

A POSTERIORI ERROR ESTIMATION FOR DEIM REDUCED NONLINEAR DYNAMICAL SYSTEMS*

D. WIRTZ[†], D. C. SORENSEN[‡], AND B. HAASDONK[†]

Abstract. In this work an efficient approach for a posteriori error estimation for POD-DEIM reduced nonlinear dynamical systems is introduced. The considered nonlinear systems may also include time- and parameter-affine linear terms as well as parametrically dependent inputs and outputs. The reduction process involves a Galerkin projection of the full system and approximation of the system's nonlinearity by the DEIM method [S. Chaturantabut and D. C. Sorensen, *SIAM J. Sci. Comput.*, 32 (2010), pp. 2737–2764]. The proposed a posteriori error estimator can be efficiently decomposed in an offline/online fashion and is obtained by a one-dimensional auxiliary ODE during reduced simulations. Key elements for efficient online computation are partial similarity transformations and matrix-DEIM approximations of the nonlinearity Jacobians. The theoretical results are illustrated by application to an unsteady Burgers equation and a cell apoptosis model.

Key words. model reduction, nonlinear dynamical systems, discrete empirical interpolation method, error estimation, offline/online decomposition, Jacobian approximation, partial similarity transform

AMS subject classifications. 78M34, 65M15, 34C99, 65D15, 33F05, 26B10

DOI. 10.1137/120899042

1. Introduction. Model order reduction (MOR) has become an important field of modern scientific computing, as it can dramatically reduce simulation costs while preserving or approximating behavior of the full scale models. This cost reduction is accomplished by replacing a high-dimensional full order dynamical system with one of much lower dimension and complexity which produces nearly the same response as the full system. A mechanism for a posteriori error estimation is highly desirable for evaluating the accuracy of the approximation produced by the reduced model. Such a capability can aid in selecting the dimension of the reduced model, validating reduced order simulation results, and also in devising adaptive refinement schemes. The construction of such an error estimator for nonlinear dynamical systems is the focus of this work.

In the context of dynamical systems, considerable work on MOR schemes has been done for the linear, time-invariant case [1], and recently time-dependent and parameter-affine linear systems have been investigated [17]. Several reduction techniques have been proposed for MOR of nonlinear dynamical systems; see [26, 3, 24] for examples. In this work we shall consider the discrete empirical interpolation method (DEIM) as the underlying nonlinear MOR scheme, which was introduced in [5] and is based on the empirical interpolation method (EIM) [2]. The key idea behind

*Submitted to the journal's Methods and Algorithms for Scientific Computing section November 15, 2012; accepted for publication (in revised form) October 4, 2013; published electronically March 4, 2014. This work was supported in part by AFOSR grant FA9550-09-1-0225 and by NSF grant CCF-1017401 and the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart and the Baden-Württemberg Stiftung gGmbH through the German Research Foundation (DFG).

<http://www.siam.org/journals/sisc/36-2/89904.html>

[†]Institute for Applied Analysis and Numerical Simulation (IANS), University of Stuttgart, D-70569 Stuttgart, Germany (daniel.wirtz@mathematik.uni-stuttgart.de, bernard.haasdonk@mathematik.uni-stuttgart.de).

[‡]Department of Computational & Applied Mathematics (CAAM), Rice University, Houston, TX 77005-1892 (sorensen@rice.edu).

DEIM/EIM in this context is to replace the orthogonal projection typical of a Galerkin scheme with a far more efficient interpolatory projection of the system's nonlinearities. This greatly reduces computational complexity as it only requires evaluation of the nonlinearities at a few selected components indexed by suitable interpolation points. In this work, we will develop a new a posteriori error estimation scheme for this setting. Our approach involves a preliminary offline phase to construct the components of the estimator and an efficient online calculation as the reduced order model simulation proceeds.

We first introduce some notation. Throughout this work we assume $\Omega \subset \mathbb{R}^d$ to be an open set, $T > 0$, and $\mathbf{f} : \Omega \rightarrow \Omega$ is Lipschitz-continuous. We will indicate matrix- and vector-valued variables by bold upper- and lower-case Latin letters and scalar values by normal typesetting. To introduce the basic ideas of the estimator, we shall begin with a complete discussion of the basic system

$$(1.1) \quad \mathbf{y}'(t) = \mathbf{f}(\mathbf{y}(t)), \quad \mathbf{y}(0) = \mathbf{y}_0,$$

where $\mathbf{y}(t) \in \mathbb{R}^d$ denotes the system's state at $t \in [0, T]$ and $\mathbf{y}_0 \in \mathbb{R}^d$ is the initial state. Once the ideas have been explained in this setting, we shall introduce more complexity into the system by including affine linear parametric terms and parametrically dependent inputs and remark on systems with outputs.

The structure of this paper is as follows: After a short review of the MOR principles in section 1.1 we introduce our a posteriori error estimation procedure for the system (1.1) in section 2. In section 3 we transfer the previous results to a more general parameterized setting. Section 4 illustrates the applicability of our theoretical results, and we conclude in section 5.

1.1. Reduction methodology background. One key element of the MOR process is to apply a Galerkin projection of (1.1) into a suitable linear subspace of \mathbb{R}^d , whose spanning *reduced basis* vectors shall be given by the columns of the matrix $\mathbf{V} \in \mathbb{R}^{d \times r}$ with $r \ll d$. The aim is to design the reduced basis in such a way that it preserves as much as possible of the full system's behavior at minimal size. The reduction works independently of the actual choice of the basis \mathbf{V} , and our analysis retains this independence. The original DEIM method [5] obtains \mathbf{V} via a proper orthogonal decomposition (POD) of *snapshots*, which are discrete samples of trajectories of the full system. This approach has proven to be quite useful [22] as it satisfies certain optimality criteria with respect to the snapshots used. See [20] and the references therein for an overview. When parameters are present, the adaptive POD-Greedy algorithm [16, 15] can also be applied to obtain \mathbf{V} , especially in conjunction with the a posteriori error estimator derived in this paper. See [12] for a similar application context.

In addition, to allow for more general projection scenarios with different test spaces, we assume there are two given biorthogonal matrices $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{d \times r}$, $\mathbf{V}^T \mathbf{W} = \mathbf{I}_r$, where \mathbf{I}_r denotes the r -dimensional identity matrix. In the previous setting we would simply have $\mathbf{W} = \mathbf{V}$ (Galerkin) instead of $\mathbf{W} \neq \mathbf{V}$ (Petrov–Galerkin).

The second ingredient to the reduction approach is the DEIM approximation of \mathbf{f} in (1.1). For details of the DEIM methodology we refer the reader to the original work [5]; here we just assume that the DEIM approximation specifications are available. Hence, for a maximum order $M \leq d$, we are given a set of linearly independent vectors $\mathcal{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_M\} \subset \mathbb{R}^d$ (referred to as the DEIM basis) along with “interpolation points” $\mathcal{E} = \{\varrho_1, \dots, \varrho_M\} \subseteq \{1, \dots, d\}$. The DEIM approach then chooses a linear combination of the vectors from \mathcal{U} such that the component functions of \mathbf{f} selected

by \mathcal{E} are interpolated by the corresponding components of the DEIM approximating function $\tilde{\mathbf{f}}_m$. Mathematically, for $m \leq M$, the m th order DEIM approximation of \mathbf{f} takes the form

$$(1.2) \quad \tilde{\mathbf{f}}_m(\mathbf{y}) := \mathbf{U}_m(\mathbf{P}_m^T \mathbf{U}_m)^{-1} \mathbf{P}_m^T \mathbf{f}(\mathbf{y}),$$

where $\mathbf{P}_m := [\mathbf{e}_{\varrho_1}, \dots, \mathbf{e}_{\varrho_m}]$, $\mathbf{U}_m := [\mathbf{u}_1, \dots, \mathbf{u}_m]$, and $\mathbf{e}_i \in \mathbb{R}^d$ denotes the i th unit vector in \mathbb{R}^d . The interpolation property is easily verified by left-multiplication of (1.2) with \mathbf{P}_m^T . The DEIM algorithm uses a greedy maximum-residual criteria for selecting the interpolation indices \mathcal{E} which ensures that $\mathbf{P}_m^T \mathbf{U}_m$ is invertible. In our applications we obtain \mathcal{U} from a POD of \mathbf{f} -evaluations on trajectory snapshots that have already been computed in the process of obtaining \mathbf{V} .

Application of the above methods to the full system (1.1) yields the reduced order model

$$\begin{aligned} \mathbf{z}'(t) &= \mathbf{W}^T \tilde{\mathbf{f}}_m(\mathbf{V}\mathbf{z}(t)) = \tilde{\mathbf{U}}_m \mathbf{P}_m^T \mathbf{f}(\mathbf{V}\mathbf{z}(t)), \\ \mathbf{z}(0) &= \mathbf{z}_0 := \mathbf{W}^T \mathbf{y}_0, \end{aligned}$$

whose reduced state we denote by $\mathbf{z}(t) \in \mathbb{R}^r$ and $\tilde{\mathbf{U}}_m := \mathbf{W}^T \mathbf{U}_m (\mathbf{P}_m^T \mathbf{U}_m)^{-1}$. We do not include \mathbf{P}_m^T in $\tilde{\mathbf{U}}_m$ because the matrix \mathbf{P}_m^T effectively selects the $\varrho_1, \dots, \varrho_m$ components of \mathbf{f} . Thus, $\tilde{\mathbf{U}}_m$ can be precomputed and stored and later multiplied with the vector of the selected component evaluations $\mathbf{P}_m^T \mathbf{f}$ during the course of the reduced order simulation.

2. Error estimation for DEIM reduced systems. In the context of error estimation, previous work on error analysis for a variety of reduction schemes and nonlinear systems can be found in, e.g., [19, 29, 25, 23, 26]. For the POD-DEIM approach considered in this work, an a priori error estimate in terms of neglected singular values (for both projection and DEIM basis) was recently developed in [6]. However, while theoretically satisfying, this a priori estimate has little practical value with respect to assessing the accuracy of a reduced order solution during a simulation.

To address this need, we introduce an efficient a posteriori error estimate whose computation can be fully decomposed into an offline and online stage, inheriting some of the principles introduced in [33, 32, 17]. Related work on a posteriori error estimation has also been done for nonlinear parametrized evolution equations [9] and linear nonaffine time-varying PDEs [12].

The key ideas of our error estimation procedure are application of the comparison lemma [18, p. 32], the use of logarithmic Lipschitz constants [27], and efficient approximations thereof. Moreover, an estimation of the DEIM approximation error using a higher order DEIM approximation is employed, which extends the ideas from [9, 13, 28]. Note that in many cases differential inequalities such as the comparison lemma are used to bound solutions or errors a priori. Instead, we will use it to provide a posteriori error bounds which can be computed along with the reduced system for any given configuration.

After some preliminaries in section 2.1, we will introduce the resulting estimates in section 2.2. A more practical version involving matrix-DEIM (MDEIM) approximations of Jacobians and partial similarity transformations for efficient local logarithmic Lipschitz constant estimations will be introduced in section 2.3. Finally, we present the offline/online decomposition of the error estimators in section 2.4.

2.1. Preliminaries. Before developing our main results, we need to establish some required concepts and useful intermediate results. For the remainder of this

work, we let $\mathbf{G} \in \mathbb{R}^{d \times d}$ denote a symmetric positive definite weighting matrix. Then \mathbf{G} defines a scalar product $\langle \mathbf{x}, \mathbf{y} \rangle_G := \mathbf{x}^T \mathbf{G} \mathbf{y}$ on \mathbb{R}^d with corresponding norm $\|\mathbf{x}\|_G := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_G}$. For any square matrix \mathbf{A} , we denote the spectrum of \mathbf{A} by $\sigma(\mathbf{A})$.

A crucial part for the error estimation process is the concept of logarithmic Lipschitz constants for functions. They have been introduced in [7] for the linear case and a more general theory has become available since. See [27] for an elegant overview.

DEFINITION 2.1 (logarithmic Lipschitz constants). *For a function $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ we define the logarithmic Lipschitz constant with respect to \mathbf{G} by*

$$(2.1) \quad L_G[\mathbf{f}] := \lim_{h \rightarrow 0^+} \frac{1}{h} \left(\sup_{\mathbf{x} \neq \mathbf{y} \in \mathbb{R}^d} \frac{\|\mathbf{x} - \mathbf{y} + h(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}))\|_G}{\|\mathbf{x} - \mathbf{y}\|_G} - 1 \right).$$

Next we give an equivalent representation that is more suitable for applications.

LEMMA 2.2 (equivalent representations for logarithmic Lipschitz constants). *If \mathbf{f} is Lipschitz-continuous, the logarithmic Lipschitz constant of \mathbf{f} is given by*

$$L_G[\mathbf{f}] = \sup_{\mathbf{x} \neq \mathbf{y} \in \mathbb{R}^d} \frac{\langle \mathbf{x} - \mathbf{y}, \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}) \rangle_G}{\|\mathbf{x} - \mathbf{y}\|_G^2}.$$

Proof. Let L denote the Lipschitz constant of \mathbf{f} and consider $h < \frac{1}{L}$. Put $\Delta \mathbf{f} := \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})$ and note $\|\Delta \mathbf{f}\|_G \leq L \|\mathbf{x} - \mathbf{y}\|_G$. It is straightforward to show that

$$\frac{1}{h} \left(\frac{\|\mathbf{x} - \mathbf{y} + h\Delta \mathbf{f}\|_G}{\|\mathbf{x} - \mathbf{y}\|_G} - 1 \right) = \frac{2 \langle \mathbf{x} - \mathbf{y}, \Delta \mathbf{f} \rangle_G + h \|\Delta \mathbf{f}\|_G^2}{\|\mathbf{x} - \mathbf{y}\|_G (\|\mathbf{x} - \mathbf{y} + h\Delta \mathbf{f}\|_G + \|\mathbf{x} - \mathbf{y}\|_G)}.$$

Now, using

$$\|\mathbf{x} - \mathbf{y}\|_G (1 - hL) \leq \|\mathbf{x} - \mathbf{y} + h\Delta \mathbf{f}\|_G \leq \|\mathbf{x} - \mathbf{y}\|_G (1 + hL)$$

gives

$$\frac{2 \langle \mathbf{x} - \mathbf{y}, \Delta \mathbf{f} \rangle_G}{\|\mathbf{x} - \mathbf{y}\|_G^2 (2 + hL)} \leq \frac{1}{h} \left(\frac{\|\mathbf{x} - \mathbf{y} + h\Delta \mathbf{f}\|_G}{\|\mathbf{x} - \mathbf{y}\|_G} - 1 \right) \leq \frac{2 \langle \mathbf{x} - \mathbf{y}, \Delta \mathbf{f} \rangle_G}{\|\mathbf{x} - \mathbf{y}\|_G^2 (2 - hL)} + \frac{hL^2}{2 - hL}.$$

Thus,

$$\begin{aligned} \frac{2}{2 + hL} \sup_{\mathbf{x} \neq \mathbf{y} \in \mathbb{R}^d} \frac{\langle \mathbf{x} - \mathbf{y}, \Delta \mathbf{f} \rangle_G}{\|\mathbf{x} - \mathbf{y}\|_G^2} &\leq \sup_{\mathbf{x} \neq \mathbf{y} \in \mathbb{R}^d} \frac{1}{h} \left(\frac{\|\mathbf{x} - \mathbf{y} + h\Delta \mathbf{f}\|_G}{\|\mathbf{x} - \mathbf{y}\|_G} - 1 \right) \\ &\leq \frac{2}{2 - hL} \sup_{\mathbf{x} \neq \mathbf{y} \in \mathbb{R}^d} \frac{\langle \mathbf{x} - \mathbf{y}, \Delta \mathbf{f} \rangle_G}{\|\mathbf{x} - \mathbf{y}\|_G^2} + \frac{hL^2}{2 - hL}, \end{aligned}$$

and finally, taking limits as $h \rightarrow 0^+$ across the inequalities will establish the result. \square

COROLLARY 2.3 (logarithmic Lipschitz constants for linear functions). *The logarithmic Lipschitz constant of a linear function $\mathbf{f} = \mathbf{A}\mathbf{y}$ with $\mathbf{A} \in \mathbb{R}^{d \times d}$ is given by*

$$(2.2) \quad L_G[\mathbf{A}] := \lim_{h \rightarrow 0^+} \frac{\|\mathbf{I} + h\mathbf{A}\|_G - 1}{h},$$

where $\|\cdot\|_G$ is the \mathbf{G} -induced matrix norm. Furthermore, (2.2) is equivalent to both

$$(2.3) \quad L_G[\mathbf{A}] = \sup_{\mathbf{x} \in \mathbb{R}^d \setminus \{0\}} \frac{\langle \mathbf{x}, \mathbf{Ax} \rangle_G}{\langle \mathbf{x}, \mathbf{x} \rangle_G},$$

$$(2.4) \quad L_G[\mathbf{A}] = \max \left\{ \sigma \left(\frac{1}{2} (\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^T) \right) \right\},$$

where $\tilde{\mathbf{A}} := \mathbf{C}^T \mathbf{AC}^{-T}$ and $\mathbf{G} = \mathbf{CC}^T$ denotes the Cholesky factorization of \mathbf{G} .

Proof. We obtain (2.2) directly from (2.1) using linearity and the matrix norm definition. As the map $\mathbf{f}(\mathbf{y}) = \mathbf{Ay}$ is Lipschitz, application of Lemma 2.2 directly yields (2.3). Let $\mathbf{x} \in \mathbb{R}^d \setminus \{0\}$ and $\mathbf{y} := \mathbf{C}^T \mathbf{x}$. Then

$$\frac{\langle \mathbf{x}, \mathbf{Ax} \rangle_G}{\langle \mathbf{x}, \mathbf{x} \rangle_G} = \frac{\mathbf{x}^T \mathbf{G} \mathbf{Ax}}{\mathbf{x}^T \mathbf{G} \mathbf{x}} = \frac{\mathbf{y}^T \mathbf{C}^T \mathbf{AC}^{-T} \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \frac{1}{2} \frac{\mathbf{y}^T \tilde{\mathbf{A}} \mathbf{y} + \mathbf{y}^T \tilde{\mathbf{A}}^T \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \frac{1}{2} \frac{\mathbf{y}^T (\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^T) \mathbf{y}}{\mathbf{y}^T \mathbf{y}},$$

which shows equality of (2.3) and (2.4) via

$$\sup_{\mathbf{x} \in \mathbb{R}^d \setminus \{0\}} \frac{\langle \mathbf{x}, \mathbf{Ax} \rangle_G}{\langle \mathbf{x}, \mathbf{x} \rangle_G} = \sup_{\mathbf{y} \in \mathbb{R}^d \setminus \{0\}} \frac{1}{2} \frac{\mathbf{y}^T (\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^T) \mathbf{y}}{\mathbf{y}^T \mathbf{y}} = \max \left\{ \sigma \left(\frac{1}{2} (\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^T) \right) \right\}. \quad \square$$

Remark 2.4. In our expression for the logarithmic Lipschitz constant of a linear function given above, we abused notation a bit by entering \mathbf{A} rather than \mathbf{f} as the argument in order to keep with the notation introduced by Dahlquist [7]. The notion of logarithmic Lipschitz constants is a natural generalization of the logarithmic norm [27].

As demonstrated in [33, 32] for reduced kernel-based systems, certain local information given by the reduced state-space coordinates can be useful. In this context, the notion of *local* logarithmic Lipschitz constants will be important.

DEFINITION 2.5 (local logarithmic Lipschitz constants). *For a function $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ we define the local logarithmic Lipschitz constant at $\mathbf{y} \in \mathbb{R}^d$ with respect to \mathbf{G} by*

$$L_G[\mathbf{f}](\mathbf{y}) = \sup_{\mathbf{x} \in \mathbb{R}^d} \frac{\langle \mathbf{x} - \mathbf{y}, \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y}) \rangle_G}{\|\mathbf{x} - \mathbf{y}\|_G^2}.$$

As the application of the comparison lemma [18, p. 32] is a key ingredient to both our error estimators and their computation, we state it here in a tailored version for completeness.

LEMMA 2.6 (comparison lemma). *Let $T > 0$, $u, \alpha, \beta : [0, T] \rightarrow \mathbb{R}$ be integrable functions, u differentiable, and assume*

$$(2.5) \quad u'(t) \leq \beta(t)u(t) + \alpha(t) \quad \forall t \in [0, T].$$

Then

$$(2.6) \quad u(t) \leq \int_0^t \alpha(s) e^{\int_s^t \beta(\tau) d\tau} ds + e^{\int_0^t \beta(\tau) d\tau} u(0) \quad \forall t \in [0, T].$$

Furthermore, (2.6) is an equality if and only if (2.5) is an equality.

Proof. Define $v(t) := \int_0^t \beta(\tau) d\tau$. Then (2.6) follows from

$$\begin{aligned} u(t) &= e^{v(t)} e^{-v(t)} u(t) - e^{v(t)} e^{-v(0)} u(0) + e^{v(t)} e^{-v(0)} u(0) \\ &= e^{v(t)} \int_0^t \left(e^{-v(s)} u(s) \right)' ds + e^{v(t)} u(0) \\ &\leq e^{v(t)} \int_0^t e^{-v(s)} \alpha(s) ds + e^{v(t)} u(0) = \int_0^t \alpha(s) e^{\int_s^t \beta(\tau) d\tau} ds + e^{\int_0^t \beta(\tau) d\tau} u(0). \end{aligned}$$

This derivation also directly shows the equality of (2.6) if (2.5) is an equality. Equality in the other case follows from simple differentiation. \square

Finally, we shall establish the formalism needed in order to compute the DEIM approximation error. The following lemma facilitates an efficient computation by providing a useful decomposition of the DEIM projection matrix relative to an intermediate lower order DEIM approximation.

LEMMA 2.7 (DEIM matrix decomposition). *Let $d, m, m' \in \mathbb{N}, m + m' \leq d$ and matrices $\mathbf{U}_m, \mathbf{P}_m \in \mathbb{R}^{d \times m}, \mathbf{U}_{m'}, \mathbf{P}_{m'} \in \mathbb{R}^{d \times m'}$ be given. Assume that the matrices $\mathbf{U} := [\mathbf{U}_m \ \mathbf{U}_{m'}]$, $\mathbf{P} := [\mathbf{P}_m \ \mathbf{P}_{m'}]$ each have linearly independent columns and that the matrices $\mathbf{P}^T \mathbf{U}$ and $\mathbf{P}_m^T \mathbf{U}_m$ are nonsingular. Define the oblique projectors*

$$(2.7a) \quad \Pi := \mathbf{U}(\mathbf{P}^T \mathbf{U})^{-1} \mathbf{P}^T,$$

$$(2.7b) \quad \Pi_m := \mathbf{U}_m(\mathbf{P}_m^T \mathbf{U}_m)^{-1} \mathbf{P}_m^T,$$

$$(2.7c) \quad \Pi_{m'} := (\mathbf{I} - \Pi_m) \mathbf{U}_{m'} (\mathbf{P}_{m'}^T (\mathbf{I} - \Pi_m) \mathbf{U}_{m'})^{-1} \mathbf{P}_{m'}^T.$$

Then we have

$$\Pi = \Pi_m + \Pi_{m'} (\mathbf{I} - \Pi_m).$$

Proof. First note that Π may be written as $\Pi = \mathbf{U}_m \mathbf{X} + \mathbf{U}_{m'} \mathbf{Y}$ with

$$(2.8) \quad \begin{bmatrix} \mathbf{P}_m^T \mathbf{U}_m & \mathbf{P}_m \mathbf{U}_{m'} \\ \mathbf{P}_{m'}^T \mathbf{U}_m & \mathbf{P}_{m'}^T \mathbf{U}_{m'} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_m^T \\ \mathbf{P}_{m'}^T \end{bmatrix}.$$

One step of block Gaussian elimination applied to (2.8) provides the transformed equation

$$(2.9) \quad \begin{bmatrix} \mathbf{P}_m^T \mathbf{U}_m & \mathbf{P}_m \mathbf{U}_{m'} \\ \mathbf{0} & \mathbf{P}_{m'}^T (\mathbf{I} - \Pi_m) \mathbf{U}_{m'} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_m^T \\ \mathbf{P}_{m'}^T (\mathbf{I} - \Pi_m) \end{bmatrix}.$$

The nonsingularity of $\mathbf{P}^T \mathbf{U}$ implies the nonsingularity of $\mathbf{P}_{m'}^T (\mathbf{I} - \Pi_m) \mathbf{U}_{m'}$ to give the expression

$$\mathbf{U}_{m'} \mathbf{Y} = \mathbf{U}_{m'} (\mathbf{P}_{m'}^T (\mathbf{I} - \Pi_m) \mathbf{U}_{m'})^{-1} \mathbf{P}_{m'}^T (\mathbf{I} - \Pi_m).$$

Now (2.9) implies $\mathbf{X} = \Pi_m - \Pi_m \mathbf{U}_{m'} \mathbf{Y}$ and it follows that

$$\Pi = \mathbf{U}_m \mathbf{X} + \mathbf{U}_{m'} \mathbf{Y} = \Pi_m + (\mathbf{I} - \Pi_m) \mathbf{U}_{m'} \mathbf{Y} = \Pi_m + \Pi_{m'} (\mathbf{I} - \Pi_m)$$

as claimed. \square

For simplicity, we have used $\mathbf{U}_{m'}, \Pi_{m'}$ instead of introducing new symbols $\mathbf{U}'_{m'}, \Pi'_{m'}$, etc. Similar to [9, 14], the key idea is to assume having a maximum DEIM order M at

which the approximation is essentially exact (meaning it is accurate to within working precision). An application of Lemma 2.7 with $m' := M - m$ directly leads to the following theorem, which shows how to efficiently compute the DEIM approximation error using only DEIM matrices of sizes m and $m' = M - m$ instead of $m + m' = M$.

THEOREM 2.8 (error representation). *Let a DEIM basis $\mathcal{U} := \{\mathbf{u}_1, \dots, \mathbf{u}_M\}$ and a set of interpolation points $\mathcal{E} = \{\varrho_1, \dots, \varrho_M\}$ be given and assume that the M th order DEIM approximation of \mathbf{f} is exact, i.e., $\tilde{\mathbf{f}}_M \equiv \mathbf{f}$ on Ω . For $m \leq M - 1$, $m' = M - m$, set $\mathbf{P}_m := [\mathbf{e}_{\varrho_1}, \dots, \mathbf{e}_{\varrho_m}]$, $\mathbf{P}_{m'} := [\mathbf{e}_{\varrho_{m+1}}, \dots, \mathbf{e}_{\varrho_{m+m'}}]$, $\mathbf{U}_m := [\mathbf{u}_1, \dots, \mathbf{u}_m]$, and $\mathbf{U}_{m'} := [\mathbf{u}_{m+1}, \dots, \mathbf{u}_{m+m'}]$. Then the approximation error for the m th order DEIM approximation of \mathbf{f} is given by*

$$(2.10) \quad \mathbf{f}(\mathbf{y}) - \tilde{\mathbf{f}}_m(\mathbf{y}) = (\Pi - \Pi_m)\mathbf{f}(\mathbf{y}) = \Pi_{m'}(\mathbf{I} - \Pi_m)\mathbf{f}(\mathbf{y}),$$

with matrices $\Pi, \Pi_m, \Pi_{m'}$ as defined in (2.7).

This form of the error has a revealing interpretation. It expresses the error as an oblique projection of the DEIM approximation error $(\mathbf{I} - \Pi_m)\mathbf{f}(\mathbf{y})$ whose norm is within a factor of $\|\Pi_m\| = \|(\mathbf{P}_m^T \mathbf{U}_m)^{-1}\|$ of the optimal POD error when the basis \mathbf{U}_m is orthogonal [5]. Of course, the hypothesis $\mathbf{f} = \tilde{\mathbf{f}}_M = \Pi\mathbf{f}$ is generally satisfied only when $M = d$ suggesting that a large m' would be required. However, our experiments show that quite small values of m' can be used to obtain very accurate estimates of the DEIM approximation error in practice.

2.2. A posteriori error estimation. With these preliminaries, we can present our first result regarding a posteriori error estimation. Let $\mathbf{y}^r(t) := \mathbf{V}\mathbf{z}(t)$, $t \in [0, T]$, denote the reconstructed state-space variable and $\mathbf{e}(t) := \mathbf{y}(t) - \mathbf{y}^r(t)$ the state-space error. Then $\mathbf{e}(t)$ is given as the solution of the error system

$$(2.11) \quad \mathbf{e}'(t) = \mathbf{f}(\mathbf{y}(t)) - \mathbf{V}\mathbf{W}^T \tilde{\mathbf{f}}_m(\mathbf{y}^r(t)), \quad \mathbf{e}(0) = \mathbf{y}_0 - \mathbf{V}\mathbf{W}^T \mathbf{y}_0.$$

THEOREM 2.9 (a posteriori error estimation for DEIM reduced systems). *Let the conditions from Theorem 2.9 hold. Then the state-space error is bounded via*

$$\|\mathbf{e}(t)\|_G \leq \Delta(t) \quad \forall t \in [0, T],$$

with

$$(2.12) \quad \Delta(t) := \int_0^t \alpha(s) e^{\int_s^t \beta(\tau) d\tau} ds + e^{\int_0^t \beta(\tau) d\tau} \left\| \mathbf{y}_0 - \mathbf{V}\mathbf{W}^T \mathbf{y}_0 \right\|_G,$$

$$(2.13) \quad \alpha(t) := \left\| \left(\Pi_{m'}(\mathbf{I} - \Pi_m) + (\mathbf{I} - \mathbf{V}\mathbf{W}^T)\Pi_m \right) \mathbf{f}(\mathbf{y}^r(t)) \right\|_G,$$

$$(2.14) \quad \beta(t) := L_G[\mathbf{f}](\mathbf{y}^r(t)),$$

with $\Pi_m, \Pi_{m'}$ defined as in Lemma 2.7, (2.7).

Proof. Note that for $m' = M - m$, Theorem 2.8 implies

$$\begin{aligned} & \mathbf{f}(\mathbf{y}^r(t)) - \mathbf{V}\mathbf{W}^T \tilde{\mathbf{f}}_m(\mathbf{y}^r(t)) \\ &= \mathbf{f}(\mathbf{y}^r(t)) - \tilde{\mathbf{f}}_m(\mathbf{y}^r(t)) + \tilde{\mathbf{f}}_m(\mathbf{y}^r(t)) - \mathbf{V}\mathbf{W}^T \tilde{\mathbf{f}}_m(\mathbf{y}^r(t)) \\ (2.15) \quad &= \Pi_{m'}(\mathbf{I} - \Pi_m)\mathbf{f}(\mathbf{y}^r(t)) + (\mathbf{I} - \mathbf{V}\mathbf{W}^T)\Pi_m\mathbf{f}(\mathbf{y}^r(t)). \end{aligned}$$

Now, from (2.11) and (2.15) we obtain

$$\begin{aligned} \langle \mathbf{e}(t), \mathbf{e}'(t) \rangle_G &= \langle \mathbf{e}(t), \mathbf{f}(\mathbf{y}(t)) - \mathbf{f}(\mathbf{y}^r(t)) \rangle_G + \left\langle \mathbf{e}(t), \mathbf{f}(\mathbf{y}^r(t)) - \mathbf{V}\mathbf{W}^T \tilde{\mathbf{f}}_m(\mathbf{y}^r(t)) \right\rangle_G \\ &\leq L_G[\mathbf{f}](\mathbf{y}^r(t)) \|\mathbf{e}(t)\|_G^2 \\ &\quad + \left\langle \mathbf{e}(t), \Pi_{m'}(\mathbf{I} - \Pi_m)\mathbf{f}(\mathbf{y}^r(t)) + (\mathbf{I} - \mathbf{V}\mathbf{W}^T)\Pi_m\mathbf{f}(\mathbf{y}^r(t)) \right\rangle_G \\ &= \beta(t) \|\mathbf{e}(t)\|_G^2 + \|\mathbf{e}(t)\|_G \alpha(t). \end{aligned}$$

This finally gives

$$\frac{d}{dt} \|\mathbf{e}(t)\|_G = \frac{\langle \mathbf{e}(t), \mathbf{e}'(t) \rangle_G}{\|\mathbf{e}(t)\|_G} \leq \beta(t) \|\mathbf{e}(t)\|_G + \alpha(t),$$

and application of Lemma 2.6 with $u(t) := \|\mathbf{e}(t)\|_G$, $u(0) = \|\mathbf{y}_0 - \mathbf{V}\mathbf{W}^T \mathbf{y}_0\|_G$ yields the desired results. \square

This form of the driving function α shows that $\alpha(t)$ remains small when $\mathbf{f}(\mathbf{y}^r(\cdot))$ is well approximated from the range of \mathbf{U} and $\Pi_m\mathbf{f}(\mathbf{y}^r(\cdot))$ is well approximated from the range of \mathbf{V} simultaneously.

The exactness assumption of the DEIM approximation of order M deserves some discussion. Our experiments in section 4.2 show the practicality of the above approach even for substantially smaller m' values than $M - m$. For scenarios that lead to prohibitively large m' values in order to obtain good accuracy, we suggest investigating DEIM splitting techniques like the “hp” empirical interpolation [11], implicit partitioning [30], or the localized DEIM method [30] in order to ensure a sufficient POD singular value decay of the DEIM basis modes.

Remark 2.10. As an alternative to the exactness assumption in Theorem 2.9, if an a priori bound ϵ on the DEIM approximation error on Ω is available, the estimation procedure would lead directly to a rigorous a posteriori error estimator by including ϵ in (2.13). Another possible estimate could be based upon the recent results on a posteriori EIM approximation errors in [10], which work without the exactness assumption but use Taylor expansions around suitable points instead.

Finally, note that zero error is achieved if the DEIM approximation is exact, i.e., $m = M$ in our context, and the trajectory is completely contained in the subspace \mathbf{V} . In this case we will have zero initial error and $\alpha(t) \equiv 0$, which means the estimator correctly predicts zero error.

2.3. Efficient error estimation. The error estimator introduced in Theorem 2.9 makes explicit use of the local logarithmic Lipschitz constant $L_G[\mathbf{f}](\mathbf{y}^r(t))$, which is (as well as its global counterpart $L_G[\mathbf{f}]$) not readily available in most practical situations. Therefore, let $\mathbf{J} : \Omega \rightarrow \mathbb{R}^{d \times d}$ denote the Jacobian $\mathbf{J}(\mathbf{y})$ of \mathbf{f} at $\mathbf{y} \in \Omega$. With the Taylor expansion of \mathbf{f} around $\mathbf{y}^r(t)$ we obtain

$$\begin{aligned} \frac{\langle \mathbf{e}(t), \mathbf{f}(\mathbf{y}(t)) - \mathbf{f}(\mathbf{y}^r(t)) \rangle_G}{\|\mathbf{e}(t)\|_G^2} &= \frac{\left\langle \mathbf{e}(t), \mathbf{J}(\mathbf{y}^r(t))\mathbf{e}(t) + \mathcal{O}(\|\mathbf{e}(t)\|_G^2) \right\rangle_G}{\|\mathbf{e}(t)\|_G^2} \\ (2.16) \quad &= \frac{\langle \mathbf{e}(t), \mathbf{J}(\mathbf{y}^r(t))\mathbf{e}(t) \rangle_G}{\|\mathbf{e}(t)\|_G^2} + \mathcal{O}(\|\mathbf{e}(t)\|_G) \end{aligned}$$

for any $t \in [0, T]$. With Definition 2.5 and (2.3), this directly gives a first order approximation of the local logarithmic Lipschitz constant

$$(2.17) \quad L_G[\mathbf{f}](\mathbf{y}^r(t)) = L_G[\mathbf{J}(\mathbf{y}^r(t))] + \mathcal{O}(\|\mathbf{e}(t)\|_G).$$

Using the approximation (2.17) avoids the need to obtain $L_G[\mathbf{f}](\mathbf{y}^r(t))$, but it comes with additional cost: The computation of the Jacobian logarithmic norm is expensive as it involves solving an eigenvalue problem of high dimension d due to the representation (2.4).

We will address this issue in the following discussion, where we propose applying a suitable partial similarity transformation to the Jacobians, which has been designed to preserve the largest eigenvalues of their symmetric parts. The key ingredient is to perform a POD of their corresponding eigenvectors, which in turn allows us to bound the resulting eigenvalue approximation error in terms of the remaining eigenvalues of their covariance matrix. We explain this idea for general symmetric matrices in the following theorem, and refer to [21, 29] for details on POD and related error estimates. We will make use of some MATLAB style notation in the following, e.g., matrix indexing as $\mathbf{A}(:, 1:k)$ for the first k columns of a matrix \mathbf{A} or $[a, b] = f(c)$ for a function f that returns two arguments a, b .

THEOREM 2.11 (approximation of eigenvalues for a family of symmetric matrices). *Let a continuous family of symmetric matrices $\mathbf{H}(t) \in \mathbb{R}^{d \times d}$ over $t \in [a, b]$ be given and let $[\lambda(t), \mathbf{q}(t)] := \lambda_{\max}(\mathbf{H}(t))$ denote the largest eigenvalue $\lambda(t)$ with corresponding normalized eigenvector $\mathbf{q}(t)$ of $\mathbf{H}(t)$. Let $C_H > 0$ be chosen so that $\sup_{t \in [a, b]} \|\mathbf{H}(t)\| \leq C_H$ holds. Further, define $\mathbf{R} = \int_a^b \mathbf{q}(t) \mathbf{q}(t)^T dt$ and let $\mathbf{Q} \Sigma^2 \mathbf{Q}^T = \mathbf{R}$ be the eigendecomposition with $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$ and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_d)$ with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_d \geq 0$. For $k \leq d$ let $\mathbf{Q}_k := \mathbf{Q}(:, 1:k)$ and $\lambda_k(t) := \lambda_{\max}(\mathbf{Q}_k^T \mathbf{H}(t) \mathbf{Q}_k)$. Then*

$$\int_a^b |\lambda(t) - \lambda_k(t)| dt \leq 4C_H \sum_{j>k} \sigma_j^2.$$

Proof. Consider the case \mathbf{R} is nonsingular, i.e., $\sigma_d > 0$. Define the vector-valued function $\mathbf{w}(t) := \Sigma^{-1} \mathbf{Q}^T \mathbf{q}(t)$, $t \in [a, b]$. Note that

$$\int_a^b \mathbf{w}(t) \mathbf{w}(t)^T dt = \int_a^b \Sigma^{-1} \mathbf{Q}^T \mathbf{q}(t) \mathbf{q}(t)^T \mathbf{Q} \Sigma^{-1} dt = \Sigma^{-1} \mathbf{Q}^T \mathbf{R} \mathbf{Q} \Sigma^{-1} = \mathbf{I}.$$

Thus, $\int_a^b w_i(t) w_j(t) dt = \delta_{ij}$ (the Kronecker delta), where $w_i(t)$ is the i th component of $\mathbf{w}(t)$. Now, partition

$$\mathbf{Q} = [\mathbf{Q}_k, \tilde{\mathbf{Q}}_k], \quad \Sigma = \text{diag}(\Sigma_k, \tilde{\Sigma}_k), \quad \mathbf{w}(t) = [\mathbf{w}_k^T(t), \tilde{\mathbf{w}}_k^T(t)]^T,$$

with $\tilde{\mathbf{Q}}_k := \mathbf{Q}(:, (k+1):d)$, $\Sigma_k, \tilde{\Sigma}_k$ denoting the appropriate diagonal blocks of Σ , and $\mathbf{w}_k(t), \tilde{\mathbf{w}}_k(t)$ denoting the corresponding subvectors of $\mathbf{w}(t)$. Put

$$\mathbf{q}(t) = \mathbf{q}_k(t) + \tilde{\mathbf{q}}_k(t), \quad \text{with } \mathbf{q}_k(t) := \mathbf{Q}_k \Sigma_k \mathbf{w}_k(t), \quad \tilde{\mathbf{q}}_k(t) := \tilde{\mathbf{Q}}_k \tilde{\Sigma}_k \tilde{\mathbf{w}}_k(t),$$

and note that $\mathbf{q}_k^T(t) \tilde{\mathbf{q}}_k(t) = 0$ for all $t \in [a, b]$. We observe that

$$(2.18) \quad \int_a^b \tilde{\mathbf{q}}_k^T(t) \tilde{\mathbf{q}}_k(t) dt = \int_a^b \tilde{\mathbf{w}}_k^T(t) \tilde{\Sigma}_k^2 \tilde{\mathbf{w}}_k(t) dt = \sum_{j>k} \sigma_j^2 \int_a^b w_j^2(t) dt = \sum_{j>k} \sigma_j^2.$$

In the following we omit the argument t and set $\mathbf{q} = \mathbf{q}(t)$, $\mathbf{q}_k = \mathbf{q}_k(t)$, $\tilde{\mathbf{q}}_k = \tilde{\mathbf{q}}_k(t)$, etc. Then

$$(2.19) \quad \begin{aligned} \mathbf{q}_k^T \mathbf{H} \mathbf{q}_k &= (\mathbf{q} - \tilde{\mathbf{q}}_k)^T \mathbf{H} (\mathbf{q} - \tilde{\mathbf{q}}_k) = \lambda - 2\mathbf{q}^T \mathbf{H} \tilde{\mathbf{q}}_k + \tilde{\mathbf{q}}_k^T \mathbf{H} \tilde{\mathbf{q}}_k \\ &= \lambda - 2\lambda \mathbf{q}^T \tilde{\mathbf{q}}_k + \tilde{\mathbf{q}}_k^T \mathbf{H} \tilde{\mathbf{q}}_k = \lambda - 2\lambda \tilde{\mathbf{q}}_k^T \tilde{\mathbf{q}}_k + \tilde{\mathbf{q}}_k^T \mathbf{H} \tilde{\mathbf{q}}_k. \end{aligned}$$

Moreover, since $\mu \leq \|\mathbf{H}\| \leq C_H$ for any eigenvalue μ of \mathbf{H} , the definitions of $\lambda = \lambda(t)$ and $\lambda_k = \lambda_k(t)$ imply

$$(2.20) \quad \lambda = \sup_{\mathbf{v} \neq 0} \frac{\mathbf{v}^T \mathbf{H} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \geq \sup_{\mathbf{Q}_k \mathbf{v}_k \neq 0} \frac{\mathbf{v}_k^T \mathbf{Q}_k^T \mathbf{H} \mathbf{Q}_k \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{v}_k} = \lambda_k \geq \frac{\mathbf{q}_k^T \mathbf{H} \mathbf{q}_k}{\mathbf{q}_k^T \mathbf{q}_k}.$$

Combining (2.19) and (2.20) provides with $\|\mathbf{q}_k\|^2 + \|\tilde{\mathbf{q}}_k\|^2 = \|\mathbf{q}\|^2 = 1$ that

$$\begin{aligned} 0 \leq \lambda - \lambda_k &\leq \lambda - \frac{\mathbf{q}_k^T \mathbf{H} \mathbf{q}_k}{\mathbf{q}_k^T \mathbf{q}_k} = \lambda - \frac{\lambda - 2\lambda \tilde{\mathbf{q}}_k^T \tilde{\mathbf{q}}_k + \tilde{\mathbf{q}}_k^T \mathbf{H} \tilde{\mathbf{q}}_k}{1 - \tilde{\mathbf{q}}_k^T \tilde{\mathbf{q}}_k} \\ &= \frac{\lambda \tilde{\mathbf{q}}_k^T \tilde{\mathbf{q}}_k - \tilde{\mathbf{q}}_k^T \mathbf{H} \tilde{\mathbf{q}}_k}{1 - \|\tilde{\mathbf{q}}_k\|^2} = \left(\lambda - \frac{\tilde{\mathbf{q}}_k^T \mathbf{H} \tilde{\mathbf{q}}_k}{\|\tilde{\mathbf{q}}_k\|^2} \right) \frac{\|\tilde{\mathbf{q}}_k\|^2}{1 - \|\tilde{\mathbf{q}}_k\|^2} \leq 2 \|\mathbf{H}\| \frac{\|\tilde{\mathbf{q}}_k\|^2}{1 - \|\tilde{\mathbf{q}}_k\|^2}, \end{aligned}$$

which is equivalent to

$$0 \leq (\lambda - \lambda_k)(1 - \|\tilde{\mathbf{q}}_k\|^2) \leq 2 \|\mathbf{H}\| \|\tilde{\mathbf{q}}_k\|^2.$$

Hence,

$$0 \leq \lambda - \lambda_k \leq (2 \|\mathbf{H}\| + \lambda - \lambda_k) \|\tilde{\mathbf{q}}_k\|^2 \leq 4 \|\mathbf{H}\| \|\tilde{\mathbf{q}}_k\|^2 \leq 4C_H \|\tilde{\mathbf{q}}_k\|^2,$$

and therefore, using (2.18),

$$\int_a^b |\lambda(t) - \lambda_k(t)| dt \leq 4C_H \int_a^b \|\tilde{\mathbf{q}}_k(t)\|^2 dt = 4C_H \sum_{j>k} \sigma_j^2$$

as claimed. If \mathbf{R} is rank deficient, the above argument is still valid simply by replacing Σ^{-1} with the pseudoinverse Σ^+ . \square

This result can be applied directly in the context of approximating the logarithmic norms of the local Jacobians.

PROPOSITION 2.12 (Jacobian partial similarity transform). *As before let $\mathbf{C}\mathbf{C}^T$ denote the Cholesky decomposition of the weighting matrix \mathbf{G} , and consider the family of symmetric matrices*

$$\mathbf{H}(t) := \frac{1}{2} \left(\mathbf{C}^T \mathbf{J}(\mathbf{y}^r(t)) \mathbf{C}^{-T} + \left(\mathbf{C}^T \mathbf{J}(\mathbf{y}^r(t)) \mathbf{C}^{-T} \right)^T \right).$$

Then, from Theorem 2.11, we have the corresponding values $C_H > 0$, $\{\sigma_i\}_{i=1,\dots,d}$, $\mathbf{Q} \in \mathbb{R}^{d \times d}$, and

$$(2.21) \quad \begin{aligned} \lambda(t) &= L_G[\mathbf{J}(\mathbf{y}^r(t))], \\ \lambda_k(t) &= L_{I_k}[\mathbf{Q}_k^T \mathbf{C}^T \mathbf{J}(\mathbf{y}^r(t)) \mathbf{C}^{-T} \mathbf{Q}_k], \end{aligned}$$

from which we directly obtain the estimate

$$(2.22) \quad \int_0^T \left| L_G[\mathbf{J}(\mathbf{y}^r(t))] - L_{I_k}[\mathbf{Q}_k^T \mathbf{C}^T \mathbf{J}(\mathbf{y}^r(t)) \mathbf{C}^{-T} \mathbf{Q}_k] \right| dt \leq C_H \sum_{j>k} \sigma_j^2.$$

Since the matrix \mathbf{R} in Theorem 2.11 is generally not available, \mathbf{Q} is obtained as the set of left singular vectors of the singular value decomposition (SVD) of a *snapshot*

matrix $\mathbf{S}_N = \sqrt{(b-a)/N}[\mathbf{q}(t_1), \dots, \mathbf{q}(t_N)]$. Assuming equally spaced points $t_j \in [a, b]$ with $t_1 = a, t_n = b$, we have

$$\lim_{N \rightarrow \infty} \mathbf{S}_N \mathbf{S}_N^T = \int_a^b \mathbf{q}(t) \mathbf{q}(t)^T dt = \mathbf{R}.$$

Thus, for a sufficiently large N , the sum of the neglected squared singular values of \mathbf{S}_N will be arbitrarily close to the corresponding neglected eigenvalues of \mathbf{R} and can safely be used in the estimate. The detailed argument is similar to one given in [21].

For a given set of training data $\Xi := \{\mathbf{y}_i \mid i = 1, \dots, N\} \subseteq \Omega$, Algorithm 1 describes how to obtain \mathbf{Q} in practice. Notationally, the *SVD* method performs the singular value decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ for a matrix \mathbf{A} . However, in practice a truncated SVD that just computes the requisite dominant singular vectors is used.

Algorithm 1. $\mathbf{Q} = \text{getTransformationMatrix}(\Xi)$.

```

1:  $\mathbf{S}_N \leftarrow []$ 
2: for all  $\mathbf{y}_i \in \Xi$  do
3:    $\mathbf{M} \leftarrow \mathbf{C}^T \mathbf{J}(\mathbf{y}_i) \mathbf{C}^{-T}$ 
4:    $[\lambda_i, \mathbf{q}_i] \leftarrow \lambda_{\max}(\frac{1}{2}(\mathbf{M} + \mathbf{M}^T))$ 
5:    $\mathbf{S}_N \leftarrow [\mathbf{S}_N \ \mathbf{q}_i]$ 
6: end for
7:  $[\mathbf{Q}, \Sigma, \mathbf{V}^T] \leftarrow \text{SVD}(\sqrt{\frac{b-a}{N}} \mathbf{S}_N)$ 
8: return  $\mathbf{Q}$ 
```

Before we can derive an efficient error estimator variant with the transformation introduced above, there is one problem left to deal with. The reduced matrix from the right-hand side of (2.21) is of small size $k \times k$; however, its computation involves the transformed Jacobian $\mathbf{C}^T \mathbf{J}(\mathbf{y}^r(t)) \mathbf{C}^{-T} \in \mathbb{R}^{d \times d}$ which makes its computation infeasible during reduced simulations. Thus, we propose to apply a MDEIM approximation, which not only reduces evaluation costs for the Jacobian but also allows an efficient offline/online decomposition of (2.21), which we will discuss in section 2.4. A similar idea named ‘‘Multi-Component EIM’’ has been formulated and successfully applied in [28, section 4.3.2]. Consequently, for any $\mathbf{A} \in \mathbb{R}^{d \times d}$ we define the transformation

$$\begin{aligned} \mathcal{V} : \mathbb{R}^{d \times d} &\rightarrow \mathbb{R}^{d^2}, \\ \mathbf{A} &\mapsto \mathcal{V}[\mathbf{A}] := (A_1^T, A_2^T, \dots, A_d^T)^T, \end{aligned}$$

which maps the matrix entries of \mathbf{A} columnwise into a vector (equivalent to the MATLAB operation $\mathbf{A}(:)$, also known as *vec*-operation).

PROPOSITION 2.13 (matrix DEIM). Choose $M_J \leq d$ and let $\hat{\mathbf{U}}_{M_J}, \hat{\mathbf{P}}_{M_J} \in \mathbb{R}^{d^2 \times M_J}$ denote the corresponding matrices for the DEIM basis and interpolation points obtained by application of the DEIM approximation procedure [5] to the vector-valued function $\mathcal{V}[\mathbf{J}(\mathbf{y})]$. Then, for $m_J \leq M_J$, the m_J th order MDEIM approximation of \mathbf{J} is given via

$$\tilde{\mathbf{J}}_{m_J}(\mathbf{y}) := \mathcal{V}^{-1} \left[\hat{\mathbf{U}}_{m_J} (\hat{\mathbf{P}}_{m_J}^T \hat{\mathbf{U}}_{m_J})^{-1} \hat{\mathbf{P}}_{m_J}^T \mathcal{V}[\mathbf{J}(\mathbf{y})] \right],$$

where $\hat{\mathbf{U}}_{m_J} := \hat{\mathbf{U}}_{M_J}(:, 1:m_J)$ and $\hat{\mathbf{P}}_{m_J} := \hat{\mathbf{P}}_{M_J}(:, 1:m_J)$.

With this method, no extra assumptions on \mathbf{f} need to be made that are not already required by the standard DEIM, as the pointwise evaluations of the Jacobian entries can always be approximated via finite differences and pointwise evaluation of the underlying \mathbf{f} . Of course, direct computation of Jacobian entries is preferred if possible. Moreover, the evaluation technique for reduced variables $\mathbf{V}\mathbf{z}(t)$ carries over directly as proposed in the original work [5, section 3.5]. Note that any matrix approximation technique using linear combinations of basis matrices could be applied in this context. See [4] for an approach using a POD-basis with least squares weights and structure-preserving constraints.

Now, using the above results from Propositions 2.12 and 2.13, we shall derive an efficient approximation of the estimator introduced in Theorem 2.9, where the computational complexity of the part (2.14) is independent of the high dimension d .

COROLLARY 2.14 (a posteriori estimate with local Jacobian logarithmic norms). *Let the conditions from Theorem 2.9 and Propositions 2.12 and 2.13 hold and replace (2.14) by*

$$(2.23) \quad \beta(t) := L_{I_k} [\mathbf{Q}_k^T \mathbf{C}^T \tilde{\mathbf{J}}_{m_J}(\mathbf{y}^r(t)) \mathbf{C}^{-T} \mathbf{Q}_k].$$

Then, the estimator (2.12) can be approximated efficiently and more accurately with increasing m_J and k . The first order approximation (2.17) of (2.14) is exactly reproduced for $m_J = d^2, k = d$.

Proof. We have $\mathbf{J} \equiv \tilde{\mathbf{J}}_{m_J}$ for $m_J = d^2$ as each entry of the Jacobian is interpolated. Together with (2.22) for $k = d$ we obtain $\beta(t) = L_G[\mathbf{J}(\mathbf{y}^r(t))]$. \square

Remark 2.15. An obvious alternative to approximate the Jacobian logarithmic norm is to use the reduced projected Jacobian $\mathbf{W}^T \mathbf{J}(\mathbf{y}^r(t)) \mathbf{V}$ and thus solve only an eigenvalue problem of size $r \ll d$. Since \mathbf{Q} does not have to be computed in this case, this might seem preferable. Our preliminary experiments employed this strategy with quite disappointing results, as the subspaces spanned by \mathbf{V}, \mathbf{W} are (generally) not designed to contain the eigenspace of the largest eigenvalues of the Jacobians. This in turn motivated the development of the partial similarity transformation introduced by Proposition 2.12.

2.4. Offline/online decomposition. The computations for the error estimator of Theorem 2.9 and Corollary 2.14 allow a decomposition into an offline/online fashion as already applied in many contexts [17, 33, 9]. We state the decomposition for the context of Corollary 2.14, as it includes the case of Theorem 2.9. From the reduction process, we are given a subspace projection matrix \mathbf{V} and the DEIM approximation (1.2) with corresponding maximal order M , basis \mathcal{U} , and interpolation points \mathcal{E} as in section 1. Moreover, for the offline computations of the $\alpha(t)$ term (2.13), we prefer the equivalent expression

$$(2.24) \quad \alpha(t) := \left\| \mathbf{M}_1 \mathbf{P}_m^T \mathbf{f}(\mathbf{y}^r(t)) - \mathbf{M}_2 \mathbf{P}_{m'}^T \mathbf{f}(\mathbf{y}^r(t)) \right\|_G,$$

with $\mathbf{M}_1, \mathbf{M}_2$ defined as

$$(2.25a) \quad \mathbf{M}_1 := \left((\mathbf{I} - \mathbf{V}\mathbf{W}^T) - \Pi_{m'} \right) \mathbf{U}_m \left(\mathbf{P}_m^T \mathbf{U}_m \right)^{-1},$$

$$(2.25b) \quad \mathbf{M}_2 := (\mathbf{I} - \Pi_m) \mathbf{U}_{m'} \left(\mathbf{P}_{m'}^T (\mathbf{I} - \Pi_m) \mathbf{U}_{m'} \right)^{-1}.$$

These expressions are easily verified by rearranging the expression for $\alpha(t)$ as

$$\alpha(t) := \left\| \left((\mathbf{I} - \mathbf{V}\mathbf{W}^T) - \Pi_{m'} \right) \Pi_m \mathbf{f}(\mathbf{y}^r(t)) + \Pi_{m'} \mathbf{f}(\mathbf{y}^r(t)) \right\|_G,$$

and applying the definitions of Π_m and $\Pi_{m'}$. Basically, this formulation allows us to nicely separate the evaluation of \mathbf{f} at both sets of interpolation points from pre-computable offline quantities. Then the following stages can be identified in order to compute the proposed error estimators:

Offline stage I: This has to be run only once for each system setting.

1. Choose a $M_J \leq d$ as maximum order and compute the Jacobian MDEIM as in Proposition 2.13, yielding the matrices $\hat{\mathbf{U}}_{M_J}, \hat{\mathbf{P}}_{M_J} \in \mathbb{R}^{d^2 \times M_J}$.
2. Compute $\mathbf{Q} \in \mathbb{R}^{d \times \min\{d, N\}}$ (or $\mathbf{Q} \in \mathbb{R}^{d \times K}$ for a truncated SVD with prescribed maximal order K) as in Proposition 2.12 or Algorithm 1.

Offline stage II: For every (new) choice of $m \leq M, m' \leq M - m, m_J \leq M_J, k \leq d$ (or $k \leq K$) perform the following steps:

1. Assemble the matrices $\mathbf{M}_1, \mathbf{M}_2$ as given by (2.25).
2. Compute the offline quantities for the $\alpha(t)$ term (2.24)

$$\mathbf{M}_3 := \mathbf{M}_1^T \mathbf{G} \mathbf{M}_1, \quad \mathbf{M}_4 := \mathbf{M}_1^T \mathbf{G} \mathbf{M}_2, \quad \mathbf{M}_5 := \mathbf{M}_2^T \mathbf{G} \mathbf{M}_2.$$

3. Compute offline vectors for the Jacobian MDEIM via

$$\hat{\mathbf{U}} := \hat{\mathbf{U}}_{m_J} (\hat{\mathbf{P}}_{m_J}^T \hat{\mathbf{U}}_{m_J})^{-1}, \quad \hat{\mathbf{U}}_{m_J} := \hat{\mathbf{U}}_{M_J}(:, 1:m_J), \quad \hat{\mathbf{P}}_{m_J} := \hat{\mathbf{P}}_{M_J}(:, 1:m_J).$$

4. Select partial similarity transform matrix of size k as $\mathbf{Q}_k := \mathbf{Q}(:, 1:k)$ and compute

$$\tilde{\mathbf{U}}(:, j) := \mathcal{V}_k [\mathbf{Q}_k^T \mathbf{C}^T \mathcal{V}^{-1} [\hat{\mathbf{U}}(:, j)] \mathbf{C}^{-T} \mathbf{Q}_k] \in \mathbb{R}^{k^2}, \quad j = 1, \dots, m_J,$$

where \mathcal{V}_k denotes the same transformation as \mathcal{V} but for $k \times k$ matrices.

Online stage: In the online stage we can compute the error estimator by solving

$$\Delta'(t) = \beta(t)\Delta(t) + \alpha(t), \quad \Delta(0) = \left\| (\mathbf{I} - \mathbf{V}\mathbf{W}^T) \mathbf{y}_0 \right\|_G$$

for $t \in [0, T]$, where in both cases the $\alpha(t)$ term (2.24) is given by

$$\alpha(t) = (\mathbf{v}_1^T \mathbf{M}_3 \mathbf{v}_1 - 2\mathbf{v}_1^T \mathbf{M}_4 \mathbf{v}_2 + \mathbf{v}_2^T \mathbf{M}_5 \mathbf{v}_2)^{\frac{1}{2}},$$

with the low-dimensional quantities

$$\mathbf{v}_1 := \mathbf{P}_{m'}^T \mathbf{f}(\mathbf{y}^r(t)) \in \mathbb{R}^m, \quad \mathbf{v}_2 := \mathbf{P}_{m'}^T \mathbf{f}(\mathbf{y}^r(t)) \in \mathbb{R}^{m'}.$$

Depending on the setting, use $\beta(t)$ as given in (2.14).

Notice that the computational complexity of the offline stages is $\mathcal{O}(d^2 N)$ and $\mathcal{O}(d)$, respectively, and the online stage has complexity $\mathcal{O}(\max\{k^3, m^2, m'^2, m_J^2\})$. We emphasize that the MDEIM approximation of the Jacobian not only gives a fast approximation, but is also crucial in order to apply the similarity transformation in the offline phase, which would in general not be possible if full Jacobians were used.

3. Error estimation for general, parametrized setting. The results so far have been developed around the relatively simple system (1.1). However, the methodology developed here can be readily transferred to a more general, parametrized setting similar to [17], which we shall now describe. All previous definitions remain valid unless explicitly indicated otherwise. We also need to introduce some additional notation. Let $\mathcal{P} \subset \mathbb{R}^p$ be a parameter domain, $T > 0$ and let $\mathbf{f} : \mathcal{D} \rightarrow \Omega$ be Lipschitz-continuous with respect to the first argument on the domain $\mathcal{D} := \Omega \times [0, T] \times \mathcal{P}$ for the remainder of this work. We shall consider the nonlinear dynamical system

$$(3.1) \quad \mathbf{y}'(t) = \mathbf{A}(t, \boldsymbol{\mu})\mathbf{y}(t) + \mathbf{f}(\mathbf{y}(t), t, \boldsymbol{\mu}) + \mathbf{B}(t, \boldsymbol{\mu})\mathbf{u}(t), \quad \mathbf{y}(0) = \mathbf{y}_0(\boldsymbol{\mu}),$$

where $\boldsymbol{\mu} \in \mathcal{P}$, and an input/control is given by $\mathbf{u} : [0, T] \rightarrow \mathbb{R}^l$. The linear part of the system is given by a time- and parameter-affine matrix

$$(3.2) \quad \mathbf{A}(t, \boldsymbol{\mu}) = \sum_{i=1}^{Q_A} \theta_i^A(t, \boldsymbol{\mu}) \mathbf{A}_i,$$

with $Q_A \in \mathbb{N}$ small, constant matrices $\mathbf{A}_i \in \mathbb{R}^{d \times d}$, and low-complexity scalar coefficient functions $\theta_i^A : [0, T] \times \mathcal{P} \rightarrow \mathbb{R}$. Analogous definitions hold for the components $\mathbf{B}(t, \boldsymbol{\mu})$ and $\mathbf{y}_0(\boldsymbol{\mu})$ with $\mathbf{B}_i \in \mathbb{R}^{d \times l}$, $i = 1, \dots, Q_B$, and $\mathbf{y}_i^0 \in \mathbb{R}^d$, respectively, where in the latter case only parameter-dependent functions $\theta_i^0(\boldsymbol{\mu})$, $i = 1, \dots, Q_0$, are used. Note here that (3.1) could of course also be merged into the setting (1.1) by considering $\mathbf{A} + \mathbf{f} + \mathbf{B}\mathbf{u}$ as one function, but the refined structure allows for more insight and thus numerical methods (e.g., solvers) to treat it.

Now, applying the (Petrov–)Galerkin projection with \mathbf{V}, \mathbf{W} and substituting the DEIM approximation of $\mathbf{f}(\mathbf{y}, t, \boldsymbol{\mu})$ to the system (3.1) yields

$$(3.3a) \quad \mathbf{z}'(t) = \sum_{i=1}^{Q_A} \theta_i^A(t, \boldsymbol{\mu}) \tilde{\mathbf{A}}_i \mathbf{z}(t) + \tilde{\mathbf{U}}_m \mathbf{P}_m^T \mathbf{f}_m(\mathbf{y}^r(t), t, \boldsymbol{\mu}) + \sum_{i=1}^{Q_B} \theta_i^B(t, \boldsymbol{\mu}) \tilde{\mathbf{B}}_i \mathbf{u}(t),$$

$$(3.3b) \quad \mathbf{z}(0) = \sum_{i=1}^{Q_0} \theta_i^0(\boldsymbol{\mu}) \tilde{\mathbf{y}}_i^0 =: \mathbf{z}_0(\boldsymbol{\mu}),$$

with reduced quantities as before and

$$\tilde{\mathbf{A}}_i := \mathbf{W}^T \mathbf{A}_i \mathbf{V}, \quad \tilde{\mathbf{B}}_i := \mathbf{W}^T \mathbf{B}_i, \quad \tilde{\mathbf{y}}_i^0 := \mathbf{W}^T \mathbf{y}_i^0.$$

Analogous to (2.11), we obtain from (3.1) and (3.3) an error system

$$(3.4) \quad \begin{aligned} \mathbf{e}'(t) &= \mathbf{A}(t, \boldsymbol{\mu})\mathbf{y}(t) - \mathbf{V}\mathbf{W}^T \mathbf{A}(t, \boldsymbol{\mu})\mathbf{V}\mathbf{z}(t) + \mathbf{f}(\mathbf{y}(t), t, \boldsymbol{\mu}) \\ &\quad - \mathbf{V}\mathbf{W}^T \tilde{\mathbf{f}}_m(\mathbf{y}^r(t), t, \boldsymbol{\mu}) + (\mathbf{I} - \mathbf{V}\mathbf{W}^T)\mathbf{B}(t, \boldsymbol{\mu})\mathbf{u}(t), \\ \mathbf{e}(0) &= \mathbf{y}_0(\boldsymbol{\mu}) - \mathbf{V}\mathbf{W}^T \mathbf{y}_0(\boldsymbol{\mu}), \end{aligned}$$

where we omit the implicit dependence of $\mathbf{e}(t)$ on the current parameter $\boldsymbol{\mu} \in \mathcal{P}$ for improved readability. Let $E_{y_0}(\boldsymbol{\mu}) := \|(\mathbf{I} - \mathbf{V}\mathbf{W}^T)\mathbf{y}_0(\boldsymbol{\mu})\|_G$ denote the initial error. Next we state our main a posteriori error estimation result from Theorem 2.9 in the context of the generalized system 3.1.

THEOREM 3.1 (a posteriori error estimation for parametrized DEIM reduced systems). *Let the conditions from Theorem 2.8 hold for $\mathbf{f}(\mathbf{y}(t), t, \boldsymbol{\mu})$. Then $\forall \boldsymbol{\mu} \in \mathcal{P}$ and $\mathbf{u}(t)$ the state-space error is bounded via*

$$\|\mathbf{e}(t; \boldsymbol{\mu}, \mathbf{u}(t))\|_G \leq \Delta(t, \boldsymbol{\mu}, \mathbf{u}) \quad \forall t \in [0, T],$$

with

$$\begin{aligned} \Delta(t, \boldsymbol{\mu}, \mathbf{u}) &:= \int_0^t \alpha(s, \boldsymbol{\mu}, \mathbf{u}) e^{\int_s^t \beta(\tau, \boldsymbol{\mu}) d\tau} ds + e^{\int_0^t \beta(\tau, \boldsymbol{\mu}) d\tau} E_{y_0}(\boldsymbol{\mu}), \\ \alpha(t, \boldsymbol{\mu}, \mathbf{u}) &:= \left\| (\mathbf{I} - \mathbf{V}\mathbf{W}^T) \mathbf{A}(t, \boldsymbol{\mu}) \mathbf{V} \mathbf{z}(t) \right. \\ &\quad \left. + \left(\Pi_{m'}(\mathbf{I} - \Pi_m) + (\mathbf{I} - \mathbf{V}\mathbf{W}^T) \Pi_m \right) \mathbf{f}(\mathbf{y}^r(t), t, \boldsymbol{\mu}) \right. \\ &\quad \left. + (\mathbf{I} - \mathbf{V}\mathbf{W}^T) \mathbf{B}(t, \boldsymbol{\mu}) \mathbf{u}(t) \right\|_G, \end{aligned} \quad (3.5)$$

$$(3.6) \quad \beta(t, \boldsymbol{\mu}) := L_G[\mathbf{A}(t, \boldsymbol{\mu})] + L_G[\mathbf{f}(\cdot, t, \boldsymbol{\mu})](\mathbf{y}^r(t)),$$

with $\Pi_m, \Pi_{m'}$ defined as in Lemma 2.7, (2.7). For efficient approximation of $\Delta(t, \boldsymbol{\mu}, \mathbf{u})$, replace $\beta(t, \boldsymbol{\mu})$ by

$$(3.7) \quad \beta(t, \boldsymbol{\mu}) = \sum_{i=1}^{Q_A} |\theta_i^A(t, \boldsymbol{\mu})| L_G[\mathbf{A}_i] + L_{I_k}[\mathbf{Q}_k^T \mathbf{C}^T \tilde{\mathbf{J}}_{m_J}(\mathbf{y}^r(t), t, \boldsymbol{\mu}) \mathbf{C}^{-T} \mathbf{Q}_k].$$

Proof. The proof is along the lines of the proof of Theorem 2.9. We introduce the notation

$$\tilde{\mathbf{A}}(t, \boldsymbol{\mu}) := (\mathbf{I} - \mathbf{V}\mathbf{W}^T) \mathbf{A}(t, \boldsymbol{\mu}) \mathbf{V}, \quad \tilde{\mathbf{B}}(t, \boldsymbol{\mu}) := (\mathbf{I} - \mathbf{V}\mathbf{W}^T) \mathbf{B}(t, \boldsymbol{\mu}),$$

and easily verify

$$(3.8) \quad \mathbf{A}(t, \boldsymbol{\mu}) \mathbf{y}(t) - \mathbf{V}\mathbf{W}^T \mathbf{A}(t, \boldsymbol{\mu}) \mathbf{V} \mathbf{z}(t) = \mathbf{A}(t, \boldsymbol{\mu}) \mathbf{e}(t) + \tilde{\mathbf{A}}(t, \boldsymbol{\mu}) \mathbf{z}(t).$$

Now, from (3.4) we obtain, using (3.8), (3.9), and (2.15), that

$$\begin{aligned} &\langle \mathbf{e}(t), \mathbf{e}'(t) \rangle_G \\ &= \langle \mathbf{e}(t), \mathbf{A}(t, \boldsymbol{\mu}) \mathbf{e}(t) \rangle_G + \langle \mathbf{e}(t), \mathbf{f}(\mathbf{y}(t), t, \boldsymbol{\mu}) - \mathbf{f}(\mathbf{y}^r(t), t, \boldsymbol{\mu}) \rangle_G \\ &\quad + \left\langle \mathbf{e}(t), \tilde{\mathbf{A}}(t, \boldsymbol{\mu}) \mathbf{z}(t) + \mathbf{f}(\mathbf{y}^r(t), t, \boldsymbol{\mu}) - \mathbf{V}\mathbf{W}^T \tilde{\mathbf{f}}_m(\mathbf{y}^r(t), t, \boldsymbol{\mu}) + \tilde{\mathbf{B}}(t, \boldsymbol{\mu}) \mathbf{u}(t) \right\rangle_G \\ &\leq L_G[\mathbf{A}(t, \boldsymbol{\mu})] \|\mathbf{e}(t)\|_G^2 + L_G[\mathbf{f}(\cdot, t, \boldsymbol{\mu})](\mathbf{y}^r(t)) \|\mathbf{e}(t)\|_G^2 \\ &\quad + \left\langle \mathbf{e}(t), \tilde{\mathbf{A}}(t, \boldsymbol{\mu}) \mathbf{z}(t) + \left(\Pi_{m'}(\mathbf{I} - \Pi_m) + (\mathbf{I} - \mathbf{V}\mathbf{W}^T) \Pi_m \right) \mathbf{f}(\mathbf{y}^r(t), t, \boldsymbol{\mu}) + \tilde{\mathbf{B}}(t, \boldsymbol{\mu}) \mathbf{u}(t) \right\rangle_G \\ &\leq \beta(t, \boldsymbol{\mu}) \|\mathbf{e}(t)\|_G^2 + \|\mathbf{e}(t)\|_G \alpha(t, \boldsymbol{\mu}). \end{aligned}$$

Application of the comparison lemma (Lemma 2.6) as before completes the proof for $\Delta(t, \boldsymbol{\mu}, \mathbf{u})$.

Finally, following the argumentation of Corollary 2.14, efficient approximations of $\Delta(t, \boldsymbol{\mu}, \mathbf{u})$ can be obtained using (2.16), Propositions 2.12 and 2.13, and

$$(3.9) \quad L_G[\mathbf{A}(t, \boldsymbol{\mu})] = \sup_{\mathbf{x} \in \mathbb{R}^d \setminus \{0\}} \sum_{i=1}^{Q_A} \theta_i^A(t, \boldsymbol{\mu}) \frac{\langle \mathbf{x}, \mathbf{A}_i \mathbf{x} \rangle_G}{\|\mathbf{x}\|_G^2} \leq \sum_{i=1}^{Q_A} |\theta_i^A(t, \boldsymbol{\mu})| L_G[\mathbf{A}_i],$$

where $L_G[\mathbf{A}_i]$ can be efficiently precomputed. \square

Remark 3.2. As this paper deals with state-space error estimation, systems with outputs are not considered here. However, if outputs $\mathbf{w}(t) = \mathbf{C}(t, \boldsymbol{\mu}) \mathbf{y}(t)$ with affine $\mathbf{C}(t, \boldsymbol{\mu})$ in the fashion of (3.2) or [17] are present, the estimation of the output error can straightforwardly be done using Theorem 3.1 via

$$e_y(t; \boldsymbol{\mu}, \mathbf{u}) \leq \sum_{i=1}^{Q_C} |\theta_i^C(t, \boldsymbol{\mu})| \|\mathbf{C}_i\| \Delta(t, \boldsymbol{\mu}, \mathbf{u}), \quad t \in [0, T].$$

3.1. Offline/online decomposition for generalized setting. The estimator of Theorem 3.1 can also be fully decomposed in an offline/online fashion. In addition to the steps already introduced in section 2.4, the following computations have to be performed. We assume the indices i, j to run over the appropriate ranges given by Q_A, Q_B , and Q_0 in both offline stages.

Offline stage I: Compute offline terms for the components as

$$\begin{aligned}\tilde{\mathbf{y}}_i^0 &:= (\mathbf{I} - \mathbf{VW}^T)\mathbf{y}_i^0, & \tilde{\mathbf{y}}_{ij}^0 &:= (\tilde{\mathbf{y}}_i^0)^T \mathbf{G} \tilde{\mathbf{y}}_j^0, & \tilde{\mathbf{A}}_i &:= (\mathbf{I} - \mathbf{VW}^T) \mathbf{A}_i \mathbf{V}, \\ \tilde{\mathbf{B}}_i &:= (\mathbf{I} - \mathbf{VW}^T) \tilde{\mathbf{B}}_i, & \mathbf{M}_{6,ij} &:= \tilde{\mathbf{A}}_i^T \mathbf{G} \tilde{\mathbf{A}}_j & \mathbf{M}_{9,ij} &:= \tilde{\mathbf{A}}_i^T \mathbf{G} \tilde{\mathbf{B}}_j, \\ \mathbf{M}_{12,ij} &:= \tilde{\mathbf{B}}_i^T \mathbf{G} \tilde{\mathbf{B}}_j.\end{aligned}$$

Offline stage II: Compute additional offline quantities for the $\alpha(t)$ term (3.5):

$$\mathbf{M}_{7,i} := \mathbf{M}_1^T \mathbf{G} \tilde{\mathbf{A}}_i, \quad \mathbf{M}_{8,i} := \mathbf{M}_2^T \mathbf{G} \tilde{\mathbf{A}}_i, \quad \mathbf{M}_{10,i} := \mathbf{M}_1^T \mathbf{G} \tilde{\mathbf{B}}_i, \quad \mathbf{M}_{11,i} := \mathbf{M}_2^T \mathbf{G} \tilde{\mathbf{B}}_i.$$

Online stage: Compute $E_{y_0}(\mu) = (\sum_{i,j}^{Q_0} \theta_i^0(\mu) \theta_j^0(\mu) \tilde{\mathbf{y}}_{ij}^0)^{\frac{1}{2}}$, and (3.5) is given by

$$\begin{aligned}\alpha(t, \mu) &= \left(\mathbf{v}_1^T \mathbf{M}_3 \mathbf{v}_1 - 2\mathbf{v}_1^T \mathbf{M}_4 \mathbf{v}_2 + \mathbf{v}_2^T \mathbf{M}_5 \mathbf{v}_2 + \mathbf{z}(t)^T \mathbf{M}_6(t, \mu) \mathbf{z}(t) \right. \\ &\quad + 2\mathbf{v}_1 \mathbf{M}_7(t, \mu) \mathbf{z}(t) - 2\mathbf{v}_2 \mathbf{M}_8(t, \mu) \mathbf{z}(t) + 2\mathbf{z}(t)^T \mathbf{M}_9(t, \mu) \mathbf{u}(t) \\ &\quad \left. + 2\mathbf{v}_1 \mathbf{M}_{10}(t, \mu) \mathbf{u}(t) - 2\mathbf{v}_2 \mathbf{M}_{11}(t, \mu) \mathbf{u}(t) + \mathbf{u}(t)^T \mathbf{M}_{12}(t, \mu) \mathbf{u}(t) \right)^{\frac{1}{2}},\end{aligned}$$

with the additional low-dimensional quantities

$$\begin{aligned}\mathbf{M}_6(t, \mu) &:= \sum_{i,j=1}^{Q_A} \theta_i^A(t, \mu) \theta_j^A(t, \mu) \mathbf{M}_{6,ij}, & \mathbf{M}_{\{7,8\}}(t, \mu) &:= \sum_{i=1}^{Q_A} \theta_i^A(t, \mu) \mathbf{M}_{\{7,8\},i}, \\ \mathbf{M}_9(t, \mu) &:= \sum_{i=1}^{Q_A} \sum_{j=1}^{Q_B} \theta_i^A(t, \mu) \theta_j^B(t, \mu) \mathbf{M}_{9,ij}, & \mathbf{M}_{\{10,11\}}(t, \mu) &:= \sum_{i=1}^{Q_B} \theta_i^B(t, \mu) \mathbf{M}_{\{10,11\},i}, \\ \mathbf{M}_{12}(t, \mu) &:= \sum_{i,j=1}^{Q_B} \theta_i^B(t, \mu) \theta_j^B(t, \mu) \mathbf{M}_{12,ij}.\end{aligned}$$

At this point, all of the offline quantities required by the estimator have been described. Moreover, the error estimate may be computed along with the reduced order trajectory by adjoining a single scalar equation to the system as described in section 2.4. In the next section we illustrate the practicality of this estimate with several numerical experiments.

4. Numerical experiments. We shall demonstrate the effectiveness of the derived a posteriori error estimation procedure of Theorem 3.1 with two computational examples. The first involves a one-dimensional (1D) viscous Burger's equation and the second involves a more realistic two-dimensional (2D) reaction-diffusion model for cell apoptosis. With these examples, we demonstrate the validity of the assumption of near exactness for modest values of m' (see section 2.1). We also show the effectiveness of the local logarithmic Lipschitz constant estimates (see section 2.3) and we compare our estimate based on inexact quantities to one that uses exact quantities. The estimator has performed quite well in these experiments.

4.1. 1D viscous Burgers' equation. In this section we will investigate different aspects of the error estimation process. We first consider a parametrized 1D unsteady viscous Burgers' equation over the unit interval $\Omega := [0, 1]$ and time $t \in [0, T]$ with $T = 1$:

$$(4.1) \quad \frac{\partial y}{\partial t}(x, t) = \mu \frac{\partial^2 y}{\partial x^2}(x, t) - \frac{\partial}{\partial x} \left(\frac{y(x, t)^2}{2} \right) + \langle \mathbf{b}(x), \mathbf{u}(t) \rangle,$$

with diffusion coefficient $\mu \in \mathcal{P} := [0.01, 0.06]$ and homogeneous initial and Dirichlet boundary conditions. Further there are external forces $\mathbf{u}(t)$ at locations $\mathbf{b}(x) = (b_1(x), b_2(x))^T$ given by

$$\begin{aligned} u_1(t) &= \sin(2\pi t), & b_1(x) &= \begin{cases} 4e^{-(\frac{x-0.2}{0.03})^2}, & x \in [0.1, 0.3], \\ 0 & \text{else,} \end{cases} \\ u_2(t) &= \begin{cases} 1, & t \in [0.2, 0.4], \\ 0 & \text{else,} \end{cases} & b_2(x) &= \begin{cases} 4, & x \in [0.6, 0.7], \\ 0 & \text{else.} \end{cases} \end{aligned}$$

The component $b_1(x)u_1(t)$ realizes an oscillating excitation centered at $x = 0.2$ in the shape of a Gaussian curve, and $b_2(x)u_2(t)$ gives a discontinuous signal over a limited time in the interval $[0.6, 0.7]$. Next, spatial discretization of the model via finite differences yields a system of $d = 500$ ordinary differential equations

$$(4.2) \quad \frac{d}{dt} \mathbf{y}(t) = \mu \mathbf{A} \mathbf{y}(t) + \mathbf{f}(\mathbf{y}(t)) + \mathbf{B} \mathbf{u}(t),$$

where $\mathbf{A} \in \mathbb{R}^{d \times d}$ is the discrete Laplacian and $\mathbf{B} \in \mathbb{R}^{d \times 2}$. Further we have $\mathbf{f}(\mathbf{y}) = -\mathbf{y}.*\mathbf{A}_y \mathbf{y}$ with first order central finite difference operator $\mathbf{A}_y \in \mathbb{R}^{d \times d}$. Here “.*” denotes elementwise multiplication. We choose $\mathbf{G} = \mathbf{I}_d$ (L^2 -norm in state-space), and the time integration of (4.2) is performed via a semi-implicit Euler scheme

$$(\mathbf{I} - \Delta t \mu \mathbf{A}) \mathbf{y}(t_{i+1}) = \mathbf{y}(t_i) + \Delta t (\mathbf{f}(\mathbf{y}(t_i)) + \mathbf{B} \mathbf{u}(t_i)),$$

on $n_t = 100$ equidistant time-steps $t_i := (i - 1)\Delta t$, $\Delta t = \frac{T}{n_t - 1}$, $i = 1, \dots, n_t$.

Figure 4.1 shows solutions of the system (4.1) for minimal (left) and maximal (middle) μ value in \mathcal{P} . The rightmost plot is the difference of both previous solutions and illustrates the behavioral change of the model over \mathcal{P} . In order to compute the reduced basis, we chose a discrete parameter set $\Xi \subseteq \mathcal{P}$ of 100 log-equidistant values to generate the training trajectories. The state-space projection matrices

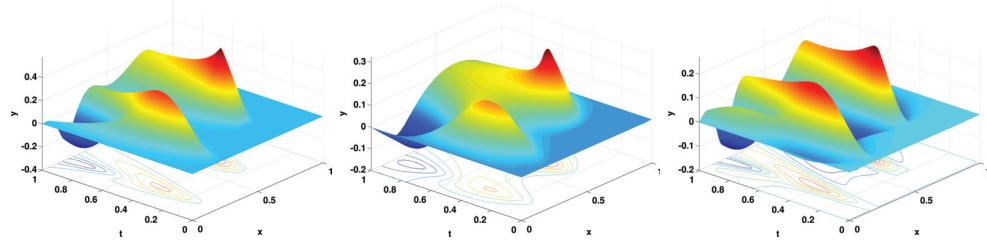


FIG. 4.1. Simulation results for minimal (left), maximal (middle) parameter values in \mathcal{P} , and their difference (right).

$\mathbf{V} = \mathbf{W}$ were obtained via the POD-Greedy algorithm [16] with maximum subspace size 100. However, in this experiment we computed the true maximum L^2 state-space error over all training trajectories $T_y^\mu := \{\mathbf{y}(t_1; \mu), \dots, \mathbf{y}(t_{n_t}; \mu)\}$ in each extension step, where each trajectory data was augmented by \mathbf{f} -evaluations $T_f^\mu := \{\mathbf{f}(\mathbf{y}(t_1; \mu)), \dots, \mathbf{f}(\mathbf{y}(t_{n_t}; \mu))\}, \mu \in \Xi$. We further used the span of the \mathbf{B} -matrix columns as an initial space, yielding a total maximum error $\max_{\mu \in \Xi} \max_{\mathbf{y} \in T_y^\mu \cup T_f^\mu} \|\mathbf{y} - \mathbf{V}\mathbf{V}^T \mathbf{y}\| \leq 1.5099 \times 10^{-6}$. Inclusion of T_f^μ and the \mathbf{B} -span in \mathbf{V} is not necessary for good reduced trajectories, but enables higher error estimator precision due to the componentwise projection of the system components at the offline stage; see sections 2.4 and 3.1. The DEIM approximation basis for $\tilde{\mathbf{f}}$ was obtained by performing a POD with output size $M = 200$ on $T_f := \bigcup_{\mu \in \Xi} T_f^\mu$. This gives a maximum relative error $\max_{\mathbf{y} \in T_f} \|\mathbf{f}(\mathbf{y}) - \tilde{\mathbf{f}}_M(\mathbf{y})\| / \|\mathbf{f}(\mathbf{y})\| \approx 3.15 \times 10^{-11}$, which we consider to practically satisfy our assumption $\tilde{\mathbf{f}}_M \equiv \mathbf{f}$ of Theorem 2.8. Figure 4.2 shows full, reduced simulation and the absolute error for $\mu = 0.04 \notin \Xi$ and DEIM approximation order $m = 12$. For this setting, the reduced model already yields visually indistinguishable results with a maximum relative error of 0.0062. In order to focus our discussion on the influences of the various choices m' , m_J , and k , we shall fix $m = 12$ and $\mu = 0.04$ for the remainder of the experiments. However, we emphasize that the error estimators are applicable for any choice of $m < M$, $m' \leq M - m$, $\mu \in \mathcal{P}$, and input $\mathbf{u}(t)$.

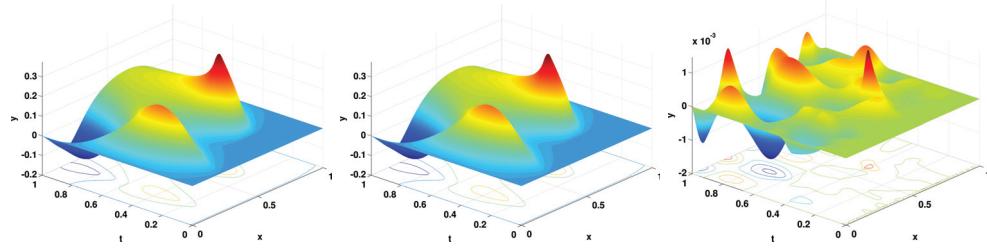


FIG. 4.2. Full (left), reduced simulation (middle), and the absolute error (right) for $\mu = 0.04$.

Next, we will discuss the effects for different m' values in section 4.2 and focus on the local logarithmic Lipschitz constant estimation in sections 4.3 and 4.4.

4.2. DEIM approximation error estimation analysis. In the estimator structure from Theorem 3.1, different choices of m' influence the α term (3.5). As this is independent of its embedding into the error estimation process for reduced systems, we first investigate the DEIM error estimation quality for different m, m' choices. As test data we use the snapshot training data $T_y := \bigcup_{\mu \in \Xi} T_y^\mu$ and compare against the true error given via T_f . For this, Figures 4.3 and 4.4 display true and estimated DEIM approximation errors, using L^2 in state-space and L^∞ over T_y . The left plot of Figure 4.3 shows the true error decay for increasing m on T_y . The middle plot shows the estimated error via (2.10) over the same DEIM orders for all remaining possible m' values (i.e., $m < m' \leq M$), plotted in an overlay. This shows that the estimation is very closely following the true DEIM error, independently of the current m' choice. In fact, the visible deviations are all essentially caused by the first $1 \leq m' \leq 4$ values. Finally, the rightmost plot of Figure 4.3 shows the singular values of the POD on T_f to obtain the DEIM basis \mathcal{U} . The change of decay rate indicates that the main dynamics of \mathbf{f} are captured at around 140, which matches with the true error decay observable in the leftmost plot. In order to give more insight, the left plot of Figure 4.4 displays

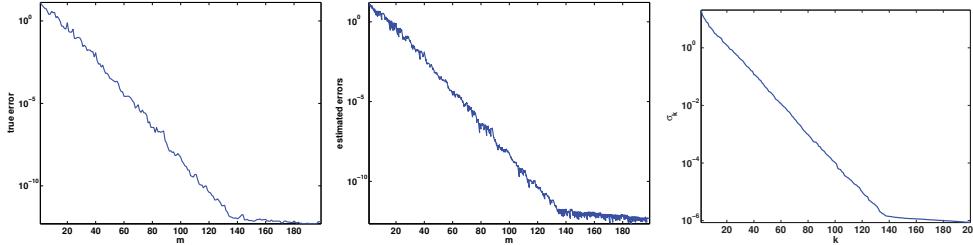


FIG. 4.3. True (left) and estimated (middle) absolute DEIM approximation errors. Right: Singular value decay of POD on T_f .

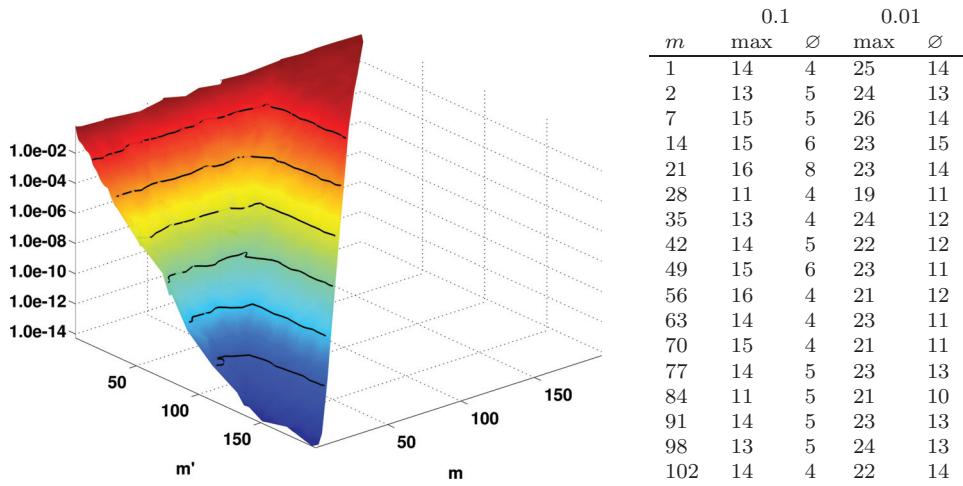


FIG. 4.4. Mean relative error between true and estimated DEIM approximation error over T_y .

the mean relative error

$$\text{mean}_{\mathbf{y} \in T_y} \|\mathbf{f}(\mathbf{y}) - \tilde{\mathbf{f}}(\mathbf{y}) - (\mathbf{M}_1 \mathbf{P}_m^T - \mathbf{M}_2 \mathbf{P}_{m'}^T) \mathbf{f}(\mathbf{y})\| / \|\mathbf{f}(\mathbf{y}) - \tilde{\mathbf{f}}(\mathbf{y})\|$$

between the true and estimated DEIM approximation errors on T_y . The contours are located at the levels 10^{-2} to 10^{-12} . The “bends” of the contours in the image are where $m + m' \approx 130$, meaning that no real estimation improvement can be achieved with higher m' values. This is in accordance with the stagnating decay of the singular values as shown in the right plot of Figure 4.3, as no essential new information is covered using larger m' values. Thus, further comments on the figure focus on the area where $m + m' \leq 130$. Most importantly, the estimation accuracy for increasing m' values at fixed m is improving exponentially. On the other hand, it is interesting to see that the contours are basically straight lines on each level, which means that for fixed m' the same error estimation precision is achieved for all m values.

To provide some values of the plot on the left-hand side in Figure 4.4, the table on the right shows how large m' must be chosen in order to obtain 10% or 1% maximum or mean relative error on T_y , respectively, for different DEIM orders m . The mean values on the 0.01 column are corresponding to points on the contour of the left plot for $1 \leq m \leq 102$, e.g., the contour is located around $m' = 14$, which is sufficient (in average) to ensure a relative error estimation error of less than 1%. In summary, this

illustrates that a very good estimation of the actual DEIM error is possible in our experiment.

The above experiments show that the DEIM approximation error estimation may indeed be very useful in practical applications. However, the above analysis so far gives insight to the applicability of the m, m' -estimation technique by itself but does not consider its application in the context of the error estimator derived in Theorem 3.1. In order to exclusively focus on this aspect, we used a modified variant of the estimator which avoids using MDEIM approximated Jacobians and partial similarity transformations in (3.7) but still uses directly computable local Lipschitz values. Therefore, during the next experiment we replaced (3.7) by

$$(4.3) \quad \beta(t, \mu) := \mu L_G[A] + \frac{\langle \mathbf{y}(t) - \mathbf{y}^r(t), \mathbf{f}(\mathbf{y}(t)) - \mathbf{f}(\mathbf{y}^r(t)) \rangle_G}{\|\mathbf{y}(t) - \mathbf{y}^r(t)\|_G^2}.$$

Moreover, we introduced a reference estimate by additionally using the true DEIM approximation error $\mathbf{f}(\mathbf{y}^r(t)) - \mathbf{V}\mathbf{W}^T \tilde{\mathbf{f}}_m(\mathbf{y}^r(t))$ instead of $(\mathbf{M}_1 \mathbf{P}_m^T - \mathbf{M}_2 \mathbf{P}_{m'}^T) \mathbf{f}(\mathbf{y}^r(t))$ within the $\alpha(t, \mu, u)$ term (3.5). Those two changes make this reference estimator as sharp as this estimator structure allows. Of course neither of the above modifications are applicable in practice as they both require the full system's trajectory. They are introduced here for demonstration purposes only.

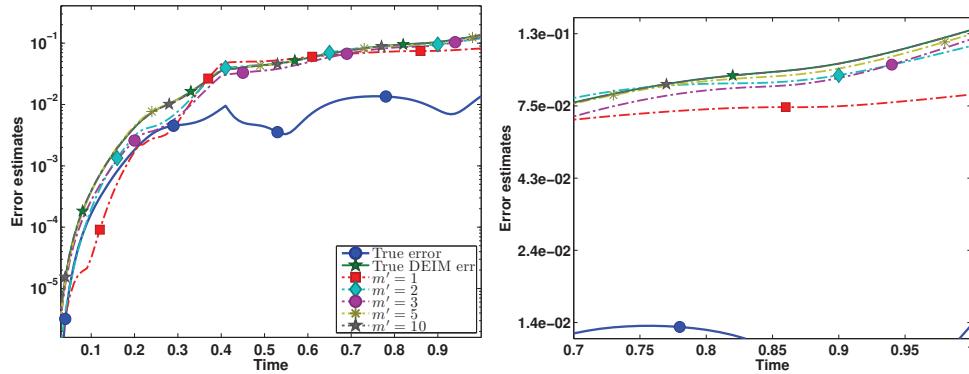


FIG. 4.5. Absolute errors over time for different DEIM m' approximation error orders.

Finally, Figure 4.5 shows the resulting error estimations for a selection of m' approximation error orders in a full view (left) and a zoom (right). The solid blue line is the true reduction error, and the green star-marked line is the reference estimate using the true DEIM approximation error. For the reduced simulation we have $\|e(T)\|_G = 0.0137$ and the rigorous reference estimate yields $\Delta(T) = 0.135$, which means an effectivity of 9.89 for the current $\mu = 0.04$. Most importantly, this illustrates that the overall estimator structure is suitable for error estimation. Now, regarding the m' influence, at first we notice that except for $m' = 1$ the reduction error estimates using the m' -order DEIM error estimate are very close to the reference estimate, where $m' = 1$ also is the only setting which is not an upper bound as the true error is underestimated at $t \in [0.05, 0.2]$. Moreover, for $m' = 10$ the results are already almost indistinguishable from the reference estimate even in the zoomed detail. As this is already quite satisfactory, we set $m' = 10$ for all subsequent experiments.

* All colors in figures appear in the electronic version only.

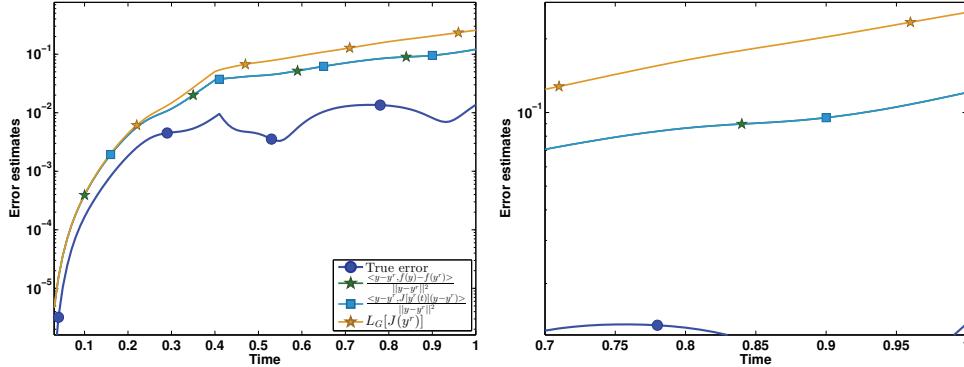


FIG. 4.6. Absolute errors over time with full local Jacobian logarithmic norms.

4.3. Local logarithmic Lipschitz constant estimations using Jacobians.

Our next experiment illustrates the occurring loss in accuracy when using the first order approximation (2.17) in (4.3). Figure 4.6 displays the resulting estimates in full (left) and zoom (right). Here, the green * star-marked line again represents the reference estimate (4.3) and the orange star-marked line uses the estimation via the logarithmic norm of the local Jacobian (2.17). As an intermediate estimate, the blue square-marked estimate uses

$$\beta(t, \mu) = \mu L_G[A] + \frac{\langle y(t) - y^r(t), J(y^r(t))(y(t) - y^r(t)) \rangle_G}{\|y(t) - y^r(t)\|_G^2}.$$

This shows two things: At first, the estimation results for either using $f(y(t)) - f(y^r(t))$ or $J(y^r(t))(y(t)) - y^r(t)$ inside the $\beta(t)$ computation are visibly indistinguishable and match on the first three digits ($\Delta(T) = 0.1354$), which makes the first order approximation already very useful even when the true error is about 1%. Second, using $L_G[J(y^r(t))]$ yields a very similar estimate with $\Delta(T) = 0.308$, which is roughly twice as large as the sharp reference with an effectivity (ratio of estimated to true error) of ≈ 22 . This is of course a loss in accuracy, but the estimate is obtained without the need for the full system's solution. On the downside, the computation time for the first reference estimate was $0.43s$, but the estimate involving $L_G[J(y^r(t))]$ took $28.27s$ to compute. This leaves us with dealing with the possibly high computation cost for the logarithmic norm of the Jacobians, which we address in the next section.

4.4. Jacobian logarithmic norm approximation. In Theorem 3.1, the quality of the efficient replacement (3.7) for (3.6) is influenced by different choices for m_J (Jacobian MDEIM approximation order) and k (partial similarity transformation size). Similar to section 4.2, the approximation quality of the involved terms $L_{I_k}[\mathbf{Q}_k^T \mathbf{C}^T \tilde{\mathbf{J}}_{m_J}(y^r(t), t, \mu) \mathbf{C}^{-T} \mathbf{Q}_k]$ versus $L_G[J(y^r(t), t, \mu)]$ can be investigated outside the scope of the error estimation process.

First, we analyze the singular value decay of the SVD used to obtain the partial similarity transformation matrix \mathbf{Q} as introduced by Theorem 2.11 and Proposition 2.12, which is shown in Figure 4.7. We used $N = 10100$ eigenvector snapshots for

* All colors in figures appear in the electronic version only.

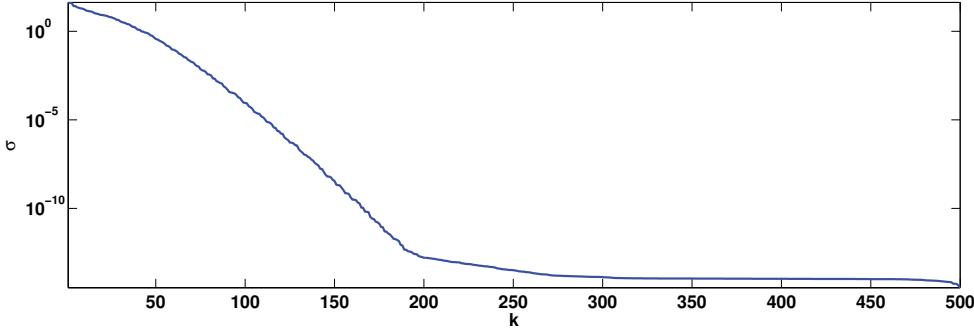


FIG. 4.7. Singular value decay of the eigenvector correlation matrix \mathbf{R} (respectively, using its discrete representation \mathbf{S}_N).

\mathbf{S}_N , and similar to Figure 4.3, we observe an exponential decay, in this case down to about 10^{-13} for $k = 180$. Besides showing that a suitable subspace for eigenvector approximation can be found, we shall see in the next experiment that even much smaller values of k are suitable for accurate logarithmic norm approximation.

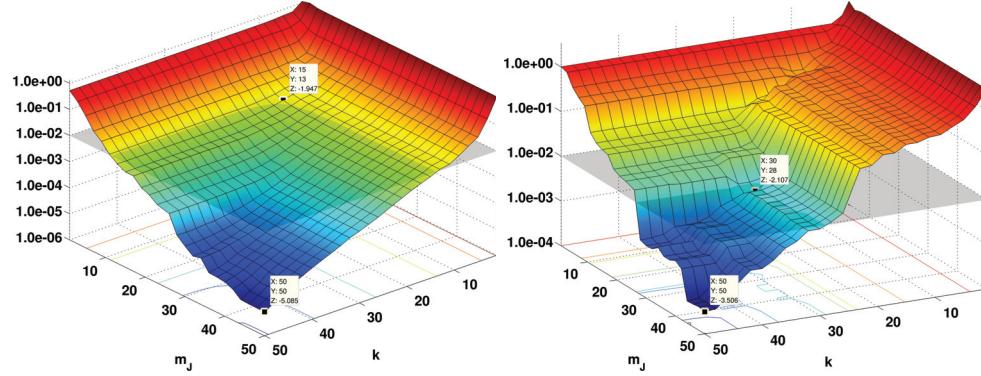


FIG. 4.8. Maximum (left) and mean (right) logarithmic norm approximation error over T_y .

Thus, Figure 4.8 shows the maximum (left) and mean (right) logarithmic norm approximation error over T_y for different m_J and k values. While values of $m_J = 13, k = 15$ are sufficient on average to have less than 1% relative error, in the worst case $m_J = 28, k = 30$ are needed to ensure the same tolerance over T_y . But even for maximal $m_J = k = 50$, the relative errors are already only 3.162×10^{-4} (worst case) and 8.222×10^{-6} (average). These results strongly indicate that both performing the similarity transformation and MDEIM approximation of Propositions 2.12 and 2.13, respectively, are suitable for a low-cost but accurate approximation of Jacobian logarithmic norms.

Finally, we will look into the estimation quality for different configurations of the practically applicable error estimator from Theorem 3.1, which involves the previously investigated approximations. Recall that the results of section 4.2 led us to fix $m' = 10$ for the following experiments. Figure 4.9 shows the absolute errors over time for different MDEIM orders m_J and partial similarity transformation sizes k in full (left) and zoom (right). Both star-marked estimations denote the already known reference estimates. In all other plots, the line style is identical for different m_J values and

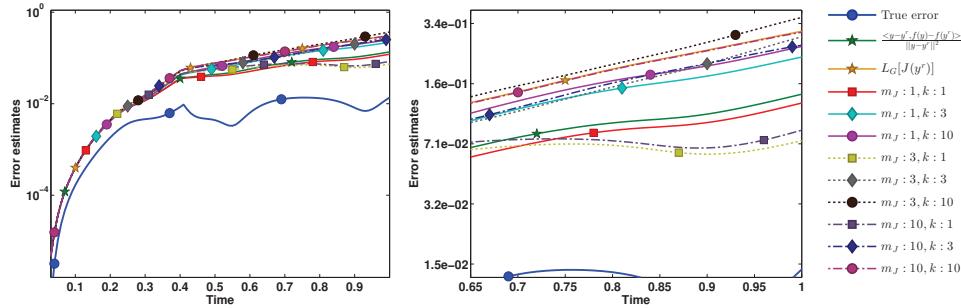


FIG. 4.9. Absolute errors over time for different MDEIM orders m_J and partial similarity transformation sizes k .

the marker style identical for different k values. One can clearly see that for both increased m_J and k the estimations get closer to the orange^{*} reference as expected. For $m_J = k = 10$ we already obtain $\Delta(T) = 0.3045$, which matches the orange reference up to two digits but needs only a computation time of $0.48s$ (speedup of 58.8). On closer examination, one can see that estimates with the same k are relatively similar and approach the orange reference from below with increasing k . This is due to the fact that for any $\mathbf{J} \in \mathbb{R}^{d \times d}$

$$L_{I_k}[\mathbf{Q}_k^T \mathbf{J} \mathbf{Q}_k] = \max_{\mathbf{z} \in \mathbb{R}^k} \frac{\langle \mathbf{z}, \mathbf{Q}_k^T \mathbf{J} \mathbf{Q}_k \mathbf{z} \rangle}{\|\mathbf{z}\|^2} = \max_{\mathbf{y} \in \text{range}(\mathbf{Q}_k)} \frac{\langle \mathbf{y}, \mathbf{J} \mathbf{y} \rangle}{\|\mathbf{y}\|^2} \leq \max_{\mathbf{y} \in \mathbb{R}^d} \frac{\langle \mathbf{y}, \mathbf{J} \mathbf{y} \rangle}{\|\mathbf{y}\|^2} = L_{I_d}[\mathbf{J}];$$

i.e., the maximum eigenvalues of the similarity transformed Jacobians are bounded by the maximum eigenvalues of the full Jacobians as $\text{range}(\mathbf{Q}_k) \subseteq \mathbb{R}^d$.

4.5. 2D reaction-diffusion model for cell apoptosis. As no significant speed-up is to be expected for most 1D cases, we considered a more realistic problem on a 2D domain $\Omega = [0, 1] \times [0, 1.5]$ next. The model of choice is a reaction-diffusion system originally introduced for 1D in [8] in order to study the dynamical behavior of protein concentrations of a cell apoptosis model in space and time. Due to the large variety of possible choices, a comprehensive analysis cannot be carried out within this paper. Instead, we chose to focus on aspects related to our error estimation procedure. The involved proteins build a network called “caspase cascade,” consisting of four different reactants x_i, y_i, x_a, y_a called caspase-8, caspase-3, procaspase-8, and procaspase-3, respectively. Their interaction is modeled by the system

$$\begin{aligned} \frac{\partial x_a}{\partial t} &= k_{c1}x_i y_a - k_{d1}x_a + D_1 \Delta x_a, & \frac{\partial y_a}{\partial t} &= k_{c2}y_i x_a^2 - k_{d2}y_a + D_2 \Delta y_a, \\ \frac{\partial x_i}{\partial t} &= -k_{c1}x_i y_a - k_{d3}x_i + k_{p1} + D_3 \Delta x_i, & \frac{\partial y_i}{\partial t} &= -k_{c2}y_i x_a^2 - k_{d4}y_i + k_{p2} + D_4 \Delta y_i, \end{aligned}$$

where k_* and D_i are suitably scaled constants controlling creation, interaction, and diffusion of the involved quantities that are detailed in [8]. Furthermore, the procaspase-8 gets activated by receptors located in the cellular membrane, which naturally leads to geometrically parameter-dependent boundary conditions. Consequently, we introduce

* All colors in figures appear in the electronic version only.

for $\mu_1 \in [0, 1]$ a boundary part $\Gamma_{\mu_1} := \{\mathbf{x} \in \partial\Omega \mid |x_1 - 0.5| \leq 0.5\mu_1 \vee |x_2 - 0.75| \leq 0.75\mu_1\} \subseteq \partial\Omega$ on which we impose the Neumann conditions

$$\frac{\partial x_a}{\partial n} \Big|_{\Gamma_{\mu_1}} = -\frac{\partial x_i}{\partial n} \Big|_{\Gamma_1} = \mu_2 x_i,$$

where $\mu_2 \in [10^{-5}, 10^{-2}]$ describes the reaction rate of the activation of procaspase-8. On all other parts of $\partial\Omega$ we enforce homogeneous Neumann conditions and have $x_a = y_a = 10^{-2}$, $x_i = y_i = 0.01$ as initial conditions.

Next, discretization on a 100×150 grid leads to a discrete system

$$\mathbf{y}'(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{f}(\mathbf{y}(t), \boldsymbol{\mu}),$$

with $d = 60000$ -dimensional state-space $\mathbf{y}(t) = (\mathbf{x}_a, \mathbf{y}_a, \mathbf{x}_i, \mathbf{y}_i)^T$, discrete Laplacian \mathbf{A} , and nonlinear reaction operator \mathbf{f} . We simulate up to $T = 500s$ and again discretize in time using a semi-implicit Euler with $\Delta t = 5s$. For model reduction we apply the DEIM procedure on \mathbf{f} with $M = 200$ and choose 200 random parameters $\Xi \subset \mathcal{P} = [0, 1] \times [10^{-5}, 10^{-2}]$ to obtain training trajectories $T_{\mathbf{y}} := \{T_{\mathbf{y}}^{\boldsymbol{\mu}} \mid \boldsymbol{\mu} \in \Xi\}$. The projection subspace \mathbf{V} was generated by a POD-Greedy procedure on $T_{\mathbf{y}}$ augmented with evaluations of \mathbf{f} and \mathbf{A} on $T_{\mathbf{y}}$ similar to the previous experiment. Running up to an error tolerance of 10^{-6} the resulting subspace is of dimension 282, a reduction by a factor of ≈ 212 . In order to compute the MDEIM and similarity transformations, we used 20200 uniformly chosen samples of $T_{\mathbf{y}}$ and set $M_J = 200, k_{max} = 50$. Figure 4.10 shows the simulation results on the middle slice $[0.5] \times [0, 1.5] \subset \Omega$ over time for $\boldsymbol{\mu} = (0.777, 0.00132)^T \notin \Xi$. The upper images show the concentrations and the corresponding lower images the absolute errors of full versus reduced solution, where a DEIM order of $m = 107$ has been used. The resulting relative error on the caspase-3/8 concentrations is about 1%, while the relative error for the procaspase concentrations is roughly at the order of 10^{-5} . The simulation time for the full model was $\approx 11s$, while the reduced model simulation (excluding error estimation) takes $0.3864s$ (averaged over 10 runs). This is a speedup factor of roughly 28, which is acceptable keeping in mind the threshold introduced by the natural overhead regarding the simulation procedure. Next, Figure 4.11 shows error estimation results for different configurations of the error estimator from Theorem 3.1. As in section 4.1, we included reference estimates using the true local logarithmic Lipschitz constants of both \mathbf{f} and the

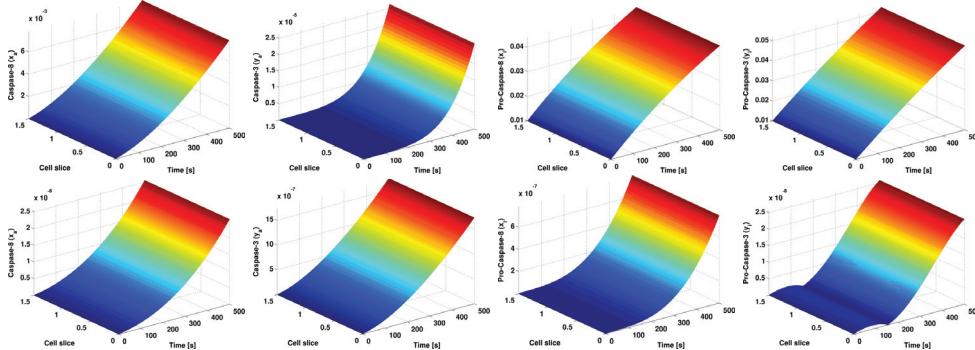
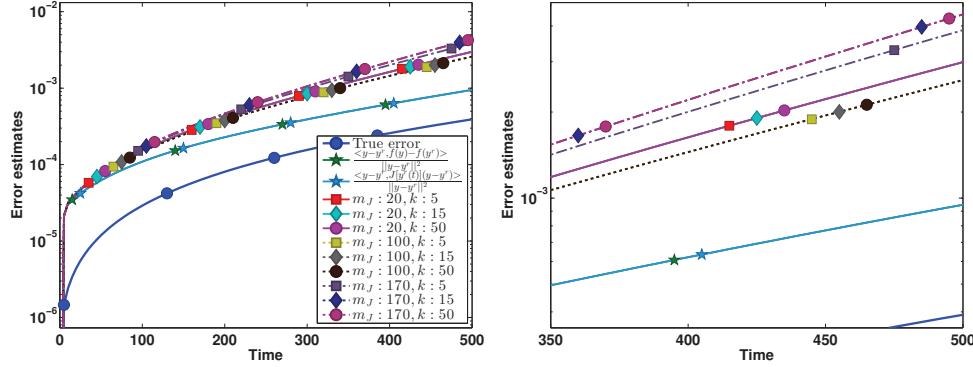
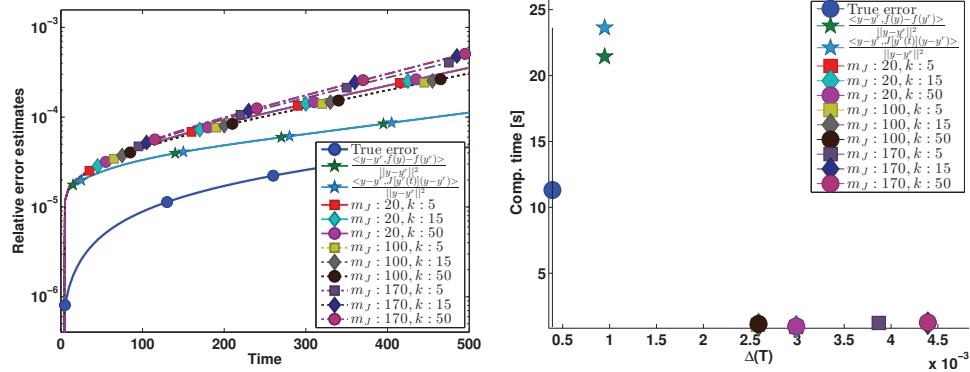


FIG. 4.10. Upper row: Caspase concentrations of full model for $\boldsymbol{\mu} = (0.777, 0.00132)^T$. Lower row: Absolute errors.

FIG. 4.11. Absolute errors over time for different k, m_J combinations.FIG. 4.12. Relative errors (left) and computation times against estimates at $T = 500$ (right) for different estimator configurations.

Jacobian \mathbf{J} and use identical line- and marker-types for the same m_J/k values as before. We decided to use the true DEIM approximation error for the references and fixed $m' = 10$ for all other estimates. Analogous to the previous experiment, both reference estimates are visually indistinguishable and confirm that the approximation (2.16) via Jacobians is applicable. Surprisingly for this case, different m_J values have more impact on the estimation results than different k values, as the estimation results for fixed m_J are lying close to each other. However, the connection between increasing m_J and the estimation results is not monotonous as all estimates for $m_J = 100$ are below the ones using $m_J = 20$.

Figure 4.12 shows the relative errors for the same configurations on the left, and the image on the right illustrates the computation times required for the different configurations plotted against the estimated error at $T = 500$. Note here that the times for the reference estimates also include the time needed to compute the full solution required for the local constants. Finally, Table 4.1 shows an overview of the used estimator configurations regarding computation times and effectivities. Remarkable at this point is that the effectivities of all estimator configurations are in the order of ten at $T = 500$. As expected, the computation times are increasing along with higher

TABLE 4.1
Overview for all used estimator configurations.

Version	$\Delta(500)$	Time	Effectivity
True error	3.908×10^{-4}	11.31s	1.000
$\frac{\langle y - y^r, f(y) - f(y^r) \rangle}{\ y - y^r\ ^2}$	9.476×10^{-4}	21.44s	2.425
$\frac{\langle y - y^r, J[y^r(t)](y - y^r) \rangle}{\ y - y^r\ ^2}$	9.476×10^{-4}	23.63s	2.425
$m_J : 20, k : 5$	2.989×10^{-3}	0.92s	7.649
$m_J : 20, k : 15$	2.989×10^{-3}	0.94s	7.649
$m_J : 20, k : 50$	2.989×10^{-3}	0.97s	7.649
$m_J : 100, k : 5$	2.580×10^{-3}	1.08s	6.602
$m_J : 100, k : 15$	2.588×10^{-3}	1.09s	6.622
$m_J : 100, k : 50$	2.588×10^{-3}	1.15s	6.622
$m_J : 170, k : 5$	3.869×10^{-3}	1.22s	9.900
$m_J : 170, k : 15$	4.392×10^{-3}	1.20s	1.124×10^1
$m_J : 170, k : 50$	4.392×10^{-3}	1.27s	1.124×10^1

m_J and k values, but compared to the full solution the speedup including the error estimator is still about a factor of ten. This higher cost of about twice the pure reduced simulation time is the price we need to pay for the a posteriori error estimation. Of course, seen as an absolute value this still is subject to further improvement; however, given the achieved effectivities the estimation results are already very satisfying. In particular, the results do not show the exponential growth that is typical of many Gronwall/comparison lemma-type estimates.

5. Conclusions. In this work we introduced a novel approach for a posteriori error estimation of nonlinear dynamical systems reduced by subspace projection and DEIM approximation of the system's nonlinearities. The main ingredients are an error estimation for the DEIM approximation using a higher number of DEIM basis functions and approximations of local logarithmic Lipschitz constants of the nonlinearities f . The latter is achieved by a partial similarity transformation of the Jacobians preserving the space of the largest eigenvectors in combination with an MDEIM to allow for an efficient offline/online decomposition of the logarithmic norm computation. Our experiments demonstrate the usability of the proposed methods to obtain efficient and practically applicable a posteriori error estimators for a quite general class of nonlinear dynamical systems. Future work will focus on further improving the local Lipschitz constant estimation by possible inclusion of state-space error directions and more eigenvalues of the Jacobians [31].

Acknowledgments. The authors would like to thank the anonymous referees for numerous helpful comments. We are particularly grateful to one of the reviewers for suggesting the elegant reformulation of the original version of Lemma 2.7.

REFERENCES

- [1] A.C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.
- [2] M. BARRAULT, Y. MADAY, N.C. NGUYEN, AND A.T. PATERA, *An ‘empirical interpolation’ method: Application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672.
- [3] P. BENNER AND T. BREITEN, *Krylov-subspace based model reduction of nonlinear circuit models using bilinear and quadratic-linear approximations*, in Progress in Industrial Mathematics at ECMI 2010, Math. Ind. 17, Springer, Berlin, Heidelberg, 2012, pp. 153–159.
- [4] K. CARLBERG, R. TUMINARO, AND P. BOGGS, *Efficient structure-preserving model reduction for nonlinear mechanical systems with application to structural dynamics*, in Proceedings of

- the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, AIAA Paper 2012-1969, Honolulu, Hawaii, 2012, pp. 9532–9547.
- [5] S. CHATURANTABUT AND D.C. SORENSEN, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764.
 - [6] S. CHATURANTABUT AND D.C. SORENSEN, *A state space error estimate for POD-DEIM nonlinear model reduction*, SIAM J. Numer. Anal., 50 (2012), pp. 46–63.
 - [7] G. DAHLQUIST, *Stability and Error Bounds in the Numerical Integration of Ordinary Differential Equations*, J. Appl. Math. Mech., 41 (1961), pp. 267–268.
 - [8] M. DAUB, S. WALDHERR, F. ALLGÖWER, P. SCHEURICH, AND G. SCHNEIDER, *Death wins against life in a spatially extended apoptosis model*, Biosystems, 108 (2012), p. 45–51.
 - [9] M. DROHMAN, B. HAASDONK, AND M. OHLBERGER, *Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation*, SIAM J. Sci. Comput., 34 (2012), pp. A937–A969.
 - [10] J.L. EFTANG, M.A. GREPL, AND A.T. PATERA, *A posteriori error bounds for the empirical interpolation method*, C. R. Math. Acad. Sci. Paris, 348 (2010), pp. 575–579.
 - [11] J.L. EFTANG AND B. STAMM, *Parameter multi-domain “hp” empirical interpolation*, Internat. J. Numer. Methods Engrg., 90 (2012), pp. 412–428.
 - [12] M. GREPL, *Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations*, Math. Models Methods Appl. Sci., 22 (2012), 1150015.
 - [13] M.A. GREPL, Y. MADAY, N.C. NGUYEN, AND A.T. PATERA, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations*, M2AN Math. Model. Numer. Anal., 41 (2007), pp. 575–605.
 - [14] M.A. GREPL, *Reduced-Basis Approximation and A Posteriori Error Estimation for Parabolic Partial Differential Equations*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2005.
 - [15] B. HAASDONK, *Convergence rates of the POD-greedy method*, ESAIM Math. Model. Numer. Anal., 47 (2013), pp. 859–873.
 - [16] B. HAASDONK AND M. OHLBERGER, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, M2AN Math. Model. Numer. Anal., 42 (2008), pp. 277–302.
 - [17] B. HAASDONK AND M. OHLBERGER, *Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition*, Math. Comput. Model. Dyn., 17 (2011), pp. 145–161.
 - [18] J.K. HALE, *Ordinary Differential Equations*, Pure and Applied Mathematics XXI, Wiley-Interscience, New York, 1969.
 - [19] M. HINZE AND S. VOLKWEIN, *Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control*, in Dimension Reduction of Large-Scale Systems, Lect. Notes Comput. Sci. Eng. 45, Springer, Berlin, 2005, pp. 261–306.
 - [20] I.T. JOLLIFFE, *Principal Component Analysis*, Springer-Verlag, New York, 2002.
 - [21] K. KUNISCH AND S. VOLKWEIN, *Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics*, SIAM J. Numer. Anal., 40 (2003), pp. 492–515.
 - [22] K. KUNISCH AND V. VOLKWEIN, *Proper orthogonal decomposition for optimality systems*, M2AN Math. Model. Numer. Anal., 42 (2008), pp. 1–23.
 - [23] M. MEYER AND H.G. MATTHIES, *Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods*, Comput. Mech., 31 (2003), pp. 179–191.
 - [24] J. PHILLIPS, J. AFONSO, A. OLIVEIRA, AND L.M. SILVEIRA, *Analog macromodeling using kernel methods*, in Proceedings of ICCAD 2003, 2003, pp. 446–453.
 - [25] T. REIS AND M. HEINKENSCHLOSS, *Model reduction with a-priori error bounds for a class of nonlinear electrical circuits*, in Proceedings of CDC/CCC 2009, 2009, pp. 5376–5383.
 - [26] M. REWIENSKI AND J. WHITE, *A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices*, IEEE Trans. Computer-Aided Design, 22 (2003), pp. 155–170.
 - [27] G. SÖDERLIND, *The logarithmic norm. History and modern theory*, BIT, 46 (2006), pp. 631–652.
 - [28] T. TONN, *Reduced-Basis Method (RBM) for Non-Affine Elliptic Parametrized PDEs*, Ph.D. thesis, Universität Ulm, Ulm, Germany, 2011.
 - [29] S. VOLKWEIN, *Model Reduction Using Proper Orthogonal Decomposition*, lecture notes, University of Graz, Graz, Austria, 2008.
 - [30] B. WIELAND, *Reduced Basis Methods for Partial Differential Equations with Stochastic Influences*, Ph.D. thesis, Universität Ulm, Ulm, Germany, 2013.

- [31] D. WIRTZ, *Model Reduction for Nonlinear Systems: Kernel Methods and Error Estimation*, Ph.D. thesis, University of Stuttgart, Stuttgart, Germany, 2013.
- [32] D. WIRTZ AND B. HAASDONK, *A-posteriori error estimation for parameterized kernel-based systems*, in Proceedings of MATHMOD 2012—7th Vienna International Conference on Mathematical Modelling, 2012.
- [33] D. WIRTZ AND B. HAASDONK, *Efficient a-posteriori error estimation for nonlinear kernel-based reduced systems*, Systems Control Lett., 61 (2012), pp. 203–211.