_{j=1}^{n_t} \subset V \subset X$. The goal is to determine additional k snapshots at time instances $\tau = (\tau_1, \ldots, \tau_k)$ with $0 \leq \tau_j \leq T$, $j = 1, \ldots, k$. In [59] we propose to determine $\tau = (\tau_1, \ldots, \tau_k)$ by solving the optimization problem

$$\min_{0 \leq \tau_1, \ldots, \tau_k \leq T} \int_0^T \|y(t) - y^\ell(t)\|_V^2 \, dt, \tag{44}$$

where y and y^ℓ are the solutions to (16) and its POD Galerkin approximation, respectively. Clearly, the definition of the operator $\mathcal{R}$ given in (6) has to be modified as follows:

$$\mathcal{R}^\tau \Psi = \sum_{j=1}^{n_t} \alpha_j^\tau \langle y(t_j), \Psi \rangle_X \, y(t_j) + \sum_{j=1}^{k} \alpha_{n_t+j}^\tau \langle y(\tau_j), \Psi \rangle_X \, y(\tau_j)$$

with appropriately modified (trapezoidal) weights α_j^τ, $j = 1, \ldots, k + n_t$. Consequently, (44) becomes an optimization problem subject to the equality constraints

$$\mathcal{R}^\tau \Psi_i = \lambda_i \Psi_i, \quad i = 1, \ldots, \ell.$$

Note that no precautions are made in (44) to avoid multiple appearance of a snapshot. In fact, this would simply imply that a specific snapshot location should be given a higher weight than others. While the presented approach shows how to choose optimal snapshots in evolution equations, a similar strategy is applicable in the context of parameter dependent systems.

It turns out in our numerical tests carried out in [59] that the proposed criterion is sensitive with respect to the choice of the time instances. Moreover, the tests demonstrate the feasibility of the method in determining optimal snapshot locations for concrete diffusion equations.

Run 10 (cf. [59, Run 1]). For $T = 1$ let $Q = (0, T) \times \Omega$ and $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$. For the FE triangulation we choose a uniform grid with mesh size $h = 1/40$, i.e., we have 900 degrees of freedom for the spatial discretization. Then, we consider

$$
\begin{aligned}
y_t(t, \boldsymbol{x}) - c\Delta y(t, \boldsymbol{x}) + \beta \cdot \nabla y(t, \boldsymbol{x}) + y(t, \boldsymbol{x}) &= f(\boldsymbol{x}) && \text{for } (t, \boldsymbol{x}) \in Q, \\
c\frac{\partial y}{\partial \boldsymbol{x}}(t, \boldsymbol{x}) + q(\boldsymbol{x})y(t, \boldsymbol{x}) &= g(\boldsymbol{x}) && \text{for } (t, \boldsymbol{x}) \in \Sigma, \\
y(0, \boldsymbol{x}) &= y_\circ(\mathbf{x}) && \text{for } \boldsymbol{x} \in \Omega,
\end{aligned}
$$

where $c = 0.1$, $\beta = (0.1, -10)^\top \in \mathbb{R}^2$,

$$f(\boldsymbol{x}) = \begin{cases} 4 & \text{for all } \boldsymbol{x} = (x_1, x_2) \text{ with } (x_1 - 0.25)^2 + (x_2 - 0.65)^2 \leq 0.05, \\ 0 & \text{otherwise,} \end{cases}$$

and $y_\circ(\boldsymbol{x}) = \sin(\pi x_1)\cos(\pi x_2)$ for $\boldsymbol{x} = (x_1, x_2) \in \Omega$ (see Figure 17, left plot). Furthermore, we have
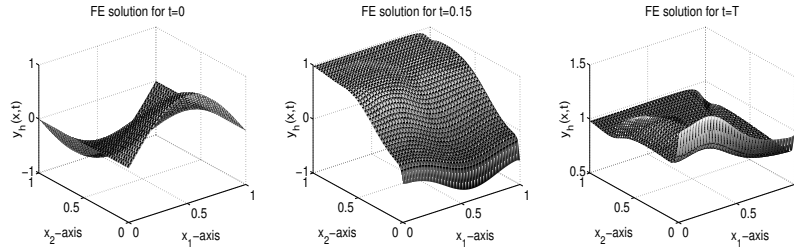
FIGURE 17. Run 10 (cf. [59, Figures 3 and 4]). initial condition $y_\circ$ (left plot), FE solution y^h for $t = 0.3$ (middle) and $t = T$ (right plot).
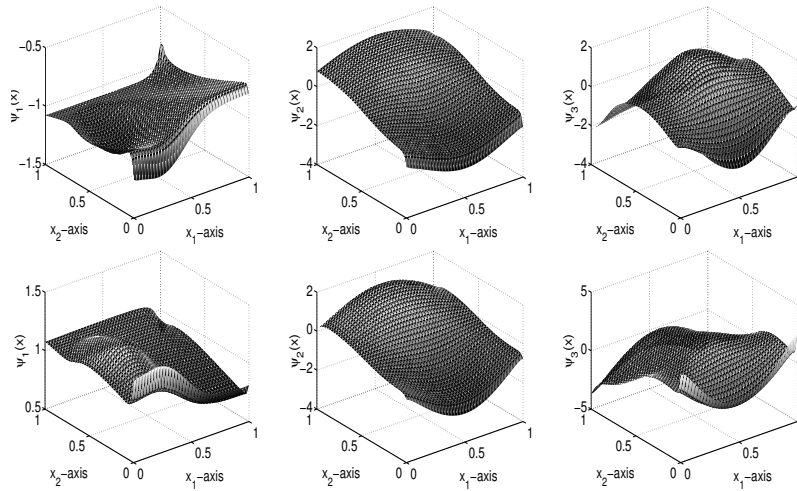

FIGURE 18. Run 10 (cf. [59, Figures 5 and 7]). POD basis Ψ_1, Ψ_2, Ψ_3 for the initial additional time instances $\tau^0 \in \mathbb{R}^4$ (upper three plots) and for the optimal additional time instances $\bar\tau \in \mathbb{R}^4$ (lower three plots).

$$q(\boldsymbol{x}) = \begin{cases} 1 & \text{for } \boldsymbol{x} = (x_1, 1) \text{ with } 0 < x_1 < 1, \\ x_2 & \text{for } \boldsymbol{x} = (1, x_2) \text{ with } 0 < x_2 < 1, \\ -2 & \text{for } \boldsymbol{x} = (x_1, 0) \text{ with } 0 < x_1 < 1, \\ 0 & \text{for } \boldsymbol{x} = (0, x_2) \text{ with } 0 < x_2 < 1, \end{cases}$$

$$g(\boldsymbol{x}) = \begin{cases} 1 & \text{for } \boldsymbol{x} = (x_1, 1) \text{ with } 0 < x_1 < 1, \\ 0 & \text{for } \boldsymbol{x} = (1, x_2) \text{ with } 0 < x_2 < 1, \text{ for } \boldsymbol{x} = (0, x_2) \text{ with } 0 < x_2 < 1, \\ -1 & \text{for } \boldsymbol{x} = (x_1, 0) \text{ with } 0 < x_1 < 1. \end{cases}$$

We utilize piecewise linear FE functions. The FE solutions $y^h = y^h(t, \boldsymbol{x})$ for $t = 0.15$ and $t = T$ are shown in Figure 17. Next we take snapshots on the fixed uniform time grid $t_j = (j-1)\Delta t$, $1 \le j \le n_t$, with $n_t = 10$ and $\Delta t = T/n_t = 0.1$. The goal is to determine four additional time instances $\bar{t} = (\bar{t}_1, \ldots, \bar{t}_4) \in [0, T]$ based on a FE approximation for (44). Since the behavior of the solution exhibits more change during the initial time interval $[0, 0.3]$ than later on, we initialize our Quasi-Newton method by the starting value $\tau^0 = (0.05, 0.15, 0.25, 0.35) \in [0, T]$. The number of POD ansatz functions is fixed to be $\ell = 3$. The corresponding value of the ROM error is approximately 0.1093. The optimal solution is given as $\bar\tau = (0.0092, 0.0076, 0.1336, 0.2882) \in [0, T]$, while the associated ROM error is approximately 0.0165, which is a reduction of about 85 %. In Figure 18 we can see that the shapes of the three POD bases changes significantly from the initial time instances $\tau^0 \in \mathbb{R}^4$ to the optimal ones $\bar\tau \in \mathbb{R}^4$. ◇

6. Optimal control with POD surrogate models

Reduced-order models are used in PDE-constrained optimization in various ways; see, e.g., [50, 86] for a survey. In optimal control problems it is sometimes necessary to compute a feedback control law instead of a fixed optimal control. In the implementation of these feedback laws models of reduced-order can play an important, and very useful role, see [11, 40, 60, 65, 68, 79]. Another useful application is the use in optimization problems, where a PDE solver is part of the function evaluation. Obviously, thinking of a gradient evaluation or even a step-size rule in the optimization algorithm, an expensive function evaluation leads to an enormous amount of computing time. Here, the reduced-order model can replace the system given by a PDE in the objective function. It is quite common that a PDE can be replaced by a five- or ten-dimensional system of ordinary differential equations. This results computationally in a very fast method for optimization compared to the effort for the computation of a single solution of a PDE. There is a large amount of literature in engineering applications in this regard, we mention only the papers [67, 71]. Recent applications can also be found in finance using the reduced models generated with the reduced basis (RB) method [76] and the POD model [85, 88] in the context of calibration for models in option pricing.

We refer to the survey article [42], where a linear-quadratic optimal control problem in an abstract setting is considered. Error estimates for the POD Galerkin approximations of the optimal control are proved. This is achieved by combining techniques from [28, 29, 44] and [56, 58]. For nonlinear problems we refer the reader to [50, 75, 86]. However, unless the snapshots are generating a sufficiently rich state space or are computed from the exact (unknown) optimal controls, it is not a priorly clear how far the optimal solution of the POD problem is from the exact one. On the other hand, the POD method is a universal tool that is applicable also to problems with time-dependent coefficients or to nonlinear equations. Moreover, by generating snapshots from the real (large) model, a space is constructed that inhibits the main and relevant physical properties of the state system. This, and its ease of use makes POD very competitive in practical use, despite of a certain heuristic flavor. In this context results for a POD a posteriori analysis are important, see e.g., [94] and [41, 54, 55, 91, 93, 96, 98]. Using a fairly standard perturbation method it is deduced how far the suboptimal control, computed on the basis of the POD model, is from the (unknown) exact one. This idea turned out to be very efficient in our examples. It is able to compensate for the lack of a priori analysis for POD methods. Let us also refer to the papers [30, 36, 69], where a posteriori error bounds are computed for linear-quadratic optimal control problems approximated by the reduced basis method.

Data- and/or simulation-based POD models depend on the data (e.g. initial values, right hand sides, boundary conditions, oberservations, etc.) which is used to generate the snapshots. If those models are used as surrogates in e.g. optimization problems with PDE constraints the algorithmical framework has to account for this fact with providing mechanisms for accordingly updating the surrogate model during the solution process. Strategies proposed in this context for optimal flow control can be found in e.g. [3, 4, 9, 34, 17]. One of the most mature methods developed in this context is Trust-Region POD proposed in [9], which since then has successfully been applied in many applications. We also refer to the work [38], where strategies for updating the POD bases are compared.

The quality of the surrogate model highly depends on its information basis, which for snapshot-based methods is given by the snapshot set, compare Section 5. The location of snapshots and also the choice of the initial control in surrogate-based optimal control is discussed in [5]. There, techniques from time-adaptive schemes for optimality systems of parabolic optimal control problems are adjusted to compute optimal time locations for snapshots generation in POD surrogate modeling for parabolic optimal control problems.

Concepts for the construction and use of POD surrogate modeling in robust optimal control of electrical machines are presented in [63, 6]. Those problems are governed by nonlinear partial

differential equations with uncertain parameters, so that robustness can be achieved by considering a worst case formulation. The resulting optimization problem then is of bilevel structure and POD reduced-order models in combination with a posteriori error estimators are used to speed up the numerical computations.

7. Miscellaneous

POD model order reduction can also be applied to provide surrogate models for high-fidelity components in networks. The general perspective is discussed in e.g. [48]. Related research for MOR of electrical networks is reported in e.g. [16, 46, 47]. The basic idea here consists in a decoupling of MOR approaches for the network and high-fidelity components which in general are modeled by PDE systems. For the latter, simulation-based POD MOR techniques are used to construct surrogate models which then are stamped back into the (reduced) electrical network. Details and performance tests are reported e.g. in [45, 47]. A short lecture series with related topics is presented under Hinze-Pilsen[2]. Further contributions to this topic can be found in [15].

Recent trends in data-driven and nonlinear MOR methods are discussed within a YouTube lecture series under Carlberg-YouTube[3].

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, and 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), 2012.

[3] K. Afanasiev and M. Hinze. Adaptive control of a wake flow using proper orthogonal decomposition. Preprint No. 648/1999, Fachbereich Mathematik, TU Berlin, 1999.

[4] K. Afanasiev and M. Hinze. Adaptive control of a wake flow using proper orthogonal decomposition. *Lecture Notes in Pure and Applied Mathematics*, 216:317-332, 2001.

[5] A. Alla, C. Gräßle, M. Hinze. A-posteriori snapshot location for POD in optimal control of linear parabolic equations. ESAIM: Mathematical Modelling and Numerical Analysis (M2AN), 52(5):1847-1873, 2018.

[6] A. Alla, M. Hinze, P. Kolvenbach, O. Lass, S. Ulbrich. A certified model reduction approach for robust parameter optimization with PDE constraints. *Adv. Comput. Math.*, 45:1221-1250, 2019.

[7] A. Alla and J. N. Kutz. Nonlinear model order reduction via dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 39:B778-B796, 2017.

[8] M.S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M.E. Rognes, and G.N. Wells. The FEniCS Project Version 1.5. *Archive of Numerical Software*, 100:9-23, 2015.

[9] E. Arian, M. Fahl, and E.W. Sachs. Trust-region proper orthogonal decomposition for flow control. Technical Report 2000-25, ICASE, 2000.

[10] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Trans. Automat. Contr.*, 53:2237–2251, 2008.

[11] J.A. Atwell, J.T. Borggaard, and B.B. King. Reduced-order controllers for Burgers' equation with a nonlinear observer. *International Journal of Applied Mathematics and Computer Science*, 11:1311–1330, 2001.

[12] S. Banholzer, E. Makarov, and S. Volkwein. POD-based multiobjective optimal control of time-variant heat phenomena. *Lecture Notes in Computational Science and Engineering*, 126:881-888, 2019.

[13] H.T. Banks, M.L. Joyner, B. Winchesky, and W.P. Winfree. Nondestructive evaluation using a reduced-order computational methodology. *Inverse Problems*, 16:1–17, 2000.

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

[15] P. Benner. (ed.). System Reduction for Nanoscale IC Design. Springer International Publishing, Cham, Switzerland, 2017.

[16] P. Benner, M. Hinze, and E.J.W. ter Maten (eds). *Model Reduction for Circuit Simulation*. Lecture Notes in Electrical Engineering, vol. 74, Springer, 2011.

---

[2]https://slideslive.com/38894790/mathematical-aspects-of-proper-orthogonal-decomposition-pod-iii
[3]https://www.youtube.com/watch?v=KOHxCIx04Dg

[17] M. Bergmann, L.Cordier. Control of the cylinder wake in the laminar regime by Trust-Region methods and POD Reduced Order Models. J. Comp. Phys. 227:7813-7840, 2009.

[18] M. Bergmann, C.-H. Bruneau, A. Iollo. Enablers for robust POD models. J. Comp. Phys. 228:516-538, 2009.

[19] J. F. Blowey and 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.

[20] T. Bui-Thanh. *Model-constrained optimization methods for reduction of parameterized systems*. PhD thesis, MIT, USA, 2007.

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

[22] D. Chapelle, A. Gariah, and J. Saint-Marie. Galerkin approximation with proper orthogonal decomposition: new error estimates and illustrative examples. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:731-757, 2012.

[23] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78:539–575, 2000.

[24] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32:2737-2764, 2010.

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

[26] M. Damodoran, T. Bui-Thanh, and K.Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *American Institute of Aeronautics and Astronautics (AIAA) Journal*, 42:1505-1516, 2004.

[27] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5: Evolution Problems I.* Springer, Berlin, Heidelberg, New York, 2000.

[28] K. Deckelnick and M. Hinze. Error estimates in space and time for tracking-type control of the instationary Stokes system. International Series of Numerical Mathematics, 143:87-103, 2002.

[29] K. Deckelnick and M. Hinze. Semidiscretization and error estimates for distributed control of the instationary Navier-Stokes equations. *Numerische Mathematik*, 97:297-320, 2004.

[30] L. Dede. Reduced basis method and a posteriori error estimation for parametrized linear-quadratic optimal control problems. *SIAM Journal on Scientific Computing*, 32:997-1019, 2010.

[31] Z. Drmac, and 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.

[32] R. Everson and L. Sirovich. The Karhunen-Loève precedure for gappy data. *Journal of the Optical Society of America*, 12:1657–1664, 1995.

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

[34] M. Fahl. Trust region methods for flow control based reduced order modeling. Dissertation, Fachbereich IV, Universität Trier, 2000.

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

[36] M. Grepl and M. Kärcher. A posteriori error estimation for reduced order solutions of parametrized parabolic optimal control problems. *Mathematical Modelling and Numerical Analysis*, 48:1615-1638, 2014.

[37] M.A. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41:575-605, 2007.

[38] C. Gräßle, M. Gubisch, S. Metzdorf, S. Rogg and S. Volkwein. POD basis updates for nonlinear PDE control. *at – Automatisierungstechnik*, 65: 298-307, 2017.

[39] C. Gräßle and M. Hinze. POD reduced-order modeling for evolution equations utilizing arbitrary finite element discretizations. *Adv. Comput. Math.*, 44:1941-1978, 2018.

[40] L. Grüne, L. Mechelli, S. Pirkelmann and S. Volkwein. Performance estimates for economic model predictive control and their application in POD-based implementations. *To appear in System Modeling and Optimization, Proceedings of the 28th IFIP TC 7 Conference 2018 on System Modelling and Optimization*, 2020.

[41] M. Gubisch and S. Volkwein. POD a-posteriori error analysis for optimal control problems with mixed control-state constraints. *Computational Optimization and Applications*, 58:619-644, 2014.

[42] M. Gubisch and S. Volkwein. Proper orthogonal decomposition for linear-quadratic optimal control. In P. Benner, A. Cohen, M. Ohlberger, and K Willcox, editors, *Model Reduction and Approximation: Theory and Algorithms*, pages 5-66. SIAM, Philadelphia, PA, 2017.

[43] M. Hintermüller, M. Hinze, and 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.

[44] M. Hinze. A variational discretization concept in control constrained optimization: the linear-quadratic case. *Computational Optimization and Applications*, 30:45-61, 2005.

[45] M. Hinze and M. Kunkel. Residual based sampling in POD model order reduction of drift-diffusion equations in parametrized electrical networks. *Zeitschrift für Angewandte Mathematik und Mechanik*, 92:91-104, 2012.

[46] M. Hinze, M. Kunkel, U. Matthes, and M. Vierling. Model order reduction of integrated circuits in electrical networks. System Reduction for Nanoscale IC Design Hamburger, Peter Benner (Ed), Mathematics in Industry 20, 2014.

[47] M. Hinze, M. Kunkel, A. Steinbrecher, and T. Stykel. Model order reduction of coupled circuit-device systems. *Int. J. Numer. Model.* 25:362-377, 2012.

[48] M. Hinze and U. Matthes. Model order reduction for networks of ODE and PDE systems. D. Hömberg and F. Tröltzsch (Eds.): CSMO 2011, IFIP AICT 391:92-101, 2013.

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

[50] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control. *Lecture Notes in Computational Science and Engineering*, 45:261–306, 2005.

[51] M. Hinze and S. Volkwein. Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition. *Computational Optimization and Applications*, 39:319-345, 2008.

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

[53] P. Holmes, J.L. Lumley, G. Berkooz, and C.W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge Monographs on Mechanics. Cambridge University Press, Cambridge, 2nd ed., 2012.

[54] M. Kahlbacher and S. Volkwein. POD a-posteriori error based inexact SQP method for bilinear elliptic optimal control problems. *Mathematical Modelling and Numerical Analysis*, 46:491-511, 2012.

[55] E. Kammann, F. Tröltzsch, and S. Volkwein. A method of a-posteriori error estimation with application to proper orthogonal decomposition. *Mathematical Modelling and Numerical Analysis*, 47:555-581, 2013.

[56] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90:117-148, 2001.

[57] K. Kunisch and S. Volkwein. Crank-Nicolson Galerkin proper orthogonal decomposition approximations for a general equation in fluid dynamics. In *Proceedings of the 18th GAMM Seminar on Multigrid and Related Methods for Optimization Problems*, pages 97-114, 2002.

[58] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40:492-515, 2002.

[59] K. Kunisch and S. Volkwein. Optimal snapshot location for computing POD basis functions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44:509-529, 2010.

[60] K. Kunisch, S. Volkwein, and L. Xie. HJB-POD based feedback design for the optimal control of evolution problems. *SIAM Journal on Applied Dynamical Systems*, 3:701-722, 2004.

[61] S. Lall, J.E. Marsden, and S. Glavaski. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal on Robust Nonlinear Control*, 12:519-535, 2002.

[62] J. Lang, S. Ullmann, and M. Rotkvic. POD-Galerkin reduced-order modeling with adaptive finite element snapshots. *J. Comput. Phys.*, 325:244-258, 2016.

[63] O. Lass and S. Ulbrich. Model order reduction techniques with a posteriori error control for nonlinear robust optimization governed by partial differential equations. *SIAM J. Sci. Comput.*, 39:S112-S139, 2016.

[64] O. Lass and S. Volkwein. Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location. *Computational Optimization and Applications*, 58:645-677, 2014.

[65] F. Leibfritz and S. Volkwein. Reduced order output feedback control design for PDE systems using proper orthogonal decomposition and nonlinear semidefinite programming. *Linear Algebra and Its Applications*, 415:542-575, 2006.

[66] A. Logg, K.A. Mardal and 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.*

[67] H.V. Ly and H.T. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer Modeling*, 33:223-236, 2001.

[68] L. Mechelli and S. Volkwein. POD-based economic model predictive control for heat-convection phenomena. *Lecture Notes in Computational Science and Engineering*, 126:663-670, 2019.

[69] F. Negri, G. Rozza, A. Manzoni, and A. Quateroni. Reduced basis method for parametrized elliptic optimal control problems. *SIAM Journal on Scientific Computing*, 35:A2316-A2340, 2013.

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

[71] B. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335-363, 2003.

[72] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*. MIT-Pappalardo Graduate Monographs in Mechanical Engineering. (C) MIT, Massachusetts Institute of Technology, Cambridge, MA, US, 2007.

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

[74] J. R. Phillips. Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* , 22:171-187, 2003.

[75] R. Pinnau. Model reduction via proper orthogonal decomposition. In W.H.A. Schilder, H.A. van der Vorst, and J. Rommes, editors, *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry*, pages 95-109. Springer, Berlin, Heidelberg, 2008.

[76] O. Pironneau. Calibration of options on a reduced basis. *Journal of Computational and Applied Mathematics*, 232:139-147, 2006.

[77] R. Rannacher. Finite element solution of diffusion problems with irregular data. Numerische Mathematik 43:309-327, 1984.

[78] M. Rathinam and L. Petzold. Dynamic iteration using reduced order models: a method for simulation of large scale modular systems. *SIAM Journal on Numerical Analysis*, 40:1446-1474, 2002.

[79] S.S. Ravindran. Reduced-order adaptive controllers for fluid flows using POD. *SIAM Journal on Scientific Computing*, 15:457-478, 2000.

[80] S.S. Ravindran. Error analysis for Galerkin POD approximation of the nonstationary Boussinesq equations. *Numerical Methods for Partial Differential Equations*, 27:1639-1665, 2011.

[81] M. Reed and B. Simon. *Methods of Modern Mathematical Physics I: Functional Analysis*. Academic Press, New York, 1980.

[82] C.W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15:997-1013, 2005.

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

[84] M. Rewiewski and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22:155-170, 2003.

[85] E.W. Sachs and M. Schu. A priori error estimates for reduced order models in finance. *ESAIM: M2AN*, 47:449-469, 2013.

[86] E.W. Sachs and S. Volkwein. POD Galerkin approximations in PDE-constrained optimization. *GAMM-Mitteilungen*, 33:194-208, 2010.

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

[88] M. Schu. *Adaptive Trust-Region POD Methods and their Application in Finance*. PhD thesis, University of Trier, 2013.

[89] J.R. Singler. New POD expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs. *SIAM Journal on Numerical Analysis*, 52:852-876, 2014.

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

[91] A. Studinger and S. Volkwein. Numerical analysis of POD a-posteriori error estimation for optimal control. *International Series of Numerical Mathematics*, 164:137-158, 2013.

[92] R. Temam. *Infinite-Dimensional Dynamical Systems in Mechanics and Physics. Appl. Math. Sci.*, volume 68, Springer, New York, 1997.

[93] T. Tonn, K. Urban, and S. Volkwein. Comparison of the reduced-basis and POD a-posteriori error estimators for an elliptic linear quadratic optimal control problem. *Mathematical and Computer Modelling of Dynamical Systems*, 17:355-369, 2011.

[94] F. Tröltzsch and S. Volkwein. POD a-posteriori error estimates for linear-quadratic optimal control problems. *Computational Optimization and Applications*, 44:83-115, 2009.

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

[96] S. Volkwein. Optimality system POD and a-posteriori error analysis for linear-quadratic problems. *Control and Cybernetics*, 40:1109-1125, 2011.

[97] S. Volkwein. Optimal control of a phase-field model using proper orthogonal decomposition. *Zeitschrift für Angewandte Mathematik und Mechanik*, 81:83-97, 2001.

[98] G. Vossen and S. Volkwein. Model reduction techniques with a-posteriori error analysis for linear-quadratic optimal control problems. Numerical Algebra, Control and Optimization, 2:465-485, 2012.

[99] Z. Whang. Nonlinear model reduction based on the finite element method with interpolated coefficients: semilinear parabolic equations. *Numer. Meth. Partial. Diff. Eqs.*, 31:1713-1741, 2015.

[100] D. Wells, Z. Wang, X. Xie, T. Iliescu. An Evolve-Then-Filter Regularized Reduced Order Model For Convection-Dominated Flows. arXiv:1506.07555v2, 2018.

[101] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *American Institute of Aeronautics and Astronautics*, 40:2323–2330, 2002.