Max Planck Institute Magdeburg
Preprints

Ulrike Baur Peter Benner Bernard Haasdonk
Christian Himpe Immanuel Maier Mario Ohlberger

Comparison of methods for parametric model order reduction of instationary problems

MPIMD/15-01 **February 20, 2015**

Abstract

Dynamical systems of large order appear in many applications. For an efficient simulation it can become necessary to reduce the system dimension using a reliable model order reduction method, in particular in a many-query context when the system is to be solved for varying parameters and input signals. Nowadays, it is often required that the models include physical parameters to allow more flexibility in simulation. These parameters should be preserved in the reduced-order system; a task that motivates the development of new approaches to model order reduction referred to collectively as parametric model order reduction. In this work, we compare several methods for parametric model order reduction using common benchmark problems from the literature.
Keywords. Parametric model order reduction, frequency response, transient simulation, input-output map.

Impressum:

Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Publisher:
Max Planck Institute for
Dynamics of Complex Technical Systems

Address:
Max Planck Institute for
Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg

http://www.mpi-magdeburg.mpg.de/preprints/

1 Introduction

In this work, parameterized dynamical systems of order n

$$\begin{aligned} E(p)\,\dot{x}(t;p) &= A(p)\,x(t;p) + B(p)\,u(t), \\ y(t;p) &= C(p)\,x(t;p) \end{aligned} \tag{1}$$

with a parameter (vector) $p \in \mathbb{R}^d$ and parameter-dependent system matrices $E(p)$, $A(p) \in \mathbb{R}^{n\times n}$, $B(p) \in \mathbb{R}^{n\times m}$, $C(p) \in \mathbb{R}^{\ell\times n}$ are considered.

It is assumed that, for all considered parameter values, $E(p)$ is invertible and the system is stable, i.e., the eigenvalues of $E^{-1}(p)A(p)$ lie in the open left half of the complex plane. In the following, the parameter dependency of the state x and of the output y is omitted in notation for a simplified presentation. In many research fields, like signal processing or control theory, the system is analyzed in the frequency domain. The system's response in the frequency domain is described by a linear mapping, called *transfer function* which maps the Laplace transform of the inputs to the Laplace transform of the outputs. The parameterized transfer function corresponding to (1) for $s \in \overline{\mathbb{C}}_+ := \{s \in \mathbb{C} \mid \operatorname{Re}(s) \geq 0\}$ is defined by

$$G(s,p) = C(p)(sE(p) - A(p))^{-1}B(p). \tag{2}$$

Parametric model order reduction (PMOR) based on projection seeks (full column rank) matrices V, $W \in \mathbb{R}^{n\times r}$ with $r \ll n$ such that the output error between the original and the reduced-order system

$$\begin{aligned} W^T E(p) V\,\dot{\hat{x}}(t) &= W^T A(p) V\,\hat{x}(t) + W^T B(p)\,u(t), \\ \hat{y}(t) &= C(p) V\,\hat{x}(t), \end{aligned} \tag{3}$$

or, analogously, in the frequency domain

$$\hat{G}(s,p) = C(p)V(sW^T E(p)V - W^T A(p)V)^{-1}W^T B(p), \tag{4}$$

is small and the computational time for the simulation of (1) and (2) is decreased significantly by using (3) or (4) instead. The simulation time of a system will be called the *online complexity* since this computation has to be done for every new value of p and input u. The time which is required to compute a reduced-order, parameterized model will be called the *offline complexity* of the PMOR method.

The reduction is especially of value (with respect to reduced computational complexity) if the parameter dependency in (1) is affine in the system matrices [24, 3], i.e., we have the following matrix representation

$$\begin{aligned} E(p) &= E_0 + e_1(p)E_1 + \ldots + e_{P_E}(p)E_{P_E}, \\ A(p) &= A_0 + f_1(p)A_1 + \ldots + f_{P_A}(p)A_{P_A}, \\ B(p) &= B_0 + g_1(p)B_1 + \ldots + g_{P_B}(p)B_{P_B}, \\ C(p) &= C_0 + h_1(p)C_1 + \ldots + h_{P_C}(p)C_{P_C}, \end{aligned}$$

leading to reduced-order matrices

$$\begin{array}{rcrcrcl} \hat{E}(p) & := & W^T E(p) V & = & W^T E_0 V & + & \sum_{i=1}^{P_E} e_i(p) W^T E_i V, \\ \hat{A}(p) & := & W^T A(p) V & = & W^T A_0 V & + & \sum_{i=1}^{P_A} f_i(p) W^T A_i V, \\ \hat{B}(p) & := & W^T B(p) & = & W^T B_0 & + & \sum_{i=1}^{P_B} g_i(p) W^T B_i, \\ \hat{C}(p) & := & C(p) V & = & C_0 V & + & \sum_{i=1}^{P_C} h_i(p) C_i V. \end{array} \tag{5}$$

It is assumed that the number of summands P_E, P_A, P_B, P_C is moderate. The reduced parameter-independent matrices $W^T E_i V$, $W^T A_i V \in \mathbb{R}^{r\times r}$, $W^T B_i \in \mathbb{R}^{r\times m}$ and $C_i V \in \mathbb{R}^{\ell\times r}$ can be pre-computed. The computation of these constant reduced matrices is part of the offline phase of the method (and will not be carried out for every new value of p).

It is also assumed that the initial state is zero, although the methods in principle can also be used in case of varying or parameter-dependent initial state. Some of the methods can easily be modified for the case of singular E while others are still under investigation. Also, not all approaches require stability. The comparison is restricted to the preassigned assumptions in order to make it applicable to a wide range of different methods.

In the following, several methods for PMOR are compared. All approaches are briefly introduced in Section 2 with error measures described in Section 3. Each method is applied to three benchmarks selected from the MOR Wiki benchmark collection [1], see Section 5 for details. The results and a discussion of the comparison can be found in Section 6.

This work complements a recent survey on model reduction methods for parametric systems [7] where several approaches and aspects for PMOR are discussed in detail but no numerical experiments are included. For a comparison of MOR methods for systems without parameter dependency, see, for instance, [18].

The comparisons described here should be thought of as a first attempt to look at the advantages and disadvantages of various state-of-the-art strategies in PMOR. For a simplified presentation of the results and a better comparability of the approaches, we limit the comparison to single-input single-output (SISO) systems which depend affinely on one parameter ($d = 1$). This will be extended to multi-input multi-output systems (MIMO) with more than one parameter ($d > 1$) in future work.

2 Methods for parametric model order reduction

The computation of the projection matrices V and W is of main interest in PMOR and differs very much along the presented approaches. We will shortly describe the methods which will be compared in the following.

2.1 POD and POD-Greedy

First, two approaches are applied for computing matrices V and W that are based on Proper Orthogonal Decomposition (POD). POD is a state-space approximation

method providing optimal approximation spaces in the mean-squared error sense [35, 8, 36]. To be more precise, the POD assumes a set of points $x_1, \dots, x_N \in \mathbb{R}^n$, compactly written as a matrix $X = [x_1, \dots, x_N]$ and computes for arbitrarily chosen $r \in \{1, \dots, N\}$:

$$POD_r(X) := \arg\min_V \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - VV^T x_i \right\|^2$$

over all matrices $V \in \mathbb{R}^{n \times r}$ that satisfy $V^T V = I$. This can be directly computed by a singular value decomposition (SVD) $V_1 S V_2^T \overset{\text{SVD}}{=} X$ and taking the r leftmost columns of V_1 as matrix V. These columns then are also denoted as POD-modes.

In the following methods I) and II) we only address the construction of a matrix V with orthogonal columns in the setting of parametric systems on a finite time-interval $[0, T]$ with a fixed input $u(\cdot)$. After construction of V, we simply choose $W = V$.

I) *Global POD*: We assume to have a set of training parameter samples $\{p_1, \dots, p_K\}$ and time discretization by choosing $J \in \mathbb{N}$ and setting $\Delta t := T/J$ and $t_i := i\Delta t, i = 0, \dots, J$. These define the set $\{x_1 \dots, x_N\} := \{x(t_0; p_1), \dots, x(t_J; p_K)\}$ by computing $K \cdot (J + 1)$ solution snapshots of the full model. The POD of this (potentially huge) set of snapshots yields a matrix $V = POD_r(X)$ that can be used for system projection and promises good average approximation over the time and parameter range, if J, K and r are chosen sufficiently large.

II) *POD-Greedy*: The size of the SVD problem of the global POD can be prohibitive, which is circumvented by the following greedy procedure that incrementally constructs a basis V by several small POD computations: Starting with an initial basis V and the corresponding reduced system, one can detect that single parameter p_j among the parameter samples, which currently is worst resolved by the reduced model (e.g., measured by error norms as specified in the next section). For this "worst" sample p_j, the high dimensional trajectory $x(\cdot; p_j)$ is computed and the new information of this trajectory is extracted by computing $r' \geq 1$ POD-modes of the orthogonal projection error trajectory to the current V. This means, we first orthogonalize all trajectory elements and collect them into a matrix, i.e. X' is chosen as the matrix with columns $\{x(t_i; p_j) - VV^T x(t_i; p_j)\}_{i=0}^{J}$ in arbitrary order. Then, we extract the new POD-mode(s) $V' := POD'_r(X')$. This matrix V' is orthonormal to V by construction and we extend the basis $V := [V, V']$. This extension loop is repeated until a desired accuracy is obtained on the training set of parameters or a desired basis size r is reached. The choice $r' = 1$, i.e., adding a single POD-mode in each iteration, promises to result in the most compact basis, while $r' > 1$ could result in an accelerated basis generation time. This POD-Greedy algorithm [23, 37] meanwhile is standard in Reduced Basis Methods [24, 10, 26] and has provable quasi-optimal convergence rates [20]. Adaptive techniques of selecting parameters or snapshots for reducing the complexity of the reduced basis generation have been proposed for example in [22, 40].

In the above procedures I) and II) also variation of the input u can be allowed via a parametrization of the input. First, if a finite number n_u of input signals u_i is to be expected, one can extend the parameter vector by $\bar{p} = (p, i)$ and hence directly include the input variation into the POD or POD-Greedy procedure by varying $\bar{p}$ over

$\mathbb{R}^d \times \{1, ..., n_u\}$.

Second, if a parametric model of the prospective input signals is available, e.g., $u(t) = u(t; p')$ with $p' \in \mathbb{R}^{d'}$ this can also be directly be included in a parametric sampling of the input signal manifold by extending the parameter vector by $\bar{p} := (p, p')$ and varying $\bar{p}$ over $\mathbb{R}^d \times \mathbb{R}^{d'}$.

2.2 Interpolatory methods for PMOR

Next, interpolatory methods for PMOR are considered which are based on well-known techniques for model order reduction (MOR) of deterministic (non-parametric) systems such as moment-matching, the iterative rational Krylov algorithm (IRKA) [19] and balanced truncation (BT). These (deterministic) MOR methods are used to reduce the order of the system (1) at a certain number of fixed parameter values $p_1, \ldots, p_K$, i.e., we apply MOR K-times on the systems with transfer functions:

$$G_j(s) := G(s, p_j) = C(p_j)(sE(p_j) - A(p_j))^{-1}B(p_j), \quad j = 1, \ldots, K.$$

The following (local) quantities are computed and stored after reduction for further use in interpolation-based PMOR:

1. the projection matrices $V_j,\ W_j \in \mathbb{R}^{n \times r'}$, for $j = 1, \ldots K$,
2. reduced system matrices, for $j = 1, \ldots K$,

$$\begin{array}{rclcrcl} \hat{E}_j & = & W_j^T E(p_j) V_j \in \mathbb{R}^{r' \times r'}, & & \hat{A}_j & = & W_j^T A(p_j) V_j \in \mathbb{R}^{r' \times r'}, \\ \hat{B}_j & = & W_j^T B(p_j) \in \mathbb{R}^{r' \times m}, & & \hat{C}_j & = & C(p_j) V_j \in \mathbb{R}^{\ell \times r'}. \end{array} \tag{6}$$

The PMOR approaches considered here use different interpolation strategies which employ some of these quantities to derive a reduced-order system (3) which approximates (1) over the whole parameter interval.

A short description of the methods follows.

I) The first method is called *PMOR by matrix interpolation* (MatrInt) [33]. A parameterized reduced-order system (3) of order r' is obtained by interpolation of the locally reduced system matrices (6), where

$$\begin{array}{rclcrcl} \hat{E}(p) & = & \sum_{j=1}^K \omega_j(p) M_j \hat{E}_j T_j^{-1}, & & \hat{A}(p) & = & \sum_{j=1}^K \omega_j(p) M_j \hat{A}_j T_j^{-1}, \\ \hat{B}(p) & = & \sum_{j=1}^K \omega_j(p) M_j \hat{B}_j, & & \hat{C}(p) & = & \sum_{j=1}^K \omega_j(p) \hat{C}_j T_j^{-1}, \end{array}$$

with properly chosen transformation matrices $M_j,\ T_j \in \mathbb{R}^{r' \times r'}$ and weights ω_j. The transformation matrices are chosen in order to give a common physical meaning to all reduced state vectors:

$M_j = (W_j^T R)^{-1}$, $T_j = R^T V_j$ with $R \in \mathbb{R}^{n \times r'}$ obtained from a thin SVD of $[\omega_1(p)V_1, \omega_2(p)V_2, \ldots, \omega_K(p)V_K]$. Note that a single R is used to compute the transformation matrices M_j, T_j for $j = 1, \ldots, K$ and that the computation is part of the online phase since the weights depend on p. Modifications of the approach which avoid this online step are proposed in [2, 33].

Stability of the parameterized reduced-order system can be achieved by a further offline step which includes the solution of K low-dimensional Lyapunov equations [17]. Transformations based on these solutions make the locally reduced systems contractive.

II) The next approach for PMOR is called *transfer function interpolation* (TransFncInt). This is based on interpolation of locally reduced-order models in the frequency domain [4]. Here, reduced-order transfer functions

$$\hat{G}_j(s) = \hat{C}_j(s\hat{E}_j - \hat{A}_j)^{-1}\hat{B}_j, \tag{7}$$

with $\hat{E}_j$, $\hat{A}_j$, $\hat{B}_j$, $\hat{C}_j$ computed by (6), are taken as 'data points' for the construction of a (parameter-dependent) interpolant. Using polynomial interpolation, the reduced-order transfer function looks as follows,

$$\hat{G}(s,p) = \sum_{j=1}^{K} L_j(p)\hat{G}_j(s),$$

where $L_j(p)$ are the Lagrange basis polynomials. A possible realization of order $K \cdot r'$, here exemplarily described for a system with $p \in \mathbb{R}$, is

$$\hat{G}(s,p) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B}(p),$$

with

$$\hat{C} := \begin{bmatrix} \hat{C}_1, & \ldots, & \hat{C}_K \end{bmatrix}, \ \hat{B}_j(p) := \left(\prod_{i=1, i\neq j}^{K} \frac{p - p_i}{p_j - p_i} \right) \hat{B}_j,$$

$$s\hat{E} - \hat{A} := \begin{bmatrix} s\hat{E}_1 - \hat{A}_1 & & \\ & \ddots & \\ & & s\hat{E}_K - \hat{A}_K \end{bmatrix}, \ \hat{B}(p) := \begin{bmatrix} \hat{B}_1(p) \\ \vdots \\ \hat{B}_K(p) \end{bmatrix}.$$

Note that this realization does not allow a reconstruction of the state in the required form $x \approx V\hat{x}$.

TransFncInt also does not provide a reduced-order model in parameterized state-space form for more than one parameter (and for other interpolation techniques than polynomial interpolation). For higher dimensional parameter spaces, piecewise polynomial interpolation on sparse grids provides an efficient implementation of this method [4]. Other interpolation techniques, such as rational interpolation, can also be used to construct $\hat{G}(s,p)$, see [5].

Note that the preservation of stability can be guaranteed in the parameterized reduced-order system if BT is taken for the local reduction. BT preserves the stability of the local transfer functions $\hat{G}_j$ in (7) and this guarantees that the eigenvalues of the interpolated transfer function $\hat{G}(s,p)$ also lie in $\mathbb{C}_-$.

III) We further consider an approach called *piecewise $\mathcal{H}_2$ tangential interpolation* (with $\mathcal{H}_2$ optimal frequency points) (PWH2TanInt) [3]. The local projection matrices are computed by IRKA (with local reduced order r') and concatenated

$$V = [V_1,\ V_2, \ldots, V_K], \quad W = [W_1,\ W_2, \ldots, W_K]$$

to obtain (4). Thus, the dimension of the reduced-order system is $K \cdot r'$. Note that the number of columns $K \cdot r'$ of V and W can further be reduced by an SVD or a rank-revealing QR factorization to ensure that V and W have full rank. IRKA computes optimal (frequency) shifts s_i and corresponding tangential directions b_{ij} and c_{ij} such that (4) matches the p-gradient and p-Hessian of the original system response (2) with respect to the parameters:

$$\nabla_p c_{ij}^T G(s_i, p_j) b_{ij} = \nabla_p c_{ij}^T \hat{G}(s_i, p_j) b_{ij}, \quad \nabla_p^2 c_{ij}^T G(s_i, p_j) b_{ij} = \nabla_p^2 c_{ij}^T \hat{G}(s_i, p_j) b_{ij},$$

for $i = 1, \ldots, r',\ j = 1, \ldots, K$. Additionally, the usual tangential interpolation properties hold:

$$G(s_i, p_j) b_{ij} = \hat{G}(s_i, p_j) b_{ij}, \quad c_{ij}^T G(s_i, p_j) = c_{ij}^T \hat{G}(s_i, p_j).$$

IV) The generalization of moment matching MOR called multi- (parameter) moment matching (MultiPMomMtch) or multivariate Padé approximation was first considered in [9, 38]. Improvements which avoid explicitly moment matching can be found in [12, 13, 30, 11]. This method is based on a multivariate Taylor expansion with expansion points in frequency and parameter space. We denote the frequency expansion points by $s_1, \ldots, s_L$ in the following. It ensures the following moment matching,

$$\frac{\partial^k}{\partial s^k} \frac{\partial^l}{\partial p^l} G(s_i, p_j) = \frac{\partial^k}{\partial s^k} \frac{\partial^l}{\partial p^l} \hat{G}(s_i, p_j),$$

for $i = 1, \ldots, L,\ j = 1, \ldots, K,\ k = 0, \ldots, q,\ l = 0, \ldots, q$ and in all directions (in contrast to tangential interpolation). The dimension of the reduced-order system is $K \cdot L \cdot \frac{q}{2} \cdot (1 + q)$ (for single-input single-output systems). This dimension might get reduced by truncating linearly dependent columns during the (repeated modified Gram-Schmidt) orthogonalization process [13].

2.3 Empirical cross Gramian

The empirical cross Gramian (emWX) is a snapshot-based method to compute the cross Gramian, which is applicable to square systems[1]. This method requires an invertible matrix E, thus it can be described in terms of linear control systems (1) with $E^{-1}A \to A$, $E \to I$ and $E^{-1}B \to B$. The cross Gramian [14] is defined as:

$$W_X := \int_0^\infty e^{At} BC e^{At} dt. \tag{8}$$

For a system with a symmetric transfer function, which includes SISO systems, the absolute values of the cross Gramian's eigenvalues equal the Hankel singular values on which BT is based. Instead of computing the cross Gramian by solving a Sylvester matrix equation or through the empirical cross Gramian from [25], a controllability-based approach is chosen here. To compute the cross Gramian, the underlying linear

[1] Systems with the same number of inputs and outputs.

control system's (1) vector field is augmented by the negative adjoint vector field and the output functional is augmented by the adjoint output functional:

$$\begin{pmatrix}\dot{x}\\ \dot{\bar{x}}\end{pmatrix} = \begin{pmatrix}A & 0\\ 0 & A^T\end{pmatrix}\begin{pmatrix}x\\ \bar{x}\end{pmatrix} + \begin{pmatrix}B\\ C^T\end{pmatrix}\begin{pmatrix}u\\ \bar{u}\end{pmatrix},$$
$$\begin{pmatrix}y\\ \bar{y}\end{pmatrix} = \begin{pmatrix}C & B^T\end{pmatrix}\begin{pmatrix}x\\ \bar{x}\end{pmatrix},$$

with $\bar{x}_0$ and $\bar{u}$ chosen appropriately; a simple choice would be $\bar{x}_0 = x_0$, $\bar{u} = u$. The controllability Gramian of this augmented system $\widehat{W}_C$ has the form [15]:

$$\widehat{W}_C = \begin{pmatrix}W_C & W_X\\ W_X^T & W_O\end{pmatrix}, \tag{9}$$

which contains the cross Gramian W_X as upper right block.

To compute this version of the cross Gramian, the empirical controllability Gramian is employed. Empirical Gramians are computed using impulse-response snapshots. Hence, in order to compute the empirical cross Gramian using the relation from (9) amounts to computing an empirical controllability Gramian from [28, 29]. Since only the upper right block of the augmented system's controllability Gramian is required to obtain the empirical cross Gramian, it is sufficient to compute W_X using snapshots $X := [x(t_0), \dots, x(t_J)] \in \mathbb{R}^{n\times J}$ and adjoint snapshots $\bar{X} := [\bar{x}(t_0), \dots, \bar{x}(t_J)] \in \mathbb{R}^{n\times J}$, given by:

$$x(t) = \int_0^t e^{A\tau}Bu(\tau)d\tau, \quad \bar{x}(t) = \int_0^t e^{A^T\tau}C^T u(\tau)d\tau.$$

Then, the empirical cross Gramian is computed by:

$$W_X \approx \Delta t\, X\bar{X}^T,$$

which corresponds to a discrete evaluation of (8). The snapshots can be obtained by solving $x(t)$ and $\bar{x}(t)$ at discrete times $t_0, \dots, t_J$, for instance using a Runge-Kutta method. This method is closely related to POD as described in Section 2.1 and to balanced POD from [39], see also [6, 21].

For a parameterized system (1), the parameter space is discretized and the mean empirical cross Gramian $\overline{W}_X$ is computed over all points p_j, $j = 1, \dots, K$, in the discretized parameter space, using the snapshots $X(p_j) := [x(t_0; p_j), \dots, x(t_J; p_j)]$ and adjoint snapshots $\bar{X}(p_j) := [\bar{x}(t_0; p_j), \dots, \bar{x}(t_J; p_j)]$, with discrete times $t_0, \dots, t_J$ and $\Delta t = |t_i - t_j|$, $i \neq j$:

$$\overline{W}_X = \frac{1}{K}\sum_{j=1}^{K} W_X(p_j) \approx \frac{1}{K}\sum_{j=1}^{K} \Delta t\, X(p_j)\bar{X}(p_j)^T =: \widetilde{W}_X.$$

An SVD of the empirical cross Gramian provides an approximate balancing projection V_1:

$$\widetilde{W}_X \overset{\text{SVD}}{=} V_1 S V_2.$$

With the Galerkin projection V_1, the reduced order model is constructed similar to the POD method. The columns of V_1 which correspond to the lowest entries of the diagonal matrix S are truncated. Then, the reduced system is given by (5) with $V = V_1(:, 1:r)$ and $W = V$.

3 Performance measures

PMOR seeks for a reduced-order, parameterized representation of the original system such that the output error between the original and the reduced-order system, i.e., $\|y - \hat{y}\|$, is small in some norm. This demand can be satisfied by a good approximation of (2) by a reduced-order, parameterized transfer function (4), i.e., forcing $\hat{G}(s,p) \approx G(s,p)$ over a wide frequency range for s and a wide parameter range for p. Alternatively, a good approximation of $x(t)$ by $V\hat{x}(t)$ also will ensure a small output error.

The choice of an appropriate error measure depends on the application. In this work, we consider many different error norms in order to get good insights in the qualities of the presented methods.

All error measures have in common that they require a discretization of the parameter space as a first step. To this end, we introduce the parameter test grid $p_1, \ldots, p_{\bar{K}}$ with $\bar{K} > K$ and compute the errors for every parameter grid point p_j, $j = 1, \ldots, \bar{K}$. We denote the corresponding quantities by $G_j := G(\cdot, p_j)$, $\hat{G}_j := \hat{G}(\cdot, p_j)$, $y_j := y(\cdot\,; p_j)$, $x_j := x(\cdot\,; p_j)$.

The state and the output errors are computed in the time domain for a finite time interval $[0, T]$. We drive the original and the reduced-order system with the same input $u(\cdot)$. The time interval is discretized by $J + 1$ time points $t_0, \ldots, t_J$.

The error in state space is estimated by the $\mathcal{L}_2$-norm for square-integrable functions,

$$\|x_j - V\hat{x}_j\|^2_{\mathcal{L}_2([0,T])} = \int_0^T \|x_j(t) - V\hat{x}_j(t)\|_2^2 \, dt \approx \Delta t \sum_{i=0}^{J} \|(x_j(t_i) - V\hat{x}_j(t_i))\|_2^2.$$

The output error is computed in two different norms,

$$\|y_j - \hat{y}_j\|^2_{\mathcal{L}_2([0,T])} = \int_0^T \|y_j(t) - \hat{y}_j(t)\|_2^2 \, dt \approx \Delta t \sum_{i=0}^{J} \|(y_j(t_i) - \hat{y}_j(t_i))\|_2^2,$$

and the $\mathcal{L}_\infty$-norm

$$\|y_j - \hat{y}_j\|_{\mathcal{L}_\infty([0,T])} \approx \max_{i=0,\ldots,J} \|y_j(t_i) - \hat{y}_j(t_i)\|_2.$$

These errors are computed as relative errors with denominator $\|x_j\|_{\mathcal{L}_2}$, $\|y_j\|_{\mathcal{L}_2}$, or $\|y_j\|_{\mathcal{L}_\infty}$, respectively.

The frequency response error can be computed by the Hardy $\mathcal{H}_2$- and the $\mathcal{H}_\infty$-norm which provide good error measures for many classes of input signals.

The computation of the $\mathcal{H}_\infty$-norm additionally requires a fine grid of frequency points $\omega_1, \ldots, \omega_{\bar{L}} \in \mathbb{R}$ ($\bar{L} > L$) for computing an estimate of

$$\|G_j - \hat{G}_j\|_{\mathcal{H}_\infty} = \sup_{\omega \in \mathbb{R}} \bar{\sigma}(G_j(\imath\omega) - \hat{G}_j(\imath\omega)) \approx \max_{1 \le i \le \bar{L}} \bar{\sigma}(G_j(\imath\omega_i) - \hat{G}_j(\imath\omega_i)),$$

with $\imath = \sqrt{-1}$ and $\bar{\sigma}(G_j(\imath\omega))$ as largest singular value of the $\ell \times m$ matrix $G_j(\imath\omega)$. The error in the $\mathcal{H}_\infty$-norm is computed as scaled error, where the maximum of all $\|G_j(\imath\omega_i) - \hat{G}_j(\imath\omega_i)\|_2$ divided by $\max_{1\le l\le \bar{L}} \|G_j(\imath\omega_l)\|_2$ for $i = 1, \ldots, \bar{L}$ is taken.

The error in the $\mathcal{H}_2$-norm is computed without sampling of the frequency space by

$$\begin{aligned} \|G_j - \hat{G}_j\|_{\mathcal{H}_2}^2 &= \tfrac{1}{2\pi} \int_{-\infty}^{\infty} \operatorname{trace}\left[(G_j(\imath\omega) - \hat{G}_j(\imath\omega))^*(G_j(\imath\omega) - \hat{G}_j(\imath\omega))\right] d\omega \\ &= \operatorname{trace}\left[C_j,\, \hat{C}_j] P\, [C_j,\, \hat{C}_j]^T\right]. \end{aligned}$$

P is the controllability Gramian of the error system and is obtained as solution of a Lyapunov equation associated to the error system [41, Section 4.6]. In Section 6, the $\mathcal{H}_2$-error is computed as relative error with denominator $\|G_j\|_{\mathcal{H}_2}^2$.

For $u \in \mathcal{L}_2([0,\infty) \to \mathbb{R}^m)$, the useful estimates

$$\|y_j - \hat{y}_j\|_{\mathcal{L}_2([0,\infty))} \le \|G_j - \hat{G}_j\|_{\mathcal{H}_\infty} \|u\|_{\mathcal{L}_2([0,\infty))}$$

and

$$\|y_j - \hat{y}_j\|_{\mathcal{L}_\infty([0,\infty))} \le \|G_j - \hat{G}_j\|_{\mathcal{H}_2} \|u\|_{\mathcal{L}_2([0,\infty))},$$

connect the time domain errors with the error measures in the frequency domain.

The complexity of methods for PMOR can be divided into two parts, operations in the offline phase, where the original system size is reduced, and the online complexity which describes the costs for computing simulations for a new parameter using the reduced-order, parameterized model. The online complexity can be considered for the transient and for the frequency response.

The number of simulation runs M in an application times the online costs of the reduced system plus the offline costs should be smaller compared to M times the online costs of the full-order system to justify the use of PMOR.

4 Expectations

Before showing the numerical results, we provide some formal comparison and discussion of the methods.

The POD method is based solely on state-space simulation. Thus, it is expected to perform well for time-domain error measures, especially with respect to the state error. The POD-Greedy method is a variant of the POD method, which not only requires fewer large dense matrix operations, but also can give more accurate results. Note that by construction, POD minimizes the mean error over the training parameter set, whereas the POD-Greedy minimizes the maximal error. A good approximation of the state leads to an equally good approximation of the output, so that we expect also the output errors to be quite small. In contrast, we cannot guess anything about the quality of the frequency space approximation. Regarding the computational effort in the offline phase, the POD-Greedy algorithm is expected to be more expensive than a global POD, as it requires many iterations of solving small PODs. However, the online computational times should not show much of a difference, because the reduced orders are equal.

Interpolatory methods based on a combination of local reduction at a set of parameter sampling points and interpolation, i.e., transfer-function interpolation (TransFncInt) and matrix interpolation (MatrInt), are expected to produce smaller errors at the selected parameter values. The error between the parameter points depends on the interpolation technique whereas the error at the parameter sampling points depends on the applied MOR method. A good approximation of the transfer function with respect to the $\mathcal{H}_\infty$-norm and of the output with respect to the $\mathcal{L}_2$-norm is expected by combining TransFncInt with BT. PWH2TanInt is an approach for rational interpolation that computes a reduced-order model which matches the gradient (with respect to the parameters) of the original transfer function at the parameter sampling points. The reduced-order model is locally $\mathcal{H}_2$-optimal if IRKA is applied for computing the local projection matrices V_j and W_j by determining optimal frequency expansion points (and tangent directions). Thus, this method is anticipated to produce small $\mathcal{H}_2$-errors. The offline costs of these interpolatory methods depend heavily on the costs for the underlying MOR approach (times the number of parameter sampling points).

The quality of the reduced-order system obtained by MultiPMomMtch depends on the initial choice of frequency and parameter expansion points and on the chosen order of matched moments. The selection of all of these quantities is largely ad-hoc and not automated in this work. The accuracy in all error measures cannot be predicted *a priori*.

Since state-space simulations of the system and the adjoint system are employed, the empirical cross Gramian should perform similarly to the POD methods with respect to the state-space norms. Yet due to the incorporation of both controllability and observability information, better accuracy in the frequency-space norms is expected.

The online complexity of all methods, except of MatrInt, is comparable since the computed reduced-order systems are of the same dimension. MatrInt and TransFncInt have an additional online step: the computation of transformation matrices and the interpolation of the system matrices in MatrInt; and the construction of a state-space realization in the implementation considered of TransFncInt. Note also that MatrInt computes a system which is much smaller than all other considered approaches and is therefore less expensive during the simulation.

5 Benchmarks

We consider three examples from the benchmark collection [1] and apply the methods which were introduced in Section 2. All systems are SISO systems with an (affine) parameter dependency on a single parameter.

5.1 Synthetic system[2]

A synthetic parameterized system of order n can be constructed as

$$G(s,p) = \sum_{i=1}^{n} \frac{r_i}{s-\sigma_i} = C(sI - A_0 - pA_1)^{-1}B,$$

where r_i and σ_i are the residues and poles of the transfer function G. The parameter p scales the real part of the system poles $\sigma_i = p\,a_i + \imath b_i$. The smaller p, the closer the poles are to the imaginary axis and the amplitude of the frequency response changes, see Figures 1 and 2. Also the decay of the Hankel singular values is influenced by p. In our particular setting, we took the following entries in the system matrices

$$A_0 = \begin{bmatrix} 0 & b_1 & & & 0 \\ -b_1 & 0 & & & \\ & & \ddots & & \\ & & & 0 & b_k \\ 0 & & & -b_k & 0 \end{bmatrix}, \; A_1 = \begin{bmatrix} a_1 & & & & 0 \\ & a_1 & & & \\ & & \ddots & & \\ & & & a_k & \\ 0 & & & & a_k \end{bmatrix},$$

$$B = [2,0,2,0,\ldots]^T, \quad C = [1,0,1,0,\ldots]$$

with $b_i \in [10,\ 10^3]$, $a_i \in [-10^3,\ -10]$ for $i = 1,\ldots,k$, $k = 500$ and an original system size $n = 1000$. The parameter range is chosen as $p \in [0.1,\ 1]$. The frequency response of the system over the whole parameter interval is shown in Figure 1.

The motivation for this benchmark comes from the simple construction which allows a flexible choice of n and of the system poles and residues. For further details, see the MOR Wiki page of the benchmark [1].

5.2 Microthruster unit[3]

The second benchmark is a real-world example from microsystems technology. This parametric model was originally presented in the Oberwolfach Model Reduction Benchmark Collection [27] under the name "Boundary Condition Independent Thermal Model" [34] and is also listed in the MOR Wiki [1]. The model describes the thermal conduction in a semiconductor chip where a flexibility in specifying the boundary conditions allows the simulation of temperature changes in the environment. This allows independent designers to observe how the surrounding influences the temperature distribution in the chip. The thermal problem is modeled as homogenous heat diffusion with heat exchange occurring at three device interfaces modeled with convection boundary conditions. These conditions introduce film coefficients describing the heat exchange on the three device interfaces. We assume two film coefficients as being fixed at 10^4, to stay within our specified setting, and one parameter (the film coefficient at the top) as variable within the range $[1,\ 10^4]$.

[2] http : //www.modelreduction.org/index.php/Synthetic_parametric_model
[3] http : //www.modelreduction.org/index.php/Microthruster_Unit

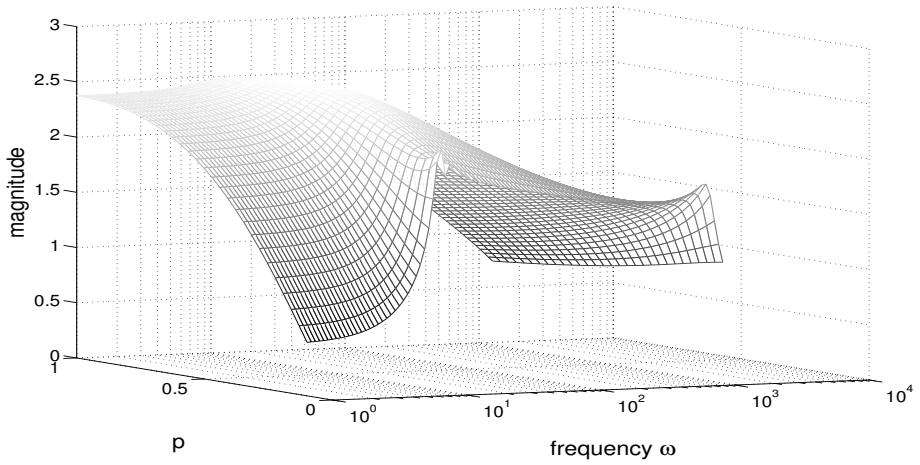

Figure 1: Frequency response of the synthetic system.

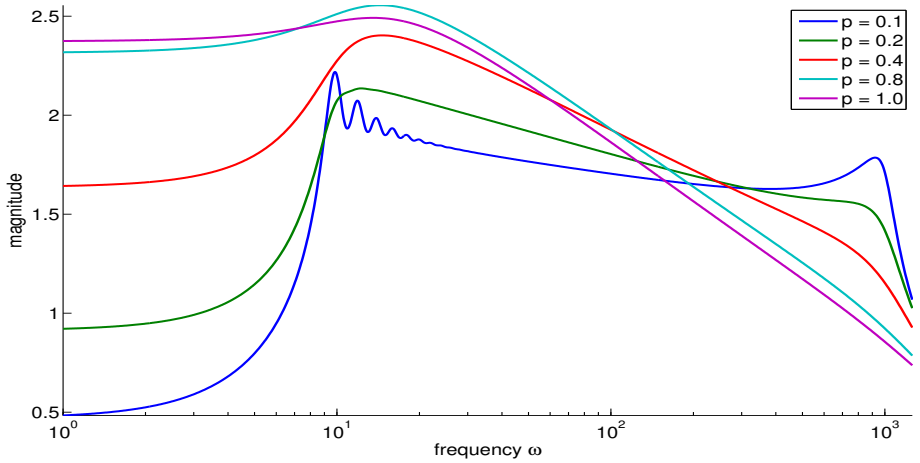

Figure 2: Frequency response of the synthetic system for some parameter values.

Discretization leads to a system of ordinary differential equations

$$E\dot{x}(t) = (A_0 + pA_1)x(t) + Bu(t), \quad y(t) = Cx(t),$$

where $E \in \mathbb{R}^{4257\times4257}$ and $A_0 \in \mathbb{R}^{4257\times4257}$ are system matrices, $A_1 \in \mathbb{R}^{4257\times4257}$ is a diagonal matrix arising from the discretization of the convection boundary condition on the top interface, and $B \in \mathbb{R}^{4257}$ is a constant load vector. Originally, the system had seven outputs. We take a sum over all rows for obtaining a single output with $C \in \mathbb{R}^{1\times4257}$. The frequency response of the system is shown in Figure 3 for p varying in $[1, 10^4]$ and for five selected values of p in Figure 4.

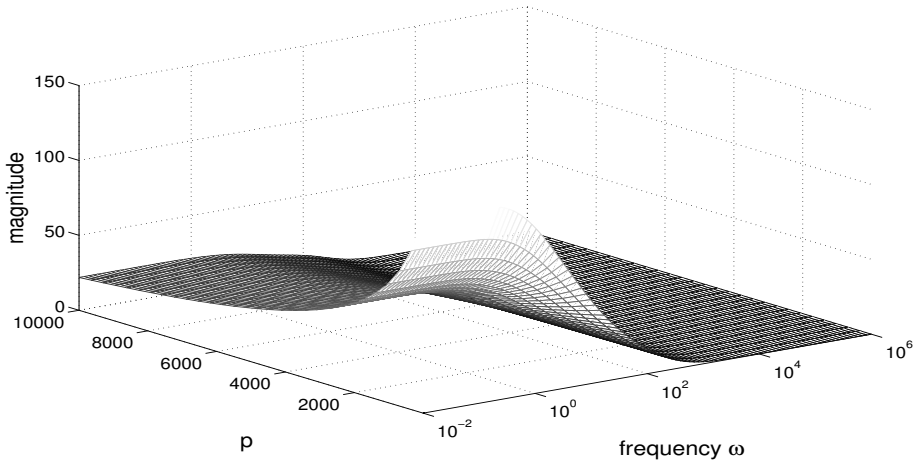

Figure 3: Frequency response of the microthruster.

5.3 Anemometer[4]

An anemometer is a flow sensing device that consists of a heater and temperature sensors placed both before and after the heater, either directly in the flow or in its vicinity. With no flow, the heat dissipates symmetrically into the fluid. This symmetry is disturbed if a flow is applied to the fluid, which leads to convection of the temperature field and therefore to a difference between the temperature sensors (Figure 5) from which the fluid velocity can be determined.

The physical model can be expressed by the convection-diffusion partial differential equation [31]:

$$\rho c\frac{\partial T}{\partial t} = \nabla\cdot(\kappa\nabla T) - \rho cv\nabla T + \dot{q}, \tag{10}$$

where ρ denotes the mass density, c is the specific heat, κ is the thermal conductivity, v is the fluid velocity, T is the temperature and $\dot{q}$ the heat flow into the system caused by the heater.

[4] http : //www.modelreduction.org/index.php/Anemometer

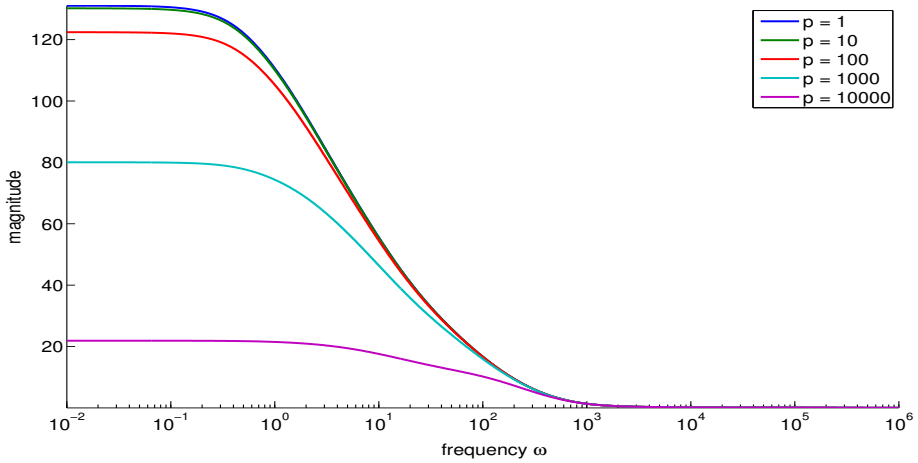

Figure 4: Frequency response of the microthruster for some parameter values.

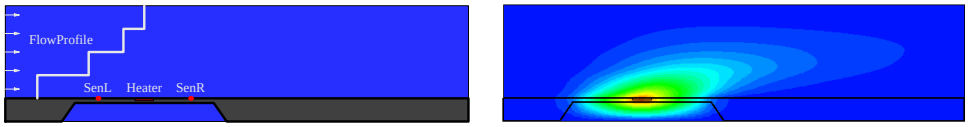

Figure 5: 2D model of an anemometer by courtesy of [32]. Left: schematics. Right: calculated temperature profile.

The model (10) is discretized in space using the finite element method (FEM) with triangular elements. The order of the discretized system is $n = 29,008$ after applying (zero) Dirichlet boundary conditions. The n dimensional ODE system has the following transfer function

$$G(s,p) = C(sE - A_0 - pA_1)^{-1}B$$

with the fluid velocity $p(= v)$ as scalar parameter. Here, E is the heat capacitance matrix, A_0 describes the thermal conduction and A_1 contains the convection terms which are given by a cascaded flow profile. B is the load vector which characterizes the spatial heat distribution into the fluid introduced by the heater. The initial conditions are set to zero. The dependency of the frequency response on the parameter $p \in [0,\ 1]$, i.e., $\bar{\sigma}(G(\imath\omega, p))$, can be seen in Figure 6 and in Figure 7 for some parameter points in $[0,\ 1]$. For more informations about the system see [32].

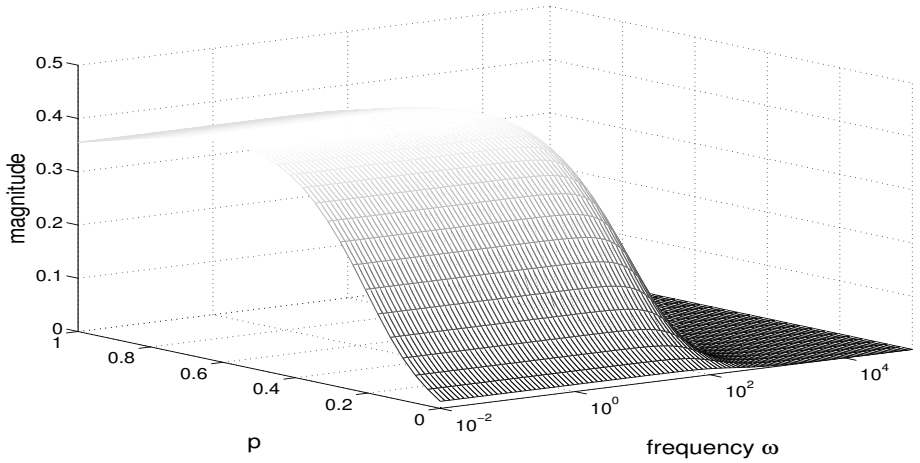

Figure 6: Frequency response of the anemometer.

6 Numerical results

All methods use the same initial sampling of the parameter space, i.e., the same choice of $\{p_1, \ldots, p_K\}$, for computing either snapshots (POD, emWX) or local reduced quantities (in interpolatory approaches). Furthermore, the reduced order in the parameterized, reduced-order system is either fixed by r (in POD, emWX) or determined by $K \cdot r'$, where r' is the local reduced order in the interpolatory approaches. In order to maintain comparability, we set $r = K \cdot r'$ for POD and emWX. The final reduced order in MatrInt is r' which leads then to smaller online costs.

All methods, except of TransFncInt and MatrInt, are computed with one-sided (Galerkin) projections in order to preserve the stability in the reduced-order system. Note that there is a loss in fidelity from using one-sided projections. TransFncInt

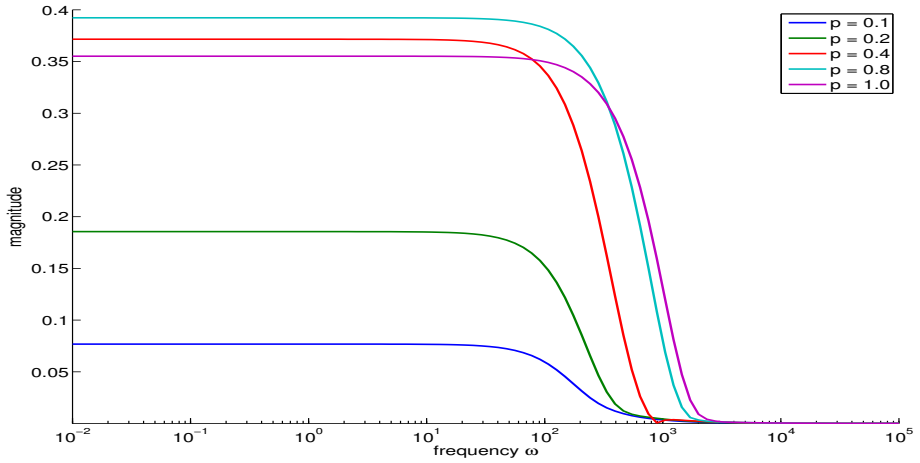

Figure 7: Frequency response of the anemometer for some parameter values.

used in combination with balanced truncation (i.e., with two-sided projections for computing the local reduced-order systems) guarantees the preservation of stability of the resulting parameterized reduced-order system [4]. Thus, this approach still uses two-sided (Petrov-Galerkin) projections. MatrInt uses a two-sided IRKA method followed by solving a low-order Lyapunov equation for each parameter sampling point in order to preserve stability [16, 17].

All error measures as described in Section 3 are computed and compared in the following, using an error grid with $\bar{K} = 100$ (with $K < \bar{K}$) parameter points. The pointwise (in p) errors are plotted in the corresponding figures in the following subsections. The maximum values of these errors (on the error grid) are listed in the Tables 1, 2 and 3. The time domain grid for all time domain errors equals the time discretization $t_i = i\Delta t$, $i = 0, \ldots, J$, in the snapshot-based PMOR approaches. A unit impulse is applied to all systems for computing the snapshots as well as the time domain error measures. For the computation of the frequency response, $\bar{L} = 100$ frequency points are taken.

6.1 Synthetic system

All PMOR approaches introduced in Section 2 are applied to the synthetic system with $n = 1000$ from Section 5.1 in order to compute parameterized, reduced-order systems. To this end, an initial discretization of the parameter interval $[0.1, 1]$ is required. We choose an equidistant grid with 4 sampling points for the computation of snapshots as well as for the computation of local reduced quantities for interpolation. In the interpolatory approaches, the local reduced order is $r' = 25$. This leads to a reduced order of $r = 100$ in all approaches except of MatrInt where a summation over the reduced system matrices leads to an overall reduced dimension of $r = 25$. The reduced order in all approaches based on snapshots is also set to 100.

The state is computed by using the MATLAB® solver ode45 (an explicit Runge-Kutta method) with a chosen relative error tolerance of 1.e-6 (and absolute tolerance 1.e-8) on the time interval $[0,\ 1]$. The frequency-domain error measures are computed on $[\imath 1,\ \imath 10^{3.1})$, the domain where the frequency responses differ along the parameter interval, see Figure 2. In MultiPMomMtch also an initial discretization of the considered frequency range is required. We took 16 logarithmically scaled frequency expansion points $s_1, \ldots, s_L$. The highest number of moments, included in the projection matrix V, is chosen as $q = 2$.

The results can be found in Table 1 and in Figures 8-13. A discussion of all results can be found in Section 6.4.

method	r	$\mathcal{L}_2$-state	$\mathcal{L}_2$-output	$\mathcal{H}_\infty$	$\mathcal{H}_2$
POD	100	3.2e-7	3.3e-7	1.1e-1	1.1e-2
POD-Greedy	100	3.2e-7	3.5e-7	1.2e-1	1.1e-2
MatrInt	25	1.0e+0	4.1e-1	∞	∞
TransFncInt	100	-	3.7e-7	3.0e-2	5.6e-4
PWH2TanInt	100	3.0e-3	4.2e-5	1.1e-2	6.5e-3
MultiPMomMtch	100	2.4e+0	1.5e+0	3.2e-2	2.1e-2
emWX	100	3.4e-7	3.5e-7	1.1e-1	1.0e-2

Table 1: Synthetic results for all PMOR methods considered (relative errors).

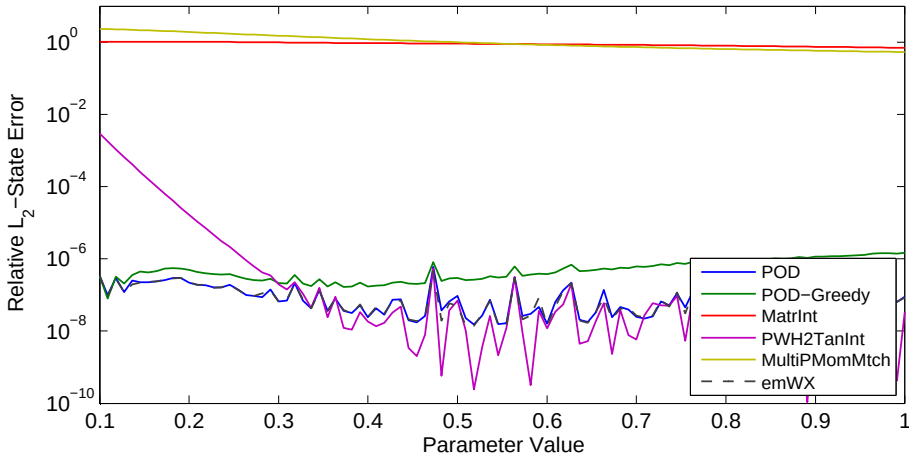

Figure 8: Relative $\mathcal{L}_2$-state error for the synthetic system.

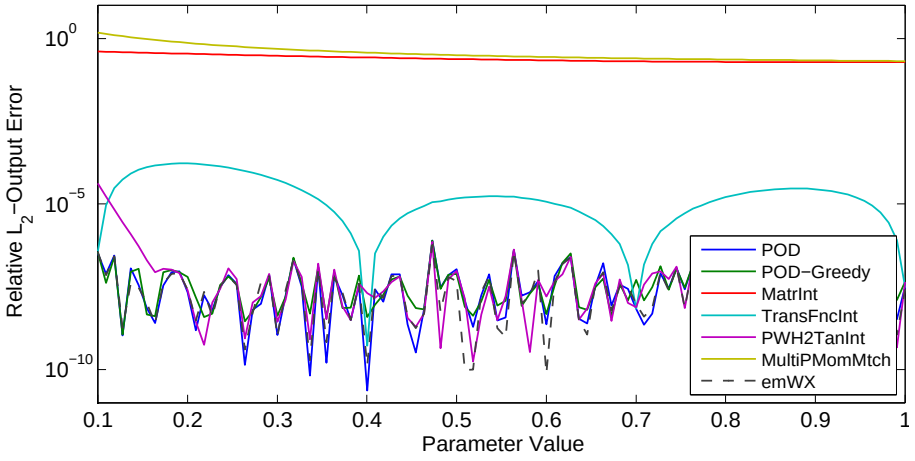

Figure 9: Relative $\mathcal{L}_2$-output error for the synthetic system.

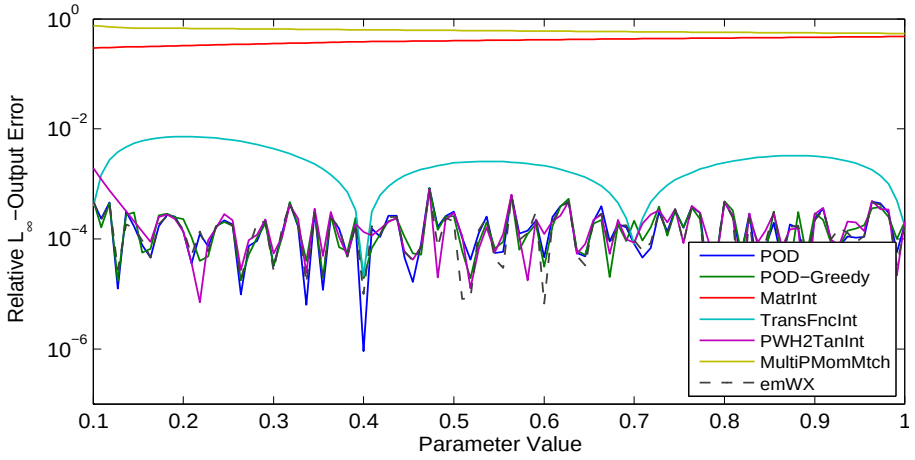

Figure 10: Relative $\mathcal{L}_\infty$-output error for the synthetic system.

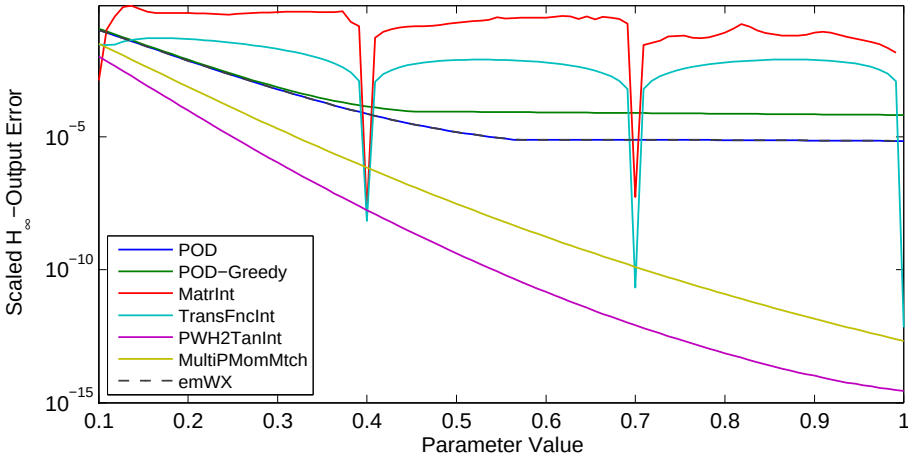

Figure 11: Scaled $\mathcal{H}_\infty$-error for the synthetic system.

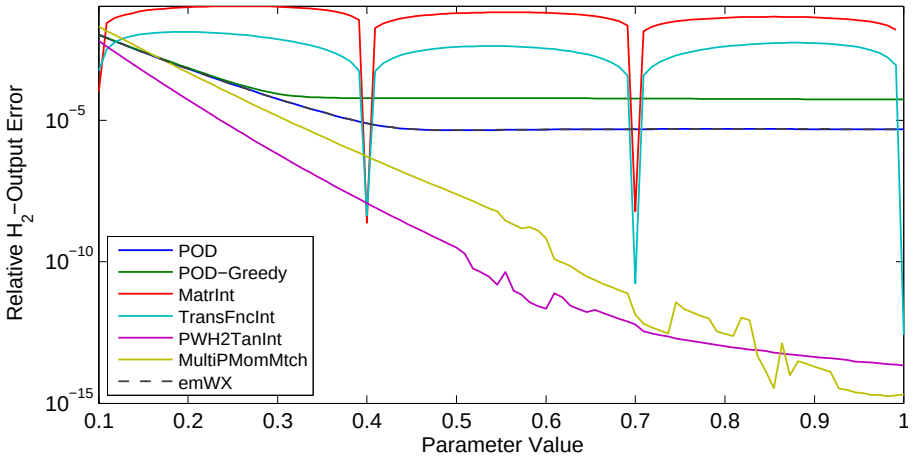

Figure 12: Relative $\mathcal{H}_2$-error for the synthetic system.

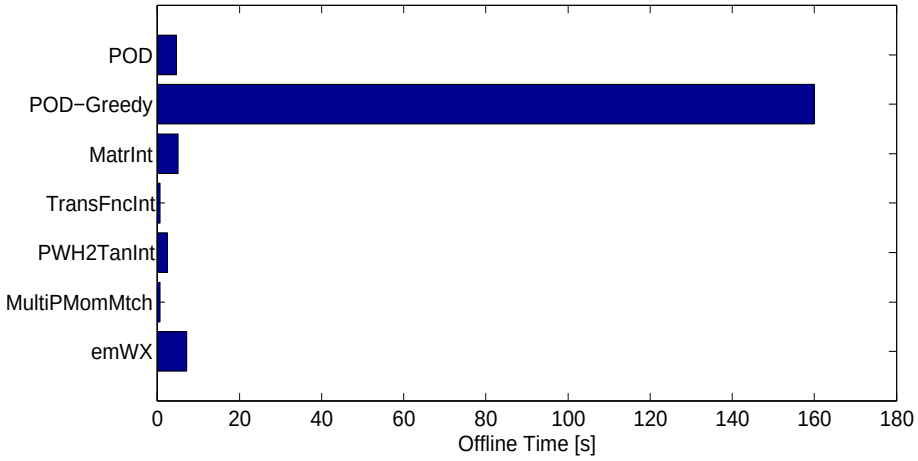

Figure 13: Offline times for the synthetic system.

6.2 Microthruster unit

The film coefficient p at the top interface is in the interval $[1,\ 10000]$. This parameter interval is discretized with 10 parameter sampling points for computing either snapshots, locally reduced-order systems for interpolation or Krylov subspaces for projection. The local reduced order is set to $r' = 10$. This gives a reduced order of $r = 100$ (MatrInt $r = 10$) in the parameterized reduced-order system. We consider a frequency range of $[\imath\ 0.001, \imath\ 10^6]$ and select 4 points in the interval as expansion points in MultiPMomMtch. The highest order of matched moments is prescribed by $q = 2$.

Trajectories and time-domain errors are computed in the time horizon $[0,\ 20]$ with constant time steps of size $\Delta t = 0.1$ (200 time points) by use of the backward Euler method.

See Table 2 and Figures 14-19 for the results achieved.

method	r	$\mathcal{L}_2$-state	$\mathcal{L}_2$-output	$\mathcal{H}_\infty$	$\mathcal{H}_2$
POD	100	1.4e-19	1.5e-19	2.9e-3	9.2e-2
POD-Greedy	100	2.6e-19	2.3e-19	1.3e-3	9.0e-2
MatrInt	10	7.3e-1	2.9e-3	1.1e-1	5.0e-2
TransFncInt	100	-	8.7e-6	9.2e-3	4.1e-2
PWH2TanInt	100	2.0e-8	1.2e-12	3.3e-6	2.3e-2
MultiPMomMtch	100	3.2e-4	7.5e-5	2.9e-2	3.5e-2
emWX	100	3.8e-7	6.6e-9	1.5e-2	3.4e-1

Table 2: Microthruster results for all PMOR methods considered (relative errors).

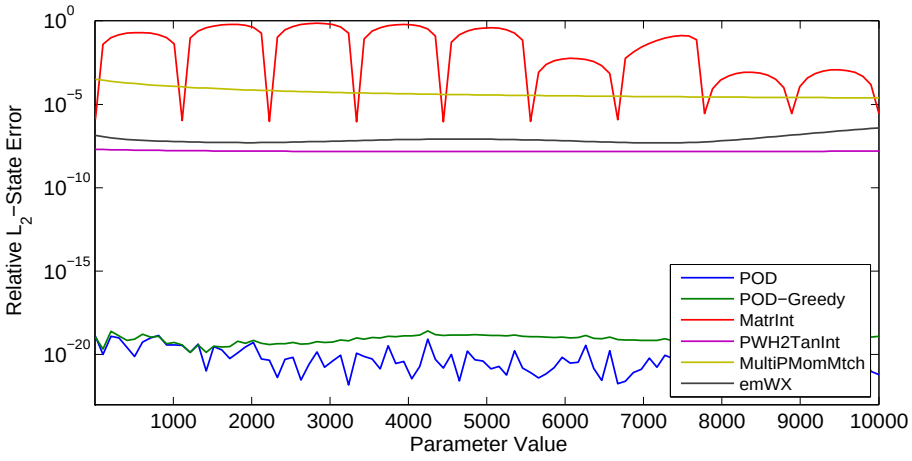

Figure 14: Relative $\mathcal{L}_2$-state error for the microthruster.

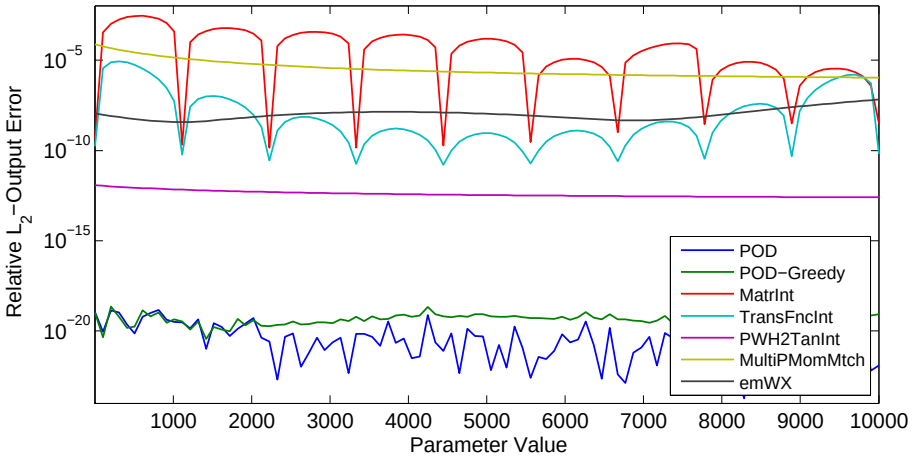

Figure 15: Relative $\mathcal{L}_2$-output error for the microthruster.

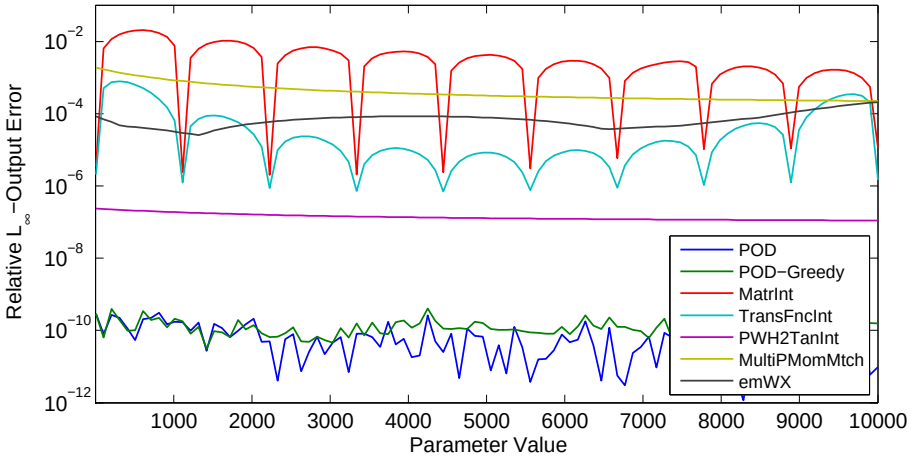

Figure 16: Relative $\mathcal{L}_\infty$-output error for the microthruster.

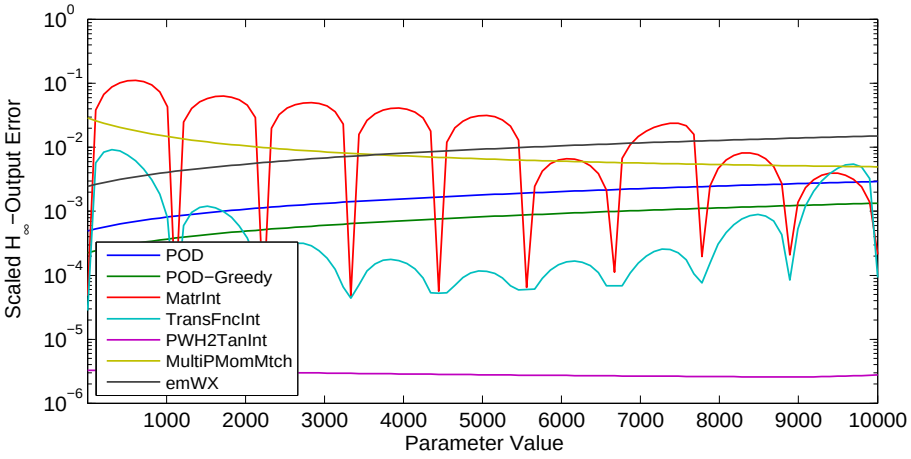

Figure 17: Scaled $\mathcal{H}_\infty$-error for the microthruster.

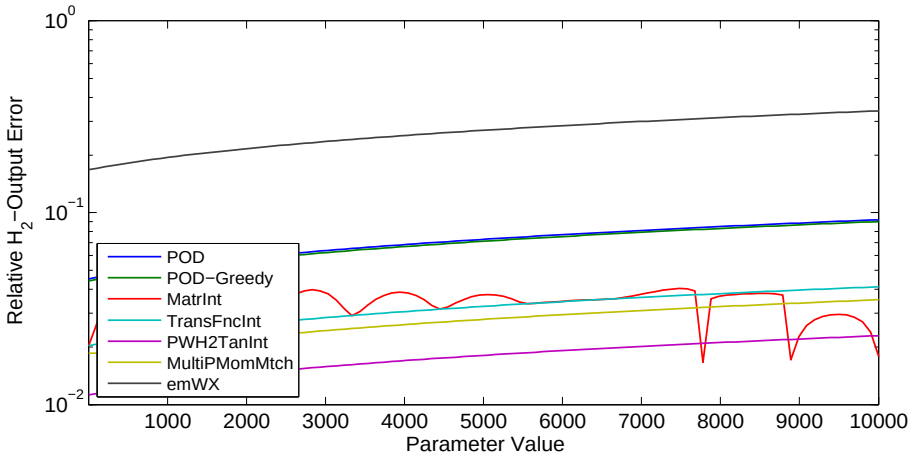

Figure 18: Relative $\mathcal{H}_2$-error for the microthruster.

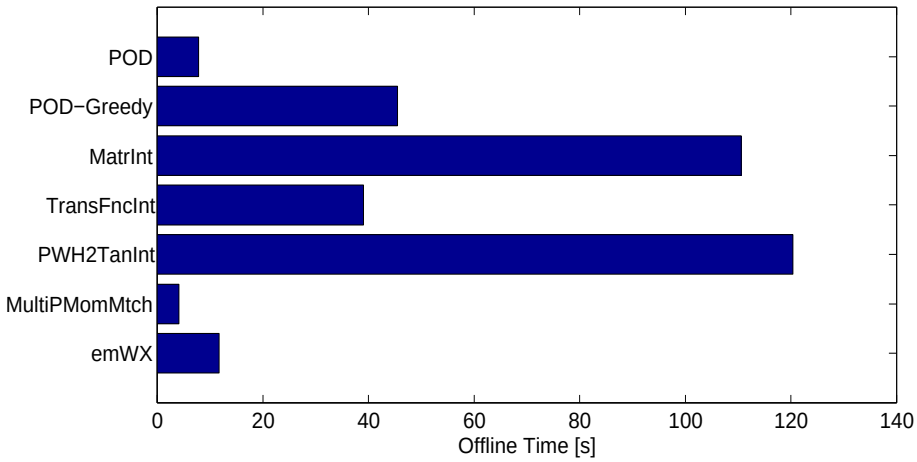

Figure 19: Offline time for the microthruster.

6.3 Anemometer

We consider the discretized convection-diffusion equation (10) from Section 5.3, i.e., a sparse ODE system of order $n = 29,008$ with one parameter $p \in [0,\ 1]$ that influences the convection. 16 parameter sampling points are taken from this interval. The local reduced order is set to $r' = 10$ which leads to $r = 160$ for all approaches except of MatrInt (with $r = 10$). The state and output errors are computed for the time horizon $[0,\ 0.05]$ with 50 time points. Like in the microthruster example, the trajectories are computed by the backward Euler method with constant step size of 0.001.

The errors in frequency domain are computed in the interval $[\imath\, 0.01,\ \imath\, 10^5]$. A choice of 10 frequency expansion points and $q = 2$ is taken for MultiPMomMtch. This gives a dimension of 480 in the projection matrix V which is truncated to $r = 160$ in order to get a comparable online time.

method	r	$\mathcal{L}_2$-state	$\mathcal{L}_2$-output	$\mathcal{H}_\infty$	$\mathcal{H}_2$
POD	160	1.6e-14	3.0e-11	9.3e-2	3.0e+0
POD-Greedy	160	1.4e-14	9.0e-11	3.5e-2	5.8e-1
MatrInt	10	6.9e+1	6.4e-3	1.1e-1	8.3e-2
TransFncInt	160	-	8.2e-3	2.3e-1	3.5e-1
PWH2TanInt	160	2.5e-6	2.9e-10	6.0e-5	2.5e-3
MultiPMomMtch	160	2.3e-7	2.9e-6	3.8e-1	1.3e+1
emWX	160	2.3e-7	1.3e-3	1.5e-1	1.3e+0

Table 3: Anemometer results for all PMOR methods considered (relative errors).

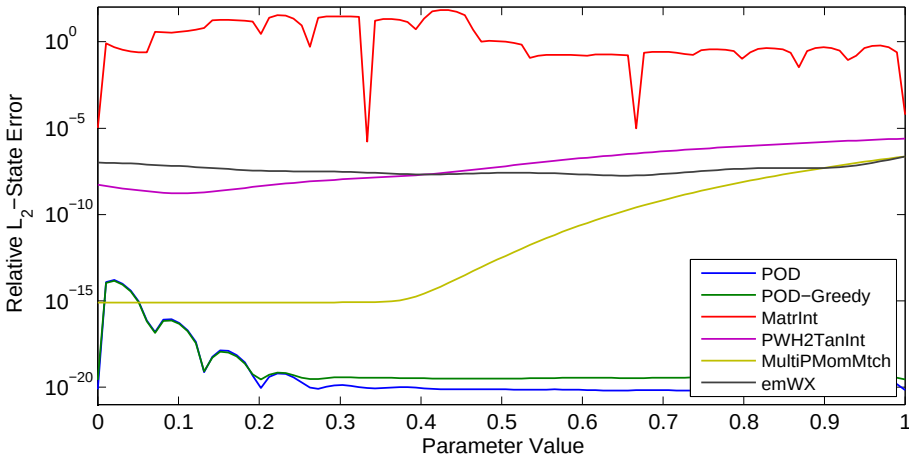

Figure 20: Relative $\mathcal{L}_2$-state error for the anemometer.

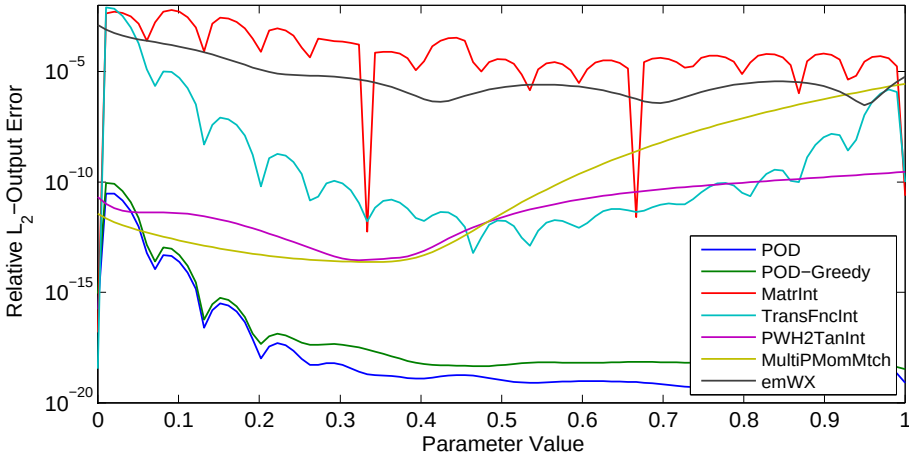

Figure 21: Relative $\mathcal{L}_2$-output error for the anemometer.

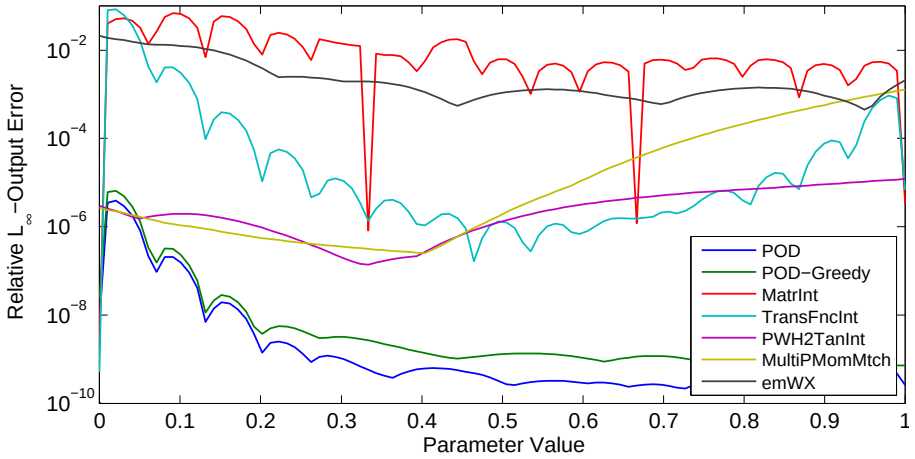

Figure 22: Relative $\mathcal{L}_\infty$-output error for the anemometer.

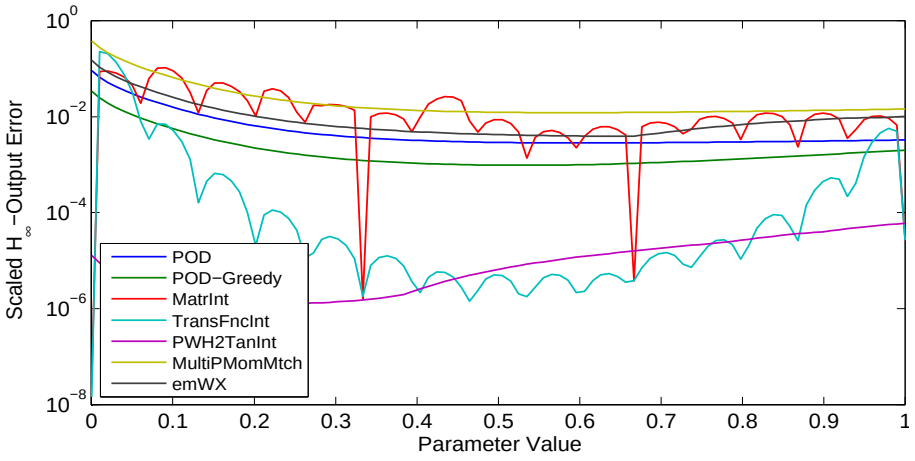

Figure 23: Scaled $\mathcal{H}_\infty$-error for the anemometer.

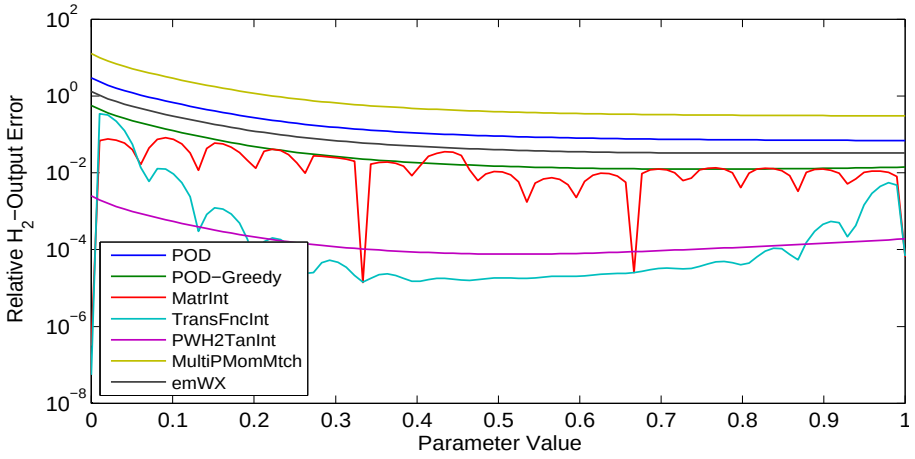

Figure 24: Relative $\mathcal{H}_2$-error for the anemometer.

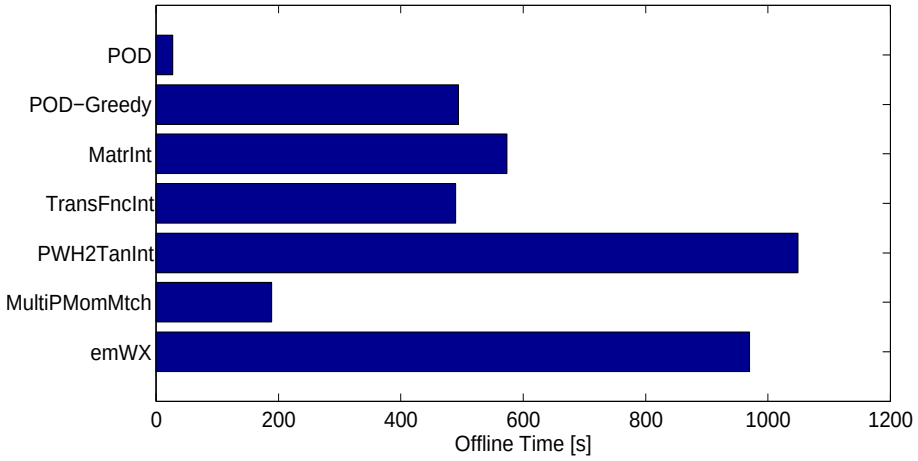

Figure 25: Offline times for the anemometer.

6.4 Discussion of the results

The error measures from Section 3 are computed for all the PMOR methods considered here as applied to the benchmarks described in Section 5.

6.4.1 Time domain errors

Figures 8, 14 and 20 show the pointwise $\mathcal{L}_2$-errors in state space. Note that state-space errors for TransFncInt cannot be computed because of of the absence of a lifting map that maps the reduced state back to the original (full order) state space for the realization considered. It can be seen that the results for POD and POD-Greedy are consistent with expectations. The errors appear to be uniform and very small throughout the parameter interval. The size of the error evidently depends on the accuracy of the chosen integrator, i.e., it is smaller than machine precision when an implicit solver is used (anemometer and microthruster) and is about 1.e-7 (6.e-7 for POD-Greedy) by restricting the absolute tolerance to 1.e-8 in the explicit MATLAB® solver ode45 (for the synthetic system).

It can be seen in Figure 8 that emWX and PWH2TanInt perform comparably well to POD for the synthetic system. MatrInt and MultiPMomMtch fail to reproduce the state.

The errors in state space are smaller for the microthruster in Figure 14. The errors of MultiPMomMtch, PWH2TanInt and emWX are very uniform over p with average size of 2.e-8 (PWH2TanInt), 6.e-5 (MultiPMomMtch) and 9.e-8 (emWX). For MatrInt, the errors are of moderate size (about 1.e-5) at the parameter sampling points but very much worse (close to 1) between them.

PWH2TanInt and emWX approximate the state uniformly well with a small average error size of 5.e-7 in PWH2TanInt and 5.e-8 in emWX for the anemometer example,

see Figure 20. Here, the state-space error is smaller (about 8.e-16) in MultiPMomMtch for $p \in [0,\ 0.4]$ and is increasing to 2.e-7 in $[0.4,\ 1]$.

Regarding the output errors, POD, POD-Greedy, PWH2TanInt and emWX perform comparably well for the synthetic system in Figures 9, 10. MatrInt and MultiPMomMtch perform worst for the synthetic system as could be expected by the bad approximation of the state in Figure 8.

In the other two benchmarks, the POD methods yield better approximations than all other approaches, see Figures 15, 16, 21, 22. The $\mathcal{L}_2$-output errors for PWH2TanInt, MultiPMomMtch and emWX are uniformly distributed over the interval for the microthruster benchmark in Figure 15 but of different size on average: 4.e-13 (PWH2TanInt), 6.e-6 (MultiPMomMtch) and 1.e-8 (emWX). The errors obtained by TransFncInt are of medium size. The error curve is wavelike with minima at the parameter sampling points and with average size of 5.e-7.

MultiPMomMtch approximates the output in the anemometer benchmark much better than MatrInt and of similar quality like PWH2TanInt. The errors obtained by TransFncInt are of medium size. The errors for emWX are larger for the anemometer example. In most of the cases, we observe that the approximation quality for the states and the outputs are related. An exception are the PWH2TanInt and emWX methods for the anemometer. Although the $\mathcal{L}_2$-state errors of the two methods are comparable, the output errors are several orders of magnitude lower for the PWH2TanInt, see Figure 22.

6.4.2 Frequency domain errors

PWH2TanInt computes the smallest $\mathcal{H}_\infty$-errors for all benchmarks considered. The error is about 3.e-6 for the microthruster example and 2.e-5 on average for the anemometer and uniform over the parameter interval, see Figures 17 and 23. It is nearly exponentially decreasing for the synthetic benchmark with an error of 1.e-2 for $p = 0.1$ in Figure 11. This can be explained by the shape of the frequency response in Figures 1 and 2. The frequency response slightly oscillates for $p = 0.1$ which makes the approximation at the beginning of the parameter interval much more difficult.

The frequency domain errors of MatrInt and TransFncInt show the expected behavior for all examples. The error curves have minima at the parameter sampling points, showing waves between them. The errors are smaller in TransFncInt, especially in the anemometer and microthruster examples. Both interpolatory approaches produce larger errors between the sampling points for the synthetic system. These results also can be explained by the frequency response of the system, given in Figure 1. Note that the reduced synthetic system obtained by applying MatrInt is unstable at $p = 1$ (indicated by ∞ in Table 1). The preservation of stability fails because of approximation errors in the numerical solution of the Lyapunov equation. A reduction to a smaller local dimension, say $r' = 18$, would result in a stable reduced-order system. However, individual variations in r' are beyond the scope of this comparison.

The $\mathcal{H}_\infty$-errors in POD, POD-Greedy and emWX appear to be uniform in p and lie, on average, between 1.e-2 and 8.e-4 for all examples. The frequency domain error curves of POD and emWX nearly coincide for the synthetic benchmark.

The errors in MultiPMomMtch are uniformly distributed for the microthruster and the anemometer. With an average size of 9.e-3 it is of similar approximation quality like emWX for the microthruster benchmark. For the anemometer example, MultiPMomMtch produces a uniformly distributed larger error of about 3.e-2, the average error size in emWX is 1.e-2. The $\mathcal{H}_\infty$-error in MultiPMomMtch is much smaller and exponentially decreasing in the synthetic benchmark. This is due to a reduced frequency range (up to 10^3 instead of 10^5 and 10^6 in the anemometer and in the microthruster benchmark, respectively). MultiPMomMtch is the only approach which requires an initial discretization of the frequency range. Thus, a smaller domain results in smaller errors and more frequency expansion points are recommended if larger frequency domains are considered.

The $\mathcal{H}_2$-errors are plotted in Figures 12, 18 and 24. The error curves in Figure 12 show a very similar behavior compared to the $\mathcal{H}_\infty$-errors for the synthetic system (even with smaller maximum values, see Table 1) in Figure 11. The errors for the microthruster in Figure 18 are of comparable size between 2.e-2 and 3.e-1 in all approaches and again of similar shape compared to the $\mathcal{H}_\infty$-error curves for the anemometer. Here, MultiPMomMtch and POD produce errors larger than 1.

6.4.3 Computational time

The offline computational costs for the (small) synthetic system in Figure 13 are low for all methods except of POD-Greedy. Since we considered only 4 training parameters for the synthetic system, the number of iterations (and so the offline time) in the POD-Greedy approach could be reduced drastically by choosing more POD modes in each iteration.

For the medium-size benchmark microthruster in Figure 19, POD, MultiPMomMtch and emWX are efficient with respect to the offline time. MatrInt, TransFncInt, PWH2TanInt and POD-Greedy have higher offline costs. The main complexity of the offline phase in the interpolatory approaches comes from applying K-times a deterministic MOR method on $G(s, p_j)$. The times are comparable for MatrInt and PWH2TanInt since both approaches use IRKA. TransFncInt uses BT which is (on average) faster than IRKA for the benchmarks considered here.

It can be seen in Figure 25 that for the anemometer, an example with larger original system size, again POD and MultiPMomMtch have the lowest offline times. The snapshot-based methods benefit here (and in the microthruster example) from an efficient implementation of the backward Euler method. A single LU decomposition per parameter value can be used to solve the linear systems because of the choice of a constant time step size. MultiPMomMtch has very low offline costs since it only requires a few factorizations of (sparse) matrices followed by a number of forward/backward solves. The anemometer offline times for POD-Greedy, MatrInt and TransFncInt are higher but nearly the half of the times for PWH2TanInt and emWX. The computational cost for emWX arises from three sources. First, the computation of snapshots; second, the Gramian matrix assembly and lastly, the SVD of the cross Gramian which is also the dominant component of the computational time for large-scale systems. In PWH2TanInt, the computation of an SVD of $V \in \mathbb{R}^{n \times Kr'}$ leads to the highest offline

time.

The averaged (over the parameter interval) online times for the computation of the transient and of the frequency response with corresponding break-even quantities are shown in Tables 4-6 for the three benchmarks. The break-even quantity of a PMOR approach is the number of online simulations (in frequency or in time) such that offline plus online time is smaller than the simulation time (again in frequency or in time) of the original system.

The simulation (online) time in frequency domain is not computed for the small synthetic benchmark in Table 4. Here, it requires more time to compute the frequency response of the reduced system than to simulate the full (but sparse) original system in frequency domain. This is different for the larger benchmarks where the online complexity in frequency domain and the corresponding break-even quantities can be found in Tables 5-6.

The online times of all PMOR approaches, i.e., the simulation times of the computed reduced-order systems, are comparable when the reduced dimensions are equal. This is the case for all approaches considered except of MatrInt. The reduced order of the system computed by MatrInt is much smaller. This results in a smaller online simulation time for systems of smaller dimension. For larger systems, the additional online steps in MatrInt (and TransFncInt) are visible in the (transient) online times, see Section 4 for details. Note that the transient online costs for the microthruster and the anemometer benefit also from the constant time step in the ODE solver. Here, the application of PMOR only pays off for simulations in time domain when the transient response has to be computed many times. This would be different (smaller break-even quantities) when the time steps vary and also when simulations are computed in the frequency domain. This can be seen by small frequency-domain break-even quantities which are in a range from 1 to 33, see Tables 5-6. There are also differences between the transient online times for reduced systems of the same dimension in Table 4. The higher costs in MultiPMomMtch and in TransFncInt for the synthetic benchmark can be explained by a closer look at the number of time steps of the ODE solver. MultiPMomMtch needs (in average) about 540 time steps until the solution is accurate to the given error tolerances. TransFncInt needs 500 steps, the other approaches between 116 and 422. The break-even quantities in time domain show that all approaches except of POD-Greedy are useful when applied to the small synthetic benchmark. POD-Greedy in this scenario obviously should be run with $r' > 1$ in order to considerably reduce the offline time.

		transient	
method	offline	online	break-even
original	-	0.77	-
POD	4.64	0.25	9
POD-Greedy	160.01	0.25	309
MatrInt	5.00	0.06	7
TransFncInt	0.60	0.28	2
PWH2TanInt	2.49	0.25	5
MultiPMomMtch	0.61	0.31	1
emWX	7.14	0.25	14

Table 4: Simulation times in the synthetic benchmark for all PMOR methods considered.

		transient		frequency	
method	offline	online	break-even	online	break-even
original	-	0.20	-	5.25	-
POD	7.83	0.0078	40	0.16	2
POD-Greedy	45.48	0.0078	231	0.16	9
MatrInt	110.60	0.011	570	0.01	22
TransFncInt	39.06	0.0088	200	0.11	8
PWH2TanInt	120.39	0.0077	611	0.16	24
MultiPMomMtch	4.06	0.0077	21	0.15	1
emWX	11.73	0.0077	60	0.15	3

Table 5: Simulation times in the microthruster benchmark for all PMOR methods considered.

		transient		frequency	
method	offline	online	break-even	online	break-even
original	-	1.39	-	32.13	-
POD	26.88	0.0058	17	0.16	1
POD-Greedy	494.36	0.0071	311	0.20	16
MatrInt	573.56	0.0393	369	0.04	18
TransFncInt	489.41	0.0125	309	0.18	16
PWH2TanInt	1049.20	0.0073	660	0.21	33
MultiPMomMtch	188.55	0.0074	119	0.22	6
emWX	970.26	0.0096	612	0.24	31

Table 6: Simulation times in the anemometer benchmark for all PMOR methods considered.

7 Conclusions

POD seems to give the best results for state-space approximations. However, it may not be feasible if the number of training parameters or the dimension of the state is too large. In this case, POD-Greedy should be preferred.

PWH2TanInt computes best approximations in the frequency domain and also provides good results with respect to the time-domain error measures. However, it requires larger offline times when applied to large-scale systems like the anemometer. So it will depend on the application (on the number of computed simulations) if a reduction by PWH2TanInt pays off for larger systems. TransFncInt performs well for mostly all error measures and benchmarks considered. The errors in frequency domain are small for the anemometer and the microthruster as expected. TransFncInt has problems when applied to transfer functions with peaks as in the synthetic system. Surprisingly, the output errors for this example are relatively small even between the parameter sampling points. MultiPMomMtch is very efficient with respect to the offline time. However, it requires a good tuning and therefore knowledge about sensitive regions in frequency and parameter space. Much smaller errors could be obtained by a more sophisticated choice of expansion points. The results obtained by applying MatrInt do not fit exactly in the framework of this comparison. This is due to a much smaller reduced order which, not surprisingly, leads to larger errors. MatrInt is well-adapted to systems with dominant eigenmodes since mode veering and crossing can be recognized by the approach. Furthermore, MatrInt can be applied to all kinds of parameter dependency, even if they are not analytically given. This is not the case for the benchmarks considered such that MatrInt could not prove its strengths in this work.

emWX exhibits a nearly constant behaviour over the tested parameter space for most of the different error norms and benchmarks. The expected better match in the frequency space norms in comparison with the POD methods was not achieved, however. Improvements to the accuracy (especially in the frequency domain errors) can be achieved in three ways: including more snapshots, using two-sided projections, and utilizing an enhanced decomposition algorithm to obtain the projection matrices.

An extension of these experiments to MIMO and multi-parameter systems is a necessary step for future work. Future work could also include a more challenging computation of the time domain error measures by using different inputs for snapshot and error computation in and by extending the time interval.

Acknowledgments

We would like to thank Lihong Feng, Matthias Geuß and Heiko Panzer for providing their code and for helpful discussions and Chris Beattie for reading a draft version of this manuscript and giving various advices for improvement.

References

[1] *MOR Wiki - Model Order Reduction Wiki.* http://www.modelreduction.org,

2014.

[2] D. Amsallem and C. Farhat, *An online method for interpolating linear parametric reduced-order models*, SIAM J. Sci. Comput., 33 (2011), pp. 2169–2198.

[3] U. Baur, C. A. Beattie, P. Benner, and S. Gugercin, *Interpolatory projection methods for parameterized model reduction*, SIAM J. Sci. Comput., 33 (2011), pp. 2489–2518.

[4] U. Baur and P. Benner, *Modellreduktion für parametrisierte Systeme durch balanciertes Abschneiden und Interpolation (Model Reduction for Parametric Systems Using Balanced Truncation and Interpolation)*, at-Automatisierungstechnik, 57 (2009), pp. 411–420.

[5] U. Baur, P. Benner, A. Greiner, J. G. Korvink, J. Lienemann, and C. Moosmann, *Parameter preserving model reduction for MEMS applications*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 297–317.

[6] P. Benner and T. Breiten, *Model order reduction based on system balancing*, in Model Reduction and Approximation for Complex Systems, Luminy book, 2015.

[7] P. Benner, S. Gugercin, and K. Willcox, *A survey of model reduction methods for parametric systems*, Preprint MPIMD/13-14, Max Planck Institute Magdeburg, Aug. 2013. Available from http://www.mpi-magdeburg.mpg.de/preprints/.

[8] G. Berkooz, P. Holmes, and J. L. Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annual review of fluid mechanics, 25 (1993), pp. 539–575.

[9] L. Daniel, O. C. Siong, L. S. Chay, K. H. Lee, and J. White, *A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models*, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 23 (2004), pp. 678–693.

[10] J. L. Eftang, D. J. Knezevic, and A. T. Patera, *An hp certified reduced basis method for parametrized parabolic partial differential equations*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 395–422.

[11] O. Farle, V. Hill, P. Ingelström, and R. Dyczij-Edlinger, *Multi-parameter polynomial order reduction of linear finite element models*, Math. Comput. Model. Dyn. Syst., 14 (2008), pp. 421–434.

[12] L. Feng and P. Benner, *Parametrische Modellreduktion durch impliziten Momentenabgleich*, in Tagungsband GMA-FA 1.30 'Modellbildung, Identifizierung und Simulation in der Automatisierungstechnik', Workshop in Anif, 26.-28.9.2007, B. Lohmann and A. Kugi, eds., 2007, pp. 34–47.

[13] ——, *Reduced Order Methods for modeling and computational reduction, MS&A Series*, vol. 9, Springer-Verlag, Berlin, Heidelberg, New York, 2014, ch. 6: A robust algorithm for parametric model order reduction based on implicit moment matching, pp. 159–186.

[14] K. V. Fernando and H. Nicholson, *On the structure of balanced and other principal representations of SISO systems*, IEEE Trans. Automat. Control, 28 (1984), pp. 228–231.

[15] K. V. Fernando and H. Nicholson, *On the cross-gramian for symmetric MIMO systems*, IEEE Trans. Circuits Syst., 32 (1985), pp. 487–489.

[16] M. Geuss, H. Panzer, A. Wirtz, and B. Lohmann, *A general framework for parametric model order reduction by matrix interpolation*, in Workshop on Model Reduction of Parametrized Systems II (MoRePaS II), 2012.

[17] M. Geuss, H. Panzer, T. Wolf, and B. Lohmann, *Stability preservation for parametric model order reduction by matrix interpolation*, in Proc. European Control Conf. ECC 2014, Strasbourg, 2014, pp. 1098–1103.

[18] S. Gugercin and A. C. Antoulas, *A comparative study of 7 algorithms for model reduction*, in Proc. 39th IEEE Conference on Decision and Control, Sydney, Australia, vol. 3, 2000, pp. 2367–2372.

[19] S. Gugercin, A. C. Antoulas, and C. A. Beattie, *$\mathcal{H}_2$ model reduction for large-scale dynamical systems*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 609–638.

[20] B. Haasdonk, *Convergence rates of the POD-Greedy method*, ESAIM: Math. Model. Numer. Anal., 47 (2013), pp. 859–873.

[21] ——, *Reduced basis methods for parametrized PDEs - a tutorial introduction for stationary and instationary problems*, in Model Reduction and Approximation for Complex Systems, Luminy book, 2015.

[22] B. Haasdonk, M. Dihlmann, and M. Ohlberger, *A training set and multiple basis generation approach for parametrized model reduction based on adaptive grids in parameter space*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 423–442.

[23] B. Haasdonk and M. Ohlberger, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, ESAIM: Math. Model. Numer. Anal., 42 (2008), pp. 277–302.

[24] ——, *Efficient reduced models and a-posteriori error estimation for parametrized dynamical systems by offline/online decomposition*, Math. Comput. Model. Dyn. Syst., 17 (2011), pp. 145–161.

[25] C. Himpe and M. Ohlberger, *Cross-gramian based combined state and parameter reduction for large-scale control systems*, Mathematical Problems in Engineering, 2014 (2014), pp. 1–13.

[26] D. J. Knezevic and A. T. Patera, *A certified reduced basis method for the Fokker-Planck equation of dilute polymeric fluids: FENE dumbbells in extensional flow.*, SIAM J. Sci. Comput., 32 (2010), pp. 793–817.

[27] J. G. Korvink and E. B. Rudnyi, *Oberwolfach benchmark collection*, in Dimension Reduction of Large-Scale Systems, P. Benner, D. C. Sorensen, and V. Mehrmann, eds., vol. 45 of Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, 2005, pp. 311–315.

[28] S. Lall, J. E. Marsden, and S. Glavaški, *Empirical model reduction of controlled nonlinear systems*, in Proc. of the IFAC World Congress, vol. F, 1999, pp. 473–478.

[29] S. Lall, J. E. Marsden, and S. Glavaški, *A subspace approach to balanced truncation for model reduction of nonlinear control systems*, Internat. J. Robust and Nonlinear Cont., 12 (2002), pp. 519–535.

[30] Y. T. Li, Z. Bai, Y. Su, and X. Zeng, *Parameterized model order reduction via a two-directional Arnoldi process*, in Proc. of the 2007 IEEE/ACM international conference on Computer-aided design, IEEE Press Piscataway, NJ, USA, 2007, pp. 868–873.

[31] C. Moosmann, E. B. Rudnyi, and J. G. K. A. Greiner, *Model order reduction for linear convective thermal flow*, in THERMINIC 2004, Sophia Antipolis, France, 2004, pp. 317–321.

[32] C. Moosmann, E. B. Rudnyi, A. Greiner, J. G. Korvink, and M. Hornung, *Parameter preserving model order reduction of a flow meter*, in 2005 NSTI Nanotech, Nanotechnology Conference and Trade Show 2005, 2005, pp. 8–12.

[33] H. Panzer, J. Mohring, R. Eid, and B. Lohmann, *Parametric model order reduction by matrix interpolation*, at-Automatisierungstechnik, 58 (2010), pp. 475–484.

[34] E. B. Rudnyi and J. G. Korvink, *Boundary condition independent thermal model*, in Dimension Reduction of Large-Scale Systems, P. Benner, D. C. Sorensen, and V. Mehrmann, eds., vol. 45 of Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, 2005, pp. 345–348.

[35] L. Sirovich, *Turbulence and the dynamics of coherent structures. I. coherent structures*, Quart. Appl. Math., 45 (1987), pp. 561–571.

[36] S. Volkwein, *Model reduction using proper orthogonal decomposition*, lecture notes, University of Konstanz, 2013.

[37] S. Waldherr and B. Haasdonk, *Efficient parametric analysis of the chemical master equation through model order reduction*, BMC Systems Biology, 6 (2012), p. 81.

[38] D. S. Weile, E. Michielssen, E. Grimme, and K. Gallivan, *A method for generating rational interpolant reduced order models of two-parameter linear systems*, Appl. Math. Lett., 12 (1999), pp. 93–102.

[39] K. Willcox and P. J., *Balanced model reduction via the proper orthogonal decomposition*, AIAA J., 40 (2002), pp. 2323–2330.

[40] Y. Zhang, L. Feng, S. Li, and P. Benner, *Accelerating PDE constrained optimization by the reduced basis method: application to batch chromatography*, Preprint MPIMD/14-09, Max Planck Institute Magdeburg, May 2014. Available from http://www.mpi-magdeburg.mpg.de/preprints/.

[41] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ, 1996.