Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms

Peter Benner[a], Pawan Goyal[a], Boris Kramer[b,*], Benjamin Peherstorfer[c], Karen Willcox[d]

[a] *Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany*
[b] *Department of Mechanical and Aerospace Engineering, University of California San Diego, CA, United States*
[c] *Courant Institute of Mathematical Sciences, New York University, NY, United States*
[d] *Oden Institute for Computational Engineering & Sciences, The University of Texas at Austin, TX, United States*

Abstract

This work presents a non-intrusive model reduction method to learn low-dimensional models of dynamical systems with non-polynomial nonlinear terms that are spatially local and that are given in analytic form. In contrast to state-of-the-art model reduction methods that are intrusive and thus require full knowledge of the governing equations and the operators of a full model of the discretized dynamical system, the proposed approach requires only the non-polynomial terms in analytic form and learns the rest of the dynamics from snapshots computed with a potentially black-box full-model solver. The proposed method learns operators for the linear and polynomially nonlinear dynamics via a least-squares problem, where the given non-polynomial terms are incorporated in the right-hand side. The least-squares problem is linear and thus can be solved efficiently in practice. The proposed method is demonstrated on three problems governed by partial differential equations, namely the diffusion-reaction Chafee-Infante model, a tubular reactor model for reactive flows, and a batch-chromatography model that describes a chemical separation process. The numerical results provide evidence that the proposed approach learns reduced models that achieve comparable accuracy as models constructed with state-of-the-art intrusive model reduction methods that require full knowledge of the governing equations.

Keywords: Model reduction, data-driven modeling, nonlinear dynamical systems, scientific machine learning, operator inference

1. Introduction

Model reduction constructs computationally efficient reduced models of large-scale dynamical systems by approximating the high-dimensional states in low-dimensional subspaces of the state space. Reduced models are especially beneficial in the many-query and outer-loop setting, such as in optimization, design, uncertainty quantification, and control, where models are evaluated many times for different inputs. An overview of the field of model reduction can be found in, e.g., [1, 2]. Most traditional model reduction techniques are of intrusive nature, which means that full knowledge of the governing equations and the discretization of the full models of the dynamical systems of interest are required. The intrusive nature limits the scope of traditional model reduction methods. For example, consider proprietary software that implements the simulation of a physical process. Details, and access to, the governing equations, discretization, and solver typically are unavailable when working with proprietary software and thus traditional, intrusive model reduction methods are not applicable. In contrast, non-intrusive model reduction techniques aim to learn reduced models from snapshots, i.e., either numerical approximations or measurements of the states and the outputs of the dynamical systems. There is a variety of non-intrusive model reduction techniques for linear systems, yet methods for learning reduced-order nonlinear systems—which is the focus of this paper—are more scarce.

*Corresponding author
Email address: bmkramer@ucsd.edu; +1 858-246-5327 (Boris Kramer)

For linear systems, the Loewner framework constructs reduced models from input-output measurements by fitting a linear-time invariant model to frequency or time-domain data [3, 4, 5, 6]. System identification techniques such as the matrix pencil approach [7], eigensystem realization algorithm [8, 9, 10, 11], and vector fitting [12, 13] can accurately approximate dynamical systems and in some cases even retain structure. For nonlinear systems, non-intrusive model reduction generally requires a choice of parametrization of the nonlinear term. For example, the Loewner approach has been extended from linear to bilinear [14] and quadratic-bilinear systems [15]. Techniques based on libraries select the nonlinear functions that compose the right-hand-side of a dynamical system from candidate functions that are chosen either by expert knowledge [16] or through sparse approximation techniques [17, 18]. Dynamic mode decomposition [19, 20] learns reduced models for nonlinear dynamical systems, yet its success depends on knowledge of suitable observables that make the input-output map low-dimensional [21]. A promising approach to consider DMD modes together with the fluctuation-dissipation theorem allows the authors in [22] to learn a linear ROM for nonlinear turbulent flows. There are also hybrid approaches to learn only part of the nonlinear system, e.g., in data-driven closure modeling, where a projection-based framework is pursued, but correction terms are learned from data [23], or the Galerkin-ROMs can be re-calibrated [24, 25] to achieve greater accuracy. Another approach to learning nonlinear reduced models is operator inference [26] that is applicable if the nonlinear terms are polynomials in the state. The operator inference framework has been extended to general nonlinear systems by utilizing variable transformations [27].

In practice, one is often confronted with a mixture of the intrusive and non-intrusive setting, such as when the governing equations of the system of interest are known yet details about the discretization are unavailable. For example, if one considers a reaction-diffusion process (such as the tubular reactor example in Section 4.2), then the reaction term might be given analytically, whereas the discretization of the terms corresponding to the diffusion are unavailable. We thus propose a novel non-intrusive model reduction method that incorporates terms that are available analytically and that learns reduced operators for all other terms from snapshot data. To this end, we build on the operator inference method and extend it in multiple directions. First, we explicitly incorporate knowledge about non-polynomial terms that are given analytically. Second, we combine operator inference with empirical interpolation [28, 29, 30] to make the learned models online efficient in case of non-polynomial nonlinear terms. Third, we demonstrate that structure in coupled (partial) differential equations can be retained in the learned reduced models, which increases the predictive capabilities of the reduced models. We show that under certain conditions the learned model converges to the same model that is derived with intrusive model reduction methods that require full knowledge of the governing equations and the full-model solvers. Numerical results demonstrate that our approach learns predictive reduced models for diffusion-reaction systems and simulations of chemical processes.

The manuscript is structured as follows. Section 2 describes the problem setting and presents background material on model reduction. Section 3 presents the proposed operator inference problem for nonlinear non-polynomial systems. Section 4 presents numerical results that provide evidence of the wide scope of our approach. Conclusions are drawn in Section 5.

2. Problem setting and preliminaries

In Section 2.1, we introduce the partial differential equation (PDE) model considered herein, followed by its discretized version in Section 2.2. In Section 2.3, we describe the standard projection-based model reduction technique and in Section 2.4 we state the problem setting of this paper.

2.1. Partial differential equation model

Let $0 < T \in \mathbb{R}$ and consider a PDE of the form

$$\frac{\partial s}{\partial t} = \mathcal{A}(s) + \mathcal{H}(s) + f(t, s) + \mathcal{B}(u) \tag{1}$$

for time $t \in (0, T]$, input $u(t)$, state $s(x, t)$ with $x \in \Omega \subseteq \mathbb{R}^d$, $d = 1, 2, 3$, and Ω a bounded open domain with boundary $\partial\Omega$. We assume boundary conditions $s_{|\partial\Omega}$ and initial conditions $s(x, 0)$ are given such that

the corresponding initial boundary value problem with governing PDE (1) is well posed up to time T. In equation (1), $\mathcal{A}$ is a linear operator (satisfying $\mathcal{A}(as) = a\mathcal{A}(s)$ for all $a \in \mathbb{R}$ and $\mathcal{A}(s_1 + s_2) = \mathcal{A}(s_1) + \mathcal{A}(s_2)$, e.g., a difference or integral operator), the operator $\mathcal{H}$ is quadratic in s (satisfying $\mathcal{H}(as) = a^2\mathcal{H}(s)$ for all s, e.g., $\mathcal{H}(s) = s \cdot \nabla s$), the nonlinear function is $f(t, s)$, and $\mathcal{B}$ is a linear input operator.

2.2. Discretized model

A semi-discrete numerical model for the PDE (1) is given by

$$\dot{\mathbf{s}}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{H}(\mathbf{s}(t) \otimes \mathbf{s}(t)) + \mathbf{f}(t, \mathbf{s}(t)) + \mathbf{B}\mathbf{u}(t), \tag{2}$$

with finite-dimensional state $\mathbf{s}(t) \in \mathbb{R}^n$, input $\mathbf{u}(t) \in \mathbb{R}^m$, linear operator $\mathbf{A} \in \mathbb{R}^{n \times n}$, quadratic operator $\mathbf{H} \in \mathbb{R}^{n \times n^2}$, input operator $\mathbf{B} \in \mathbb{R}^{n \times m}$. The initial condition for the semi-discrete model (2) is $\mathbf{s}(0) = \mathbf{s}_0$ and the input at time $t = 0$ is $\mathbf{u}(0) = \mathbf{u}_0$. Throughout this paper, we call (2) the full-order model (FOM). Here, $\otimes$ denotes the column-wise Kronecker product, which for a column vector $\mathbf{s} = [s_1, s_2, \ldots, s_n]^\top$ is given by

$$\mathbf{s} \otimes \mathbf{s} = [s_1^2 \;\; s_1 s_2 \;\; \ldots s_1 s_n \;\; s_2 s_1 \;\; s_2^2 \;\; \ldots \;\; s_2 s_m \;\; \ldots s_m^2] \in \mathbb{R}^{n^2},$$

and for a matrix $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k] \in \mathbb{R}^{n \times k}$ is given by

$$\mathbf{S} \otimes \mathbf{S} = [\mathbf{s}_1 \otimes \mathbf{s}_1 \;\; \mathbf{s}_2 \otimes \mathbf{s}_2 \;\; \ldots \;\; \mathbf{s}_k \otimes \mathbf{s}_k] \in \mathbb{R}^{n^2 \times k}.$$

In the following, we assume that each component function $f_1, \ldots, f_n : [0, T] \times \mathbb{R} \to \mathbb{R}$ of $\mathbf{f}$ requires evaluating f from the continuous model (1) at a single component of the state vector $\mathbf{s}(t) = [s_1(t), \ldots, s_n(t)]^\top$. Formally, this means that $\mathbf{f}$ is

$$\mathbf{f}(t, \mathbf{s}) = \begin{bmatrix} f(t, s_1) \\ \vdots \\ f(t, s_n) \end{bmatrix}. \tag{3}$$

We introduced the PDE model (1) and the definition of $\mathbf{f}$ in (3) for a scalar model (describing the evolution of one physical quantity, e.g., velocity) for ease of notation. For PDEs with ℓ physical variables, that means that each component function in (3) can depend on other physical variables at the same spatial location. Next, we give a concrete example of a PDE and its discretization with non-polynomial nonlinear terms as in (3).

Example 1. *Consider the PDE*

$$\frac{\partial}{\partial t} s(x, t) = \frac{\partial^2}{\partial x^2} s(x, t) + e^{-\beta t} s(x, t)^{-\alpha} + b(x)u(t)$$

with state $s(x, t)$, diffusion operator $\frac{\partial}{\partial x^2}$, time-dependent reaction term $f(t, s) = e^{-\beta t} s^{-\alpha}$, one-dimensional input $u(t)$, input function $b(x)$ and constants $\alpha, \beta \in \mathbb{R}_+$. Let $\mathbf{s}$ denote the discretized, finite-dimensional state vector obtained through a finite difference approximation. A discretization of the diffusion operator results in a matrix representation $\mathbf{A}$, whose structure depends on the discretization scheme, approximation order, and boundary conditions. Likewise, the discretized version of $b(x)$ is $\mathbf{B}$. For a vector $\mathbf{s}$, we define $\mathbf{s}^{-\alpha} := [s_i]^{-\alpha}$, $i = 1, 2, \ldots, n$ as the componentwise power of the vector entries. The discretized system reads as

$$\dot{\mathbf{s}}(t) = \mathbf{A}\mathbf{s}(t) + e^{-\beta t}\mathbf{s}(t)^{-\alpha} + \mathbf{B}u(t).$$

Note that the nonlinear function $\mathbf{f}(t, \mathbf{s}(t)) = e^{-\beta t}\mathbf{s}(t)^{-\alpha}$ does not require approximation of spatial derivatives. In particular, evaluating the semi-discrete nonlinear function $\mathbf{f}(t, \mathbf{s})$ only requires application of $f(t, s)$ at every component of $\mathbf{s}$.

In this work, we consider functions $f(t, s)$ that are continuous in s and that are *spatially local*, which means they involve nonlinear expressions in the state s; the class of problems not considered in this paper are functions f that include differentials or integrals of s. Other examples of spatially local nonlinear terms are the Arrhenius reaction model $\exp(\gamma - \frac{\gamma}{s})$ in the tubular reactor model in Section 4.2 and the function $\frac{s}{\alpha + s}$ in the Batch Chromatography example in Section 4.3; see also [28, 30].

2.3. Traditional projection-based model reduction for dynamical systems

In projection-based model reduction, the semi-discrete model (2) is projected onto a low-dimensional subspace. Let $r \ll n$ and let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_r] \in \mathbb{R}^{n \times r}$ be an orthonormal matrix, where the columns of $\mathbf{V}$ span an r-dimensional subspace of $\mathbb{R}^n$. A basis matrix $\mathbf{V}$ of such a subspace can be computed with, e.g., proper orthogonal decomposition (POD) [31, 32, 33]. Let $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_k]$ be snapshot matrix containing solution snapshots $\mathbf{s}_i \approx \mathbf{s}(t_i)$, where the $\mathbf{s}_i$ are solutions to (2) computed with a time-stepping scheme. POD computes the basis matrix $\mathbf{V}$ from $\mathbf{S}$ via singular value decomposition (SVD).

Consider now the reduced operators

$$\widetilde{\mathbf{A}} = \mathbf{V}^\top \mathbf{A}\mathbf{V} \in \mathbb{R}^{r \times r}, \quad \check{\mathbf{H}} = \mathbf{V}^\top \mathbf{H}(\mathbf{V} \otimes' \mathbf{V}) \in \mathbb{R}^{r \times r^2}, \quad \widetilde{\mathbf{B}} = \mathbf{V}^\top \mathbf{B} \in \mathbb{R}^{r \times m}, \tag{4}$$

which are the restrictions of the matrices of the FOM from (2) to the image of $\mathbf{V}$. To remove redundant terms in the original Kronecker product, we use throughout the manuscript the product $\mathbf{s} \otimes' \mathbf{s}$ that contains as components $\mathbf{s}_i\mathbf{s}_j, i, j = 1, 2, \ldots, n, j \geq i$, which is the Kronecker product yet the duplicate terms are removed. For instance, for $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2]^\top$, the standard Kronecker product yields $\mathbf{s} \otimes \mathbf{s} = [\mathbf{s}_1^2 \ \mathbf{s}_1\mathbf{s}_2 \ \mathbf{s}_2\mathbf{s}_1 \ \mathbf{s}_2^2]^\top$ and without redundant terms we have $\mathbf{s} \otimes' \mathbf{s} = [\mathbf{s}_1^2 \ \mathbf{s}_1\mathbf{s}_2 \ \mathbf{s}_2^2]^\top$. To this end, we define the matrix $\widetilde{\mathbf{H}} \in \mathbb{R}^{r \times (r^2+r)/2}$ (without redundant terms) as

$$\widetilde{\mathbf{H}} = \check{\mathbf{H}}_{:,r(i-1)+j}, \qquad i = 1, \ldots r, \ i \leq j \leq r,$$

where $\check{\mathbf{H}}_{:,j}$ denotes jth column of the matrix $\check{\mathbf{H}}$. Moreover, we define the function $\widetilde{\mathbf{f}} : [0, T] \times \mathbb{R}^r \to \mathbb{R}^r$ as

$$\widetilde{\mathbf{f}}(t, \widetilde{\mathbf{s}}(t)) = \mathbf{V}^\top \mathbf{f}(t, \mathbf{V}\widetilde{\mathbf{s}}(t)),$$

where $\widetilde{\mathbf{s}} \in \mathbb{R}^r$. Note that there are efficient reduction strategies that approximately compute $\widetilde{\mathbf{f}}$ so that $\mathbf{f}$ does not have to be evaluated at all n components of $\mathbf{V}\widetilde{\mathbf{s}}$ [34, 28, 35, 30, 36]; we will return to such reduction strategies of $\widetilde{\mathbf{f}}$ in Section 3.3. The reduced operators and $\widehat{\mathbf{f}}$ define the reduced-order model (ROM)

$$\dot{\widetilde{\mathbf{s}}}(t) = \widetilde{\mathbf{A}}\widetilde{\mathbf{s}}(t) + \widetilde{\mathbf{H}}(\widetilde{\mathbf{s}}(t) \otimes' \widetilde{\mathbf{s}}(t)) + \widetilde{\mathbf{f}}(t, \widetilde{\mathbf{s}}(t)) + \widetilde{\mathbf{B}}\mathbf{u}(t) \tag{5}$$

with the reduced state $\widetilde{\mathbf{s}}(t) \in \mathbb{R}^r$ at time t and with initial condition $\widetilde{\mathbf{s}}(0) = \widetilde{\mathbf{s}}_0$. Constructing reduced operators (4) via projection requires that the operators $\mathbf{A}, \mathbf{B}, \mathbf{H}$ of the system (2) are available in assembled form or as matrix-vector products. Furthermore, each evaluation of $\widetilde{\mathbf{f}}$ requires evaluating components of $\mathbf{f}$.

2.4. Problem setting

In this work, we consider the situation that we have a PDE model (1) with spatially local nonlinear term f as described in Section 2.1, and that we can obtain snapshot data $\mathbf{s}(t_i), \mathbf{u}(t_i)$ of the model (2) at times $0 = t_0 < t_1 < \cdots < t_k$. However, the discretized operators $\mathbf{A}, \mathbf{H}, \mathbf{B}$ are unavailable. The goal of this work is to learn a nonlinear ROM of the form (5).

3. Operator inference for nonlinear systems

In Section 3.1, we extend the operator inference framework for low-order polynomial systems from [26] to the setting described in Section 2.1 with general non-polynomial nonlinear function f. In Section 3.2, we present a result that shows that under certain conditions, the learned operators converge to their projection-based counterparts. In Section 3.3, we discuss empirical interpolation [28, 30] to efficiently reduce the non-polynomial nonlinear terms in the learned ROMs.

3.1. Operator inference method

We now extend the operator inference method introduced in [26] for polynomially nonlinear systems to systems with non-polynomial nonlinear terms that are spatially local. Let $\mathbf{s}_1, \ldots, \mathbf{s}_k$ be the solutions of the FOM (2) at time steps $t_1, \ldots, t_k$ computed with a time stepping scheme and initial condition $\mathbf{s}_0$. Let the inputs $\mathbf{u}(t_0), \ldots, \mathbf{u}(t_k)$ be sampled at the same time steps. Then, the snapshot and input trajectory are defined as

$$\mathbf{S} := \begin{bmatrix} | & | & & | \\ \mathbf{s}_0 & \mathbf{s}_1 & \cdots & \mathbf{s}_k \\ | & | & & | \end{bmatrix}, \qquad \mathbf{U} := \begin{bmatrix} | & | & & | \\ \mathbf{u}(t_0) & \mathbf{u}(t_1) & \cdots & \mathbf{u}(t_k) \\ | & | & & | \end{bmatrix}. \tag{6}$$

Note that we can evaluate $\mathbf{f}(t_i, \mathbf{s}(t_i))$ using the solution $\mathbf{s}(t)$ and the analytical form of the spatially local nonlinear term f. That is, we can post-process the snapshots to obtain

$$\mathbf{F} = \begin{bmatrix} | & | & & | \\ \mathbf{f}(t_0, \mathbf{s}(t_0)) & \mathbf{f}(t_1, \mathbf{s}(t_1)) & \cdots & \mathbf{f}(t_k, \mathbf{s}(t_k)) \\ | & | & & | \end{bmatrix}. \tag{7}$$

In the operator inference framework, the goal is to learn a ROM of the form (5) directly from data. One starts with the projected trajectories of $\mathbf{s}(t)$, $\dot{\mathbf{s}}(t)$ and $\mathbf{f}(t, \mathbf{s})$, which are obtained via the following projections onto the basis matrix $\mathbf{V} \in \mathbb{R}^{n \times r}$:

$$\widehat{\mathbf{S}} = \mathbf{V}^\top \mathbf{S}, \qquad \widehat{\mathbf{F}} = \mathbf{V}^\top \mathbf{F}. \tag{8}$$

We denote with $\dot{\widehat{\mathbf{s}}}_k$ the time derivative approximation of $\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{s}}(t_k)$, which can be computed from $\widehat{\mathbf{s}}$ using a time derivative approximation (see, e.g., [37, 38, 39]). We store the time-derivative approximations in the matrix

$$\dot{\widehat{\mathbf{S}}} := \begin{bmatrix} | & | & & | \\ \dot{\widehat{\mathbf{s}}}(t_0) & \dot{\widehat{\mathbf{s}}}(t_1) & \cdots & \dot{\widehat{\mathbf{s}}}(t_k) \\ | & | & & | \end{bmatrix}. \tag{9}$$

The optimization problem to compute $\widehat{\mathbf{A}}$, $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{H}}$ from the above projected data is:

$$\min_{\widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{H}}} \| \underbrace{\dot{\widehat{\mathbf{S}}} - \widehat{\mathbf{F}}}_{:= \widehat{\mathbf{R}}} - \widehat{\mathbf{A}}\widehat{\mathbf{S}} - \widehat{\mathbf{B}}\mathbf{U} - \widehat{\mathbf{H}}(\widehat{\mathbf{S}} \otimes' \widehat{\mathbf{S}}) \|_F \tag{10}$$

where $\widehat{\mathbf{R}}$ contains only known terms. Taken together, the solution of the operator inference problem for $\widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{H}}$ yields a learned ROM of the form

$$\dot{\widehat{\mathbf{s}}}(t) = \widehat{\mathbf{A}}\widehat{\mathbf{s}}(t) + \widehat{\mathbf{H}}(\widehat{\mathbf{s}}(t) \otimes' \widehat{\mathbf{s}}(t)) + \mathbf{V}^\top \mathbf{f}(t, \mathbf{V}\widehat{\mathbf{s}}(t)) + \widehat{\mathbf{B}}\mathbf{u}(t), \tag{11}$$

which can then be used for predictive simulations. Algorithm 1 summarizes the algorithmic steps of this section. In Figure 1, we illustrate the differences of the proposed approach of using operator inference together with knowledge of spatially nonlinear functions to learn ROMs as compared to an intrusive projection-based ROM.

Remark 1. *We introduced the operator inference framework in the context of continuous-time nonlinear systems. However, the framework carries over directly to discrete time systems of the form*

$$\mathbf{s}_{k+1} = \mathbf{A}\mathbf{s}_k + \mathbf{H}(\mathbf{s}_k \otimes' \mathbf{s}_k) + \mathbf{f}(t_k, \mathbf{s}_k) + \mathbf{B}\mathbf{u}_k. \tag{12}$$

In this case, we replace $\mathbf{S}$ *with* $\mathbf{S}_0 = [\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{k-1}]$*, set* $\mathbf{U} = [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{k-1}]$*, and replace the time-derivative matrix* $\dot{\mathbf{S}}$ *with* $\mathbf{S}_1 = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k]$*.*

Algorithm 1 Operator inference for nonlinear systems.

1: **Input:** $\mathbf{s}_0, \mathbf{S}, \mathbf{U}, \mathbf{F}$ and a user-specified tolerance `tol`.
2: Collect derivative data $\dot{\mathbf{S}}$ by numerical approximation or evaluation of a right-hand side residual on the snapshot matrix $\mathbf{S}$.
3: Compute the r dominant POD basis vectors of $\mathbf{S}$, resulting in $\mathbf{V}$ such that

$$\frac{\|\mathbf{S} - \mathbf{V}\mathbf{V}^\top \mathbf{S}\|}{\|\mathbf{S}\|} \leq \texttt{tol}.$$

4: Determine projected initial condition $\widehat{\mathbf{s}}_0$, reduced states $\widehat{\mathbf{S}}$, reduced derivative $\dot{\widehat{\mathbf{S}}}$, and reduced nonlinear terms $\widehat{\mathbf{F}}$:

$$\widehat{\mathbf{s}}_0 := \mathbf{V}^\top \mathbf{s}_0, \quad \widehat{\mathbf{S}} := \mathbf{V}^\top \mathbf{S}, \quad \dot{\widehat{\mathbf{S}}} = \mathbf{V}^\top \dot{\mathbf{S}}, \quad \widehat{\mathbf{F}} := \mathbf{V}^\top \mathbf{F}.$$

5: Solve the optimization problem (10), yielding $\widehat{\mathbf{A}}$, $\widehat{\mathbf{B}}$, $\widehat{\mathbf{H}}_{\text{sq}}$.
6: **Output:** $\widehat{\mathbf{s}}_0$, $\widehat{\mathbf{A}}$, $\widehat{\mathbf{B}}$, $\widehat{\mathbf{H}}$.

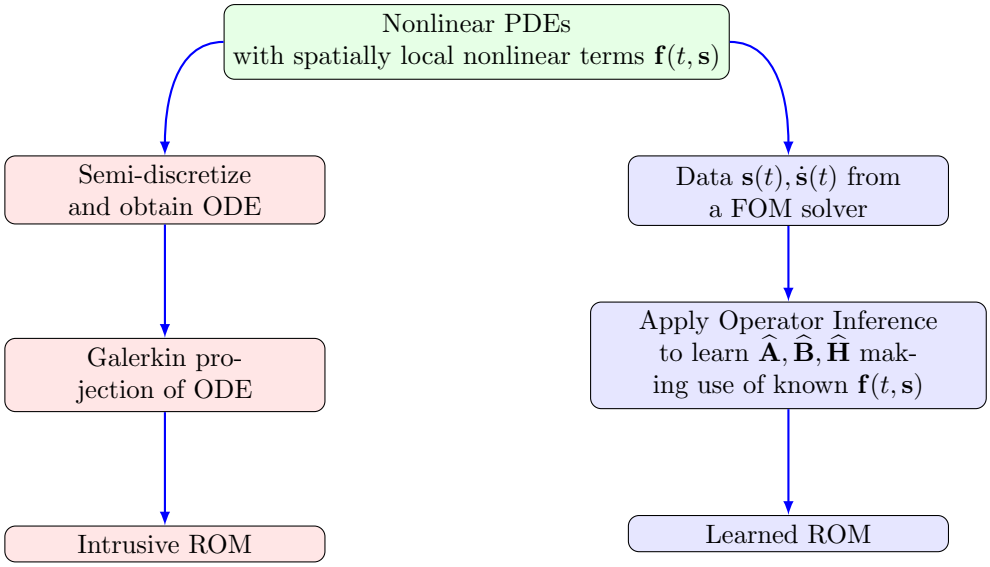


Figure 1: A comparison of a fully intrusive ROM approach with the proposed learning approach.

3.2. Analysis of operator inference framework

We extend the results from [26, Sec 3.2] to show that under certain conditions on the time discretization, the learned operators converge to the intrusively obtained reduced operators. For this, we make two assumptions on the time discretization of the FOM and ROM.

Assumption 1. *The time stepping scheme for the FOM* (2) *is convergent, i.e.,*

$$\max_{i \in \{1, \ldots T/\Delta t\}} \|\mathbf{s}_i - \mathbf{s}(t_i)\|_2 \to 0 \quad \textit{as} \quad \Delta t \to 0.$$

Assumption 2. *The derivatives approximated from projected states,* $\dot{\widehat{\mathbf{s}}}_k$*, converge to* $\frac{d}{dt}\widehat{\mathbf{s}}(t_k)$ *as the discretization time step* $\Delta t \to 0$*, i.e.,*

$$\max_{i \in \{1, \ldots T/\Delta t\}} \|\dot{\widehat{\mathbf{s}}}_i - \frac{d}{dt}\widehat{\mathbf{s}}(t_i)\|_2 \to 0 \quad \textit{as} \quad \Delta t \to 0.$$

Assumption 2 relates to the specific derivative approximation scheme that is used, its approximation order and the time-step size. In this work, we employ a fourth-order time-stepping scheme, see Section 4 for details. Other choices of time discretizations are possible, see e.g., [37, 38, 39] and the references therein.

Theorem 1. *Let Assumptions 1–2 hold and let a basis matrix* $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r] \in \mathbb{R}^{n\times r}$ *be given. Let* $\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{H}}$ *be the intrusively projected ROM operators from* (4)*. If the data matrix* $\widehat{\mathbf{D}} = [\widehat{\mathbf{S}},\ \mathbf{U},\ \widehat{\mathbf{S}} \otimes' \widehat{\mathbf{S}}]$ *has full column rank, then for every* $\varepsilon > 0$*, there exists* $r \leq n$ *and a time step size* $\Delta t > 0$ *such that for the difference between the learned operators* $\widehat{\mathbf{A}}, \widehat{\mathbf{H}}, \widehat{\mathbf{B}}$ *and the (intrusive) projection-based* $\widetilde{\mathbf{A}}, \widetilde{\mathbf{H}}, \widetilde{\mathbf{B}}$*, we have*

$$\|\widehat{\mathbf{A}} - \widetilde{\mathbf{A}}\|_F \leq \varepsilon, \quad \|\widehat{\mathbf{B}} - \widetilde{\mathbf{B}}\|_F \leq \varepsilon, \qquad \|\widehat{\mathbf{H}} - \widetilde{\mathbf{H}}\|_F \leq \varepsilon.$$

Proof. The proof follows the ideas of the proof of [26, Theorem 1]. With the data matrices from (8)–(9), we set $\widehat{\mathbf{D}} = [\widehat{\mathbf{S}},\ \mathbf{U},\ \widehat{\mathbf{S}} \otimes' \widehat{\mathbf{S}}]$ as the data matrix, $\widehat{\mathbf{R}} = \dot{\widehat{\mathbf{S}}} - \widehat{\mathbf{F}}$ as the residual, and $\widehat{\mathbf{O}} = [\widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{H}}]$ are the operators to be inferred, so the operator inference problem (10) can be formulated as

$$\min_{\widehat{\mathbf{O}}}\ \|\widehat{\mathbf{D}}\widehat{\mathbf{O}}^\top - \widehat{\mathbf{R}}^\top\|_F. \tag{13}$$

Denote by $\widetilde{\mathbf{S}} = [\widetilde{\mathbf{s}}_1, \widetilde{\mathbf{s}}_2, \ldots, \widetilde{\mathbf{s}}_k]$ the data matrix that is collected from taking k snapshots of the intrusive ROM (5), and let $\widetilde{\mathbf{D}} = [\widetilde{\mathbf{S}},\ \mathbf{U},\ \widetilde{\mathbf{S}} \otimes' \widetilde{\mathbf{S}}]$ and $\widetilde{\mathbf{R}} = \dot{\widetilde{\mathbf{S}}} - \widetilde{\mathbf{F}}$. With $\widetilde{\mathbf{D}}, \widetilde{\mathbf{R}}$ as data, the projected operators $\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{H}}$ are a solution of the minimization problem (13), and if $\widetilde{\mathbf{D}}$ has full column rank, then the solution is also unique. Next, we observe that $\widehat{\mathbf{S}} = \widetilde{\mathbf{S}} + \Delta\widetilde{\mathbf{S}}$, i.e., the projected FOM data can be interpreted as a perturbation of the ROM-simulated data, and as the dimension r increases, we have $\|\Delta\widetilde{\mathbf{S}}\|_F \to 0$ because of Assumption 1, and the fact that for the limiting case $r = n$ the FOM (2) and ROM (5) are equivalent up to a rotation of coordinates. Since $\widehat{\mathbf{S}} \to \widetilde{\mathbf{S}}$ we also have by the continuity of $\mathbf{f}_r$ that $\widehat{\mathbf{F}} \to \widehat{\mathbf{F}}$. Moreover, Assumption 2 ensures convergence of $\dot{\widehat{\mathbf{s}}}_k \to \frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{s}}(t_k)$ and using again that for the limiting case $r = n$ the FOM (2) and ROM (5) are equivalent up to a rotation of coordinates, we have that $\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{s}}(t_k) \to \frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\mathbf{s}}(t_k)$ as $r \to n$ so $\|\Delta\widetilde{\mathbf{R}}\|_F \to 0$. Taken together we have $\|\widetilde{\mathbf{R}} + \Delta\widetilde{\mathbf{R}}\|_F \to \|\widetilde{\mathbf{R}}\|_F$. Therefore, we have as a limiting result that

$$\min_{\widehat{\mathbf{O}}}\left[\lim_{\substack{r\to n\\ \Delta t\to 0}} \|\widehat{\mathbf{D}}\widehat{\mathbf{O}}^\top - \widehat{\mathbf{R}}^\top\|_F\right] = \min_{\widehat{\mathbf{O}}}\left[\lim_{\substack{r\to n\\ \Delta t\to 0}} \|[\widetilde{\mathbf{D}} + \Delta\widetilde{\mathbf{D}}]\widehat{\mathbf{O}}^\top - [\widetilde{\mathbf{R}} + \Delta\widetilde{\mathbf{R}}]^\top\|_F\right] = \min_{\widehat{\mathbf{O}}} \|\widetilde{\mathbf{D}}\widehat{\mathbf{O}}^\top - \widetilde{\mathbf{R}}^\top\|_F, \tag{14}$$

so the learned operators $\widehat{\mathbf{O}}$ converge to the intrusively projected operators $\widetilde{\mathbf{O}}$. In the pre-asymptotic case, we then get the stated result in the theorem, which also uses the full-rank assumption on $\widehat{\mathbf{D}}$ to deduce $\|\widehat{\mathbf{A}} - \widetilde{\mathbf{A}}\|_F \leq \varepsilon,\ \|\widehat{\mathbf{B}} - \widetilde{\mathbf{B}}\|_F \leq \varepsilon,\ \|\widehat{\mathbf{H}} - \widetilde{\mathbf{H}}\|_F \leq \varepsilon$. □

The theorem shows that the learned ROM operators converge to the projected ROM operators as $r \to n$. However, we do not obtain a convergence rate, and in our practical experience there are examples where the difference in operators might not monotonically decrease for low orders of the ROM, such as the tubular reactor example in Section 4.2. For the other examples in Sections 4.1 and 4.3, the norm did decrease for low orders of the ROM. Most importantly, however, we see in the numerical results in Section 4 that in all cases we can obtain accurate learned ROMs with convergent state-errors for $r \ll n$ via the operator inference framework.

The cost of the proposed framework of operator inference for nonlinear systems is similar to a standard projection-based POD framework when $k < n$, i.e., we have more states than snapshots. Both the projection-based ROM and the learned ROM require computing the POD basis from the snapshot set as well as time integration of r-dimensional ROMs. The methods differ in that the operator inference requires projecting the data $\mathbf{S}, \dot{\mathbf{S}}$ onto the subspace $\mathbf{V}$ (see step 4 of Algorithm 1), forming $\mathbf{S} \otimes' \mathbf{S}$ and then solving a least-squares problem in reduced dimension. The cost scales as $\mathcal{O}(rn)$. In contrast, the projection-based ROM requires projecting the model matrices onto the subspace $\mathbf{V}$, see (4), which also scales as $\mathcal{O}(rn)$. Note, that in that last step we need to form $\mathbf{V}^\top\mathbf{H}(\mathbf{V} \otimes' \mathbf{V})$ which is potentially expensive.

Algorithm 2 Discrete empirical interpolation method [30, Algorithm 1]

Input: $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_m]$.
$p_1 = \arg\max_i(|\mathbf{w}_1(i)|)$; $\mathbf{W}_1 = [\mathbf{w}_1]$; $\mathbb{S}_1 = [e_{p_1}]$; $\rho_1 = [p_1]$
for $j = 2 : m$ **do**
    Solve $\mathbb{S}_{j-1}^\top \mathbf{W}_{j-1} z = \mathbb{S}_{j-1}^\top \mathbf{w}_j$ for z;
    $r_j = \mathbf{w}_j - \mathbf{W}_{j-1} z$; $p_j = \arg\max_i(|r_j(i)|)$;
    $U_j = [\mathbf{W}_{j-1}, \mathbf{w}_j]$; $\mathbb{S}_j = [\mathbb{S}_{j-1}, e_{p_j}]$; $\rho_j = (\rho_{j-1}, p_j)$;
end for
Output: Selection operator $\mathbb{S} = \mathbb{S}_m$

3.3. Empirical interpolation

In this section, we discuss empirical interpolation [28, 30] to accelerate the evaluation of $\mathbf{V}^\top \mathbf{f}(t, \mathbf{V}\widehat{\mathbf{s}})$ in the learned ROM. We employ the discrete empirical interpolation method (DEIM) from [30]. To this end, we approximate

$$\widehat{\mathbf{f}}(t, \mathbf{V}\widehat{\mathbf{s}}(t)) \approx \widehat{\mathbf{f}}_r(t, \mathbf{V}\widehat{\mathbf{s}}(t)) = \mathbf{V}^\top \mathbf{W}(\mathbb{S}^\top \mathbf{W})^{-1} \mathbb{S}^\top \mathbf{f}(t, \mathbf{V}\widehat{\mathbf{s}}(t)). \tag{15}$$

The matrix $\mathbf{W}$ is computed by taking the SVD of the nonlinear snapshot matrix $\mathbf{F}$ in equation (7), and setting $\mathbf{W}$ to the leading m left singular vectors of $\mathbf{F}$. Here, $\mathbb{S}$ is an $n \times m$ matrix obtained by selecting certain columns of the $n \times n$ identity matrix. The choice of the selected entries in $\mathbb{S}$ follows Algorithm 2, which enforces interpolation $\mathbb{S}^\top \mathbf{f}(t, \cdot) = \mathbb{S}^\top [\mathbf{W}(\mathbb{S}^\top \mathbf{W})^{-1} \mathbb{S}^\top \mathbf{f}(t, \cdot)]$ and simultaneously limits the local growth of the spectral norm $\|(\mathbb{S}^\top \mathbf{W})^{-1}\|_2$, as this term occurs in the DEIM error bound [30]. The learned ROM with DEIM interpolation is written as

$$\dot{\widehat{\mathbf{s}}}(t) = \widehat{\mathbf{A}}\widehat{\mathbf{s}}(t) + \widehat{\mathbf{H}}(\widehat{\mathbf{s}}(t) \otimes' \widehat{\mathbf{s}}(t)) + \widehat{\mathbf{f}}_r(t, \widehat{\mathbf{s}}(t)) + \widehat{\mathbf{B}}\mathbf{u}(t), \tag{16}$$

where only $m \ll n$ entries of $\mathbf{f}$ are evaluated, thus achieving computational speedup.

4. Numerical results

In this section, we test the efficiency of the proposed operator inference approach by means of several problems, arising in different areas of science and engineering. We compare the quality of the learned ROM with an intrusive ROM. Here, we use a projection-based POD method. Below we give some information that applies to all our numerical simulations:

- All models are simulated using the routine `ode15s` in MATLAB® with relative error and absolute error tolerances of 10^{-10}.
- In the operator inference framework, we approximate the time derivative $\dot{\widehat{\mathbf{s}}}_i$ with a five-point stencil $\dot{\widehat{\mathbf{s}}}_i \approx (-\widehat{\mathbf{s}}_{i+2} + 8\widehat{\mathbf{s}}_{i+1} - 8\widehat{\mathbf{s}}_{i-1} + \widehat{\mathbf{s}}_{i-2})/(12\Delta t)$, which has fourth-order accuracy. The first two and last two time derivatives are computed using first-order forward and backward Euler approximations, respectively.

We present three examples with increasing complexity to illustrate our approach.

4.1. Chafee-Infante model

We consider the one-dimensional Chafee-Infante equation [40], which is closely related to the Cahn-Hillard equation [41]. The Chafee-Infante model has been studied extensively as a benchmark problem for nonlinear MOR techniques, and has been reduced by means of various methods, see, e.g. [42, 43]. The Chafee-Infante equation is a diffusion-reaction model with the governing equation, boundary and initial conditions as follows:

$$\begin{aligned} &\frac{dv}{dt} + v^3 = \frac{\partial^2 v}{\partial x^2} + u, && (x,t) \in (0,1) \times (0,T], && v(0,t) = u(t), && t \in (0,T), \\ &\frac{\partial v}{\partial x}(1,t) = 0, && t \in (0,T), && v(x,0) = 0, && x \in (0,1). \end{aligned} \tag{17}$$

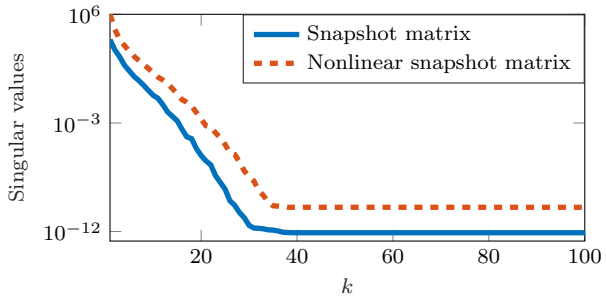


Figure 2: Chafee-Infante example: Decay of singular values of the snapshot matrix and the nonlinear snapshot matrix.

As a quantity of interest, we observe the output at the right boundary, i.e., $y(t) = v(L,t)$. The model (17) contains a cubic nonlinear term, which fits our discussion of spatially local nonlinear terms of Section 2.1. Hence, we can use our proposed non-intrusive approach to construct a learned ROM.

The governing equations are spatially discretized via a finite difference scheme using 500 grid points. We generate snapshots of the discretized state $\mathbf{s}(t)$ via numerical simulations of the semi-discrete system in the time interval $T = [0, 10]$s with time step 10^{-3}. We also collect snapshot data for the input $u(t) = 10(\sin(\pi t) + 1)$. This allows us to build a snapshot matrix $\mathbf{S}$ and input matrix $\mathbf{U}$. Furthermore, we construct a nonlinear snapshot matrix $\mathbf{F}$ from the snapshot matrix $\mathbf{S}$ by applying the nonlinear function $\mathbf{f}$ from equation (3), i.e., $\mathbf{F}(:,i) = \mathbf{f}(\mathbf{S}(:,i), t_i)$ where $(:,i)$ denotes the ith column of a matrix.

In Figure 2, we show the decay of the singular values of the snapshot matrix and nonlinear snapshot matrix, indicating exponential decay for both. By taking the dominant 12 POD basis vectors, we construct an intrusive ROM by explicitly computing the projected matrices, and also employ our non-intrusive framework from Algorithm 1 to obtain a learned ROM. For the operator inference method, we leave out the first 10 recorded snapshots, as the transient response in the first 10 state vectors is very fast. A much smaller time step could solve this issue, but we are often faced with a situation where a data set with a fixed time step is given to us, and we have little control over the time step selection, which is the situation we consider here. We employ DEIM approximation for the reduced nonlinear terms as discussed in Section 3.3. We select DEIM basis vectors for the nonlinear term corresponding to relative singular values up to 10^{-8}.

In Figure 3, we compare the output of interest for the FOM and both ROMs for the same input. Note that the system is simulated for $T = 20$s, thus predicting the output for 10s, i.e., 100% past the training interval. We observe that the learned ROM is more accurate at most time steps. While this appears counter-intuitive at first, it might be possible that unresolved features (sometimes called closure terms) are learned by the ROM. This remains a topic of further investigation.

In Figure 4, we compare the state approximation error of both ROMs. Both POD and our learned ROM are accurate, with the learned ROM being slightly more accurate. We found that after $r = 14$, the non-intrusive approach starts yielding unstable ROMs. This could be due to the fact that the singular values corresponding to higher basis vectors are relatively small (order 10^{-10} in this example). Moreover, since the training data remains the same as r is increased, there can be a situation where there is not enough information in the training data to learn a more accurate ROM that inherits the stability from the FOM. We do not observe instability in non-intrusive ROMs in general. Nevertheless, at this point the ROMs already predict the output with accuracy 10^{-7}, which is sufficient in most practical applications.

4.2. Tubular reactor model

We consider a one-dimensional non-adiabatic tubular reactor model with a single reaction, describing the evolution of the species concentration $\psi(x,t)$ and temperature $\theta(x,t)$. The dynamics of the model are given

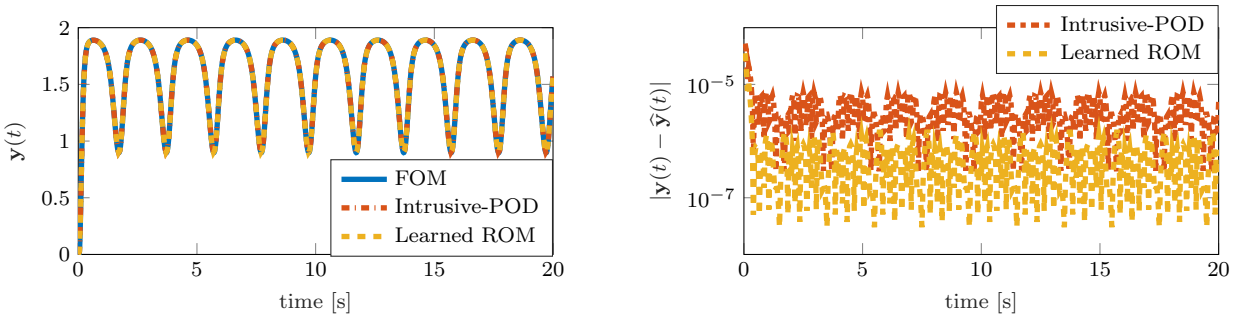


Figure 3: Chafee-Infante example: Output (left) and output errors (right) of the POD intrusive and the learned ROMs of order $r = 12$.

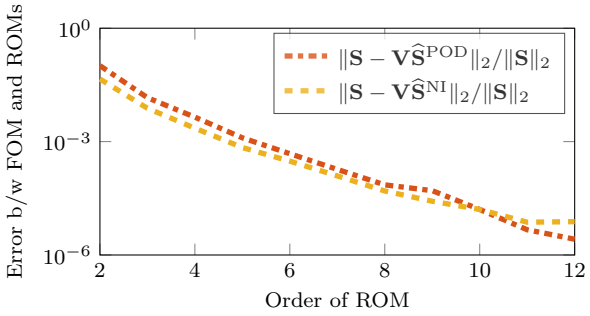


Figure 4: Chafee-Infante example: Convergence of relative state errors with increasing order of the POD intrusive ROMs and the learned ROMs.

by the PDEs [44] as follows:

$$\frac{\partial \psi}{\partial t} = \frac{1}{\mathrm{Pe}}\frac{\partial^2 \psi}{\partial x^2} - \frac{\partial \psi}{\partial x} - \mathcal{D}\mathcal{F}(\psi, \theta; \gamma),$$
$$\frac{\partial \theta}{\partial t} = \frac{1}{\mathrm{Pe}}\frac{\partial^2 \theta}{\partial x^2} - \frac{\partial \theta}{\partial x} - \beta(\theta - \theta_{\mathrm{ref}}) + \mathcal{B}\mathcal{D}\mathcal{F}(\psi, \theta; \gamma),$$

with spatial variable $x \in (0, 1)$, time $t > 0$ and Arrhenius reaction term

$$\mathcal{F}(\psi, \theta; \gamma) = \psi \exp\left(\gamma - \frac{\gamma}{\theta}\right). \tag{19}$$

The parameters are the Damköhler number $\mathcal{D}$, the Péclet number Pe, and the reaction rate γ. We set $\mathcal{D} = 0.167$, $\mathrm{Pe} = 5$, and $\gamma = 25$. The reference temperature is $\theta_{\mathrm{ref}}(x, t) \equiv 1$, and the constants are $\mathcal{B} = 0.5$ and $\beta = 2.5$. Robin boundary conditions are imposed on the left boundary of the domain as follows:

$$\frac{\partial \psi}{\partial x}(0, t) = \mathrm{Pe}(\psi(0, t) - 1), \qquad \frac{\partial \theta}{\partial x}(0, t) = \mathrm{Pe}(\theta(0, t) - 1),$$

and the Neumann boundary conditions on the right boundary are

$$\frac{\partial \psi}{\partial x}(1, t) = 0, \quad \frac{\partial \theta}{\partial x}(1, t) = 0.$$

Moreover, the initial conditions are prescribed as

$$\psi(x, 0) = \psi_0(x), \qquad \theta(x, 0) = \theta_0(x).$$

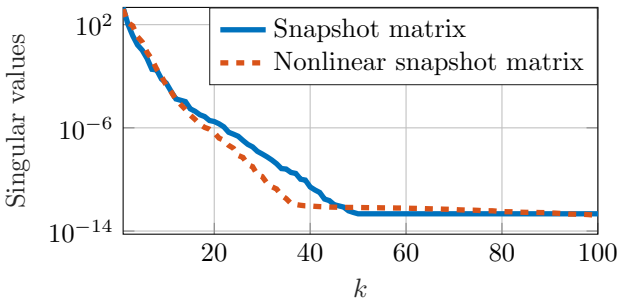


Figure 5: Tubular reactor example: Decay of the singular values of the snapshot matrix and the nonlinear snapshot matrix.

The quantity of interest is the temperature oscillation at the reactor exit, i.e., the quantity

$$y(t) = \theta(x = 1, t).$$

The governing PDE is discretized with a finite difference approximation (see [45] for details) leading to a discretized state $\mathbf{s}(t) \in \mathbb{R}^{198}$ which takes the form

$$\dot{\mathbf{s}}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{f}(\mathbf{s}(t)) + \mathbf{B}. \tag{20}$$

Observe that in this example the input $\mathbf{u}(t)$ is constant, i.e., $\mathbf{u}(t) \equiv 1$. The Arrhenius nonlinear term (19) requires pointwise evaluations (local in space), which fits to our framework presented in Section 3. Thus, we can apply the proposed non-intrusive operator inference framework to obtain a ROM by including PDE-level information about the nonlinear terms into the learning framework. We build the snapshot matrix by collecting snapshots in the time interval $T = (0, 30]$s which are $\delta t = 10^{-3}$ apart.

Figure 5 shows the decay of the singular values of the snapshot matrix $\mathbf{S}$ and the nonlinear snapshot matrix $\mathbf{F}$. We use the leading $r = 10$ POD modes in the projection matrix $\mathbf{V}$. We construct ROMs both via explicitly computing the projected model terms, and via our non-intrusive approach. As in the previous example, we make use of DEIM to efficiently evaluate the reduced nonlinear terms in both ROMs. We select DEIM basis vectors for the nonlinear term corresponding to relative singular values up to 10^{-8}.

Both ROMs are simulated until $T = 60$s, which is 100% longer than the training interval. Figure 6a shows the quantity of interest obtained from the FOM and the two ROMs. Note, that for this set of parameters, the model enters a self-excited limit-cycle oscillation. Figure 6b shows the output error $|\mathbf{y}(t) - \widehat{\mathbf{y}}(t)|$ for the two ROMs. Both models produce similar output errors. This competitiveness of the learned ROM with the projection-based POD model is remarkable, since after $t = 30$s outputs are purely predictive as the ROMs exit the range of training data.

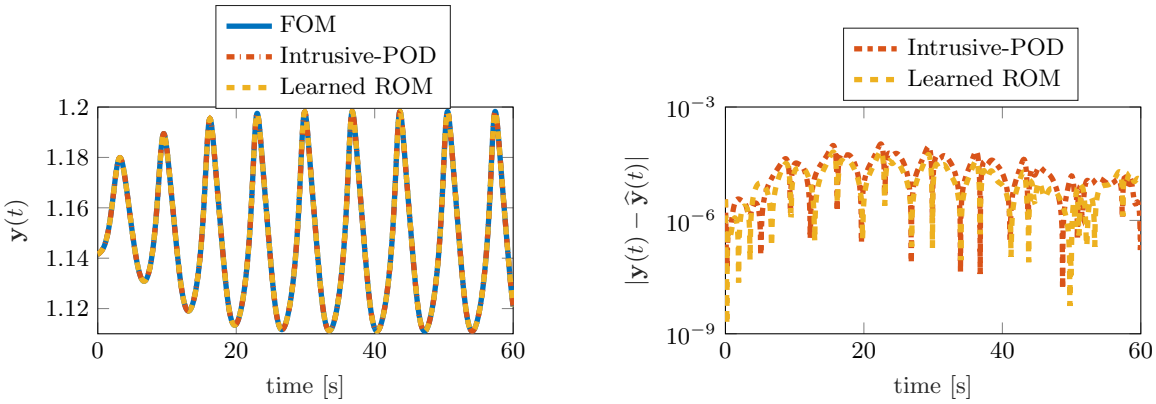
Figure 7 shows the relative error in the approximated states for both ROMs with increasing ROM order. The learned ROM is more accurate than the POD-ROM in most cases. The errors for both models level-off after $r = 18$ at a relative state error of 10^{-7}, which is again accurate enough for most applications, certainly for the temperature field considered in this example.

4.3. Batch Chromatography

In Section 4.3.1 we describe the PDE model and its discretization. Section 4.3.2 presents the structure-preserving ROM generation in the projection-based setting, and Section 4.3.3 describes how we enforce the coupling structure in the non-intrusive learning framework. Section 4.3.4 presents our numerical results.

4.3.1. Problem setup

Chromatography is a separation and purification process used in the chemical and pharmaceutical industry, and we consider one of the simplest chromatography processes, the so-called batch chromatography. For a detailed description of the process, we refer to [46, 47]. The primary principle of the batch chromatography process for binary separation is shown in Figure 8. A mixture of products A and B is injected



(a) Transient response.

(b) Absolute output error.

Figure 6: Tubular reactor example: A comparison of the transient behavior of the projection-based POD ROM and the learned ROM.

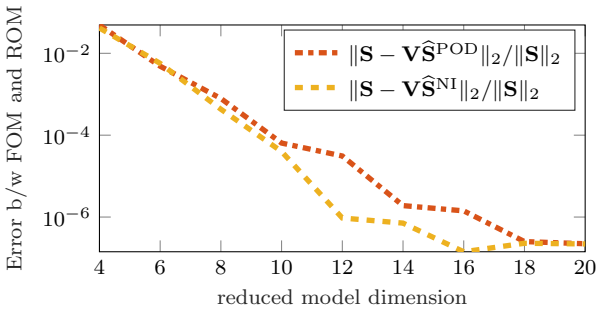


Figure 7: Tubular reactor example: Relative state errors produced by both ROMs, i.e., $\|\mathbf{S}-\mathbf{V}\widehat{\mathbf{S}}^{\mathrm{POD}}\|/\|\mathbf{S}\|$ and $\|\mathbf{S}-\mathbf{V}\widehat{\mathbf{S}}^{\mathrm{NI}}\|/\|\mathbf{S}\|$.

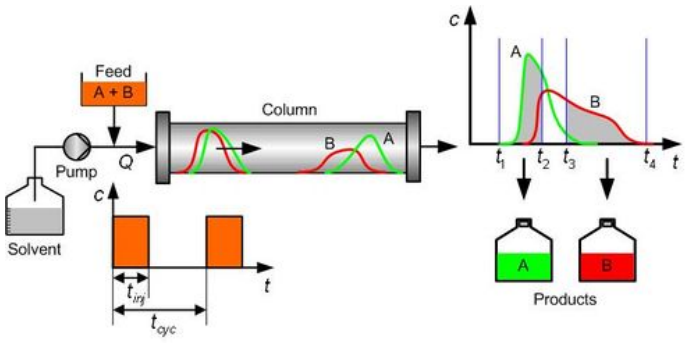


Figure 8: The figure shows a chromatography process, separating a mixture, c.f. [48].

at the inlet of the column and then the feed mixture flows through the column. Since the to-be-separated solutes exhibit different adsorption affinities to the stationary phase, they move at different velocities, and thus separate from each other when exiting the column. At the column outlet, component A, which moves faster, is collected between t_1 and t_2, and later, component B is collected between t_3 and t_4. Here, time t_1 and t_4 are determined by a minimum concentration threshold that the detector can resolve, and time t_2 and t_3 are determined by the purity specifications imposed on the products.

The dynamics of the batch chromatographic column can be described by the following dimensionless

coupled PDE system:

$$\frac{\partial c_i}{\partial t} + \frac{1-\epsilon}{\epsilon}\frac{\partial q_i}{\partial t} + \frac{\partial c_i}{\partial x} - \frac{1}{\mathrm{Pe}}\frac{\partial^2 c_i}{\partial x^2} = 0, \tag{21a}$$

$$\frac{\partial q_i}{\partial t} = \frac{L}{Q/\epsilon A_c}\kappa_i \left(q_i^{Eq} - q_i\right), \tag{21b}$$

where $i \in \{1,2\}$, $x \in (0,L)$ is the spatial variable, and the liquid and solid phase concentrations of the compound i are denoted by c_i and q_i, respectively; the constants ϵ, L are the interstitial liquid velocity, the column porosity, and the column length, respectively; Pe denotes the Péclet number; A_c denotes the cross-sectional area, defined as $\frac{\pi D^2}{4}$, where D is the diameter of the tube. The constant κ_i is the mass-transfer coefficient of component i; moreover, q_i^{Eq} is the adsorption equilibrium concentration calculated by the isotherm equation for the component i, given as follows:

$$q_i^{\mathrm{Eq}} = \frac{H_{i,1}\, c_i}{1 + \sum\limits_{j=1,2} K_{j,1} c_j^f\, c_j} + \frac{H_{i,2}\, c_i}{1 + \sum\limits_{j=1,2} K_{j,2} c_j^f\, c_j}, \tag{22}$$

where $H_{i,1}$ and $H_{i,2}$ are the Henry constants, and $K_{j,1}$ and $K_{j,2}$ the thermodynamic coefficients. Furthermore, the boundary conditions are specified by the Danckwerts relations:

$$\left.\frac{\partial c_i}{\partial x}\right|_{x=0} = \mathrm{Pe}\,(\,c_i|_{x=0} - c_i^{\mathrm{in}}), \qquad \left.\frac{\partial c_i}{\partial x}\right|_{x=L} = 0, \tag{23}$$

where c_i^{in} is the concentration of component i at the inlet of the column. A sigmoid-type injection profile is assumed, which is a differentiable approximation of a rectangular profile, given by

$$c_i^{\mathrm{in}}(t) = \frac{1}{1 + e^{-5(t - t_{\mathrm{inj}})}}, \tag{24}$$

where t_{inj} is the injection period. In addition, we have the following initial conditions:

$$c_i(t=0,x) = q_i(t=0,x) = 0, \quad x \in [0,\ L],\ i \in \{1,2\}. \tag{25}$$

System (21) is a coupled set of PDEs, and more importantly, (21a) involves the derivatives of c_i and q_i. However, we can modify (21a) such that the resulting equation does not have a $\partial q_i/\partial t$ term. For this, we set $\epsilon_c := \dfrac{\epsilon - 1}{\epsilon}$ and insert (21b) by into (21a), yielding

$$\frac{\partial c_i}{\partial t} + \frac{\partial c_i}{\partial x} - \frac{1}{\mathrm{Pe}}\frac{\partial^2 c_i}{\partial x^2} = \epsilon_c \kappa_i \left(q_i^{Eq} - q_i\right), \tag{26a}$$

$$\frac{\partial q_i}{\partial t} = \kappa_i \left(q_i^{\mathrm{Eq}} - q_i\right). \tag{26b}$$

A finite volume discretization of the governing equations yields a discretized model of the form:

$$\begin{bmatrix} \dot{\mathbf{c}}_1 \\ \dot{\mathbf{q}}_1 \\ \dot{\mathbf{c}}_2 \\ \dot{\mathbf{q}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{A}_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{q}_1 \\ \mathbf{c}_2 \\ \mathbf{q}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \\ \mathbf{B} \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} \epsilon_c \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{f}_1(\mathbf{c}_1, \mathbf{q}_1, \mathbf{c}_2, \mathbf{q}_2) \\ \mathbf{f}_2(\mathbf{c}_1, \mathbf{q}_1, \mathbf{c}_2, \mathbf{q}_2) \end{bmatrix}, \tag{27}$$

where $\mathbf{c}_1, \mathbf{q}_1, \mathbf{c}_2, \mathbf{q}_2 \in \mathbb{R}^n$, $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^n$, and n is the number of degrees of freedom in the discretized field. The term $\mathbf{f}_i(\mathbf{c}_1, \mathbf{q}_1, \mathbf{c}_2, \mathbf{q}_2) \in \mathbb{R}^n$ is the evaluation of the nonlinear term $\kappa_i \left(q_i^{\mathrm{Eq}} - q_i\right)$ on the right-hand side of (26b) on the spatial grid. Note that this nonlinear term involves a rational function, see (22), which again translates to a nonlinear term that is local in space, and hence the nonlinear function fits the setting described in Section 2.1.

4.3.2. Maintaining the coupling structure: Projection-based ROM

Classically, to obtain the basis for POD, one would stack all variables in one vector, i.e., $[\mathbf{c}_1^\top, \mathbf{q}_1^\top, \mathbf{c}_2^\top, \mathbf{q}_2^\top]^\top$ and determine the common subspace $\mathbf{V}$ by taking the SVD of that data. However, by doing so, we observe that both methods produce unstable ROMs. This is potentially due to the fact that the governing PDEs are highly coupled and the resulting ROMs do not preserve the coupling topology structure. We present an approach that uses the knowledge about the topology structure of a coupled system in the ROM.

To preserve the coupled topology structure, we first compute a basis for each subsystem, i.e., $\mathbf{c}_i \approx \mathbf{V}_{\mathbf{c}_i}\widehat{\mathbf{c}}_i$ and $\mathbf{q}_i \approx \mathbf{V}_{\mathbf{q}_i}\widehat{\mathbf{q}}_i$, $i \in \{1, 2\}$. Next, we construct a projection matrix with a block structure, i.e., we approximate

$$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{q}_1 \\ \mathbf{c}_2 \\ \mathbf{q}_2 \end{bmatrix} \approx \begin{bmatrix} \mathbf{V}_{\mathbf{c}_1} & & & \\ & \mathbf{V}_{\mathbf{q}_1} & & \\ & & \mathbf{V}_{\mathbf{c}_2} & \\ & & & \mathbf{V}_{\mathbf{q}_2} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{c}}_1 \\ \widehat{\mathbf{q}}_1 \\ \widehat{\mathbf{c}}_2 \\ \widehat{\mathbf{q}}_2 \end{bmatrix}. \tag{28}$$

With the above projection matrix $\mathbf{V}$, a projection-based ROM preserves this structure, i.e.,

$$\begin{bmatrix} \dot{\widehat{\mathbf{c}}}_1 \\ \dot{\widehat{\mathbf{q}}}_1 \\ \dot{\widehat{\mathbf{c}}}_2 \\ \dot{\widehat{\mathbf{q}}}_2 \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{A}}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \widehat{\mathbf{A}}_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{c}}_1 \\ \widehat{\mathbf{q}}_1 \\ \widehat{\mathbf{c}}_2 \\ \widehat{\mathbf{q}}_2 \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{B}}_1 \\ 0 \\ \widehat{\mathbf{B}}_2 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} \epsilon_c \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \widehat{\mathbf{f}}_1(\widehat{\mathbf{c}}_1, \widehat{\mathbf{q}}_1, \widehat{\mathbf{c}}_2, \widehat{\mathbf{q}}_2) \\ \widehat{\mathbf{f}}_2(\widehat{\mathbf{c}}_1, \widehat{\mathbf{q}}_1, \widehat{\mathbf{c}}_2, \widehat{\mathbf{q}}_2) \end{bmatrix}, \tag{29}$$

where $\widehat{\mathbf{c}}_1, \widehat{\mathbf{q}}_1, \widehat{\mathbf{c}}_2, \widehat{\mathbf{q}}_2 \in \mathbb{R}^r$, $\widehat{\mathbf{A}}_1, \widehat{\mathbf{A}}_2 \in \mathbb{R}^{r\times r}$, $\widehat{\mathbf{B}}_1, \widehat{\mathbf{B}}_2 \in \mathbb{R}^r$. Note, that the dimension of the coupled ROM (29) is $4r$, however, it has sparse structure, e.g., the linear system matrix has only $2r^2$ nonzero entries and the linear input matrix has only $2r$ nonzero entries. Moreover, it is also possible to use different ROM dimensions for each of the four physical states. Thus, enforcing the coupling structure does not necessarily lead to drastically more expensive ROMs.

Remark 2. *Maintaining the coupling structure in a ROM of the aggregate system is important for physical interpretability, and can have numerical implications as well [49, 50, 51, 52]. For instance, structure-preserving ROMs can lead to convergent feedback controllers for coupled systems [53] and have implications on stability. We observe in our numerical experiments on the batch chromatography example, that unstable ROMs arise when we do not preserve this structure. Conversely, when the coupling structure of the FOM is maintained in the ROMs, we obtain stable ROMs. A theoretical justification of this observation for nonlinear systems remains an open problem, though, for linear systems, stability analysis for coupled systems is performed in [51].*

4.3.3. Maintaining the coupling structure: Learned ROM

Having derived the form of the projection-based structure-preserving ROM leads to a strategy to enforce the coupling structure in the operator inference framework as follows. First, we compute the block-diagonal projection matrix $\mathbf{V}$ from (28). Second, we collect projected state and time-derivative data:

$$\widehat{\mathbf{S}} = \begin{bmatrix} \mathbf{V}_{\mathbf{c}_1}^\top \mathbf{C}_1 \\ \mathbf{V}_{\mathbf{q}_1}^\top \mathbf{Q}_1 \\ \mathbf{V}_{\mathbf{c}_2}^\top \mathbf{C}_2 \\ \mathbf{V}_{\mathbf{q}_2}^\top \mathbf{Q}_2 \end{bmatrix} =: \begin{bmatrix} \widehat{\mathbf{C}}_1 \\ \widehat{\mathbf{Q}}_1 \\ \widehat{\mathbf{C}}_2 \\ \widehat{\mathbf{Q}}_2 \end{bmatrix}, \qquad \dot{\widehat{\mathbf{S}}} = \begin{bmatrix} \mathbf{V}_{\mathbf{c}_1}^\top \dot{\mathbf{C}}_1 \\ \mathbf{V}_{\mathbf{q}_1}^\top \dot{\mathbf{Q}}_1 \\ \mathbf{V}_{\mathbf{c}_2}^\top \dot{\mathbf{C}}_2 \\ \mathbf{V}_{\mathbf{q}_2}^\top \dot{\mathbf{Q}}_2 \end{bmatrix} =: \begin{bmatrix} \dot{\widehat{\mathbf{C}}}_1 \\ \dot{\widehat{\mathbf{Q}}}_1 \\ \dot{\widehat{\mathbf{C}}}_2 \\ \dot{\widehat{\mathbf{Q}}}_2 \end{bmatrix}. \tag{30}$$

The input matrix $\mathbf{U}$ is assembled as given in (6) and the nonlinear term matrix $\mathbf{F}$ is built in a similar way as given in (7). We then solve *separate* least-squares problems of the form

$$\min_{\widehat{\mathbf{A}}_i, \widehat{\mathbf{B}}_i} \|\dot{\widehat{\mathbf{C}}}_i - \epsilon_c \mathbf{V}_{\mathbf{c}_i}^\top \mathbf{F} - \widehat{\mathbf{A}}_i \widehat{\mathbf{C}}_i - \widehat{\mathbf{B}}_i \mathbf{U}\|_F, \quad i \in \{1, 2\}. \tag{31}$$

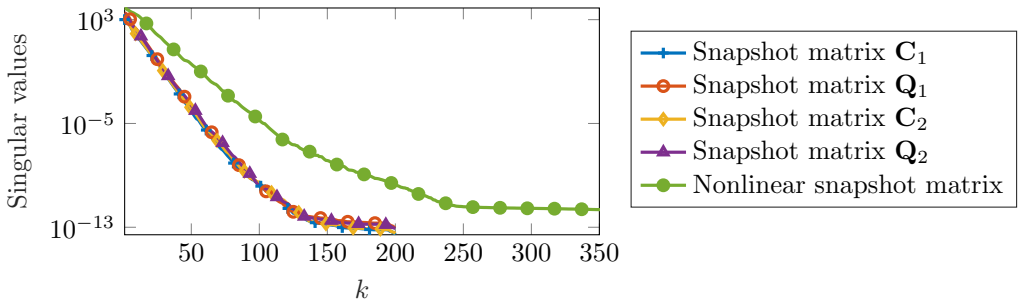


Figure 9: Batch chromatography example: Decay of singular values of the collected snapshots for all four components separately and the nonlinear snapshot matrix.

Recall the structure of the projection-based ROM in (29). For $\dot{\widehat{\mathbf{q}}}_i$, $i \in \{1, 2\}$, there are no linear terms on the right hand side, and so there is no need to solve a least-squares problem. The expressions for $\dot{\widehat{\mathbf{q}}}_i$ are

$$\dot{\widehat{\mathbf{q}}}_i = \mathbf{f}_i(\widehat{\mathbf{c}}_1, \widehat{\mathbf{q}}_1, \widehat{\mathbf{c}}_2, \widehat{\mathbf{q}}_2) = \mathbf{V}_{\mathbf{q}_i}^\top \mathbf{f}_i(\mathbf{V}_{\mathbf{c}_1}\mathbf{c}_1, \mathbf{V}_{\mathbf{q}_1}\mathbf{q}_1, \mathbf{V}_{\mathbf{c}_2}\mathbf{c}_2, \mathbf{V}_{\mathbf{q}_2}\mathbf{q}_2).$$

4.3.4. Results

For the numerical experiment, we choose the model parameters as follows: $\epsilon = 0.4$, $L = 10.5$, Pe $= 2000$, $D = 2.6$, $\kappa_1 = \kappa_2 = 0.1$, $H_{1,1} = 3.728, H_{1,2} = 0.3, H_{2,1} = 2.688, H_{2,2} = 0.1$, $K_{1,1} = 46.6, K_{1,2} = 33.6, K_{2,1} = 3000, K_{2,2} = 1000$, $Q = 0.1018$, $c_1^f = c_2^f = 2.9 \times 10^{-3}$, and $t_{\text{inj}} = 1.3$s. For spatial discretization we choose $n = 400$ (thus the overall state is $4n = 1600$). The model is simulated until $T = 10$s and snapshots are collected every $\delta t = 5 \times 10^{-5}$s. Such a fine time stepping allows us to obtain an accurate approximation of the time derivative; however, we take every 100th snapshot of the state and time derivative while inferring the operators in the least squares procedure. Figure 9 shows the decay of the singular values of each component, showing a rather slow decay, which can be expected due to the transport nature of the problem. Next, we compute ROMs via intrusive POD and our non-intrusive approach with the same matrix $\mathbf{V}$ as shown in (28), where each component is reduced to order $r = 22$. As in the previous two examples, we incorporate a DEIM approximation of the reduced nonlinear terms for both ROMs. We select DEIM basis vectors for the nonlinear term corresponding to relative singular values up to 10^{-10}. Figure 10, left, shows the quantity of interest obtained from the time-domain simulations of the FOM and the two ROMs. Figure 10, right, shows the (mean) relative error of the two outputs. The results indicate that the non-intrusive and intrusive approaches yield ROMs of comparative accuracy. Figure 11 shows the relative error in the approximated states for both ROMs with increasing ROM order. For ROM orders of $r < 30$, both the intrusive POD ROM and the learned ROMs perform similarly. However, as r increases further, the error for the non-intrusive models does not decrease as favorably as for POD intrusive models. This could be due to the condition number of the operator inference problem (11). Adding regularization to the least-squares problem could improve this as seen in [54] and also using data from several different simulations, similar to [26, Sec 3.6].

5. Conclusions

We have presented a data-driven model reduction method that non-intrusively learns reduced models from data collected of the full model states. The presented method applies to non-polynomial nonlinear models that are spatially local, and extends previous results for the polynomial case [26]. The method exploits the spatially local structure of the nonlinear terms to generate nonlinear snapshots from state data without requiring access to the discretization of the nonlinear term explicitly. Moreover, we presented a convergence result that shows that under mild assumptions on the time stepping scheme, and if the time step of the sampled data is sufficiently small, the non-intrusively learned reduced models converge to the same reduced models as obtained with intrusive model reduction methods. The numerical results show that the learned reduced models provide accurate predictions when compared to the full-model predictions.

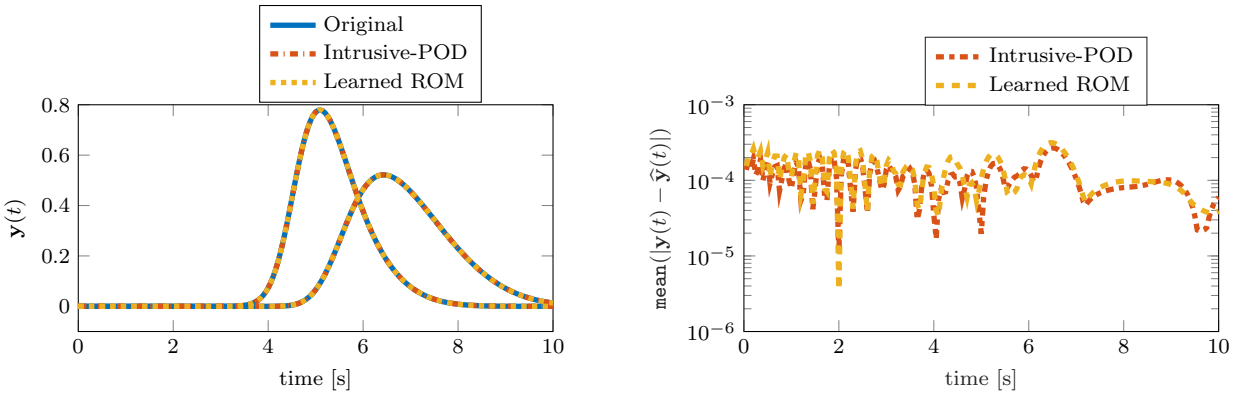


Figure 10: Batch chromatography example: A comparison of the POD intrusive model with the learned model of order $r = 4\times 22$.

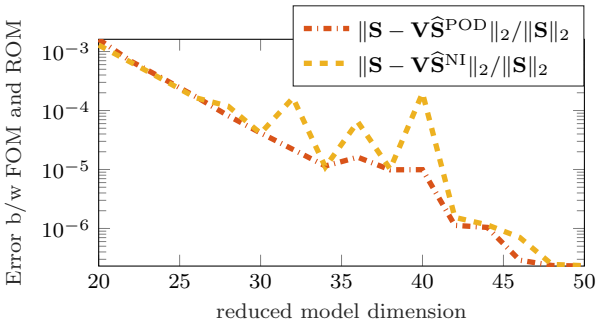


Figure 11: Batch Chromatography example: Relative errors $\|\mathbf{S} - \mathbf{V}\widehat{\mathbf{S}}^{\mathrm{POD}}\|/\|\mathbf{S}\|$ and $\|\mathbf{S} - \mathbf{V}\widehat{\mathbf{S}}^{\mathrm{NI}}\|/\|\mathbf{S}\|$.

The three numerical experiments with reactive flows and chemical purification processes demonstrate that the method works well for strongly nonlinear systems, and further information in form of known structure can also be embedded into the learning framework. Future research directions motivated by this work are: addressing the effect of measurement noise (from real data) on the learning procedure and ultimately the learned ROMs; developing a suitable regularization strategy (possibly problem dependent) that could improve the learned ROMs accuracy and stability; and including additional information, such as physical constraints in the least-squares problem as suggested by [55].

Acknowledgments

This work was supported in part by the US Air Force Center of Excellence on Multi-Fidelity Modeling of Rocket Combustor Dynamics award FA9550-17-1-0195, the Air Force Office of Scientific Research MURI on managing multiple information sources of multi-physics systems awards FA9550-15-1-0038 and FA9550-18-1-0023, and the AEOLUS center under US Department of Energy Applied Mathematics MMICC award DE-SC0019303. The fourth author was partially supported by the US Department of Energy, Office of Advanced Scientific Computing Research, Applied Mathematics Program, DOE Award DESC0019334 and the National Science Foundation under Grant No. 1901091.

References

[1] P. Benner, A. Cohen, M. Ohlberger, K. Willcox (Eds.), Model Reduction and Approximation: Theory and Algorithms, Computational Science & Engineering, SIAM Publications, Philadelphia, PA, 2017. doi:10.1137/1.9781611974829.

[2] A. Quarteroni, G. Rozza (Eds.), Reduced Order Methods for Modeling and Computational Reduction, Vol. 9 of MS & A, Springer, Berlin, Heidelberg, New York, 2014. doi:10.1007/978-3-319-02090-7_6.

[3] A. J. Mayo, A. C. Antoulas, A framework for the solution of the generalized realization problem, Linear Algebra Appl. 425 (2-3) (2007) 634–662.

[4] A. Ionita, A. Antoulas, Data-driven parametrized model reduction in the Loewner framework, SIAM J. Sci. Comput. 36 (3) (2014) A984–A1007. doi:10.1137/130914619.

[5] B. Peherstorfer, S. Gugercin, K. Willcox, Data-driven reduced model construction with time-domain Loewner models, SIAM J. Sci. Comput. 39 (5) (2017) A2152–A2178.

[6] P. Schulze, B. Unger, C. Beattie, S. Gugercin, Data-driven structured realization, Linear Algebra Appl. 537 (2018) 250–286.

[7] A. C. Ionita, A. C. Antoulas, Matrix pencils in time and frequency domain system identification, in: Developments in Control Theory Towards Global Control, Control, Robotics and Sensors, Institution of Engineering and Technology, 2012, pp. 79–88.

[8] S.-Y. Kung, A new identification and model reduction algorithm via singular value decomposition, in: Proc. 12th Asilomar Conf. Circuits, Syst. Comput., Pacific Grove, CA, 1978, pp. 705–714.

[9] Z. Ma, S. Ahuja, C. W. Rowley, Reduced-order models for control of fluids using the eigensystem realization algorithm, Theoretical Comput. Fluid Dyn. 25 (1-4) (2011) 233–247.

[10] B. Kramer, S. Gugercin, Tangential interpolation-based eigensystem realization algorithm for MIMO systems, Math. Comput. Model. Dyn. Syst. 22 (4) (2016) 282–306. doi:10.1080/13873954.2016.1198389.

[11] B. Kramer, A. Gorodetsky, System identification via CUR-factored Hankel approximation, SIAM J. Sci. Comput. 40 (2) (2018) A848–A866. doi:10.1137/17M1137632.

[12] B. Gustavsen, A. Semlyen, Rational approximation of frequency domain responses by vector fitting, IEEE Trans. Power Del. 14 (3) (1999) 1052–1061. doi:10.1109/61.772353.

[13] Z. Drmač, S. Gugercin, C. Beattie, Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation, SIAM J. Sci. Comput. 37 (2) (2015) A625–A652. doi:10.1137/140961511.

[14] A. C. Antoulas, I. V. Gosea, A. C. Ionita, Model reduction of bilinear systems in the Loewner framework, SIAM J. Sci. Comput. 38 (5) (2016) B889–B916.

[15] I. V. Gosea, A. C. Antoulas, Data-driven model order reduction of quadratic-bilinear systems, Numer. Lin. Alg. Appl. 25 (6) (2018) e2200.

[16] S. L. Brunton, J. L. Proctor, J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl Acad. Sci. 113 (15) (2016) 3932–3937.

[17] S. H. Rudy, S. L. Brunton, J. L. Proctor, J. N. Kutz, Data-driven discovery of partial differential equations, Sci. Adv. 3 (4) (2017) e1602614.

[18] H. Schaeffer, G. Tran, R. Ward, Extracting sparse dynamics from limited data, SIAM J. Appl. Math. 78 (6) (2018) 3279–3295.

[19] C. Rowley, I. Mezić, S. Bagheri, P. Schlatter, D. Henningson, Spectral analysis of nonlinear flows, J. Fluid Mech. 641 (2009) 115–127.

[20] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28.

[21] M. O. Williams, I. G. Kevrekidis, C. W. Rowley, A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition, J. Nonlinear Sci. 25 (6) (2015) 1307–1346.

[22] M. A. Khodkar, P. Hassanzadeh, Data-driven reduced modelling of turbulent Rayleigh-Bènard convection using DMD-enhanced fluctuation-dissipation theorem, J. Fluid Mech. 852 (2018) R3. doi:10.1017/jfm.2018.586.

[23] X. Xie, M. Mohebujjaman, L. G. Rebholz, T. Iliescu, Data-driven filtered reduced order modeling of fluid flows, SIAM J. Sci. Comput. 40 (3) (2018) B834–B857.

[24] M. Couplet, C. Basdevant, P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, J. Comput. Phys. 207 (1) (2005) 192–220. doi:https://doi.org/10.1016/j.jcp.2005.01.008.

[25] L. Cordier, B. A. El Majd, J. Favier, Calibration of POD reduced-order models using Tikhonov regularization, Internat. J. Numer. Methods in Fluids 63 (2) (2010) 269–296.

[26] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Comp. Meth. Appl. Mech. Eng. 306 (2016) 196–215.

[27] E. Qian, B. Kramer, A. Marques, K. Willcox, Transform & learn: A data-driven approach to nonlinear model reduction, in: AIAA Aviation and Aeronautics Forum and Exposition, Dallas, TX, 2019. doi:10.2514/6.2019-3707.

[28] M. Barrault, Y. Maday, N. C. Nguyen, A. T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, C. R. Math. Acad. Sci. Paris 339 (9) (2004) 667–672.

[29] M. A. Grepl, Y. Maday, N. C. Nguyen, A. T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, ESAIM: Math. Model. Numer. Anal. 41 (3) (2007) 575–605.

[30] S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764. doi:10.1137/090766498.

[31] J. L. Lumley, The structure of inhomogeneous turbulent flows, Atmospheric Turbulence and Radio Wave Propagation (1967) 166–177.

[32] L. Sirovich, Turbulence and the dynamics of coherent structures. I-coherent structures, Quarterly of Applied Mathematics 45 (1987) 561–571.

[33] P. Holmes, J. L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge Monographs on Mechanics, Cambridge University Press, 1996.

[34] P. Astrid, S. Weiland, K. Willcox, T. Backx, Missing point estimation in models described by proper orthogonal decomposition, IEEE Trans. Autom. Control 53 (10) (2008) 2237–2251.

[35] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows, J. Comput. Phys. 242 (2013) 623–647.

[36] N. Nguyen, A. T. Patera, J. Peraire, A best points interpolation method for efficient approximation of parametrized functions, Internat. J. Numer. Methods Engrg. 73 (4) (2008) 521–543.

[37] J. R. Martins, J. T. Hwang, Review and unification of methods for computing derivatives of multidisciplinary computational models, AIAA J. 51 (11) (2013) 2582–2599.

[38] I. Knowles, R. J. Renka, Methods for numerical differentiation of noisy data, Electron. J. Differ. Equ 21 (2014) 235–246.

[39] R. Chartrand, Numerical differentiation of noisy, nonsmooth, multidimensional data, in: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), IEEE, 2017, pp. 244–248.

[40] N. Chafee, E. F. Infante, A bifurcation problem for a nonlinear partial differential equation of parabolic type, Applicable Analysis 4 (1) (1974) 17–37.

[41] X. Chen, Generation, propagation, and annihilation of metastable patterns, J. Differential Equations 206 (2) (2004) 399–437.

[42] P. Benner, T. Breiten, Two-sided projection methods for nonlinear model order reduction, SIAM J. Sci. Comput. 37 (2) (2015) B239–B260. doi:10.1137/14097255X.

[43] P. Benner, P. Goyal, S. Gugercin, $\mathcal{H}_2$-quasi-optimal model order reduction for quadratic-bilinear control systems, SIAM J. Matrix Anal. Appl. 39 (2) (2018) 983–1032. doi:10.1137/16M1098280.

[44] R. F. Heinemann, A. B. Poore, Multiplicity, stability, and oscillatory dynamics of the tubular reactor, Chemical Eng. Sci. 36 (8) (1981) 1411–1419.

[45] Y. B. Zhou, Model reduction for nonlinear dynamical systems with parametric uncertainties, Ph.D. thesis, Massachusetts Institute of Technology (2012).

[46] G. Guiochon, A. Felinger, D. G. Shirazi, Fundamentals of Preparative and Nonlinear Chromatography, Elsevier, 2006.

[47] Y. Zhang, L. Feng, S. Li, P. Benner, Accelerating PDE constrained optimization by the reduced basis method: application to batch chromatography, Internat. J. Numer. Methods Engrg. 104 (11) (2015) 983–1007. doi:10.1002/nme.4950.

[48] The MORwiki Community, Batch chromatography, MORwiki – Model Order Reduction Wiki (2018). URL `http://modelreduction.org/index.php/Batch_Chromatography`

[49] P. Benner, L. Feng, Model order reduction for coupled problems, Applied and Computational Mathematics: An International journal 14 (1) (2015) 3–22.
URL `http://acmij.az/view.php?lang=az&menu=journal&id=372`

[50] B.-S. Liao, Z. Bai, W. Gao, The important modes of subsystems: A moment-matching approach, Internat. J. Numer. Methods Engrg. 70 (13) (2007) 1581–1597.

[51] T. Reis, T. Stykel, Stability analysis and model order reduction of coupled systems, Math. Comput. Model. Dyn. Syst. 13 (5) (2007) 413–436.

[52] T. Reis, T. Stykel, A survey on model reduction of coupled systems, in: W. H. A. Schilders, H. A. van der Vorst, J. Rommes (Eds.), Model Order Reduction: Theory, Research Aspects and Applications, Springer-Verlag, 2008, pp. 133–155.

[53] B. Kramer, Model reduction for control of a multiphysics system: Coupled Burgers' equation, in: American Control Conference (ACC), IEEE, 2016, pp. 6146–6151. doi:10.1109/ACC.2016.7526635.

[54] R. Swischuk, B. Kramer, C. Huang, K. Willcox, Learning physics-based reduced-order models for a single-injector combustion process, AIAA J. 58:6 (2020) 2658–2672. doi:https://doi.org/10.2514/1.J058943.

[55] J.-C. Loiseau, S. L. Brunton, Constrained sparse Galerkin regression, J. Fluid Mech. 838 (2018) 42–67. doi:10.1017/jfm.2017.823.