

# Operator inference of non-Markovian terms for learning reduced models from partially observed state trajectories

Wayne Isaac Tan Uy and Benjamin Peherstorfer\*

March 3, 2021

This work introduces a non-intrusive model reduction approach for learning reduced models from partially observed state trajectories of high-dimensional dynamical systems. The proposed approach compensates for the loss of information due to the partially observed states by constructing non-Markovian reduced models that make future-state predictions based on a history of reduced states, in contrast to traditional Markovian reduced models that rely on the current reduced state alone to predict the next state. The core contributions of this work are a data sampling scheme to sample partially observed states from high-dimensional dynamical systems and a formulation of a regression problem to fit the non-Markovian reduced terms to the sampled states. Under certain conditions, the proposed approach recovers from data the very same non-Markovian terms that one obtains with intrusive methods that require the governing equations and discrete operators of the high-dimensional dynamical system. Numerical results demonstrate that the proposed approach leads to non-Markovian reduced models that are predictive far beyond the training regime. Additionally, in the numerical experiments, the proposed approach learns non-Markovian reduced models from trajectories with only 20% observed state components that are about as accurate as traditional Markovian reduced models fitted to trajectories with 99% observed components.

**Keywords:** scientific machine learning, non-intrusive model reduction, dynamical systems, non-Markovian reduced models, operator inference, partial observations

## 1. Introduction

There is an increasing interest in methods to learn reduced models of high-dimensional dynamical systems from data. Such non-intrusive—scientific-machine-learning—model reduction approaches are applicable even if little is known about the governing equations and instead mostly data are available. This differs from traditional intrusive model reduction that is typically a rather manual process requiring extensive expertise and full knowledge of the high-dimensional governing equations and their discretizations [39, 6, 37, 20, 18]. This work proposes a non-intrusive model reduction method to learn reduced models from partially observed state trajectories, rather than fully observed states, of high-dimensional systems. To compensate for the information lost due to the partially observed states, the proposed reduced models make future-state predictions based on a time history of the reduced states and thus describe non-Markovian dynamics

---

\*{wayne.uy,pehersto}@cims.nyu.edu, Courant Institute of Mathematical Sciences, New York University, New York, NY 10012

[16, 11, 48]. Notice that many traditional reduced models from non-/intrusive model reduction are Markovian and thus make future-state predictions based on the current reduced state alone [39, 6, 37, 20, 18]. The two core contributions of this work are a data sampling scheme and a formulation of a regression problem to learn the non-Markovian terms of the proposed reduced models from partially observed state trajectories. Under certain conditions, in particular if the high-dimensional system dynamics are linear, the proposed data sampling scheme together with the proposed regression problem guarantees recovery of the very same non-Markovian reduced models that one would obtain with intrusive projection-based methods; note that the latter require the high-dimensional system operators in either assembled form or implicitly via methods that provide the action of the high-dimensional operators on vectors in contrast to the proposed approach. If the high-dimensional system dynamics have nonlinear state dependencies, then the proposed non-Markovian reduced models are approximations of the models obtained with intrusive methods.

There is a growing body of literature on learning reduced models from data. If frequency-response or impulse-response measurements are available, then widely used approaches for learning dynamical-system models include the Loewner approach [1, 26, 29, 4, 2, 17, 22], vector fitting [19, 15], and eigensystem realization [23, 24]. With respect to learning from time-domain state trajectories, dynamic mode decomposition (DMD) [44, 43, 38, 51, 25, 58] has been widely adopted which fits linear dynamical systems in the  $L_2$  norm. Tools from sparse regression and compressive sensing have been applied in [41, 8, 42] to extract sparse representations of governing equations from data. These methods focus on learning the high-dimensional dynamical system while our objective is to learn reduced models that reduce the computational costs of many-query applications. One approach that explicitly targets reduced models is operator inference [35], which learns reduced models of nonlinear dynamical systems from projected state trajectories; see also [5, 30, 50, 33, 34]. Together with the re-projection data sampling scheme, which alternates between querying the full system and performing projection, operator inference guarantees recovery of the very same reduced models that would be obtained with intrusive model reduction for nonlinear polynomial systems [33]. There are also *a posteriori* error estimators available for models learned with operator inference from data of high-dimensional linear dynamical systems [52]. The work on Lift & Learn introduced in [36] proposes a non-intrusive model reduction approach that is applicable to dynamical systems with nonlinear terms beyond polynomials.

This work builds on operator inference to learn non-Markovian reduced models from partially observed states. The re-projection scheme [33] is extended to generate data for learning non-Markovian operators. There are other methods for learning non-Markovian models from data. Closest to our work is the use of time-delay coordinates in [13, 9] to generalize DMD to non-Markovian models if under-resolved and incomplete dynamics are observed; however, our focus is on learning polynomial systems rather than DMD models and to recover the same non-Markovian dynamics that one would obtain with intrusive projection approaches in certain situations. There are also model reduction methods for fitting reduced models with delay to data sampled from high-dimensional dynamical systems that include a time delay term [3, 45, 46]. In contrast, the high-dimensional systems that we consider do not necessarily have time-delay terms but rather we introduce non-Markovian (time-delay) terms in our reduced models to compensate for partial state information. Widely used non-Markovian models in nonlinear system identification [7] and time series analysis [47] are based on the nonlinear autoregressive moving average model with exogenous inputs (NARMAX) and its linear counterparts such as the autoregressive moving average model (ARMA) and the autoregressive moving average model with exogenous inputs (ARMAX). Hidden Markov models [60] seek to quantify the uncertainty on the unobserved Markovian states given measurements on observed states which are non-Markovian. Non-Markovian dynamics also play an important role in molecular dynamics simulations [54, 55, 49]. Another area of model reduction and data-fit modeling where non-Markovian models have been investigated is in the fluids and reduced modeling community. There, closure modeling aims to capture the discrepancy in the low-dimensional approximation due to truncation; see for example large eddy simulation [40] and closure models for reduced systems obtained via proper orthogonal decomposition [57]. More recently, data-driven closure for reduced models [32, 56, 53, 28, 31, 59] have been proposed, many of which are based on time-delay coordinates and the Mori-Zwanzig formalism; note that the works [32, 56, 53] consider intrusive settings

where the full system and/or its discretization are used to construct the closure model. In contrast, the goal of this work is learning models from data, rather than deriving a closure. Another key distinction is that we derive non-Markovian models explicitly to compensate for the limited information due to the partially observed state trajectories, rather than closing reduced models and compensating for other data limitations.

This manuscript is organized as follows. Section 2 discusses preliminaries on intrusive model reduction for nonlinear polynomial systems and reviews non-intrusive model reduction based on operator inference and re-projection [35, 33]. A formulation of the problem is presented and an example is discussed that highlights the challenges of constructing reduced models from partially observed states. Section 3 introduces the proposed approach to learn the non-Markovian terms from data. The proposed reduced models describe the dynamics of the observed states and are learned via a novel extended re-projection algorithm and linear regression that can be implemented with off-the-shelf linear algebra packages. The numerical experiments in Section 4 show that the proposed non-intrusive approach and the resulting non-Markovian reduced models achieve approximations which are orders of magnitude more accurate than models obtained by directly applying operator inference to partially observed state trajectories.

## 2. Preliminaries

We summarize fundamental concepts of intrusive model reduction in Section 2.1. Section 2.2 then discusses non-intrusive model reduction from data via operator inference and re-projection. Section 2.3 formulates the problem of learning reduced models from partially observed states.

### 2.1. Intrusive (traditional) model reduction

Let  $\boldsymbol{\mu} \in \mathcal{D} \subseteq \mathbb{R}^d$  be a parameter in the parameter domain  $\mathcal{D}$  and consider the full model that is a dynamical system with nonlinear polynomial terms up to degree  $\ell \in \mathbb{N}$

$$\begin{aligned} \mathbf{x}_{k+1}(\boldsymbol{\mu}) &= \mathbf{f}(\mathbf{x}_k(\boldsymbol{\mu}), \mathbf{u}_k(\boldsymbol{\mu})) = \sum_{j=1}^{\ell} \mathbf{A}_j(\boldsymbol{\mu}) \mathbf{x}_k^j(\boldsymbol{\mu}) + \mathbf{B}(\boldsymbol{\mu}) \mathbf{u}_k(\boldsymbol{\mu}), \quad k = 0, \dots, K-1, \\ \mathbf{y}_{k+1}(\boldsymbol{\mu}) &= \mathbf{C}(\boldsymbol{\mu}) \mathbf{x}_{k+1}(\boldsymbol{\mu}), \end{aligned} \quad (2.1)$$

where  $k \in \mathbb{N}$  is the time step,  $\mathbf{x}_k(\boldsymbol{\mu}) \in \mathbb{R}^N$  is the state,  $\mathbf{u}_k(\boldsymbol{\mu}) \in \mathbb{R}^p$  is the input,  $\mathbf{y}_{k+1}(\boldsymbol{\mu}) \in \mathbb{R}^s$  is the output,  $\mathbf{B}(\boldsymbol{\mu}) \in \mathbb{R}^{N \times p}$ ,  $\mathbf{C}(\boldsymbol{\mu}) \in \mathbb{R}^{s \times N}$ ,  $\mathbf{A}_j(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N_j}$  are the system matrices with  $N_j = \binom{N+j-1}{j}$  for  $j = 1, \dots, \ell$  and  $K \in \mathbb{N}$  with  $K \geq 1$ . The nonlinear terms in (2.1) depend on vectors  $\mathbf{x}_k^j(\boldsymbol{\mu}) \in \mathbb{R}^{N_j}$  for  $j \geq 2$  which are defined by retaining only the components of the Kronecker product

$$\underbrace{\mathbf{x}_k(\boldsymbol{\mu}) \otimes \dots \otimes \mathbf{x}_k(\boldsymbol{\mu})}_j$$

whose factors are unique up to permutation [35].

To obtain a reduced model with state dimension  $n \ll N$  from (2.1), a projection-based approach via the proper orthogonal decomposition (POD) is performed as follows. For the parameter values  $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_m \in \mathcal{D}$  with their corresponding state trajectories  $\mathbf{X}(\boldsymbol{\mu}_i) = [\mathbf{x}_0(\boldsymbol{\mu}_i), \dots, \mathbf{x}_{K-1}(\boldsymbol{\mu}_i)] \in \mathbb{R}^{N \times K}$  for  $i = 1, \dots, m$ , the first  $n$  left singular vectors of  $[\mathbf{X}(\boldsymbol{\mu}_1), \dots, \mathbf{X}(\boldsymbol{\mu}_m)] \in \mathbb{R}^{N \times mK}$  are computed to form the reduced basis matrix  $\mathbf{V}_x = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{N \times n}$ . A solution to (2.1) is sought in the subspace  $\mathcal{V}_x$  spanned by the columns of  $\mathbf{V}_x$  to obtain the approximation  $\mathbf{V}_x \tilde{\mathbf{x}}_k(\boldsymbol{\mu})$  of  $\mathbf{x}_k(\boldsymbol{\mu})$  where  $\tilde{\mathbf{x}}_k \in \mathbb{R}^n$  is the state of the reduced model.

Applying Galerkin projection onto  $\mathcal{V}_x$ , the reduced model is

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1}(\boldsymbol{\mu}) &= \sum_{j=1}^{\ell} \tilde{\mathbf{A}}_j(\boldsymbol{\mu}) \tilde{\mathbf{x}}_k^j(\boldsymbol{\mu}) + \tilde{\mathbf{B}}(\boldsymbol{\mu}) \mathbf{u}_k(\boldsymbol{\mu}), \quad k = 0, \dots, K-1, \\ \tilde{\mathbf{y}}_{k+1}(\boldsymbol{\mu}) &= \tilde{\mathbf{C}}(\boldsymbol{\mu}) \tilde{\mathbf{x}}_{k+1}(\boldsymbol{\mu}), \end{aligned} \quad (2.2)$$

---

**Algorithm 1.** Data sampling with re-projection

---

- 1: Set  $\bar{\mathbf{x}}_0(\boldsymbol{\mu}) = \mathbf{V}_x^T \mathbf{x}_0(\boldsymbol{\mu})$
  - 2: **for**  $k = 0, \dots, K-1$  **do**
  - 3:   Query (2.1) for a single time step to obtain  $\mathbf{x}_{\text{tmp}} = \sum_{j=1}^{\ell} \mathbf{A}_j(\boldsymbol{\mu})(\mathbf{V}_x \bar{\mathbf{x}}_k(\boldsymbol{\mu}))^j + \mathbf{B}(\boldsymbol{\mu}) \mathbf{u}_k(\boldsymbol{\mu})$
  - 4:   Set  $\bar{\mathbf{x}}_{k+1}(\boldsymbol{\mu}) = \mathbf{V}_x^T \mathbf{x}_{\text{tmp}}$
  - 5: **end for**
  - 6: Return  $[\bar{\mathbf{x}}_0(\boldsymbol{\mu}), \dots, \bar{\mathbf{x}}_K(\boldsymbol{\mu})]$
- 

for  $\boldsymbol{\mu} \in \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_m\}$  where  $\tilde{\mathbf{B}}(\boldsymbol{\mu}) = \mathbf{V}_x^T \mathbf{B}(\boldsymbol{\mu}) \in \mathbb{R}^{n \times p}$ ,  $\tilde{\mathbf{C}}(\boldsymbol{\mu}) = \mathbf{C}(\boldsymbol{\mu}) \mathbf{V}_x \in \mathbb{R}^{s \times n}$ ,  $\tilde{\mathbf{A}}_1(\boldsymbol{\mu}) = \mathbf{V}_x^T \mathbf{A}_1(\boldsymbol{\mu}) \mathbf{V}_x \in \mathbb{R}^{n \times n}$ . The reduced operators  $\tilde{\mathbf{A}}_j(\boldsymbol{\mu}) \in \mathbb{R}^{n \times n_j}$  with  $n_j = \binom{n+j-1}{j}$  for  $j \geq 2$  can be derived in a similar manner; see, e.g., [33]. The process of constructing the reduced operators via matrix-matrix multiplications of the basis matrix  $\mathbf{V}_x$  and the full model operators is intrusive in the sense that the full model operators are required either in assembled form or implicitly through a routine that provides the action of the full model operators on a vector. To derive the reduced model for parameters  $\boldsymbol{\mu} \in \mathcal{D} \setminus \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_m\}$ , the operators of the system (2.2) for  $\boldsymbol{\mu} \in \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_m\}$  are interpolated element-wise [14].

## 2.2. Non-intrusive model reduction with operator inference and re-projection

Operator inference [35] is a non-intrusive model reduction approach that learns operators of a low-dimensional dynamical-system model from data without requiring the full-model operators (2.1) in assembled or implicit form. Together with the data sampling via re-projection [33], operator inference can exactly recover the reduced operators  $\tilde{\mathbf{A}}(\boldsymbol{\mu})_1, \dots, \tilde{\mathbf{A}}(\boldsymbol{\mu})_\ell, \tilde{\mathbf{B}}(\boldsymbol{\mu}), \tilde{\mathbf{C}}(\boldsymbol{\mu})$ .

In data sampling with re-projection, for each  $\boldsymbol{\mu} \in \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_m\}$ , the full system (2.1) is queried at an initial condition  $\mathbf{x}_0(\boldsymbol{\mu}) \in \mathcal{V}_x$  and inputs  $\mathbf{u}_0(\boldsymbol{\mu}), \dots, \mathbf{u}_{K-1}(\boldsymbol{\mu})$  for a single time step followed by a projection step to derive the re-projected trajectories  $\bar{\mathbf{x}}_0(\boldsymbol{\mu}), \dots, \bar{\mathbf{x}}_K(\boldsymbol{\mu}) \in \mathbb{R}^n$ ; see Algorithm 1. Let  $\bar{\mathbf{X}}^j(\boldsymbol{\mu}) = [\bar{\mathbf{x}}_0^j(\boldsymbol{\mu}), \dots, \bar{\mathbf{x}}_{K-1}^j(\boldsymbol{\mu})] \in \mathbb{R}^{n_j \times K}$  for  $j = 1, \dots, \ell$ ,  $\mathbf{W}(\boldsymbol{\mu}) = [\bar{\mathbf{x}}_1(\boldsymbol{\mu}), \dots, \bar{\mathbf{x}}_K(\boldsymbol{\mu})] \in \mathbb{R}^{n \times K}$ , and  $\mathbf{U}(\boldsymbol{\mu}) = [\mathbf{u}_0(\boldsymbol{\mu}), \dots, \mathbf{u}_{K-1}(\boldsymbol{\mu})] \in \mathbb{R}^{p \times K}$ . For each  $\boldsymbol{\mu} \in \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_m\}$ , the least squares problem

$$\min_{\hat{\mathbf{O}}(\boldsymbol{\mu})} \|\mathbf{D}(\boldsymbol{\mu})^T \hat{\mathbf{O}}(\boldsymbol{\mu})^T - \mathbf{W}(\boldsymbol{\mu})^T\|_F^2 \quad (2.3)$$

with the data matrix

$$\mathbf{D}(\boldsymbol{\mu}) = \begin{bmatrix} \bar{\mathbf{X}}^1(\boldsymbol{\mu}) \\ \vdots \\ \bar{\mathbf{X}}^\ell(\boldsymbol{\mu}) \\ \mathbf{U}(\boldsymbol{\mu}) \end{bmatrix} \in \mathbb{R}^{(\sum_{j=1}^{\ell} n_j + p) \times K}$$

is then solved for

$$\hat{\mathbf{O}}(\boldsymbol{\mu}) = [\hat{\mathbf{A}}_1(\boldsymbol{\mu}) \quad \dots \quad \hat{\mathbf{A}}_\ell(\boldsymbol{\mu}) \quad \hat{\mathbf{B}}(\boldsymbol{\mu})] \in \mathbb{R}^{n \times (\sum_{j=1}^{\ell} n_j + p)}.$$

Under mild conditions on  $\mathbf{D}(\boldsymbol{\mu})$ , it was shown in [33] that the unique optimal solution to (2.3) is attained at  $\hat{\mathbf{A}}_j(\boldsymbol{\mu}) = \tilde{\mathbf{A}}_j(\boldsymbol{\mu})$  for  $j = 1, \dots, \ell$  and  $\hat{\mathbf{B}}(\boldsymbol{\mu}) = \tilde{\mathbf{B}}(\boldsymbol{\mu})$  which are exactly the reduced operators from intrusive model reduction as discussed in Section 2.1.

## 2.3. Problem formulation

Consider now the situation where only partially observed state trajectories  $\mathbf{Z}(\boldsymbol{\mu})$  are available, rather than fully observed state trajectories  $\mathbf{X}(\boldsymbol{\mu})$  of the full model. Formally, define  $\mathbf{P}_o \in \{0, 1\}^{r \times N}$  the selection

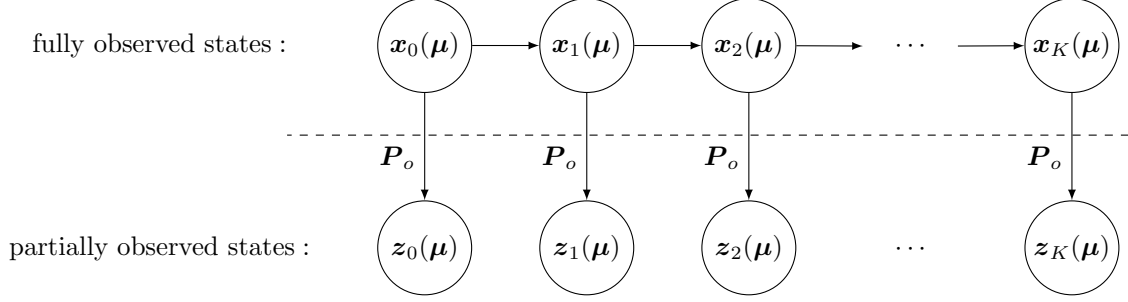


Figure 1.: The goal of this work is to learn reduced models from observations  $z_k(\mu) = P_o x_k(\mu)$  which represent a few components of the full-model states  $x_k(\mu)$ .

matrix whose  $r$  rows are a subset of the rows of the  $N \times N$  identity matrix. We consider the setting in which the system (2.1) can be simulated at various initial conditions and inputs to generate observation trajectories  $\mathbf{Z}(\mu) = [z_0(\mu), \dots, z_K(\mu)]$  and outputs  $\mathbf{y}_0(\mu), \dots, \mathbf{y}_K(\mu)$  that are related to the (unobserved) state trajectories  $\mathbf{X}(\mu)$  of the full model (2.1) via

$$z_k(\mu) = P_o x_k(\mu), \quad k = 0, \dots, K, \quad (2.4)$$

see Figure 1. Implicit and explicit availability of the matrix  $P_o$  is not required in the following.

The aim is now to derive a low-dimensional model with operator inference that can predict the observations (2.4) of the full model at new parameter values in  $\mathcal{D}$  and new inputs. Directly applying operator inference and re-projection (Section 2.2) to trajectories of observations  $\mathbf{Z}(\mu)$  can lead to models that poorly approximate the dynamics of the full model. To see this, consider a POD basis matrix  $\mathbf{V}$  with  $n$  columns derived from the snapshot matrix of observations  $\mathbf{Z} = [\mathbf{Z}(\mu_1), \dots, \mathbf{Z}(\mu_m)]$ . Then, the projection from the fully observed, high-dimensional state  $x_k(\mu)$  to the reduced observation  $\tilde{z}_k(\mu) \in \mathbb{R}^n, n < r$ , is given by  $(P_o^T \mathbf{V})(P_o^T \mathbf{V})^T$ . If a reduced model of the form (2.2) is learned from trajectories of observations, then the reduced states are in the subspace spanned by columns of  $P_o$ ; however, in contrast to, e.g., the POD basis  $\mathbf{V}_x$  of the fully observed states, the columns of  $P_o^T$  are canonical unit vectors that form a basis of a subspace that typically offers an inadequate approximation quality of the full model states.

As an illustration, consider the nonlinear polynomial full model of degree  $\ell = 3$  stemming from the spatio-temporal discretization of the parameter-independent Chafee-Infante equation; details of this numerical experiment are presented in Section 4.3. First, consider a reduced model constructed with operator inference and re-projection as discussed in Section 2.2 when all components of the states can be observed. Figure 2a shows that with  $n = 10$  dimensions, the reduced model output approximates the full model output well. In contrast, learning a reduced model with the non-intrusive model reduction process described in Section 2.2 with observations that contain only 60% of the components of the full model states, rather than all components, leads to the results in Figure 2b. The reduced model trajectories plotted for the output are computed using a test input; cf. Section 4.3 for details. The model learned from the partially observed states provides a poor approximation of the output of the full model.

### 3. Learning non-Markovian reduced models with operator inference

To compensate for the loss of information of learning from partially observed state trajectories, we learn non-Markovian terms that take into account the history (memory) of reduced states at previous time steps to correct reduced models; cf. Mori-Zwanzig formalism [12, 10, 27] and neural-network architectures in machine learning such as the long short-term memory network (LSTM) [21]. This is in stark contrast to traditional, Markovian reduced models of the form (2.2) where the reduced state at the current time step only depends on a single reduced state at the previous time step.

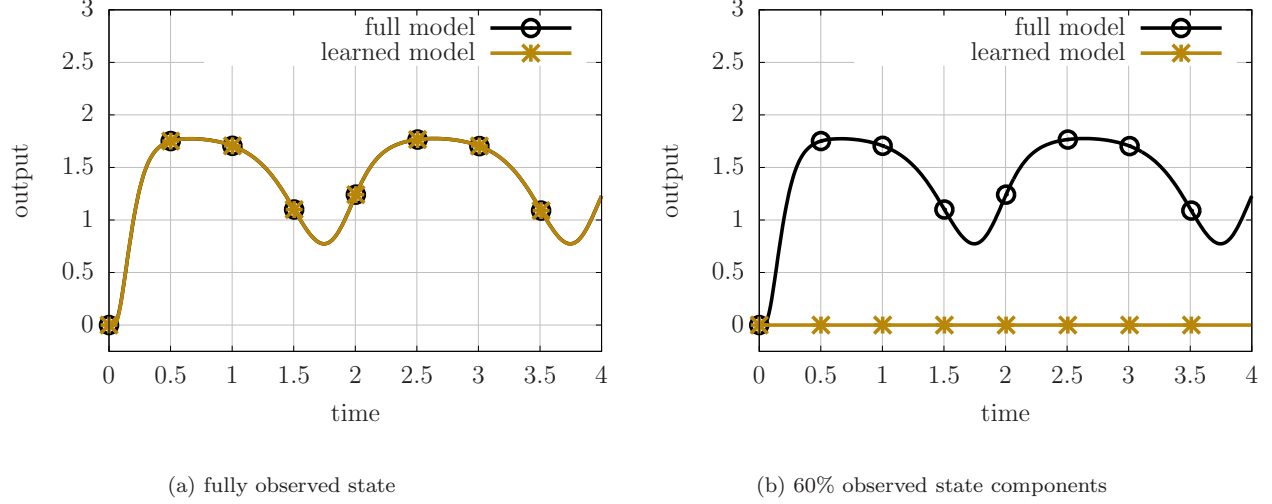


Figure 2.: Chafee-Infante: The left panel shows that a reduced model learned with operator inference and re-projection (Section 2.2) from fully observed state trajectories approximates well the output of the full model. However, applying operator inference when only 60% of the state components (observations) are available, rather than all state components, leads to a reduced model that fails to predict the output of the full model in this example, as shown in the right panel.

Sections 3.1 and 3.2 propose a parametrization of reduced models with non-Markovian terms for linear and nonlinear polynomial full models. A computational procedure to learn the reduced model operators of the non-Markovian terms is introduced in Section 3.3, which discusses an extension of the re-projection algorithm as well as two modes in which operator inference can be applied.

For ease of notation, we drop the parametric dependence on the states, inputs, and operators because the non-Markovian terms described below are learned for each  $\mu \in \{\mu_1, \dots, \mu_m\}$  separately, analogous to the Markovian operators as discussed in Section 2.1.

### 3.1. Reduced systems for partially observed states with linear full dynamics

We first address the setting in which the full model (2.1) is linear in the state variables. We have available observation trajectories  $\mathbf{Z} = [\mathbf{z}_0, \dots, \mathbf{z}_K]$  that contain selected components of the state trajectory  $\mathbf{X} = [\mathbf{x}_0, \dots, \mathbf{x}_K]$  as defined in (2.4). Let  $\mathbf{V} \in \mathbb{R}^{r \times n}$  be the reduced basis matrix obtained via POD from snapshots of observations and  $\mathcal{V}$  be the subspace spanned by its columns; cf Section 2.3. The goal is to derive a model that describes exactly the dynamics of the projected observations  $\tilde{\mathbf{z}}_k = \mathbf{V}^T \mathbf{z}_k \in \mathbb{R}^r$  and the output  $\mathbf{y}_k = \mathbf{C} \mathbf{x}_k$ . Note that the projected observations  $\tilde{\mathbf{z}}_k$  are obtained by first projecting the state  $\mathbf{x}_k$  according to  $\mathbf{P}_o$  and then projecting the observation  $\mathbf{z}_k$  onto the space spanned by the columns of  $\mathbf{V}$ . The output can also be viewed as a projection from  $\mathbb{R}^N$  to  $\mathbb{R}^s$  if the output dimension  $s$  satisfies  $s < N$ .

#### 3.1.1. Dynamics of the projected observations

Define  $\mathbf{P}_o^\perp \in \mathbb{R}^{(N-r) \times N}$  to be the matrix that extracts the components of  $\mathbf{x}_k$  which are missing from  $\mathbf{z}_k$ . If  $\mathbf{0}_{m \times n}$  is an  $m \times n$  matrix of zeros,  $\mathbf{P}_o^\perp$  satisfies the relationship  $\mathbf{P}_o (\mathbf{P}_o^\perp)^T = \mathbf{0}_{r \times N-r}$  such that the columns of  $\mathbf{P}_o$  and  $\mathbf{P}_o^\perp$  are orthonormal. We can therefore express  $\mathbf{x}_k$  as a direct sum via

$$\mathbf{x}_k = \mathbf{P}_o^T \mathbf{z}_k + (\mathbf{P}_o^\perp)^T \mathbf{z}_k^\perp, \quad k = 0, \dots, K, \quad (3.1)$$

for  $\mathbf{z}_k^\perp \in \mathbb{R}^{N-r}$ . In addition, denote by  $\mathbf{V}^\perp \in \mathbb{R}^{r \times (r-n)}$  the matrix whose columns form an orthonormal basis for the orthogonal complement of  $\mathcal{V}$ . If  $\hat{\mathbf{z}}_k \in \mathbb{R}^{r-n}$ ,  $\mathbf{z}_k$  can also be expressed as a direct sum

$$\mathbf{z}_k = \mathbf{V}\tilde{\mathbf{z}}_k + \mathbf{V}^\perp \hat{\mathbf{z}}_k, \quad k = 0, \dots, K. \quad (3.2)$$

Substituting (3.2) into (3.1),  $\mathbf{x}_k$  thus admits the direct sum decomposition

$$\mathbf{x}_k = \mathbf{Q}\tilde{\mathbf{z}}_k + \mathbf{Q}^\perp \mathbf{w}_k, \quad k = 0, \dots, K, \quad (3.3)$$

where  $\mathbf{Q} = \mathbf{P}_o^T \mathbf{V} \in \mathbb{R}^{N \times n}$ ,  $\mathbf{Q}^\perp = [\mathbf{P}_o^T \mathbf{V}^\perp (\mathbf{P}_o^\perp)^T] \in \mathbb{R}^{N \times (N-n)}$ , and  $\mathbf{w}_k = [\hat{\mathbf{z}}_k^T (\mathbf{z}_k^\perp)^T]^T$ . It can be verified that the columns of the matrices  $\mathbf{Q}, \mathbf{Q}^\perp$  are orthonormal and that  $\mathbf{Q}^T \mathbf{Q}^\perp = \mathbf{0}_{n \times (N-n)}$ .

A system of equations of the time evolution of the projected states  $\tilde{\mathbf{z}}_k$  and the orthogonal complement  $\mathbf{w}_k$  can be obtained by substituting (3.3) to (2.1) and pre-multiplying the resulting expression by  $\mathbf{Q}^T$  or  $(\mathbf{Q}^\perp)^T$ , thereby giving the system

$$\tilde{\mathbf{z}}_{k+1} = \mathbf{Q}^T \mathbf{A}_1 \mathbf{Q} \tilde{\mathbf{z}}_k + \mathbf{Q}^T \mathbf{A}_1 \mathbf{Q}^\perp \mathbf{w}_k + \mathbf{Q}^T \mathbf{B} \mathbf{u}_k, \quad k = 0, \dots, K-1, \quad (3.4)$$

$$\mathbf{w}_{k+1} = (\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q} \tilde{\mathbf{z}}_k + (\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q}^\perp \mathbf{w}_k + (\mathbf{Q}^\perp)^T \mathbf{B} \mathbf{u}_k, \quad k = 0, \dots, K-1. \quad (3.5)$$

To extract a dynamical system in terms of  $\tilde{\mathbf{z}}_k$  only, we choose the initial condition  $\mathbf{x}_0$  such that  $\mathbf{w}_0 = (\mathbf{Q}^\perp)^T \mathbf{x}_0 = \mathbf{0}_{N-n}$ . We view  $\tilde{\mathbf{z}}_k$  as a constant in (3.5), solve for  $\mathbf{w}_k$  as

$$\mathbf{w}_k = \sum_{l=0}^{k-1} ((\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q}^\perp)^{k-l-1} ((\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q} \tilde{\mathbf{z}}_l + (\mathbf{Q}^\perp)^T \mathbf{B} \mathbf{u}_l), \quad (3.6)$$

and combine (3.6) with (3.4) to deduce

$$\tilde{\mathbf{z}}_{k+1} = \underbrace{\tilde{\mathbf{A}}_1 \tilde{\mathbf{z}}_k + \tilde{\mathbf{B}} \mathbf{u}_k}_{\text{Markovian term}} + \underbrace{\sum_{l=0}^{k-1} (\mathbf{E}_{k-l} \tilde{\mathbf{z}}_l + \mathbf{F}_{k-l} \mathbf{u}_l)}_{\text{non-Markovian term}}, \quad k = 0, \dots, K-1, \quad (3.7)$$

where  $\tilde{\mathbf{A}}_1 = \mathbf{Q}^T \mathbf{A}_1 \mathbf{Q}$  and  $\tilde{\mathbf{B}} = \mathbf{Q}^T \mathbf{B}$  are the operators for the Markovian term while the operators

$$\begin{aligned} \mathbf{E}_{k-l} &= \mathbf{Q}^T \mathbf{A}_1 \mathbf{Q}^\perp ((\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q}^\perp)^{k-l-1} (\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q} \in \mathbb{R}^{n \times n}, \\ \mathbf{F}_{k-l} &= \mathbf{Q}^T \mathbf{A}_1 \mathbf{Q}^\perp ((\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q}^\perp)^{k-l-1} (\mathbf{Q}^\perp)^T \mathbf{B} \in \mathbb{R}^{n \times p}, \end{aligned} \quad (3.8)$$

for  $l = 0, \dots, k-1$  give rise to the non-Markovian term. Deriving the dynamical system (3.7) is analogous to what the Mori-Zwanzig formalism suggests [12, 10, 27] for linear systems with inputs. Observe that (3.7) shows the dependence of the projected observation  $\tilde{\mathbf{z}}_{k+1}$  at time  $k+1$  on all previous projected observations and inputs. It is comprised of the Markovian term  $\tilde{\mathbf{A}}_1 \tilde{\mathbf{z}}_k + \tilde{\mathbf{B}} \mathbf{u}_k$  and the non-Markovian term that introduces dependence on time steps before  $k$ .

**Remark 1.** The condition that  $(\mathbf{Q}^\perp)^T \mathbf{x}_0 = \mathbf{0}_{N-n}$  is met, for instance, if  $\mathbf{x}_0 = \mathbf{P}_o^T \mathbf{V} \mathbf{V}^T \mathbf{z}_0$  where  $\mathbf{z}_0$  is the initial state in the available observation trajectory  $\mathbf{Z}$ . If this is not the case, the dynamical system (3.7) for  $\tilde{\mathbf{z}}_k$  becomes

$$\tilde{\mathbf{z}}_{k+1} = \tilde{\mathbf{A}}_1 \tilde{\mathbf{z}}_k + \tilde{\mathbf{B}} \mathbf{u}_k + \sum_{l=0}^{k-1} (\mathbf{E}_{k-l} \tilde{\mathbf{z}}_l + \mathbf{F}_{k-l} \mathbf{u}_l) + \mathbf{\Psi}_k \mathbf{w}_0, \quad k = 0, \dots, K-1, \quad (3.9)$$

where  $\mathbf{\Psi}_k = (\mathbf{Q}^T \mathbf{A}_1 \mathbf{Q}^\perp) ((\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q}^\perp)^k \in \mathbb{R}^{n \times (N-n)}$ . Throughout this work,  $\mathbf{x}_0$  is chosen such that  $(\mathbf{Q}^\perp)^T \mathbf{x}_0 = \mathbf{0}_{N-n}$  is satisfied.



### 3.1.2. Dynamics of the output

Following analogous steps as above, the system describing the time evolution of the output can be derived as

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k = \mathbf{C}\mathbf{Q}\tilde{\mathbf{z}}_k + \mathbf{C}\mathbf{Q}^\perp \mathbf{w}_k, \quad k = 0, \dots, K.$$

With (3.6), this results into

$$\mathbf{y}_k = \tilde{\mathbf{C}}\tilde{\mathbf{z}}_k + \sum_{l=0}^{k-1} (\mathbf{G}_{k-l}\tilde{\mathbf{z}}_l + \mathbf{H}_{k-l}\mathbf{u}_l), \quad k = 0, \dots, K, \quad (3.10)$$

where  $\tilde{\mathbf{C}} = \mathbf{C}\mathbf{Q}$  and

$$\begin{aligned} \mathbf{G}_{k-l} &= \mathbf{C}\mathbf{Q}^\perp ((\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q}^\perp)^{k-l-1} (\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q} \in \mathbb{R}^{s \times n}, \\ \mathbf{H}_{k-l} &= \mathbf{C}\mathbf{Q}^\perp ((\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q}^\perp)^{k-l-1} (\mathbf{Q}^\perp)^T \mathbf{B} \in \mathbb{R}^{s \times p}, \end{aligned} \quad (3.11)$$

for  $l = 0, \dots, k-1$ .

### 3.1.3. Linear reduced models for the state and output with non-Markovian terms

Typically, the norm of the operators in the non-Markovian term in (3.7) and (3.10) decay as one goes further back in time. This motivates truncating the non-Markovian term and taking only at most the previous  $L \in \mathbb{N}$  states and inputs into account; we refer to  $L$  as the lag. Thus, the reduced models we seek to learn from observation trajectories are parameterized as

$$\begin{aligned} \tilde{\mathbf{z}}_{k+1}^{(L)} &= \tilde{\mathbf{A}}_1 \tilde{\mathbf{z}}_k^{(L)} + \tilde{\mathbf{B}}\mathbf{u}_k + \sum_{l=k-L}^{k-1} (\mathbf{E}_{k-l}\tilde{\mathbf{z}}_l^{(L)} + \mathbf{F}_{k-l}\mathbf{u}_l), \quad k = 0, \dots, K-1, \\ \mathbf{y}_k^{(L)} &= \tilde{\mathbf{C}}\tilde{\mathbf{z}}_k^{(L)} + \sum_{l=k-L}^{k-1} (\mathbf{G}_{k-l}\tilde{\mathbf{z}}_l^{(L)} + \mathbf{H}_{k-l}\mathbf{u}_l), \end{aligned} \quad (3.12)$$

where we use the convention that  $\tilde{\mathbf{z}}_l^{(L)} = \mathbf{0}_{n \times 1}$  and  $\mathbf{u}_l = \mathbf{0}_{p \times 1}$  for negative integers  $l \in \mathbb{Z}^-$  in the remainder of this work.

Let us remark on the error of non-Markovian versus Markovian reduced models. Denote by  $\tilde{\mathbf{z}}_k^{(0)}$  the reduced state resulting from retaining the Markovian term only in (3.7), i.e. the state of the Markovian reduced model  $\tilde{\mathbf{z}}_{k+1}^{(0)} = \tilde{\mathbf{A}}_1 \tilde{\mathbf{z}}_k^{(0)} + \tilde{\mathbf{B}}\mathbf{u}_k$ . The initial conditions for the non-Markovian and Markovian model are set to be identical  $\tilde{\mathbf{z}}_0 = \tilde{\mathbf{z}}_0^{(0)} = \tilde{\mathbf{z}}_0^{(L)}$ . For a fixed time step  $k$ , if

$$\|\mathbf{V}(\tilde{\mathbf{z}}_k - \tilde{\mathbf{z}}_l^{(L)})\|_2 = \|\tilde{\mathbf{z}}_k - \tilde{\mathbf{z}}_l^{(L)}\|_2 < \|\mathbf{z}_k - \mathbf{V}\tilde{\mathbf{z}}_k^{(0)}\|_2 - \|\mathbf{z}_k - \mathbf{V}\tilde{\mathbf{z}}_k\|_2 \quad (3.13)$$

holds, then, by the triangle inequality, we obtain

$$\|\mathbf{z}_k - \mathbf{V}\tilde{\mathbf{z}}_k^{(L)}\|_2 = \|\mathbf{z}_k - \mathbf{V}\tilde{\mathbf{z}}_k + \mathbf{V}(\tilde{\mathbf{z}}_k - \tilde{\mathbf{z}}_k^{(L)})\|_2 < \|\mathbf{z}_k - \mathbf{V}\tilde{\mathbf{z}}_k^{(0)}\|_2. \quad (3.14)$$

Inequality (3.14) implies that, if (3.13) holds, the error of the non-Markovian state  $\tilde{\mathbf{z}}_k^{(L)}$  is lower than the error of the Markovian state  $\tilde{\mathbf{z}}_k^{(0)}$ . From (3.2), we see that  $\tilde{\mathbf{z}}_k$  is the projection of  $\mathbf{z}_k$  onto  $\mathcal{V}$ . Therefore,  $\|\mathbf{z}_k - \mathbf{V}\tilde{\mathbf{z}}_k^{(0)}\|_2 - \|\mathbf{z}_k - \mathbf{V}\tilde{\mathbf{z}}_k\|_2 \geq 0$ . The condition (3.13) holds, for example, if  $k < L + 2$  since  $\tilde{\mathbf{z}}_k = \tilde{\mathbf{z}}_k^{(L)}$ .

In general, however, examples can be constructed for which (3.14) is violated for selected time steps  $k$  with  $L$  fixed; see Appendix A. A theoretical analysis remains future work.



### 3.2. Reduced models with non-Markovian terms for nonlinear polynomial systems

For polynomial full models (2.1), an analogous procedure as in Section 3.1 leads to polynomial dynamics of the projected observations  $\tilde{\mathbf{z}}_k$  with a polynomial non-Markovian term that takes the history of states and inputs into account. However, the number of summands in the non-Markovian term grows combinatorially in the degree  $\ell$  of the full model and involves powers of the projected observations  $\tilde{\mathbf{z}}_k$  and the inputs  $\mathbf{u}_k$  and the Kronecker products of their powers. Thus, even truncating the non-Markovian term at lag  $L$ —retaining only the summands that are functions of the Markovian state and input and the  $L$  previous states and inputs—quickly becomes computationally intractable in terms of number of degrees of freedom and costs of simulating the corresponding model.

Instead, we propose a two step approximation to design non-Markovian terms for nonlinear polynomial full models: first, we consider a linear approximation of the non-Markovian term and second, truncate the linearized non-Markovian term at lag  $L$ . This results in the reduced model

$$\begin{aligned}\tilde{\mathbf{z}}_{k+1}^{(L)} &= \sum_{j=1}^{\ell} \tilde{\mathbf{A}}_j (\tilde{\mathbf{z}}_k^{(L)})^j + \tilde{\mathbf{B}} \mathbf{u}_k + \sum_{l=k-L}^{k-1} (\mathbf{E}_{k-l} \tilde{\mathbf{z}}_l^{(L)} + \mathbf{F}_{k-l} \mathbf{u}_l), \quad k = 0, \dots, K-1, \\ \mathbf{y}_k^{(L)} &= \tilde{\mathbf{C}} \tilde{\mathbf{z}}_k^{(L)} + \sum_{l=k-L}^{k-1} (\mathbf{G}_{k-l} \tilde{\mathbf{z}}_l^{(L)} + \mathbf{H}_{k-l} \mathbf{u}_l),\end{aligned}\tag{3.15}$$

with the reduced state  $\tilde{\mathbf{z}}_k^{(L)}$  and output  $\mathbf{y}_k^{(L)}$  at time  $k$ . Notice that the model (3.15) is nonlinear in the state variable only in the Markovian term but linear in the non-Markovian term.

### 3.3. Operator inference for non-Markovian terms

We now introduce a procedure to learn the non-Markovian operators in (3.12) and (3.15) from data. In particular, we will show that if the full model (2.1) is linear, then the operators  $(\mathbf{E}_l, \mathbf{F}_l)$  and  $(\mathbf{G}_l, \mathbf{H}_l)$ ,  $l = 1, \dots, L$ , in the non-Markovian term of (3.12) defined in (3.8) and (3.11) are recovered with our procedure.

We first discuss a data sampling scheme to generate trajectories of observations. The non-Markovian operators are then learned either simultaneously (“batch”) or in a stagewise manner with operator inference.

#### 3.3.1. Data generation: Extended re-projection algorithm

We extend the re-projection algorithm of Section 2.2 to generate data for learning the non-Markovian terms. We emphasize that in the following, re-projection is applied to observations, rather than state trajectories. Instead of performing re-projection after each time step as in Algorithm 1, we propose to query the full model for  $K_r > 1$  time steps before the next re-projection step is performed; this cycle is repeated as illustrated in Figure 3.

Denote the initial condition by  $\mathbf{z}_0^{(0)} \in \mathbb{R}^r$  and let  $N_r$  be the number re-projection steps. For the  $i$ -th re-projection step, denote by  $\tilde{\mathbf{z}}_0^{(i)} = \mathbf{V}^T \mathbf{z}_{K_r}^{(i-1)} \in \mathbb{R}^n$  the projected observation and set  $\mathbf{z}_0^{(i)} = \mathbf{V} \tilde{\mathbf{z}}_0^{(i)} \in \mathbb{R}^r$ . We then query the full model for  $K_r$  time steps starting from  $\mathbf{z}_0^{(i)}$  (or the corresponding  $\mathbf{P}_o^T \mathbf{z}_0^{(i)} \in \mathbb{R}^N$ ) with inputs  $[\mathbf{u}_0^{(i)}, \dots, \mathbf{u}_{K_r-1}^{(i)}]$  to compute the observation trajectory  $[\mathbf{z}_0^{(i)}, \dots, \mathbf{z}_{K_r-1}^{(i)}, \mathbf{z}_{K_r}^{(i)}]$  and the corresponding output trajectory  $[\mathbf{y}_0^{(i)}, \dots, \mathbf{y}_{K_r-1}^{(i)}]$ . Set  $\tilde{\mathbf{z}}_k^{(i)} = \mathbf{V}^T \mathbf{z}_k^{(i)}$  for  $k = 1, \dots, K_r-1$ . The  $(i+1)$ -th re-projection step is subsequently initiated with  $\tilde{\mathbf{z}}_0^{(i+1)} = \mathbf{V}^T \mathbf{z}_{K_r}^{(i)}$  and the full model is sampled again for  $K_r$  time steps. Thus, the proposed data sampling scheme yields the re-projected trajectories  $\{[\tilde{\mathbf{z}}_0^{(i)}, \dots, \tilde{\mathbf{z}}_{K_r-1}^{(i)}]\}_{i=1}^{N_r}$  and output data  $\{[\mathbf{y}_0^{(i)}, \dots, \mathbf{y}_{K_r-1}^{(i)}]\}_{i=1}^{N_r}$  as a result. The extended re-projection procedure is summarized in Algorithm 2.

**Remark 2.** If querying the full model requires an initial condition at each of the  $N$  state components, in contrast to Algorithm 2 that provides an initial condition  $\mathbf{z}_0^{(0)}$  only at the  $r$  observed state components,

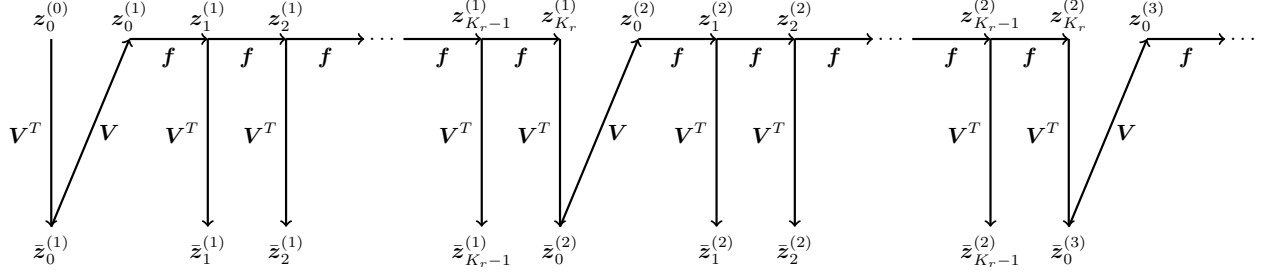


Figure 3.: Diagram of the extended re-projection algorithm. Instead of performing re-projection steps after each time step, the full model is queried for  $K_r$  time steps before re-projection is performed. The observations resulting from the queried full model are then projected to the reduced subspace.

---

**Algorithm 2.** Data sampling with extended re-projection

---

- 1: Set  $\bar{z}_0^{(1)} = \mathbf{V}^T \mathbf{z}_0^{(0)}$
  - 2: **for**  $i = 1, \dots, N_r$  **do**
  - 3:   Set  $\mathbf{z}_0^{(i)} = \mathbf{V} \bar{z}_0^{(i)}$
  - 4:   Query (2.4) at  $\mathbf{z}_0^{(i)}$  (or at  $\mathbf{P}_o^T \mathbf{z}_0^{(i)}$ ) and  $[\mathbf{u}_0^{(i)}, \dots, \mathbf{u}_{K_r-1}^{(i)}]$  to obtain  $[\mathbf{z}_0^{(i)}, \dots, \mathbf{z}_{K_r-1}^{(i)}, \mathbf{z}_{K_r}^{(i)}]$  and  $[\mathbf{y}_0^{(i)}, \dots, \mathbf{y}_{K_r-1}^{(i)}]$
  - 5:   Set  $\bar{z}_k^{(i)} = \mathbf{V}^T \mathbf{z}_k^{(i)}$  for  $k = 1, \dots, K_r - 1$
  - 6:   Set  $\bar{z}_0^{(i+1)} = \mathbf{V}^T \mathbf{z}_{K_r}^{(i)}$
  - 7: **end for**
  - 8: Return  $\{[\bar{z}_0^{(i)}, \dots, \bar{z}_{K_r-1}^{(i)}]\}_{i=1}^{N_r}$  and  $\{[\mathbf{y}_0^{(i)}, \dots, \mathbf{y}_{K_r-1}^{(i)}]\}_{i=1}^{N_r}$
- 

then such an initial condition can be derived with  $\mathbf{P}_o^T \mathbf{z}_0^{(0)}$  and the proposed re-projection procedure is still applicable.

### 3.3.2. Stagewise operator inference of non-Markovian operators

As shown in [33], under appropriate conditions on  $N_r$  and the resulting data matrix in the least squares problem, the trajectories  $\{[\bar{z}_0^{(i)}, \bar{z}_1^{(i)}]\}_{i=1}^{N_r}$  and  $\{(\bar{z}_0^{(i)}, \mathbf{y}_0^{(i)})\}_{i=1}^{N_r}$  are sufficient to recover the Markovian operators in (3.12) and (3.15). This is because for  $i = 1, \dots, N_r$ ,  $\bar{z}_0^{(i)}$  and  $\bar{z}_1^{(i)}$  satisfy the Markovian dynamics

$$\bar{z}_1^{(i)} = \sum_{j=1}^{\ell} \tilde{\mathbf{A}}_j (\bar{z}_0^{(i)})^j + \tilde{\mathbf{B}} \mathbf{u}_0^{(i)}.$$

Thus, we will only focus on inferring the non-Markovian operators. Observe that for fixed  $i$ , the dynamical system satisfied by  $\bar{z}_k^{(i)}$  is exactly that of  $\bar{z}_k$  defined in Sections 3.1 and 3.2 since  $\bar{z}_k^{(i)} = \mathbf{V}^T \mathbf{z}_k^{(i)}$  for  $k = 1, \dots, K_r - 1$ .

We now propose stagewise operator inference where the non-Markovian operators  $(\mathbf{E}_l, \mathbf{F}_l)$  and  $(\mathbf{G}_l, \mathbf{H}_l)$  in (3.12) and (3.15) are learned sequentially from data for each  $l = 1, \dots, L$ . Denote by  $(\hat{\mathbf{E}}_l, \hat{\mathbf{F}}_l)$  and  $(\hat{\mathbf{G}}_l, \hat{\mathbf{H}}_l)$  the estimates of  $(\mathbf{E}_l, \mathbf{F}_l)$  and  $(\mathbf{G}_l, \mathbf{H}_l)$  for  $l \in \mathbb{N}$ . For fixed  $l$ , suppose that the operators  $(\hat{\mathbf{E}}_j, \hat{\mathbf{F}}_j)$  and  $(\hat{\mathbf{G}}_j, \hat{\mathbf{H}}_j)$  for  $j = 1, \dots, l - 1$  have already been inferred. The operators  $(\mathbf{E}_l, \mathbf{F}_l)$  and  $(\mathbf{G}_l, \mathbf{H}_l)$  are learned by solving for  $(\hat{\mathbf{E}}_l, \hat{\mathbf{F}}_l)$  and  $(\hat{\mathbf{G}}_l, \hat{\mathbf{H}}_l)$  in the least squares problems given by

$$\min_{\hat{\mathbf{E}}_l \in \mathbb{R}^{n \times n}, \hat{\mathbf{F}}_l \in \mathbb{R}^{n \times p}} \sum_{i=1}^{N_r} \|\Delta_{\mathbf{z}}^{(i)}(l, 1) - (\hat{\mathbf{E}}_l \bar{z}_0^{(i)} + \hat{\mathbf{F}}_l \mathbf{u}_0^{(i)})\|_2^2 \quad (3.16)$$

and

$$\min_{\hat{\mathbf{G}}_l \in \mathbb{R}^{s \times n}, \hat{\mathbf{H}}_l \in \mathbb{R}^{s \times p}} \sum_{i=1}^{N_r} \|\Delta_{\mathbf{y}}^{(i)}(l, 1) - (\hat{\mathbf{G}}_l \bar{\mathbf{z}}_0^{(i)} + \hat{\mathbf{H}}_l \mathbf{u}_0^{(i)})\|_2^2, \quad (3.17)$$

where for  $\alpha \in \{0, 1\}$ ,

$$\begin{aligned} \Delta_{\mathbf{z}}^{(i)}(l, \alpha) &= \bar{\mathbf{z}}_{l+1}^{(i)} - \left( \sum_{j=1}^{\ell} \tilde{\mathbf{A}}_j (\bar{\mathbf{z}}_l^{(i)})^j + \tilde{\mathbf{B}} \mathbf{u}_l^{(i)} + \alpha \sum_{j=1}^{l-1} (\hat{\mathbf{E}}_{l-j} \bar{\mathbf{z}}_j^{(i)} + \hat{\mathbf{F}}_{l-j} \mathbf{u}_j^{(i)}) \right), \\ \Delta_{\mathbf{y}}^{(i)}(l, \alpha) &= \mathbf{y}_l^{(i)} - \left( \tilde{\mathbf{C}} \bar{\mathbf{z}}_l^{(i)} + \alpha \sum_{j=1}^{l-1} (\hat{\mathbf{G}}_{l-j} \bar{\mathbf{z}}_j^{(i)} + \hat{\mathbf{H}}_{l-j} \mathbf{u}_j^{(i)}) \right). \end{aligned} \quad (3.18)$$

The quantities  $\Delta_{\mathbf{z}}^{(i)}(l, 1)$  and  $\Delta_{\mathbf{y}}^{(i)}(l, 1)$  are the residuals of  $\bar{\mathbf{z}}_{l+1}^{(i)}$  and  $\mathbf{y}_l^{(i)}$  with respect to the reduced model with non-Markovian term with lag  $l-1$ .

To write the least-squares problems in matrix form, set  $\Delta_{\mathbf{z}}(l, 1) = [\Delta_{\mathbf{z}}^{(1)}(l, 1), \dots, \Delta_{\mathbf{z}}^{(N_r)}(l, 1)] \in \mathbb{R}^{n \times N_r}$  and  $\Delta_{\mathbf{y}}(l, 1) = [\Delta_{\mathbf{y}}^{(1)}(l, 1), \dots, \Delta_{\mathbf{y}}^{(N_r)}(l, 1)] \in \mathbb{R}^{s \times N_r}$  and let  $\hat{\mathbf{O}}_l^{\mathbf{z}} = [\hat{\mathbf{E}}_l \ \hat{\mathbf{F}}_l] \in \mathbb{R}^{n \times (n+p)}$  and  $\hat{\mathbf{O}}_l^{\mathbf{y}} = [\hat{\mathbf{G}}_l \ \hat{\mathbf{H}}_l] \in \mathbb{R}^{s \times (n+p)}$  be the matrices of unknowns. Set the data matrix as

$$\hat{\mathbf{D}} = \begin{bmatrix} \bar{\mathbf{z}}_0^{(1)} & \dots & \bar{\mathbf{z}}_0^{(N_r)} \\ \mathbf{u}_0^{(1)} & \dots & \mathbf{u}_0^{(N_r)} \end{bmatrix} \in \mathbb{R}^{(n+p) \times N_r} \quad (3.19)$$

so that (3.16) and (3.17) are, in matrix form,

$$\min_{\hat{\mathbf{O}}_l^{\mathbf{z}} \in \mathbb{R}^{n \times (n+p)}} \|\hat{\mathbf{D}}^T (\hat{\mathbf{O}}_l^{\mathbf{z}})^T - \Delta_{\mathbf{z}}(l, 1)^T\|_F^2 \quad (3.20)$$

and

$$\min_{\hat{\mathbf{O}}_l^{\mathbf{y}} \in \mathbb{R}^{s \times (n+p)}} \|\hat{\mathbf{D}}^T (\hat{\mathbf{O}}_l^{\mathbf{y}})^T - \Delta_{\mathbf{y}}(l, 1)^T\|_F^2, \quad (3.21)$$

respectively. Notice that the data matrix in the least squares problems (3.20) and (3.21) is the same for all  $l = 1, \dots, L$ .

Algorithm 3 summarizes the stagewise operator inference procedure for the non-Markovian term. To learn the operators up to lag  $L$ , the number of time steps in the extended re-projection algorithm has to be  $K_r \geq L + 2$ . If the full model (2.1) is linear, the non-Markovian operators as defined in (3.8) and (3.11) can be recovered exactly under appropriate conditions on  $N_r$  and  $\hat{\mathbf{D}}$ . This is shown by the following proposition.

**Proposition 3.** *Let the full model in (2.1) be linear. If  $K_r \geq L + 2$ ,  $N_r \geq n + p$  and the data matrix  $\hat{\mathbf{D}}$  in (3.19) has full rank, then for  $l = 1, \dots, L$ , the least squares problems (3.16) and (3.17) are uniquely solved at  $(\hat{\mathbf{E}}_l, \hat{\mathbf{F}}_l) = (\mathbf{E}_l, \mathbf{F}_l)$  and  $(\hat{\mathbf{G}}_l, \hat{\mathbf{H}}_l) = (\mathbf{G}_l, \mathbf{H}_l)$  defined in (3.8) and (3.11), respectively, with objective value 0.*

*Proof.* The equations (3.7) and (3.10) describe the dynamics of  $\tilde{\mathbf{z}}_k = \mathbf{V}^T \mathbf{z}_k$  and  $\mathbf{y}_k$ . Therefore, following the extended re-projection procedure in Algorithm 2, for fixed  $i = 1, \dots, N_r$ , the reduced state  $\bar{\mathbf{z}}_k^{(i)}$ , the input  $\mathbf{u}_k^{(i)}$ , and the output  $\mathbf{y}_k^{(i)}$  satisfy

$$\begin{aligned} \bar{\mathbf{z}}_{k+1}^{(i)} &= \tilde{\mathbf{A}}_1 \bar{\mathbf{z}}_k^{(i)} + \tilde{\mathbf{B}} \mathbf{u}_k^{(i)} + \sum_{l=0}^{k-1} (\mathbf{E}_{k-l} \bar{\mathbf{z}}_l^{(i)} + \mathbf{F}_{k-l} \mathbf{u}_l^{(i)}), \\ \mathbf{y}_k^{(i)} &= \tilde{\mathbf{C}} \bar{\mathbf{z}}_k^{(i)} + \sum_{l=0}^{k-1} (\mathbf{G}_{k-l} \bar{\mathbf{z}}_l^{(i)} + \mathbf{H}_{k-l} \mathbf{u}_l^{(i)}), \end{aligned}$$

---

**Algorithm 3.** Stagewise operator inference for learning non-Markovian correction terms

---

- 1: Formulate the data matrix  $\hat{\mathbf{D}}$  in (3.19)
  - 2: **for**  $l = 1, \dots, L$  **do**
  - 3:   Compute the components of  $\Delta_{\mathbf{z}}(l, 1)$  and  $\Delta_{\mathbf{y}}(l, 1)$  defined in (3.18)
  - 4:   Solve for  $\hat{\mathbf{O}}_l^{\mathbf{z}}$  and  $\hat{\mathbf{O}}_l^{\mathbf{y}}$  in (3.20) and (3.21)
  - 5: **end for**
  - 6: Return  $(\hat{\mathbf{E}}_l, \hat{\mathbf{F}}_l)$  and  $(\hat{\mathbf{G}}_l, \hat{\mathbf{H}}_l)$  for  $l = 1, \dots, L$
- 

for  $k = 1, \dots, K_r - 2$ , where  $(\mathbf{E}_l, \mathbf{F}_l)$  and  $(\mathbf{G}_l, \mathbf{H}_l)$  are defined in (3.8) and (3.11). This implies that (3.16) and (3.17) have objective value 0 when  $(\hat{\mathbf{E}}_l, \hat{\mathbf{F}}_l) = (\mathbf{E}_l, \mathbf{F}_l)$  and  $(\hat{\mathbf{G}}_l, \hat{\mathbf{H}}_l) = (\mathbf{G}_l, \mathbf{H}_l)$ . The solutions to these least squares problems are unique since the problems (3.16) and (3.20) and the problems (3.17) and (3.21) each have the same solution if  $\hat{\mathbf{D}}$  is full rank.  $\square$

### 3.3.3. Batch operator inference of non-Markovian operators

We now propose batch operator inference that infers all non-Markovian operators simultaneously instead of proceeding in a sequential manner as the stagewise approach. Batch operator inference learns the operators  $(\mathbf{E}_l, \mathbf{F}_l)$  and  $(\mathbf{G}_l, \mathbf{H}_l)$  by solving for  $(\check{\mathbf{E}}_l, \check{\mathbf{F}}_l)$  and  $(\check{\mathbf{G}}_l, \check{\mathbf{H}}_l)$  in the least squares problems given by

$$\min_{\substack{\check{\mathbf{E}}_l \in \mathbb{R}^{n \times n}, \check{\mathbf{F}}_l \in \mathbb{R}^{n \times p} \\ l=1, \dots, L}} \sum_{k=1}^{K_r-2} \sum_{i=1}^{N_r} \left\| \Delta_{\mathbf{z}}^{(i)}(k, 0) - \sum_{l=k-L}^{k-1} \left( \check{\mathbf{E}}_{k-l} \bar{\mathbf{z}}_l^{(i)} + \check{\mathbf{F}}_{k-l} \mathbf{u}_l^{(i)} \right) \right\|_2^2 \quad (3.22)$$

and

$$\min_{\substack{\check{\mathbf{G}}_l \in \mathbb{R}^{s \times n}, \check{\mathbf{H}}_l \in \mathbb{R}^{s \times p} \\ l=1, \dots, L}} \sum_{k=1}^{K_r-2} \sum_{i=1}^{N_r} \left\| \Delta_{\mathbf{y}}^{(i)}(k, 0) - \sum_{l=k-L}^{k-1} \left( \check{\mathbf{G}}_{k-l} \bar{\mathbf{z}}_l^{(i)} + \check{\mathbf{H}}_{k-l} \mathbf{u}_l^{(i)} \right) \right\|_2^2, \quad (3.23)$$

where  $\Delta_{\mathbf{z}}^{(i)}(k, 0)$  and  $\Delta_{\mathbf{y}}^{(i)}(k, 0)$  defined in (3.18) represent the discrepancy between the current reduced state and output with the corresponding Markovian model.

To write the least-squares problems (3.22) and (3.23) in matrix form, set  $\Delta_{\mathbf{z}}(k, 0) = [\Delta_{\mathbf{z}}^{(1)}(k, 0), \dots, \Delta_{\mathbf{z}}^{(N_r)}(k, 0)] \in \mathbb{R}^{n \times N_r}$  and  $\Delta_{\mathbf{y}}(k, 0) = [\Delta_{\mathbf{y}}^{(1)}(k, 0), \dots, \Delta_{\mathbf{y}}^{(N_r)}(k, 0)] \in \mathbb{R}^{s \times N_r}$  for  $k = 1, \dots, K_r - 2$ . The right hand side matrix is then given by

$$\begin{aligned} \Delta_{\mathbf{z}} &= [\Delta_{\mathbf{z}}(1, 0), \dots, \Delta_{\mathbf{z}}(K_r - 2, 0)] \in \mathbb{R}^{n \times N_r(K_r-2)}, \\ \Delta_{\mathbf{y}} &= [\Delta_{\mathbf{y}}(1, 0), \dots, \Delta_{\mathbf{y}}(K_r - 2, 0)] \in \mathbb{R}^{s \times N_r(K_r-2)}. \end{aligned} \quad (3.24)$$

The matrices of unknowns are

$$\begin{aligned} \check{\mathbf{O}}^{\mathbf{z}} &= [\check{\mathbf{E}}_1 \ \check{\mathbf{F}}_1 \ \dots \ \check{\mathbf{E}}_L \ \check{\mathbf{F}}_L] \in \mathbb{R}^{n \times (n+p)L}, \\ \check{\mathbf{O}}^{\mathbf{y}} &= [\check{\mathbf{G}}_1 \ \check{\mathbf{H}}_1 \ \dots \ \check{\mathbf{G}}_L \ \check{\mathbf{H}}_L] \in \mathbb{R}^{s \times (n+p)L}. \end{aligned} \quad (3.25)$$

Finally, for  $k = 1, \dots, K_r - 2$ , define  $\check{\mathbf{D}}(k) \in \mathbb{R}^{(n+p)L \times N_r}$  as

$$\check{\mathbf{D}}^T(k) = \begin{bmatrix} (\bar{\mathbf{z}}_{k-1}^{(1)})^T & (\mathbf{u}_{k-1}^{(1)})^T & \dots & (\bar{\mathbf{z}}_{k-L}^{(1)})^T & (\mathbf{u}_{k-L}^{(1)})^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (\bar{\mathbf{z}}_{k-1}^{(N_r)})^T & (\mathbf{u}_{k-1}^{(N_r)})^T & \dots & (\bar{\mathbf{z}}_{k-L}^{(N_r)})^T & (\mathbf{u}_{k-L}^{(N_r)})^T \end{bmatrix}$$

---

**Algorithm 4.** Batch operator inference for learning non-Markovian correction terms

---

- 1: **for**  $k = 1, \dots, K_r - 2$  **do**
  - 2:   Compute  $\Delta_z(k, 0), \Delta_y(k, 0), \check{D}(k)$
  - 3: **end for**
  - 4: Formulate  $\Delta_z$  and  $\Delta_y$  in (3.24)
  - 5: Formulate the data matrix  $\check{D}$  in (3.26)
  - 6: Solve for  $\check{O}^z$  and  $\check{O}^y$  in (3.27) and (3.28)
  - 7: Return  $(\check{E}_l, \check{F}_l)$  and  $(\check{G}_l, \check{H}_l)$  for  $l = 1, \dots, L$
- 

and set the data matrix to

$$\check{D} = [\check{D}(1) \ \dots \ \check{D}(K_r - 2)] \in \mathbb{R}^{(n+p)L \times (K_r - 2)N_r}. \quad (3.26)$$

The problems (3.22) and (3.23) are then, in matrix form,

$$\min_{\check{O}^z \in \mathbb{R}^{n \times (n+p)L}} \|\check{D}^T (\check{O}^z)^T - \Delta_z^T\|_F^2 \quad (3.27)$$

and

$$\min_{\check{O}^y \in \mathbb{R}^{n \times (n+p)L}} \|\check{D}^T (\check{O}^y)^T - \Delta_y^T\|_F^2, \quad (3.28)$$

respectively.

The batch operator inference procedure is summarized in Algorithm 4. Like the stagewise approach, batch operator inference recovers the non-Markovian operators if the full model (2.1) is linear.

**Proposition 4.** *If (2.1) is linear,  $K_r = L + 2$ ,  $(K_r - 2)N_r \geq (n + p)L$  and the data matrix  $\check{D}$  in (3.26) is full rank, the least squares problems (3.22) and (3.23) are uniquely minimized at  $(\check{E}_l, \check{F}_l) = (E_l, F_l)$  and  $(\check{G}_l, \check{H}_l) = (G_l, H_l)$  for  $l = 1, \dots, L$  defined in (3.8) and (3.11), respectively, with objective value 0.*

*Proof.* The proof is analogous to the proof of Proposition 3. □

## 4. Numerical experiments

We conduct numerical experiments with data from linear full models based on the convection-diffusion equation in Section 4.1 and nonlinear models corresponding to diffusion-reaction processes in Section 4.2 and the Chafee-Infante equation in Section 4.3. The number of the observed state components ranges from 20, 40, 60 to 80 percent of the total number of state components  $N$ , the dimension of the full system (2.1). The observed components correspond to equidistant points in the spatial domain for problems with one-dimensional spatial domains. In problems with two-dimensional spatial domains, the observed state components are selected equidistantly with respect to the indexing of the grid points. The Markovian operators are learned according to the methodology in [35, 33] which guarantees recovery of the operators in intrusive model reduction from data in our examples up to numerical errors.

#### 4.1. Convection-diffusion equation

Consider a linear parabolic PDE described by the convection-diffusion equation on the time domain  $t \in (0, 0.5)$  and spatial domain  $\Omega = (0, 1) \times (0, 0.25)$  with boundary  $\partial\Omega$  given by

$$\begin{aligned} \frac{\partial}{\partial t}x(\xi_1, \xi_2, t) &= \nabla \cdot (\nabla x(\xi_1, \xi_2, t)) - (1, 1) \cdot \nabla x(\xi_1, \xi_2, t), \\ x(\xi_1, \xi_2, t) &= 0 \quad \text{for } (\xi_1, \xi_2) \in \partial\Omega \setminus \bigcup_{i=1}^5 E_i, \\ \nabla x(\xi_1, \xi_2, t) \cdot \mathbf{n} &= u_i(t) \quad \text{for } (\xi_1, \xi_2) \in E_i, i = 1, \dots, 5, \\ x(\xi_1, \xi_2, 0) &= 0 \end{aligned} \tag{4.1}$$

for  $(\xi_1, \xi_2, t) \in \Omega \times (0, 0.5)$ . The pieces of the boundary  $\partial\Omega$  with prescribed Neumann conditions are  $E_1 = \{(\xi_1, \xi_2) : \xi_1 \in (0.8, 1), \xi_2 = 0\} \cup \{(\xi_1, \xi_2) : \xi_1 = 1, \xi_2 \in (0, 0.25)\}$ ,  $E_2 = \{(\xi_1, \xi_2) : \xi_1 \in (0.6, 0.8), \xi_2 = 0.25\}$ ,  $E_3 = \{(\xi_1, \xi_2) : \xi_1 \in (0.2, 0.4), \xi_2 = 0.25\}$ ,  $E_4 = \{(\xi_1, \xi_2) : \xi_1 \in (0, 0.2), \xi_2 = 0\}$ , and  $E_5 = \{(\xi_1, \xi_2) : \xi_1 \in (0.4, 0.6), \xi_2 = 0\}$ . The Neumann boundary condition at each of these edges is driven by an input  $u_i(t)$  for  $i = 1, \dots, 5$ . We set the output to be the integral of the PDE solution along  $E_5$ , i.e.  $y(t) = \int_{E_5} x(\xi_1, \xi_2, t) \partial\Omega$ .

The PDE (4.1) is temporally discretized using finite difference with time step size  $\delta t = 10^{-5}$  and spatially discretized using square finite elements with width  $\Delta\xi_1 = \Delta\xi_2 = 1/75$  and  $N = 1121$  linear hat basis functions. This yields the high-dimensional system

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_1 \mathbf{x}_k + \mathbf{B} \mathbf{u}_k, & k = 0, \dots, K-1, \\ \mathbf{y}_{k+1} &= \mathbf{C} \mathbf{x}_{k+1}. \end{aligned} \tag{4.2}$$

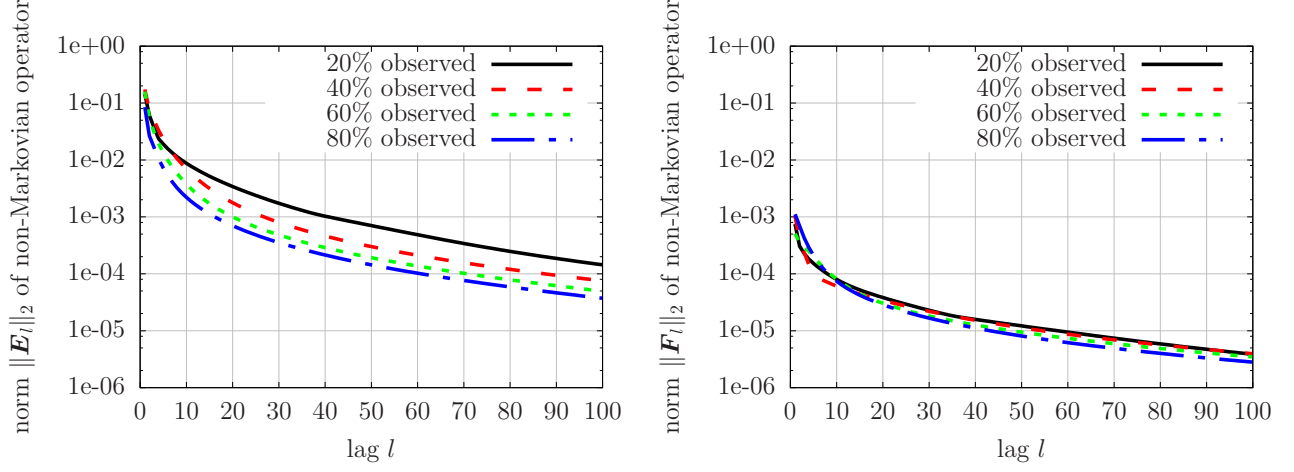
The basis  $\mathbf{V}$  of dimension  $n$  is obtained from snapshots of the observation trajectory (partially observed states) using the input  $\mathbf{u}^{\text{basis}}(t) = [\sin(2t), \sin(4t), \dots, \sin(10t)]^T \in \mathbb{R}^5$  at time  $t$  and  $K = 50001$  time steps. Together with the basis  $\mathbf{V}$ , the extended re-projection algorithm (Algorithm 2) is applied with  $N_r$  re-projection steps and  $K_r$  time steps per re-projection step, where inputs to the full model (4.2) at each time step are realizations of a  $p$ -dimensional random vector with independent components that are uniformly distributed in  $[0, 2]$ . The test input trajectory is given by  $\mathbf{u}^{\text{test}}(t) = [e^t \sin(1.75t), e^t \sin(3.5t), \dots, e^t \sin(8.75t)]^T \in \mathbb{R}^5$ .

##### 4.1.1. Recovering non-Markovian operators

Figure 4 plots the norms  $\|\mathbf{E}_l\|_2$  (left panel) and  $\|\mathbf{F}_l\|_2$  (right panel) of the non-Markovian operators (3.8) obtained with intrusive model reduction for lag values  $l = 1, \dots, 100$ . The four curves in each panel correspond to 20%, 40%, 60%, and 80% observed state components, respectively. The norms of the non-Markovian operators decay with increasing lag  $l$ , which supports truncating the non-Markovian term.

Set now the lag to  $L = 100$  with 60% of all state components observed. Consider stagewise inference (Algorithm 3) of the non-Markovian operators applied to trajectories sampled with the extended re-projection algorithm with  $N_r = 25$  re-projection steps  $K_r = 102$  time steps per re-projection step. This means that Proposition 3 applies because the full model (4.2) is linear in the state variable and we have numerically ensured that the data matrix is full rank. Thus, stagewise inference recovers the very same non-Markovian operators (3.8) that are obtained with intrusive model reduction. Figure 5 compares the norms of the operators  $\mathbf{E}_l, \mathbf{F}_l$  from intrusive model reduction (3.8) with the norms of the inferred operators  $\hat{\mathbf{E}}_l, \hat{\mathbf{F}}_l$  from stagewise inference. Notice that the norms coincide which is in agreement with Proposition 3.

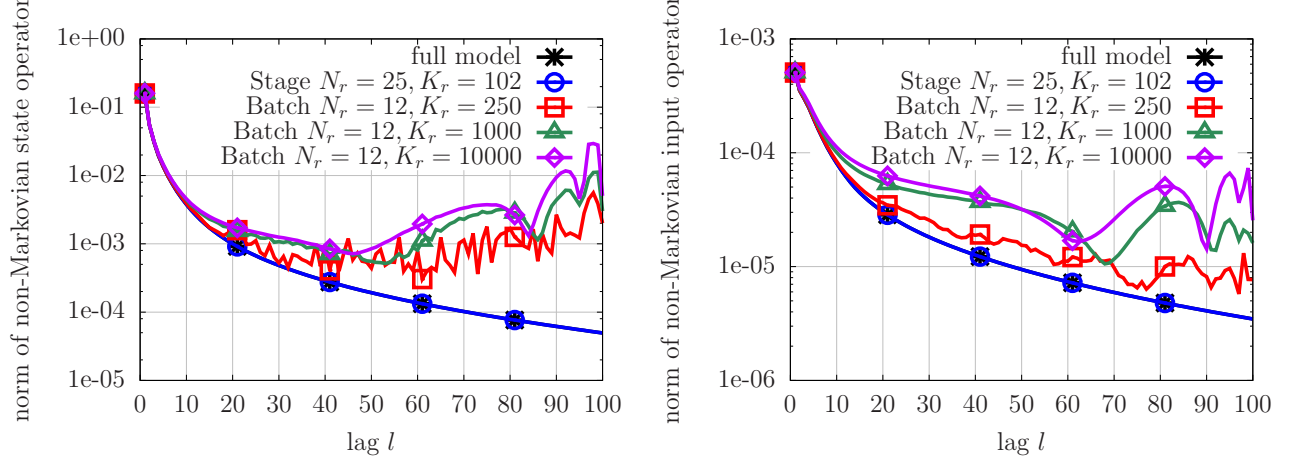
Consider now batch inference with trajectories sampled with the extended re-projection algorithms with  $N_r = 12$  and  $K_r \in \{250, 1000, 10000\}$  which correspond to larger training data sets than what is used with  $N_r = 25$  and  $K_r = 102$  for stagewise inference. The data sets generated for increasing values of  $K_r$  are nested. Since  $K_r > L + 2$ , Proposition 4 does not apply and therefore we do not expect that the operators inferred with batch inference coincide with the operators from intrusive model reduction in this setting. This is indicated by the results shown in Figure 5. Note, however, that the operators obtained by batch inference have norms that closely approximate the norm of the intrusive non-Markovian operators for  $l \leq 10$ .



(a) non-Markovian state operator  $\mathbf{E}_l$ , dimension  $n = 10$ .

(b) non-Markovian input operator  $\mathbf{F}_l$ , dimension  $n = 10$ .

Figure 4.: Convection-diffusion equation (Section 4.1). The norm of the non-Markovian operators decays with the lag  $l$  and thus supports truncating the non-Markovian term.



(a) Stage vs batch for  $\mathbf{E}_l$ ,  $n = 10$ , 60% observed.

(b) Stage vs batch for  $\mathbf{F}_l$ ,  $n = 10$ , 60% observed.

Figure 5.: Convection-diffusion equation (Section 4.1). Stagewise operator inference recovers the non-Markovian operators from intrusive model reduction (cf. Proposition 3) in this example. The batch approach only gives approximations of the non-Markovian operators of intrusive model reduction because  $K_r > L + 2$  (cf. Proposition 4).



#### 4.1.2. State error with number of time steps at most the lag of non-Markovian term

Let  $\mathbf{Z}_{\text{test}} = [\mathbf{z}_0^{\text{test}}, \dots, \mathbf{z}_{K-1}^{\text{test}}] \in \mathbb{R}^{r \times K}$  be the trajectory of observations under the test input  $\mathbf{u}^{\text{test}}(t)$  generated with the full model for  $K = 102$  time steps. Note that the number of rows of  $\mathbf{Z}_{\text{test}}$  depends on the number of observed state components. Figure 6 plots the relative projection error

$$\epsilon_z^{\text{proj}} = \frac{\|\mathbf{Z}_{\text{test}} - \mathbf{V}\mathbf{V}^T\mathbf{Z}_{\text{test}}\|_F}{\|\mathbf{Z}_{\text{test}}\|_F} \quad (4.3)$$

for trajectories  $\mathbf{Z}_{\text{test}}$  with 20% and 80% observed state components with reduced dimensions  $n = 4$  and  $n = 10$ . Additionally, Figure 6 plots the relative error of the observations

$$\epsilon_z^{\text{stage}} = \frac{\|\mathbf{Z}_{\text{test}} - \mathbf{V}\tilde{\mathbf{Z}}_{\text{test}}^{\text{stage}}\|_F}{\|\mathbf{Z}_{\text{test}}\|_F} \quad (4.4)$$

where  $\tilde{\mathbf{Z}}_{\text{test}}^{\text{stage}} \in \mathbb{R}^{n \times K}$  is the trajectory of observations computed with the learned reduced model with non-Markovian operators obtained with stagewise inference and test input  $\mathbf{u}^{\text{test}}(t)$ . Note that the observation error  $\epsilon_z^{\text{stage}}$  depends on the number of observed state components, the dimension  $n$  of the reduced space, and the lag  $L$ . Stagewise inference recovers the non-Markovian operators from intrusive model reduction in this example (cf. Proposition 3) and thus the relative observation error (4.4) of the learned reduced model equals the projection error if the lag  $L$  of the non-Markovian term satisfies  $L \geq K - 2$ . This corresponds to model (3.7) without truncation.

We now set  $L = 100$ . Figure 7 shows the absolute difference  $|\epsilon_z^{\text{stage}} - \epsilon_z^{\text{proj}}|$  between the relative stagewise observation error (4.4) and the projection error (4.3) after  $K = L + 2$  time steps for 20%, 40%, 60%, and 80% observed components. Define the analog of (4.4) for batch operator inference as

$$\epsilon_z^{\text{batch}} = \frac{\|\mathbf{Z}_{\text{test}} - \mathbf{V}\tilde{\mathbf{Z}}_{\text{test}}^{\text{batch}}\|_F}{\|\mathbf{Z}_{\text{test}}\|_F} \quad (4.5)$$

where  $\tilde{\mathbf{Z}}_{\text{test}}^{\text{batch}} \in \mathbb{R}^{n \times K}$  is the trajectory of observations computed with the reduced model whose non-Markovian operators are learned simultaneously. Also shown in Figure 7 is the absolute difference  $|\epsilon_z^{\text{batch}} - \epsilon_z^{\text{proj}}|$  between the relative batch observation error (4.5) and the projection error (4.3). The results demonstrate that the reduced model with non-Markovian terms obtained with stagewise operator inference in this example achieves a state error (4.4) that equals, up to numerical errors, the projection error (4.3). Meanwhile, batch inference in this situation does not recover the non-Markovian operators of intrusive model reduction and thus the corresponding difference between observation (4.5) and projection error (4.3) is higher than with stagewise inference in this example.

#### 4.1.3. State error with number of time steps larger than lag of non-Markovian term

We now consider  $K = 50001$  time steps and a lag  $L \ll K$  so that the observation error of a reduced model obtained with stagewise inference does not necessarily coincide with the projection error (4.3). Figure 8 shows the observation errors (4.4) and (4.5) for stagewise and batch inference for 20% observed states with dimension  $n = 10$  (left panel) and 80% observed states with dimension  $n = 4$  (right panel). Batch inference with  $N_r = 12$  and  $K_r = 250$  leads to a model with a large observation error  $\epsilon_z^{\text{batch}}$  near  $L = 20$  for 20% observed states and dimension  $n = 10$ ; this is consistent with the graphs of Figure 5 that report that the norm of the learned operators can differ significantly from the norm of the intrusive operators. However, if the number of time steps  $K_r$  sampled from the full model with re-projection is increased to  $K_r = 1000$  and  $K_r = 10000$ , the non-Markovian models obtained with batch inference outperform stagewise inference in this example. While stagewise inference is confined to recovering the non-Markovian operators of intrusive model reduction, batch inference offers more flexibility that can lead to a non-Markovian model that achieves lower errors than the stagewise approach. The right panel of Figure 8 shows results for 80%

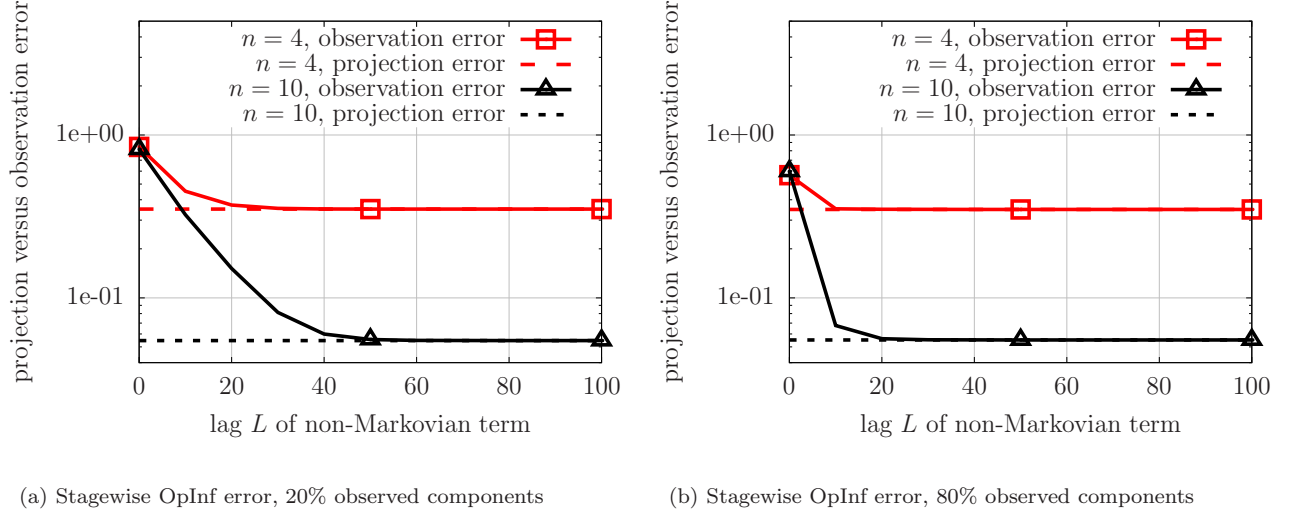


Figure 6.: Convection-diffusion equation (Section 4.1). The projection error  $\epsilon_z^{\text{proj}}$  (4.3) and the stagewise observation error  $\epsilon_z^{\text{stage}}$  (4.4) are equal at the maximal lag  $L = 100$  since  $L = K - 2$ .

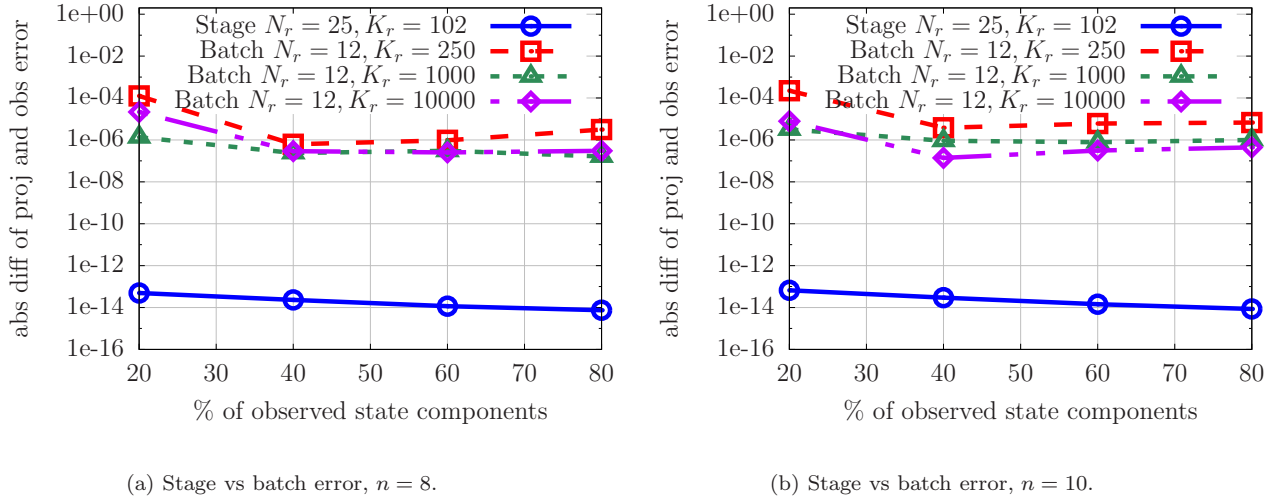


Figure 7.: Convection-diffusion equation (Section 4.1). If the lag of the non-Markovian term satisfies  $L = K - 2$ , the observation error corresponding to stagewise operator inference coincides with the projection error. Since the batch inference does not recover the non-Markovian operators of intrusive model reduction in this example, the batch observation error differs from the projection error.

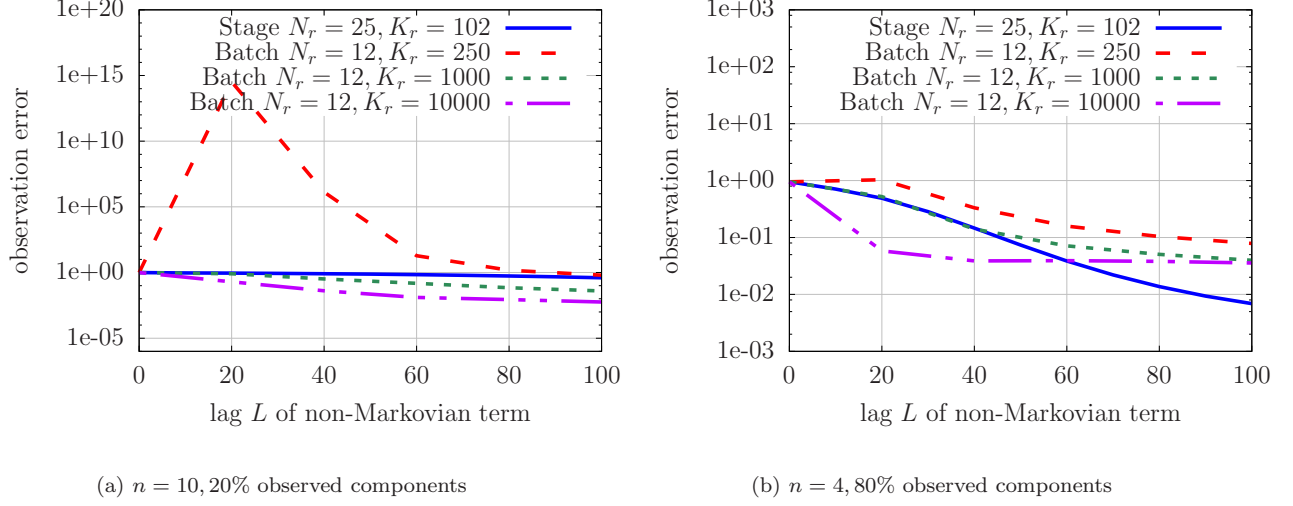


Figure 8.: Convection-diffusion equation (Section 4.1). The non-Markovian model trained via batch inference may produce unstable dynamics (left) or result in a model whose observation error stagnates despite an increase in the lag  $L$  (right).

observed state components and dimension  $n = 4$  where batch inference leads to a non-Markovian model whose observation error does not improve as the lag  $L$  is increased. In contrast, stagewise operator inference yields a non-Markovian model that eventually attains a lower error with increasing lag  $L$ .

We now fix the lag to  $L = 100$  and consider the relative observation errors shown in Figure 9 over the number of observed components. Here, batch inference was also applied to trajectories sampled with the extended re-projection algorithm with  $N_r = 25$  and  $K_r = \{500, 1000, 10000\}$  which represent nested training data sets. The results demonstrate that batch inference of the operators seems to outperform stagewise learning if training is done on a sufficiently large data set. In terms of computational cost, stagewise uses  $N_r \times K_r = 25 \times 102$  data points whereas for batch inference, up to  $N_r \times K_r = 25 \times 10000$  data points are used.

#### 4.1.4. Output error

Consider now the output trajectory  $\mathbf{Y}_{\text{test}}$  of the full model with the test input trajectory and let  $\tilde{\mathbf{Y}}_{\text{test}}^{\text{Markovian}}$  be the output trajectory obtained with a Markovian reduced model learned via non-intrusive model reduction from partially observed state trajectories, i.e. the lag of the non-Markovian term is  $L = 0$ . Denote by  $\tilde{\mathbf{Y}}_{\text{test}}^{\text{stage}}$  the output trajectory computed with the non-Markovian reduced model learned via stagewise inference. Note that  $\tilde{\mathbf{Y}}_{\text{test}}^{\text{stage}}$  depends on the lag  $L$ , the dimension  $n$  of the reduced space, and the number of observed state components. Figure 10 shows the output trajectory of the full model  $\mathbf{Y}_{\text{test}}$ , the trajectory  $\tilde{\mathbf{Y}}_{\text{test}}^{\text{Markovian}}$  of the Markovian reduced model ( $L = 0$ ), and the trajectory  $\tilde{\mathbf{Y}}_{\text{test}}^{\text{stage}}$  of the non-Markovian reduced model with stagewise operator inference and lag  $L = 100$  for  $n = 8$  and 60% observed state components (left panel) and for  $n = 10$  and 20% observed state components (right panel). The learned non-Markovian reduced model provides a more accurate approximation of the full model output than the traditional Markovian reduced model.

Consider now the relative output error

$$\epsilon^{\text{output}}(\tilde{\mathbf{Y}}) = \frac{\|\mathbf{Y}_{\text{test}} - \tilde{\mathbf{Y}}\|_F}{\|\mathbf{Y}_{\text{test}}\|_F} \quad (4.6)$$

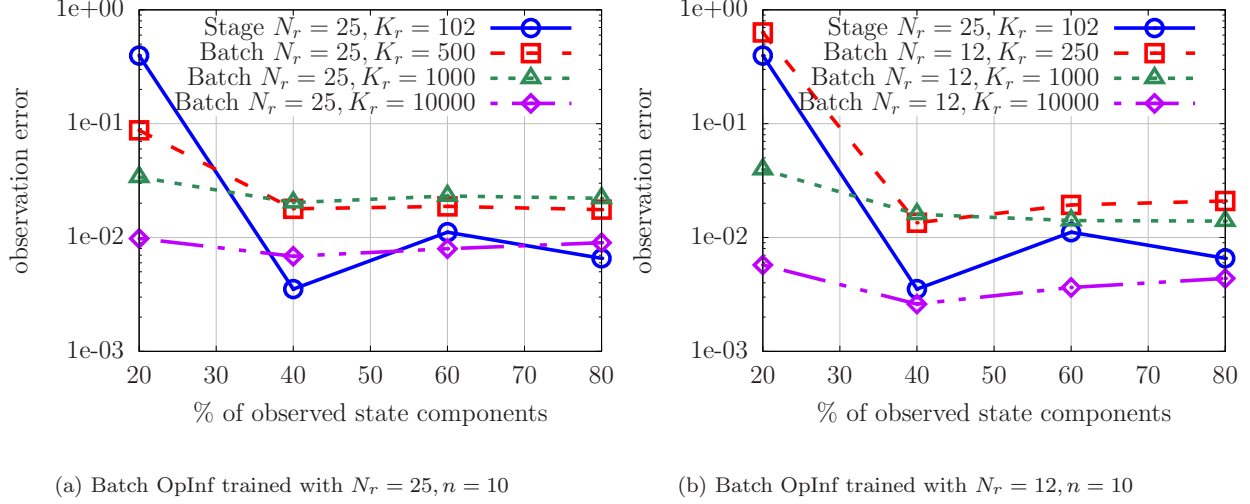


Figure 9.: Convection-diffusion equation (Section 4.1). Stagewise inference achieves a smaller observation error than batch inference in this example except when the latter is trained on data sets with sufficiently large  $K_r$ .

where  $\tilde{Y}$  is a trajectory from a reduced model for the output. Figure 11 shows the output error (4.6) of the trajectory computed with the Markovian reduced model ( $\tilde{Y} = \tilde{Y}_{\text{test}}^{\text{Markovian}}$ ) and compares it to the error (4.6) of the trajectories computed with the stagewise learned non-Markovian reduced model with lag  $L = 100$  ( $\tilde{Y} = \tilde{Y}_{\text{test}}^{\text{stage}}$ ) for 20%, 40%, 60%, and 80% observed state components and dimension  $n = 8$  (left panel) and  $n = 10$  (right panel). The non-Markovian reduced model achieves errors of orders of magnitude lower than the Markovian reduced model in this example.

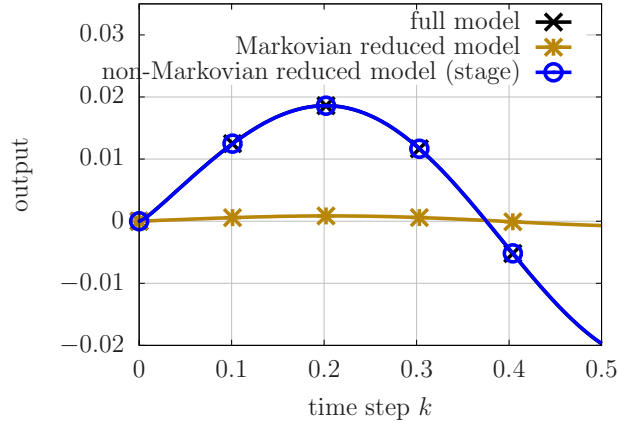
If instead 99% of the state components of the full model were observed, the approximation quality of the learned Markovian reduced model for the output improves as depicted in Figure 12 for  $n = 10$  (left panel) and  $n = 8$  (right panel). We also emphasize that, depending on the nature of the output, it is still possible for the learned Markovian model to provide a poor approximation despite increasing the rate of observed components to 99%. This underscores the benefits of incorporating a non-Markovian term to the reduced model.

## 4.2. Diffusion-reaction equation

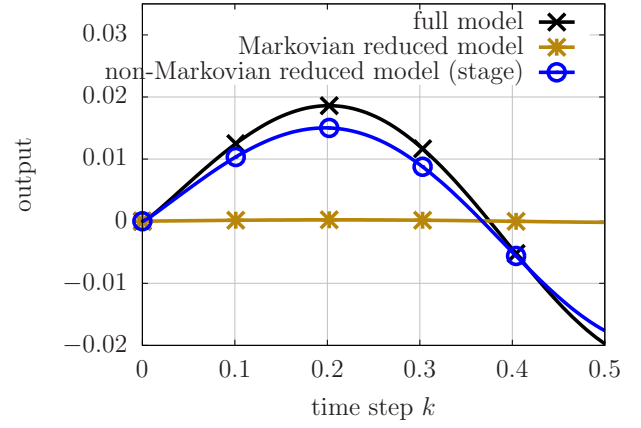
We now learn a non-Markovian reduced model of a parametric nonlinear polynomial system arising from the diffusion-reaction example in [33]. Set the spatial domain as  $\Omega = (0, 1) \times (0, 1)$  with boundary  $\partial\Omega$ , the time domain as  $t \in (0, 100)$ , and the parameter domain as  $\mu \in \mathcal{D} = [1, 1.5]$ . For  $(\xi_1, \xi_2, t) \in \Omega \times (0, 100)$  and  $\mu \in \mathcal{D}$ , the PDE is described by

$$\begin{aligned} \frac{\partial}{\partial t} x(\xi_1, \xi_2, t; \mu) &= -\Delta x(\xi_1, \xi_2, t; \mu) + (\sin(2\pi\xi_1) \sin(2\pi\xi_2)/10)u(t) + g(x(\xi_1, \xi_2, t; \mu); \mu), \\ \nabla x(\xi_1, \xi_2, t; \mu) \cdot \mathbf{n} &= 0 \quad \text{for } (\xi_1, \xi_2) \in \partial\Omega, \\ x(\xi_1, \xi_2, 0; \mu) &= 0 \end{aligned} \quad (4.7)$$

where  $u(t)$  is the input and  $g$  is defined by  $g(x; \mu) = -(0.1 \sin(\mu) + 2)e^{-2.7\mu^2}(1 + 1.8\mu x + 1.62\mu^2 x^2)$ . The PDE (4.7) is spatially discretized on a grid with 64 equidistant points in  $\xi_1$  and  $\xi_2$  via the finite difference method. It is then temporally discretized with forward Euler using time step size of  $\delta t = 10^{-2}$  to obtain the

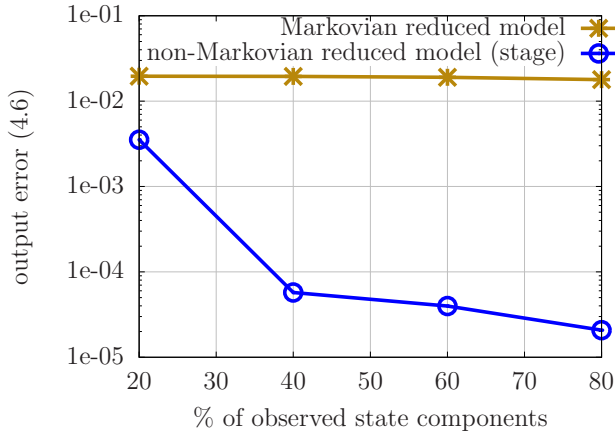


(a)  $n = 8$ , 60% observed components

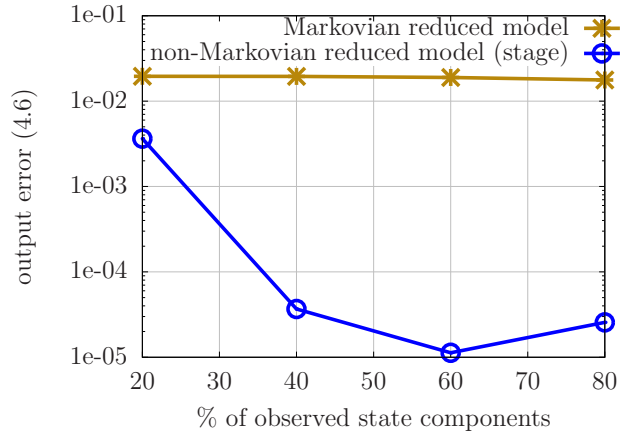


(b)  $n = 10$ , 20% observed components

Figure 10.: Convection-diffusion equation (Section 4.1). The correction terms in the non-Markovian reduced model improve the approximation offered by the Markovian reduced model.



(a) output,  $n = 8$



(b) output,  $n = 10$

Figure 11.: Convection-diffusion equation (Section 4.1). The non-Markovian reduced model for the output achieves errors which are orders of magnitude lower than the Markovian model for this example.

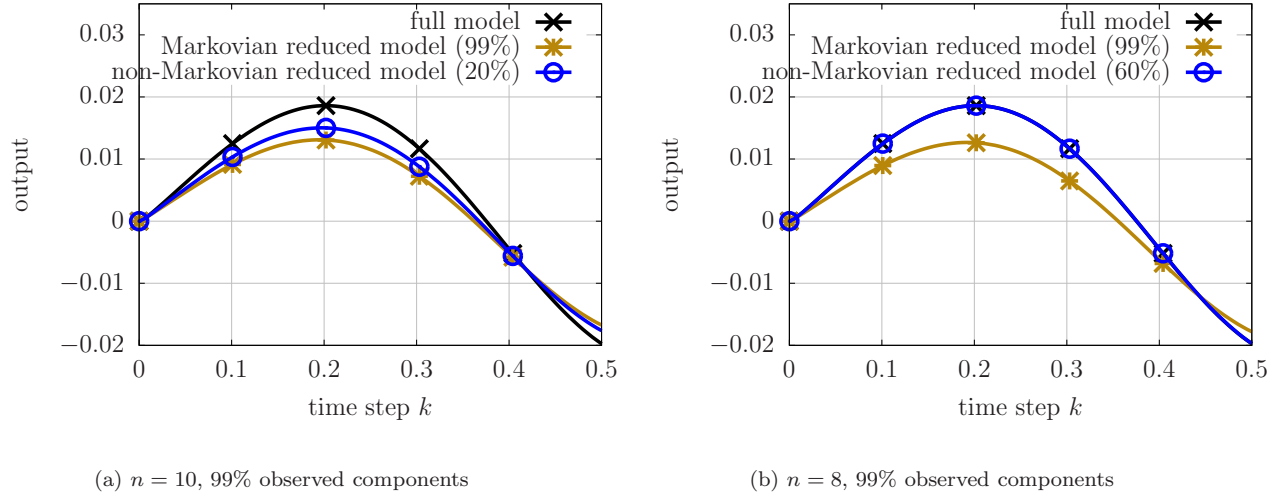


Figure 12.: Convection-diffusion equation (Section 4.1). A Markovian reduced model needs up to 99% of the components of the states to be observed to achieve comparable accuracy as a non-Markovian reduced model learned from only 20% observed components in this example.

parametric full model

$$\mathbf{x}_{k+1}(\mu) = \mathbf{A}_1(\mu)\mathbf{x}_k(\mu) + \mathbf{A}_2(\mu)\mathbf{x}_k^2(\mu) + \mathbf{A}_3(\mu)\mathbf{x}_k^3(\mu) + \mathbf{B}u_k, \quad k = 0, \dots, K-1, \quad (4.8)$$

with  $\mathbf{x}_k(\mu) \in \mathbb{R}^N$ ,  $N = 64^2$  and  $u_k \in \mathbb{R}$ .

The inputs for basis generation, training, and testing are independently sampled at each time step from a uniform distribution on  $[-3, 996]$ . The parameters for basis generation and training are  $\mu_i^{\text{train}} = 1 + (i-1)/18$  for  $i = 1, \dots, 10$  while the test parameters are  $\mu_i^{\text{test}} = 1 + (i-1)/16$  for  $i = 1, \dots, 9$ . The Markovian and non-Markovian operators are interpolated in the testing phase as described in Section 2.1 and [14]. For each training parameter  $\mu_1^{\text{train}}, \dots, \mu_{10}^{\text{train}}$ , we generate 20 trajectories with the extended re-projection algorithm with a total of  $N_r$  re-projection steps and  $K_r$  time steps per re-projection step; each trajectory corresponds to different realizations of the input random variable. For fixed  $N_r$ , a fair comparison across different data sets is aimed for by nesting the data sets with respect to increasing  $K_r$ . The lag  $L = 40$  and  $N_r, K_r$  are always chosen so that the resulting data matrices for the least squares problems are numerically full rank.

#### 4.2.1. State error

Consider the projection error averaged over  $N_\mu$  parameters

$$e_\mu^{\text{proj}} = \frac{1}{N_\mu} \sum_{i=1}^{N_\mu} \frac{\|\mathbf{Z}(\mu_i) - \mathbf{V}\mathbf{V}^T\mathbf{Z}(\mu_i)\|_F}{\|\mathbf{Z}(\mu_i)\|_F} \quad (4.9)$$

where  $\mathbf{Z}(\mu_i)$  is an observation trajectory for fixed parameter value  $\mu_i$ . Analogously, define the observation error as

$$e_\mu^{\text{obs}}(\tilde{\mathbf{Z}}) = \frac{1}{N_\mu} \sum_{i=1}^{N_\mu} \frac{\|\mathbf{Z}(\mu_i) - \mathbf{V}\tilde{\mathbf{Z}}(\mu_i)\|_F}{\|\mathbf{Z}(\mu_i)\|_F} \quad (4.10)$$

where  $\tilde{\mathbf{Z}}(\mu_i)$  is an observation trajectory obtained with a learned reduced model for parameter  $\mu_i$ . The projection error  $e_\mu^{\text{proj}}$  depends on the dimension  $n$  of the reduced space  $\mathcal{V}$ , the number of observed state

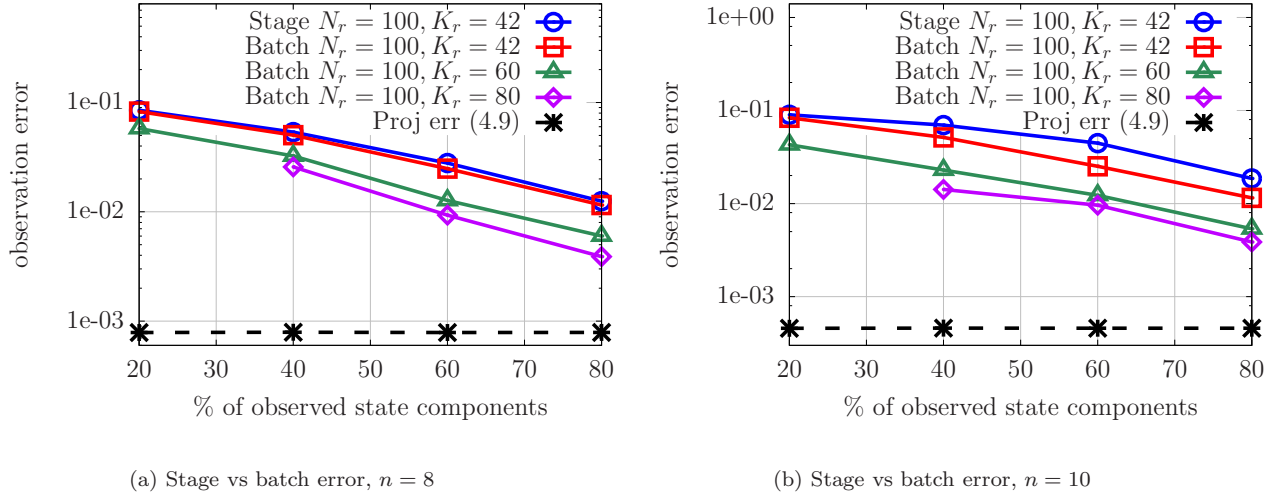


Figure 13.: Diffusion-reaction equation (Section 4.2). For nonlinear polynomial full models, neither mode of operator inference necessarily recovers the non-Markovian operators of intrusive model reduction. In this example, for a fixed number of re-projection steps, batch inference can achieve a lower error if the trajectories in the training data are sufficiently long.

components  $r$ , and whether the average is taken over the training parameters  $\{\mu_i^{\text{train}}\}_{i=1}^{10}$  or the test parameters  $\{\mu_i^{\text{test}}\}_{i=1}^9$ . The observation error (4.10) additionally depends on all parameters that the reduced model depends on such as the lag  $L$  and the data set used to train the reduced model.

Figure 13 shows the projection error (4.9) and the observation error (4.10) over the test parameters and inputs for dimensions  $n = 8$  (left panel) and  $n = 10$  (right panel). Stagewise inference is applied to data sets derived from re-projection with  $N_r = 100$  re-projection steps and  $K_r = 42$  time steps per re-projection step; batch inference is applied to data sets with  $K_r = 42, 60, 80$  time steps per re-projection step. Recall that these data sets are nested. The results reported in Figure 13 show that a non-Markovian reduced model learned with batch inference can achieve a lower error if the data set is sufficiently large, which is in agreement with the results reported for the linear full model in Section 4.1.3. Notice that if  $K_r = 42$ , both modes of operator inference are applied to the same training data. Missing markers indicate that models led numerically to NaN (Not a Number) after a finite number of time steps, e.g., for the non-Markovian model obtained with batch inference from 20% observed states and  $n = 8$  dimensions. Unlike in the linear case, there is no guarantee here that either mode of operator inference is able to recover the non-Markovian operators from intrusive model reduction because the full model is nonlinear polynomial and we consider linear non-Markovian terms only (cf. Section 3.2).

Figure 14 shows the projection and observation error for dimension  $n = 10$  with re-projection parameters  $N_r = 12, K_r \in \{100, 500, 2500\}$  for batch inference. The left panel shows the errors over the training parameters and inputs while the right panel shows the errors over the test parameters and inputs. The results show that batch inference achieves an up to one order of magnitude lower test error than stagewise inference in this experiment. This is because batch inference is trained on longer trajectories (larger  $K_r$  for fixed  $N_r$ ) unlike stagewise inference which is constrained to utilize data only up to  $K_r = L + 2$  time steps if the non-Markovian term has lag  $L$ . Notice also that batch inference starts to overfit for  $K_r = 2500$ , which we think is because the full model has reached steady state by 2500 time steps and so the additional training data provided by increasing the number of time steps  $K_r$  per re-projection step is skewing the least squares problem towards the steady state behavior of the full model present in the training data set.



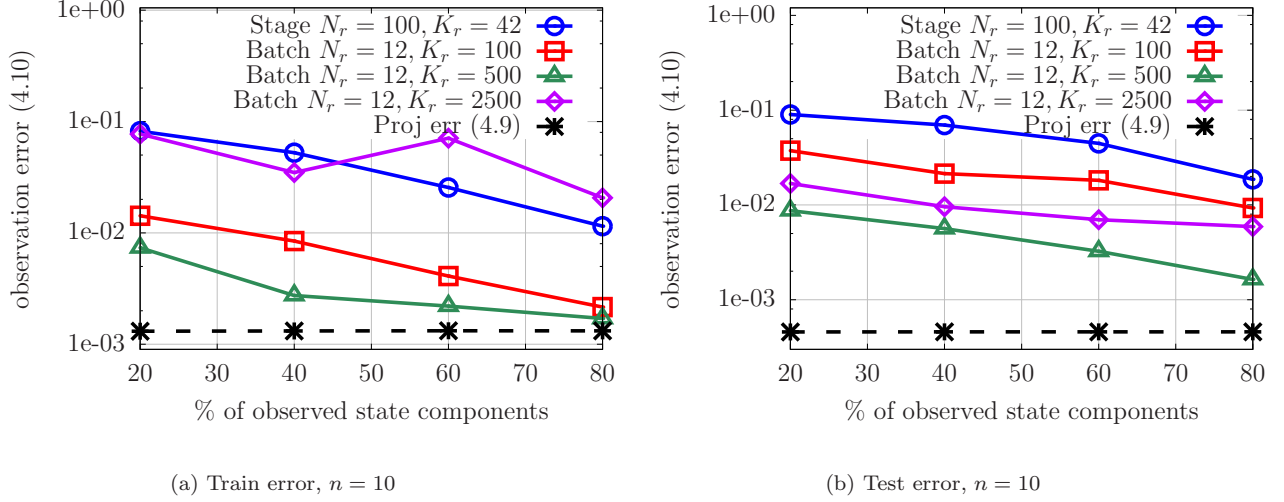


Figure 14.: Diffusion-reaction equation (Section 4.2). Batch inference is flexible in that it can be applied to training data with long trajectories (large  $K_r$ ) to learn a non-Markovian reduced model with lag  $L$  in contrast to stagewise inference which only utilizes data up to  $K_r = L + 2$  time steps. In this example, this results in a non-Markovian model learned with batch inference that achieves lower errors than its stagewise counterpart.

### 4.3. Chafee-Infante equation

Consider the Chafee-Infante equation on the spatial domain  $\Omega = (0, 1)$  and time domain  $t \in (0, 4)$ . For  $(\xi, t) \in \Omega \times (0, 4)$ , if  $u(t)$  is the input, the PDE satisfies

$$\begin{aligned} \frac{\partial}{\partial t} x(\xi, t) &= \frac{\partial^2}{\partial \xi^2} x(\xi, t) - x^3(\xi, t) + x(\xi, t), \\ \frac{\partial}{\partial \xi} x(1, t) &= 0, \\ x(0, t) &= u(t), \\ x(\xi, 0) &= 0. \end{aligned} \quad (4.11)$$

The quantity of interest is modeled as  $y(t) = x(1, t)$  which is the solution at the boundary point  $\xi = 1$ . An approximation to the solution of (4.11) is sought at 128 equidistant spatial nodes in  $(0, 1)$  via the finite difference method. Forward Euler is then employed to discretize the PDE temporally with  $\delta t = 10^{-5}$ . This gives the full model

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_1 \mathbf{x}_k + \mathbf{A}_2 \mathbf{x}_k^2 + \mathbf{A}_3 \mathbf{x}_k^3 + \mathbf{B} u_k, \quad k = 0, \dots, K-1, \\ y_{k+1} &= \mathbf{C} \mathbf{x}_{k+1} \end{aligned} \quad (4.12)$$

with  $\mathbf{x}_k \in \mathbb{R}^N$ ,  $N = 128$ , and  $u_k, y_k \in \mathbb{R}$ .

The basis matrix  $\mathbf{V}$  for the observations is computed with inputs sampled from a uniform distribution on  $[0, 10]$ . The training data set is generated as follows. First, trajectories  $i = 1, \dots, 15$  are obtained with re-projection using the inputs  $u_{k,i}^{\text{train}} = 10\theta_k^{(i)}(\cos(50\pi k\delta t\gamma_k^{(i)}) + 1)$  for the  $i$ -th trajectory, where  $\theta_k^{(i)}, \gamma_k^{(i)}$  are realizations of a random variable with uniform distribution on  $[0, 1]$ . Second, training trajectories  $i = 16, \dots, 30$  are obtained with inputs that are realizations of a uniform distribution on  $[0, 10]$ . The lag is set to  $L = 60$  and the non-Markovian operators are then learned as a batch with  $N_r = 600$  (20 re-projection

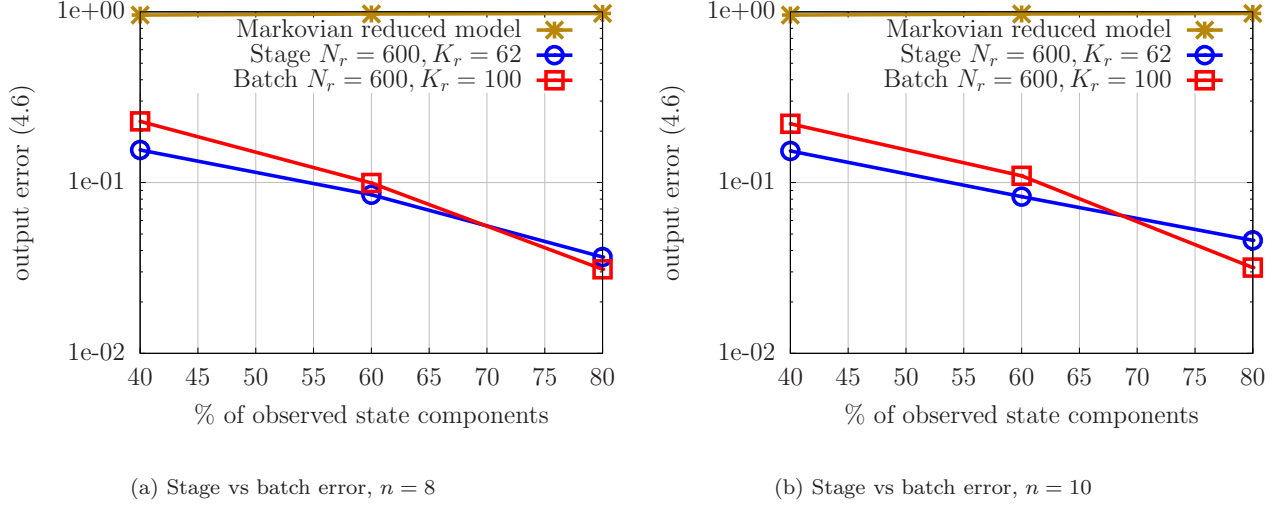


Figure 15.: Chafee-Infante equation (Section 4.3). The non-Markovian reduced models achieve an error of more than one order of magnitude lower than its Markovian counterpart.

steps per input) and  $K_r = 100$ , and in a stagewise manner on the subset generated by the  $K_r = 62$  time steps. The test input is  $u^{\text{test}}(t) = 5(\sin(\pi t) + 1)$ .

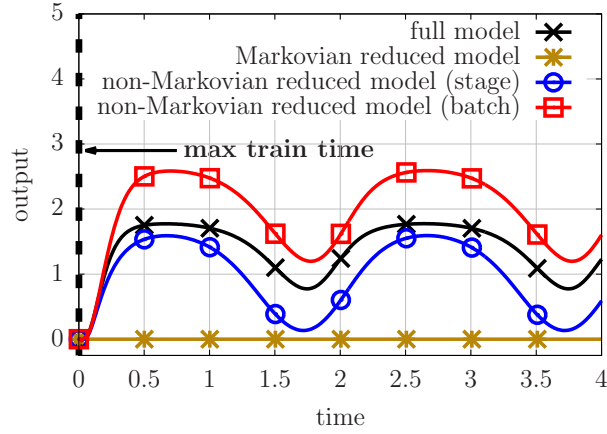
#### 4.3.1. Output error

Consider the output error defined analogously as (4.6) for the non-Markovian reduced model obtained with batch and stagewise inference; see Figure 15. The results indicate that a Markovian reduced model fails to make predictions about the full model output when one only observes partial components of the state. In contrast, the proposed non-Markovian reduced models learned with stagewise or batch inference achieve more than one order of magnitude improvement compared to the Markovian reduced model.

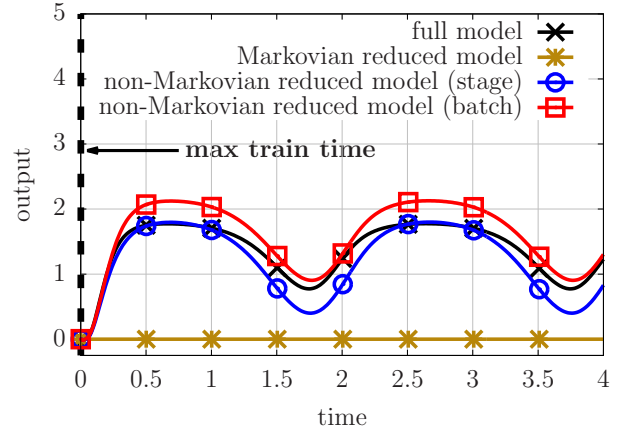
Figure 16 compares the full model outputs over time with the approximations given by the Markovian and the non-Markovian reduced models learned from 40%, 60%, and 80% observed state components. In agreement with the results shown in Figure 2, the Markovian reduced model is unable to capture the oscillatory behavior of the output of the full model. In contrast, the non-Markovian reduced models capture the oscillatory output behavior even if only 40% of the state components are observed and provide a close approximation as this rate is increased to 80% of the state components. In the training data, each re-projection step is succeeded by at most  $K_r = 100$  time steps with the full model. This corresponds to  $t = 0.001$  and is plotted as a dashed vertical line in each panel of Figure 16. This training time length is not long enough to cover the oscillatory nature of the output. Yet, the resulting approximation with the non-Markovian reduced model provides a reasonable approximation far outside of the training regime, which emphasizes that the dynamics of the underlying system are learned rather than mere interpolations between training data samples.

## Acknowledgments

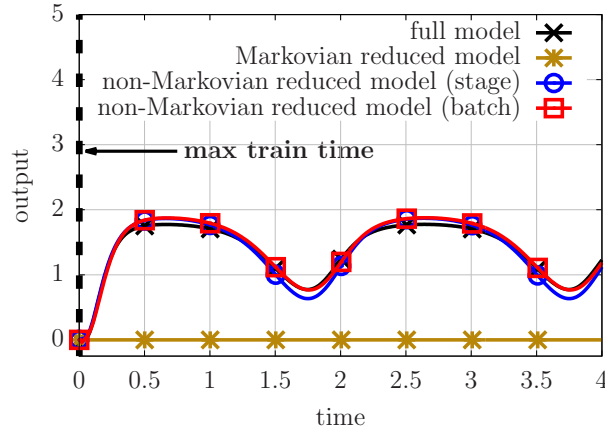
This work was partially supported by US Department of Energy, Office of Advanced Scientific Computing Research, Applied Mathematics Program (Program Manager Dr. Steven Lee), DOE Award DESC0019334, and by the National Science Foundation under Grant No. 1901091 and under Grant No. 1761068.



(a) 40% observed state components



(b) 60% observed state components



(c) 80% observed state components

Figure 16.: Chafee-Infante equation (Section 4.3). The non-Markovian reduced model for the output captures the oscillatory nature of the full model output even though the models are learned from much shorter training trajectories than the prediction end time. Thus, the learned non-Markovian model provide reasonable approximations far beyond the training regime in this example.

## Appendix A Error analysis of the non-Markovian reduced model

We build on the analysis in Section 3.1.3 and demonstrate numerically that Markovian reduced models for linear autonomous full systems can achieve lower errors than the proposed non-Markovian reduced models (3.12). To motivate the numerical experiments that follow, consider a full model with dimension  $N = 2$  and with observed state of dimension  $r = 1$  and reduced dimension  $n = 1$ . A sufficient condition for which the proposed reduced model with non-Markovian term (3.12) yields a lower error than a Markovian reduced model is when  $\mathbf{A}_1$  is symmetric positive definite. To see this, observe that for positive integers  $l$  ( $l \in \mathbb{Z}^+$ ),  $\tilde{\mathbf{A}}_1, \mathbf{E}_l \in \mathbb{R}$  and that  $\tilde{\mathbf{A}}_1 > 0$ ,

$$\mathbf{E}_l = (\mathbf{Q}^T \mathbf{A}_1 \mathbf{Q}^\perp)^2 ((\mathbf{Q}^\perp)^T \mathbf{A}_1 \mathbf{Q}^\perp)^{l-1} > 0.$$

Provided that  $\tilde{\mathbf{z}}_0 = \tilde{\mathbf{z}}_0^{(0)} = \tilde{\mathbf{z}}_0^{(L)}$ , for fixed  $k \in \mathbb{Z}^+$ , if  $\tilde{\mathbf{z}}_k^{(0)}$  and  $\tilde{\mathbf{z}}_k^{(L)}$  are expressed in terms of the initial condition  $\tilde{\mathbf{z}}_0$ , algebraic calculations show that

$$\|\tilde{\mathbf{z}}_k - \tilde{\mathbf{z}}_k^{(L)}\|_2 \leq \|\tilde{\mathbf{z}}_k - \tilde{\mathbf{z}}_k^{(0)}\|_2$$

since  $\tilde{\mathbf{A}}_1, \mathbf{E}_l$  are positive for all  $l \in \mathbb{Z}^+$ . Therefore, since

$$\begin{aligned} \|\mathbf{z}_k - \mathbf{V} \tilde{\mathbf{z}}_k^{(0)}\|_2 &= \|\mathbf{z}_k - \mathbf{V} \tilde{\mathbf{z}}_k\|_2 + \|\mathbf{V}(\tilde{\mathbf{z}}_k - \tilde{\mathbf{z}}_k^{(0)})\|_2, \\ \|\mathbf{z}_k - \mathbf{V} \tilde{\mathbf{z}}_k^{(L)}\|_2 &= \|\mathbf{z}_k - \mathbf{V} \tilde{\mathbf{z}}_k\|_2 + \|\mathbf{V}(\tilde{\mathbf{z}}_k - \tilde{\mathbf{z}}_k^{(L)})\|_2, \end{aligned}$$

we conclude that

$$\|\mathbf{z}_k - \mathbf{V} \tilde{\mathbf{z}}_k^{(L)}\|_2 \leq \|\mathbf{z}_k - \mathbf{V} \tilde{\mathbf{z}}_k^{(0)}\|_2,$$

i.e., the reduced model with non-Markovian term achieves a lower error than its Markovian counterpart.

However, the symmetric positive definiteness of the matrix  $\mathbf{A}_1$  is insufficient when  $N > 2, n > 1$ . To see this, consider the following two examples with lag  $L = 1$ . A numerical implementation is available in Python<sup>1</sup> which reproduces Figure 17 below. We set  $N = 10, n = 2$  and consider 30% observed state components for the first example while for the second, we use  $N = 50, n = 40$  and consider 95% observed state components. In both cases, the initial condition  $\tilde{\mathbf{z}}_0$  is chosen such that its components are realizations of independent standard normal random variables. The initial condition for the full system is then  $\mathbf{x}_0 = \mathbf{Q} \tilde{\mathbf{z}}_0$  so that  $\mathbf{x}_0$  satisfies  $(\mathbf{Q}^\perp)^T \mathbf{x}_0 = \mathbf{0}_{N-n}$ .

The symmetric positive definite matrix  $\mathbf{A}_1$  is constructed as follows. Its eigenvalues are sampled from a uniform distribution on  $(0, 1)$  to ensure that the system is stable. Its orthonormal eigenvectors are then chosen to be the eigenvectors of  $(\mathbf{R} + \mathbf{R}^T)/2$  where  $\mathbf{R}^{N \times N}$  is a matrix whose entries are independently sampled from a uniform distribution on  $(0, 10)$ . The components with indices 1,6,10 of the full state are

---

<sup>1</sup><https://github.com/wayneisaacuy/OpInfPartialObs>

observed in the first example with the initial condition and basis and system matrices given by

$$\begin{aligned} \mathbf{x}_0 &= [-0.5960 \ 0 \ 0 \ 0 \ 0 \ 1.0333 \ 0 \ 0 \ 0 \ 0.8346]^T, \\ \mathbf{V} &= \begin{bmatrix} -0.9889 & 0.0294 \\ 0.0767 & -0.7374 \\ -0.1269 & -0.6748 \end{bmatrix}, \\ \mathbf{V}^\perp &= \begin{bmatrix} -0.1453 \\ -0.6710 \\ 0.7270 \end{bmatrix}, \\ \mathbf{A}_1 &= \begin{bmatrix} 0.3603 & 0.0184 & -0.2192 & 0.0435 & -0.1624 & -0.0602 & 0.0758 & -0.0872 & 0.0634 & -0.0252 \\ 0.0184 & 0.2907 & -0.1049 & 0.1334 & 0.0087 & 0.0951 & -0.0594 & -0.0602 & -0.0717 & 0.1366 \\ -0.2192 & -0.1049 & 0.2978 & -0.1695 & 0.0887 & 0.0648 & -0.0924 & 0.0624 & -0.0213 & 0.0079 \\ 0.0435 & 0.1334 & -0.1695 & 0.3700 & 0.0529 & -0.0074 & 0.1284 & 0.0196 & -0.0115 & 0.0273 \\ -0.1624 & 0.0087 & 0.0887 & 0.0529 & 0.4582 & 0.0913 & 0.1194 & -0.0375 & 0.0449 & 0.1615 \\ -0.0602 & 0.0951 & 0.0648 & -0.0074 & 0.0913 & 0.4311 & -0.0781 & -0.0263 & 0.2070 & 0.1714 \\ 0.0758 & -0.0594 & -0.0924 & 0.1284 & 0.1194 & -0.0781 & 0.3804 & 0.0296 & 0.1548 & -0.1197 \\ -0.0872 & -0.0602 & 0.0624 & 0.0196 & -0.0375 & -0.0263 & 0.0296 & 0.3470 & 0.1123 & -0.1761 \\ 0.0634 & -0.0717 & -0.0213 & -0.0115 & 0.0449 & 0.2070 & 0.1548 & 0.1123 & 0.5707 & -0.1059 \\ -0.0252 & 0.1366 & 0.0079 & 0.0273 & 0.1615 & 0.1714 & -0.1197 & -0.1761 & -0.1059 & 0.3255 \end{bmatrix}. \end{aligned}$$

The details of the second example are provided in the repository<sup>1</sup>.

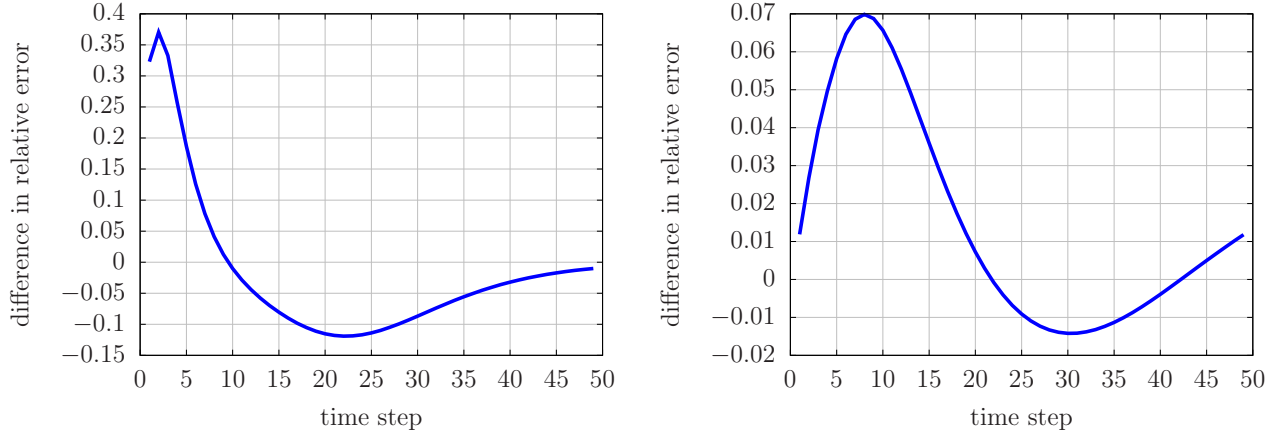
Figure 17 shows the difference in the relative error

$$\frac{1}{\|\mathbf{z}_k\|_2} (\|\mathbf{z}_k - \mathbf{V}\tilde{\mathbf{z}}_k^{(0)}\|_2 - \|\mathbf{z}_k - \mathbf{V}\tilde{\mathbf{z}}_k^{(L)}\|_2)$$

against the time step  $k$ . At certain time instances, the Markovian reduced model has a smaller error (negative values on the  $y$ -axis) than the model with non-Markovian term of lag  $L = 1$ . Thus, the conclusion we derived for  $N = 2, n = 1$  does not generalize and these examples show that it is possible that the Markovian model gives a more accurate approximation than the truncated non-Markovian model even if the matrix  $\mathbf{A}_1$  is symmetric positive definite. A more rigorous analysis is warranted but is beyond the scope of this work.

## References

- [1] A. C. Antoulas and B. D. Q. Anderson. On the scalar rational interpolation problem. *IMA Journal of Mathematical Control & Information*, 3(2-3):61–88, 1986.
- [2] A. C. Antoulas, I. V. Gosea, and A. C. Ionita. Model reduction of bilinear systems in the Loewner framework. *SIAM Journal on Scientific Computing*, 38(5):B889–B916, 2016.
- [3] C. Beattie and S. Gugercin. Interpolatory projection methods for structure-preserving model reduction. *Systems & Control Letters*, 58(3):225–232, 2009.
- [4] C. Beattie and S. Gugercin. Realization-independent  $\mathcal{H}_2$ -approximation. In *Proc. IEEE Conf. Decis. Control*, pages 4953–4958, Maui, HI, USA, 2012.
- [5] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox. Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms. *Computer Methods in Applied Mechanics and Engineering*, 2020. (accepted).
- [6] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.



(a)  $N = 10, n = 2$ , 30% observed state components

(b)  $N = 50, n = 40$ , 95% observed state components

Figure 17.: The Markovian model yields a more accurate approximation of the observed state dynamics at certain time points than the model with truncated non-Markovian term in this example.

- [7] S. A. Billings. *Nonlinear system identification : NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, Inc., Chichester, West Sussex, United Kingdom, 2013.
- [8] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.
- [9] K. P. Champion, S. L. Brunton, and J. N. Kutz. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1):312–333, 2019.
- [10] A. Chorin and O. H. Hald. *Stochastic Tools in Mathematics and Science*. Springer New York, 2009.
- [11] A. Chorin and P. Stinis. Problem reduction, renormalization, and memory. *Commun. Appl. Math. Comput. Sci.*, 1(1):1–27, 2006.
- [12] A. J. Chorin, O. H. Hald, and R. Kupferman. Optimal prediction with memory. *Physica D: Nonlinear Phenomena*, 166(3-4):239–257, 2002.
- [13] S. L. Clainche and J. M. Vega. Higher order dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 16(2):882–925, 2017.
- [14] J. Degroote, J. Vierendeels, and K. Willcox. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *International Journal for Numerical Methods in Fluids*, 63:207–230, 2009.
- [15] Z. Drmač, S. Gugercin, and C. Beattie. Vector fitting for matrix-valued rational approximation. *SIAM Journal on Scientific Computing*, 37(5):A2346–A2379, 2015.
- [16] D. Givon, R. Kupferman, and A. Stuart. Extracting macroscopic dynamics: model problems and algorithms. *Nonlinearity*, 17(6):R55–R127, 2004.
- [17] I. V. Gosea and A. C. Antoulas. Data-driven model order reduction of quadratic-bilinear systems. *Numerical Linear Algebra with Applications*, 25(6):e2200, 2018.

- [18] S. Gugercin and A. C. Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766, 2004.
- [19] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on Power Delivery*, 14(3):1052–1061, 1999.
- [20] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer International Publishing, 2016.
- [21] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [22] A. C. Ionita and A. C. Antoulas. Data-driven parametrized model reduction in the Loewner framework. *SIAM Journal on Scientific Computing*, 36(3):A984–A1007, 2014.
- [23] J.-N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics*, 8(5):620–627, 1985.
- [24] B. Kramer and S. Gugercin. The eigensystem realization algorithm from tangentially interpolated data. *Mathematical and Computer Modelling of Dynamical Systems*, 2016. to appear.
- [25] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic mode decomposition: Data-driven modeling of complex systems*. SIAM, 2016.
- [26] S. Lefteriu and A. Antoulas. A new approach to modeling multiport systems from frequency-domain data. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 29(1):14–27, Jan 2010.
- [27] K. K. Lin and F. Lu. Data-driven model reduction, Wiener projections, and the Koopman-Mori-Zwanzig formalism. *Journal of Computational Physics*, 424:109864, 2021.
- [28] R. Maulik, A. Mohan, B. Lusch, S. Madireddy, P. Balaprakash, and D. Livescu. Time-series learning of latent-space dynamics for reduced-order model closure. *Physica D: Nonlinear Phenomena*, 405:132368, 2020.
- [29] A. Mayo and A. Antoulas. A framework for the solution of the generalized realization problem. *Linear Algebra and its Applications*, 425(2–3):634 – 662, 2007.
- [30] S. A. McQuarrie, C. Huang, and K. Willcox. Data-driven reduced-order models via regularized operator inference for a single-injector combustion process. *arXiv e-prints*, page arXiv:2008.02862, 2020.
- [31] C. Mou, H. Liu, D. R. Wells, and T. Iliescu. Data-driven correction reduced order models for the quasi-geostrophic equations: a numerical investigation. *International Journal of Computational Fluid Dynamics*, 34(2):147–159, 2020.
- [32] S. Pan and K. Duraisamy. Data-driven discovery of closure models. *SIAM Journal on Applied Dynamical Systems*, 17(4):2381–2413, 2018.
- [33] B. Peherstorfer. Sampling low-dimensional Markovian dynamics for pre-asymptotically recovering reduced models from data with operator inference. *SIAM Journal on Scientific Computing*, 42(5):A3489–A3515, 2020.
- [34] B. Peherstorfer and K. Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.
- [35] B. Peherstorfer and K. Willcox. Data-driven operator inference for non-intrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.



- [36] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox. Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406:132401, 2020.
- [37] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(1):3, 2011.
- [38] C. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.
- [39] G. Rozza, D. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1–47, 2008.
- [40] P. Sagaut. *Large eddy simulation for incompressible flows : An introduction*. Springer-Verlag, Berlin New York, 2006.
- [41] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher. Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences*, 110(17):6634–6639, 2013.
- [42] H. Schaeffer, G. Tran, and R. Ward. Extracting sparse high-dimensional dynamics from limited data. *SIAM Journal on Applied Mathematics*, 78(6):3279–3295, 2018.
- [43] P. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.
- [44] P. Schmid and J. Sesterhenn. Dynamic mode decomposition of numerical and experimental data. In *Bull. Amer. Phys. Soc., 61st APS meeting*, page 208. American Physical Society, 2008.
- [45] P. Schulze and B. Unger. Data-driven interpolation of dynamical systems with delay. *Systems & Control Letters*, 97:125 – 131, 2016.
- [46] P. Schulze, B. Unger, C. Beattie, and S. Gugercin. Data-driven structured realization. *Linear Algebra and its Applications*, 537:250–286, 2018.
- [47] R. Shumway and D. S. Stoffer. *Time series analysis and its applications : With R examples*. Springer, New York, 2011.
- [48] P. Stinis. Renormalized Mori-Zwanzig-reduced models for systems without scale separation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2176):20140446, 2015.
- [49] E. Suárez, J. L. Adelman, and D. M. Zuckerman. Accurate estimation of protein folding and unfolding times: Beyond markov state models. *Journal of Chemical Theory and Computation*, 12(8):3473–3481, 2016.
- [50] R. Swischuk, B. Kramer, C. Huang, and K. Willcox. Learning physics-based reduced-order models for a single-injector combustion process. *AIAA Journal*, 58(6):2658–2672, 2020.
- [51] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.
- [52] W. I. T. Uy and B. Peherstorfer. Probabilistic error estimation for non-intrusive reduced models learned from data of systems governed by linear parabolic partial differential equations. *arXiv e-prints*, page arXiv:2005.05890, 2020.
- [53] Z. Y. Wan, P. Vlachas, P. Koumoutsakos, and T. Sapsis. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PLOS ONE*, 13(5):e0197704, 2018.

- [54] J. Wang and A. L. Ferguson. Nonlinear reconstruction of single-molecule free-energy surfaces from univariate time series. *Physical Review E*, 93(3), 2016.
- [55] J. Wang and A. L. Ferguson. Recovery of protein folding funnels from single-molecule time series by delay embeddings and manifold learning. *The Journal of Physical Chemistry B*, 122(50):11931–11952, 2018.
- [56] Q. Wang, N. Ripamonti, and J. S. Hesthaven. Recurrent neural network closure of parametric POD-galerkin reduced-order models based on the Mori-Zwanzig formalism. *Journal of Computational Physics*, 410:109402, 2020.
- [57] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Computer Methods in Applied Mechanics and Engineering*, 237-240:10–26, 2012.
- [58] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, June 2015.
- [59] X. Xie, M. Mohebbujjaman, L. G. Rebholz, and T. Iliescu. Data-driven filtered reduced order modeling of fluid flows. *SIAM Journal on Scientific Computing*, 40(3):B834–B857, 2018.
- [60] W. Zucchini. *Hidden Markov models for time series : An introduction using R*. CRC Press, Boca Raton, FL, 2016.