

# PROBABILISTIC ERROR ESTIMATION FOR NON-INTRUSIVE REDUCED MODELS LEARNED FROM DATA OF SYSTEMS GOVERNED BY LINEAR PARABOLIC PARTIAL DIFFERENTIAL EQUATIONS \*

WAYNE ISAAC T. UY<sup>1</sup> AND BENJAMIN PEHERSTORFER<sup>2</sup>

**Abstract.** This work derives a residual-based *a posteriori* error estimator for reduced models learned with non-intrusive model reduction from data of high-dimensional systems governed by linear parabolic partial differential equations with control inputs. It is shown that quantities that are necessary for the error estimator can be either obtained exactly as the solutions of least-squares problems in a non-intrusive way from data such as initial conditions, control inputs, and high-dimensional solution trajectories or bounded in a probabilistic sense. The computational procedure follows an offline/online decomposition. In the offline (training) phase, the high-dimensional system is judiciously solved in a black-box fashion to generate data and to set up the error estimator. In the online phase, the estimator is used to bound the error of the reduced-model predictions for new initial conditions and new control inputs without recourse to the high-dimensional system. Numerical results demonstrate the workflow of the proposed approach from data to reduced models to certified predictions.

**AMS Subject Classification.** 65M60, 65P99, 65Y99, 65F99, 93C05, 62G15, 65C05, 65N15.

The dates will be set by the publisher.

## 1. INTRODUCTION

Model reduction constructs reduced models that rapidly approximate solutions of differential equations by solving in problem-dependent, low-dimensional subspaces of classical, high-dimensional (e.g., finite-element) solution spaces [2, 12, 17, 32, 34]. Traditional model reduction methods typically are intrusive in the sense that full knowledge about the underlying governing equations and their discretizations are required to derive reduced models. In contrast, this work considers non-intrusive model reduction that aims to learn reduced models from data with only little knowledge about the governing equations and their discretizations. However, constructing reduced models is only one aspect of model reduction. Another aspect is deriving *a posteriori* error estimators that bound the error of reduced-model predictions with respect to the high-dimensional solutions that are obtained numerically with, e.g., finite-element methods [11, 15, 30, 37, 41–43]. This work builds on *a posteriori* error estimators [11, 15] from intrusive model reduction to establish error estimation for reduced models that are learned with non-intrusive methods. The key contribution is to show that all quantities required for deriving the

---

*Keywords and phrases:* model reduction, error estimation, non-intrusive model reduction, small sample statistical estimates

\* 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.

<sup>1</sup> wayne.uy@cims.nyu.edu, Courant Institute of Mathematical Sciences, New York, NY 10012

<sup>2</sup> pehersto@cims.nyu.edu, Courant Institute of Mathematical Sciences, New York, NY 10012

error estimator can be either obtained in a non-intrusive way via least-squares regression from initial conditions, control inputs, and solution trajectories or bounded in a probabilistic sense, if the system of interest is known to be governed by a linear parabolic partial differential equation (PDE) with control inputs. The key requirement to make the estimator practical is that the high-dimensional system is queryable in the sense that during a training (offline) phase one has access to a black box that one can feed with initial conditions and inputs and that returns the corresponding numerical approximations of the high-dimensional solution trajectories. If one considers learning reduced models from data as a machine learning task, then the proposed error estimator can be considered as pre-asymptotic computable generalization bound [25] of the learned models because the proposed estimator provides an upper bound on the error of the reduced model for initial conditions and inputs that have not been seen during learning (training) the reduced model and the error-estimator quantities. The bound is pre-asymptotic with respect to the number of data points and the dimension of the reduced model.

We review literature on non-intrusive and data-driven model reduction. First, the systems and control community has developed methods for identifying dynamical systems from frequency-response or impulse-response data, e.g., the Loewner approach by Antoulas and collaborators [1, 10, 19], vector fitting [6, 13], and eigensystem realization [21, 22]. In contrast, our approach will learn from time-domain data; not necessarily impulse-response data. Second, dynamic mode decomposition [23, 33, 35, 36, 40] has been shown to successfully derive linear dynamical systems that best fit data in the  $L_2$  norm. However, the authors are unaware of error estimators for models derived with dynamic mode decomposition. Third, there is operator inference [29] that coincides with dynamic mode decomposition in case of linear systems but is also applicable to data from systems with nonlinear terms; see also the work on lift & learn for general nonlinear systems [31] and the work on dynamic reduced models [28]. The error estimators proposed in the following will build on operator inference for non-intrusive model reduction because, together with a particular data-sampling scheme [27], operator inference exactly recovers the reduced models that are obtained via traditional intrusive model reduction. Thus, the learned models are the traditional reduced models with well-studied properties known from intrusive model reduction.

We now review literature on error estimators developed for intrusive model reduction. First, the reduced-basis community has developed error estimators for elliptic PDEs [30] and parabolic PDEs [11] with affine parameter dependence, time-dependent viscous Burgers' equation [20, 26], and linear evolution equations [14, 15], among others. For systems that are nonlinear and/or have non-affine parameter dependence, error bounds have been established for reduced models with empirical interpolation in, e.g., [4, 7, 16, 44]. These error estimators typically depend on the dual norm of the reduced-model residual and on other quantities of the underlying PDE discretizations such as coercivity and inf-sup stability constants [18] that require knowledge about the weak form of the governing equations that are unavailable in the setting of non-intrusive model reduction where one has access to data alone. The work [38] proposes a probabilistic error bound involving randomized residuals which overcomes the need to compute constants in the error estimators; however, the reduced models are constructed with traditional intrusive model reduction and, in particular, residuals are computed in an intrusive way which conflicts with non-intrusive model reduction. In the systems and control community, the discrepancy between the high-dimensional solutions of systems of ordinary differential equations and reduced-model solutions is bounded in terms of the transfer functions, see, e.g., [9, 45].

This manuscript is organized as follows: Section 2 outlines preliminaries on spatial and temporal discretization of linear parabolic PDEs and intrusive model reduction. Section 3 describes the proposed error estimator for reduced models learned with operator inference from data. First, least-squares problems are derived to infer residual-norm operators from data. Second, constants required for error estimation are bounded in a probabilistic sense. These two novel components are combined together with an intrusive error estimator [15] into a computational procedure that realizes the full workflow from data to reduced models to certification of reduced-model predictions, under certain conditions that are made precise. Numerical results are presented in Section 4 and conclusions are drawn in Section 5.

## 2. PRELIMINARIES

Section 2.1 reviews linear parabolic PDEs with spatial and time discretization discussed in Sections 2.2 and 2.3, respectively. The continuous-time problem is transformed into a discrete linear time-invariant system. Intrusive model reduction is then recalled in Section 2.4. Section 2.5 outlines the problem formulation.

### 2.1. Linear parabolic PDEs with time-independent coefficients

Let  $\Omega \subset \mathbb{R}^m$  be a bounded domain and let  $\mathcal{T} = (0, T)$  be a time interval with  $T > 0$  fixed. Consider the linear parabolic PDE on  $(\mathbf{x}, t) \in \Omega \times \mathcal{T}$  given by

$$\begin{aligned} \frac{\partial}{\partial t} w(\mathbf{x}, t) &= \nabla \cdot (\mathbf{b}^T(\mathbf{x}) \nabla w(\mathbf{x}, t)) - \mathbf{c}(\mathbf{x}) \cdot \nabla w(\mathbf{x}, t) - d(\mathbf{x})w(\mathbf{x}, t) + R(\mathbf{x}), \\ w(\mathbf{x}, t) &= 0 \quad \text{for } \mathbf{x} \in \Gamma_D, \\ \mathbf{b}^T(\mathbf{x}) \nabla w(\mathbf{x}, t) \cdot \mathbf{n} &= u_j(t) \quad \text{for } \mathbf{x} \in \Gamma_N^j, \quad j = 1, \dots, n_\Gamma, \\ w(\mathbf{x}, 0) &= h(\mathbf{x}), \end{aligned} \quad (2.1)$$

where  $w : \Omega \times \mathcal{T} \rightarrow \mathbb{R}$  is the solution,  $\mathbf{b} : \Omega \rightarrow \mathbb{R}^{m \times m}$ ,  $\mathbf{c} : \Omega \rightarrow \mathbb{R}^m$ ,  $d : \Omega \rightarrow \mathbb{R}$  are time-independent coefficients,  $R : \Omega \rightarrow \mathbb{R}$  is the source term and the boundary  $\partial\Omega$  is decomposed into the  $n_\Gamma$  disjoint segments  $\cup_{j=1}^{n_\Gamma} \Gamma_N^j = \Gamma_N$  with Neumann conditions and the remaining portion  $\Gamma_D$  with Dirichlet condition. The control inputs are  $\{u_j(t)\}_{j=1}^{n_\Gamma}$  for  $t \in \mathcal{T}$ . Define  $[\cdot]_i$  as the  $i$ -th component of a vector and  $[\cdot]_{ij}$  as the  $(i, j)$ -th component of a matrix. Let further  $[\mathbf{b}]_{ij}, [\mathbf{c}]_i, d \in L^\infty(\Omega)$  for  $i, j \in \{1, \dots, m\}$ ,  $R, h \in L^2(\Omega)$ , and  $u_j \in L^2(\mathcal{T})$  for  $j = 1, \dots, n_\Gamma$  where  $L^2, L^\infty$  correspond to the space of square-integrable and essentially bounded measurable functions, respectively. For (2.1) to be parabolic, it is required that for any  $\boldsymbol{\xi} \in \mathbb{R}^m$  and  $\mathbf{x} \in \Omega$ , there exists a constant  $\theta > 0$  such that  $\boldsymbol{\xi}^T \mathbf{b}(\mathbf{x}) \boldsymbol{\xi} \geq \theta \|\boldsymbol{\xi}\|_2^2$  [8].

### 2.2. Spatial discretization

For the Sobolev space  $H^1(\Omega)$ , define  $\mathcal{V} = \{v \in H^1(\Omega) \mid v|_{\Gamma_D} = 0\}$  which is equipped with the norm  $\|\cdot\|_{\mathcal{V}}$ . Denote by  $C^0(\mathcal{T}; L^2(\Omega))$  the space of continuous functions  $w(t) : \mathcal{T} \rightarrow L^2(\Omega)$  with  $\max_{t \in \mathcal{T}} \|w(t)\|_{L^2(\Omega)} < \infty$  and  $L^2(\mathcal{T}; \mathcal{V})$  the space of all strongly measurable functions  $w(t) : \mathcal{T} \rightarrow \mathcal{V}$  for which  $(\int_{\mathcal{T}} \|w(t)\|_{\mathcal{V}}^2 dt)^{1/2} < \infty$ . We seek  $w \in C^0(\mathcal{T}; L^2(\Omega)) \cap L^2(\mathcal{T}; \mathcal{V})$  such that

$$\int_{\Omega} v \frac{\partial}{\partial t} w \, d\mathbf{x} = -a(w, v) + f(v) \quad \forall v \in \mathcal{V} \quad (2.2)$$

where

$$a(w, v) = \int_{\Omega} \nabla v \cdot (\mathbf{b}^T \nabla w) \, d\mathbf{x} + \int_{\Omega} v [\nabla w \cdot \mathbf{c} + wd] \, d\mathbf{x}$$

and

$$f(v) = \int_{\Omega} v R \, d\mathbf{x} + \sum_{j=1}^{n_\Gamma} u_j(t) \int_{\Gamma_N^j} v \, d\Gamma_N,$$

see [8, 17, 39] for details. In the following, we assume that the bilinear form  $a$  in (2.2) is coercive and continuous, i.e.,  $\exists \alpha > 0$  and  $\gamma < \infty$  for which  $a(v, v) \geq \alpha \|v\|_{\mathcal{V}}^2$  and  $a(w, v) \leq \gamma \|w\|_{\mathcal{V}} \|v\|_{\mathcal{V}}$  for  $v, w \in \mathcal{V}$  and  $f$  in (2.2) is continuous. To discretize (2.2), consider a finite-dimensional approximation space  $\mathcal{V}_N \subset \mathcal{V}$  with basis  $\{\varphi_i\}_{i=1}^N$  such that for  $w \in \mathcal{V}_N$ ,  $w(\mathbf{x}, t) = \sum_{i=1}^N w_i(t) \varphi_i(\mathbf{x})$ . Setting  $v = \varphi_i, i = 1, \dots, N$  in (2.2) results in

$$\mathbf{M} \frac{d\mathbf{w}(t)}{dt} = \mathbf{K} \mathbf{w}(t) + \mathbf{F} \mathbf{u}(t) \quad (2.3)$$

where  $\mathbf{w}(t) = [w_1(t), \dots, w_N(t)]^T \in \mathbb{R}^N$ ,  $\mathbf{M} \in \mathbb{R}^{N \times N}$  such that  $[\mathbf{M}]_{ij} = \int_{\Omega} \varphi_j \varphi_i d\mathbf{x}$ ,  $\mathbf{K} \in \mathbb{R}^{N \times N}$  such that  $[\mathbf{K}]_{ij} = -a(\varphi_j, \varphi_i)$ ,  $\mathbf{u}(t) = [1, u_1(t), \dots, u_{n_{\Gamma}}(t)]^T \in \mathbb{R}^p$  with  $p = n_{\Gamma} + 1$ , while

$$\mathbf{F} = \begin{bmatrix} \int_{\Omega} \varphi_1 R d\mathbf{x} & \int_{\Gamma_N^1} \varphi_1 d\Gamma_N & \dots & \int_{\Gamma_N^{n_{\Gamma}}} \varphi_1 d\Gamma_N \\ \vdots & \vdots & \ddots & \vdots \\ \int_{\Omega} \varphi_N R d\mathbf{x} & \int_{\Gamma_N^1} \varphi_N d\Gamma_N & \dots & \int_{\Gamma_N^{n_{\Gamma}}} \varphi_N d\Gamma_N \end{bmatrix} \in \mathbb{R}^{N \times p}.$$

If the source term  $R = 0$ ,  $p = n_{\Gamma}$  and the resulting  $\mathbf{u}(t)$ ,  $\mathbf{F}$  are obtained by truncating the first component of  $\mathbf{u}(t)$  and the first column of  $\mathbf{F}$  defined above.

### 2.3. Time discretization

To temporally discretize the time-continuous system (2.3), let  $\{t_k\}_{k=0}^K \subset \mathcal{T}$  be equally spaced points with  $t_{k+1} - t_k = \delta t$  and denote by  $\mathbf{w}_k, \mathbf{u}_k$  the discrete time approximations to  $\mathbf{w}(t_k), \mathbf{u}(t_k)$ . A one-step scheme can be expressed as

$$\frac{\mathbf{w}_{k+1} - \mathbf{w}_k}{\delta t} = \beta \mathbf{M}^{-1}(\mathbf{K}\mathbf{w}_{k+1} + \mathbf{F}\mathbf{u}_{k+1}) + (1 - \beta)\mathbf{M}^{-1}(\mathbf{K}\mathbf{w}_k + \mathbf{F}\mathbf{u}_k), \quad \beta \in [0, 1] \quad (2.4)$$

in which we recover the forward Euler, backward Euler, and Crank-Nicolson method with  $\beta = 0$ ,  $\beta = 1$ , and  $\beta = \frac{1}{2}$ , respectively. We rewrite (2.4) as

$$\mathbf{w}_{k+1} = \mathbf{A}\mathbf{w}_k + \mathbf{B}\mathbf{g}_{k+1} \quad (2.5)$$

with

$$\begin{aligned} \mathbf{A} &= (\mathbf{I}_N - \beta \delta t \mathbf{M}^{-1} \mathbf{K})^{-1} (\mathbf{I}_N + (1 - \beta) \delta t \mathbf{M}^{-1} \mathbf{K}), \\ \mathbf{B} &= (\mathbf{I}_N - \beta \delta t \mathbf{M}^{-1} \mathbf{K})^{-1} \delta t \mathbf{M}^{-1} \mathbf{F}, \\ \mathbf{g}_{k+1} &= \beta \mathbf{u}_{k+1} + (1 - \beta) \mathbf{u}_k, \end{aligned}$$

and the  $N \times N$  identity matrix  $\mathbf{I}_N$ . Note that  $\mathbf{g}_{k+1} = \mathbf{u}_k$  for  $\beta = 0$  while  $\mathbf{g}_{k+1} = \mathbf{u}_{k+1}$  for  $\beta = 1$ . We refer to  $\mathbf{W} = [\mathbf{w}_0, \dots, \mathbf{w}_K]$  as a trajectory. We further define  $\mathcal{G}$  as the set of input trajectories  $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_K] \in \mathbb{R}^{p \times K}$  of arbitrary but finite length  $K$  so that  $\sum_{k=1}^K [\mathbf{g}_k]_i^2 < \infty$  for  $i = 1, \dots, p$ , i.e. each component of the discrete-time input has finite norm on the time interval  $\mathcal{T}$ . Since  $u_j \in L^2(\mathcal{T})$ , we only consider input trajectories  $\mathbf{G} \in \mathcal{G}$ . In the following, we restrict ourselves to equidistant time discretizations. Non-equidistant discretizations in time of (2.3) lead to time-varying systems instead of the time-invariant system (2.5) that we consider here; see Section 5 for a discussion about the extension of the proposed approach to time-varying systems.

### 2.4. Traditional (intrusive) model reduction

Model reduction seeks an approximate solution to (2.5) which lies in a low-dimensional subspace  $\mathcal{V}_n$  spanned by the columns of  $\mathbf{V}_n = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{N \times n}$  with  $n \ll N$ . Various approaches exist for constructing the low-dimensional subspace, see, e.g., [2, 12, 17, 32, 34]. In the following, we use the proper orthogonal decomposition (POD) to construct  $\mathbf{V}_n$ . Let  $[\mathbf{w}_0, \dots, \mathbf{w}_K]$  be the snapshot matrix whose columns are the states  $\mathbf{w}_k$ . The basis  $\mathbf{V}_n$  for  $\mathcal{V}_n$  is derived from the left singular vectors of the snapshot matrix corresponding to the  $n$  largest singular values. Via Galerkin projection, the low-dimensional (reduced) system can then be derived as

$$\tilde{\mathbf{w}}_{k+1} = \tilde{\mathbf{A}}\tilde{\mathbf{w}}_k + \tilde{\mathbf{B}}\mathbf{g}_{k+1} \quad (2.6)$$

where

$$\tilde{\mathbf{A}} = \mathbf{V}_n^T \mathbf{A} \mathbf{V}_n \in \mathbb{R}^{n \times n}, \quad \tilde{\mathbf{B}} = \mathbf{V}_n^T \mathbf{B} \in \mathbb{R}^{n \times p}. \quad (2.7)$$

The low-dimensional solution  $\tilde{\mathbf{w}}_k$  approximates the solution  $\mathbf{w}_k$  to (2.5) through  $\mathbf{V}_n \tilde{\mathbf{w}}_k$ . We refer to  $\tilde{\mathbf{W}} = [\tilde{\mathbf{w}}_0, \dots, \tilde{\mathbf{w}}_{K-1}]$  as a reduced trajectory.

## 2.5. Non-intrusive model reduction and problem formulation

Deriving reduced model (2.6) by forming the matrix-matrix products (2.7) of the basis matrix  $\mathbf{V}_n$  and the operators  $\mathbf{A}$  and  $\mathbf{B}$  of the high-dimensional system is intrusive in the sense that  $\mathbf{A}$  and  $\mathbf{B}$  are required either in assembled form or implicitly through a routine that provides the action of  $\mathbf{A}$  and  $\mathbf{B}$  to a vector. In the following, we are interested in the situation where  $\mathbf{A}$  and  $\mathbf{B}$  are unavailable. Rather, we can simulate the high-dimensional system (2.5) at initial conditions and control inputs to generate state trajectories. Building on non-intrusive model reduction, we learn the reduced operators  $\tilde{\mathbf{A}}$  and  $\tilde{\mathbf{B}}$  from state trajectories without having  $\mathbf{A}$  and  $\mathbf{B}$  available. A major component of intrusive model reduction, besides constructing reduced models, is deriving error estimators that rigorously upper bound the approximation error of the reduced models with respect to the high-dimensional solutions [11, 15, 30, 37, 41–43]. However, such error estimators typically depend on quantities such as norms of  $\mathbf{A}$  and residuals that are unavailable in non-intrusive model reduction. Thus, error estimators developed for intrusive model reduction typically cannot be directly applied when reduced models are learned with non-intrusive model reduction methods.

## 3. CERTIFYING REDUCED MODELS LEARNED FROM DATA

Our goal is two-fold: (i) learning the reduced operators (2.7) from state trajectories of the high-dimensional system and (ii) learning quantities to establish *a posteriori* error estimators to rigorously bound the error  $\|\mathbf{w}_k - \mathbf{V}_n \tilde{\mathbf{w}}_k\|_2$  in the 2-norm  $\|\cdot\|_2$  of the reduced solution  $\tilde{\mathbf{w}}_k$  with respect to the high-dimensional solution  $\mathbf{w}_k$  at time step  $k$  for different initial conditions and different inputs than what was used during (i). The reduced operators and the quantities for the error estimators are learned under the setting that the high-dimensional operators in (2.5) are unavailable in assembled and implicit form. We build on a non-intrusive approach for model reduction based on operator inference [29] and re-projection [27] and on an error estimator for linear evolution equations [15]. We show that the required quantities for the error estimator can be recovered from residual trajectories corresponding to training control inputs in a non-intrusive way similar to learning the reduced operators with operator inference and re-projection. These quantities then allow bounding the state error for other inputs and initial conditions.

Section 3.1 reviews operator inference with re-projection introduced in [27] and provides novel results on conditions which permit recovery of the reduced system operators. Section 3.2 discusses an error estimator from intrusive model reduction as presented in [15]. To carry over the error estimator [15] to the non-intrusive model reduction case, an optimization problem is formulated in Section 3.3 whose unique solution leads to the required quantities for error estimation under certain conditions. Sections 3.4 and 3.5 address prediction of the state *a posteriori* error for other control inputs. The former utilizes a deterministic bound for the state error. In contrast, the latter offers a probabilistic error estimator whose reliability, the probability of failure of the error estimator, can be controlled by the number of samples. A summary of the proposed approach comprised of an offline (training) and online (prediction) phase is then given in Section 3.6.

### 3.1. Recovering reduced models from data with operator inference and re-projection

Let  $\mathbf{V}_n$  be the basis matrix with  $n$  columns. Building on [29], the work [27] introduces a re-projection scheme to generate the reduced trajectory  $\tilde{\mathbf{W}} = [\tilde{\mathbf{w}}_0, \dots, \tilde{\mathbf{w}}_{K-1}]$  that would be obtained with the reduced model (2.6) as if it were available by querying the high-dimensional system (2.5) alone with input trajectory  $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_K]$ . We define a queryable system as follows.

**Definition 1.** A system (2.5) is queryable if the trajectory  $[\mathbf{w}_0, \dots, \mathbf{w}_K]$  with  $K \geq 1$  can be computed for any initial condition  $\mathbf{w}_0 \in \mathcal{V}_n$  and any input trajectory  $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_K] \in \mathcal{G}$ .

For example, system (2.5) can be black-box and queryable in the sense that the operators  $\mathbf{A}$  and  $\mathbf{B}$  are unavailable but  $\mathbf{w}_0$  and  $\mathbf{G}$  can be provided to a black box to produce  $\mathbf{W}$ . In contrast, if there is a high-dimensional system for which a trajectory  $\mathbf{W}$  for an input trajectory  $\mathbf{G}$  is given, without being able to choose  $\mathbf{G}$  and initial condition, then such a system is not queryable.

For a queryable system, the re-projection scheme alternates between time-stepping the high-dimensional system (2.5) and projecting the state onto the space  $\mathcal{V}_n$  spanned by the columns of  $\mathbf{V}_n$ . Let  $\mathbf{w}_0 \in \mathcal{V}_n$  be the initial condition and define  $\bar{\mathbf{w}}_0 = \mathbf{V}_n^T \mathbf{w}_0$ . The re-projection scheme takes a single time step with the high-dimensional system (2.5) with initial condition  $\mathbf{V}_n \bar{\mathbf{w}}_0$  and control input  $\mathbf{g}_1$  to obtain  $\mathbf{w}_1$ . The state  $\mathbf{w}_1$  is projected to obtain  $\bar{\mathbf{w}}_1 = \mathbf{V}_n^T \mathbf{w}_1$ , and the process is repeated by taking a single time step with the high-dimensional system (2.5) with initial condition  $\mathbf{V}_n \bar{\mathbf{w}}_1$  and control input  $\mathbf{g}_2$ . The high-dimensional system is re-simulated to obtain the re-projected trajectory  $\bar{\mathbf{W}} = [\bar{\mathbf{w}}_0, \dots, \bar{\mathbf{w}}_{K-1}]$ . It is shown in [27] that  $\bar{\mathbf{W}}$  is the reduced trajectory  $\tilde{\mathbf{W}} = [\tilde{\mathbf{w}}_0, \dots, \tilde{\mathbf{w}}_{K-1}]$  in our case of a linear system (2.5). Furthermore, the least-squares problem

$$\min_{\hat{\mathbf{A}}, \hat{\mathbf{B}}} \sum_{k=0}^{K-1} \left\| \hat{\mathbf{A}} \bar{\mathbf{w}}_k + \hat{\mathbf{B}} \mathbf{g}_{k+1} - \bar{\mathbf{w}}_{k+1} \right\|_2^2 \quad (3.1)$$

has as the unique solution the reduced operators  $\tilde{\mathbf{A}}$  and  $\tilde{\mathbf{B}}$  if the data matrix

$$\Psi = \begin{bmatrix} \bar{\mathbf{W}}^T & \mathbf{G}^T \end{bmatrix} \in \mathbb{R}^{K \times (n+p)} \quad (3.2)$$

has full rank and  $K \geq n+p$ ; see Corollary 3.2 in [27] for more details.

The following proposition generalizes the least-squares problem (3.1) to trajectories from multiple initial conditions and shows that in this case there always exist initial conditions and input trajectories such that the unique solution of the corresponding least-squares problem is given by the reduced operators  $\tilde{\mathbf{A}}$  and  $\tilde{\mathbf{B}}$ . This may be necessary in scenarios for which  $\Psi$  in (3.2), which only depends on a single initial condition, is not full rank despite  $K \geq n+p$ . This may occur if the system (2.5) evolves slowly such that the rows of  $\Psi$  are numerically linearly dependent.

**Proposition 2.** *There exist  $n+p$  input trajectories  $\mathbf{G}^{(1)}, \dots, \mathbf{G}^{(n+p)} \in \mathcal{G}$ , each of finite length  $K_\ell \in \mathbb{N}$  for  $\ell = 1, \dots, n+p$ , and  $n+p$  initial conditions  $\mathbf{w}_0^{(1)}, \dots, \mathbf{w}_0^{(n+p)} \in \mathcal{V}_n$  such that the generalized data matrix*

$$\Phi = \begin{bmatrix} \bar{\mathbf{W}}^{(1)} & \dots & \bar{\mathbf{W}}^{(n+p)} \\ \mathbf{G}^{(1)} & \dots & \mathbf{G}^{(n+p)} \end{bmatrix}^T \in \mathbb{R}^{(\sum_{\ell=1}^{n+p} K_\ell) \times (n+p)}$$

with re-projected trajectories  $\bar{\mathbf{W}}^{(\ell)} = [\bar{\mathbf{w}}_0^{(\ell)}, \dots, \bar{\mathbf{w}}_{K_\ell-1}^{(\ell)}] \in \mathbb{R}^{n \times K_\ell}$  has full rank, thereby guaranteeing the recovery of the reduced operators  $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$  via least squares regression.

*Proof.* The generalized data matrix  $\Phi$  is induced by the least squares problem

$$\min_{\hat{\mathbf{A}}, \hat{\mathbf{B}}} \sum_{\ell=1}^{n+p} \sum_{k=0}^{K_\ell-1} \left\| \hat{\mathbf{A}} \bar{\mathbf{w}}_k^{(\ell)} + \hat{\mathbf{B}} \mathbf{g}_{k+1}^{(\ell)} - \bar{\mathbf{w}}_{k+1}^{(\ell)} \right\|_2^2 \quad (3.3)$$

which is an extension of the least squares problem (3.1) for the case when there are  $\ell = 1, \dots, n+p$  initial conditions  $\mathbf{w}_0^{(1)}, \dots, \mathbf{w}_0^{(n+p)}$  and input trajectories  $\mathbf{G}^{(\ell)} = [\mathbf{g}_1^{(\ell)}, \dots, \mathbf{g}_{K_\ell}^{(\ell)}] \in \mathbb{R}^{p \times K_\ell}$ . If  $\Phi$  is full rank,  $\hat{\mathbf{A}}, \hat{\mathbf{B}}$  in (3.3) recover the reduced operators  $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$  as discussed in [27].

We now derive specific initial conditions and control inputs that lead to a full-rank  $\Phi$ . First, we select  $n$  linearly independent initial conditions  $\mathbf{w}_0^{(1)}, \dots, \mathbf{w}_0^{(n)} \in \mathcal{V}_n$ , which exist because  $\mathcal{V}_n$  has  $n$  dimensions. Correspondingly,  $\bar{\mathbf{w}}_0^{(1)}, \dots, \bar{\mathbf{w}}_0^{(n)} \in \mathbb{R}^n$  are linearly independent as well. To see this, note that  $\bar{\mathbf{w}}_0^{(i)} = \mathbf{V}_n^T \mathbf{w}_0^{(i)}$

holds for  $i = 1, \dots, n$  and thus  $\mathbf{V}_n[\bar{\mathbf{w}}_0^{(1)}, \dots, \bar{\mathbf{w}}_0^{(n)}] = [\mathbf{w}_0^{(1)}, \dots, \mathbf{w}_0^{(n)}]$  because  $\mathbf{w}_0^{(1)}, \dots, \mathbf{w}_0^{(n)} \in \mathcal{V}_n$ . Because  $\mathbf{V}_n$  has orthonormal columns, the rank of  $[\bar{\mathbf{w}}_0^{(1)}, \dots, \bar{\mathbf{w}}_0^{(n)}] = \mathbf{V}_n^T[\mathbf{w}_0^{(1)}, \dots, \mathbf{w}_0^{(n)}]$  is equal to the rank of  $\mathbf{V}_n(\mathbf{V}_n^T[\mathbf{w}_0^{(1)}, \dots, \mathbf{w}_0^{(n)}]) = [\mathbf{w}_0^{(1)}, \dots, \mathbf{w}_0^{(n)}]$ , which is  $n$ . Set  $\mathbf{g}_1^{(i)} = \mathbf{0}_{p \times 1}$  for  $i = 1, \dots, n$  where  $\mathbf{0}_{m \times n}$  represents an  $m \times n$  matrix of zeros. Second, set  $\mathbf{w}_0^{(n+1)} = \dots = \mathbf{w}_0^{(n+p)} = \mathbf{0}_{N \times 1} \in \mathcal{V}_n$  and select  $p$  linearly independent control inputs  $\mathbf{g}_1^{(n+1)}, \dots, \mathbf{g}_1^{(n+p)} \in \mathcal{G}$ , which exist because  $\mathbb{R}^p \subset \mathcal{G}$  per definition; see Section 2.3. Taking these  $n+p$  initial conditions and input signals and time-stepping with re-projection the high-dimensional system for a finite number of times steps leads to a generalized data matrix  $\Phi$  that contains at least the following rows

$$\begin{bmatrix} \bar{\mathbf{w}}_0^{(1)} \\ \mathbf{g}_1^{(1)} \end{bmatrix}^T, \dots, \begin{bmatrix} \bar{\mathbf{w}}_0^{(n+p)} \\ \mathbf{g}_1^{(n+p)} \end{bmatrix}^T.$$

The matrix  $\Phi$  therefore contains  $n+p$  linearly independent rows and thus has full rank. Note that  $K_\ell \geq 1$  for  $\ell = 1, \dots, n+p$ .  $\square$

**Remark 3.** Proposition 2 considers trajectories from multiple initial conditions to show that initial conditions and input trajectories exist to recover the reduced model via operator inference and re-projection. It is possible for the generalized data matrix  $\Phi$  to be full rank even though fewer than  $n+p$  initial conditions are utilized. However, Proposition 2 demonstrates in a constructive manner that only at most  $n+p$  initial conditions are necessary to recover the reduced model. To ease exposition, we build on the formulation with a single initial condition (3.1) in the following and in all our numerical results. However, the following results immediately generalize to the formulation with multiple initial conditions used in Proposition 2.

### 3.2. Error estimation for linear reduced models in intrusive model reduction

We now elaborate on an *a posteriori* estimator for the state error in intrusive model reduction by following the presentation by Haasdonk and Ohlberger [15]; note, however, that intrusive error estimation for reduced models of parabolic PDEs has been studied by Grepl and Patera in [11] as well and the following non-intrusive approach may extend to their error estimators too. For  $k \in \mathbb{N}$ , define the state error at time  $k$  as  $\mathbf{w}_k - \mathbf{V}_n \tilde{\mathbf{w}}_k$  and the residual  $\mathbf{r}_k$  as

$$\mathbf{r}_k = \mathbf{A} \mathbf{V}_n \tilde{\mathbf{w}}_k + \mathbf{B} \mathbf{g}_{k+1} - \mathbf{V}_n \tilde{\mathbf{w}}_{k+1}. \quad (3.4)$$

The state error is

$$\mathbf{w}_k - \mathbf{V}_n \tilde{\mathbf{w}}_k = \mathbf{A}^k (\mathbf{w}_0 - \mathbf{V}_n \tilde{\mathbf{w}}_0) + \sum_{l=0}^{k-1} \mathbf{A}^{k-l-1} \mathbf{r}_l. \quad (3.5)$$

Define

$$\Delta_k^{\mathbf{w}}(c_0, \dots, c_k; \mathbf{w}_0, \mathbf{G}) = c_0 \|\mathbf{w}_0 - \mathbf{V}_n \tilde{\mathbf{w}}_0\|_2 + \sum_{l=0}^{k-1} c_{l+1} \|\mathbf{r}_l\|_2 \quad (3.6)$$

which relies on the initial condition  $\mathbf{w}_0$ , input trajectory  $\mathbf{G} \in \mathcal{G}$ , and constants  $c_0, \dots, c_k \in \mathbb{R}$ . The norm of (3.5) is then bounded by

$$\|\mathbf{w}_k - \mathbf{V}_n \tilde{\mathbf{w}}_k\|_2 \leq \Delta_k^{\mathbf{w}}(\|\mathbf{A}^k\|_2, \dots, \|\mathbf{A}^0\|_2; \mathbf{w}_0, \mathbf{G}) = \|\mathbf{A}^k\|_2 \|\mathbf{w}_0 - \mathbf{V}_n \tilde{\mathbf{w}}_0\|_2 + \sum_{l=0}^{k-1} \|\mathbf{A}^{k-l-1}\|_2 \|\mathbf{r}_l\|_2. \quad (3.7)$$

If  $\max_{0 \leq l \leq k} \|\mathbf{A}^l\|_2 \leq C$  for a constant  $C \in \mathbb{R}$ , then the following holds

$$\|\mathbf{w}_k - \mathbf{V}_n \tilde{\mathbf{w}}_k\|_2 \leq \Delta_k^{\mathbf{w}}(\underbrace{C, \dots, C}_{k+1}; \mathbf{w}_0, \mathbf{G}).$$

The error  $\mathbf{w}_0 - \mathbf{V}_0 \tilde{\mathbf{w}}_0$  of the initial condition is the projection error  $\mathbf{w}_0 - \mathbf{V}_n \mathbf{V}_n^T \mathbf{w}_0$  and can be computed if  $\mathbf{V}_n$  and the initial condition  $\mathbf{w}_0$  are known.

### 3.3. Recovering the residual operators from residual trajectories

The residual norm  $\|\mathbf{r}_k\|_2$  at time step  $k$  is a critical component for the error estimator in [15]; directly computing  $\|\mathbf{r}_k\|_2$  using formula (3.4) would require either the high-dimensional system operators  $\mathbf{A}$  and  $\mathbf{B}$  or querying the system (2.5) at each  $\tilde{\mathbf{w}}_k$ . Following [15], the squared residual norm is expanded as

$$\|\mathbf{r}_k\|_2^2 = \tilde{\mathbf{w}}_k^T \mathbf{M}_1 \tilde{\mathbf{w}}_k + \mathbf{g}_{k+1}^T \mathbf{M}_2 \mathbf{g}_{k+1} + 2\mathbf{g}_{k+1}^T \mathbf{M}_3 \tilde{\mathbf{w}}_k + \tilde{\mathbf{w}}_{k+1}^T \mathbf{M}_4 \tilde{\mathbf{w}}_{k+1} - 2\tilde{\mathbf{w}}_{k+1}^T \tilde{\mathbf{A}} \tilde{\mathbf{w}}_k - 2\tilde{\mathbf{w}}_{k+1}^T \tilde{\mathbf{B}} \mathbf{g}_{k+1} \quad (3.8)$$

with the matrices

$$\mathbf{M}_1 = \mathbf{V}_n^T \mathbf{A}^T \mathbf{A} \mathbf{V}_n, \quad \mathbf{M}_2 = \mathbf{B}^T \mathbf{B}, \quad \mathbf{M}_3 = \mathbf{B}^T \mathbf{A} \mathbf{V}_n, \quad (3.9)$$

and  $\mathbf{M}_4 = \mathbf{V}_n^T \mathbf{V}_n$ . Observe that after the reduced model has been obtained with operator inference and re-projection (Section 3.1), the matrices  $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$ , and  $\mathbf{M}_4$  can be readily computed without  $\mathbf{A}$  and  $\mathbf{B}$ . Only matrices  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$  are needed additionally to compute the squared residual norm with (3.8).

Let  $\tilde{\mathbf{W}} = [\tilde{\mathbf{w}}_0, \dots, \tilde{\mathbf{w}}_{K-1}]$  be the re-projected trajectory using an input trajectory  $\mathbf{G}$ . Let further  $\tilde{\mathbf{R}} = [\tilde{\mathbf{r}}_0, \dots, \tilde{\mathbf{r}}_{K-1}]$  be the residual trajectory corresponding to the re-projected trajectory defined as

$$\tilde{\mathbf{r}}_k = \mathbf{A} \mathbf{V}_n \tilde{\mathbf{w}}_k + \mathbf{B} \mathbf{g}_{k+1} - \mathbf{V}_n \tilde{\mathbf{w}}_{k+1},$$

following the residual expression in (3.4). The following proposition shows that  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$  can be derived via a least-squares problem using  $\tilde{\mathbf{R}}, \tilde{\mathbf{W}}, \mathbf{G}$ .

**Proposition 4.** Define the data matrix  $\mathbf{D} \in \mathbb{R}^{K \times \frac{1}{2}(n+p)(n+p+1)}$  as

$$\mathbf{D} = \begin{bmatrix} \text{vech}(2\tilde{\mathbf{w}}_0 \tilde{\mathbf{w}}_0^T - \text{diag}(\tilde{\mathbf{w}}_0 \tilde{\mathbf{w}}_0^T)) & \cdots & \text{vech}(2\tilde{\mathbf{w}}_{K-1} \tilde{\mathbf{w}}_{K-1}^T - \text{diag}(\tilde{\mathbf{w}}_{K-1} \tilde{\mathbf{w}}_{K-1}^T)) \\ \text{vech}(2\mathbf{g}_1 \mathbf{g}_1^T - \text{diag}(\mathbf{g}_1 \mathbf{g}_1^T)) & \cdots & \text{vech}(2\mathbf{g}_K \mathbf{g}_K^T - \text{diag}(\mathbf{g}_K \mathbf{g}_K^T)) \\ 2 \text{vec}(\mathbf{g}_1 \tilde{\mathbf{w}}_0^T) & \cdots & 2 \text{vec}(\mathbf{g}_K \tilde{\mathbf{w}}_{K-1}^T) \end{bmatrix}^T \quad (3.10)$$

where  $\text{vec}(\cdot)$  is the vectorization operator,  $\text{vech}(\cdot)$  is the half-vectorization operator of a symmetric matrix, and  $\text{diag}(\cdot)$  is a diagonal matrix preserving only the diagonal entries of its matrix argument. Let  $\mathbf{f} \in \mathbb{R}^K$  whose  $(k+1)$ -th entry is

$$[\mathbf{f}]_{k+1} = \|\tilde{\mathbf{r}}_k\|_2^2 - \tilde{\mathbf{w}}_{k+1}^T \mathbf{M}_4 \tilde{\mathbf{w}}_{k+1} + 2\tilde{\mathbf{w}}_{k+1}^T \tilde{\mathbf{A}} \tilde{\mathbf{w}}_k + 2\tilde{\mathbf{w}}_{k+1}^T \tilde{\mathbf{B}} \mathbf{g}_{k+1}$$

and consider the least squares problem

$$\min_{\substack{\hat{\mathbf{M}}_1 \in \mathbb{R}^{n \times n}, \\ \hat{\mathbf{M}}_2 \in \mathbb{R}^{p \times p}, \\ \hat{\mathbf{M}}_3 \in \mathbb{R}^{p \times n}}} \sum_{k=0}^{K-1} \left( \tilde{\mathbf{w}}_k^T \hat{\mathbf{M}}_1 \tilde{\mathbf{w}}_k + \mathbf{g}_{k+1}^T \hat{\mathbf{M}}_2 \mathbf{g}_{k+1} + 2\mathbf{g}_{k+1}^T \hat{\mathbf{M}}_3 \tilde{\mathbf{w}}_k - [\mathbf{f}]_{k+1} \right)^2. \quad (3.11)$$

If  $K \geq (n+p)(n+p+1)/2$  and the data matrix  $\mathbf{D}$  has full rank, the unique solution to (3.11) is  $\hat{\mathbf{M}}_1 = \mathbf{M}_1, \hat{\mathbf{M}}_2 = \mathbf{M}_2, \hat{\mathbf{M}}_3 = \mathbf{M}_3$  with objective value 0.

*Proof.* The least squares problem (3.11) is equivalent to

$$\min_{\hat{\mathbf{o}}} \|\mathbf{D}\hat{\mathbf{o}} - \mathbf{f}\|_2^2 \quad (3.12)$$

where

$$\hat{\mathbf{o}} = \begin{bmatrix} \text{vech}(\hat{\mathbf{M}}_1) \\ \text{vech}(\hat{\mathbf{M}}_2) \\ \text{vec}(\hat{\mathbf{M}}_3) \end{bmatrix} \in \mathbb{R}^{\frac{1}{2}(n+p)(n+p+1)}.$$

As the data matrix  $\mathbf{D}$  is full rank with  $K \geq (n+p)(n+p+1)/2$ , it follows that (3.12) has a unique solution. This implies that (3.11) also has a unique solution due to the equivalence between (3.11) and (3.12). From the residual norm expression (3.8), notice that  $\hat{\mathbf{M}}_1 = \mathbf{M}_1, \hat{\mathbf{M}}_2 = \mathbf{M}_2, \hat{\mathbf{M}}_3 = \mathbf{M}_3$  yields an objective value of 0 for (3.11). Therefore, it is the unique minimizer for the least squares problem (3.11).  $\square$

#### 3.4. Error estimator based on the learned residual norm operators

Consider a queryable system (2.5). The residual trajectory of the re-projected state trajectory can be computed during the re-projection step. Let  $\mathbf{V}_n$  be a basis matrix,  $\mathbf{w}_0^{\text{train}} \in \mathcal{V}_N$  an initial condition, and  $\mathbf{G}^{\text{train}} = [\mathbf{g}_1^{\text{train}}, \dots, \mathbf{g}_K^{\text{train}}] \in \mathcal{G}$  an input trajectory. Consider further the corresponding re-projected trajectory  $\bar{\mathbf{W}}^{\text{train}} = [\bar{\mathbf{w}}_0^{\text{train}}, \dots, \bar{\mathbf{w}}_{K-1}^{\text{train}}]$  and the corresponding residual trajectory  $\bar{\mathbf{R}}^{\text{train}} = [\bar{\mathbf{r}}_0^{\text{train}}, \dots, \bar{\mathbf{r}}_{K-1}^{\text{train}}]$ . Denote by

$$\Psi^{\text{train}} = [(\bar{\mathbf{W}}^{\text{train}})^T \quad (\mathbf{G}^{\text{train}})^T] \in \mathbb{R}^{K \times (n+p)}$$

the data matrix for operator inference and  $\mathbf{D}^{\text{train}}$  the data matrix (3.10) with  $\bar{\mathbf{w}}_k = \bar{\mathbf{w}}_k^{\text{train}}$  and  $\mathbf{g}_k = \mathbf{g}_k^{\text{train}}$ . If  $\Psi^{\text{train}}$  and  $\mathbf{D}^{\text{train}}$  have full rank with  $K \geq (n+p)(n+p+1)/2$ , the reduced model (2.6) can be recovered together with  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$  defined in (3.8) following Section 3.1 and Proposition 4.

Set  $J > 0$  as the number of time steps for prediction and let  $\mathbf{W}^{\text{test}} = [\mathbf{w}_1^{\text{test}}, \dots, \mathbf{w}_J^{\text{test}}]$  be the state trajectory resulting from system (2.5) subject to the initial state  $\mathbf{w}_0^{\text{test}}$  and the input trajectory  $\mathbf{G}^{\text{test}} = [\mathbf{g}_1^{\text{test}}, \dots, \mathbf{g}_J^{\text{test}}] \in \mathcal{G}$ . For the initial state  $\tilde{\mathbf{w}}_0^{\text{test}} = \mathbf{V}_n^T \mathbf{w}_0^{\text{test}}$ , denote by  $\tilde{\mathbf{W}}^{\text{test}} = [\tilde{\mathbf{w}}_1^{\text{test}}, \dots, \tilde{\mathbf{w}}_J^{\text{test}}]$  the associated reduced state trajectory produced by the recovered reduced model derived from operator inference and re-projection. The norm of the residual of the trajectory  $\tilde{\mathbf{W}}^{\text{test}}$  with respect to the high-dimensional model can be computed via (3.8) by invoking  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$  learned as in Proposition 4. Under certain conditions, the state error of  $\tilde{\mathbf{w}}_k^{\text{test}}$  can be bounded as follows.

**Proposition 5.** *If  $\|\mathbf{r}_k^{\text{test}}\|_2, k \in \mathbb{N}$ , is the residual norm of  $\tilde{\mathbf{w}}_k^{\text{test}}$  calculated through (3.8), under the assumption that  $\|\mathbf{A}\|_2 \leq 1$ , the state error of the learned reduced model can be bounded via*

$$\|\mathbf{w}_k^{\text{test}} - \mathbf{V}_n \tilde{\mathbf{w}}_k^{\text{test}}\|_2 \leq \Delta_k^{\mathbf{w}}(\underbrace{1, \dots, 1}_{k+1}; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}}) = \|\mathbf{w}_0^{\text{test}} - \mathbf{V}_n \tilde{\mathbf{w}}_0^{\text{test}}\|_2 + \sum_{l=0}^{k-1} \|\mathbf{r}_l^{\text{test}}\|_2. \quad (3.13)$$

*Proof.* Using the basis matrix  $\mathbf{V}_n$ , the input trajectory  $\mathbf{G}^{\text{test}}$ , the recovered reduced operators  $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$  from Section 3.1, and the recovered matrices  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$  from Proposition 4, the residual norm  $\|\mathbf{r}_k^{\text{test}}\|_2$  can be computed for  $k = 0, \dots, J-1$ .

From (3.5), we deduce

$$\|\mathbf{w}_k^{\text{test}} - \mathbf{V}_n \tilde{\mathbf{w}}_k^{\text{test}}\|_2 \leq \|\mathbf{A}^k\|_2 \|\mathbf{w}_0^{\text{test}} - \mathbf{V}_n \tilde{\mathbf{w}}_0^{\text{test}}\|_2 + \sum_{l=0}^{k-1} \|\mathbf{A}^{k-l-1}\|_2 \|\mathbf{r}_l^{\text{test}}\|_2 \leq \Delta_k^{\mathbf{w}}(\underbrace{1, \dots, 1}_{k+1}; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}}). \quad (3.14)$$

The second inequality in (3.14) holds as  $\|\mathbf{A}^l\|_2 \leq \|\mathbf{A}\|_2^l \leq 1$  for  $0 \leq l \leq k$ .  $\square$

**Remark 6.** Proposition 5 shows that  $\Delta_k^w$  is a pre-asymptotic, computable upper bound on the generalization error of the learned reduced model with respect to control inputs.

The condition stated in Proposition 5 is met, for example, in the following situations. Let the bilinear form  $a$  in (2.2) be symmetric. If  $\beta = 0$  in (2.5) (forward Euler) and the basis functions  $\varphi_i$  are, e.g., orthonormal such that  $\mathbf{M}$  is a multiple of the identity matrix, then  $\mathbf{A}$  is symmetric and there exists a sufficiently small time-step size  $\delta t$  such that the spectral radius  $\rho(\mathbf{A}) = \|\mathbf{A}\|_2 \leq 1$ . Alternatively, certain mass lumping techniques [39] may be applied to attain an  $\mathbf{M}$  with such structure. Finally, if  $\beta = 1$  in (2.5) (backward Euler), it can be shown that there exists  $\delta t$  such that the maximum singular value of  $(\mathbf{I} - \delta t \mathbf{M}^{-1} \mathbf{K})^{-1}$  is at most 1, which relies on the symmetry of  $\mathbf{M}$  and  $\mathbf{K}$ .

**Remark 7.** The error estimator  $\Delta_k^w$  can be reformulated when the state error is measured with respect to a different norm. Consider the reduced basis solution to a symmetric parabolic linear PDE. Let  $(\cdot, \cdot)_{L^2}$  be the inner product on  $L^2$  and  $\{\eta_i\}_{i=1}^n$ ,  $n \ll N$ , be the low-dimensional finite element basis such that the reduced basis solution  $\tilde{w}(\mathbf{x}, t)$  to  $w(\mathbf{x}, t)$  is expressed as  $\tilde{w}(\mathbf{x}, t) = \sum_{i=1}^n \tilde{w}_i(t) \eta_i(\mathbf{x})$ . Define the energy norm for  $v \in \mathcal{V}$  at  $t = t_k$  as

$$\|v(t_k)\|_a^2 = (v(t_k), v(t_k))_{L^2} + \delta t \sum_{k'=1}^k a(v(t_{k'}), v(t_{k'})).$$

It was shown in [11] that at  $t = t_k$ , the error of the reduced basis solution satisfies

$$\|w(\cdot, t_k) - \tilde{w}(\cdot, t_k)\|_a \leq \delta t \sum_{k'=1}^k \check{\mathbf{r}}(t_{k'})^T \mathbf{K}^{-1} \check{\mathbf{r}}(t_{k'})$$

where  $\check{\mathbf{r}}(\cdot) \in \mathbb{R}^N$  is the residual of the reduced basis solution at a particular time and  $\mathbf{K}$  is the stiffness matrix defined in Section 2.2. Using computations analogous to those carried out in [15],  $\check{\mathbf{r}}(t_{k'})^T \mathbf{K}^{-1} \check{\mathbf{r}}(t_{k'})$  admits an expression similar to (3.8) whose operators can be inferred using the strategy in Section 3.3. This enables the calculation of the above bound for other control inputs without recourse to the high-dimensional system. However, obtaining trajectories of  $\check{\mathbf{r}}(t_{k'})^T \mathbf{K}^{-1} \check{\mathbf{r}}(t_{k'})$  poses a challenge to training the error operators since this presumes knowledge of  $\mathbf{K}$  which is unavailable in our setting.

### 3.5. Probabilistic *a posteriori* error estimator for the state

We discuss an approach to bound  $\|\mathbf{A}^l\|_2$ ,  $0 \leq l \leq J$ , if the condition  $\|\mathbf{A}\|_2 \leq 1$  in Proposition 5 is not met or if it is unknown if  $\|\mathbf{A}\|_2 \leq 1$  holds. We seek an upper bound for  $\|\mathbf{A}^l\|_2$  with probabilistic guarantees in order to derive a probabilistic *a posteriori* error estimator for the state in Section 3.5.1. The practical implementation of this error estimator is then discussed in Section 3.5.2. In the following, denote by  $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  the multivariate Gaussian distribution with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ .

#### 3.5.1. Probabilistic upper bound for $\|\mathbf{A}^l\|_2$ and the state error

**Lemma 8.** For  $l \in \mathbb{N}$ , let  $\boldsymbol{\Theta}^{(l)} = \mathbf{A}^l \mathbf{Z}_1$  where  $\mathbf{Z}_1 \sim N(\mathbf{0}_{N \times 1}, \mathbf{I}_N)$  so that  $\boldsymbol{\Theta}^{(l)}$  is an  $N$ -dimensional Gaussian random vector with mean zero and covariance  $\mathbf{A}^l (\mathbf{A}^l)^T$ . Suppose that  $\{\boldsymbol{\Theta}_i^{(l)}\}_{i=1}^M$  are  $M \in \mathbb{N}$  independent and identically distributed  $N$ -dimensional random vectors with the same law as  $\boldsymbol{\Theta}^{(l)}$ . Then, for  $\gamma_l > 0$ ,

$$P \left( \gamma_l \max_{i=1, \dots, M} \|\boldsymbol{\Theta}_i^{(l)}\|_2^2 \geq \|\mathbf{A}^l\|_2^2 \right) \geq 1 - \left[ F_{\chi_1^2} \left( \frac{1}{\gamma_l} \right) \right]^M \quad (3.15)$$

where  $F_{\chi_1^2}$  is the cumulative distribution function of the chi-squared distribution with 1 degree of freedom.

*Proof.* It suffices to show that

$$P\left(\gamma_l \|\Theta^{(l)}\|_2^2 \geq \|\mathbf{A}^l\|_2^2\right) \geq 1 - F_{\chi_1^2}\left(\frac{1}{\gamma_l}\right) \quad (3.16)$$

because using the fact

$$P\left(\gamma_l \max_{i=1,\dots,M} \|\Theta_i^{(l)}\|_2^2 \leq \|\mathbf{A}^l\|_2^2\right) = \left[P(\gamma_l \|\Theta_i^{(l)}\|_2^2 \leq \|\mathbf{A}^l\|_2^2)\right]^M,$$

we conclude that

$$P\left(\gamma_l \max_{i=1,\dots,M} \|\Theta_i^{(l)}\|_2^2 \geq \|\mathbf{A}^l\|_2^2\right) = 1 - \left[P(\gamma_l \|\Theta_i^{(l)}\|_2^2 \leq \|\mathbf{A}^l\|_2^2)\right]^M \geq 1 - \left[F_{\chi_1^2}\left(\frac{1}{\gamma_l}\right)\right]^M$$

as desired. The proof of (3.16) uses ideas similar to that in [5]. Recall that

$$\|\mathbf{A}^l\|_2 = \sqrt{\lambda_{\max}((\mathbf{A}^l)^T \mathbf{A}^l)}$$

where  $\lambda_{\max}(\cdot)$  represents the largest eigenvalue of the matrix argument. Since  $(\mathbf{A}^l)^T \mathbf{A}^l$  is real and symmetric,  $(\mathbf{A}^l)^T \mathbf{A}^l = \mathbf{Q} \Lambda \mathbf{Q}^T$  where  $\mathbf{Q} \in \mathbb{R}^{N \times N}$ ,  $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_N$ , and  $\Lambda$  is a diagonal matrix whose entries  $[\Lambda]_{ii} = \lambda_i$  satisfy  $0 \leq \lambda_1 \leq \dots \leq \lambda_N = \|\mathbf{A}^l\|_2^2$ . By setting  $\mathbf{Z}_2 = \mathbf{Q}^T \mathbf{Z}_1$ , we have  $\mathbf{Z}_2 \sim N(\mathbf{0}_{N \times 1}, \mathbf{I}_N)$  and that

$$\|\Theta^{(l)}\|_2^2 = (\Theta^{(l)})^T \Theta^{(l)} = \mathbf{Z}_1^T (\mathbf{A}^l)^T \mathbf{A}^l \mathbf{Z}_1 = \mathbf{Z}_1^T \mathbf{Q} \Lambda \mathbf{Q}^T \mathbf{Z}_1 = \mathbf{Z}_2^T \Lambda \mathbf{Z}_2 \geq \|\mathbf{A}^l\|_2^2 ([\mathbf{Z}_2]_N)^2$$

where  $[\mathbf{Z}_2]_N$  is the  $N$ -th component of  $\mathbf{Z}_2$ . Since  $[\mathbf{Z}_2]_N \sim N(0, 1)$ ,  $([\mathbf{Z}_2]_N)^2 \sim \chi_1^2$ , i.e. it is a chi-squared random variable with 1 degree of freedom. It follows that for a constant  $\gamma_l > 0$  with

$$P(\gamma_l ([\mathbf{Z}_2]_N)^2 \geq 1) = 1 - F_{\chi_1^2}\left(\frac{1}{\gamma_l}\right)$$

we obtain

$$P(\gamma_l \|\Theta^{(l)}\|_2^2 \geq \|\mathbf{A}^l\|_2^2) \geq P(\gamma_l ([\mathbf{Z}_2]_N)^2 \geq 1) = 1 - F_{\chi_1^2}\left(\frac{1}{\gamma_l}\right).$$

□

**Remark 9.** Results similar to (3.15) can be obtained for other distributions on  $\Theta^{(l)}$  building on, e.g., [3, 5].

Using (3.15), we derive a probabilistic *a posteriori* error estimator as the next result demonstrates.

**Proposition 10.** For  $l = 1, \dots, J$ , let  $\Theta^{(l)} = \mathbf{A}^l \mathbf{Z}$  where  $\mathbf{Z} \sim N(\mathbf{0}_{N \times 1}, \mathbf{I}_N)$  so that  $\Theta^{(l)}$  is an  $N$ -dimensional Gaussian random vector with mean zero and covariance  $\mathbf{A}^l (\mathbf{A}^l)^T$ . Let  $\{\Theta_i^{(l)}\}_{i=1}^M$  be independent and identically distributed  $N$ -dimensional random vectors with the same law as  $\Theta^{(l)}$  and define

$$\Xi_l = \sqrt{\gamma_l \max_{i=1,\dots,M} \|\Theta_i^{(l)}\|_2^2},$$

for  $\gamma_l > 0, l \geq 1$  with  $\Xi_0 = 1$ . For an initial state  $\mathbf{w}_0 \in \mathcal{V}_N$  and an input trajectory  $\mathbf{G}$ , the following holds

$$P\left(\bigcap_{k=1}^J \left\{ \|\mathbf{w}_k - \mathbf{V}_n \tilde{\mathbf{w}}_k\|_2 \leq \Delta_k^{\mathbf{w}}(\Xi_k, \dots, \Xi_0; \mathbf{w}_0, \mathbf{G}) \right\}\right) \geq \max\left(0, 1 - \sum_{l=1}^J \left[F_{\chi_1^2}\left(\frac{1}{\gamma_l}\right)\right]^M\right). \quad (3.17)$$

*Proof.* Define the events  $E_l = \{\Xi_l^2 \geq \|\mathbf{A}^l\|_2^2\}$  for  $l = 1, \dots, J$  and the event

$$E = \bigcap_{k=1}^J \left\{ \|\mathbf{w}_k - \mathbf{V}_n \tilde{\mathbf{w}}_k\|_2 \leq \Delta_k^{\mathbf{w}}(\Xi_k, \dots, \Xi_0; \mathbf{w}_0, \mathbf{G}) \right\}.$$

Recall from (3.7) that

$$\|\mathbf{w}_k - \mathbf{V}_n \tilde{\mathbf{w}}_k\|_2 \leq \Delta_k^{\mathbf{w}}(\|\mathbf{A}^k\|_2, \dots, \|\mathbf{A}^0\|_2; \mathbf{w}_0, \mathbf{G}), \quad k = 1, \dots, J,$$

holds, which means that we obtain

$$P(E) \geq P(\bigcap_{l=1}^J E_l).$$

Using the Fréchet inequality, we obtain

$$P(\bigcap_{l=1}^J E_l) \geq \max \left( 0, \sum_{l=1}^J P(E_l) - (J-1) \right).$$

Therefore, using Lemma 8,

$$P(E) \geq \max \left( 0, \sum_{l=1}^J \left( 1 - \left[ F_{\chi_1^2} \left( \frac{1}{\gamma_l} \right) \right]^M \right) - (J-1) \right) = \max \left( 0, 1 - \sum_{l=1}^J \left[ F_{\chi_1^2} \left( \frac{1}{\gamma_l} \right) \right]^M \right).$$

□

### 3.5.2. Sampling random vectors from queryable systems

We now discuss a practical implementation of the probabilistic error bound in Proposition 10. We resume the setup outlined in Section 3.4. Recall that the reduced model (2.6) and  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$  are recovered using the input trajectory  $\mathbf{G}^{\text{train}}$ . Also,  $\mathbf{W}^{\text{test}}$  and  $\tilde{\mathbf{W}}^{\text{test}}$  are the state and reduced state trajectories owing to the input trajectory  $\mathbf{G}^{\text{test}}$  while  $\|\mathbf{r}_k^{\text{test}}\|_2$  is the residual norm of  $\tilde{\mathbf{w}}_k^{\text{test}}$  calculated through (3.8).

To construct an upper bound for  $\|\mathbf{w}_k^{\text{test}} - \mathbf{V}_n \tilde{\mathbf{w}}_k^{\text{test}}\|_2$  according to Proposition 10, realizations of the random vectors  $\Theta^{(l)} \sim N(\mathbf{0}_{N \times 1}, \mathbf{A}^l (\mathbf{A}^l)^T)$ ,  $l = 1, \dots, k$  need to be simulated. Therefore, for fixed  $M$ , if  $\{\mathbf{z}_i\}_{i=1}^M$  are realizations of  $\mathbf{Z}$ , realizations  $\{\theta_i^{(l)}\}_{i=1}^M$  of  $\Theta^{(l)}$  and hence a single realization

$$\xi_l = \sqrt{\gamma_l \max_{i=1, \dots, M} \|\theta_i^{(l)}\|_2^2}$$

of  $\Xi_l$  for  $l = 1, \dots, J$  can be simulated by querying (2.5) for  $J$  time steps with control input  $\mathbf{g}_k = \mathbf{0}_{p \times 1}$  for all  $k$  and with the realizations  $\{\mathbf{z}_i\}_{i=1}^M$  serving as  $M$  initial states, i.e.  $\mathbf{w}_0 = \mathbf{z}_i$  for  $i = 1, \dots, M$ . This produces  $M$  trajectories of  $\mathbf{w}_k = \mathbf{A}^k \mathbf{w}_0$ . Note that  $\xi_0 = 1$ .

For specified  $\gamma_l > 0$  which controls the confidence level (failure probability) of the probabilistic error estimator in (3.17), an error estimate for  $\|\mathbf{w}_k^{\text{test}} - \mathbf{V}_n \tilde{\mathbf{w}}_k^{\text{test}}\|_2$  for  $k = 1, \dots, J$  is provided by

$$\Delta_k^{\mathbf{w}}(\xi_k, \dots, \xi_0; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}}) = \xi_k \|\mathbf{w}_0^{\text{test}} - \mathbf{V}_n \tilde{\mathbf{w}}_0^{\text{test}}\|_2 + \sum_{l=0}^{k-1} \xi_{k-l-1} \|\mathbf{r}_l^{\text{test}}\|_2 \quad (3.18)$$

which we refer to as learned error estimate. Observe that  $\Delta_k^{\mathbf{w}}(\xi_k, \dots, \xi_0; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}})$  is a single realization of the error estimator  $\Delta_k^{\mathbf{w}}(\Xi_k, \dots, \Xi_0; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}})$ .

**Remark 11.** *Bounds on an output, a quantity of interest which is obtained via a linear functional of the state  $\mathbf{w}_k$ , can also be formulated if the norm of the output operator is available. Let the output at time  $k$  be expressed as*

$$y_k = \mathbf{C}\mathbf{w}_k$$

for which it is assumed that  $\|\mathbf{C}\|_2$  is known. The output for the low-dimensional system  $\tilde{y}_k$  is therefore

$$\tilde{y}_k = \mathbf{C}\mathbf{V}_n\tilde{\mathbf{w}}_k.$$

Following [15], since

$$\|y_k - \tilde{y}_k\|_2 = \|\mathbf{C}(\mathbf{w}_k - \mathbf{V}_n\tilde{\mathbf{w}}_k)\|_2 \leq \|\mathbf{C}\|_2\|\mathbf{w}_k - \mathbf{V}_n\tilde{\mathbf{w}}_k\|_2,$$

and  $\Delta_k^w(\xi_k, \dots, \xi_0; \mathbf{w}_0, \mathbf{G})$  is an error estimate for  $\|\mathbf{w}_k - \mathbf{V}_n\tilde{\mathbf{w}}_k\|_2$ , we deduce that  $\tilde{y}_k - \Delta_k^y$  and  $\tilde{y}_k + \Delta_k^y$  are lower and upper bound estimates for  $y_k \in \mathbb{R}$  where

$$\Delta_k^y = \|\mathbf{C}\|_2\Delta_k^w(\xi_k, \dots, \xi_0; \mathbf{w}_0, \mathbf{G}). \quad (3.19)$$

### 3.6. Computational procedure for offline and online phase

The proposed offline-online computational procedure for non-intrusive model reduction of certified reduced models is summarized in Algorithm 2. It builds on the reprojection scheme in Algorithm 1 introduced in [27] which is modified to include computation of the residual trajectory. The offline phase serves as a training stage to determine the unknown quantities while the online phase utilizes these for certified predictions.

The inputs to Algorithm 2 include the number of time steps  $K$  (training),  $J$  (prediction), initial condition  $\mathbf{w}_0$ , the snapshot matrix  $\mathbf{W}^{\text{basis}} = [\mathbf{w}_0, \mathbf{w}_1^{\text{basis}}, \dots, \mathbf{w}_K^{\text{basis}}]$  owing to the input trajectory  $\mathbf{G}^{\text{basis}} = [\mathbf{g}_1^{\text{basis}}, \dots, \mathbf{g}_K^{\text{basis}}] \in \mathcal{G}$ , the basis dimension  $n$ , the input trajectories  $\mathbf{G}^{\text{train}}, \mathbf{G}^{\text{test}}$  for training and prediction,  $M \geq 1$ ,  $\{\gamma_l\}_{l=1}^J$ , and the input trajectory  $\mathbf{G}^{\text{norm}} = [\mathbf{0}_{p \times 1}, \dots, \mathbf{0}_{p \times 1}] \in \mathbb{R}^{p \times J}$  for finding an upper bound for  $\|\mathbf{A}^l\|_2$ ,  $1 \leq l \leq J$ , and the computational model (2.5) that can be queried.

The offline stage constitutes operator inference with reprojection (Section 3.1) and estimation of state error upper bounds (Sections 3.3, 3.5) with the input trajectory  $\mathbf{G}^{\text{train}} \in \mathcal{G}$ . It is composed of three parts: inferring the reduced system, inferring the residual-norm operator, and finding an upper bound for the norm of  $\mathbf{A}$  in the error estimator. The offline phase proceeds by building the low-dimensional basis  $\mathbf{V}_n$  from trajectories of the state contained in  $\mathbf{W}^{\text{basis}}$ . The re-projection algorithm is then invoked to generate the re-projected states  $\bar{\mathbf{w}}_k^{\text{train}}$  and its residual  $\bar{\mathbf{r}}_k^{\text{train}}$  corresponding to the control input  $\mathbf{G}^{\text{train}}$ . Using data on  $\bar{\mathbf{w}}_k^{\text{train}}$  and  $\mathbf{G}^{\text{train}}$ , the least squares problem (3.1) is formulated in order to recover the reduced system (2.6) in a non-intrusive manner. The second part of the offline stage utilizes the inferred reduced system and data on the residual  $\bar{\mathbf{r}}^{\text{train}}$  to set up the least squares problem (3.11). Solving (3.11) yields the operators  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$ , which enable the computation of the residual norm (3.8) at any time for a specified control input. Finally, upper bounds for the operator norms  $\|\mathbf{A}^l\|_2$  in the *a posteriori* error expression (3.6) are sought by querying the system (2.5) at initial conditions consisting of  $M$  realizations of  $\mathbf{Z} \sim N(\mathbf{0}_{N \times 1}, \mathbf{I}_N)$ . The  $M$  trajectories corresponding to each initial condition are employed in the definition of  $\xi_l$  which is a realization of the probabilistic bound  $\Xi_l$ , i.e.  $\|\mathbf{A}^l\|_2 \leq \Xi_l$ . Notice that Algorithms 1 and 2 do not rely on knowledge of  $\mathbf{A}, \mathbf{B}$  in (2.5) and  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$  in (3.8). Furthermore, it is unnecessary to use the same input trajectory  $\mathbf{G}^{\text{train}}$  for solving the least squares problems (3.1) and (3.11).

In the online stage, the deduced quantities in the offline stage are invoked to compute the low-dimensional solution (2.6), the norm of its residual (3.8), and consequently an upper bound for the state error (3.13) or (3.18) for an input trajectory  $\mathbf{G}^{\text{test}} \in \mathcal{G}$ .

Algorithm 2 serves as the reference for the numerical examples undertaken in Section 4.

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

- 
- 1: Set  $\bar{\mathbf{w}}_0 = \mathbf{V}_n^T \mathbf{w}_0$
  - 2: **for**  $k = 0, \dots, K - 1$  **do**
  - 3:   Query (2.5) for a single time step to obtain  $\mathbf{w}_{tmp} = \mathbf{A}\mathbf{V}_n \bar{\mathbf{w}}_k + \mathbf{B}\mathbf{g}_{k+1}$
  - 4:   Set  $\bar{\mathbf{w}}_{k+1} = \mathbf{V}_n^T \mathbf{w}_{tmp}$
  - 5:   Compute the residual  $\bar{\mathbf{r}}_k = \mathbf{w}_{tmp} - \mathbf{V}_n \bar{\mathbf{w}}_{k+1}$
  - 6: **end for**
  - 7: Return  $[\bar{\mathbf{w}}_0, \bar{\mathbf{w}}_1, \dots, \bar{\mathbf{w}}_K]$  and  $[\bar{\mathbf{r}}_0, \bar{\mathbf{r}}_1, \dots, \bar{\mathbf{r}}_{K-1}]$
- 

**Algorithm 2** Data-driven model reduction**Offline phase**

- 1: Construct a low-dimensional basis  $\mathbf{V}_n$  from the snapshot matrix  $\mathbf{W}^{\text{basis}}$
- 2: Generate  $\{\bar{\mathbf{w}}_k^{\text{train}}\}_{k=0}^K$  via re-projection and its residual  $\{\bar{\mathbf{r}}_k^{\text{train}}\}_{k=0}^{K-1}$  (Algorithm 1) using  $\mathbf{G}^{\text{train}}$
- 3: Perform operator inference by solving (3.1) to deduce  $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$
- 4: Infer  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$  from (3.11) for the computation of (3.8)
- 5: Simulate  $M$  realizations  $\{\mathbf{z}_i\}_{i=1}^M$  of  $\mathbf{Z} \sim N(\mathbf{0}_{N \times 1}, \mathbf{I}_N)$
- 6: Produce  $M$  realizations  $\{\boldsymbol{\theta}_i^{(l)}\}_{i=1}^M$  of  $\boldsymbol{\Theta}^{(l)}$  for  $l = 1, \dots, J$  by querying (2.5) for  $J$  time steps with  $\mathbf{w}_0 = \mathbf{z}_i$ ,  $i = 1, \dots, M$  and input  $\mathbf{G}^{\text{norm}}$
- 7: Compute  $\xi_l = \sqrt{\gamma_l \max_{i=1, \dots, M} \|\boldsymbol{\theta}_i^{(l)}\|_2^2}$  for  $l = 1, \dots, J$

**Online phase**

- 8: Calculate the low-dimensional solution  $\{\tilde{\mathbf{w}}_k^{\text{test}}\}_{k=1}^J$  to (2.6) using the inferred  $\tilde{\mathbf{A}}, \tilde{\mathbf{B}}$  and input  $\mathbf{G}^{\text{test}}$
  - 9: Evaluate  $\|\mathbf{r}_k^{\text{test}}\|_2^2$  for  $k = 0, \dots, J - 1$  in (3.8) utilizing the deduced  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$
  - 10: Estimate the *a posteriori* error for the state via (3.13) or (3.18) for  $k = 1, \dots, J$
- 

**3.7. Computational cost comparison**

We compare the computational cost for certifying low-dimensional models under intrusive and non-intrusive model reduction. The cost for both approaches in the online phase is identical. In the offline stage, the procedure of generating snapshots and a reduced basis are the same in the intrusive and the proposed non-intrusive approach. The intrusive approach then assembles the reduced operators via matrix-matrix products (2.7) and sets up the error estimator by computing the quantities  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \{\|\mathbf{A}^l\|_2\}_{l=1}^J$  which involves the matrix-matrix products in (3.9) and computing the largest singular value of a matrix.

In contrast, the proposed non-intrusive approach samples trajectories via re-projection in which the full model is queried. The reduced operators are then fitted to data by solving the least squares problem (3.1) which scales according to  $\mathcal{O}(K(n+p)^2)$ . Finally, the error-estimator quantities  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \{\xi_l\}_{l=1}^J$  are learned via least-squares regression (3.12), which scales as

$$\mathcal{O}\left(K \left(\frac{(n+p)(n+p+1)}{2}\right)^2\right),$$

and by simulating the full model to realize probabilistic bounds of  $\{\|\mathbf{A}^l\|_2\}_{l=1}^J$ .

**4. NUMERICAL RESULTS**

The numerical examples in this section illustrate the following points: 1) the quantities for error estimators are learned from data up to numerical errors, 2) the learned low-dimensional system and the residual norm for

the *a posteriori* error estimators are exact reconstructions of those resulting from intrusive model reduction, 3) the learned quantities can be used to predict the low-dimensional solution and provide error estimates for specified control inputs, and 4) error estimators for the output, i.e. quantity of interest, can be deduced if the output operator is linear in the state and its norm is available.

An implementation of operator inference with the proposed non-intrusive error estimation is available in Python<sup>1</sup>. It requires a blackbox function that timesteps the high-dimensional system to perform non-intrusive model reduction. It computes the reduced model operators and operators in the error estimator for both types of model reduction. The sample script provided reproduces Figure 1b and 2c below.

#### 4.1. Error quantities

We compute the following quantities to assess the predictive capabilities of reduced models learned from data for test input trajectories  $\mathbf{G}^{\text{test}}$  and test initial conditions  $\mathbf{w}_0^{\text{test}}$ .

Error of the reduced solution:

$$e^{(1)} = \frac{\|\mathbf{W}^{\text{test}} - \mathbf{V}_n \check{\mathbf{W}}^{\text{test}}\|_F}{\|\check{\mathbf{W}}^{\text{test}}\|_F} \quad (4.1)$$

where  $\check{\mathbf{W}}^{\text{test}}$  refers to the trajectory of the reduced system inferred via intrusive model reduction ( $\hat{\mathbf{W}}^{\text{test}}$ ) or operator inference ( $\tilde{\mathbf{W}}^{\text{test}}$ ) and  $\|\cdot\|_F$  is the Frobenius norm.

Time-averaged residual norm:

$$e^{(2)} = \frac{1}{J} \sum_{k=0}^{J-1} \|\mathbf{r}_k^{\text{test}}\|_2 \quad (4.2)$$

where the residual norm  $\|\mathbf{r}_k^{\text{test}}\|_2$  is computed through the two approaches for model reduction we compare: intrusive ( $\|\hat{\mathbf{r}}_k^{\text{test}}\|_2$ ) vs operator inference ( $\|\tilde{\mathbf{r}}_k^{\text{test}}\|_2$ ).

Relative average state error over time and its corresponding *a posteriori* error estimates tabulated in Table 1.

In (4.5),  $\xi_k$ ,  $k = 0, \dots, J$ , are realizations of the random variables  $\Xi_k$  defined in Proposition 10. In our experiments, we set  $\gamma_l = \gamma$  for  $l = 1, \dots, J$  so that the probability lower bound in (3.17) becomes

$$P^{\text{LB}}(\gamma, M, J) = \max \left( 0, 1 - J \left[ F_{\chi_1^2} \left( \frac{1}{\gamma} \right) \right]^M \right). \quad (4.7)$$

Relative state error at a particular time point  $k$  and its corresponding *a posteriori* error estimates tabulated in Table 2.

#### 4.2. Heat transfer

The setup for non-intrusive model reduction applied to this example is first described which is followed by the numerical results.

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

Errors and error estimators	Definition
error of reduced solution via operator inference	$e^{(3)} = \frac{\sum_{k=0}^{J-1} \ \mathbf{w}_k^{\text{test}} - \mathbf{V}_n \tilde{\mathbf{w}}_k^{\text{test}}\ _2}{J \sum_{k=0}^{J-1} \ \mathbf{w}_k^{\text{test}}\ _2} \quad (4.3)$
intrusive model reduction upper bound for the state error	$\Delta^{(1)} = \frac{\sum_{k=0}^{J-1} \Delta_k^{\mathbf{w}}(\ \mathbf{A}^k\ _2, \dots, \ \mathbf{A}^0\ _2; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}})}{J \sum_{k=0}^{J-1} \ \mathbf{w}_k^{\text{test}}\ _2} \quad (4.4)$
realization of probabilistic upper bound for the state error	$\Delta^{(2)} = \frac{\sum_{k=0}^{J-1} \Delta_k^{\mathbf{w}}(\xi_k, \dots, \xi_0; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}})}{J \sum_{k=0}^{J-1} \ \mathbf{w}_k^{\text{test}}\ _2} \quad (4.5)$
learned deterministic upper bound for the state error	$\Delta^{(3)} = \frac{\sum_{k=0}^{J-1} \Delta_k^{\mathbf{w}}(1, \dots, 1; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}})}{J \sum_{k=0}^{J-1} \ \mathbf{w}_k^{\text{test}}\ _2} \quad (4.6)$

TABLE 1. Relative average state error over time and its corresponding error estimates obtained from intrusive model reduction and operator inference.

Errors and error estimators	Definition
error of reduced solution via operator inference	$e^{(4)} = \frac{\ \mathbf{w}_k^{\text{test}} - \mathbf{V}_n \tilde{\mathbf{w}}_k^{\text{test}}\ _2}{\ \mathbf{w}_k^{\text{test}}\ _2} \quad (4.8)$
intrusive model reduction upper bound for the state error	$\Delta^{(4)} = \frac{\Delta_k^{\mathbf{w}}(\ \mathbf{A}^k\ _2, \dots, \ \mathbf{A}^0\ _2; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}})}{\ \mathbf{w}_k^{\text{test}}\ _2} \quad (4.9)$
realization of probabilistic upper bound for the state error	$\Delta^{(5)} = \frac{\Delta_k^{\mathbf{w}}(\xi_k, \dots, \xi_0; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}})}{\ \mathbf{w}_k^{\text{test}}\ _2} \quad (4.10)$
learned deterministic upper bound for the state error	$\Delta^{(6)} = \frac{\Delta_k^{\mathbf{w}}(1, \dots, 1; \mathbf{w}_0^{\text{test}}, \mathbf{G}^{\text{test}})}{\ \mathbf{w}_k^{\text{test}}\ _2} \quad (4.11)$

TABLE 2. Relative state error at a particular time point  $k$  and its corresponding error estimates obtained from intrusive model reduction and operator inference.

#### 4.2.1. Setup

For  $\Omega = (0, 1)$ ,  $\mathcal{T} = (0, T)$ ,  $T = 5$ , consider the heat equation on  $(x, t) \in \Omega \times \mathcal{T}$  given by

$$\begin{aligned} \frac{\partial}{\partial t} w(x, t) &= \mu \frac{\partial^2}{\partial x^2} w(x, t), \\ w(0, t) &= 0, \\ \frac{\partial}{\partial x} w(1, t) &= u(t) \\ w(x, 0) &= 0. \end{aligned}$$

To discretize the PDE,  $\Omega$  is subdivided into  $N = 133$  intervals with width  $\Delta x = 1/N$ . Let  $\{\varphi_i\}_{i=1}^N$  be linear hat basis functions with  $\varphi_i(j\Delta x) = \delta_{ij}$  where  $\delta_{ij}$  is the Kronecker delta function. We obtain the continuous-time system

$$\mathbf{M} \frac{d\mathbf{w}(t)}{dt} = \mathbf{K} \mathbf{w}(t) + \mu \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u(t) \end{bmatrix}$$

where  $[\mathbf{K}]_{ij} = -\mu \int_0^1 \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} dx$  for  $i, j = 1, \dots, N$ . In our simulation, we set  $\mu = 0.1$  for the diffusivity parameter and temporally discretized the continuous system using backward Euler with  $\delta t = 0.01$  being the time step size.

The basis  $\mathbf{V}_n$  was constructed from the snapshot matrix of  $K = 500$  time steps driven by the control input  $u^{\text{basis}}(t) = e^t \sin(20\pi t/T)$ . The objective functions (3.1) and (3.12) were optimized using the input trajectory  $\mathbf{G}^{\text{train}} = [0, z_1, \dots, z_{K-1}]$  where  $z_i$  is a realization of  $Z_i$ ,  $i = 1, \dots, K-1$ , which are independent and identically distributed  $N(0, 1)$  random variables. The training control input need not be random but it aids in exciting the system to produce a full rank data matrix for the least squares problems we solve. Random signals are also utilized in system identification [24]. If information is available on a specific signal that is appropriate for a particular system, it could also be utilized in the training stage.

#### 4.2.2. Results

We now assess the performance of the learned reduced model and quantities required for *a posteriori* error estimation. In the online stage, the control input  $u^{\text{test}}(t) = e^t \sin(12\pi t/T)$  was discretized using 500 time steps. The quantities listed in Section 4 are calculated up to  $n = 8$  basis vectors.

Figure 1 demonstrates that the reduced system and quantities required for error estimation can be recovered up to numerical errors. In particular, Figure 1a plots (4.1) for the reduced solution resulting from intrusive model reduction compared to that from operator inference. It demonstrates the theory established in earlier work [27] on the recovery of the reduced operators in the system (2.6). Due to this, the error of the reduced trajectory from either approach is almost identical. The quantity (4.2) involving the residual norm for both approaches of model reduction is presented in Figure 1b. The plot shows that both methods are in close agreement. If the conditions in Proposition 4 are met, the matrices  $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$  in (3.8) and hence the residual norm itself can also be recovered.

In this simulation, our knowledge of  $\mathbf{M}, \mathbf{K}$  informed the choice of  $\delta t$  so that  $\|\mathbf{A}\|_2 \leq 1$  and thus, the deterministic error estimator (3.13) is applicable. If this is not the case, the probabilistic error estimator introduced in Section 3.5 can be utilized instead. Figure 2 shows the deterministic and probabilistic *a posteriori* error estimates. For the probabilistic error estimate, we chose  $\gamma = 1, M = 25, J = 500$  so that  $P^{\text{LB}}(\gamma, M, J) \approx 0.9641$ . Only one realization of each of the random variables  $\Xi_l$ ,  $l = 1, \dots, J$  was generated for this example. Figures 2a and 2b display the learned reduced model error (4.8) and the intrusive (4.9), probabilistic (4.10), and deterministic (4.11) error estimates at  $t = 1$  and  $t = 5$ , respectively. These plots depict the intrusive

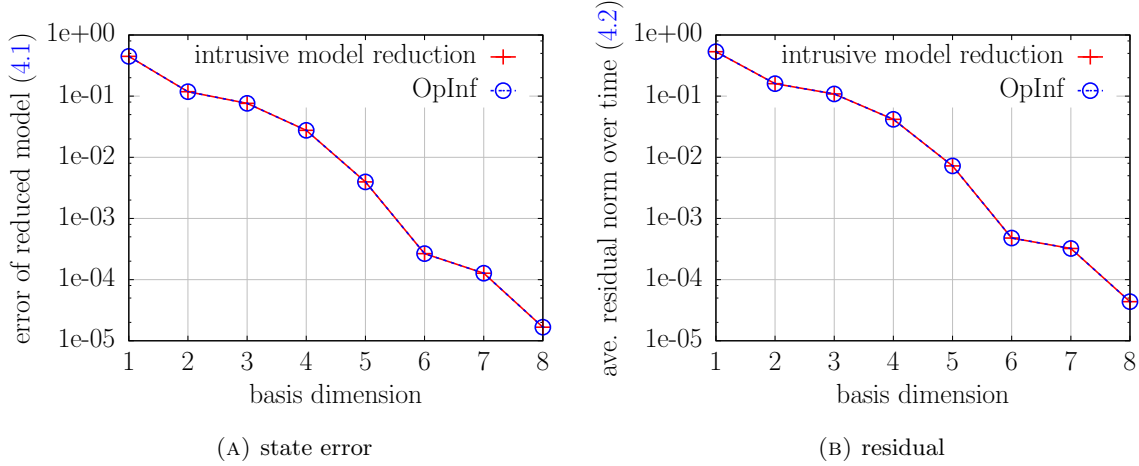


FIGURE 1. Heat equation: The results in plots (a)-(b) indicate that the reduced system and the quantities required for error estimation under operator inference, i.e. residual norm operators, are equal to their intrusive counterparts up to numerical errors.

model reduction error estimate (3.7) for the state error. We notice that the intrusive and deterministic (non-intrusive) error estimates are almost identical. In addition, the plots convey that the learned error estimate (3.18) under operator inference is roughly of the same order of magnitude as the error estimate provided by the intrusive approach. The calculated quantities for the time-averaged learned reduced model error (4.3) and its corresponding intrusive (4.4), probabilistic (4.5), and deterministic (4.6) error estimates are likewise shown in Figure 2c. The plot reveals that the behavior of the time-averaged relative state error is similar to that of the relative state error at various time instances.

In practice, the learned error estimator may depend on the realizations of the random variables  $\Theta^{(l)}$  simulated. In all simulations described above, we performed calculations using only a single realization of  $\Xi_l$ . We therefore generate multiple realizations of  $\Xi_l$  and study the variability in the resulting learned error estimate associated with various sets of realizations of  $\Theta^{(l)}$ . Figure 3 compiles the mean (solid) of 100 realizations of the learned error estimator (4.10) for  $t = 1$  and  $t = 5$  and (4.5) in Figures 3a, 3a, and 3c respectively. In each panel, the vertical bars symbolize the minimum and maximum among the simulated realizations while the error estimate from the intrusive approach is also shown. We observe from the minimum and maximum values that there is low variability in the learned error estimates generated.

#### 4.2.3. Computational cost

Finally, we compare the offline runtime for certifying low-dimensional models under intrusive and non-intrusive model reduction. Note that the online runtime for both is identical. The comparison is carried out for the experiment corresponding to Figure 2. The results are shown in Figure 4 where the runtime is reported for reduced dimensions between 1 and 8, averaged over 100 runs. The runtime of both approaches is dominated by setting up the error estimator. For intrusive model reduction, this is due to computing  $\|\mathbf{A}^l\|_2$  for  $l = 1, \dots, J$ . For the proposed non-intrusive approach, this is due to querying the high-dimensional system to obtain realizations of  $\xi_l, l = 1, \dots, J$ . The runtime of solving the least-squares problem (3.12) is negligible in this example.

### 4.3. Convection-diffusion in a pipe

The setup for this problem is first described followed by the numerical results for two types of control inputs and bounds on the output error.

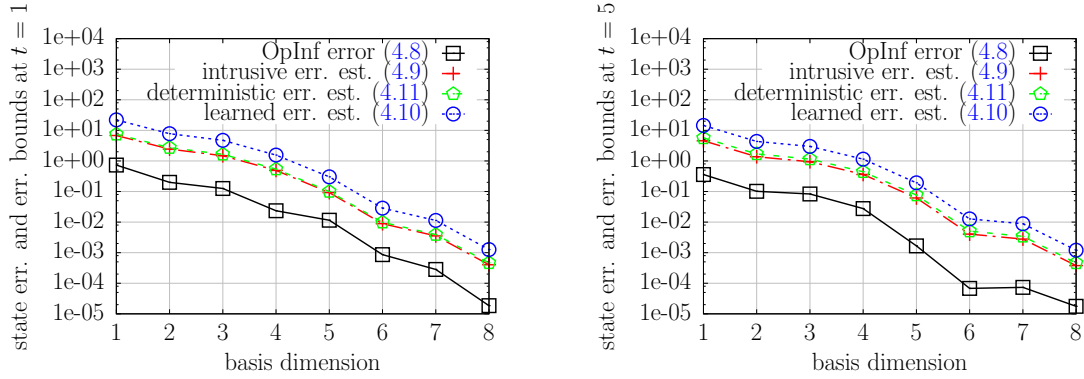
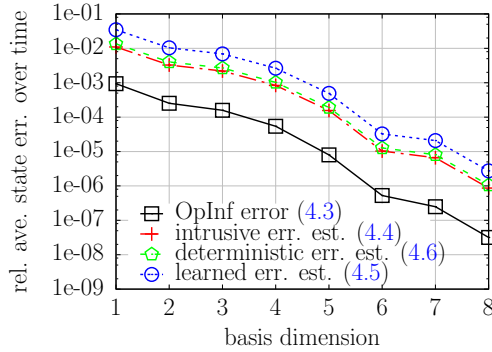
(A) error and error estimates at  $t = 1$ (B) error and error estimates at  $t = 5$ (C) average error and error estimates over  $t \in [0, 5]$ 

FIGURE 2. Heat equation: The plots illustrate that the deterministic (non-intrusive) error estimator (3.13) and the learned (probabilistic) error estimator derived in Proposition 10 bound the error of the reduced solution in this example. The intrusive and deterministic error estimates are close. In addition, the learned error estimator indicates an error of the same order of magnitude as the intrusive error estimator. The parameters used for the learned error estimator were chosen as  $\gamma = 1, M = 25, J = 500$  so that the learned estimator gives an upper with probability  $P^{LB} \approx 0.9641$ .

#### 4.3.1. Setup

We now consider a parabolic PDE over a 2-D spatial domain according to the convection-diffusion equation. Let  $\mathcal{T} = (0, 0.5)$  and  $\Omega = (0, 1) \times (0, 0.25)$ . For  $(x_1, x_2, t) \in \Omega \times \mathcal{T}$ , the PDE examined is

$$\begin{aligned} \frac{\partial}{\partial t} w(x_1, x_2, t) &= \nabla \cdot (\mu \nabla w(x_1, x_2, t)) - (1, 1) \cdot \nabla w(x_1, x_2, t), \\ w(x_1, x_2, t) &= 0 \text{ for } (x_1, x_2) \in \partial\Omega \setminus \cup_{i=1}^5 E_i, \\ \nabla w(x_1, x_2, t) \cdot \mathbf{n} &= u_i(t) \text{ for } (x_1, x_2) \in E_i, i = 1, \dots, 5, \\ w(x_1, x_2, 0) &= 0. \end{aligned} \tag{4.12}$$

where the domain  $\Omega$  and the segments  $E_i, i = 1, \dots, 5$  with Neumann conditions are visualized in Figure 5.

The finite element discretization is constructed using square elements with width  $\Delta x_1 = \Delta x_2 = 1/75$  and associated linear hat basis functions  $\{\varphi_i(x_1, x_2)\}_{i=1}^N$  where  $N = 1121$ . The continuous-time system resulting

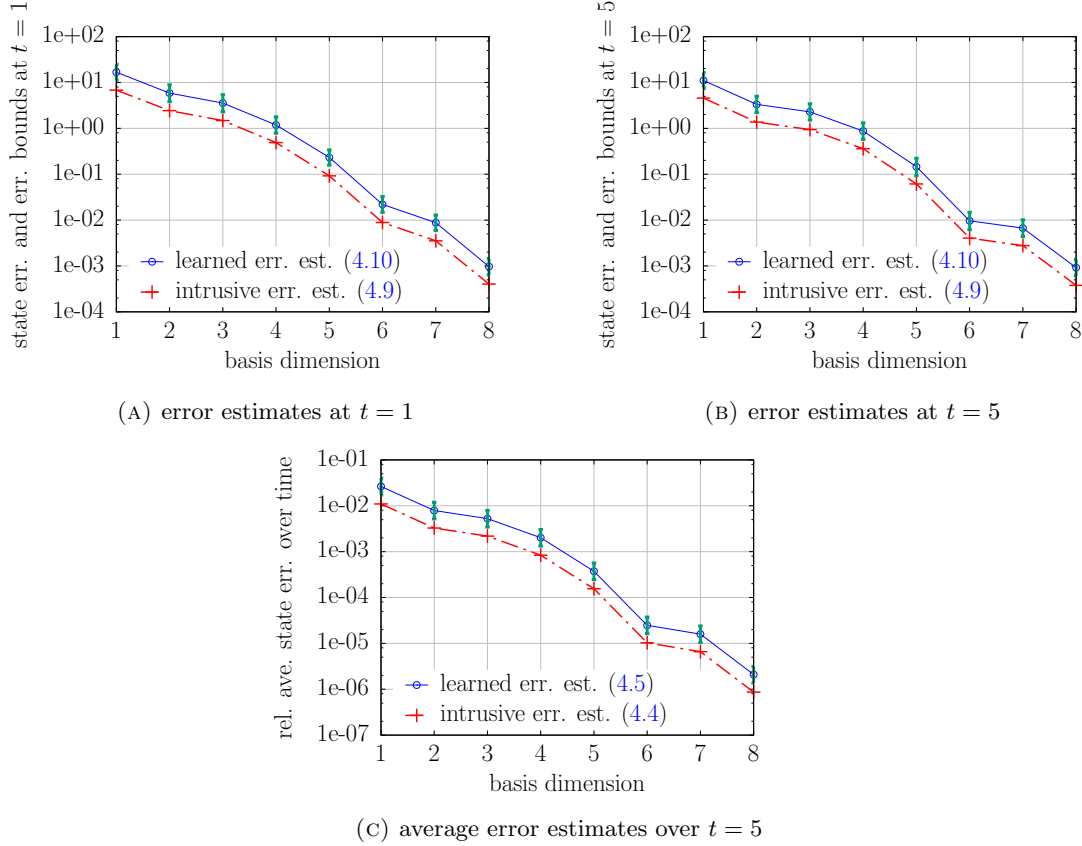


FIGURE 3. Heat equation: The mean, minimum, and maximum of the quantities (4.10) and (4.5) computed from 100 samples of the learned error estimator are shown. It is observed that there is low variation among the samples of the learned error estimator.

from this PDE is

$$\mathbf{M} \frac{d\mathbf{w}(t)}{dt} = \mathbf{K}\mathbf{w}(t) + \mathbf{F}\mathbf{u}(t)$$

where  $\mathbf{M}$  is the mass matrix as before,  $[\mathbf{K}]_{ij} = -\mu \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i d\mathbf{x} - \int_{\Omega} ((1,1) \cdot \nabla \varphi_j) \varphi_i d\mathbf{x}$  for  $i, j = 1, \dots, N$  and  $[\mathbf{F}]_{ij} = \mu \int_{E_j} \varphi_i d\mathbf{x}$  for  $i = 1, \dots, N, j = 1, \dots, 5$ . This was then discretized using forward Euler with the time step size  $\delta t = 10^{-5}$ .

Two variants of this problem are investigated in Sections 4.3.2 and 4.3.3 in which we implemented different pairs of control signals ( $\mathbf{u}^{\text{basis}}(t), \mathbf{u}^{\text{test}}(t)$ ) in each variation. The control inputs for basis generation and the testing stage are more similar in nature in Section 4.3.2 than in Section 4.3.3. The same control input  $\mathbf{u}^{\text{train}}(t)$  is used to solve the optimization problems (3.1) and (3.10) for both variants which is discretized to obtain  $\mathbf{G}^{\text{train}}$ . The trajectory  $\mathbf{G}^{\text{train}}$  was simulated as follows: for the time points  $\{t_k\}_{k=0}^{K-1}$ ,  $K = 5 \times 10^4$ ,  $[\mathbf{g}_k]_j$  is a realization of  $Z_k^{(j)} \sim N(0, \sin^2(j\pi t_k))$  such that  $Z_k^{(j)}, Z_l^{(j)}$  are independent for  $k, l = 1, \dots, K-1, k \neq l$ .

#### 4.3.2. Results for exponentially growing sinusoidal control input

The diffusivity parameter in this example is set to  $\mu = 0.5$ . The basis  $\mathbf{V}_n$  is constructed using the control input  $u_j^{\text{basis}}(t) = \sin(2jt), j = 1, \dots, 5$  while the control input  $u_j^{\text{test}}(t) = e^t \sin(1.75jt), j = 1, \dots, 5$  is used for prediction in the online stage. Both of these control inputs are discretized in time using  $K = 5 \times 10^4$  (basis)

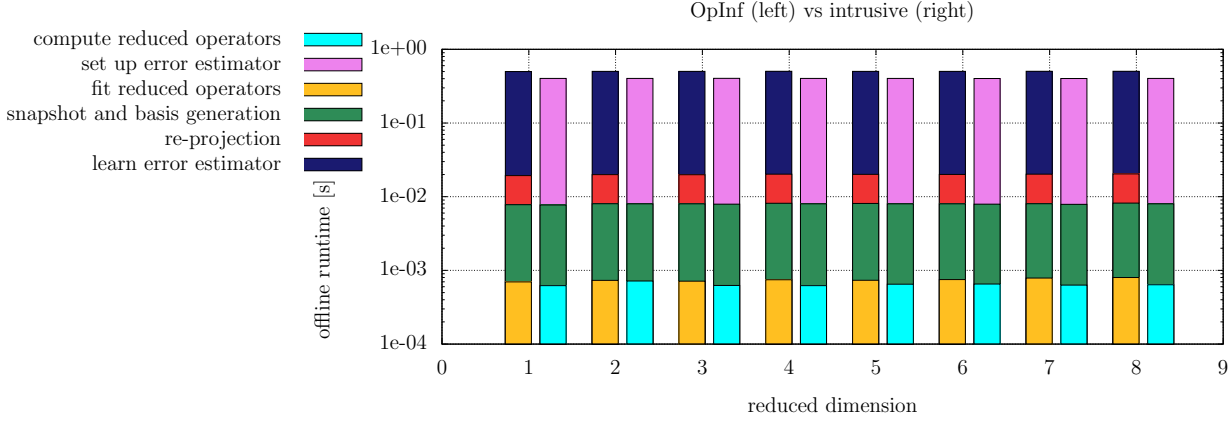


FIGURE 4. Heat equation. Comparison of runtime of the offline phase for intrusive model reduction and the proposed non-intrusive approach. Setting up the error estimator dominates the offline costs. In case of intrusive model reduction, the offline costs are dominated by computing  $\|\mathbf{A}^l\|_2, l = 1, \dots, J$ . In case of the proposed non-intrusive approach, the offline costs are dominated by generating realizations  $\xi_l, l = 1, \dots, J$ , for the probabilistic bound.

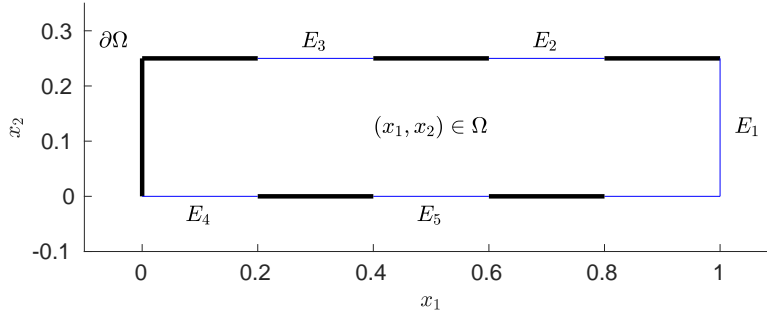


FIGURE 5. Domain  $\Omega$  for the convection-diffusion PDE in Section 4.3 with segments of the boundary with Neumann conditions indicated by thin solid lines.

and  $J = 5 \times 10^4$  (prediction) intervals of equal width. To visualize trajectories of the high-dimensional system resulting from the control input  $\mathbf{u}^{\text{test}}(t)$ , Fig 6 illustrates  $w(x_1, x_2, t)$  for  $t = 0.1, 0.5$ .

We now examine the accuracy of the inferred reduced model and its state error estimate under operator inference by computing the errors listed above. The quantity (4.2) corresponding to the intrusive and operator inference approach as a function of the basis dimension  $n$  is contrasted in Figure 7. The plot demonstrates the recovery of the residual norm (3.8) in the latter method. The reduced system operators for both methods are also almost identical.

We then investigate the effect of the parameters  $\gamma$  and  $M$  in the learned error estimator (3.18) in Figures 8 and 9. Figures 8a and 8b depict the learned reduced model error (4.8) and the intrusive (4.9) and learned (probabilistic) (4.10) error estimates at times  $t = 0.1$  and  $t = 0.5$ . Each panel presents, for  $\gamma = 7, 20, 50$ , a single realization (3.18) of the probabilistic error estimator (3.17) using  $M = 10$  with their respective lower bound probabilities of  $P^{\text{LB}} \approx 0.7543, 0.9985, 0.9999$ . The same set of realizations of  $\Xi_l$  for  $l = 1, \dots, J$  were utilized for the values of  $\gamma$  considered. For fixed  $M$ , the learned error estimates become more conservative with

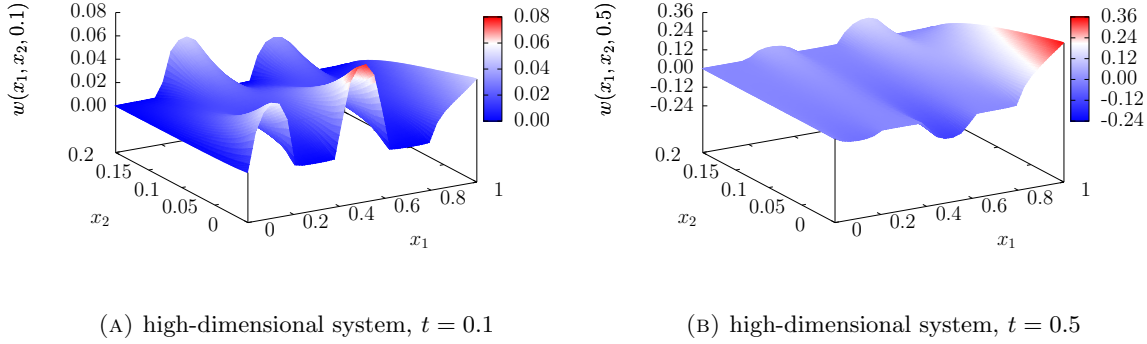


FIGURE 6. Convection-diffusion equation: Numerical approximation of the solution to (4.12) at times  $t = 0.1$  and  $t = 0.5$  for  $\mu = 0.5$  and control input  $\mathbf{G}^{\text{test}}$ .

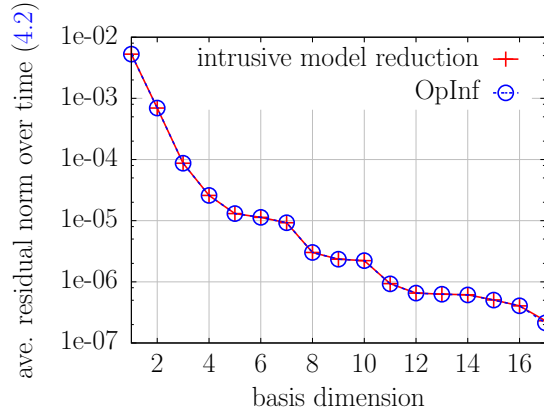


FIGURE 7. Convection-diffusion equation (Section 4.3.2): The graph indicates that the residual norm (3.8) needed for the *a posteriori* estimate (3.6) can be recovered under operator inference.

respect to the intrusive error estimate in favor of increased confidence in the estimate; cf. the definition of  $\Xi_l$  in Proposition 10.

Figure 9 plots the same quantities shown in Figure 8 but for the parameters  $\gamma = 1$  and  $M = 35, 100, 500$ , i.e.  $M$  is varied while  $\gamma$  is fixed. The lower bound probability values for each  $M$  are  $P^{\text{LB}} \approx 0.9212, 0.9999, 1$ . The sets of the  $M = 35, 100, 500$  realizations of  $\Theta^{(l)}$  for  $l = 1, \dots, J$  are nested. For this example, increasing  $M$  did lead only to slight changes in the learned error estimate. The influence of  $M$  is more difficult to gauge numerically since the maximum of  $\{\theta_i^{(l)}\}_{i=1}^M$  may not differ substantially as a function of  $M$ . The results in Figures 8 and 9 indicate that in this example, for a fixed value for  $P^{\text{LB}}$ , it is more favorable to choose a larger value of  $M$  and a smaller value of  $\gamma$  to obtain a tighter learned error estimate that is close to the error estimate from intrusive model reduction with a high confidence in the estimate. This is because increasing  $\gamma$  results in scaling  $\xi_l$  by  $\sqrt{\gamma}$  which then yields a more conservative error estimate (3.18) while only gaining a modest increase in  $P^{\text{LB}}$ . In contrast, increasing  $M$  (which requires additional queries to the full system) implies a faster convergence of  $P^{\text{LB}}$  to 1 without necessarily incurring a scaling of  $\xi_l$ . We therefore see that there is a trade off between a conservative bound and high  $P^{\text{LB}}$  for  $\gamma$  while there is a trade off between computational expense and high  $P^{\text{LB}}$  for  $M$ .

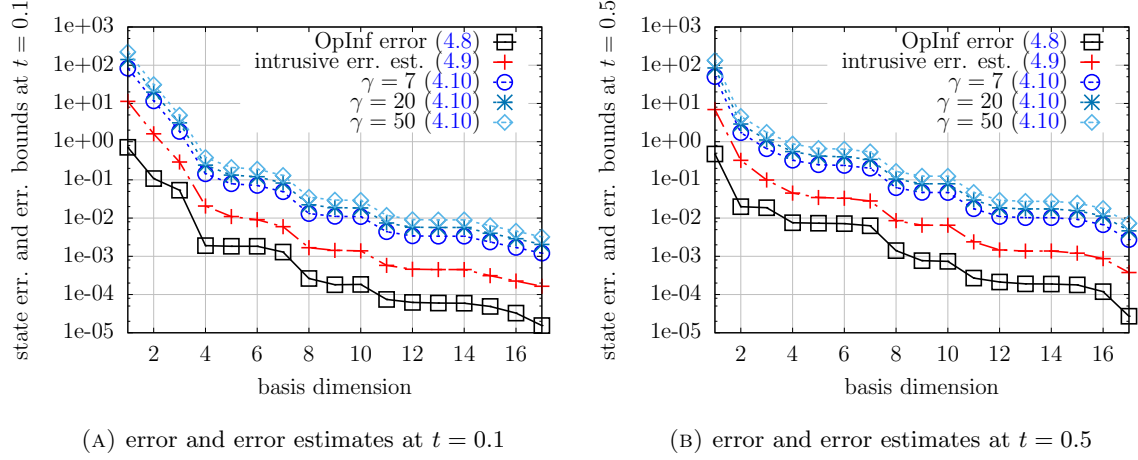


FIGURE 8. Convection-diffusion equation (Section 4.3.2). Increasing  $\gamma$  leads to a more conservative learned error estimate (3.18) for fixed  $M, J$  with respect to the intrusive error estimate. This simultaneously corresponds to a larger lower bound probability  $P^{\text{LB}}$ . The parameters used were  $M = 10$  and  $\gamma = 7, 20, 50$  for which  $P^{\text{LB}} \approx 0.75, 0.9985, 0.9999$ .

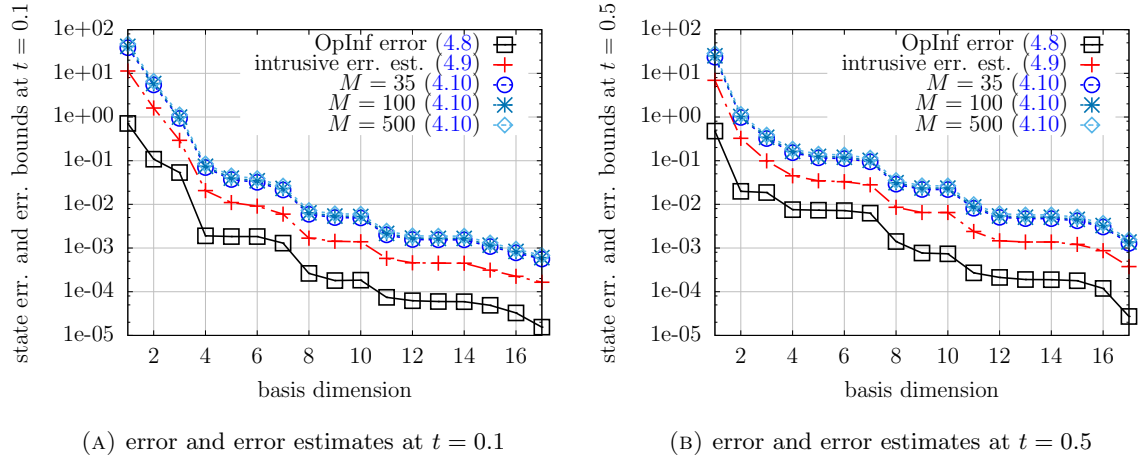


FIGURE 9. Convection-diffusion equation (Section 4.3.2). Increasing  $M$  only slightly changes the learned error estimate (3.18) for fixed  $\gamma, J$  in this example. The parameters used were  $\gamma = 1$  and  $M = 35, 100, 500$  for which  $P^{\text{LB}} \approx 0.92, 0.99, 1$ .

We now assess the variation in the realizations of the learned error estimator. The simulation is carried out for  $\gamma = 1, M = 35$ . We generated 50 sets of  $M = 35$  realizations of  $\Theta^{(l)}$  to produce 50 realizations of  $\Xi_t$  and of the learned error estimate (4.10). The mean (solid) of the 50 realizations of (4.10) for  $t = 0.1$  and  $t = 0.5$  are illustrated in the panels of Figure 10 together with their minimum and maximum values (vertical bars). For reference, the error estimate (4.8) under intrusive model reduction is also included. The plots show that the variation among samples of the learned error estimator is low.

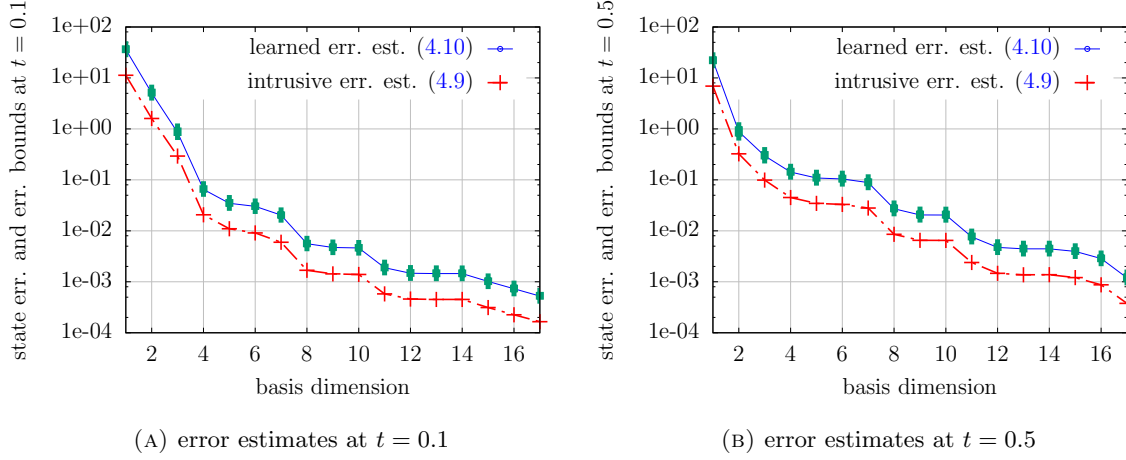


FIGURE 10. Convection-diffusion equation (Section 4.3.2). The mean, minimum, and maximum of (4.10) based on 50 samples of the learned error estimator are depicted together with the error estimate from intrusive model reduction. The parameters used were  $\gamma = 1, M = 35$ . There is a low variation among the samples of the learned error estimator.

#### 4.3.3. Results for sinusoidal control input

In this case, the diffusivity parameter is set to  $\mu = 1$ . The control input  $\mathbf{u}^{\text{basis}}(t)$  for constructing the basis  $\mathbf{V}_n$  consists of

$$\begin{aligned}
 u_1^{\text{basis}}(t) &= 5t \sin(\pi t) \\
 u_2^{\text{basis}}(t) &= e^{5t} \sin(2\pi t) \\
 u_3^{\text{basis}}(t) &= \sqrt{3 + t^2} \sin(3\pi t) \\
 u_4^{\text{basis}}(t) &= 50t^2 \sin(4\pi t) \\
 u_5^{\text{basis}}(t) &= e^{2t} \sin(5\pi t)
 \end{aligned} \tag{4.13}$$

while the components of the control input  $\mathbf{u}^{\text{test}}(t)$  for prediction were chosen as  $u_j^{\text{test}}(t) = \sin(j\pi t z_j), j = 1, \dots, 5$  where  $z_j$  is a realization of a  $N(0, 1)$  random variable.

Figure 11 summarizes the predictive capabilities of operator inference. The quantity (4.2) is plotted in Figure 11a wherein we see concordance between the intrusive and operator inference approaches. Observe that the errors in Figure 11a are larger than in Figure 7 since the basis and test control inputs are less similar in nature. Figure 11b contains graphs of the learned reduced model error (4.3) and the intrusive (4.4) and learned (probabilistic) (4.5) error estimates in which 1 sample of the learned error estimator was generated. The parameters for the learned error estimator were set to  $\gamma = 1, M = 40, J = 5 \times 10^4$  so that  $P^{\text{LB}}(\gamma, M, J) \approx 0.9883$ . The learned error estimate is close to the error estimate given by the intrusive approach.

#### 4.3.4. Results for bound on output error

We now study the efficiency of the learned error estimator for the state in constructing bounds for an output. We resume the setup in the previous subsection wherein the control input is sinusoidal. We consider two quantities of interest for this case, namely,  $y_k^{(j)} = \mathbf{C}^{(j)} \mathbf{w}_k$  for  $j \in \{1, 2\}$  with the control input  $\mathbf{G}^{\text{test}}$ . The matrices  $\mathbf{C}^{(1)}$  and  $\mathbf{C}^{(2)}$  are defined as follows: The first output is the average of the state components of  $\mathbf{w}_k$  at

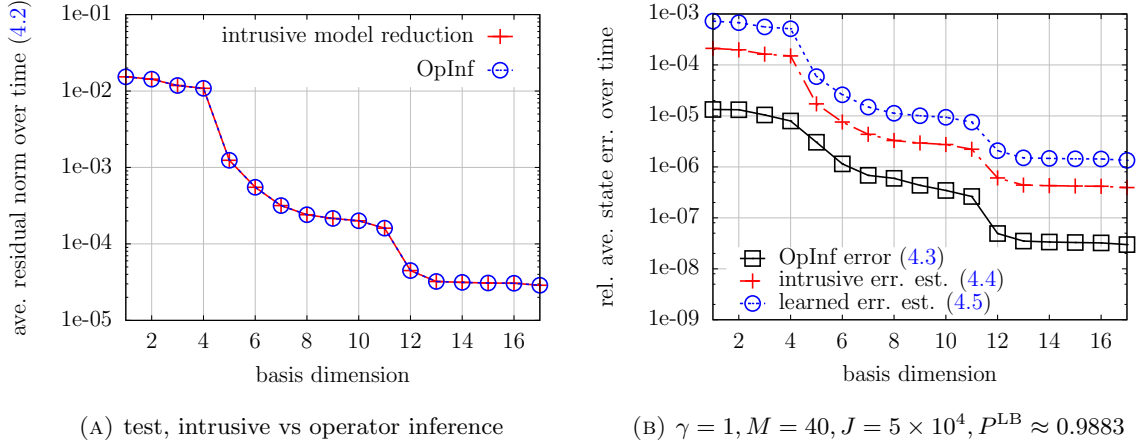


FIGURE 11. Convection-diffusion equation (Section 4.3.3). The panels show that operator inference recovers the residual norm necessary for deriving error estimates of the state. Furthermore, the learned state error estimate is only slightly higher than the error estimate provided by the intrusive method.

each time which is

$$y_k^{(1)} = \mathbf{C}^{(1)} \mathbf{w}_k \quad \text{where} \quad \mathbf{C}^{(1)} = \left[ \frac{1}{N}, \dots, \frac{1}{N} \right] \in \mathbb{R}^{1 \times N}. \quad (4.14)$$

The second output is the integral of the finite element approximation to  $w(\mathbf{x}, t)$  over the edge  $E_5$  at each time given by

$$y_k^{(2)} = \mathbf{C}^{(2)} \mathbf{w}_k \quad \text{where} \quad \mathbf{C}^{(2)} = \left[ \int_{E_5} \varphi_1 d\Gamma, \dots, \int_{E_5} \varphi_N d\Gamma \right] \in \mathbb{R}^{1 \times N}. \quad (4.15)$$

The output  $y_k$  and its bounds  $\tilde{y}_k \pm \Delta_k^y$  over time are displayed in Figures 12 (first output) and 13 (second output); cf. Remark 11. These quantities were sketched for  $n \in \{7, 12, 17\}$  basis dimensions in the first output and  $n \in \{5, 10, 15\}$  in the second output. The output bound  $\Delta_k^y$  is computed via the learned error estimator for the state  $\Delta_k^w$  with the same parameters above, i.e.  $\gamma = 1, M = 40, J = 5 \times 10^4$ . The panels show that increasing  $n$  yields a decrease in the output bound width  $2\Delta_k^y$  over time, i.e. the bounds are sharper with respect to the output value. This is supported by Figure 11 which demonstrates decrease of the learned state error estimate as a function of the basis dimension.

## 5. CONCLUSIONS

This work proposes a probabilistic *a posteriori* error estimator that is applicable with non-intrusive model reduction under certain assumptions. The key is that quantities that are necessary for error estimators developed for intrusive model reduction can be derived via least-squares regression from input and solution trajectories whereas other quantities that are necessary can be bounded in a probabilistic sense by sampling the high-dimensional system in a judicious and black-box way. The learned estimators can be used to rigorously upper bound the error of reduced models learned from data for initial conditions and inputs that are different from those used during the training phase (offline phase). Thus, the proposed approach establishes trust in decisions made from data by realizing the full workflow from data to reduced models to certified predictions.

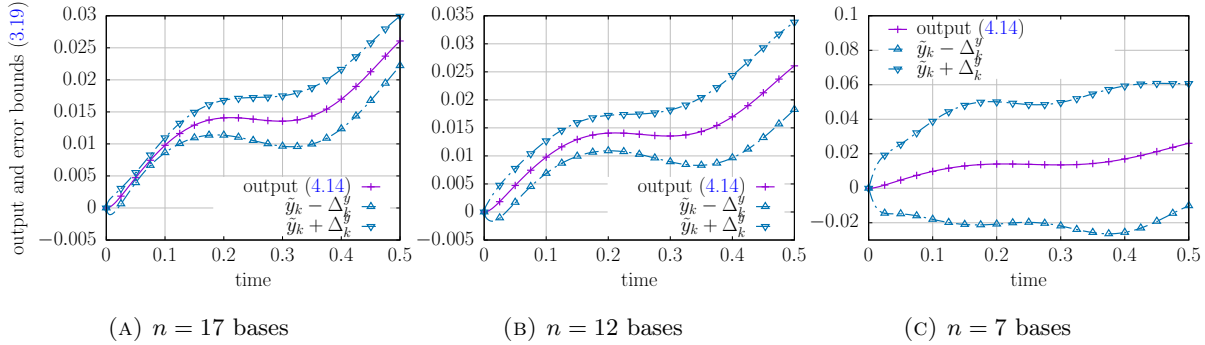


FIGURE 12. Convection-diffusion equation (Section 4.3.3). The panels show the predictive capability of the learned error estimator for the state error in constructing lower and upper bounds for the output (average of state components). The bounds correctly indicate that the errors of the predicted reduced-model outputs decreases if the basis dimension is increased.

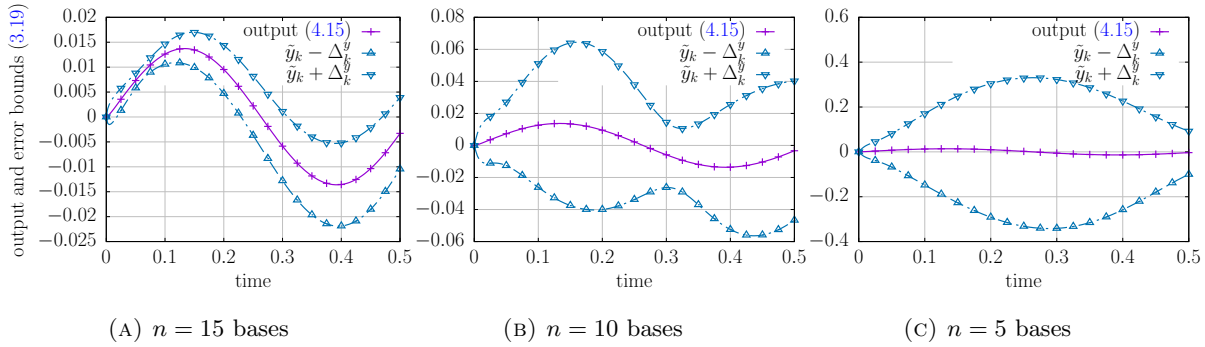


FIGURE 13. Convection-diffusion equation (Section 4.3.3). Similar behavior as described in Figure 12 is observed in these panels for the quantity of interest based on the integral over the Neumann boundary.

The framework introduced here for learning error estimators may be extended beyond linear time-invariant systems by exploiting *a posteriori* error estimators from intrusive model reduction provided that these can be derived for such systems. See [15] as an example for linear time-varying systems. While details remain future work, it seems the proposed approach is applicable to linear time-varying systems for which the time-varying operators can be decomposed as a linear combination of time-independent matrices with time-varying scalar coefficients. These matrices are then inferred from data on the squared residual norm if the time-dependent coefficients are known. For nonlinear systems, the method here should be adapted to specific nonlinear PDEs for which error analysis of the intrusive reduced model is possible. This will be the subject of future work.

## REFERENCES

- [1] 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.
- [2] 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.
- [3] Z. Bujanović and D. Kressner. Norm and trace estimation with random rank-one vectors. 2020. arXiv:2004.06433.
- [4] S. Chaturantabut and D. C. Sorensen. A state space error estimate for POD-DEIM nonlinear model reduction. *SIAM Journal on Numerical Analysis*, 50(1):46–63, 2012.

- [5] J. D. Dixon. Estimating extremal eigenvalues and condition numbers of matrices. *SIAM Journal on Numerical Analysis*, 20(4):812–814, 1983.
- [6] Z. Drmač, S. Gugercin, and C. Beattie. Vector fitting for matrix-valued rational approximation. *SIAM Journal on Scientific Computing*, 37(5):A2346–A2379, 2015.
- [7] J. L. Eftang, M. A. Grepl, and A. T. Patera. A posteriori error bounds for the empirical interpolation method. *Comptes Rendus Mathématique*, 348(9-10):575–579, 2010.
- [8] L. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.
- [9] L. Feng, A. C. Antoulas, and P. Benner. Some a posteriori error bounds for reduced-order modelling of (non-)parametrized linear systems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 51(6):2127–2158, 2017.
- [10] 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.
- [11] M. A. Grepl and A. T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.
- [12] 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.
- [13] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on Power Delivery*, 14(3):1052–1061, 1999.
- [14] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 2008.
- [15] B. Haasdonk and M. Ohlberger. Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition. *Mathematical and Computer Modelling of Dynamical Systems*, 17(2):145–161, 2011.
- [16] B. Haasdonk, M. Ohlberger, and G. Rozza. A reduced basis method for evolution schemes with parameter-dependent explicit operators. *Electron. Trans. Numer. Anal.*, 32:145–161, 2008.
- [17] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer International Publishing, 2016.
- [18] D. Huynh, G. Rozza, S. Sen, and A. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *Comptes Rendus Mathématique*, 345(8):473–478, 2007.
- [19] 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.
- [20] A. Janon, M. Nodet, and C. Prieur. Certified reduced-basis solutions of viscous Burgers equation parametrized by initial and boundary values. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(2):317–348, 2013.
- [21] 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.
- [22] B. Kramer and A. A. Gorodetsky. System identification via CUR-factored Hankel approximation. *SIAM Journal on Scientific Computing*, 40(2):A848–A866, 2018.
- [23] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic mode decomposition: Data-driven modeling of complex systems*. SIAM, 2016.
- [24] L. Ljung. *System identification: Theory for the user*. Prentice Hall information and system sciences series. Prentice Hall PTR, 2nd ed edition, 1999.
- [25] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012.
- [26] N.-C. Nguyen, G. Rozza, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers’ equation. *Calcolo*, 46(3):157–185, 2009.
- [27] B. Peherstorfer. Sampling low-dimensional markovian dynamics for preasymptotically recovering reduced models from data with operator inference. *SIAM Journal on Scientific Computing*, 42(5):A3489–A3515, 2020.
- [28] B. Peherstorfer and K. Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.
- [29] B. Peherstorfer and K. Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.
- [30] C. Prud’homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80, 2001.
- [31] E. Qian, B. Peherstorfer, and K. Willcox. Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406:132401, 2020.
- [32] 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.
- [33] C. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.
- [34] 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.

- [35] P. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.
- [36] 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.
- [37] A. Schmidt and B. Haasdonk. Reduced basis approximation of large scale parametric algebraic Riccati equations. *ESAIM: COCV*, 24(1):129–151, 2018.
- [38] K. Smetana, O. Zahm, and A. T. Patera. Randomized residual-based error estimators for parametrized equations. *SIAM Journal on Scientific Computing*, 41(2):A900–A926, 2019.
- [39] V. Thomee. *Galerkin Finite Element Methods for Parabolic Problems*. Springer Berlin Heidelberg, 2006.
- [40] 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.
- [41] K. Veroy and A. T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids*, 47(8-9):773–788, 2005.
- [42] K. Veroy, C. Prud’homme, D. Rovas, and A. T. Patera. A Posteriori Error Bounds for Reduced-Basis Approximation of Parametrized Noncoercive and Nonlinear Elliptic Partial Differential Equations. In *16th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences*. American Institute of Aeronautics and Astronautics, 2003.
- [43] K. Veroy, D. V. Rovas, and A. T. Patera. A posteriori error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations : “convex inverse” bound conditioners. *ESAIM: Control, Optimisation and Calculus of Variations*, 8:1007–1028, 2002.
- [44] D. Wirtz, D. C. Sorensen, and B. Haasdonk. A posteriori error estimation for DEIM reduced nonlinear dynamical systems. *SIAM Journal on Scientific Computing*, 36(2):A311–A338, 2014.
- [45] Y. Zhang, L. Feng, S. Li, and P. Benner. An efficient output error estimation for model order reduction of parametrized evolution equations. *SIAM Journal on Scientific Computing*, 37(6):B910–B936, 2015.