September 19, 2014

Comparison of methods for parametric model order reduction of instationary problems
Part I

U. Baur*, P. Benner*, B. Haasdonk+, C. Himpe#, I. Martini+, M. Ohlberger#

* Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg
+ Institute of Applied Analysis and Numerical Simulation, University of Stuttgart
# Institute for Computational and Applied Mathematics, University of Münster

Outline

Comparison of methods for parametric model order reduction
of instationary problems
Part I

1. General framework
2. Methods for parametric model order reduction
3. Performance measure
4. Numerical results

General framework

Consider a parameterized dynamical system

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

with $p \in \mathbb{R}^d$ and $E(p)$, $A(p) \in \mathbb{R}^{n\times n}$, $B(p) \in \mathbb{R}^{n\times m}$, $C(p) \in \mathbb{R}^{\ell\times n}$.
The associated transfer function with $s \in \overline{\mathbb{C}}_+$ is

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

General assumptions:

- $E(p)$ invertible in parameter range
- stability
- affine parameter dependency

General framework

Assumptions in the current comparison:

- zero initial state
- single input, single output ($m = \ell = 1$)
- single parameter ($d = 1$)
- affine parameter dependency (in A)

This leads to a simplified parameterized system

$$\begin{aligned} E\,\dot{x}(t;p) &= (A_0 + p\,A_1)\,x(t;p) + B\,u(t), \\ y(t;p) &= C\,x(t;p). \end{aligned}$$

Parametric model order reduction (PMOR)

Replace original system by projection with $V,\ W \in \mathbb{R}^{n\times r}$, $r \ll n$,

$$\begin{aligned} W^T E V\,\dot{\hat{x}}(t;p) &= (W^T A_0 V + p\,W^T A_1 V)\,x(t;p) + W^T B\,u(t), \\ \hat{y}(t;p) &= C V\,\hat{x}(t;p). \end{aligned}$$

How to compute V, W?

Methods considered:

1. POD and POD-Greedy
2. Interpolatory methods (matrix interpolation, transfer function interpolation, piecewise $\mathcal{H}_2$ tangential interpolation, multi-parameter moment matching)
3. Empirical Cross Gramian

POD and POD-Greedy

Inputs for POD:

- training parameter samples $\{p_1, \ldots, p_K\}$
- time discretization $\{t_0, \ldots, t_J\}$
- training input $u(t)$

Compute snapshots $\{x_1 \ldots, x_N\} := \{x(t_0; p_1), \ldots, x(t_J; p_K)\}$ and obtain

$V = POD_r(x_1, \ldots, x_N) := \arg\min_V \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - VV^T x_i \right\|^2 \in \mathbb{R}^{n \times r}$,

with $V^T V = I$ computed by

$$[x_1, \ldots, x_N] \overset{\text{SVD}}{=} V_1 S V_2^T \text{ and } V = V_1[:, 1:r].$$

POD and POD-Greedy

Inputs for POD:

- training parameter samples $\{p_1, \ldots, p_K\}$
- time discretization $\{t_0, \ldots, t_J\}$
- training input $u(t)$

POD-Greedy [*Haasdonk/Ohlberger 08, Waldherr/Haasdonk 12*]:

1. start with initial basis V and corresponding reduced system
2. determine "worst" sample p_j
3. $V' = POD_{r'}\left(\{x(t_i; p_j) - VV^T x(t_i; p_j)\}_{i=0}^{J}\right)$
4. extend matrix $V := [V, V']$

Multi-parameter moment matching

Extension of moment matching to PMOR (MultiPMomMtch)

- Early approach [*Weile et al. 99, Gunupudi et al. 03, Daniel et al. 04, Moosmann/Korvink 06, Farle et al. 08, Feng/Benner 08,...*]
- multivariate Taylor expansion of $G(s,p)$ about $(s_i, p_j) \Rightarrow$ inputs: $s_i \in \{s_1, \ldots, s_L\}$, $p_j \in \{p_1, \ldots, p_K\}$
- avoid explicit moment matching, compute V via repeated modified Gram-Schmidt [*Feng/Benner 08*]
- matched multi-moments:

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

for $i = 1, \ldots, L,\ j = 1, \ldots, K,\ k = 0, \ldots, q,\ l = 0, \ldots, q.$

Interpolatory Methods for PMOR

1. Input: parameter samples $\{p_1, \ldots, p_K\}$
2. apply MOR, e.g., balanced truncation (BT), IRKA to

$$G(s, p_j) = C(s\,E - (A_0 + p_j\,A_1))^{-1}B, \quad j = 1, \ldots, K$$

3. store some of the reduced quantities ($j = 1, \ldots, K$):

- the projection matrices: $V_j,\ W_j \in \mathbb{R}^{n \times r_j}$
  [*piecewise $\mathcal{H}_2$ tangential interpolation*]
- reduced system matrices:
  [*matrix interpolation, transfer function interpolation*]

$$\begin{aligned} \hat{E}_j &= W_j^T E V_j \in \mathbb{R}^{r_j \times r_j} & \hat{A}_j &= W_j^T (A_0 + p_j\,A_1) V_j \in \mathbb{R}^{r_j \times r_j} \\ \hat{B}_j &= W_j^T B \in \mathbb{R}^{r_j \times m} & \hat{C}_j &= C V_j \in \mathbb{R}^{\ell \times r_j} \end{aligned}$$

4. interpolate to get reduced parametric system

Interpolatory Methods for PMOR

1. Matrix interpolation (MatrInt)
   [*Panzer/Mohring/Eid/Lohmann 10, Amsallem/Farhat 11*]
   transform matrices into generalized coordinates,
   small reduced order,
   guaranteed stability by solving low-dim. Lyapunov equations
   [*Geuss/Panzer/Wolf/Lohmann 14*]
2. Transfer function interpolation (TransFncInt) [*B./Benner 09*]
   reduction by BT, polynomial interpolation,
   guaranteed stability
3. Piecewise $\mathcal{H}_2$ tangential interpolation (PWH2TanInt)
   [*B./Beattie/Benner/Gugercin 11*]
   reduction by IRKA, projection on concatenated (global) basis

Empirical Cross Gramian (emWX)

Inputs for emWX:

- training parameter samples $\{p_1, \ldots, p_K\}$
- time discretization $\{t_0, \ldots, t_J\}$
- training input $u(t)$

Compute mean empirical cross Gramian [*Himpe/Ohlberger 14*]

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

with $X(p_j) := [x(t_0; p_j), \ldots, x(t_J; p_j)]$, adjoint snapshots $\bar{X}(p_j)$.

Obtain projection V by SVD of the empirical cross gramian:

$$\overline{W}_X \overset{\text{SVD}}{=} V_1 S V_2^T \text{ and } V = V_1[:, 1:r].$$

Performance measure

1. Time-domain errors ($y_j := y(\cdot\,;\,p_j)$, $x_j := x(\cdot\,;\,p_j)$):

$$
\begin{aligned}
\|x_j - V\hat{x}_j\|^2_{\mathcal{L}_2([0,\,T])} &\approx \Delta t \sum_{i=0}^{J} \|(x_j(t_i) - V\hat{x}_j(t_i))\|_2^2 \\
\|y_j - \hat{y}_j\|^2_{\mathcal{L}_2([0,\,T])} &\approx \Delta t \sum_{i=0}^{J} \|(y_j(t_i) - \hat{y}_j(t_i))\|_2^2 \\
\|y_j - \hat{y}_j\|_{\mathcal{L}_\infty([0,\,T])} &\approx \max_{i=0,\dots,J} \|y_j(t_i) - \hat{y}_j(t_i)\|_2
\end{aligned}
$$

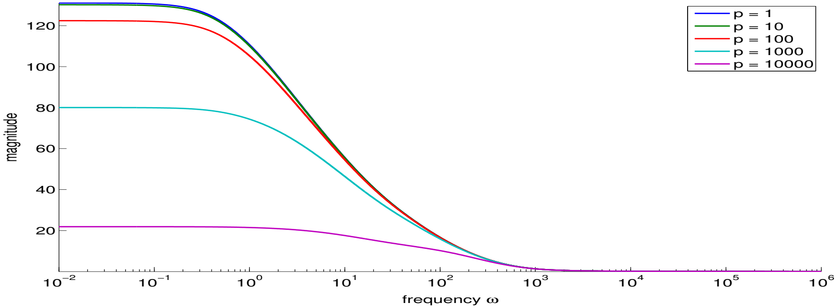
2. Errors in frequency domain ($G_j := G(\cdot\,, p_j)$):

$$
\begin{aligned}
\|G_j - \hat{G}_j\|_{\mathcal{H}_\infty} &\approx \max_{1\le i\le L} \bar{\sigma}(G_j(\imath\omega_i) - \hat{G}_j(\imath\omega_i)) \\
\|G_j - \hat{G}_j\|^2_{\mathcal{H}_2} &= \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{tr}\left[(G_j(\imath\omega) - \hat{G}_j(\imath\omega))^*(G_j(\imath\omega) - \hat{G}_j(\imath\omega))\right] \\
&= \mathrm{tr}\left[[C_j,\ \hat{C}_j]P\,[C_j,\ \hat{C}_j]^T\right]
\end{aligned}
$$

3. Offline/Online time

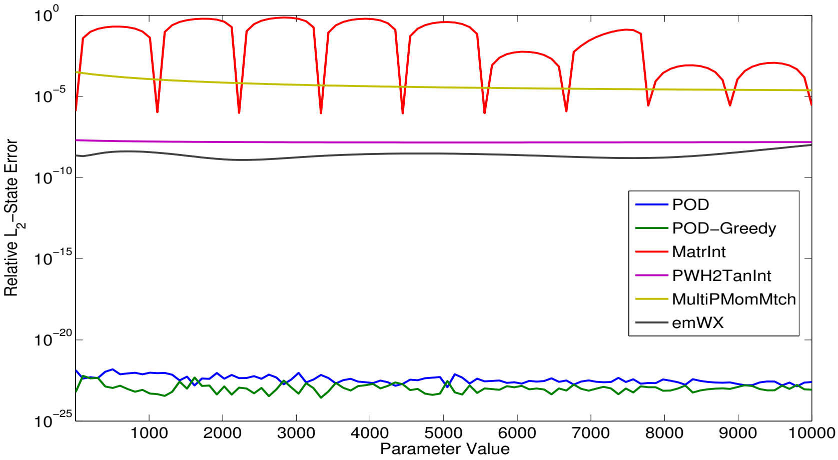
Benchmark microthruster

- thermal problem modeled as homogenous heat diffusion
- heat exchange modeled with convection boundary conditions
- film coefficient at top interface varies in $[1, 10^4]$

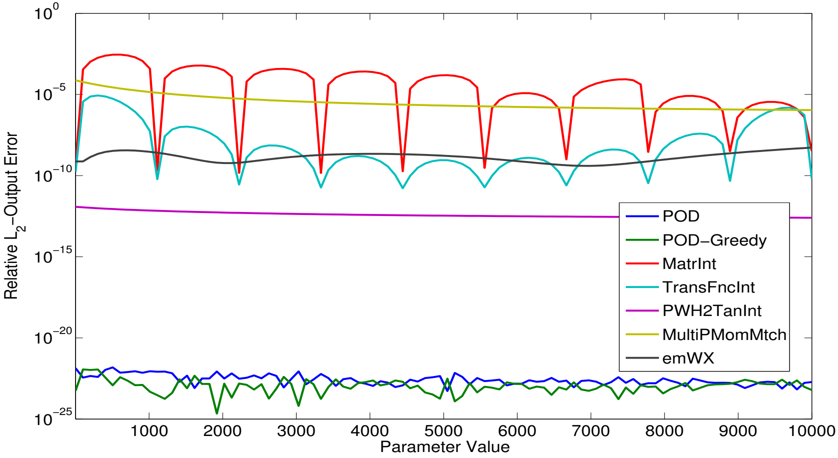


Benchmark from http://www.modelreduction.org.

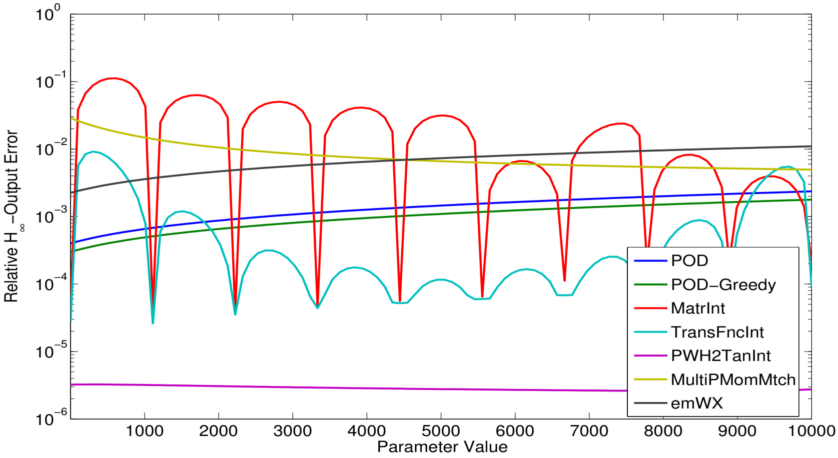
Numerical results - $\mathcal{L}_2$-state error



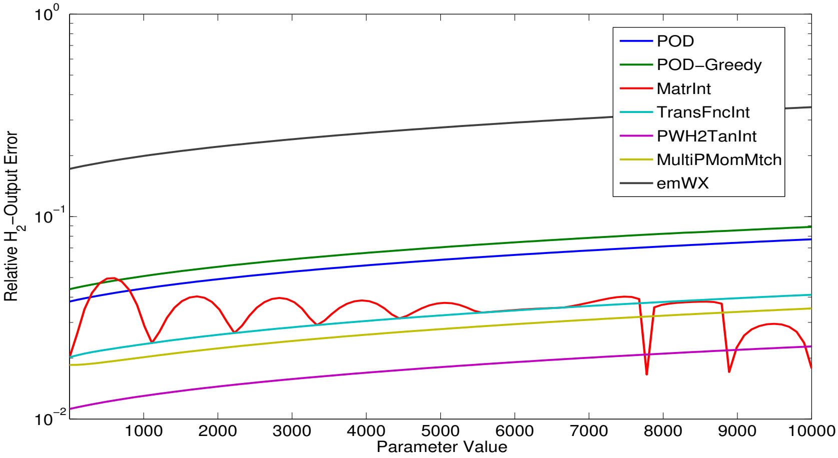
Numerical results - $\mathcal{L}_2$-output error



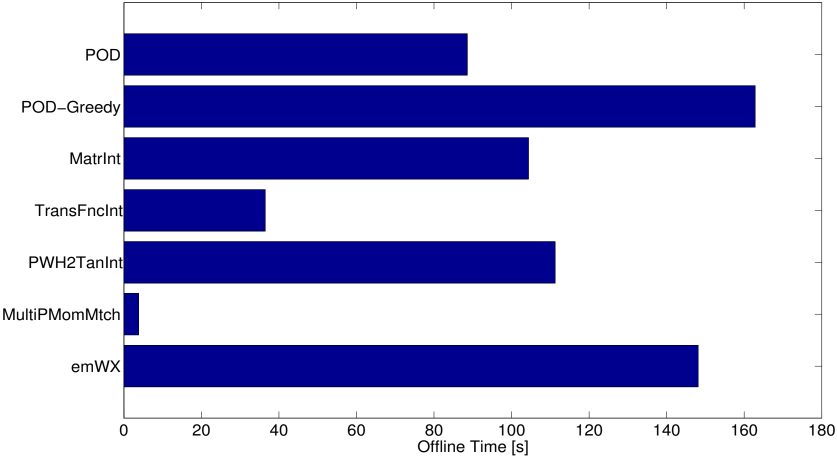
Numerical results - $\mathcal{H}_\infty$ error



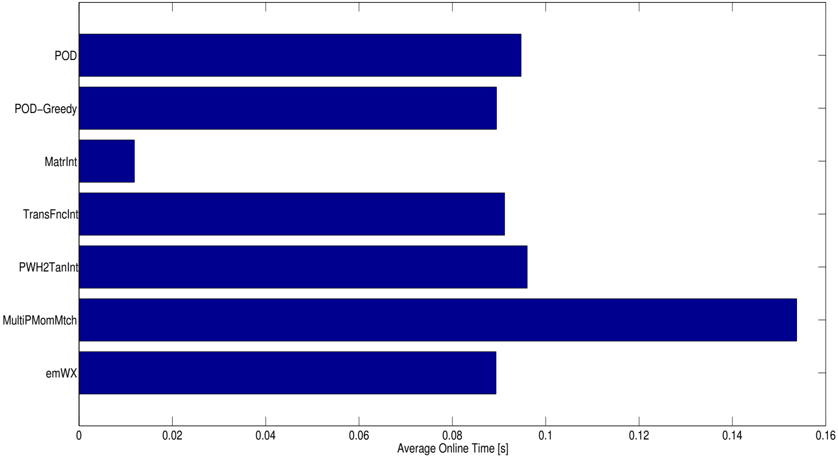
Numerical results - $\mathcal{H}_2$ error



Numerical results - Offline time



Numerical results - Online time



Conclusions

1. POD/POD-Greedy very good for state-space approximations
2. PWH2TanInt very good in frequency domain
3. emWX constant good behavior for most norms and all benchmarks
4. TransFncInt performs well for mostly all error measures, problems by peaks
5. MultiPMomMtch most efficient w.r.t. offline time but requires good tuning

Conclusions

1. POD/POD-Greedy very good for state-space approximations
2. PWH2TanInt very good in frequency domain
3. emWX constant good behavior for most norms and all benchmarks
4. TransFncInt performs well for mostly all error measures, problems by peaks
5. MultiPMomMtch most efficient w.r.t. offline time but requires good tuning

- For details/more results, see preprint

Conclusions

1. POD/POD-Greedy very good for state-space approximations
2. PWH2TanInt very good in frequency domain
3. emWX constant good behavior for most norms and all benchmarks
4. TransFncInt performs well for mostly all error measures, problems by peaks
5. MultiPMomMtch most efficient w.r.t. offline time but requires good tuning

- For details/more results, see preprint
- Outlook $\Rightarrow$ Christians talk!

Conclusions

1. POD/POD-Greedy very good for state-space approximations
2. PWH2TanInt very good in frequency domain
3. emWX constant good behavior for most norms and all benchmarks
4. TransFncInt performs well for mostly all error measures, problems by peaks
5. MultiPMomMtch most efficient w.r.t. offline time but requires good tuning

- For details/more results, see preprint
- Outlook $\Rightarrow$ Christians talk!
- Thank you very much for your attention!