

RESEARCH ARTICLE

Open Access



# A posteriori error estimation for model order reduction of parametric systems

Lihong Feng , Sridhar Chellappa and Peter Benner

\*Correspondence:  
feng@mpi-magdeburg.mpg.de

<sup>1</sup>Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany

## Abstract

This survey discusses a posteriori error estimation for model order reduction of parametric systems, including linear and nonlinear, time-dependent and steady systems. We focus on introducing the error estimators we have proposed in the past few years and comparing them with the most related error estimators from the literature. For a clearer comparison, we have translated some existing error bounds proposed in function spaces into the vector space  $\mathbb{C}^n$  and provide the corresponding proofs in  $\mathbb{C}^n$ . Some new insights into our proposed error estimators are explored. Moreover, we review our newly proposed error estimator for nonlinear time-evolution systems, which is applicable to reduced-order models solved by arbitrary time-integration solvers. Our recent work on multi-fidelity error estimation is also briefly discussed. Finally, we derive a new *inf-sup*-constant-free output error estimator for nonlinear time-evolution systems. Numerical results for three examples show the robustness of the new error estimator.

**Keywords:** A posteriori error estimation, Parametric systems, Model order reduction

## Introduction

For every model order reduction (MOR) method or algorithm to be eventually used in real applications, accuracy and efficiency of the method play key roles. While many MOR methods are numerically shown efficient, not all of them are guaranteed to be reliable. In other words, not all numerically demonstrated efficient MOR methods are associated with computable error estimators, let alone fast-to-compute error estimators. This work reviews a posteriori error estimators for projection-based MOR of parametric systems. Many projection-based MOR methods for parametric systems [1] have been proposed, for example, the multi-moment-matching methods [2, 3], methods based on (transfer function, projection matrix, or manifold) interpolation [4–11], the proper orthogonal decomposition (POD) methods [12–14], as well as the reduced basis methods [15–18]. We name those MOR methods for parametric systems pMOR methods. However, error estimation for some of the pMOR methods are not yet widely discussed, for example, error estimation for interpolation-based pMOR methods. While some a posteriori error bounds [15, 17–28] are proposed for reduced-order models (ROMs) obtained from the reduced basis method, most of them are derived using the weak form of the finite element method (FEM). In contrast, we proposed some a posteriori error estimators [29–35] which are

independent of the numerical discretization method. The error estimators are expressed with the already discretized matrices and (nonlinear) vectors. Many of the existing error bounds or error estimators are applicable to ROMs constructed via *global* projection matrices, regardless which pMOR method is used for the ROM construction. For the reduced basis method, the projection matrix and the ROM are usually constructed via a greedy process. Multi-fidelity error estimation is recently proposed in [36] to accelerate the greedy algorithm for constructing the projection matrix.

We further discuss a newly proposed error estimator [35] which is independent of the numerical time-integration scheme and therefore is able to estimate the error of the ROM solved with any time integrator. This is desired in many engineering applications, where often commercial software is used to solve the original dynamical systems. Then it is also desirable that the error estimator can be applied to measure the ROM error while the ROM is solved with the same software. However, existing error estimators (bounds) cannot achieve this, since they usually require a pre-defined non-adaptive time-integration scheme. This limits the wide use of the error estimators (bounds).

Finally, we propose an *inf-sup*-constant-free output error estimator for nonlinear time-evolution systems, which avoids the computation of the smallest singular value  $\sigma_{\min}(\mu)$  of a large matrix at each queried sample of the parameter. This not only improves the accuracy of the error estimator for problems with  $\sigma_{\min}(\mu)$  close to zero, but also reduces a large amount of computations, as computing the singular value needs computational complexity of at least  $O(N)$  for each parameter sample, where  $N$  is usually large.

Most of the error estimation methods reviewed in this work are based either on the residual of the ROM approximation or on both the residual and a dual system. Such techniques of using the residual of an approximate solution and a dual system, can be traced back to error estimation for FEM approximations, see, e.g., [37].

For clarity, we summarize the new contributions of this survey, which cannot be found in the referenced articles:

- Theorem 2. It transforms the error bound presented in function space in [19, 27] into an error bound in the vector space  $\mathbb{C}^n$ . New proofs are provided in [Appendix](#).
- Theorem 4. It derives an error bound with *quadratic* decay in  $\mathbb{C}^n$ .
- Theorem 5 and its proof. It uses a slightly different dual system (25) and a slightly different auxiliary output  $\tilde{y}^k(\mu)$  to derive the same output error bound as in [29, 30]. Please see Remark 8 for the detailed differences.
- Theorem 7. It quantifies the state error estimator proposed in [38] with computable upper and lower bounds.
- “[Inf-sup-constant-free error estimator for time-evolution systems](#)” section. It proposes a new *inf-sup*-constant-free output error estimator for parametric time-evolution systems.

In the next sections, we discuss error estimation for both time-evolution systems and steady systems. “[Problem formulation](#)” section formulates the problems considered in this work, including the original large-scale models and the corresponding ROMs. We first review rigorous error bounds for both systems and provide some new proofs in “[Rigorous a posteriori error bounds](#)” section. Then in “[A posteriori error estimators](#)” section, we review error estimators which are not rigorous anymore, but decay faster than the error bounds. The error estimators usually also have less computational complexity than

the error bounds. “[Error estimator for ROMs solved with any black-box time-integration solver](#)” section reviews the newly proposed error estimator that is applicable to black-box solvers. The recently proposed multi-fidelity error estimation for large and complex systems is reviewed in “[Multi-fidelity error estimation](#)” section. It is shown that for some complex systems, the greedy process of constructing the reduced basis can be largely accelerated with multi-fidelity error estimation. “[Inf-sup-constant-free error estimator for time-evolution systems](#)” section proposes a new *inf-sup*-constant-free output error estimator for nonlinear time-evolution systems and presents numerical results. We conclude this survey in “[Conclusion](#)” section. This review is not exhaustive, but only contains our contribution to this topic and the most related ones from the literature. Other error estimators, in particular all error estimators for different types of systems, e.g., error estimation for ROMs of second-order non-parametric systems [39, 40], are not discussed. The proper generalized decomposition (PGD) method [41] known as a non-projection-based MOR method, and the corresponding error estimation [42–44], are not considered in this survey either. The list of abbreviations is provided as below:

- MOR: model order reduction
- POD: proper orthogonal decomposition
- FEM: finite element method
- PDEs: partial differential equations
- FOM: full-order model
- ROM: reduced-order model
- PGD: proper generalized decomposition
- IMEX: implicit–explicit
- RBF: radial basis function
- FFNN: feed forward neural network
- TC1: the first test case
- TC2: the second test case

### Problem formulation

Consider the following parametric time-evolution system of differential algebraic equations (DAEs):

$$\begin{aligned} \frac{d}{dt} \mathbf{E}(\mu) \mathbf{x}(t, \mu) &= \mathbf{A}(\mu) \mathbf{x}(t, \mu) + \mathbf{f}(\mathbf{x}(t, \mu), \mu) + \mathbf{B}(\mu) \mathbf{u}(t), & \mathbf{x}(0, \mu) &= \mathbf{x}_0(\mu), \\ \mathbf{y}(t, \mu) &= \mathbf{C}(\mu) \mathbf{x}(t, \mu), \end{aligned} \quad (1)$$

where  $t \in [0, T]$  and  $\mu \in \mathcal{P} \subset \mathbb{R}^p$ ,  $\mathcal{P}$  is the parameter domain.  $\mathbf{x}(\mu) \in \mathbb{R}^N$  is the state vector of the system and  $\mathbf{E}(\mu)$ ,  $\mathbf{A}(\mu) \in \mathbb{R}^{N \times N}$ ,  $\mathbf{B}(\mu) \in \mathbb{R}^{N \times n_I}$ ,  $\mathbf{C}(\mu) \in \mathbb{R}^{n_O \times N}$ ,  $\forall \mu \in \mathcal{P}$ , are the system matrices,  $\mathbf{f} : \mathbb{R}^N \times \mathcal{P} \mapsto \mathbb{R}^N$  is the nonlinear system operator and  $\mathbf{u} : t \mapsto \mathbb{R}^{n_I}$  is the external input signal. Such systems often arise from discretizing partial differential equations (PDEs) using numerical discretization schemes, or following some physical laws. System (1) is called the full-order model (FOM) when we discuss MOR. The number of equations  $N$  in (1) is often very large to ensure high-resolution of the underlying physical process. Numerically solving the FOM is expensive, especially for multi-query tasks, where the FOM has to be solved at many instances of  $\mu$ . When  $n_I > 1$  and  $n_O > 1$ , the system has multiple inputs and multiple outputs. Such problems are common in electrical or electromagnetic simulation [36]. When we consider error estimation, we usually first

assume  $n_I = n_o = 1$ , then the obtained error estimation is extended to the more general case  $n_I > 1$  and  $n_O > 1$ . The extension is straightforward if the error is measured using the matrix-max norm [31, 33, 38]. Therefore, if not mentioned explicitly, we consider the case  $n_I = n_o = 1$  such that (1) can be written as

$$\begin{aligned} \frac{d}{dt} \mathbf{E}(\mu) \mathbf{x}(t, \mu) &= \mathbf{A}(\mu) \mathbf{x}(t, \mu) + \mathbf{f}(\mathbf{x}(t, \mu), \mu) + \mathbf{b}(\mu) u(t), & \mathbf{x}(0, \mu) &= \mathbf{x}_0(\mu), \\ y(t, \mu) &= \mathbf{c}(\mu) \mathbf{x}(t, \mu). \end{aligned} \quad (2)$$

Here, the input signal  $u(t)$  and the output response  $y(t, \mu)$  become scalar-valued functions of time and  $\mu$ , respectively. Consequently, the system matrices  $\mathbf{b} \in \mathbb{R}^N$  and  $\mathbf{c} \in \mathbb{R}^N$  are now vectors. All other quantities remain the same as in (1). We will briefly mention the extension to  $n_I > 1$  and  $n_O > 1$  at proper places. Projection-based MOR techniques obtain a ROM for (2) in the following form:

$$\begin{aligned} \frac{d}{dt} \hat{\mathbf{E}}(\mu) \hat{\mathbf{x}}(t, \mu) &= \hat{\mathbf{A}}(\mu) \hat{\mathbf{x}}(t, \mu) + \hat{\mathbf{f}}(\mathbf{V} \hat{\mathbf{x}}(t, \mu), \mu) + \hat{\mathbf{b}}(\mu) u(t), & \hat{\mathbf{x}}(0, \mu) &= \mathbf{W}^T \mathbf{x}_0(\mu), \\ \hat{y}(t, \mu) &= \hat{\mathbf{c}}(\mu) \hat{\mathbf{x}}(t, \mu), \end{aligned} \quad (3)$$

where  $\mathbf{V} \in \mathbb{R}^{N \times n}$  is the parameter  $\mu$ -independent projection matrix, whose columns are the reduced basis vectors.

$$\hat{\mathbf{E}}(\mu) = \mathbf{W}^T \mathbf{E}(\mu) \mathbf{V}, \hat{\mathbf{A}}(\mu) = \mathbf{W}^T \mathbf{A}(\mu) \mathbf{V}, \hat{\mathbf{b}}(\mu) = \mathbf{W}^T \mathbf{b}(\mu), \hat{\mathbf{c}}(\mu) = \mathbf{c}(\mu) \mathbf{V}$$

are the reduced system matrices.  $\hat{\mathbf{f}}(\cdot, \cdot) = \mathbf{V}^T \mathbf{f}(\cdot, \cdot)$  is the reduced nonlinear vector. The number of equations  $n$  in (3) should be much smaller than  $N$  in (2), i.e.,  $n \ll N$ , so that the ROM can be readily used for repeated simulations. When  $\mathbf{V} = \mathbf{W}$ , it is referred to as Galerkin projection. We focus on Galerkin projection, though the error estimators discussed in this work straightforwardly apply to Petrov–Galerkin projection, too.

For steady problems, the parametric system is time-independent,

$$\begin{aligned} \mathbf{f}(\mathbf{x}(\mu), \mu) &= \mathbf{b}(\mu), \\ \mathbf{y}(\mu) &= \mathbf{c}(\mu) \mathbf{x}(\mu), \end{aligned} \quad (4)$$

where  $\mathbf{x}(\mu) \in \mathbb{R}^N$ , and  $\mathbf{f} : \mathbb{R}^N \times \mathcal{C}^p \mapsto \mathbb{R}^N$  is the nonlinear system operator. Projection-based pMOR obtains a steady parametric ROM as below,

$$\begin{aligned} \hat{\mathbf{f}}(\mathbf{V} \hat{\mathbf{x}}(\mu), \mu) &= \hat{\mathbf{b}}(\mu), \\ \hat{y}(\mu) &= \hat{\mathbf{c}}(\mu) \hat{\mathbf{x}}(\mu), \end{aligned} \quad (5)$$

where  $\hat{\mathbf{f}}(\cdot, \cdot) = \mathbf{V}^T \mathbf{f}(\cdot, \cdot)$ . When the system is linear, the steady system then becomes

$$\begin{aligned} \mathbf{M}(\mu) \mathbf{x}(\mu) &= \mathbf{b}(\mu), \\ \mathbf{y}(\mu) &= \mathbf{c}(\mu) \mathbf{x}(\mu), \end{aligned} \quad (6)$$

where  $\mathbf{M}(\mu) \in \mathbb{R}^{N \times N}$ ,  $\forall \mu \in \mathcal{P}$ . The corresponding steady parametric ROM is

$$\begin{aligned} \hat{\mathbf{M}}(\mu) \hat{\mathbf{x}}(\mu) &= \hat{\mathbf{b}}(\mu), \\ \hat{y}(\mu) &= \hat{\mathbf{c}}(\mu) \hat{\mathbf{x}}(\mu), \end{aligned} \quad (7)$$

where  $\hat{\mathbf{M}}(\mu) = \mathbf{V}^T \mathbf{M}(\mu) \mathbf{V}$ .

In the following, we mainly discuss error estimation on the solutions obtained from the ROMs (3) and (7). The norm  $\|\cdot\|$  refers to the vector 2-norm or matrix spectral norm all through the article. The ROM in (3) or (7) is constructed using a global reduced basis

V. The error estimation methods reviewed in this work could be applied to measure the error of the ROM obtained using a global reduced basis, irrespective of the method used to construct the reduced basis. In this sense, the error estimation methods are generic and could be applicable to multi-moment-matching methods, POD methods, reduced basis methods and some interpolation-based methods.

*Remark 1* We point out that if the FOMs (1), (2), (4), (6) are obtained from numerical discretization of PDEs, the error estimation discussed in this work and those in most of the referenced works in the introduction, do not involve the discretization error. This is the case for most of the reduced basis method in the literature. As the spatial discretization and the model reduction are mostly two separate steps, this is common practice. We note that in case of knowledge of the discretization error, e.g., in adaptive FEM, one can adapt the model reduction error tolerance to this error so that model reduction does not contribute further to the magnitude of the approximation error, by, e.g., choosing the model reduction tolerance to be 1–2 orders of magnitude lower than the discretization error. This is common practice, but beyond the scope of this paper. For works on error estimation including both the discretization error and the ROM error, please refer to [45–47]. The error estimation reviewed in this work could be combined with the discretization error estimator [37] to realize adaptivity of the mesh size by checking the two estimated errors respectively, during a joint greedy process for both spatial discretization and MOR. Moreover, there are FOMs that are not derived by numerical discretization of PDEs, rather from some physical laws, for example, the modified nodal analysis (MNA) in circuit simulation directly results in systems of DAEs. For such systems, we consider the solutions to the FOMs as the *exact* solutions.

### Rigorous a posteriori error bounds

This section reviews rigorous a posteriori error bounds for estimating the ROM error, which are upper bounds of the true errors and therefore are rigorous. For time-evolution systems, most of the approaches estimate the error at discrete time instances. There are error bounds for state error and error bounds for output error. Output error bounds usually need a dual system to achieve faster decay. We review the error bound for time-evolution systems and steady systems in separate subsections.

#### Error bounds for time-evolution systems

The standard error estimation approaches proposed for the reduced basis method are residual-based [17–19, 23, 24]. In order to derive the error bound, knowledge of the temporal discretization scheme used to integrate the FOM and the ROM is assumed, e.g., using implicit Euler, Crank-Nicolson method, or an implicit–explicit (IMEX) method. Computing the error bound involves determining the residual vector  $\mathbf{r}(\mu) \in \mathbb{R}^N$  at each time instance. Some goal-oriented output error estimation approaches also require the residual of a dual system.

Suppose (2) is discretized in time using a first-order IMEX scheme [48]. The linear part is discretized implicitly, while the nonlinear vector  $\mathbf{f}(\mathbf{x}(\mathbf{t}, \mu), \mu)$  is evaluated explicitly. The resulting discretized system is

$$\begin{aligned} \mathbf{A}_t(\mu)\mathbf{x}^k(\mu) &= \mathbf{E}(\mu)\mathbf{x}^{k-1}(\mu) + \delta t(\mathbf{f}(\mathbf{x}^{k-1}(\mu), \mu) + \mathbf{b}(\mu)u^k), \\ \mathbf{y}^k(\mu) &= \mathbf{c}(\mu)\mathbf{x}^k(\mu), \quad k = 0, \dots, K \end{aligned} \quad (8)$$

with  $\mathbf{A}_t(\mu) := \mathbf{E}(\mu) - \delta t \mathbf{A}(\mu)$ . Here,  $\delta t$  is the temporal discretization step. Error estimation methods discussed in this work may also apply to time-varying  $\delta t$ . For simplicity, we use  $\delta t$  to represent the time-varying case, too. The ROM (3) can be discretized in the same way as

$$\begin{aligned}\hat{\mathbf{A}}_t(\mu) \hat{\mathbf{x}}^k(\mu) &= \hat{\mathbf{E}}(\mu) \hat{\mathbf{x}}^{k-1}(\mu) + \delta t (\hat{\mathbf{f}}(\mathbf{V} \hat{\mathbf{x}}^{k-1}(\mu), \mu) + \hat{\mathbf{b}}(\mu) u^k), \\ \hat{\mathbf{y}}^k(\mu) &= \hat{\mathbf{c}} \hat{\mathbf{x}}^k(\mu), \quad k = 0, \dots, K,\end{aligned}\quad (9)$$

where  $\hat{\mathbf{A}}_t(\mu) := \hat{\mathbf{E}}(\mu) - \delta t \hat{\mathbf{A}}(\mu)$ . The residual from the ROM approximation is computed by substituting the approximate state vector  $\hat{\mathbf{x}}^k(\mu) := \mathbf{V} \hat{\mathbf{x}}^k(\mu)$  into (8). The resulting residual at the  $k$ -th time step,  $\mathbf{r}^k(\mu)$  is

$$\mathbf{r}^k(\mu) := \mathbf{E}(\mu) \hat{\mathbf{x}}^{k-1}(\mu) + \delta t (\mathbf{f}(\hat{\mathbf{x}}^{k-1}(\mu), \mu) + \mathbf{b}(\mu) u^k) - \mathbf{A}_t(\mu) \hat{\mathbf{x}}^k(\mu). \quad (10)$$

The nonlinear part of the ROM (9) is not yet hyperreduced. When hyperreduction [49, 50], e.g., discrete empirical interpolation (DEIM), is applied to (9), we get the ROM in the form,

$$\begin{aligned}\hat{\mathbf{A}}_t(\mu) \hat{\mathbf{x}}^k(\mu) &= \hat{\mathbf{E}}(\mu) \hat{\mathbf{x}}^{k-1}(\mu) + \delta t (\mathbf{V}^T \mathcal{I}_f^k(\mu) + \hat{\mathbf{b}}(\mu) u^k), \\ \hat{\mathbf{y}}^k(\mu) &= \hat{\mathbf{c}} \hat{\mathbf{x}}^k(\mu), \quad k = 0, \dots, K,\end{aligned}\quad (11)$$

where  $\mathcal{I}_f^k(\mu)$  is an approximation of  $\mathbf{f}(\mathbf{V} \hat{\mathbf{x}}^k(\mu), \mu)$ . It is clear that in order to obtain the residual  $\mathbf{r}^k(\mu)$ , the temporal discretization scheme for the ROM should be the same as that for the FOM so that  $\hat{\mathbf{x}}^k(\mu)$  in (10) and  $\mathbf{x}^k(\mu)$  in (8) correspond to the same time instance  $t^k$ .

### State error bound

An *a posteriori* error bound  $\Delta(\mu)$  for the approximation error  $\mathbf{e}^k(\mu) := \mathbf{x}^k(\mu) - \hat{\mathbf{x}}^k(\mu)$  can be computed based on the residual as below.

**Theorem 1** (Residual-based error bound) *Suppose that the nonlinear quantity  $\mathbf{f}(\mathbf{x}(t, \mu), \mu)$  is Lipschitz continuous in the first argument for all  $\mu$  such that there exists a constant  $L_f$  for which*

$$\|\mathbf{f}(\mathbf{x}(t, \mu), \mu) - \mathbf{f}(\tilde{\mathbf{x}}(t, \mu), \mu)\| \leq L_f \|\mathbf{x}(t, \mu) - \tilde{\mathbf{x}}(t, \mu)\|, \quad \forall t \geq 0, \mu \in \mathcal{P}.$$

*Further assume that for any parameter  $\mu$  the projection error at the first time step is  $\|\mathbf{e}^0(\mu)\| = \|\mathbf{x}^0(\mu) - \hat{\mathbf{x}}^0(\mu)\| = \|\mathbf{x}^0(\mu) - \mathbf{V} \mathbf{V}^T \mathbf{x}^0(\mu)\|$ , and  $\mathbf{A}_t(\mu)$  is invertible,  $\forall \mu \in \mathcal{P}$ . The error of the approximate state vector  $\hat{\mathbf{x}}$  at the  $k$ -th time step,  $\|\mathbf{e}^k(\mu)\| = \|\mathbf{x}^k(\mu) - \hat{\mathbf{x}}^k(\mu)\|$  is given by:*

$$\|\mathbf{e}^k(\mu)\| \leq \Delta^k(\mu) := \xi(\mu)^k \|\mathbf{e}^0(\mu)\|_2 + \sum_{i=1}^k \zeta(\mu) \cdot \xi(\mu)^{k-i} \cdot \|\mathbf{r}^i(\mu)\|, \quad (12)$$

where  $\zeta(\mu) := \|\mathbf{A}_t(\mu)^{-1}\|$  and  $\xi(\mu) := \|\mathbf{A}_t(\mu)^{-1} \mathbf{E}(\mu)\| + \delta t L_f \|\mathbf{A}_t(\mu)^{-1}\|$ .

*Proof* A proof for the above theorem can be found in [35]. □

**Remark 2** When the system has multiple inputs, then the state error bound corresponding to each column of  $\mathbf{B}(\mu)$  can be obtained from Theorem 1. The final state error is taken as the maximum over all the column-wise derived state error bounds.

In [16, 18, 19, 51], similar state error bounds using the residual  $\mathbf{r}^k(\mu)$  are derived, where only linear systems are considered in [16, 18]. The error bound proposed in [19] for nonlinear systems includes the error of hyperreduction for the nonlinear function  $\mathbf{f}(\mathbf{x}(t, \mu), \mu)$ , such that there is an additional term in the error bound. The error bound [19] is expressed in function space, and is not straightforward to be translated into the vector space  $\mathbb{C}^n$  as we consider here. An error bound in the vector space  $\mathbb{C}^n$  by considering hyperreduction is provided in [29, 52]. A state error bound based on implicit temporal discretization scheme is proposed in [51], where the hyperreduction error is also considered. In summary, all the error bounds discussed in [16, 18, 19, 29, 51, 52] and the one in Theorem 1 involve summing up the residual  $\mathbf{r}^k(\mu)$  at discrete time steps.

*Remark 3* In [24], state error bound for the linear version of the time-continuous ROM in (3) is derived, i.e., the nonlinear functions in (2) and in (3) are both assumed to be zero. The error bound is also a function of continuous time and continuous parameter. The sum of the residual  $\|\mathbf{r}^i(\mu)\|$  over discrete time steps becomes the integral of the residual over the time interval  $[0, T]$ .

#### Output error bound

A straightforward output error bound for the output error

$$\mathbf{e}_o^k(\mu) := \mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu) = \mathbf{c}\mathbf{x}^k(\mu) - \mathbf{c}\hat{\mathbf{x}}^k(\mu)$$

can be derived from (12) of Theorem 1 by noticing that  $\|\mathbf{e}_o^k(\mu)\| \leq \|\mathbf{c}\| \|\mathbf{e}^k(\mu)\|$  [24]. Finally, we have

$$\|\mathbf{e}_o^k(\mu)\| \leq \|\mathbf{c}\| (\xi(\mu))^k \|\mathbf{e}^0(\mu)\|_2 + \sum_{i=1}^k \zeta(\mu) \cdot \xi(\mu)^{k-i} \cdot \|\mathbf{r}^i(\mu)\|. \quad (13)$$

The above output error bound is nevertheless rather conservative, especially when  $\|\mathbf{c}\|$  is large. Moreover, the error bound depends only on  $\|\mathbf{r}^i(\mu)\|$ , i.e., the primal residual, leading to a linear decay.

Primal-dual-based output error bounds are obtained in [19, 24, 26, 27] for linear time-evolution systems, so that the resulting error bounds possess quadratic decay w.r.t. both the primal residual and the dual residual. The output error bound in [19, 26, 27] is described in function space based on the weak form of the original PDEs. To be consistent with the system (9) using matrices and vectors, we transform the error bound in [19] into the vector space  $\mathbb{C}^n$  for the ROM in (9). Theorem 2 shows the interpreted error bound. The assumptions of the theorem correspond to those assumptions in [19] in function space. No additional or stronger limitations are assumed in Theorem 2. Although the proof for the theorem can be done by more or less using the idea of the proof in [19], it is very different and is therefore provided in this work. Note that the proof in [19] is divided into several lemmas, thus consists of a sequence of proofs.

**Theorem 2** (Output error bound for linear systems) *Given a linear FOM in (8), where  $\mathbf{f}(\mathbf{x}^k(\mu), \mu) = 0$ , consider the output error of its ROM in (9) where  $\hat{\mathbf{f}}(\mathbf{V}\hat{\mathbf{x}}^k(\mu), \mu) = 0$ . Assume that there is no error at the initial condition,  $\mathbf{e}^0(\mu) = \mathbf{x}^0(\mu) - \hat{\mathbf{x}}^0(\mu) = 0$ , and both the matrices  $-\mathbf{A}(\mu)$  and  $\mathbf{E}(\mu)$  are symmetric positive definite. An error bound for the error*

of a corrected output of the ROM is

$$|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}_c^k(\mu)| \leq \Gamma(\mu) \cdot \left( \sum_{i=1}^k \|\mathbf{r}^i(\mu)\|^2 \right)^{1/2} \cdot \left( \sum_{j=0}^{k-1} \|\mathbf{r}_{du}^{K-k+j}(\mu)\|^2 + \delta_{du}^K(\mu) \right)^{1/2}, \quad (14)$$

where  $\hat{\mathbf{y}}_c^k(\mu) = \hat{\mathbf{y}}^k(\mu) + \sum_{j=0}^{k-1} (\hat{\mathbf{x}}_{du}^{K-k+j}(\mu))^T \mathbf{r}^j(\mu)$  is a corrected output of  $\hat{\mathbf{y}}^k(\mu)$ . Here,  $\|\mathbf{r}_{du}^j(\mu)\|$  denotes the residual at the  $j$ -th time instance induced by the ROM of the dual system

$$\begin{aligned} \mathbf{A}_t(\mu)^T \mathbf{x}_{du}^k(\mu) &= \mathbf{E}(\mu)^T \mathbf{x}_{du}^{k+1}(\mu), \\ \mathbf{E}(\mu)^T \mathbf{x}_{du}^K(\mu) &= \mathbf{c}(\mu)^T, \quad k = K - 1, \dots, 0, \end{aligned} \quad (15)$$

and  $\Gamma(\mu)$  is a parameter-dependent scalar.  $\delta_{du}^K(\mu)$  is a scaled upper bound for the dual ROM state error  $\|\mathbf{x}_{du}^K(\mu) - \mathbf{V}_{du} \hat{\mathbf{x}}_{du}^K(\mu)\|^2$  at the final time step  $t = t_K$ . The dual ROM is defined as

$$\begin{aligned} \hat{\mathbf{A}}_{du}(\mu) \hat{\mathbf{x}}_{du}^k(\mu) &= \hat{\mathbf{E}}_{du}(\mu) \hat{\mathbf{x}}_{du}^{k+1}(\mu), \\ \hat{\mathbf{E}}_{du}(\mu) \hat{\mathbf{x}}_{du}^K(\mu) &= \hat{\mathbf{c}}_{du}(\mu), \quad k = K - 1, \dots, 0, \end{aligned} \quad (16)$$

where  $\hat{\mathbf{A}}_{du}(\mu) = \mathbf{V}_{du}^T \mathbf{A}_t(\mu)^T \mathbf{V}_{du}$ ,  $\hat{\mathbf{E}}_{du}(\mu) = \mathbf{V}_{du}^T \mathbf{E}(\mu)^T \mathbf{V}_{du}$ ,  $\hat{\mathbf{c}}_{du} = \mathbf{V}_{du}^T \mathbf{c}(\mu)^T$ .

*Proof* See [Appendix](#). □

*Remark 4* For multiple input and multiple output systems, an output error bound corresponding to each column of  $\mathbf{B}(\mu)$  and each row of  $\mathbf{C}(\mu)$  can be derived from [Theorem 2](#). Then the final output error bound is taken as the maximum over all column-row-wise derived output error bounds. Please refer to a more detailed derivation for steady systems in the next “[Error bounds for steady systems](#)” section.

*Remark 5* In [\[24\]](#), a similar primal-dual based output error bound is obtained for the time-continuous ROM in [\(3\)](#). The output error bound estimates the output error at the final time  $T$ . The sums of the primal residual  $\mathbf{r}(\mu)^k$  and the dual-residual  $\mathbf{r}_{du}^k$  over time instances in [\(14\)](#) then become two integrals integrating the time variable from 0 to  $T$ . The initial approximation error  $\|\mathbf{e}^0(\mu)\|_2$  was assumed to be zero in [\[19\]](#), while it exists in the error bound in [\[24\]](#).  $\Gamma(\mu)$  is also differently defined in [\[24\]](#). In contrast to [\[24\]](#) where the error estimation is derived in the vector space  $\mathbb{C}^n$ , in [\[26\]](#) a time-continuous output error bound is derived in the function space based on the weak form of the original PDEs. The error bounds proposed in [\[15, 24, 27\]](#) are also reviewed in the survey paper [\[28\]](#) on the reduced basis method.

*Remark 6* [Theorem 2](#) is restricted in the sense that both  $\mathbf{E}(\mu)$  and  $-\mathbf{A}(\mu)$  are assumed to be symmetric positive definite. Our proposed error estimators to be discussed in “[A posteriori error estimators](#)” section do not need this assumption.

The primal-dual based output error bound in [\(14\)](#) has a *quadratic* behavior in the sense that it is the multiplication of the primal residuals with the dual-residuals. Therefore, it is expected that the error bound decays faster than the primal-only output error bound



in (13). Note that all the above reviewed error bounds estimate the error by accumulating the residuals over time.

### Error bounds for steady systems

In this subsection, we discuss error bounds for steady systems as in (6). Analogous to the time-evolution systems, the error bounds for both the state error and the output error also rely on the spectral norm of  $\|\mathbf{M}^{-1}(\mu)\|$ , i.e., the smallest singular value of the matrix  $\mathbf{M}(\mu)$  for any given  $\mu$ .

### State error bound

The state error bound for the state error  $\mathbf{e}(\mu) = \mathbf{x}(\mu) - \mathbf{V}\hat{\mathbf{x}}(\mu)$  can be easily derived by noticing that

$$\begin{aligned}\mathbf{M}(\mu)\mathbf{e}(\mu) &= \mathbf{M}(\mu)\mathbf{x}(\mu) - \mathbf{M}(\mu)\mathbf{V}\hat{\mathbf{x}}(\mu), \\ &= \underbrace{\mathbf{b}(\mu) - \mathbf{M}(\mu)\mathbf{V}\hat{\mathbf{x}}(\mu)}_{=: \mathbf{r}(\mu)}.\end{aligned}$$

Finally,

$$\|\mathbf{e}(\mu)\| = \|\mathbf{M}(\mu)^{-1}\mathbf{r}(\mu)\| \leq \|\mathbf{M}(\mu)^{-1}\| \|\mathbf{r}(\mu)\|. \quad (17)$$

Similar error bounds for the state error have been proposed for the reduced-basis method [17, 18] based on the weak form of the PDEs and they are written in the functional form. Here, we derive the error bound in the vector space  $\mathbb{C}^n$  for the spatially discretized system (6) written using matrices and vectors. For systems with multiple inputs,  $\mathbf{b}(\mu)$  is a matrix, then  $\mathbf{e}(\mu)$  is also a matrix. Considering the  $i$ -th column  $\mathbf{e}_i(\mu)$  of  $\mathbf{e}(\mu)$ , we get [38]

$$\|\mathbf{e}_i(\mu)\| \leq \|\mathbf{M}(\mu)^{-1}\| \|\mathbf{r}_i(\mu)\|, \quad (18)$$

where  $\mathbf{r}_i(\mu)$  is the  $i$ -th column of  $\mathbf{r}(\mu)$ . The final bound  $\Delta_s(\mu)$  is then defined as  $\Delta_s(\mu) := \max_i \|\mathbf{e}_i(\mu)\|$  for multiple input systems. In [17, 23, 53], error bounds for the nonlinear steady systems (4) and (5) are also obtained, where  $\|\mathbf{M}(\mu)^{-1}\|$  on the right-hand side of (18) is replaced by the smallest singular value of a properly defined Jacobian matrix in [17], whereas in [23], it is replaced by a lower bound on the coercivity constant of a linear operator. In [53], with some assumptions,  $\|\mathbf{e}(\mu)\|$  is bounded as

$$\|\mathbf{e}(\mu)\| \leq 2(\|\hat{\mathbf{e}}(\mu)\| + \|\mathbf{J}_f(\mu)^{-1}\| \|\mathbf{r}_r(\mu)\|), \quad (19)$$

where  $\hat{\mathbf{e}}(\mu)$  is a properly computed approximation of  $\mathbf{e}(\mu)$  which is the solution to the residual system

$$\mathbf{J}_f(\mu)\mathbf{e}(\mu) = \mathbf{r}(\mu).$$

Here,  $\mathbf{J}_f(\mu)$  is the Jacobian matrix of  $\mathbf{f}$  in (4) w.r.t.  $\hat{\mathbf{x}}(\mu)$ . In (19),  $\mathbf{r}_r(\mu) := \mathbf{r}(\mu) - \mathbf{J}_f\hat{\mathbf{e}}(\mu)$  is the residual induced by the approximation  $\hat{\mathbf{e}}(\mu)$  to  $\mathbf{e}(\mu)$ .

In summary, the error bound  $\Delta_s(\mu)$  as well as the error bounds derived in [17, 18, 23] all depend on the spectral norm of a properly defined matrix, or a coercivity constant (usually a lower bound of it), which entail computational complexity depending on the large dimension  $N$ . Sometimes,  $\|\mathbf{M}(\mu)^{-1}\|$  or the coercivity constant is so small leading to very rough error estimation.

**Output error bound**

Analogous to the time-evolution systems, estimating the output error  $\mathbf{y}(\mu) - \hat{\mathbf{y}}(\mu)$  for the ROM (7) is also based on a dual system and its ROM defined respectively as

$$\mathbf{M}^T(\mu)\mathbf{x}_{du}(\mu) = \mathbf{c}^T(\mu), \quad (20)$$

$$\hat{\mathbf{M}}(\mu)\hat{\mathbf{x}}_{du}(\mu) = \hat{\mathbf{c}}(\mu), \quad (21)$$

where  $\hat{\mathbf{M}}(\mu) = \mathbf{V}_{du}^T \mathbf{M}^T(\mu) \mathbf{V}_{du}$ ,  $\hat{\mathbf{c}}(\mu) = \mathbf{V}_{du}^T \mathbf{c}^T(\mu)$ . The following Theorem states an output error bound using the dual system (20) and its ROM (21).

**Theorem 3** (Output error bound for linear steady systems [32, 33]) *The output error  $|\mathbf{e}_o(\mu)| := |\mathbf{y}(\mu) - \hat{\mathbf{y}}(\mu)|$  of the ROM (7) is bounded by*

$$|\mathbf{e}_o(\mu)| \leq \|\mathbf{r}_{du}(\mu)\| \|\mathbf{r}(\mu)\| / \|\mathbf{M}^{-1}(\mu)\| + \|[\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu)]^T \mathbf{r}(\mu)\|. \quad (22)$$

Here  $\mathbf{r}_{du}(\mu) = \mathbf{c}^T(\mu) - \mathbf{M}^T(\mu)\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu)$ .

*Proof* See [33]. □

The above primal-dual based output error bound is motivated by the primal-dual error bounds early proposed in [20–22], etc, though the derivations in [20–22] are in function space and therefore are different. An even earlier proposed primal-dual based output error bound in function space can be found in [25]. For systems with multiple inputs and multiple outputs, an error bound with matrix-max norm can be derived [33]. To this end, we first get the error bound for the  $(i, j)$ -th entry of the output  $\mathbf{e}_o(\mu)$  matrix, which can be straightforwardly derived from (22), i.e.,

$$|\mathbf{e}_o(\mu)^{ij}| \leq \|\mathbf{r}_{du,i}(\mu)\| \|\mathbf{r}_j(\mu)\| / \|\mathbf{M}^{-1}(\mu)\| + \|[\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu)]^T \mathbf{r}_j(\mu)\|, \quad (23)$$

where  $\mathbf{r}_{du,i}(\mu) := \mathbf{c}_i^T(\mu) - \mathbf{M}^T(\mu)\mathbf{V}_{du}\hat{\mathbf{x}}_{du,i}(\mu)$  is derived from the  $i$ -th row  $\mathbf{c}_i^T(\mu)$  of the matrix  $\mathbf{c}(\mu)$  and  $\mathbf{r}_j(\mu) := \mathbf{b}_j(\mu) - \mathbf{M}(\mu)\mathbf{V}\hat{\mathbf{x}}_j(\mu)$  is defined by the  $j$ -th column  $\mathbf{b}_j(\mu)$  of  $\mathbf{b}(\mu)$ . Here,  $\hat{\mathbf{x}}_{du,i}(\mu)$  and  $\hat{\mathbf{x}}_j(\mu)$  are the  $i$ -th and  $j$ -th columns of  $\hat{\mathbf{x}}_{du}(\mu)$  and  $\hat{\mathbf{x}}(\mu)$ , respectively. The error bound for  $\|\mathbf{e}_o(\mu)\|_{\max}$  is defined as

$$\|\mathbf{e}_o(\mu)\|_{\max} \leq \max_{1 \leq i \leq n_o, 1 \leq j \leq n_l} \|\mathbf{e}_o(\mu)^{ij}\|.$$

The error bounds in (22) and (23) do not quadratically decay, since there is a second term which is not a quadratic function of the two residual norms. However, with some modifications or assumptions, we can derive error bounds that are quadratic.

**Theorem 4** (Output error bound for linear steady systems with quadratic behavior) *If  $\mathbf{b}(\mu) = \mathbf{c}^T(\mu)$ ,*

$$|\mathbf{e}_o(\mu)| \leq \|\mathbf{r}_{du}(\mu)\| \|\mathbf{r}(\mu)\| / \|\mathbf{M}^{-1}(\mu)\|. \quad (24)$$

*When  $\mathbf{b}(\mu) \neq \mathbf{c}^T(\mu)$ , if we modify the output of the ROM in (7) to  $\bar{\mathbf{y}}(\mu) = \hat{\mathbf{y}}(\mu) + (\mathbf{V}_{du}\hat{\mathbf{x}}_{du})^T \mathbf{r}(\mu)$ , then the output error bound for the output error  $\mathbf{y}(\mu) - \bar{\mathbf{y}}(\mu)$  becomes*

$$|\mathbf{y}(\mu) - \bar{\mathbf{y}}(\mu)| \leq \|\mathbf{r}_{du}(\mu)\| \|\mathbf{r}(\mu)\| / \|\mathbf{M}^{-1}(\mu)\|.$$

*Proof* When  $\mathbf{b}(\mu) = \mathbf{c}^T(\mu)$ , the dual system (20) is the same as the primal system (6), so that  $\mathbf{V}_{du} = \mathbf{V}$  and  $\hat{\mathbf{x}}_{du}(\mu) = \hat{\mathbf{x}}(\mu)$ . This leads to  $\|[\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu)]^T \mathbf{r}(\mu)\| = 0$  in (22) or

$$\begin{aligned}
\|[\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu)]^T \mathbf{r}_j(\mu)\| &= 0 \text{ in (23). When } \mathbf{b}(\mu) \neq \mathbf{c}^T(\mu), \text{ we have} \\
\|\mathbf{y}(\mu) - \hat{\mathbf{y}}(\mu)\| &= \|\mathbf{c}(\mu)(\mathbf{x}(\mu) - \mathbf{V}\hat{\mathbf{x}}(\mu)) - (\mathbf{V}_{du}\hat{\mathbf{x}}_{du})^T \mathbf{r}(\mu)\|, \\
&= \|\mathbf{c}(\mu)\mathbf{M}(\mu)^{-1}[\mathbf{M}(\mu)\mathbf{x}(\mu) - \mathbf{M}(\mu)\mathbf{V}\hat{\mathbf{x}}(\mu)] - (\mathbf{V}_{du}\hat{\mathbf{x}}_{du})^T \mathbf{r}(\mu)\|, \\
&= \|\mathbf{c}(\mu)\mathbf{M}(\mu)^{-1}\mathbf{r}(\mu) - (\mathbf{V}_{du}\hat{\mathbf{x}}_{du})^T \mathbf{r}(\mu)\|, \\
&= \|[\mathbf{M}(\mu)^{-T}\mathbf{c}(\mu)^T - (\mathbf{V}_{du}\hat{\mathbf{x}}_{du})^T] \mathbf{r}(\mu)\|, \\
&= \|\mathbf{M}(\mu)^{-T}[\mathbf{c}(\mu)^T - \mathbf{M}(\mu)^T(\mathbf{V}_{du}\hat{\mathbf{x}}_{du})^T] \mathbf{r}(\mu)\|, \\
&= \|\mathbf{M}(\mu)^{-1}\mathbf{r}_{du}(\mu)^T \mathbf{r}(\mu)\|, \\
&\leq \|\mathbf{M}(\mu)^{-1}\| \|\mathbf{r}_{du}(\mu)\| \|\mathbf{r}(\mu)\|.
\end{aligned}$$

□

With the corrected output, the output error bound has a *quadratic* behavior. The same technique was previously used in [20–22, 25] for error analysis in function space. The error bound in (24) is in agreement with the analysis in, e.g., [20]. It is worth pointing out that using a corrected output to obtain error estimation with quadratic decay was early proposed for the finite element method (FEM) [37]. Analogous to the state error bound in (18), the smallest singular value of  $\mathbf{M}(\mu)$  for any given  $\mu$  must be computed in order to compute the output error bound.

### A posteriori error estimators

This section discusses a posteriori error estimators that may loose the rigorousness of the error bound. However, these error estimators try to reduce the big gap between the error bounds and the true error occurring in many problems. Usually the ratio  $\frac{\text{error bound}}{\text{true error}}$  or  $\frac{\text{error estimator}}{\text{true error}}$  is considered as the effectivity of an error bound/estimator. A posteriori error estimators discussed in this section are aimed to have effectivities  $\frac{\text{error estimator}}{\text{true error}}$  closer to 1 as compared to those  $\frac{\text{error bound}}{\text{true error}}$  of the error bounds. At the same time, they usually have less computational complexity than the error bound. In the following subsections, we also separately discuss error estimation for time-evolution systems and that for steady systems.

#### Error estimators for time-evolution systems

The error estimators discussed in this subsection aim to estimate the error of the time-discrete ROM in (9). The works in [29–31] propose output error estimators which avoid accumulating (summing up) the residuals over the time evolution, resulting in much tighter error estimators than the primal-dual based error bounds in “[Error bounds for time-evolution systems](#)” section. Furthermore, the output error estimators in [29, 30] apply to both nonlinear and linear systems. For nonlinear systems, the error estimators could also include the approximation error of hyperreduction. We review those error estimators in the following theorems.

#### Output error estimators

The output error estimators needs a dual system defined as,

$$\mathbf{A}_t(\mu)^T \mathbf{x}_{du}^k(\mu) = \mathbf{c}(\mu)^T. \quad (25)$$

The ROM of the dual system can be derived by

$$\hat{\mathbf{A}}_t(\mu) \hat{\mathbf{x}}_{du}^k(\mu) = \hat{\mathbf{c}}(\mu), \quad (26)$$

where  $\hat{\mathbf{A}}_t(\mu) = \mathbf{V}_{du}^T \mathbf{A}_t(\mu)^T \mathbf{V}_{du}$ ,  $\hat{\mathbf{c}}(\mu) = \mathbf{V}_{du}^T \mathbf{c}(\mu)^T$ .

*Remark 7* The dual system defined in (25) is slightly different from that in [29–31], where the right-hand side is  $-\mathbf{c}(\mu)^T$  instead of  $\mathbf{c}(\mu)^T$ . To be consistent with the definition of the corrected output  $\tilde{\mathbf{y}}(\mu)$  for the steady systems in Theorem 4, we use the dual system (25), based on which the corrected output for the nonlinear time-evolution systems to be defined later will have a uniform form as the corrected output for the steady systems.

The residual induced by the approximate solution  $\mathbf{V}_{du} \hat{\mathbf{x}}_{du}$  computed from the ROM in (26) is

$$\mathbf{r}_{du}^k(\mu) := \mathbf{c}(\mu)^T - \mathbf{A}_t(\mu)^T \mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu). \quad (27)$$

Define an auxiliary residual:

$$\tilde{\mathbf{r}}^k(\mu) := \mathbf{E}(\mu) \mathbf{x}^{k-1} + \delta t (\mathbf{f}(\mathbf{x}^{k-1}(\mu), \mu) + \mathbf{b}(\mu) u^k(\mu)) - \mathbf{A}_t(\mu) \tilde{\mathbf{x}}^k.$$

It can be seen that the differences of  $\tilde{\mathbf{r}}^k(\mu)$  from  $\mathbf{r}^k(\mu)$  in (10) are that  $\mathbf{f}(\tilde{\mathbf{x}}^{k-1}(\mu))$  in  $\mathbf{r}^k(\mu)$  is replaced by  $\mathbf{f}(\mathbf{x}^{k-1}(\mu), \mu)$ , i.e.,  $\mathbf{f}$  applied to the true solution  $\mathbf{x}^{k-1}(\mu)$  and  $\mathbf{E}(\mu) \tilde{\mathbf{x}}^{k-1}(\mu)$  is replaced by  $\mathbf{E}(\mu) \mathbf{x}^{k-1}(\mu)$ . With  $\tilde{\mathbf{r}}^k(\mu)$ , we will derive a direct relation between  $\tilde{\mathbf{r}}^k(\mu)$  and the state error  $\mathbf{x}^k(\mu) - \tilde{\mathbf{x}}^k(\mu)$ . This relation will aid the derivation of the output error estimation.

**Theorem 5** (Primal-dual based output error estimator [29, 30]) *For the time-discrete FOM (8) and the time-discrete ROM (9), assume that  $\mathbf{A}_t(\mu)$  is invertible for any  $\mu \in \mathcal{P}$ , then the output error  $e_o^k(\mu) = \mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)$  at the time instance  $t^k$  can be bounded as*

$$|e_o^k(\mu)| \leq \tilde{\rho}(\mu) \Phi^k(\mu) \|\mathbf{r}^k(\mu)\|, \quad k = 1, \dots, K, \quad (28)$$

where  $\tilde{\rho}^k(\mu) = \|\tilde{\mathbf{r}}^k(\mu)\| / \|\mathbf{r}^k(\mu)\|$  and  $\Phi^k(\mu) = \|\mathbf{A}_t(\mu)^{-1}\| \|\mathbf{r}_{du}^k(\mu)\| + \|\mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu)\|$ .

*Proof* From the dual system (25), we have

$$(\mathbf{x}_{du}^k(\mu))^T \mathbf{A}_t(\mu) = \mathbf{c}(\mu). \quad (29)$$

Multiplying  $\mathbf{x}^k(\mu) - \tilde{\mathbf{x}}^k(\mu)$  from the right on both sides of (29), we get

$$(\mathbf{x}_{du}^k(\mu))^T \mathbf{A}_t(\mu) [\mathbf{x}^k(\mu) - \tilde{\mathbf{x}}^k(\mu)] = \mathbf{c}(\mu) [\mathbf{x}^k(\mu) - \tilde{\mathbf{x}}^k(\mu)].$$

Recalling the definition of  $\tilde{\mathbf{r}}^k(\mu)$  in (7), we have

$$\tilde{\mathbf{r}}^k(\mu) = \mathbf{A}_t(\mu) [\mathbf{x}^k(\mu) - \tilde{\mathbf{x}}^k(\mu)].$$

We combine the above two equations to obtain

$$(\mathbf{x}_{du}^k(\mu))^T \tilde{\mathbf{r}}^k(\mu) = \mathbf{c}(\mu) [\mathbf{x}^k(\mu) - \tilde{\mathbf{x}}^k(\mu)] \quad (30)$$

Defining a new vector  $\tilde{\mathbf{y}}^k(\mu) = \hat{\mathbf{y}}^k(\mu) + (\mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu))^T \tilde{\mathbf{r}}^k(\mu)$  yields

$$\begin{aligned} |\mathbf{y}^k(\mu) - \tilde{\mathbf{y}}^k(\mu)| &= |\mathbf{c}(\mu) [\mathbf{x}^k(\mu) - \tilde{\mathbf{x}}^k(\mu)] - (\mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu))^T \tilde{\mathbf{r}}^k(\mu)| \\ &= |(\mathbf{x}_{du}^k(\mu))^T \tilde{\mathbf{r}}^k(\mu) - (\mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu))^T \tilde{\mathbf{r}}^k(\mu)| \quad (\text{using Eq. (30)}) \end{aligned} \quad (31)$$

Inserting the relation

$$\mathbf{A}_t(\mu)^T [\mathbf{x}_{du}^k(\mu) - \mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu)] = \mathbf{r}_{du}^k(\mu),$$

into (31), leads to

$$\begin{aligned} |\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)| &= |[\mathbf{r}_{du}^k(\mu)]^T [\mathbf{A}_t(\mu)]^{-1} \tilde{\mathbf{r}}^k(\mu)| \\ &\leq \|[\mathbf{A}_t(\mu)]^{-1}\| \|\mathbf{r}_{du}^k(\mu)\| \|\tilde{\mathbf{r}}^k(\mu)\|. \end{aligned} \quad (32)$$

Finally,

$$\begin{aligned} |\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)| &\leq \|[\mathbf{A}_t(\mu)]^{-1}\| \|\mathbf{r}_{du}^k(\mu)\| \|\tilde{\mathbf{r}}^k(\mu)\| + |(\mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu))^T \tilde{\mathbf{r}}^k(\mu)| \\ &\leq \|[\mathbf{A}_t(\mu)]^{-1}\| \|\mathbf{r}_{du}^k(\mu)\| \|\tilde{\mathbf{r}}^k(\mu)\| + \|\mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu)\| \|\tilde{\mathbf{r}}^k(\mu)\|. \end{aligned}$$

Defining  $\tilde{\rho}^k(\mu) = \|\tilde{\mathbf{r}}^k(\mu)\| / \|\mathbf{r}^k(\mu)\|$ , we obtain the desired error bound.  $\square$

However,  $\tilde{\mathbf{r}}^k(\mu)$  involves the true solution vector  $\mathbf{x}^k(\mu)$ , making  $\tilde{\rho}^k(\mu)$  not computable. In [29, 30], the nonlinear function  $\mathbf{f}(\mathbf{x}(t, \mu), \mu)$  is assumed to be Lipschitz continuous w.r.t.  $\mathbf{x}(t, \mu)$ , in order to derive a computable estimation  $\rho(\mu)$  for  $\tilde{\rho}^k(\mu)$ . The output error can then be estimated as [29, 30]

$$|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)| \approx \rho(\mu) \Phi^k(\mu) \|\mathbf{r}^k(\mu)\|. \quad (33)$$

The details of computing  $\rho(\mu)$  can be found in [29, 30].

*Remark 8* Recall that the dual system defined in (26) is a bit different, therefore the proof in [29, 30] does not directly apply here. Although the proof above is similar to the proofs in [29, 30], it is not the same. In particular the variable  $\tilde{\mathbf{y}}^k(\mu)$  is different, which is  $\tilde{\mathbf{y}}^k(\mu) = \hat{\mathbf{c}}(\mu)^T \hat{\mathbf{x}}^k(\mu) - (\mathbf{V}_{du} \hat{\mathbf{x}}_{du}(\mu))^T \tilde{\mathbf{r}}^k(\mu)$  in [29, 30].

When hyperreduction is applied to the ROM (9), we get the hyperreduced ROM in (11). An output error estimation for the hyperreduced ROM is derived in [29, 30] as

$$\|e_o^k(\mu)\| \approx \rho(\mu) \Phi^k(\mu) (\|\mathbf{r}_{\mathcal{I}}^k(\mu)\| + \epsilon_{\mathcal{I}}^k(\mu)), \quad k = 1, \dots, K, \quad (34)$$

where  $\mathbf{r}_{\mathcal{I}}^k(\mu) := \mathbf{E}(\mu) \mathbf{V} \hat{\mathbf{x}}^{k-1}(\mu) + \delta t (\mathcal{I}_{\mathbf{f}}^{k-1}(\mu) + \mathbf{b}u^k) - \mathbf{A}_t \mathbf{V} \hat{\mathbf{x}}^k(\mu)$ , and  $\epsilon_{\mathcal{I}}^k(\mu)$  is an estimation of the interpolation error  $\|\delta t [\mathbf{f}(\mathbf{x}^k(\mu), \mu) - \mathcal{I}_{\mathbf{f}}^k(\mu)]\|$ . The quantity  $\epsilon_{\mathcal{I}}^k(\mu)$  can be computed using, e.g., the technique in [51]. The output error estimators in (33) and (34) do not include the sum of the residuals over time instances, and are expected to be much tighter than the rigorous output error bound. In the numerical results in [29] for a linear system, it is shown that the error estimator yields a more accurate estimation of the true error than the error bound in [19, 27].

*Remark 9* For multiple-input multiple-output systems, the corresponding output error estimator can be obtained using the matrix-max norm as explained in (23) and (24).

The error estimators in (33) and (34) do not quadratically decay w.r.t. the two residuals because of the second part in  $\Phi^k(\mu)$ . In [31], we use a corrected output of the ROM, so that the finally derived error estimator includes much less contribution of the second part in  $\Phi^k(\mu)$ . This makes the error estimator decay almost quadratic.

Define a corrected output  $\tilde{\mathbf{y}}^k(\mu) = \hat{\mathbf{y}}^k(\mu) + (\mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu))^T \mathbf{r}^k(\mu)$  for the ROM in (9). With the same assumptions as in Theorem 5, and the Lipschitz continuity of  $\mathbf{f}(\mathbf{x}(t, \mu), \mu)$ , the output error  $\tilde{e}_o^k(\mu) = \mathbf{y}^k(\mu) - \tilde{\mathbf{y}}^k(\mu)$  can be estimated as [31]

$$\|\tilde{e}_o^k(\mu)\| \approx \tilde{\Phi}^k(\mu) \|\mathbf{r}^k(\mu)\|, \quad k = 1, \dots, K, \quad (35)$$

where  $\tilde{\Phi}^k(\mu) = \rho \|[\mathbf{A}_t(\mu)]^{-1}\| \|\mathbf{r}_{du}^k(\mu)\| + |1 - \rho| \|\mathbf{V}_{du} \hat{\mathbf{x}}_{du}^k(\mu)\|$ .

Comparing the error estimator in (35) with that in (33), we find that the second non-quadratic term is still there, but with a scaling factor  $|1 - \rho|$  instead of  $\rho$ . It is analyzed in [30] that under certain assumptions, when the POD-greedy algorithm used to compute the projection matrix  $\mathbf{V}$  converges,  $\rho$  gets closer to 1, meaning that  $|1 - \rho|$  will be closer to 0. This makes the second part in (35) tend to zero, while the second part in (28) remains away from zero. Therefore, the error estimator for the corrected output error should give a tighter estimation. The derivation follows almost that in [31] and the proof for Theorem 5, noticing that the dual system and the corrected output are slightly different from those in [31]. We will not repeat it here.

With simple calculations, the corrected output error for the hyperreduced ROM in (11) can be estimated as [31]

$$\|\tilde{e}_o^k(\mu)\| \approx \tilde{\Phi}^k(\mu)\|\mathbf{r}_Z^k(\mu)\| + \tilde{\Phi}^k(\mu)\epsilon_Z^k, \quad k = 1, \dots, K. \quad (36)$$

### Error estimators for linear steady systems

Some error estimators [33, 34, 38, 54] for linear steady systems were proposed in order to avoid estimating/computing the spectral norm  $\|\mathbf{M}(\mu)^{-1}\|$  involved in the error bounds in “Error bounds for steady systems” section. An approach based on randomized residuals is proposed in [54], where some randomized systems are defined to get error estimators for both the state error and the output error. It is discussed in [34] and [38] that the error estimators in [54] are theoretically less accurate than the estimators proposed in [33, 38]. The error estimators in [54] more easily underestimate the true error than the estimators in [33, 38], which is also numerically demonstrated in [33, 38]. Here we review the error estimators proposed in our recent work [33, 34, 38].

### State error estimators

The error estimator proposed in [38] estimates the state error for linear steady systems. For the FOM in (6), the error  $\mathbf{e}(\mu) := \mathbf{x}(\mu) - \mathbf{V}\hat{\mathbf{x}}(\mu)$  of the approximate state  $\mathbf{V}\hat{\mathbf{x}}(\mu)$  computed by the ROM (7) can be estimated as

$$|\mathbf{e}(\mu)| \approx |\mathbf{V}_r\hat{\mathbf{x}}_r(\mu)|, \quad (37)$$

where  $\hat{\mathbf{x}}_r(\mu)$  is the solution to the following ROM

$$\mathbf{M}_r(\mu)\hat{\mathbf{x}}_r(\mu) = \hat{\mathbf{r}}(\mu). \quad (38)$$

Here,  $\mathbf{M}_r(\mu) = \mathbf{V}_r^T \mathbf{M}(\mu) \mathbf{V}_r$ ,  $\hat{\mathbf{r}}(\mu) = \mathbf{V}_r \mathbf{r}(\mu)$  with  $\mathbf{V}_r$  being properly derived, and  $\mathbf{r}(\mu) = \mathbf{b}(\mu) - \mathbf{M}(\mu)\mathbf{V}\hat{\mathbf{x}}(\mu)$ . The system (38) is the ROM of the following residual system

$$\mathbf{M}(\mu)\mathbf{x}_r(\mu) = \mathbf{r}(\mu). \quad (39)$$

*Remark 10* We note that a similar technique of using an approximate solution to a residual system, as an error estimator for the state error, was already proposed for the finite element method (FEM) (see [37] and the references therein). There, the approximate solution was not obtained from a ROM of the residual system.

The accuracy of the error estimator  $|\mathbf{V}_r\hat{\mathbf{x}}_r(\mu)|$  in (37) is quantified in [38]:

**Theorem 6** (Quantifying the error estimator [38]) *The state error  $|\mathbf{e}(\mu)|$  is lower and upper bounded as*

$$|\mathbf{V}_r\hat{\mathbf{x}}_r(\mu)| - \delta(\mu) \leq |\mathbf{e}(\mu)| \leq |\mathbf{V}_r\hat{\mathbf{x}}_r(\mu)| + \delta(\mu),$$

where  $\delta(\mu) = |\mathbf{x}_r(\mu) - \mathbf{V}_r \hat{\mathbf{x}}_r(\mu)|$ .

Whenever the ROM (38) of the residual system is accurate enough,  $\delta(\mu)$  will be small. However, how to further quantify the error  $\delta(\mu)$  is left open. We derive the following theorem with computable upper and lower bounds.

**Theorem 7** (Quantifying the error estimator with computable upper and lower bound)  
The state error  $|\mathbf{e}(\mu)|$  is lower and upper bounded as

$$|\mathbf{V}_r \hat{\mathbf{x}}_r(\mu) - \bar{\delta}(\mu)| \leq |\mathbf{e}(\mu)| \leq |\mathbf{V}_r \hat{\mathbf{x}}_r(\mu) + \bar{\delta}(\mu)|, \quad (40)$$

where  $\bar{\delta}(\mu) = \|\mathbf{M}(\mu)^{-1}\| \|\mathbf{r}_r(\mu)\|$ .

*Proof* The proof can be easily done. We notice that

$$\mathbf{M}(\mu)[\mathbf{x}_r(\mu) - \mathbf{V}_r \hat{\mathbf{x}}_r(\mu)] = \mathbf{r}_r(\mu),$$

with  $\mathbf{r}_r(\mu) := \mathbf{r}(\mu) - \mathbf{M}(\mu)\mathbf{V}_r \hat{\mathbf{x}}_r(\mu)$ . Then

$$\delta(\mu) = \|\mathbf{M}(\mu)^{-1}\mathbf{r}_r(\mu)\| \leq \|\mathbf{M}(\mu)^{-1}\| \|\mathbf{r}_r(\mu)\| = \bar{\delta}(\mu).$$

□

Note that for linear systems, the upper bound in (40) is only half the upper bound in (19).

#### Output error estimators

The error estimators in [33, 34] estimate the error  $\mathbf{e}_o(\mu) := \mathbf{y}(\mu) - \hat{\mathbf{y}}(\mu)$  of the output  $\hat{\mathbf{y}}(\mu)$  computed from the ROM (7). In [34], we derive the following primal-dual based output error estimator

$$|\mathbf{e}_o(\mu)| \approx |[\mathbf{V}_{du} \hat{\mathbf{x}}_{du}(\mu)]^T \mathbf{r}(\mu)| =: \Delta_{o_1}(\mu),$$

where  $\hat{\mathbf{x}}_{du}(\mu)$  is the solution to the reduced dual system

$$\hat{\mathbf{M}}_{du}(\mu) \hat{\mathbf{x}}_{du} = \hat{\mathbf{c}}_{du}(\mu), \quad (41)$$

and  $\hat{\mathbf{M}}_{du}(\mu) = \mathbf{V}_{du}^T \mathbf{M}(\mu)^T \mathbf{V}_{du}$ ,  $\hat{\mathbf{c}}_{du}(\mu) = \mathbf{V}_{du}^T \mathbf{c}(\mu)^T$ . The reduced dual system is a ROM of the dual system,

$$\mathbf{M}(\mu)^T \mathbf{x}_{du} = \mathbf{c}(\mu)^T. \quad (42)$$

*Remark 11* In [37] and the references therein, the FEM approximation error was estimated also using a similarly defined dual system in the function space. However, the approximate solution to the dual system is not the solution of the ROM for the dual system. The approximate dual solution is then multiplied with the residual of the FEM approximation to the original PDEs to constitute a primal-dual based error estimator for the output error of the FEM approximation.

The randomized output error estimator in [54] is based on the output error estimator  $\Delta_{o_1}(\mu)$ . On the one hand, it is analysed in [34] that  $\Delta_{o_1}(\mu)$  is more accurate than the randomized output error estimator; on the other hand, it is also numerically demonstrated in [34] that  $\Delta_{o_1}(\mu)$  is nevertheless less accurate than the other estimators proposed in [33, 34]. In the following, we first introduce the primal-dual output error estimator in [33], which involves a dual-residual system defined as

$$\mathbf{M}(\mu)^T \mathbf{x}_{r_{du}}(\mu) = \mathbf{r}_{du}(\mu),$$

and its ROM

$$\hat{\mathbf{M}}_{r_{du}}(\mu)\hat{\mathbf{x}}_{r_{du}}(\mu) = \hat{\mathbf{r}}_{du}(\mu),$$

where  $\hat{\mathbf{M}}_{r_{du}}(\mu) = \mathbf{V}_{r_{du}}^T \mathbf{M}(\mu) \mathbf{V}_{r_{du}}$ ,  $\hat{\mathbf{r}}_{du}(\mu) = \mathbf{V}_{r_{du}}^T \mathbf{r}_{du}(\mu)$ , with  $\mathbf{V}_{r_{du}}$  being properly computed. The dual-residual  $\mathbf{r}_{du}(\mu) := \mathbf{c}(\mu)^T - \mathbf{M}(\mu)^T \mathbf{V}_{du} \hat{\mathbf{x}}_{du}$  is the residual induced by the approximate solution  $\mathbf{V}_{du} \hat{\mathbf{x}}_{du}$  computed from the dual ROM (41). A primal-dual and dual-residual based output error estimator proposed in [33] is stated as following. For the FOM in (6), the output error  $\mathbf{e}_o(\mu)$  of the ROM (7) can be estimated as

$$|\mathbf{e}_o(\mu)| \approx |[\mathbf{V}_{du} \hat{\mathbf{x}}_{du}(\mu)]^T \mathbf{r}(\mu)| + |(\mathbf{V}_{r_{du}} \hat{\mathbf{x}}_{r_{du}})^T \mathbf{r}(\mu)| =: \Delta_{o_2}(\mu). \quad (43)$$

The error estimator  $\Delta_{o_2}(\mu)$  in (43) has an additional term  $|(\mathbf{V}_{r_{du}} \hat{\mathbf{x}}_{r_{du}})^T \mathbf{r}(\mu)|$  as compared to  $\Delta_{o_1}(\mu)$ . Now we discuss the accuracy of both estimators through the next Theorems.

**Theorem 8** (Quantifying the output error estimator  $\Delta_{o_1}(\mu)$  [34]) *The output error  $\mathbf{e}_o(\mu)$  is bounded as*

$$\Delta_{o_1}(\mu) - \delta_1(\mu) \leq |\mathbf{e}_o(\mu)| \leq \Delta_{o_1}(\mu) + \delta_1(\mu),$$

where  $\delta_1(\mu) := |[\mathbf{x}_{du}(\mu) - \mathbf{V}_{du} \hat{\mathbf{x}}_{du}(\mu)]^T \mathbf{r}(\mu)| = |\mathbf{r}_{du}(\mu)^T (\mathbf{M}(\mu))^{-1} \mathbf{r}(\mu)|$ .

*Proof* See [34]. □

**Theorem 9** (Quantifying the output error estimator  $\Delta_{o_2}(\mu)$  [34]) *The output error  $\mathbf{e}_o(\mu)$  is bounded as*

$$\Delta_{o_2}(\mu) - \delta_2(\mu) - |(\mathbf{V}_{r_{du}} \hat{\mathbf{x}}_{r_{du}})^T \mathbf{r}(\mu)| \leq |\mathbf{e}_o(\mu)| \leq \Delta_{o_2}(\mu) + \delta_2(\mu),$$

where  $\delta_2(\mu) := |[\mathbf{x}_{r_{du}}(\mu) - \mathbf{V}_{r_{du}} \hat{\mathbf{x}}_{r_{du}}(\mu)]^T \mathbf{r}(\mu)|$ .

*Proof* See [34]. □

We can observe that

$$\begin{aligned} \delta_2(\mu) &= |[\mathbf{x}_{r_{du}}(\mu) - \mathbf{V}_{r_{du}} \hat{\mathbf{x}}_{r_{du}}(\mu)]^T \mathbf{r}(\mu)|, \\ &= |[\mathbf{M}(\mu)^T \mathbf{x}_{r_{du}}(\mu) - \mathbf{M}(\mu)^T \mathbf{V}_{r_{du}} \hat{\mathbf{x}}_{r_{du}}(\mu)]^T (\mathbf{M}(\mu))^{-1} \mathbf{r}(\mu)|, \\ &= |[\mathbf{r}_{du}(\mu) - \mathbf{M}(\mu)^T \mathbf{V}_{r_{du}} \hat{\mathbf{x}}_{r_{du}}(\mu)]^T (\mathbf{M}(\mu))^{-1} \mathbf{r}(\mu)|, \\ &= |\mathbf{r}_{r_{du}}(\mu)^T (\mathbf{M}(\mu))^{-1} \mathbf{r}(\mu)|, \end{aligned}$$

where  $\mathbf{r}_{r_{du}}(\mu) := \mathbf{r}_{du}(\mu) - \mathbf{M}(\mu)^T \mathbf{V}_{r_{du}} \hat{\mathbf{x}}_{r_{du}}(\mu)$  is the residual induced by the reduced dual-residual system.

We can further derive upper bounds for  $\delta_1(\mu)$  and  $\delta_2(\mu)$ , respectively. Actually,

$$\begin{aligned} \delta_1(\mu) &\leq \|\mathbf{r}_{du}(\mu)\| \|(\mathbf{M}(\mu))^{-1}\| \|\mathbf{r}(\mu)\| =: \bar{\delta}_1(\mu), \\ \delta_2(\mu) &\leq \|\mathbf{r}_{r_{du}}(\mu)\| \|(\mathbf{M}(\mu))^{-1}\| \|\mathbf{r}(\mu)\| =: \bar{\delta}_2(\mu). \end{aligned}$$

Although we have no proof yet, it is expected that  $\|\mathbf{r}_{r_{du}}(\mu)\| \leq \|\mathbf{r}_{du}\|$  in general, since  $\mathbf{r}_{r_{du}}$  is the residual induced by the ROM of the dual-residual system whose right-hand side is  $\mathbf{r}_{du}$ . Finally, we should have  $\bar{\delta}_2(\mu) \leq \bar{\delta}_1(\mu)$ , indicating that  $\Delta_{o_2}(\mu)$  should be more accurate than  $\Delta_{o_1}(\mu)$ . On the other hand, we know that

$$\begin{aligned} |\mathbf{e}_o| - \bar{\delta}_1(\mu) &\leq \Delta_{o_1}(\mu), \\ |\mathbf{e}_o| - \bar{\delta}_2(\mu) &\leq \Delta_{o_2}(\mu). \end{aligned}$$



Then  $\bar{\delta}_2(\mu) \leq \bar{\delta}_1(\mu)$  implies that underestimation of the true error by  $\Delta_{o_2}(\mu)$  should be less than by  $\Delta_{o_1}(\mu)$ .

In [34], we have further proposed another output error estimator variant  $\Delta_{o_3}(\mu)$ , which has less computational complexity than  $\Delta_{o_2}(\mu)$ , but has similar, or sometimes even better accuracy. It does not depend on the dual system and/or dual-residual system as  $\Delta_{o_1}(\mu)$  and  $\Delta_{o_2}(\mu)$ , but depends on the primal-residual system in (39).  $\Delta_{o_3}(\mu)$  is defined as

$$|\mathbf{e}_o(\mu)| \approx |\mathbf{c}(\mu)\mathbf{V}_r\hat{\mathbf{x}}_r(\mu)| =: \Delta_{o_3}(\mu). \quad (44)$$

Comparing  $\Delta_{o_3}(\mu)$  with the state error estimator  $|\mathbf{V}_r\hat{\mathbf{x}}_r(\mu)|$  in (37), we see that there is only a difference of the output matrix  $\mathbf{c}(\mu)$ . Both are derived by employing the primal-residual system (39).

**Theorem 10** (Quantifying the output error estimator  $\Delta_{o_3}(\mu)$  [34]) *The output error  $\mathbf{e}_o(\mu)$  is bounded as*

$$\Delta_{o_3}(\mu) - \delta_3(\mu) \leq |\mathbf{e}_o(\mu)| \leq \Delta_{o_3}(\mu) + \delta_3(\mu),$$

where  $\delta_3(\mu) := |\mathbf{c}(\mu)[\mathbf{x}_r(\mu) - \mathbf{V}_r\hat{\mathbf{x}}_r(\mu)]|$ .

With simple calculations, an upper bound of  $\delta_3(\mu)$  can be derived as

$$\delta_3(\mu) \leq \bar{\delta}_3(\mu) := \|\mathbf{c}(\mu)\| \|\mathbf{M}(\mu)^{-1}\| \|\mathbf{r}_r(\mu)\| = \|\mathbf{c}(\mu)\| \bar{\delta}(\mu) \quad (\text{Theorem 7}).$$

It can be easily seen that computing  $\Delta_{o_3}(\mu)$  needs only to compute one additional ROM, i.e., the ROM of the primal-residual system (38), while computing  $\Delta_{o_2}(\mu)$  needs to compute two additional ROMs. Theoretically, the upper bound  $\bar{\delta}_2(\mu)$  should decay faster than the upper bound  $\bar{\delta}_3(\mu)$ , implicating that  $\Delta_{o_2}(\mu)$  should be more accurate than  $\Delta_{o_3}(\mu)$ . However, from our numerical simulations on several different problems [34],  $\Delta_{o_3}(\mu)$  is even more accurate than  $\Delta_{o_2}(\mu)$ .

### Error estimator for ROMs solved with any black-box time-integration solver

The error bounds and error estimators reviewed in the previous sections are all residual based. In particular, for time-evolution systems, the error bound and estimators need to compute the residual  $\mathbf{r}^k(\mu)$  at corresponding time instances  $t_k$ ,  $k = 1, \dots, K$ . It is clear that to compute  $\mathbf{r}^k(\mu)$ , the temporal discretization scheme applied to the FOM must be known, so that  $\mathbf{r}^k(\mu)$  (10) can be derived by inserting the approximate solution  $\tilde{\mathbf{x}}^k(\mu)$  into the temporal discretization scheme, e.g., (8) and by subtracting the left-hand side of the first equation from its right-hand side. Moreover, the temporal discretization scheme (8) for the ROM (3) must be the same as that for the FOM to make sure that  $\tilde{\mathbf{x}}^k(\mu)$  computed from the ROM (8) corresponds to the true solution  $\mathbf{x}^k(\mu)$  at the same time instance  $t_k$ . These two requirements on the FOM and the ROM become limitations for the error bounds (estimators) when the FOM is simulated by a black-box time-integration solver and/or when the ROM is also desired to be solved using the same black-box time-integration solver.

In [35], we propose a new error estimator which is applicable to the situation where both the FOM and the ROM are solved by a black-box solver. We take use of a user-defined implicit–explicit (IMEX) temporal discretization scheme to derive the new error estimator. Although potentially any IMEX scheme can be applied, we consider the first-order IMEX scheme (8) in this survey. Note that the second-order IMEX scheme is also used in [35].

Since the first-order IMEX scheme (8) differs from the black-box solver, we have a defect or a mismatch when we insert the solution snapshots  $\mathbf{x}^k(\mu)$  computed from the black-box solver into the first-order IMEX scheme.

$$\mathbf{d}^k(\mu) := \mathbf{A}_t(\mu)\mathbf{x}^k(\mu) - \mathbf{E}(\mu)\mathbf{x}^{k-1}(\mu) + \delta t(\mathbf{f}(\mathbf{x}^{k-1}(\mu), \mu) + \mathbf{b}(\mu)u^k).$$

Although the time-integration scheme of the black-box solver is invisible, we can use the solution snapshots  $\mathbf{x}^k(\mu)$ ,  $k = 0, \dots, K$ , at some samples of  $\mu$  to learn the defect vector. We then use  $\mathbf{d}^k(\mu)$  to correct the user-defined scheme (8), such that its solution *recovers* the solution  $\mathbf{x}^k(\mu)$  computed by a black-box solver and the temporal discretization scheme of the black-box solver then becomes visible via the corrected time-discrete FOM as below,

$$\begin{aligned} \mathbf{A}_t(\mu)\mathbf{x}_c^k(\mu) &= \mathbf{E}(\mu)\mathbf{x}_c^{k-1}(\mu) + \delta t(\mathbf{f}(\mathbf{x}_c^{k-1}(\mu), \mu) + \mathbf{b}(\mu)u^k) + \mathbf{d}^k(\mu), \\ \mathbf{y}_c^k(\mu) &= \mathbf{c}(\mu)\mathbf{x}_c^k(\mu). \end{aligned} \quad (45)$$

It is clear that if  $\mathbf{d}^k(\mu)$  can be accurately learned, then not only  $\mathbf{x}_c^k(\mu)$  in (45) recovers  $\mathbf{x}^k(\mu)$ , but also the FOM in (45) is equivalent to the temporal discretization scheme of the black-box solver. The ROM of the FOM in (45) can be obtained as

$$\begin{aligned} \hat{\mathbf{A}}_t(\mu)\hat{\mathbf{x}}_c^k(\mu) &= \hat{\mathbf{E}}(\mu)\hat{\mathbf{x}}_c^{k-1}(\mu) + \delta t(\hat{\mathbf{f}}(\mathbf{V}\hat{\mathbf{x}}_c^{k-1}(\mu), \mu) + \hat{\mathbf{b}}(\mu)u^k) + \hat{\mathbf{d}}^k(\mu), \\ \hat{\mathbf{y}}_c^k(\mu) &= \hat{\mathbf{c}}\hat{\mathbf{x}}_c^k, \end{aligned} \quad (46)$$

where  $\hat{\mathbf{A}}_t(\mu)$ ,  $\hat{\mathbf{E}}(\mu)$ ,  $\hat{\mathbf{f}}(\cdot, \cdot)$ ,  $\hat{\mathbf{b}}(\mu)$ ,  $\hat{\mathbf{c}}(\mu)$  are defined as in (3) and  $\hat{\mathbf{d}}^k(\mu) = \mathbf{V}^T \mathbf{d}^k(\mu)$ . We make use of both the corrected FOM (45) and the corresponding ROM (46) to derive output error estimation for the output error  $|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)|$ , where  $\mathbf{y}^k(\mu)$  and  $\hat{\mathbf{y}}^k(\mu)$  are the outputs of the FOM in (2) and the ROM in (3) at any time instance  $t^k$ , respectively. Both systems can be solved using any black-box solver.

Given the FOM in (2), assuming that  $\mathbf{A}_t(\mu)$  is non-singular for all  $\mu \in \mathcal{P}$ , the nonlinear function  $\mathbf{f}(\mathbf{x}(t, \mu), \mu)$  is Lipschitz continuous w.r.t.  $\mathbf{x}(t, \mu)$ , and the defect vector  $\mathbf{d}(\mu)$  can be accurately learned, then the output error  $|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)|$  of the ROM (3) can be estimated as [35]

$$|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)| \approx \bar{\Phi}^k(\mu) \|\mathbf{r}_c^k(\mu)\| + \|\hat{\mathbf{y}}_c^k(\mu) - \hat{\mathbf{y}}^k(\mu)\|, \quad (47)$$

where  $\mathbf{r}_c^k(\mu) := \mathbf{E}(\mu)\mathbf{V}\hat{\mathbf{x}}_c^{k-1}(\mu) + \delta t(\mathbf{f}(\mathbf{V}\hat{\mathbf{x}}_c^{k-1}(\mu), \mu) + \mathbf{b}(\mu)u^k) + \mathbf{d}^k(\mu) - \mathbf{A}_t(\mu)\mathbf{V}\hat{\mathbf{x}}_c^k(\mu)$  is the residual induced by the  $\mathbf{d}$ -corrected ROM (46), and  $\hat{\mathbf{y}}_c^k(\mu) := \hat{\mathbf{y}}_c^k(\mu) + (\mathbf{V}_{du}\hat{\mathbf{x}}_{du}^k(\mu))^T \mathbf{r}_c^k(\mu)$  is a corrected output of the  $\mathbf{d}$ -corrected ROM in (46).  $\bar{\Phi}^k(\mu)$ ,  $\mathbf{V}$ ,  $\mathbf{V}_{du}$ ,  $\hat{\mathbf{x}}_{du}^k(\mu)$  and  $\mathbf{r}_{du}^k(\mu)$  are defined as before.

The corrected output  $\bar{\mathbf{y}}_c^k(\mu) := \hat{\mathbf{y}}_c^k(\mu) + (\mathbf{V}_{du}\hat{\mathbf{x}}_{du}^k(\mu))^T \mathbf{r}_c^k(\mu)$  is defined a bit differently as in [35], where  $\hat{\mathbf{y}}_c^k(\mu) := \hat{\mathbf{y}}_c^k(\mu) - (\mathbf{V}_{du}\hat{\mathbf{x}}_{du}^k(\mu))^T \mathbf{r}_c^k(\mu)$ . Its corresponding dual system in [35]:

$$\mathbf{A}_t(\mu)^T \mathbf{x}_{du}(\mu) = -\mathbf{c}^T(\mu)$$

is also slightly different from that in (25). Please also refer to Remark 7. However, derivation of the error estimator is very similar as that in [35] and is not repeated here.

When hyperreduction is considered, the ROM (3) becomes

$$\begin{aligned} \frac{d}{dt} \hat{\mathbf{E}}(\mu)\hat{\mathbf{x}}(\mu) &= \hat{\mathbf{A}}(\mu)\hat{\mathbf{x}}(\mu) + \mathbf{V}^T \mathcal{I}_f(\mu) + \hat{\mathbf{b}}(\mu)u(t), & \hat{\mathbf{x}}(0) &= \mathbf{V}^T \mathbf{x}_0, \\ \hat{\mathbf{y}}(\mu) &= \hat{\mathbf{c}}(\mu)\hat{\mathbf{x}}(\mu), \end{aligned} \quad (48)$$

**Algorithm 1** The weak greedy algorithm

Input: the FOM, a training set  $\Xi$  composed of parameter samples taken from the parameter domain  $\mathcal{P}$ , error tolerance  $\text{tol} < 1$ ,  $\Delta(\mu)$  to estimate the error.

Output: Projection matrix  $\mathbf{V}$ .

1. Choose initial parameter  $\mu^* \in \Xi$ .  $\mathbf{V} \leftarrow \emptyset$ ,  $\varepsilon = 1$ .
2. While  $\varepsilon > \text{tol}$
3. Compute the snapshot(s)  $\mathbf{x}(\mu^*)$  by solving the FOM at  $\mu = \mu^*$ .
4. Update  $\mathbf{V}$  by  $\mathbf{V} = \text{orth}\{\mathbf{V}, \mathbf{x}(\mu^*)\}$ , (e.g., using the modified Gram-Schmidt process with deflation.)
5. Compute  $\mu^*$  such that  $\mu^* = \arg \max_{\mu \in \Xi} \Delta(\mu)$ .
6.  $\varepsilon = \Delta(\mu^*)$ .
7. End.

where  $\mathcal{I}_f(\mu)$  approximates  $\mathbf{f}(\mathbf{V}\hat{\mathbf{x}}, \mu)$  via hyperreduction. The corresponding  $\mathbf{d}$ -corrected ROM is

$$\begin{aligned} \hat{\mathbf{A}}_t(\mu)\hat{\mathbf{x}}_c^k(\mu) &= \mathbf{E}(\mu)\hat{\mathbf{x}}_c^{k-1}(\mu) + \delta t(\mathbf{V}^T\mathcal{I}_f^k(\mu) + \hat{\mathbf{b}}(\mu)u^k) + \hat{\mathbf{d}}^k(\mu), & \hat{\mathbf{x}}_c(0) &= \mathbf{V}^T\mathbf{x}_0, \\ \hat{\mathbf{y}}_c^k(\mu) &= \hat{\mathbf{c}}\hat{\mathbf{x}}_c^k, \end{aligned} \quad (49)$$

which is the  $\mathbf{d}$ -corrected hyperreduced ROM. Error estimation for the output error of the ROM (48) is stated as

$$|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)| \approx \bar{\Phi}^k(\mu)(\|\mathbf{r}_{c,\mathcal{I}}^k(\mu)\| + \epsilon_{\mathcal{I}}^k(\mu)) + \|\hat{\mathbf{y}}_c^k(\mu) - \hat{\mathbf{y}}^k(\mu)\|, \quad (50)$$

where  $\mathbf{r}_{c,\mathcal{I}}^k(\mu) := \mathbf{E}(\mu)\mathbf{V}\hat{\mathbf{x}}_c^{k-1}(\mu) + \delta t(\mathcal{I}_f^k(\mu) + \mathbf{b}(\mu)u^k) + \mathbf{d}^k(\mu) - \mathbf{A}_t(\mu)\mathbf{V}\hat{\mathbf{x}}_c^k(\mu)$  is the residual induced by the  $\mathbf{d}$ -corrected ROM (49), and  $\epsilon_{\mathcal{I}}^k(\mu)$  is the hyperreduction error defined as before.

Now we come to the problem of accurately learning  $\mathbf{d}(\mu)$ , so that (47) gives an accurate error estimation for ROMs solved with black-box solvers. In [35], we have used proper orthogonal decomposition (POD) combined with radial basis function (RBF) interpolation or with feed forward neural network (FFNN) to learn  $\mathbf{d}(\mu)$ . POD is first used to project  $\mathbf{d}(\mu) \in \mathbb{R}^N$  onto a lower-dimensional subspace. RBF or FFNN is then used to learn the projected short vector  $\hat{\mathbf{d}}(\mu) \in \mathbb{R}^{n_d}$ ,  $n_d \ll N$ , where  $\mathbf{d}(\mu) \approx \mathbf{V}_d\hat{\mathbf{d}}(\mu)$ ,  $\mathbf{V}_d \in \mathbb{R}^{N \times n_d}$  is computed from a two-stage singular value decomposition (SVD) of the snapshot matrix  $\mathbf{D} := [\mathbf{d}^0(\mu_1), \dots, \mathbf{d}^K(\mu_1), \dots, \mathbf{d}^0(\mu_s), \dots, \mathbf{d}^K(\mu_s)]$ . Each  $\mathbf{d}^i(\mu_j)$  is the defect vector evaluated at time instance  $t_i$  and parameter sample  $\mu_j$ . All details can be found in [35].

*Remark 12* While the new error estimator is based on our earlier proposed output error estimator in [31], the idea can be directly applied to derive a posteriori state error estimators (bounds).

### Multi-fidelity error estimation

This section briefly reviews our recent multi-fidelity error estimation used for accelerating the weak greedy algorithm. Weak greedy algorithms are often used to iteratively construct the reduced basis ( $\mathbf{V}$ ) for MOR of parametric steady systems. A sketch of the algorithm is given as Algorithm 1.

Some key points for the greedy algorithm to converge fast are: a properly chosen training set  $\Xi$ , an efficient and fast-to-compute error estimator (bound)  $\Delta(\mu)$ . For some complex problems, although the cardinality of the training set  $\Xi$  is not large, computing  $\Delta(\mu)$

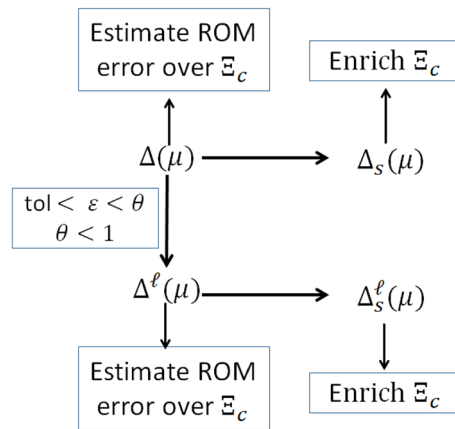
over  $\Xi$  at each iteration is slow. In [36], we propose the concept of multi-fidelity error estimation to accelerate the greedy iteration.

We start with a rough training set  $\Xi_c$  with even smaller cardinality, i.e.  $|\Xi_c| \leq |\Xi|$ , and try to evaluate  $\Delta(\mu)$  only over  $\Xi_c$  at each greedy iteration. At the same time, a surrogate estimator is constructed based on the already available values of  $\Delta(\mu)$  over  $\Xi_c$ . This surrogate is supposed to be more cheaply computed than  $\Delta(\mu)$ , so that it can be fast evaluated over a fine training set  $\Xi_f$  with much larger cardinality than  $|\Xi|$ . Using the results of the surrogate estimator over  $\Xi_f$ , we enrich  $\Xi_c$  with the parameter sample selected by the surrogate. The selected parameter sample corresponds to the largest value of the surrogate. The parameter sample that corresponds to the smallest value of  $\Delta(\mu)$  over  $\Xi_c$  is simultaneously removed from  $\Xi_c$ . This way, we can always keep  $\Xi_c$  small over iterations, but  $\Xi_c$  is kept being updated to only contain those *important* parameter samples. In the greedy process, those samples correspond to large ROM errors and are good candidates for greedy parameter selection in the next iterations.

This process of using a surrogate estimator in the greedy algorithm was originally proposed in [55] for time-evolution nonlinear systems. In [36], we define this as bi-fidelity error estimation, since both the original estimator  $\Delta(\mu)$  and a surrogate estimator are used for estimating the error in the greedy process. Based on that, we further propose multi-fidelity error estimation which depends on the structure of the original error estimator  $\Delta(\mu)$  [36]. Taking the output error estimator  $\Delta_{o_3}(\mu)$  as an example, two projection matrices  $\mathbf{V}$  and  $\mathbf{V}_r$  should be constructed in order to compute  $\Delta_{o_3}(\mu)$ . When we replace  $\Delta(\mu)$  in Algorithm 1 with  $\Delta_{o_3}(\mu)$ , we need to iteratively update both  $\mathbf{V}$  and  $\mathbf{V}_r$  with snapshots by solving the FOM in (6) at two greedily selected parameter samples twice. If at a certain stage, e.g., when the estimated ROM error is smaller than a small value  $\theta < 1$ , but is still larger than the error tolerance  $\text{tol}$ , we stop updating  $\mathbf{V}_r$ , then the FOM in (39) at one of the two selected parameter samples does not have to be solved. Consequently, we have saved runtime of solving a large FOM at the subsequent iterations. At the same time, the original  $\Delta_{o_3}(\mu)$  is degraded to a low fidelity error estimator  $\Delta_{o_3}^\ell(\mu)$ . The surrogate estimator is then constructed based on this low fidelity estimator at the latter stage of the greedy process. Finally, we have employed the original estimator  $\Delta_{o_3}(\mu)$ , a low-fidelity estimator  $\Delta_{o_3}^\ell(\mu)$ , and their respective surrogates in the whole greedy process. We call this multi-fidelity error estimation. We sketch this concept in Fig. 1. It is shown in [36] that the greedy process employing multi-fidelity error estimation is much faster than the standard weak greedy algorithm for some large-scale time-delay systems with hundreds of delays.

The error estimators presented in the previous sections have been numerically compared in the individual papers. For an overview, we list them as below. Here, the sections are those in this survey where the corresponding error estimators are reviewed.

- In [29, 30], the error estimator proposed there (“[Error estimators for time-evolution systems](#)” section) is numerically compared with the error bound in [19, 27] (“[Error bounds for time-evolution systems](#)” section) for parametric time-evolution systems.
- In [31], the error estimator with corrected output (“[Error estimators for time-evolution systems](#)” section) is numerically compared with the error estimator in [29, 30] for parametric time-evolution systems.



**Fig. 1** The concept of multi-fidelity error estimation in a greedy process, where  $\Delta(\mu)$  represents any original error estimator,  $\Delta^\ell(\mu)$  is a low-fidelity error estimator when we stop updating partial information of  $\Delta(\mu)$ , and  $\Delta_s(\mu)$  is a surrogate of  $\Delta(\mu)$ . Likewise,  $\Delta_s^\ell(\mu)$  is a surrogate of  $\Delta^\ell(\mu)$ .  $\text{tol}$  and  $\varepsilon$  are defined in Algorithm 1,  $\text{tol} < \theta < 1$  is a user-defined small value

- In [38], the proposed state error estimator (“[Error estimators for linear steady systems](#)” section) is compared with the state error bound (“[Error bounds for steady systems](#)” section) for parametric steady systems from computational electromagnetics.
- In [33], a newly proposed output error estimator  $\Delta_{o_1}(\mu)$  (“[Error estimators for linear steady systems](#)” section) is compared with the output error bound (22) in [32] “[Error bounds for steady systems](#)” section) for parametric linear steady systems. It is also compared with an existing randomized error estimator from [54].
- In [34], some more output error estimators are proposed and compared with each other; they are also compared with the output error estimator  $\Delta_{o_1}(\mu)$  proposed in [33] (“[Error estimators for linear steady systems](#)” section).
- In [35], a new error estimator (“[Error estimator for ROMs solved with any black-box time-integration solver](#)” section) which is applicable to the situation where both the FOM and the ROM are solved by a black-box solver, is compared with the output error estimator in [31] for parametric nonlinear time-evolution systems.
- In [36], the multi-fidelity error estimation (“[Multi-fidelity error estimation](#)” section) is numerically compared with the standard greedy process with only a single high-fidelity error estimator for time-delay systems with more than one hundred delays.

**Inf-sup-constant-free error estimator for time-evolution systems**

While the error estimators for time-evolution systems described in “[Error estimators for time-evolution systems](#)” section are accurate, their computation involves the quantities  $\Phi^k(\mu)$  and  $\tilde{\Phi}^k(\mu)$  for which the term  $\|\mathbf{A}_t(\mu)^{-1}\| = \frac{1}{\sigma_{\min}(\mathbf{A}_t(\mu))}$  needs to be evaluated for every  $\mu$ , where  $\sigma_{\min}(\mathbf{A}_t(\mu))$  is the smallest singular value of the matrix  $\mathbf{A}_t(\mu)$ . In function space,  $\sigma_{\min}(\mathbf{A}_t(\mu))$  corresponds to the *inf-sup*-constant of a linear operator [56]. This poses two challenges. Firstly, the complexity of computing the smallest singular value is at least linearly dependent on  $N$  for each parameter sample. When the number of parameter samples is high (typical for problems with several parameters or parameters having a wide range), this can lead to significant increase of the offline computational cost. Secondly, for some applications the matrix  $\mathbf{A}_t(\mu)$  could be poorly conditioned, leading to  $\sigma_{\min}(\mathbf{A}_t(\mu))$

close to zero, which could lead to blow up of the estimated error. While methods exist in the literature [56–58] to address the increased computational cost, these approaches are somewhat heuristic and a careful tuning of the involved parameters needs to be done to achieve good results. In the following theorem, we derive a new output error estimator applicable to time-evolution systems avoiding the inf-sup constant.

*Remark 13* For the sake of exposition, we derive the new inf-sup-constant-free error estimator based on the derivation of the output error estimator in Theorem 5. But, a similar process can be repeated to derive inf-sup-constant-free versions of the error estimators presented in (34), (35) and (36). Furthermore, a straightforward extension of the inf-sup-constant-free output error estimator is applicable to the output error estimator in (47) and (50) which deals with the case of black-box time-integration solvers.

**Theorem 11** (Primal-dual inf-sup-constant-free output error estimator) *For the time discrete FOM (8) and the time-discrete ROM (9), assume the time step  $\delta t$  is constant, so that  $\mathbf{A}_t(\mu)$  does not change with time. Let all the assumptions in Theorem 5 be met, the output error  $e_o^k(\mu) = \mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)$  at the time instance  $t^k$  can be bounded as*

$$|e_o^k(\mu)| \leq \tilde{\rho}(\mu) \check{\Phi}^k(\mu) \|\mathbf{r}^k(\mu)\|, \quad k = 1, \dots, K,$$

where  $\tilde{\rho}^k(\mu) := \|\tilde{\mathbf{r}}^k(\mu)\|/\|\mathbf{r}^k(\mu)\|$  and  $\check{\Phi}^k(\mu) = \|\mathbf{e}_{du}\| + \|\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu)\|$  with  $\mathbf{e}_{du} := \mathbf{A}_t(\mu)^{-1}\mathbf{r}_{du}(\mu)$ . Here,  $\hat{\mathbf{x}}_{du}(\mu)$  and  $\mathbf{r}_{du}(\mu)$  are defined in (26) and (27), respectively.

*Proof* We start with the expression of (32) from Theorem 5 and write

$$\begin{aligned} |\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)| &= |[\mathbf{r}_{du}(\mu)]^T [\mathbf{A}_t(\mu)]^{-1} \tilde{\mathbf{r}}^k(\mu)| \\ &\leq \|([\mathbf{A}_t(\mu)]^{-T} \mathbf{r}_{du}(\mu))^T\| \|\tilde{\mathbf{r}}^k(\mu)\|. \end{aligned}$$

Since  $\mathbf{A}_t(\mu)$  does not depend on time, we can safely remove the superscript  $k$  from  $\mathbf{r}_{du}(\mu)$ . Unlike what is done in (32) we do not apply the matrix sub-multiplicative property in the second line for the term  $\|([\mathbf{A}_t(\mu)]^{-T} \mathbf{r}_{du}^k(\mu))\|$ . The expression  $[\mathbf{A}_t(\mu)]^{-T} \mathbf{r}_{du}(\mu) =: \mathbf{e}_{du}(\mu)$  is seen to be the solution of the linear system

$$[\mathbf{A}_t(\mu)]^T \mathbf{e}_{du}(\mu) = \mathbf{r}_{du}(\mu). \quad (51)$$

We call the above linear system the dual-residual system corresponding to the dual system (25). Using this dual-residual system and the expression  $\tilde{\mathbf{y}}(\mu) = \hat{\mathbf{y}}^k(\mu) + (\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu))^T \tilde{\mathbf{r}}^k(\mu)$  we write

$$\begin{aligned} |\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)| &\leq \|\mathbf{e}_{du}(\mu)^T\| \|\tilde{\mathbf{r}}^k(\mu)\| + |(\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu))^T \tilde{\mathbf{r}}^k(\mu)| \\ &\leq \|\mathbf{e}_{du}(\mu)\| \|\tilde{\mathbf{r}}^k(\mu)\| + \|\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu)\| \|\tilde{\mathbf{r}}^k(\mu)\| \\ &= (\|\mathbf{e}_{du}(\mu)\| + \|\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu)\|) \|\tilde{\mathbf{r}}^k(\mu)\|, \end{aligned}$$

where  $\hat{\mathbf{x}}_{du}(\mu)$  does not change with time, so that the superscript  $k$  is also removed. Defining  $\check{\Phi}(\mu) := \|\mathbf{e}_{du}(\mu)\| + \|\mathbf{V}_{du}\hat{\mathbf{x}}_{du}(\mu)\|$  and using  $\tilde{\rho}^k(\mu) = \|\tilde{\mathbf{r}}^k(\mu)\|/\|\mathbf{r}^k(\mu)\|$ , we get the desired error bound.  $\square$

Finally, we approximate the ratio  $\tilde{\rho}^k(\mu)$  with the quantity  $\rho(\mu)$  to obtain the inf-sup-constant-free output error estimator as

$$|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)| \approx \rho(\mu) \check{\Phi}(\mu) \|\mathbf{r}^k(\mu)\| =: \Delta_{\text{iscf}}(\mu), \quad (52)$$

---

**Algorithm 2** Simultaneous construction of the projection bases for the inf-sup-constant-free output error estimator applicable to time-evolution systems
 

---

Input: Dual system matrices  $\mathbf{A}_t(\mu)^T$ ,  $\mathbf{c}(\mu)^T$ , a training set  $\Xi$  composed of parameter samples taken from the parameter domain  $\mathcal{P}$ , error tolerance  $\text{tol} < 1$ .

Output: Projection matrices  $\mathbf{V}_{du}$  and  $\mathbf{V}_e$ .

1. Choose initial parameters  $\mu^* \in \Xi$  and  $\mu_e^* \in \Xi$  with  $\mu^* \neq \mu_e^*$ . Set  $\mathbf{V}_{du} \leftarrow \emptyset$ ,  $\mathbf{V}_e^0 \leftarrow \emptyset$ ,  $\mathbf{V}_e \leftarrow \emptyset$ ,  $\varepsilon = 1$ .

2. While  $\varepsilon > \text{tol}$

3. Compute the dual system snapshot(s)  $\mathbf{x}_{du}(\mu)$  by solving the FOM (25) at  $\mu = \mu^*$ .

4. Update  $\mathbf{V}_{du}$  by  $\mathbf{V}_{du} = \text{orth}\{\mathbf{V}_{du}, \mathbf{x}_{du}(\mu^*)\}$  (e.g., using the modified Gram-Schmidt process with deflation.)

5. Compute the dual system snapshot(s)  $\mathbf{x}_{du}(\mu)$  by solving the FOM (25) at  $\mu = \mu_e^*$ .

6. Update  $\mathbf{V}_e^0$  by  $\mathbf{V}_e^0 = \text{orth}\{\mathbf{V}_e^0, \mathbf{x}_{du}(\mu_e^*)\}$  (e.g., using the modified Gram-Schmidt process with deflation.)

7. Form  $\mathbf{V}_e$ :  $\mathbf{V}_e = \text{orth}\{\mathbf{V}_{du}, \mathbf{V}_e^0\}$ .

8. Compute  $\mu^*$  such that

$$\mu^* = \arg \max_{\mu \in \Xi} \|\mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\|$$

where  $\hat{\mathbf{e}}_{du}(\mu)$  is obtained by solving the ROM (53).

9. Compute  $\mu_e^*$  from  $\Xi$  following

$$\mu_e^* = \arg \max_{\mu \in \Xi} \|\mathbf{r}_{du}(\mu) - [\mathbf{A}_t(\mu)]^T \mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\|$$

and ensuring  $\mu_e^* \neq \mu^*$ .

10.  $\varepsilon = \|\mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu^*)\|$ .

11. End.

---

### Computational aspects

In (52), evaluating  $\check{\Phi}(\mu)$  involves determining  $\|\mathbf{e}_{du}(\mu)\|$  by solving the dual-residual system (51) for every parameter sample  $\mu$ . This step can be computationally expensive. To address this, we propose to obtain a ROM for (51) such that we can approximate  $\mathbf{e}_{du}(\mu) \approx \mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)$ . The ROM reads

$$[\hat{\mathbf{A}}_e(\mu)]^T \hat{\mathbf{e}}_{du}(\mu) = \hat{\mathbf{r}}_{du}(\mu), \quad (53)$$

where  $\hat{\mathbf{A}}_e(\mu) = \mathbf{V}_e^T \mathbf{A}_t(\mu) \mathbf{V}_e$ ,  $\hat{\mathbf{r}}_{du}(\mu) = \mathbf{V}_e^T \mathbf{r}_{du}(\mu)$ . The dual-residual  $\mathbf{r}_{du}(\mu)$  is the residual induced by the approximate solution  $\mathbf{V}_{du} \hat{\mathbf{x}}_{du}(\mu)$  computed from the dual ROM (26). We propose a greedy algorithm in which  $\mathbf{V}_e$  and the projection matrix  $\mathbf{V}_{du}$  for the dual system ROM (26) are constructed simultaneously. For an appropriately computed  $\mathbf{V}_e$ , we have  $\|\mathbf{e}_{du}(\mu)\| \approx \|\mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\|$  and hence the inf-sup-constant-free error estimator (52) can be further approximated as

$$|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu)| \approx \rho(\mu) \check{\Phi}_e^k(\mu) \|\mathbf{r}^k(\mu)\| =: \tilde{\Delta}_{\text{isfc}}(\mu), \quad (54)$$

with  $\check{\Phi}_e(\mu) := \|\mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\| + \|\mathbf{V}_{du} \hat{\mathbf{x}}_{du}(\mu)\|$ . Next, the greedy algorithm to simultaneously construct  $\mathbf{V}_{du}$  and  $\mathbf{V}_e$  is detailed.

### Simultaneous and greedy construction of $\mathbf{V}_{du}$ and $\mathbf{V}_e$

The greedy algorithm is sketched in Algorithm 2. The inputs to the algorithm are the system matrices corresponding to the dual system, viz.,  $\mathbf{A}_t(\mu)^T$ ,  $\mathbf{c}(\mu)^T$ , a properly chosen training set  $\Xi$  and a tolerance  $\text{tol}$ . The outputs resulting from the algorithm are the two projection bases  $\mathbf{V}_{du}$  and  $\mathbf{V}_e$  which are needed to evaluate  $\check{\Phi}_e(\mu)$  in the *inf-sup*-constant-

free error estimator (54). In Step 1, the initial greedy parameters  $\mu^*$  and  $\mu_e^*$  are initialized, ensuring that  $\mu^* \neq \mu_e^*$ . The projection matrices  $\mathbf{V}_{du}$ ,  $\mathbf{V}_e^0$  and  $\mathbf{V}_e$  are initialized as empty matrices. In Steps 3 and 5, the FOM (25) is evaluated at  $\mu^*$  and  $\mu_e^*$ , respectively. The resulting dual system snapshots are then used to update  $\mathbf{V}_{du}$  and  $\mathbf{V}_e^0$  in Steps 4 and 6, respectively, using e.g., the modified Gram-Schmidt process with deflation. Step 7 involves constructing the projection matrix  $\mathbf{V}_e$ . Following this, the ROM in (53) is solved to evaluate  $\|\mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\| \forall \mu \in \Xi$ , which is then used as an error estimator to choose the next greedy parameter  $\mu^*$  in Step 8. Furthermore, in Step 9, the norm of the residual  $\|\mathbf{r}_{du}(\mu) - [\mathbf{A}_t(\mu)]^T \mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\|$  induced by the ROM (53) of the dual-residual system is evaluated to determine the second greedy parameter  $\mu_e^*$  for the next iteration. In Step 10, the maximum estimated error at the current iteration is set to be the maximum estimated error in Step 8, i.e.,  $\epsilon = \|\mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\|$ .

*Remark 14* In Step 8, we have used the criterion  $\|\mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\|$  to select the parameter  $\mu^*$  for constructing  $\mathbf{V}_{du}$  for the ROM (26). Recalling the state error estimator (37) for steady parametric systems proposed in (37), it is easy to see that  $\|\mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\|$  is exactly the state error estimator for the state error  $\|\mathbf{x}_{du}(\mu) - \hat{\mathbf{x}}_{du}(\mu)\|$  of the ROM (26). We use this state error estimator to iteratively construct the projection matrix  $\mathbf{V}_{du}$  for the ROM (26). In order to evaluate the state error estimator, we also need to construct  $\mathbf{V}_e$ . In [38], we have explained how to construct  $\mathbf{V}_e$  in detail. In particular, a different criterion is used for greedy construction of  $\mathbf{V}_e$ , i.e. the norm  $\|\mathbf{r}_{du}(\mu) - [\mathbf{A}_t(\mu)]^T \mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)\|$  to avoid  $\mu^* = \mu_e^*$ . The vector  $\mathbf{r}_{du}(\mu) - [\mathbf{A}_t(\mu)]^T \mathbf{V}_e \hat{\mathbf{e}}_{du}(\mu)$  is nothing but the residual vector induced by the ROM (53) of the dual-residual system (51).

To obtain the ROM (3) corresponding to the FOM (2), we apply the adaptive POD-Greedy algorithm [31] to construct the projection matrix  $\mathbf{V}$ . As the first test case (TC1), we apply the adaptive POD-Greedy algorithm which uses the output error estimator presented in Theorem 5. For the second test case (TC2), we apply the adaptive POD-Greedy algorithm with the new *inf-sup*-constant-free error estimator  $\tilde{\Delta}_{\text{iscf}}(\mu)$  (54). To compute the projection bases for evaluating (54), we make use of Algorithm 2. We first run Algorithm 2 to obtain the projection matrices  $\mathbf{V}_{du}$ ,  $\mathbf{V}_e$ , as well as the reduced quantities  $\hat{\mathbf{x}}_{du}(\mu)$ ,  $\hat{\mathbf{e}}_{du}(\mu)$  corresponding to  $\mathbf{V}_{du}$ ,  $\mathbf{V}_e$ . During each iteration of the POD-Greedy algorithm, those quantities are then used to compute the output error estimator  $\tilde{\Delta}_{\text{iscf}}(\mu)$  (54).

### Numerical examples

Next, we illustrate the benefits of using the *inf-sup*-constant-free error estimator  $\tilde{\Delta}_{\text{iscf}}(\mu)$  in (54) with two numerical examples: the Burgers' equation and the FitzHugh–Nagumo equations. It is demonstrated that, firstly, the *inf-sup*-constant-free error estimator offers accurate performance when used in the POD-greedy algorithm to construct  $\mathbf{V}$ . Secondly, the new approach yields a significant reduction of the offline computational costs by avoiding solving several large-scale eigenvalue problems for obtaining the *inf-sup* constant.

For the adaptive POD-Greedy greedy algorithm, we plot the maximum estimated errors computed using the respective error estimators for TC1 and TC2 over the training set  $\Xi$  at every iteration. We define this as:

$$\epsilon_{\max} := \max_{\mu \in \Xi} \Delta(\mu),$$

where  $\Delta(\mu)$  is either (33) in case of TC1 or (54) in case of TC2.



### 1-D Burgers' equation

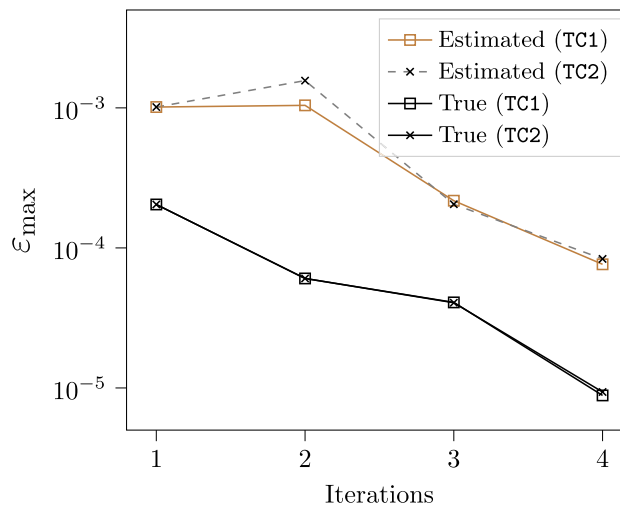
The viscous Burgers' equation defined in the 1-D domain  $\Omega := [0, 1]$  is given by

$$\begin{aligned} \frac{\partial v}{\partial t} + v \frac{\partial v}{\partial z} &= \mu \frac{\partial^2 v}{\partial z^2}, \\ v(z, 0) &= \sin(2\pi z), \\ v(0, t) = v(1, t) &= 0 \end{aligned} \tag{55}$$

with  $v := v(z, t) \in \mathbb{R}$  denoting the state variable,  $z \in \Omega$  is the spatial variable and the time variable  $t \in [0, 2]$ . We spatially discretize (55) with the finite difference method. The mesh size is  $\Delta z = 0.001$ , which results in a discretized FOM of dimension  $N = 4000$ . As the variable parameter, we consider the viscosity  $\mu \in \mathcal{P} := [0.005, 1]$ . The output variable of interest is the value of the state at the node just before the right boundary. The ROM tolerance is set to be  $\text{tol} = 1 \times 10^{-4}$ . We generate 100 sample points in  $\mathcal{P}$  using `np.logspace` in python, out of which 80 randomly chosen samples constitute the training set.

For TC1, we first use the standard greedy Algorithm 1 to compute the projection basis  $V_{du}$  for the ROM of the dual system. The error estimator used in Algorithm 1 is the state error bound (17) so that the *inf-sup* constants  $\|[\mathbf{A}_t(\mu)]^{-1}\| \forall \mu \in \Xi$  are pre-computed before starting the greedy iteration. These are then used to evaluate the output error estimator (33) during the POD-greedy algorithm for constructing  $\mathbf{V}$ . The greedy Algorithm 1 for constructing  $V_{du}$  converges in 1.1 s. However, computing the *inf-sup* constants took 164.8 s. For solving the eigenvalue problem at every parameter, we make use of the `scipy` library for Python. The POD-greedy needs 255.7 s to converge, with the ROM dimension  $n = 4$ .

In the case of TC2, Algorithm 2 is first used to compute the projection bases  $V_{du}$  and  $V_e$  simultaneously. This requires a total time of 0.98 s. The POD-greedy algorithm using the *inf-sup*-constant-free error estimator takes the same 255.7 s to converge. The final ROM dimension is also  $n = 4$ . Convergence of the POD-greedy algorithm for ROMs generation in the case of TC1 and TC2 is plotted in Fig. 2. It is clear that using the *inf-sup*-constant-



**Fig. 2** 1-D Burgers' equation: error (estimator) decay for TC1 and TC2

free error estimator results in little loss of accuracy in the error estimation, while speeding up the offline basis generation by  $1.6\times$ .

### 2-D Burgers' equation

We next consider the 2-D coupled Burgers' equation in the square domain  $\Omega := [0, 2] \times [0, 2]$ . The governing equations are as follows:

$$\frac{\partial v_1}{\partial t} + v_1 \frac{\partial v_1}{\partial z_1} + v_2 \frac{\partial v_1}{\partial z_2} = \mu \left( \frac{\partial^2 v_1}{\partial z_1^2} + \frac{\partial^2 v_1}{\partial z_2^2} \right), \quad (56a)$$

$$\frac{\partial v_2}{\partial t} + v_1 \frac{\partial v_2}{\partial z_1} + v_2 \frac{\partial v_2}{\partial z_2} = \mu \left( \frac{\partial^2 v_2}{\partial z_1^2} + \frac{\partial^2 v_2}{\partial z_2^2} \right). \quad (56b)$$

We impose the following Dirichlet boundary conditions:

$$\begin{aligned} v_1(0, z_2, t) &= 0, & v_1(2, z_2, t) &= 0; \\ v_1(z_1, 0, t) &= 0, & v_1(z_1, 2, t) &= 0; \\ v_2(0, z_2, t) &= 0, & v_2(2, z_2, t) &= 0; \\ v_2(z_1, 0, t) &= 0, & v_2(z_1, 2, t) &= 0. \end{aligned}$$

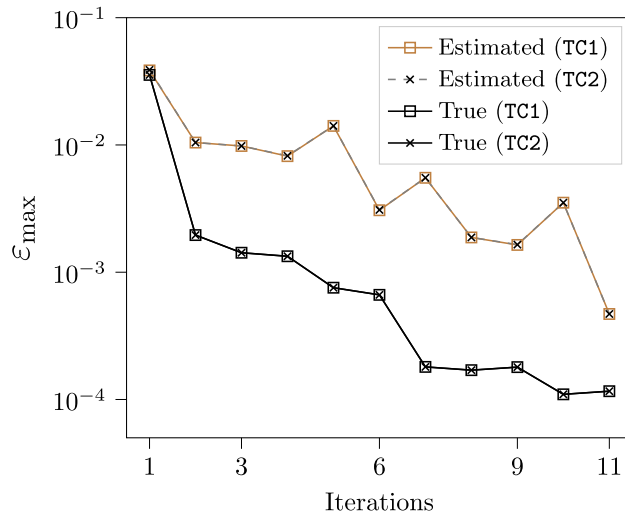
The initial conditions at time  $t = 0$  are given by

$$\begin{aligned} v_1(z_1, z_2, t = 0) &= \phi_1, \\ v_2(z_1, z_2, t = 0) &= \phi_2 \end{aligned}$$

with  $\phi_1 = \phi_2 = 10 e^{-(z_1-0.8)^2-(z_2-1.0)^2}$ . In (56),  $v_1(z_1, z_2, t)$  and  $v_2(z_1, z_2, t)$  denote the state variables and represent, respectively, the velocity components in the canonical  $x$  and  $y$  directions. Further,  $(z_1, z_2) \in \Omega$  and  $t \in [0, 1]$ . Similar to the 1-D case, we spatially discretize the 2-D Burgers' equation using the finite difference method with a step size  $\Delta z_1 = \Delta z_2 = 0.011$  (90 divisions along both  $x$ -axis and  $y$ -axis). This results in a coupled FOM of dimension  $N = 2 \cdot 8100$ . The viscosity  $\mu \in \mathcal{P} := [0.01, 1]$  is the parameter of interest. As the output, we take the mean of  $x$ -component velocities in the region  $\tilde{\Omega} := [0.7, 1.4] \times [0.7, 1.4]$ . A first-order implicit–explicit scheme with time step size  $\Delta t = 0.0025$  is used. The ROM tolerance is  $\text{tol} = 1 \times 10^{-3}$ . We generate 60 *logspace*-sampled (with `np.logspace` in python) points from  $\mathcal{P}$ , out of which 48 samples are randomly chosen to constitute the training set.

For TC1, the computation of the dual system projection matrix  $\mathbf{V}_{du}$  takes 36.5 s and computing the inf-sup-constant by solving an eigenvalue problem for every parameter in the training set  $\Xi$  took 3,380 seconds. Following this, Algorithm 1 is used to obtain the projection matrix  $\mathbf{V}$ . It requires 5,808 seconds to reach the desired tolerance of  $1 \times 10^{-3}$  in 11 iterations. The ROM dimension is  $n = 44$ .

The simultaneous generation of  $\mathbf{V}_{du}$  and  $\mathbf{V}_e$  with Algorithm 2 needs 63 s in case of TC2. The POD-Greedy algorithm using the inf-sup-constant-free error estimator takes 5,811 seconds, which is close to the time taken by the greedy algorithm in case of TC1. The resulting ROM has the same dimension as before, viz.,  $n = 44$ . The convergence of the estimated and true errors of TC1 and TC2 are plotted in Fig. 3. Evidently, the use of the inf-sup-constant-free output error estimator results in no loss of the accuracy of the



**Fig. 3** 2-D Burgers’ equation: error (estimator) decay for TC1 and TC2

estimated error. The overall speedup achieved in case of TC2 is 1.6-fold, compared to the offline time for TC1. However, since the system is of much larger dimension than the 1-D case, the offline time of computing the inf-sup-constants takes much longer time: 3,380 s. This certifies that using the inf-sup-constant-free error estimator has saved almost one hour of offline computational time.

**FitzHugh–Nagumo equations**

The FitzHugh–Nagumo system models the response of an excitable neuron or cell under an external stimulus. It finds applications in a variety of fields such as cardiac electrophysiology and brain modeling. The nonlinear coupled system of two partial differential equations defined in the domain  $\Omega := [0, L]$  is given below:

$$\begin{aligned}
 \epsilon \frac{\partial v_1(z, t)}{\partial t} &= \epsilon^2 \frac{\partial^2 v_1(z, t)}{\partial z^2} + f(v_1(z, t)) - v_2(z, t) + c, \\
 \frac{\partial v_2(z, t)}{\partial t} &= b v_1(z, t) - \gamma v_2(z, t) + c,
 \end{aligned}
 \tag{57}$$

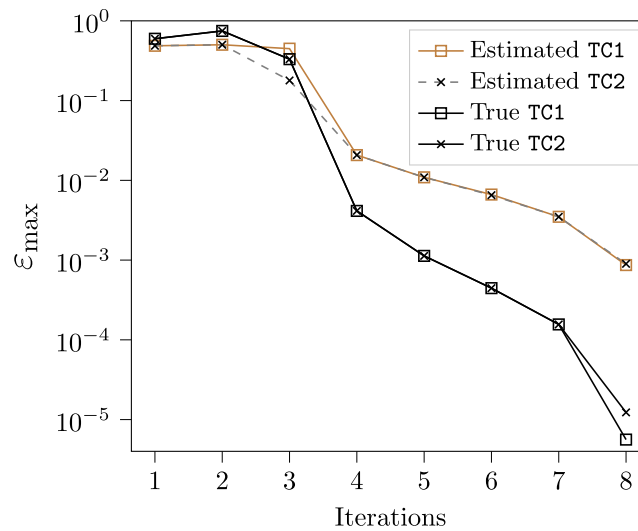
with boundary conditions

$$\frac{\partial}{\partial z} v_1(0, t) = -I_{\text{ext}}(t), \quad \frac{\partial}{\partial z} v_1(L, t) = 0,$$

and initial conditions

$$v_1(z, 0) = 0.001, \quad v_2(z, 0) = 0.001.$$

In the above equations,  $v_1(z, t)$  and  $v_2(z, t)$  represent the electric potential and the recovery rate of the potential, respectively. The spatial variable is denoted by  $z \in \Omega$  and the time  $t \in [0, 5]$ . The nonlinear term is represented by  $f(v_1(z, t)) := v_1(v_1 - 0.1)(1 - v_1)$ . The external stimulus is  $I_{\text{ext}}(t) = 50000t^3 e^{-15t}$ . The system has four free parameters  $\epsilon, c, b, \gamma$ . We fix  $b = 0.5$  and  $\gamma = 2$  while the two free parameters are  $\mu = [\epsilon, c] \in \mathcal{P} := [0.01, 0.04] \times [0.025, 0.075]$ . A finite difference scheme is employed to spatially discretize (57) with 2048 nodes used for each variable leading to a FOM of dimension  $N = 4096$ . We



**Fig. 4** FitzHugh–Nagumo equation: error (estimator) decay for TC1 and TC2

sample 100 parameters uniformly from the domain  $\mathcal{P}$  and randomly choose 70 samples from that to form the training set  $\Xi$ . The temporal discretization is done on a uniform grid with  $\delta t = 0.01$ . The output variables of interest are the values of the two state variables at the node next to the leftmost boundary. The greedy algorithm tolerance is set as  $\text{tol} = 1 \times 10^{-3}$ .

As done for the previous example, we first consider TC1 for the FitzHugh–Nagumo system. In this example, the greedy algorithm needs 1.6 s to obtain  $\mathbf{V}_{du}$  while computing the *inf-sup* constants takes 174.7 s. The POD-greedy algorithm based on the error estimator in Theorem 5 converges to the desired tolerance in 8 iterations, taking 291.2 s. The resulting ROM is of dimension  $n = 48$ .

Applying TC2 to this example, Algorithm 2 requires just 3.6 s to obtain  $\mathbf{V}_{du}$  and  $\mathbf{V}_d$ . The POD-greedy algorithm converges in 8 iterations and the runtime is 292 s. The ROM dimension is again  $n = 48$ . The convergence of the POD-greedy algorithm for ROM generation in the case of TC1 and TC2 is plotted in Fig. 4. Likewise, using the *inf-sup*-constant-free error estimator results in no loss of accuracy, but ends up with  $1.6\times$  speed-ups. For both examples, the *inf-sup* constants, i.e., the smallest singular values of  $\mathbf{A}(\mu)$  at all the training samples of  $\mu$  are close to 1, so that the effectivity of the *inf-sup*-constant-free error estimator in TC2 has almost no difference from that in TC1. This can be seen from Figs. 2–4.

## Conclusion

A posteriori error estimation is vital not only to quantify the accuracy of ROMs but also to construct ROMs of small dimension, in a computationally efficient and adaptive manner. In this review, we have presented a wide range of a posteriori error estimators applicable to (non)-linear parametric systems, covering both steady and time-dependent systems. Furthermore, we have also discussed multi-fidelity error estimators as a means to improve the computational efficiency of error estimation. As a novel contribution, we have introduced an *inf-sup*-constant-free output error estimator that is applicable to nonlinear time-dependent systems. This new error estimator is attractive for its improved efficiency

and also its ability to be applicable to systems with a potentially ill-conditioned left hand side matrix, e.g.  $\mathbf{A}_t(\mu)$  with smallest singular values being close to zeros. Results on three numerical examples were used to illustrate the reduced computational costs offered by the *inf-sup*-constant-free output error estimator, which is achieved with smaller effort but with little loss of accuracy. Going ahead, we envisage an important potential for accurate error estimation in applications such as digital twins where model updates can be done *on-the-fly* based on the accuracy quantified by error estimators.

#### Acknowledgements

Not applicable.

#### Author contributions

LF has written the paper till “Multi-fidelity error estimation” section and modified “Inf-sup-constant-free error estimator for time-evolution systems” and “Conclusion” sections; SC has written “Inf-sup-constant-free error estimator for time-evolution systems” and “Conclusion” sections and generated the numerical results; PB has given detailed comments all through the paper, including important comments on the theorems.

#### Funding

Open Access funding enabled and organized by Projekt DEAL.

#### Availability of data and materials

Not applicable.

#### Declarations

##### Ethics approval and consent to participate

Not applicable.

##### Consent for publication

Not applicable.

##### Competing interests

The authors declare no competing interests.

##### Code availability

The code will be made open access once the work is accepted for publication.

## Appendix: Proof of Theorem 2

*Proof* To simplify the proof, we first define an auxiliary dual solution  $\mathbf{x}_{du,L}^k(\mu) := \mathbf{x}_{du}^{K-L+k-1}(\mu)$ ,  $\forall L < K$  which is a time-shift of  $\mathbf{x}_{du}^k(\mu)$ , then it is easy to see that  $\mathbf{x}_{du,L}^k(\mu)$  satisfies the following dual system:

$$\begin{aligned} \mathbf{A}_t(\mu)^T \mathbf{x}_{du,L}^k(\mu) &= \mathbf{E}(\mu)^T \mathbf{x}_{du,L}^{k+1}(\mu), \\ \mathbf{E}(\mu)^T \mathbf{x}_{du,L}^{L+1}(\mu) &= \mathbf{c}(\mu)^T, k = L, \dots, 0, \forall L < K. \end{aligned} \quad (58)$$

The final condition of  $\mathbf{x}_{du,L}^k(\mu)$  is at time  $t_{L+1}$ , and coincides with the final condition of  $\mathbf{x}_{du}^K(\mu)$  at time  $t_K$ .

Multiplying  $\mathbf{e}^j(\mu) := (\mathbf{x}^j(\mu) - \tilde{\mathbf{x}}^j(\mu))$  from the left on both sides of the first equation in (58), and recalling  $\mathbf{A}_t(\mu) = \mathbf{E}(\mu) - \delta t \mathbf{A}(\mu)$ , we obtain

$$[\mathbf{e}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{x}_{du,L}^j(\mu) - \delta t [\mathbf{e}^j(\mu)]^T \mathbf{A}(\mu)^T \mathbf{x}_{du,L}^j(\mu) = [\mathbf{e}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{x}_{du,L}^{j+1}(\mu).$$

Then we have

$$[\mathbf{e}^j(\mu)]^T \mathbf{E}(\mu)^T [\mathbf{x}_{du,L}^j(\mu) - \mathbf{x}_{du,L}^{j+1}(\mu)] - \delta t [\mathbf{e}^j(\mu)]^T \mathbf{A}(\mu)^T \mathbf{x}_{du,L}^j(\mu) = 0.$$

Summing the above equation from  $j = 0$  to  $j = L$ , we get

$$\sum_{j=0}^L [\mathbf{e}^j(\mu)]^T \mathbf{E}(\mu)^T [\mathbf{x}_{du,L}^j(\mu) - \mathbf{x}_{du,L}^{j+1}(\mu)] - \sum_{j=0}^L \delta t [\mathbf{e}^j(\mu)]^T \mathbf{A}(\mu)^T \mathbf{x}_{du,L}^j(\mu) = 0.$$

Reformulating the first sum and using the assumption  $\mathbf{e}^0(\mu) = 0$  lead to

$$\begin{aligned} & \sum_{j=0}^{L-1} [\mathbf{e}^{j+1}(\mu) - \mathbf{e}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{x}_{du,L}^{j+1}(\mu) - [\mathbf{e}^L(\mu)]^T \mathbf{E}(\mu)^T \mathbf{x}_{du,L}^{L+1}(\mu) \\ & - \sum_{j=1}^L \delta t [\mathbf{e}^j(\mu)]^T \mathbf{A}(\mu)^T \mathbf{x}_{du,L}^j(\mu) = 0. \end{aligned}$$

From the second equation in (58), we see that  $\mathbf{E}(\mu)^T \mathbf{x}_{du,L}^{L+1}(\mu)$  is actually the final condition at time  $t_{L+1}$ . Therefore,  $\mathbf{E}(\mu)^T \mathbf{x}_{du,L}^{L+1}(\mu) = \mathbf{c}(\mu)^T$ . Replacing  $\mathbf{E}(\mu)^T \mathbf{x}_{du,L}^{L+1}(\mu)$  in the above equation with  $\mathbf{c}(\mu)^T$ , we get

$$\begin{aligned} & \sum_{j=0}^{L-1} [\mathbf{e}^{j+1}(\mu) - \mathbf{e}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{x}_{du,L}^{j+1}(\mu) - [\mathbf{e}^L(\mu)]^T \mathbf{c}(\mu)^T \\ & - \sum_{j=1}^L \delta t [\mathbf{e}^j(\mu)]^T \mathbf{A}(\mu)^T \mathbf{x}_{du,L}^j(\mu) = 0. \end{aligned} \quad (59)$$

Now from the primal system (8) and its ROM (9), we can get the error system,

$$\mathbf{E}(\mu) \mathbf{e}^{j+1}(\mu) - \delta t \mathbf{A}(\mu) \mathbf{e}^{j+1}(\mu) = \mathbf{E}(\mu) \mathbf{e}^j(\mu) + \mathbf{r}^{j+1}(\mu),$$

where we have used  $\mathbf{A}_t(\mu) = \mathbf{E}(\mu) - \delta t \mathbf{A}(\mu)$  and  $\mathbf{r}^{j+1}(\mu) = \mathbf{E}(\mu) \tilde{\mathbf{x}}^j(\mu) + \delta t \mathbf{b}(\mu) w^{j+1} - \mathbf{A}_t(\mu) \tilde{\mathbf{x}}^{j+1}(\mu)$  for linear systems. Multiplying  $[\mathbf{x}_{du,L}^{j+1}(\mu)]^T$  from the left on both sides of the above equation, we get,

$$\begin{aligned} & [\mathbf{x}_{du,L}^{j+1}(\mu)]^T \mathbf{E}(\mu) \mathbf{e}^{j+1}(\mu) - [\mathbf{x}_{du,L}^{j+1}(\mu)]^T \delta t \mathbf{A}(\mu) \mathbf{e}^{j+1}(\mu) \\ & = [\mathbf{x}_{du,L}^{j+1}(\mu)]^T \mathbf{E}(\mu) \mathbf{e}^j(\mu) + [\mathbf{x}_{du,L}^{j+1}(\mu)]^T \mathbf{r}^{j+1}(\mu). \end{aligned}$$

Summing up both sides of the above equation from  $j = 0$  to  $j = L - 1$ , we obtain

$$\begin{aligned} & \sum_{j=0}^{L-1} [\mathbf{x}_{du,L}^{j+1}(\mu)]^T \mathbf{E}(\mu) [\mathbf{e}^{j+1}(\mu) - \mathbf{e}^j(\mu)] - \sum_{j=0}^{L-1} [\mathbf{x}_{du,L}^{j+1}(\mu)]^T \delta t \mathbf{A}(\mu) \mathbf{e}^{j+1}(\mu) \\ & = \sum_{j=0}^{L-1} [\mathbf{x}_{du,L}^{j+1}(\mu)]^T \mathbf{r}^{j+1}(\mu) \end{aligned} \quad (60)$$

Comparing (59) with (60), we derive

$$\mathbf{c}(\mu) \mathbf{e}^L(\mu) = \sum_{j=1}^L [\mathbf{x}_{du,L}^j(\mu)]^T \mathbf{r}^j(\mu).$$

Recall  $\mathbf{x}_{du,L}^j(\mu) = \mathbf{x}_{du}^{K-L+j-1}(\mu)$ , then

$$\mathbf{c}(\mu) \mathbf{e}^L(\mu) = \sum_{j=1}^L [\mathbf{x}_{du}^{K-L+j-1}(\mu)]^T \mathbf{r}^j(\mu).$$

Taking  $L = k$  results in

$$\mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu) = \mathbf{c}(\mu) \mathbf{e}^k(\mu) = \sum_{j=1}^k [\mathbf{x}_{du}^{K-k+j-1}(\mu)]^T \mathbf{r}^j(\mu). \quad (61)$$

From the definition of the corrected output  $\hat{\mathbf{y}}_c^k(\mu)$ , we get

$$\mathbf{y}^k(\mu) - \hat{\mathbf{y}}_c^k(\mu) = \mathbf{y}^k(\mu) - \hat{\mathbf{y}}^k(\mu) - \sum_{j=1}^k [\tilde{\mathbf{x}}_{du}^{K-k+j-1}(\mu)]^T \mathbf{r}^j(\mu). \quad (62)$$

Combing (61) with (62) leads to

$$|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}_c^k(\mu)| = \left| \sum_{j=1}^k [\mathbf{e}_{du}^{K-k+j-1}(\mu)]^T \mathbf{r}^j(\mu) \right|,$$

where  $\mathbf{e}_{du}^{K-k+j-1}(\mu) = \mathbf{x}_{du}^{K-k+j-1}(\mu) - \tilde{\mathbf{x}}_{du}^{K-k+j-1}(\mu)$  is the dual error induced by the ROM (16) of the dual system. Employing the Cauchy-Schwarz inequality results in

$$|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}_c^k(\mu)| \leq \left( \sum_{j=1}^k \|\mathbf{r}^j(\mu)\|^2 \right)^{1/2} \left( \sum_{j=1}^k \|\mathbf{e}_{du}^{K-k+j-1}(\mu)\|^2 \right)^{1/2}. \quad (63)$$

Now we look for an upper bound of  $\left( \sum_{j=1}^k \|\mathbf{e}_{du}^{K-k+j-1}(\mu)\|^2 \right)^{1/2}$ , that is, an upper bound of  $\left( \sum_{j=0}^{k-1} \|\mathbf{e}_{du}^{K-k+j}(\mu)\|^2 \right)^{1/2}$ . From the dual system (15) and its ROM (16), we get the dual-error system,

$$\mathbf{E}(\mu)^T \mathbf{e}_{du}^j(\mu) - \delta t \mathbf{A}(\mu)^T \mathbf{e}_{du}^j(\mu) = \mathbf{E}(\mu)^T \mathbf{e}_{du}^{j+1}(\mu) + \mathbf{r}_{du}^j(\mu), j = K-1, \dots, 0,$$

where  $\mathbf{r}_{du}^j(\mu) := \mathbf{E}(\mu)^T \tilde{\mathbf{x}}_{du}^{j+1}(\mu) - \mathbf{A}_t(\mu)^T \tilde{\mathbf{x}}_{du}^j(\mu)$ , and  $\tilde{\mathbf{x}}_{du}(\mu) = \mathbf{V}_{du} \tilde{\mathbf{x}}_{du}^j(\mu)$ . Multiplying  $[\mathbf{e}_{du}^j(\mu)]^T$  from the left on both sides of the above error equation gives

$$\begin{aligned} & [\mathbf{e}_{du}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^j(\mu) - \delta t [\mathbf{e}_{du}^j(\mu)]^T \mathbf{A}(\mu)^T \mathbf{e}_{du}^j(\mu) \\ &= [\mathbf{e}_{du}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^{j+1}(\mu) + [\mathbf{e}_{du}^j(\mu)]^T \mathbf{r}_{du}^j(\mu), \\ &\leq ([\mathbf{e}_{du}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^j(\mu))^{1/2} ([\mathbf{e}_{du}^{j+1}(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^{j+1}(\mu))^{1/2} + \|\mathbf{e}_{du}^j(\mu)\| \|\mathbf{r}_{du}^j(\mu)\|, \\ &\leq \frac{1}{2} [\mathbf{e}_{du}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^j(\mu) + \frac{1}{2} [\mathbf{e}_{du}^{j+1}(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^{j+1}(\mu) + \|\mathbf{e}_{du}^j(\mu)\| \|\mathbf{r}_{du}^j(\mu)\|. \end{aligned}$$

Here, since  $\mathbf{E}(\mu)$  is symmetric positive definite,  $\mathbf{v}^T \mathbf{E}(\mu)^T \mathbf{w}$  defines an inner product  $(\mathbf{v}, \mathbf{w})_{\mathbf{E}}, \forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^N$ , so that the first inequality is derived from the Cauchy-Schwarz inequality. The second inequality has used  $2ab \leq a^2 + b^2 (a, b \in \mathbb{R})$ . We further apply the inequality  $2|a||b| \leq \frac{1}{\gamma^2} a^2 + \gamma^2 b^2 (\gamma \in \mathbb{R}_+)$  to the third term  $\|\mathbf{e}_{du}^j(\mu)\| \|\mathbf{r}_{du}^j(\mu)\|$  on the right-hand side of the above inequality with  $\gamma = (\delta t \lambda_{\min}^{\mathbf{A}}(\mu))^{1/2}$  to obtain

$$\begin{aligned} & [\mathbf{e}_{du}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^j(\mu) - \delta t [\mathbf{e}_{du}^j(\mu)]^T \mathbf{A}(\mu)^T \mathbf{e}_{du}^j(\mu) \\ &\leq \frac{1}{2} [\mathbf{e}_{du}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^j(\mu) + \frac{1}{2} [\mathbf{e}_{du}^{j+1}(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^{j+1}(\mu) \\ &\quad + \frac{1}{2\delta t \lambda_{\min}^{\mathbf{A}}(\mu)} \|\mathbf{r}_{du}^j(\mu)\|^2 + 1/2 \delta t \lambda_{\min}^{\mathbf{A}}(\mu) \|\mathbf{e}_{du}^j(\mu)\|^2, \end{aligned}$$

where  $\lambda_{\min}^{\mathbf{A}}(\mu) > 0$  is the smallest eigenvalue of  $-\mathbf{A}(\mu)$ . Since  $-\mathbf{A}(\mu)$  is symmetric positive definite, we have that  $-[\mathbf{e}_{du}^j(\mu)]^T \mathbf{A}(\mu)^T \mathbf{e}_{du}^j(\mu) \geq \lambda_{\min}^{\mathbf{A}}(\mu) \|\mathbf{e}_{du}^j(\mu)\|^2$ , then the above inequality gives

$$\begin{aligned} & [\mathbf{e}_{du}^j(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^j(\mu) - [\mathbf{e}_{du}^{j+1}(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^{j+1}(\mu) \\ &\leq \frac{1}{\delta t \lambda_{\min}^{\mathbf{A}}(\mu)} \|\mathbf{r}_{du}^j(\mu)\|^2 - \delta t \lambda_{\min}^{\mathbf{A}}(\mu) \|\mathbf{e}_{du}^j(\mu)\|^2. \end{aligned}$$

Summing up both sides of the above inequality from  $j = K-k$  to  $K-1$  results in

$$\begin{aligned} & [\mathbf{e}_{du}^{K-k}(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^{K-k}(\mu) - [\mathbf{e}_{du}^K(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^K(\mu) \\ &\leq \sum_{j=K-k}^{K-1} \frac{1}{\delta t \lambda_{\min}^{\mathbf{A}}(\mu)} \|\mathbf{r}_{du}^j(\mu)\|^2 - \sum_{j=K-k}^{K-1} \delta t \lambda_{\min}^{\mathbf{A}}(\mu) \|\mathbf{e}_{du}^j(\mu)\|^2. \end{aligned}$$

The above inequality is further reformulated into

$$\begin{aligned} & \sum_{j=K-k}^{K-1} \delta t \lambda_{\min}^{\mathbf{A}}(\mu) \|\mathbf{e}_{du}^j(\mu)\|^2 + [\mathbf{e}_{du}^{K-k}(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^{K-k}(\mu) \\ & \leq \sum_{j=K-k}^{K-1} \frac{1}{\delta t \lambda_{\min}^{\mathbf{A}}(\mu)} \|\mathbf{r}_{du}^j(\mu)\|^2 + [\mathbf{e}_{du}^K(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^K(\mu). \end{aligned}$$

With the assumption that  $\mathbf{E}(\mu)$  is symmetric positive definite, we can safely remove  $[\mathbf{e}_{du}^{K-k}(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^{K-k}(\mu) \geq 0$  from the above inequality and obtain

$$\sum_{j=K-k}^{K-1} \delta t \lambda_{\min}^{\mathbf{A}}(\mu) \|\mathbf{e}_{du}^j(\mu)\|^2 \leq \sum_{j=K-k}^{K-1} \frac{1}{\delta t \lambda_{\min}^{\mathbf{A}}(\mu)} \|\mathbf{r}_{du}^j(\mu)\|^2 + [\mathbf{e}_{du}^K(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^K(\mu).$$

Equivalently,

$$\sum_{j=0}^{k-1} \delta t \lambda_{\min}^{\mathbf{A}}(\mu) \|\mathbf{e}_{du}^{K-k+j}(\mu)\|^2 \leq \sum_{j=0}^{k-1} \frac{1}{\delta t \lambda_{\min}^{\mathbf{A}}(\mu)} \|\mathbf{r}_{du}^{K-k+j}(\mu)\|^2 + [\mathbf{e}_{du}^K(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^K(\mu). \tag{64}$$

From the final condition of the dual system (15) and its ROM (16), we obtain

$$\mathbf{E}(\mu)^T \mathbf{e}_{du}^K(\mu) = \mathbf{r}_{du}^K(\mu),$$

where  $\mathbf{r}_{du}^K(\mu) = \mathbf{c}^T(\mu) - \mathbf{E}(\mu)^T \tilde{\mathbf{x}}_{du}^K(\mu)$ , so that

$$\|\mathbf{e}_{du}^K(\mu)\|^2 \leq \|\mathbf{E}(\mu)^{-T}\| \|\mathbf{r}_{du}^K(\mu)\|.$$

Then

$$[\mathbf{e}_{du}^K(\mu)]^T \mathbf{E}(\mu)^T \mathbf{e}_{du}^K(\mu) \leq \|\mathbf{e}_{du}^K(\mu)\|^2 \|\mathbf{E}(\mu)^T\| \leq \|\mathbf{E}(\mu)^T\| \|\mathbf{E}(\mu)^{-T}\| \|\mathbf{r}_{du}^K(\mu)\|.$$

Inserting this inequality into (64), and noticing that  $\tilde{\sigma}(\mu) := \sigma_{\max} \mathbf{E}(\mu) / \sigma_{\min} \mathbf{E}(\mu) = \|\mathbf{E}(\mu)^T\| \|\mathbf{E}(\mu)^{-T}\|$ , we get

$$\sum_{j=0}^{k-1} \|\mathbf{e}_{du}^{K-k+j}(\mu)\|^2 \leq \sum_{j=0}^{k-1} \frac{1}{(\delta t)^2 (\lambda_{\min}^{\mathbf{A}}(\mu))^2} \|\mathbf{r}_{du}^{K-k+j}(\mu)\|^2 + \frac{\tilde{\sigma}(\mu)}{\delta t \lambda_{\min}^{\mathbf{A}}(\mu)} \|\mathbf{r}_{du}^K(\mu)\|, \tag{65}$$

where  $\sigma_{\max} \mathbf{E}(\mu)$  and  $\sigma_{\min} \mathbf{E}(\mu)$  are the largest and smallest singular values of  $\mathbf{E}(\mu)$ , respectively. Combining (65) with (63) results in

$$|\mathbf{y}^k(\mu) - \hat{\mathbf{y}}_c^k(\mu)| \leq \left( \sum_{j=1}^k \|\mathbf{r}^j(\mu)\|^2 \right)^{1/2} \Gamma(\mu) \left( \sum_{j=0}^{k-1} \|\mathbf{r}_{du}^{K-k+j}(\mu)\|^2 + \delta_{du}^K(\mu) \right)^{1/2},$$

where  $\Gamma(\mu) := \frac{1}{\delta t \lambda_{\min}^{\mathbf{A}}(\mu)}$  and  $\delta_{du}^K(\mu) := \frac{\tilde{\sigma}(\mu)}{\Gamma(\mu)} \|\mathbf{r}_{du}^K(\mu)\|$ . □

Received: 4 October 2023 Accepted: 7 February 2024

Published online: 11 March 2024

**References**

1. Benner P, Gugercin S, Willcox K. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* 2015;57(4):483–531. <https://doi.org/10.1137/130932715>.
2. Daniel L, Siong OC, Chay LS, Lee KH, White J. A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. *IEEE Trans Comput-Aided Design Integr Circuits Syst.* 2004;23(5):678–93.



3. Feng L, Benner P. Chap. 6: A robust algorithm for parametric model order reduction based on implicit moment matching. In: Quarteroni A, Rozza G, editors. *Reduced order methods for modeling and computational reduction*, vol. 9. MS&A Series. Berlin: Springer; 2014. p. 159–86. [https://doi.org/10.1007/978-3-319-02090-7\\_6](https://doi.org/10.1007/978-3-319-02090-7_6).
4. Baur U, Beattie CA, Benner P, Gugercin S. Interpolatory projection methods for parameterized model reduction. *SIAM J Sci Comput.* 2011;33(5):2489–518. <https://doi.org/10.1137/090776925>.
5. Amsallem D. Interpolation on manifolds of CFD-based fluid and finite element-based structural reduced-order models for on-line aerolastic predictions. Ph.D. Thesis, Stanford University; 2010.
6. Geuß M, Butnaru D, Peherstorfer B, Bungartz H, Lohmann B. Parametric model order reduction by sparse-grid-based interpolation on matrix manifolds for multidimensional parameter spaces. In: *Proceedings of the European control conference*, Strasbourg, France; 2014. p. 2727–32. <https://doi.org/10.1109/ECC.2014.6862414>.
7. Geuß M, Lohmann B. STABLE—a stability algorithm for parametric model reduction by matrix interpolation. *Math Comput Model Dyn Syst.* 2016;22(4):307–22. <https://doi.org/10.1080/13873954.2016.1198383>.
8. Kabir M, Khazaka R. Parametric macromodeling of high-speed modulators from frequency-domain data using Loewner matrix based method. In: *IEEE MTT-S international microwave symposium digest*; 2013. p. 1–4.
9. Ionita AC, Antoulas AC. Data-driven parametrized model reduction in the Loewner framework. *SIAM J Sci Comput.* 2014;36(3):984–1007. <https://doi.org/10.1137/130914619>.
10. Xiao YQ, Grivet-Talocia S, Manfredi P, Khazaka R. A novel framework for parametric Loewner matrix interpolation. *IEEE Trans Compon Packag Manuf Technol.* 2019;9(12):2404–17. <https://doi.org/10.1109/TCPM.2019.2948802>.
11. Yue Y, Feng L, Benner P. Reduced-order modelling of parametric systems via interpolation of heterogeneous surrogates. *Adv Model Simul Eng Sci.* 2019;6:10. <https://doi.org/10.1186/s40323-019-0134-y>.
12. Bui-Thanh T, Willcox K, Ghattas O. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM J Sci Comput.* 2008;30:3270–88.
13. Tezaur R, As'ad F, Farhat C. Robust and globally efficient reduction of parametric, highly nonlinear computational models and real time online performance. *Comput Methods Appl Mech Eng.* 2022;399:115392.
14. He W, Avery P, Farhat C. In situ adaptive reduction of nonlinear multiscale structural dynamics models. *Int J Numer Methods Eng.* 2020;121(22):4971–88.
15. Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch Comput Methods Eng.* 2008;15(3):229–75. <https://doi.org/10.1007/s11831-008-9019-9>.
16. Haasdonk B, Ohlberger M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM Math Model Numer Anal.* 2008;42(2):277–302. <https://doi.org/10.1051/m2an:2008001>.
17. Quarteroni A, Manzoni A, Negri F. *Reduced basis methods for partial differential equations*, vol. 92. *La Matematica per il 3+2*. Cham: Springer; 2016. <https://doi.org/10.1007/978-3-319-15431-2>.
18. Hesthaven JS, Rozza G, Stamm B. *Certified reduced basis methods for parametrized partial differential equations*. Springer briefs in mathematics. Cham: Springer; 2016. <https://doi.org/10.1007/978-3-319-22470-1>.
19. Grepl M. *Reduced-basis approximation a posteriori error estimation for parabolic partial differential equations*. PhD thesis, Cambridge: Massachusetts Institute of Technology (MIT); 2005. <https://dspace.mit.edu/handle/1721.1/32387>.
20. Veroy K. *Reduced-basis methods applied to problems in elasticity: analysis and applications*. PhD thesis, Cambridge: Massachusetts Institute of Technology (MIT); 2003. <https://dspace.mit.edu/handle/1721.1/29583>.
21. Rovas DV. *Reduced-basis output bound methods for parametrized partial differential equations*. PhD thesis, Cambridge: Massachusetts Institute of Technology (MIT); 2003. <https://dspace.mit.edu/handle/1721.1/16956>.
22. Sen S. *Reduced basis approximation and a posteriori error estimation for non-coercive elliptic problems: applications to acoustics*. PhD thesis, Cambridge: Massachusetts Institute of Technology (MIT); 2007. <http://dspace.mit.edu/handle/1721.1/7582>.
23. Veroy K, Prud'Homme C, Rovas DV, Patera AT. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In: *16th AIAA computational fluid dynamics conference*, Orlando, United States; 2003. <https://hal.archives-ouvertes.fr/hal-01219051>.
24. Haasdonk B, Ohlberger M. Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition. *Math Comput Model Dyn Syst.* 2011;17(2):145–61. <https://doi.org/10.1080/13873954.2010.514703>.
25. Machiels L, Maday Y, Patera AT. Output bounds for reduced-order approximations of elliptic partial differential equations. *Comput Meth Appl Mech Eng.* 2001;190(26–27):3413–26. [https://doi.org/10.1016/S0045-7825\(00\)00275-9](https://doi.org/10.1016/S0045-7825(00)00275-9).
26. Rovas DV, Machiels L, Maday Y. Reduced-basis output bound methods for parabolic problems. *IMA J Numer Anal.* 2006;26(3):423–45.
27. Grepl MA, Patera AT. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM Math Model Numer Anal.* 2005;39(1):157–81. <https://doi.org/10.1051/m2an:2005006>.
28. Quarteroni A, Rozza G, Manzoni A. Certified reduced basis approximation for parametrized partial differential equations and applications. *J Math Ind.* 2011;1:3–44. <https://doi.org/10.1186/2190-5983-1-3>.
29. Zhang Y. *Model order reduction for parameterized nonlinear evolution equations*. Dissertation, Magdeburg: Otto-von-Guericke-Universität; 2016. <https://doi.org/10.25673/4434>.
30. Zhang Y, Feng L, Li S, Benner P. An efficient output error estimation for model order reduction of parametrized evolution equations. *SIAM J Sci Comput.* 2015;37(6):910–36. <https://doi.org/10.1137/140998603>.
31. Chellappa S, Feng L, Benner P. Adaptive basis construction and improved error estimation for parametric nonlinear dynamical systems. *Int J Numer Methods Eng.* 2020;121(23):5320–49. <https://doi.org/10.1002/nme.6462>.
32. Feng L, Antoulas AC, Benner P. Some a posteriori error bounds for reduced order modelling of (non-)parametrized linear systems. *ESAIM M2AN.* 2017;51(6):2127–58. <https://doi.org/10.1051/m2an/2017014>.
33. Feng L, Benner P. A new error estimator for reduced-order modeling of linear parametric systems. *IEEE Trans Microw Theory Tech.* 2019;67(12):4848–59. <https://doi.org/10.1109/TMTT.2019.2948858>.
34. Feng L, Benner P. On error estimation for reduced-order modeling of linear non-parametric and parametric systems. *ESAIM Math Model Numer Anal.* 2021;55(2):561–94. <https://doi.org/10.1051/m2an/2021001>.

35. Chellappa S, Feng L, Benner P. Accurate error estimation for model reduction of nonlinear dynamical systems via data-enhanced error closure. *Comput Methods Appl Mech Eng.* 2024;420:116712. <https://doi.org/10.1016/j.cma.2023.116712>.
36. Feng L, Lombardi L, Antonini G, Benner P. Multi-fidelity error estimation accelerates greedy model reduction of complex dynamical systems. *Int J Numer Methods Eng.* 2023;124(23):5312–33. <https://doi.org/10.1002/nme.7348>.
37. Chamoin L, Legoll F. An introductory review on a posteriori error estimation in finite element computations. *SIAM Rev.* 2023;65(4):963–1028.
38. Chellappa S, Feng L, Rubia V, Benner P. Inf-sup-constant-free state error estimator for model order reduction of parametric systems in electromagnetics. *IEEE Trans Microw Theory Tech.* 2023;71(11):4762–77. <https://doi.org/10.1109/TMTT.2023.3288642>.
39. Bonin T, Faßbender H, Soppa A, Zaeh M. A fully adaptive rational global Arnoldi method for the model-order reduction of second-order MIMO systems with proportional damping. *Math Comput Simul.* 2016;122:1–19. <https://doi.org/10.1016/j.matcom.2015.08.017>.
40. Grunert D, Fehr J, Haasdonk B. Well-scaled, a-posteriori error estimation for model order reduction of large second-order mechanical systems. *ZAMM J Appl Math Mech / Zeitschrift für Angewandte Mathematik und Mechanik.* 2019;100(8):201900186. <https://doi.org/10.1002/zamm.201900186>.
41. Chinesta F, Ammar A, Cueto E. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Arch Comput Methods Eng.* 2010;17(4):327–50.
42. Ladevèze P, Chamoin L. On the verification of model reduction methods based on the proper generalized decomposition. *Comput Methods Appl Mech Eng.* 2011;200(23–24):2032–47.
43. Alfaro I, González D, Zlotnik S, Diez P, Cueto E, Chinesta F. An error estimator for real-time simulators based on model order reduction. *Adv Model Simul Eng Sci.* 2015. <https://doi.org/10.1186/s40323-015-0050-8>.
44. Chamoin L, Pled F, Allier P-E, Ladevèze P. A posteriori error estimation and adaptive strategy for PGD model reduction applied to parametrized linear parabolic problems. *Comput Methods Appl Mech Eng.* 2017;327:118–46.
45. Gubisch M, Neitzel I, Volkwein S. A-posteriori error estimation of discrete POD models for PDE-constrained optimal control. In: Benner P, Ohlberger M, Patera A, Rozza G, Urban K, editors. *Model reduction of parametrized systems.* Cham: Springer; 2017. p. 213–34.
46. Dogančić B, Jokić M. Discretization and model reduction error estimation of interconnected dynamical systems. *IFAC PapersOnLine.* 2022;55(4):177–82. <https://doi.org/10.1016/j.ifacol.2022.06.029>.
47. Binder A, Jadhav O, Mehrmann V. Error analysis of a model order reduction framework for financial risk analysis. *Electron Trans Numer Anal.* 2022;55:469–507.
48. Ascher UM, Ruuth SJ, Wetton BTR. Implicit–explicit methods for time-dependent partial differential equations. *SIAM J Numer Anal.* 1995;32(3):797–823. <https://doi.org/10.1137/0732037>.
49. Barrault M, Maday Y, Nguyen NC, Patera AT. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *CR Acad Sci Paris.* 2004;339(9):667–72. <https://doi.org/10.1016/j.cma.2004.08.006>.
50. Chaturantabut S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Sci Comput.* 2010;32(5):2737–64. <https://doi.org/10.1137/090766498>.
51. Drohmann M, Haasdonk B, Ohlberger M. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM J Sci Comput.* 2012;34(2):937–69. <https://doi.org/10.1137/10081157X>.
52. Zhang Y, Feng L, Li S, Benner P. Accelerating PDE constrained optimization by the reduced basis method: application to batch chromatography. *Int J Numer Methods Eng.* 2015;104(11):983–1007. <https://doi.org/10.1002/nme.4950>.
53. Schmidt A, Wittwar D, Haasdonk B. Rigorous and effective a-posteriori error bounds for nonlinear problems—application to RB methods. *Adv Comput Math.* 2020;46(2):1–30. <https://doi.org/10.1007/s10444-020-09741-x>.
54. Smetana K, Zahm O, Patera AT. Randomized residual-based error estimators for parametrized equations. *SIAM J Sci Comput.* 2019;41(2):900–26. <https://doi.org/10.1137/18M120364X>.
55. Chellappa S, Feng L, Benner P. An adaptive sampling approach for the reduced basis method. In: *Realization and model reduction of dynamical systems—a Festschrift in honor of the 70th birthday of Thanos Antoulas.* Cham: Springer; 2022. p. 137–155. [https://doi.org/10.1007/978-3-030-95157-3\\_8](https://doi.org/10.1007/978-3-030-95157-3_8).
56. Huynh DBP, Rozza G, Sen S, Patera AT. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup lower bounds. *CR Acad Sci Paris.* 2007;345(8):473–8. <https://doi.org/10.1016/j.cma.2007.09.019>.
57. Manzoni A, Negri F. Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized PDEs. *Adv Comput Math.* 2015;41(5):1255–88. <https://doi.org/10.1007/s10444-015-9413-4>.
58. Patera AT, Rozza G. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations. Technical report; 2007. MIT Pappalardo graduate monographs in mechanical engineering

## Publisher’s Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.