Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

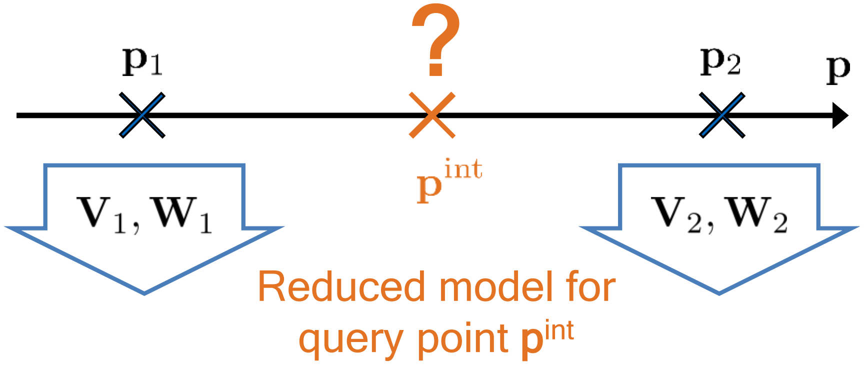
Parametric Model Order Reduction: An Introduction

Maria Cruz Varona

Chair of Automatic Control

Department of Mechanical Engineering

Technical University of Munich



Model Order Reduction Summer School

September 24th 2019

Linear Model Order Reduction

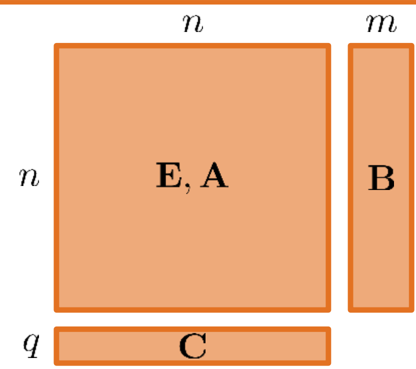
Projective Non-Parametric MOR

Linear time-invariant (LTI) system

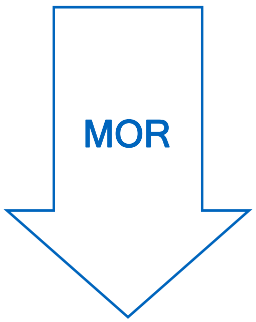
$$
\mathbf{G}(s):\begin{cases}\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)\\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)\end{cases}
$$

$\mathbf{E},\ \mathbf{A} \in \mathbb{R}^{n\times n}$

$\mathbf{B} \in \mathbb{R}^{n\times m},\ \mathbf{C} \in \mathbb{R}^{q\times n}$



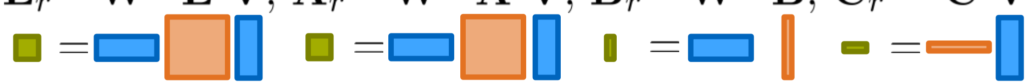
$r \ll n$



Projection

$\mathbf{V},\ \mathbf{W} \in \mathbb{R}^{n\times r}$

$\mathbf{E}_r = \mathbf{W}^T\,\mathbf{E}\,\mathbf{V},\ \mathbf{A}_r = \mathbf{W}^T\,\mathbf{A}\,\mathbf{V},\ \mathbf{B}_r = \mathbf{W}^T\mathbf{B},\ \mathbf{C}_r = \mathbf{C}\,\mathbf{V}$

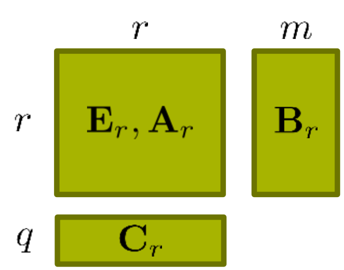


Reduced order model (ROM)

$$
\mathbf{G}_r(s):\begin{cases}\mathbf{E}_r\dot{\mathbf{x}}_r(t) = \mathbf{A}_r\mathbf{x}_r(t) + \mathbf{B}_r\mathbf{u}(t)\\ \mathbf{y}_r(t) = \mathbf{C}_r\mathbf{x}_r(t)\end{cases}
$$

$\mathbf{E}_r,\ \mathbf{A}_r \in \mathbb{R}^{r\times r}$

$\mathbf{B}_r \in \mathbb{R}^{r\times m},\ \mathbf{C}_r \in \mathbb{R}^{q\times r}$



Linear MOR methods – Overview

1. Modal Reduction (modalMOR)
   - Preservation of dominant eigenmodes
   - Frequently used in structural dynamics / second order systems

2. Truncated Balanced Realization (TBR) / Balanced Truncation (BT)
   - Retention of state-space directions with highest energy transfer
   - Requires solution of Lyapunov equations, i.e. linear matrix equations (LMEs)
   - Applicable for medium-scale models: $n \approx 5000$

3. Rational Krylov subspaces (RK)
   - “Moment Matching”: matching some Taylor-series coefficients of the transfer function
   - Requires solution of linear systems of equations (LSEs) – applicable for $n \approx 10^6$
   - Also employed for: approximate solution of eigenvalue problems, LSEs, LMEs,…

4. Iterative Rational Krylov algorithm (IRKA)
   - H2-optimal reduction
   - Adaptive choice of Krylov reduction parameters (e.g. shifts)

Truncated Balanced Realization (TBR)

Goal: Preserve state-space directions with highest enery transfer

Controllability and Observability Gramians:

$$P = \int_0^\infty e^{E^{-1}At} E^{-1} B B^\top E^{-\top} e^{A^\top E^{-\top} t} \, dt$$

$$Q = \int_0^\infty e^{A^\top E^{-\top} t} C^\top C e^{E^{-1}At} \, dt$$

Energy interpretation:

$$\min_{x(0)=0,\ x(\infty)=x_e} \int_0^\infty |u(t)|^2 \, dt = x_e^\top P^{-1} x_e$$

$$\|y(t)\|_2^2 = x_0^\top Q \, x_0$$

Lyapunov equations: $A P E^\top + E P A^\top + B B^\top = 0, \qquad A^\top Q E + E^\top Q A + C^\top C = 0$

Procedure:

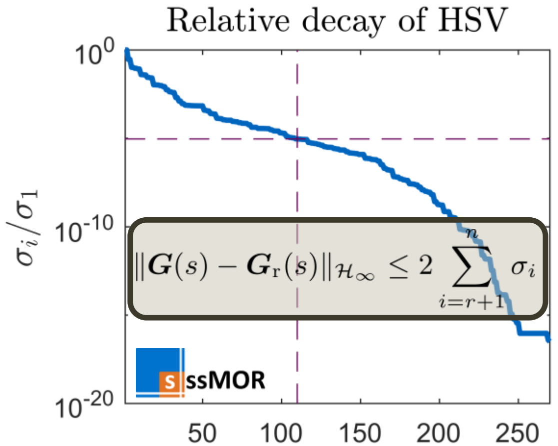
1 *Balancing step:* Compute balanced realization, where $P = E^\top Q E = \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$

$$P = RR^\top, \quad Q = SS^\top$$

$$S^\top E R = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} N_1^\top \\ N_2^\top \end{bmatrix}$$

2 *Truncation step:* $\sigma_i \gg \sigma_j, \; i = 1, \ldots, r, \; j = r+1, \ldots, n$

$$W^\top = \Sigma_1^{-1/2} U_1^\top S^\top, \qquad V = R \, N_1 \, \Sigma_1^{-1/2}$$



Rational Interpolation by Krylov subspaces

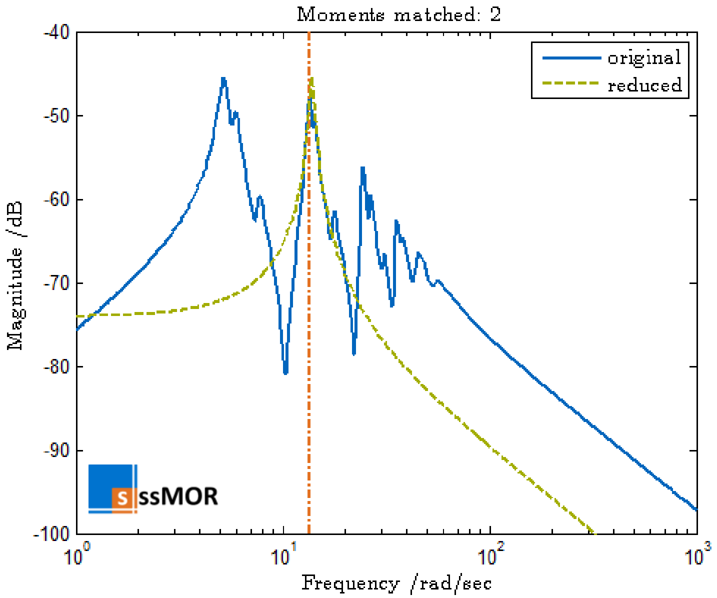
Moments of a transfer function

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

$$= G(\Delta s + \sigma) = \sum_{i=0}^{\infty} M_i(\sigma)\,(s-\sigma)^i$$

σ : interpolation point (shift)

$M_i(\sigma)$: i-th moment around σ



(Multi)-Moment Matching by Rational Krylov (RK) subspaces

Bases for input and output Krylov-subspaces:

$$\mathrm{Ran}(V) \supseteq \mathrm{span}\left\{A_\sigma^{-1}B,\ A_\sigma^{-1}EA_\sigma^{-1}B,\ \ldots,\ (A_\sigma^{-1}E)^{r-1}A_\sigma^{-1}B\right\}$$

$$\mathrm{Ran}(W) \supseteq \mathrm{span}\left\{A_\sigma^{-\mathsf{T}}C^{\mathsf{T}},\ A_\sigma^{-\mathsf{T}}E^{\mathsf{T}}A_\sigma^{-\mathsf{T}}C^{\mathsf{T}},\ \ldots,\ (A_\sigma^{-\mathsf{T}}E^{\mathsf{T}})^{r-1}A_\sigma^{-\mathsf{T}}C^{\mathsf{T}}\right\}$$

$\Rightarrow$ $$M_i(\sigma) = M_{\mathrm{r},i}(\sigma)$$

for $i = 0, \ldots, 2r-1$

Moments from full and reduced order model around certain shifts **match!**

Comparison: BT vs. Krylov subspaces

Balanced Truncation (BT)

+ stability preservation
+ automatable
+ error bound (a priori)

– computing-intensive
– storage-intensive
– n < 5000

Rational Krylov (RK) subspaces

+ numerically efficient
+ $n \approx 10^6$
+ H_2-optimal (IRKA)
+ many degrees of freedom

– many degrees of freedom
– stability gen. not preserved
– no error bounds

Subject of research

- Numerically efficient solution of large-scale Lyapunov equations

⇨ Krylov-based Low-Rank Approximation
  - ADI (Alternating Directions Implicit)
  - RKSM (Rational Krylov Subspace Method)

Subject of research

- Adaptive choice of reduction parameters
  - Reduced order
  - Interpolation data (shifts, etc.)
- Stability preservation
- Numerically efficient computation of rigorous error bounds

Parametric Model Order Reduction

Parametric Model Order Reduction (pMOR)

Large-scale parametric model

$$\mathbf{E}(\mathbf{p})\,\dot{\mathbf{x}} = \mathbf{A}(\mathbf{p})\,\mathbf{x} + \mathbf{B}(\mathbf{p})\,\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}(\mathbf{p})\,\mathbf{x}$$

$$\mathbf{p} \in \mathcal{D} \subset \mathbb{R}^d$$

pMOR →

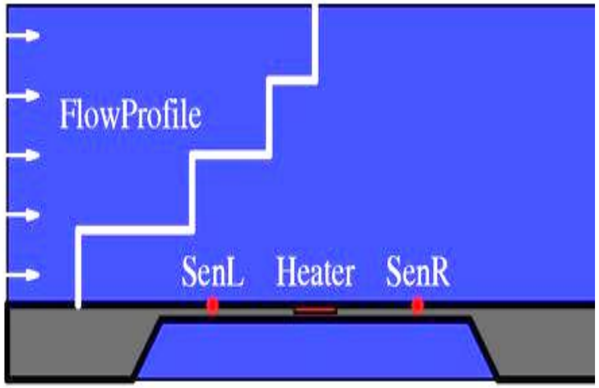
Reduced order parametric model

$$\mathbf{E}_r(\mathbf{p})\,\dot{\mathbf{x}}_r = \mathbf{A}_r(\mathbf{p})\,\mathbf{x}_r + \mathbf{B}_r(\mathbf{p})\,\mathbf{u}$$
$$\mathbf{y}_r = \mathbf{C}_r(\mathbf{p})\,\mathbf{x}_r$$

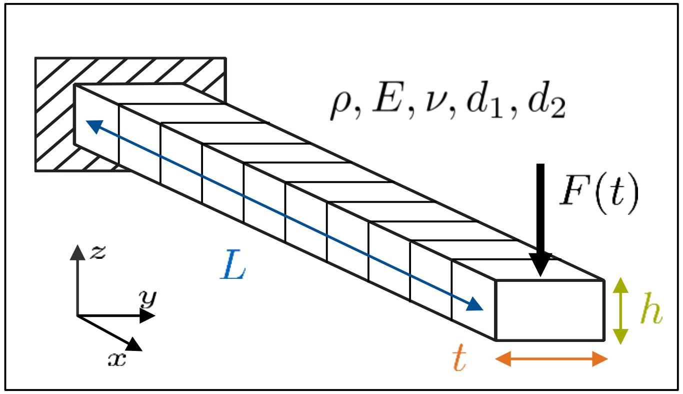
$$\mathbf{x}_r \in \mathbb{R}^r,\ r \ll n$$

- Linear dynamic systems with design parameters (e.g. material / geometry parameters,…)
- **Goal:** numerically efficient reduction with **preservation of the parameter dependency**

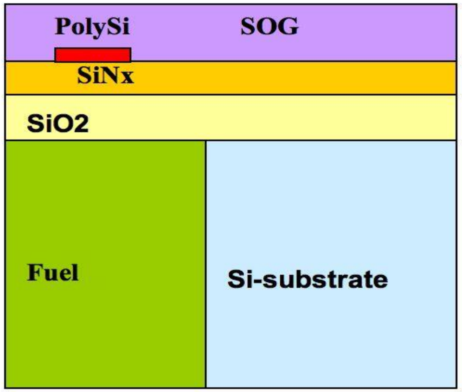
  → variation of the parameters in the ROM without having to repeat the reduction every time!



Flow sensing anemometer



Timoshenko beam



Microthruster unit

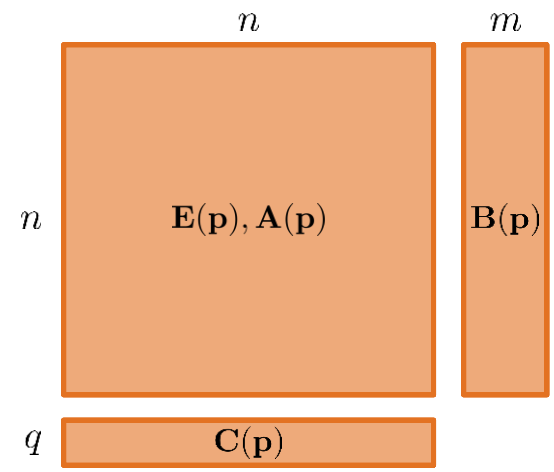
Projective Parametric MOR

Linear parametric system

$$\mathbf{G}(s,\mathbf{p}):\begin{cases}\mathbf{E}(\mathbf{p})\dot{\mathbf{x}}(t)=\mathbf{A}(\mathbf{p})\mathbf{x}(t)+\mathbf{B}(\mathbf{p})\mathbf{u}(t)\\ \mathbf{y}(t)=\mathbf{C}(\mathbf{p})\mathbf{x}(t)\end{cases}$$

$$\mathbf{x}(t)\in\mathbb{R}^n,\mathbf{u}(t)\in\mathbb{R}^m,\mathbf{y}(t)\in\mathbb{R}^q,\mathbf{p}\in\mathcal{D}\subset\mathbb{R}^d$$

High computational effort and storage requirement needed for simulation, optimization and control



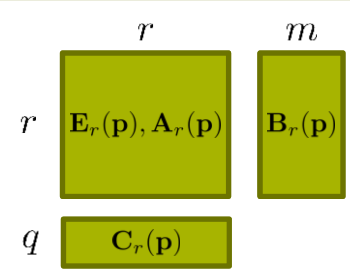
$r\ll n$

Reduced parametric system

$$\mathbf{G}_r(s,\mathbf{p}):\begin{cases}\mathbf{E}_r(\mathbf{p})\dot{\mathbf{x}}_r(t)=\mathbf{A}_r(\mathbf{p})\mathbf{x}_r(t)+\mathbf{B}_r(\mathbf{p})\mathbf{u}(t)\\ \mathbf{y}_r(t)=\mathbf{C}_r(\mathbf{p})\mathbf{x}_r(t)\end{cases}$$

$$\mathbf{E}_r(\mathbf{p})=\mathbf{W}(\mathbf{p})^T\mathbf{E}(\mathbf{p})\mathbf{V}(\mathbf{p}),\qquad \mathbf{A}_r(\mathbf{p})=\mathbf{W}(\mathbf{p})^T\mathbf{A}(\mathbf{p})\mathbf{V}(\mathbf{p})$$

$$\mathbf{B}_r(\mathbf{p})=\mathbf{W}(\mathbf{p})^T\mathbf{B}(\mathbf{p}),\qquad \mathbf{C}_r(\mathbf{p})=\mathbf{C}(\mathbf{p})\mathbf{V}(\mathbf{p})$$



Overview pMOR approaches

Global approaches

Common subspaces $\mathbf{V}(\mathbf{p}), \mathbf{W}(\mathbf{p})$
for all $\mathbf{p} \in \mathcal{D} \subset \mathbb{R}^d$

Multi-Parameter Moment Matching
[Weile '99, Daniel '04]

- ✚ Moment Matching w.r.t. s and $\mathbf{p}$
- ▬ Explicit parameter dependency requi.
- ▬ Curse of dimensionality

Concatenation of local bases
[Leung '05, Li '05, Baur et al. '11]

- ✚ Computation of $\mathbf{V}_1, \mathbf{W}_1, \ldots, \mathbf{V}_K, \mathbf{W}_K$ using BT, RK, IRKA or POD
- ✚ Concatenation of the local bases
  $\mathbf{V}(\mathbf{p}) = [\mathbf{V}_1, \ldots, \mathbf{V}_K], \mathbf{W}(\mathbf{p}) = [\mathbf{W}_1, \ldots, \mathbf{W}_K]$
- ▬ Reduced order: $r = K \cdot r'$
- ▬ Affine parameter dependency required

Local approaches

Individual subspaces $\mathbf{V}(\mathbf{p}_i), \mathbf{W}(\mathbf{p}_i)$
for local systems at $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_K$

Interpolation of transfer functions
[Baur '09]

- ✚ Local reduction using BT
  ⇨ Error bounds and stability
- ▬ Reduced order: $r = K \cdot r'$

Interpolation of subspaces
[Amsallem '08]

- ◎ Interpolation of the reduction bases
- ✚ Reduced order: $r = r'$

Interpolation of reduced matrices
[Eid '09, Panzer '10, Amsallem '11]

- ✚ No explicit or affine parameter dependency required
- ✚ Reduced order: $r = r'$

Concatenation of bases

(Weighted) concatenation of bases

- (Weighted) concatenation of local bases: $\tilde{\mathbf{V}}(\mathbf{p}^{\text{int}}) = \left[w_1(\mathbf{p}^{\text{int}})\,\mathbf{V}(\mathbf{p}_1), \ldots, w_K(\mathbf{p}^{\text{int}})\,\mathbf{V}(\mathbf{p}_K)\right] \in \mathbb{R}^{n \times K \cdot r}$
- Singular Value Decomposition: $\tilde{\mathbf{V}}(\mathbf{p}^{\text{int}}) \overset{\text{SVD}}{=} \tilde{\mathbf{U}}(\mathbf{p}^{\text{int}})\,\tilde{\mathbf{\Sigma}}(\mathbf{p}^{\text{int}})\,\tilde{\mathbf{T}}(\mathbf{p}^{\text{int}})^T$
- Reduction basis for new query point: $\mathbf{V}(\mathbf{p}^{\text{int}}) = \left[\tilde{\mathbf{u}}_1(\mathbf{p}^{\text{int}}), \ldots, \tilde{\mathbf{u}}_r(\mathbf{p}^{\text{int}})\right] \in \mathbb{R}^{n \times r}$

(Weighted) concatenation of snapshots

Concatenation of snapshots is suitable, when the local bases $\mathbf{V}(\mathbf{p}_i) = \mathbf{U}(\mathbf{p}_i)(:, 1:r), \quad i = 1, \ldots, K$ are calculated via an SVD-based technique like POD: $\mathbf{X}(\mathbf{p}_i) = \mathbf{U}(\mathbf{p}_i)\,\mathbf{\Sigma}(\mathbf{p}_i)\,\mathbf{T}(\mathbf{p}_i)^T$.

- (Weighted) concatenation of snapshots: $\tilde{\mathbf{X}}(\mathbf{p}^{\text{int}}) = \left[w_1(\mathbf{p}^{\text{int}})\,\mathbf{X}(\mathbf{p}_1), \ldots, w_K(\mathbf{p}^{\text{int}})\,\mathbf{X}(\mathbf{p}_K)\right] \in \mathbb{R}^{n \times K \cdot n_s}$
- Singular Value Decomposition: $\tilde{\mathbf{X}}(\mathbf{p}^{\text{int}}) \overset{\text{SVD}}{=} \tilde{\mathbf{U}}_{\tilde{\mathbf{X}}}(\mathbf{p}^{\text{int}})\,\tilde{\mathbf{\Sigma}}_{\tilde{\mathbf{X}}}(\mathbf{p}^{\text{int}})\,\tilde{\mathbf{T}}_{\tilde{\mathbf{X}}}(\mathbf{p}^{\text{int}})^T$
- Reduction basis for new query point: $\mathbf{V}(\mathbf{p}^{\text{int}}) = \tilde{\mathbf{U}}_{\tilde{\mathbf{X}}}(\mathbf{p}^{\text{int}})(:, 1:r) \in \mathbb{R}^{n \times r}$

Interpolation of subspaces

[Amsallem et al. '08]

Interpolation of local bases is suitable for linear and nonlinear MOR!

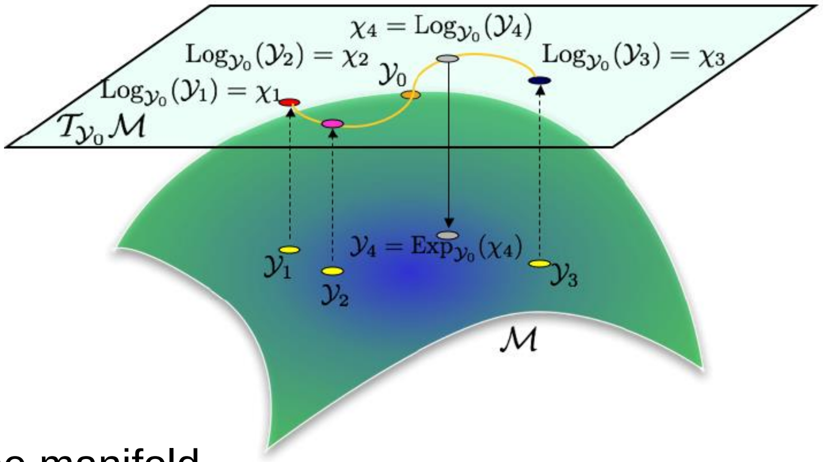
Starting point: Local bases $\{\mathbf{V}(\mathbf{p}_i)\}_{i=1}^{K}$ at parameter sample points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_K$.

Procedure:

1. Choose a local basis $\mathbf{V}_{i_0}$ for the reference subspace $\mathcal{V}_{i_0}$
2. Mapping of all subspaces $\mathcal{V}(\mathbf{p}_i)$ onto the tangent space $\mathcal{T}_{\mathcal{V}_{i_0}}$ using the logarithmic map:

$$\left(\mathbf{I} - \mathbf{V}_{i_0}\mathbf{V}_{i_0}^T\right)\mathbf{V}(\mathbf{p}_i)\left(\mathbf{V}_{i_0}^T\mathbf{V}(\mathbf{p}_i)\right)^{-1} = \mathbf{U}(\mathbf{p}_i)\,\mathbf{\Sigma}(\mathbf{p}_i)\,\mathbf{T}(\mathbf{p}_i)^T,$$

$$\mathbf{\Gamma}(\mathbf{p}_i) = \mathbf{U}(\mathbf{p}_i)\,\arctan\left(\mathbf{\Sigma}(\mathbf{p}_i)\right)\mathbf{T}(\mathbf{p}_i)^T$$



3. Interpolation in the tangent space:

$$\mathbf{\Gamma}(\mathbf{p}^{\mathrm{int}}) = \sum_{i=1}^{K} w_i(\mathbf{p}^{\mathrm{int}})\,\mathbf{\Gamma}(\mathbf{p}_i)$$

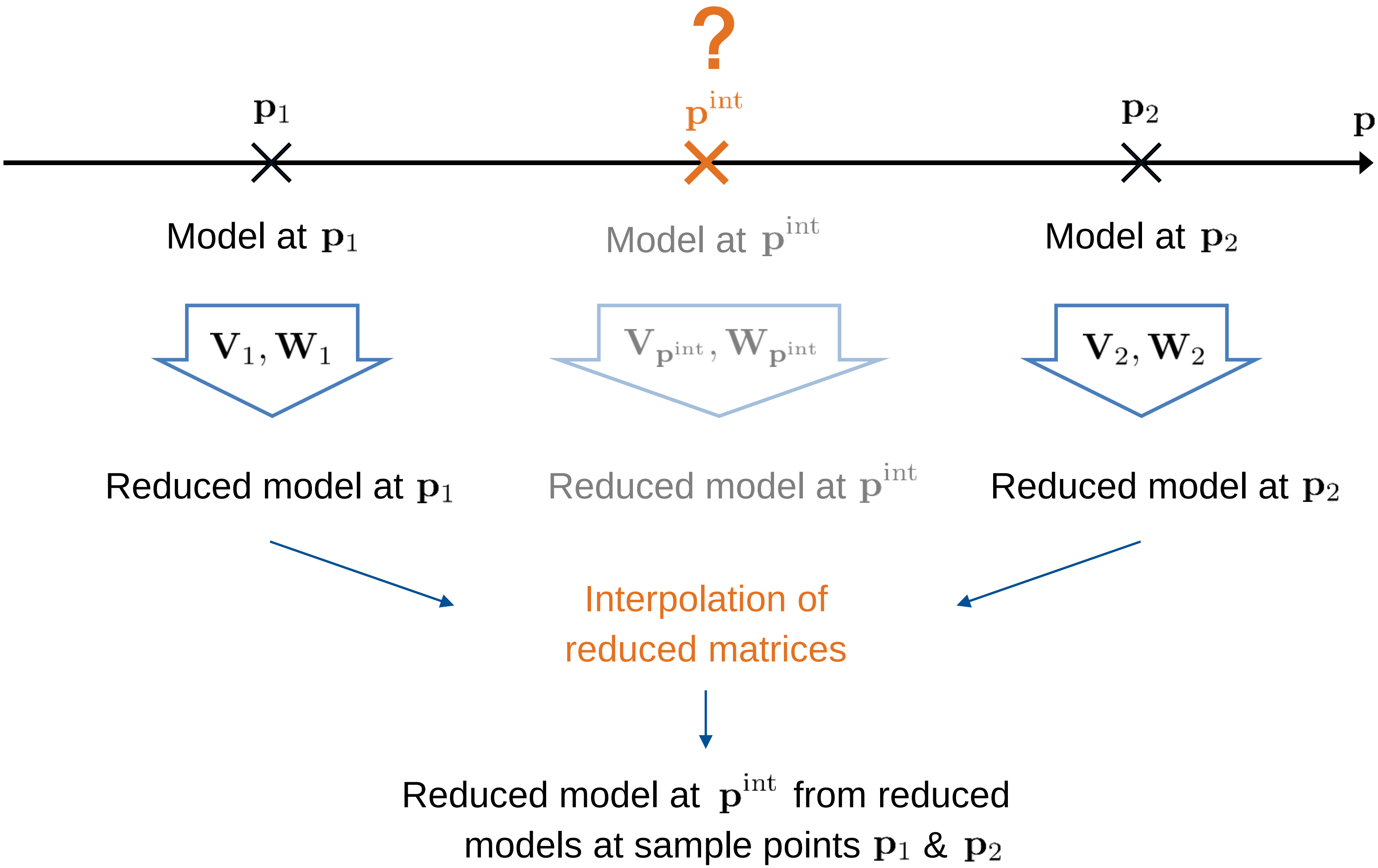
4. Backmapping of the interpolated subspace onto the manifold using the exponential map:

$$\mathbf{\Gamma}(\mathbf{p}^{\mathrm{int}}) \overset{\mathrm{SVD}}{=} \mathbf{U}(\mathbf{p}^{\mathrm{int}})\,\mathbf{\Sigma}(\mathbf{p}^{\mathrm{int}})\,\mathbf{T}(\mathbf{p}^{\mathrm{int}})^T,$$

$$\mathbf{V}(\mathbf{p}^{\mathrm{int}}) = \mathbf{V}_{i_0}\mathbf{T}(\mathbf{p}^{\mathrm{int}})\cos\left(\mathbf{\Sigma}(\mathbf{p}^{\mathrm{int}})\right) + \mathbf{U}(\mathbf{p}^{\mathrm{int}})\sin\left(\mathbf{\Sigma}(\mathbf{p}^{\mathrm{int}})\right)$$

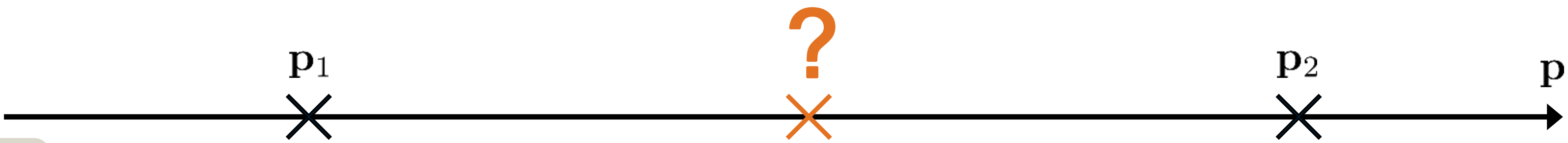
Interpolation of reduced matrices: pMOR by Matrix Interpolation

pMOR by Matrix Interpolation – Main Idea



pMOR by Matrix Interpolation – Procedure

[Panzer et al. ’10]



$\mathbf{V}_1$
$\mathbf{W}_1$

$$\mathbf{E}_{r,1}\dot{\mathbf{x}}_{r,1} = \mathbf{A}_{r,1}\mathbf{x}_{r,1} + \mathbf{B}_{r,1}\mathbf{u}$$
$$\mathbf{y}_{r,1} = \mathbf{C}_{r,1}\mathbf{x}_{r,1}$$

$\mathbf{p}^{\text{int}}$

direct interpolation not meaningful

$$\mathbf{E}_{r,2}\dot{\mathbf{x}}_{r,2} = \mathbf{A}_{r,2}\mathbf{x}_{r,2} + \mathbf{B}_{r,2}\mathbf{u}$$
$$\mathbf{y}_{r,2} = \mathbf{C}_{r,2}\mathbf{x}_{r,2}$$

$\mathbf{V}_2$
$\mathbf{W}_2$

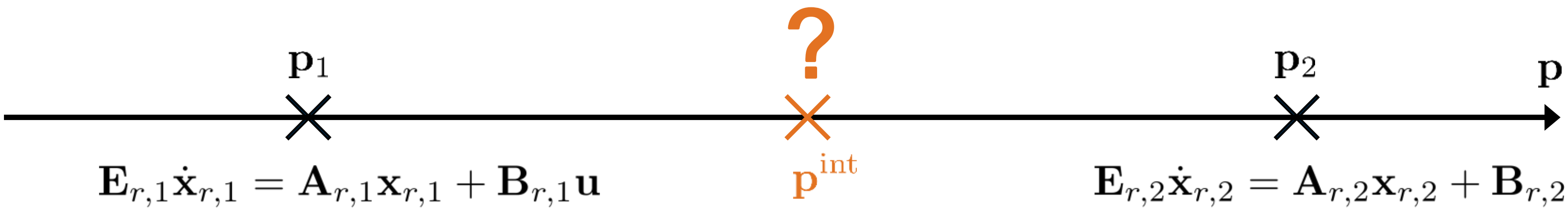
1.) Individual reduction

$$\mathbf{E}_{r,i}\dot{\mathbf{x}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t) \qquad \mathbf{E}_{r,i} = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i, \quad \mathbf{A}_{r,i} = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i$$
$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) \qquad \mathbf{B}_{r,i} = \mathbf{W}_i^T\mathbf{B}_i, \quad \mathbf{C}_{r,i} = \mathbf{C}_i\mathbf{V}_i$$

$\mathbf{p}_i, \quad i = 1, \ldots, K$
$\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i)$
$\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i)$

pMOR by Matrix Interpolation – Procedure

[Panzer et al. '10]



$$\mathbf{E}_{r,1}\dot{\mathbf{x}}_{r,1} = \mathbf{A}_{r,1}\mathbf{x}_{r,1} + \mathbf{B}_{r,1}\mathbf{u} \qquad \mathbf{E}_{r,2}\dot{\mathbf{x}}_{r,2} = \mathbf{A}_{r,2}\mathbf{x}_{r,2} + \mathbf{B}_{r,2}\mathbf{u}$$

$$\mathbf{y}_{r,1} = \mathbf{C}_{r,1}\mathbf{x}_{r,1} \qquad \mathbf{y}_{r,2} = \mathbf{C}_{r,2}\mathbf{x}_{r,2}$$

1.) Individual reduction

$$\mathbf{E}_{r,i}\dot{\mathbf{x}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t) \qquad \mathbf{E}_{r,i} = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i, \quad \mathbf{A}_{r,i} = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i$$

$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) \qquad \mathbf{B}_{r,i} = \mathbf{W}_i^T\mathbf{B}_i, \quad \mathbf{C}_{r,i} = \mathbf{C}_i\mathbf{V}_i$$

$$\mathbf{p}_i, \quad i = 1, \dots, K$$
$$\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i)$$
$$\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i)$$

2.) Transformation to generalized coordinates

$$\mathbf{T}_i = (\mathbf{R}_V^T\mathbf{V}_i)^{-1}$$

$$\mathbf{M}_i^T \cdot \Big| \quad \mathbf{E}_{r,i}\mathbf{T}_i\dot{\hat{\mathbf{x}}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{T}_i\hat{\mathbf{x}}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t)$$

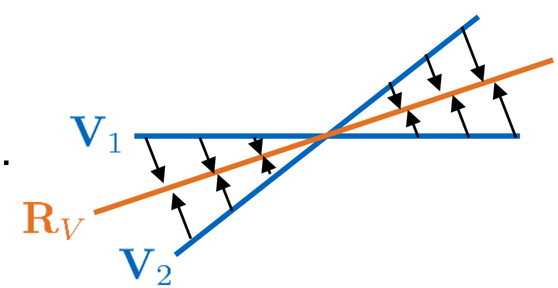
$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{T}_i\hat{\mathbf{x}}_{r,i}(t)$$

$$\mathbf{x}_{r,i} = \mathbf{T}_i\hat{\mathbf{x}}_{r,i}$$

$$\mathbf{V}_{\text{all}} = [\mathbf{V}_1, \dots, \mathbf{V}_K]$$
$$\mathbf{V}_{\text{all}} \overset{\text{SVD}}{=} \mathbf{U}\mathbf{S}\mathbf{N}^T$$
$$\mathbf{R}_V = \mathbf{U}(:, 1:r)$$

How do we choose $\mathbf{T}_i$?

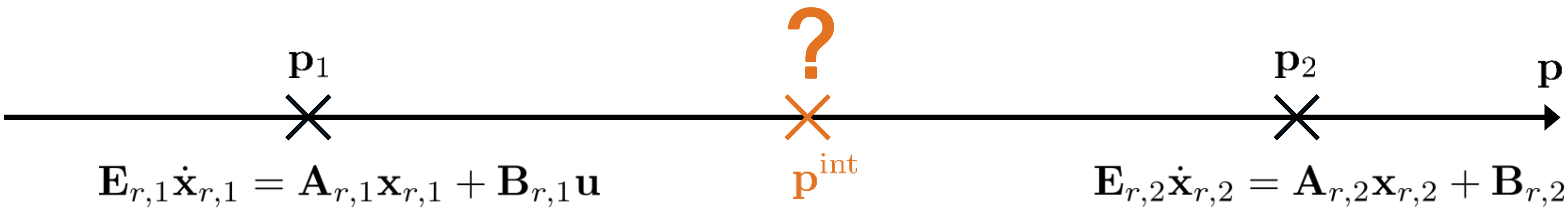
Goal: Adjustment of the local bases $\mathbf{V}_i$ to $\hat{\mathbf{V}}_i = \mathbf{V}_i\mathbf{T}_i$, in order to make the gen. coordinates $\hat{\mathbf{x}}_{r,i}$ compatible w.r.t. a reference subspace $\mathbf{R}_V$.



High correlation $\hat{\mathbf{V}}_i \leftrightarrow \mathbf{R}_V$:
$$\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{R}_V \overset{!}{=} \mathbf{I}$$

pMOR by Matrix Interpolation – Procedure

[Panzer et al. '10]



$$\mathbf{E}_{r,1}\dot{\mathbf{x}}_{r,1} = \mathbf{A}_{r,1}\mathbf{x}_{r,1} + \mathbf{B}_{r,1}\mathbf{u} \qquad \mathbf{E}_{r,2}\dot{\mathbf{x}}_{r,2} = \mathbf{A}_{r,2}\mathbf{x}_{r,2} + \mathbf{B}_{r,2}\mathbf{u}$$

$$\mathbf{y}_{r,1} = \mathbf{C}_{r,1}\mathbf{x}_{r,1} \qquad \mathbf{y}_{r,2} = \mathbf{C}_{r,2}\mathbf{x}_{r,2}$$

1.) Individual reduction

$$\mathbf{E}_{r,i}\dot{\mathbf{x}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t) \qquad \mathbf{E}_{r,i} = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i, \quad \mathbf{A}_{r,i} = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i$$

$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) \qquad \mathbf{B}_{r,i} = \mathbf{W}_i^T\mathbf{B}_i, \quad \mathbf{C}_{r,i} = \mathbf{C}_i\mathbf{V}_i$$

$\mathbf{p}_i, \quad i = 1, \ldots, K$

$\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i)$

$\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i)$

2.) Transformation to generalized coordinates

$\mathbf{M}_i^T \cdot \Big|$

$$\underbrace{\mathbf{M}_i^T\mathbf{E}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{E}}_{r,i}}\dot{\hat{\mathbf{x}}}_{r,i}(t) = \underbrace{\mathbf{M}_i^T\mathbf{A}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{A}}_{r,i}}\hat{\mathbf{x}}_{r,i}(t) + \underbrace{\mathbf{M}_i^T\mathbf{B}_{r,i}}_{\hat{\mathbf{B}}_{r,i}}\mathbf{u}(t)$$

$$\mathbf{y}_{r,i}(t) = \underbrace{\mathbf{C}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{C}}_{r,i}}\hat{\mathbf{x}}_{r,i}(t)$$

$\mathbf{T}_i = (\mathbf{R}_V^T\mathbf{V}_i)^{-1}$

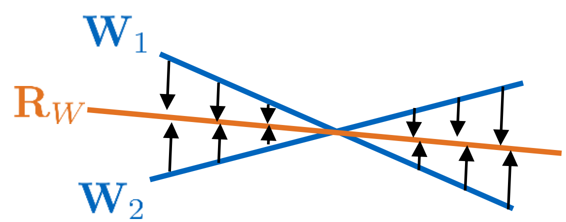
$\mathbf{M}_i = (\mathbf{R}_W^T\mathbf{W}_i)^{-1}$

$\mathbf{x}_{r,i} = \mathbf{T}_i\hat{\mathbf{x}}_{r,i}$

Analogous to $\mathbf{R}_V$ or $\mathbf{R}_W = \mathbf{R}_V := \mathbf{R}$

How do we choose $\mathbf{M}_i$?

Goal: Adjustment of the local bases $\mathbf{W}_i$ to $\hat{\mathbf{W}}_i = \mathbf{W}_i\mathbf{M}_i$, in order to describe the local reduced models w.r.t. the same reference basis $\mathbf{R}_W$.



High correlation $\hat{\mathbf{W}}_i \leftrightarrow \mathbf{R}_W$: $\mathbf{M}_i^T\mathbf{W}_i^T\mathbf{R}_W \overset{!}{=} \mathbf{I}$

pMOR by Matrix Interpolation – Procedure

[Panzer et al. '10]

$\mathbf{p}_1$ ... ? $\mathbf{p}^{\text{int}}$ ... $\mathbf{p}_2$ ... $\mathbf{p}$

$\mathbf{T}_1$, $\mathbf{M}_1$

$$\mathbf{E}_{r,1}\dot{\mathbf{x}}_{r,1} = \mathbf{A}_{r,1}\mathbf{x}_{r,1} + \mathbf{B}_{r,1}\mathbf{u}$$
$$\mathbf{y}_{r,1} = \mathbf{C}_{r,1}\mathbf{x}_{r,1}$$

$$\mathbf{E}_{r,2}\dot{\mathbf{x}}_{r,2} = \mathbf{A}_{r,2}\mathbf{x}_{r,2} + \mathbf{B}_{r,2}\mathbf{u}$$
$$\mathbf{y}_{r,2} = \mathbf{C}_{r,2}\mathbf{x}_{r,2}$$

$\mathbf{T}_2$, $\mathbf{M}_2$

1.) Individual reduction

$$\mathbf{E}_{r,i}\dot{\mathbf{x}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t) \qquad \mathbf{E}_{r,i} = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i, \quad \mathbf{A}_{r,i} = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i$$
$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) \qquad \mathbf{B}_{r,i} = \mathbf{W}_i^T\mathbf{B}_i, \quad \mathbf{C}_{r,i} = \mathbf{C}_i\mathbf{V}_i$$

$\mathbf{p}_i, \quad i = 1, \dots, K$

$\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i)$

$\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i)$

2.) Transformation to generalized coordinates

$\mathbf{M}_i^T \cdot \Big|$

$$\underbrace{\mathbf{M}_i^T\mathbf{E}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{E}}_{r,i}}\dot{\hat{\mathbf{x}}}_{r,i}(t) = \underbrace{\mathbf{M}_i^T\mathbf{A}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{A}}_{r,i}}\hat{\mathbf{x}}_{r,i}(t) + \underbrace{\mathbf{M}_i^T\mathbf{B}_{r,i}}_{\hat{\mathbf{B}}_{r,i}}\mathbf{u}(t)$$
$$\mathbf{y}_{r,i}(t) = \underbrace{\mathbf{C}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{C}}_{r,i}}\hat{\mathbf{x}}_{r,i}(t)$$

$$\mathbf{T}_i = (\mathbf{R}_V^T\mathbf{V}_i)^{-1}$$
$$\mathbf{M}_i = (\mathbf{R}_W^T\mathbf{W}_i)^{-1}$$
$$\mathbf{R}_W = \mathbf{R}_V := \mathbf{R}$$

$\mathbf{x}_{r,i} = \mathbf{T}_i\hat{\mathbf{x}}_{r,i}$

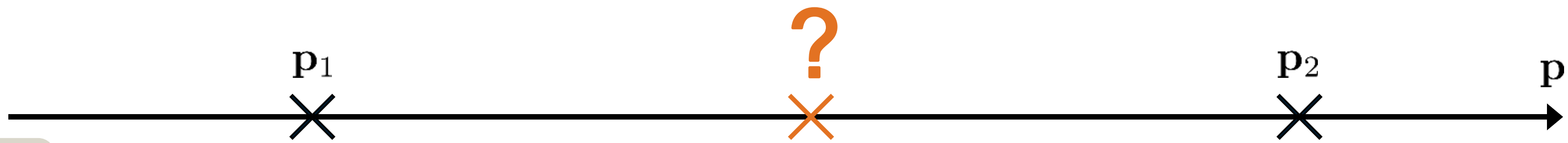
$\mathbf{V}_{\text{all}} = [\mathbf{V}_1, \dots, \mathbf{V}_K]$

$\mathbf{V}_{\text{all}} \overset{\text{SVD}}{=} \mathbf{U}\mathbf{S}\mathbf{N}^T$

$\mathbf{R}_V = \mathbf{U}(:, 1:r)$

pMOR by Matrix Interpolation – Procedure

[Panzer et al. ’10]



$\mathbf{T}_1$, $\mathbf{M}_1$

$$\hat{\mathbf{E}}_{r,1}\dot{\hat{\mathbf{x}}}_{r,1} = \hat{\mathbf{A}}_{r,1}\hat{\mathbf{x}}_{r,1} + \hat{\mathbf{B}}_{r,1}\mathbf{u}$$
$$\mathbf{y}_{r,1} = \hat{\mathbf{C}}_{r,1}\hat{\mathbf{x}}_{r,1}$$

$$\hat{\mathbf{E}}_{r,2}\dot{\hat{\mathbf{x}}}_{r,2} = \hat{\mathbf{A}}_{r,2}\hat{\mathbf{x}}_{r,2} + \hat{\mathbf{B}}_{r,2}\mathbf{u}$$
$$\mathbf{y}_{r,2} = \hat{\mathbf{C}}_{r,2}\hat{\mathbf{x}}_{r,2}$$

$\mathbf{T}_2$, $\mathbf{M}_2$

1.) Individual reduction

$$\mathbf{E}_{r,i}\dot{\mathbf{x}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t) \qquad \mathbf{E}_{r,i} = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i, \quad \mathbf{A}_{r,i} = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i$$
$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) \qquad \mathbf{B}_{r,i} = \mathbf{W}_i^T\mathbf{B}_i, \quad \mathbf{C}_{r,i} = \mathbf{C}_i\mathbf{V}_i$$

$$\mathbf{p}_i, \quad i = 1, \ldots, K$$
$$\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i)$$
$$\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i)$$

2.) Transformation to generalized coordinates

$$\mathbf{M}_i^T \cdot \Big|$$

$$\underbrace{\mathbf{M}_i^T\mathbf{E}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{E}}_{r,i}}\dot{\hat{\mathbf{x}}}_{r,i}(t) = \underbrace{\mathbf{M}_i^T\mathbf{A}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{A}}_{r,i}}\hat{\mathbf{x}}_{r,i}(t) + \underbrace{\mathbf{M}_i^T\mathbf{B}_{r,i}}_{\hat{\mathbf{B}}_{r,i}}\mathbf{u}(t)$$
$$\mathbf{y}_{r,i}(t) = \underbrace{\mathbf{C}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{C}}_{r,i}}\hat{\mathbf{x}}_{r,i}(t)$$

$$\mathbf{T}_i = (\mathbf{R}_V^T\mathbf{V}_i)^{-1}$$
$$\mathbf{M}_i = (\mathbf{R}_W^T\mathbf{W}_i)^{-1}$$
$$\mathbf{R}_W = \mathbf{R}_V := \mathbf{R}$$

$$\mathbf{x}_{r,i} = \mathbf{T}_i\hat{\mathbf{x}}_{r,i}$$

$$\mathbf{V}_{\text{all}} = [\mathbf{V}_1, \ldots, \mathbf{V}_K]$$
$$\mathbf{V}_{\text{all}} \overset{\text{SVD}}{=} \mathbf{U}\mathbf{S}\mathbf{N}^T$$
$$\mathbf{R}_V = \mathbf{U}(:, 1:r)$$

pMOR by Matrix Interpolation – Procedure

[Panzer et al. '10]

$\mathbf{p}_1$? $\mathbf{p}_2$ $\mathbf{p}$

$$\hat{\mathbf{E}}_{r,1}\dot{\hat{\mathbf{x}}}_{r,1} = \hat{\mathbf{A}}_{r,1}\hat{\mathbf{x}}_{r,1} + \hat{\mathbf{B}}_{r,1}\mathbf{u}$$
$$\mathbf{y}_{r,1} = \hat{\mathbf{C}}_{r,1}\hat{\mathbf{x}}_{r,1}$$

$\mathbf{p}^{\mathrm{int}}$

$$\hat{\mathbf{E}}_{r,2}\dot{\hat{\mathbf{x}}}_{r,2} = \hat{\mathbf{A}}_{r,2}\hat{\mathbf{x}}_{r,2} + \hat{\mathbf{B}}_{r,2}\mathbf{u}$$
$$\mathbf{y}_{r,2} = \hat{\mathbf{C}}_{r,2}\hat{\mathbf{x}}_{r,2}$$

1.) Individual reduction

$$\mathbf{E}_{r,i}\dot{\mathbf{x}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t) \qquad \mathbf{E}_{r,i} = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i, \quad \mathbf{A}_{r,i} = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i$$
$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) \qquad \mathbf{B}_{r,i} = \mathbf{W}_i^T\mathbf{B}_i, \quad \mathbf{C}_{r,i} = \mathbf{C}_i\mathbf{V}_i$$

$$\mathbf{p}_i, \quad i = 1,\dots,K$$
$$\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i)$$
$$\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i)$$

2.) Transformation to generalized coordinates

$\mathbf{M}_i^T \cdot \Big|$

$$\underbrace{\mathbf{M}_i^T\mathbf{E}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{E}}_{r,i}}\dot{\hat{\mathbf{x}}}_{r,i}(t) = \underbrace{\mathbf{M}_i^T\mathbf{A}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{A}}_{r,i}}\hat{\mathbf{x}}_{r,i}(t) + \underbrace{\mathbf{M}_i^T\mathbf{B}_{r,i}}_{\hat{\mathbf{B}}_{r,i}}\mathbf{u}(t)$$
$$\mathbf{y}_{r,i}(t) = \underbrace{\mathbf{C}_{r,i}\mathbf{T}_i}_{\hat{\mathbf{C}}_{r,i}}\hat{\mathbf{x}}_{r,i}(t)$$

$$\mathbf{T}_i = (\mathbf{R}_V^T\mathbf{V}_i)^{-1}$$
$$\mathbf{M}_i = (\mathbf{R}_W^T\mathbf{W}_i)^{-1}$$
$$\mathbf{R}_W = \mathbf{R}_V := \mathbf{R}$$

$$\mathbf{x}_{r,i} = \mathbf{T}_i\hat{\mathbf{x}}_{r,i}$$

$$\mathbf{V}_{\mathrm{all}} = [\mathbf{V}_1, \dots, \mathbf{V}_K]$$
$$\mathbf{V}_{\mathrm{all}} \overset{\mathrm{SVD}}{=} \mathbf{U}\mathbf{S}\mathbf{N}^T$$
$$\mathbf{R}_V = \mathbf{U}(:, 1:r)$$

3.) Interpolation

$$\hat{\mathbf{E}}_r(\mathbf{p}^{\mathrm{int}}) = \sum_{i=1}^K \omega_i(\mathbf{p}^{\mathrm{int}})\hat{\mathbf{E}}_{r,i}, \quad \hat{\mathbf{A}}_r(\mathbf{p}^{\mathrm{int}}) = \sum_{i=1}^K \omega_i(\mathbf{p}^{\mathrm{int}})\hat{\mathbf{A}}_{r,i}$$
$$\hat{\mathbf{B}}_r(\mathbf{p}^{\mathrm{int}}) = \sum_{i=1}^K \omega_i(\mathbf{p}^{\mathrm{int}})\hat{\mathbf{B}}_{r,i}, \quad \hat{\mathbf{C}}_r(\mathbf{p}^{\mathrm{int}}) = \sum_{i=1}^K \omega_i(\mathbf{p}^{\mathrm{int}})\hat{\mathbf{C}}_{r,i}$$

$$\sum_{i=1}^K \omega_i(\mathbf{p}^{\mathrm{int}}) = 1$$

Offline/Online decomposition

Offline phase:

1. Choose appropriate sample points $\mathbf{p}_i, \; i = 1, \ldots, K$ in the parameter space
2. Build local models at the parameter sample points
3. Reduce the local models separately with desired MOR technique (e.g. modalMOR, BT, rational Krylov, IRKA, …)
4. Compute $\mathbf{R}$, all transformation matrices $\mathbf{T}_i, \; \mathbf{M}_i$ and transform the local reduced models to generalized coordinates (step 4. in online phase, if weighted SVD)

Online phase:

1. Calculate the weights $\omega_i(\mathbf{p}^{\text{int}}), \; i = 1, \ldots, K$ depending on the actual parameter value $\mathbf{p}^{\text{int}}$ and the chosen interpolation method (linear, spline,…)
2. Interpolate between the reduced system matrices

Sampling of the parameter space

Interpolation method (weighting functions)

Sampling methods – Overview

[Benner et al. '15]

Choice of parameter sample points is a critical question, specially in high-dimensional spaces!

Small number of parameters (d < 3)

- Full grid-based sampling or Latin hypercube sampling: Structured/uniform sampling, random sampling, logarithmic sampling
- Moderate/high number of samples generated that covers the parameter space

Moderate number of parameters (3 ≤ d ≤ 10)

- In this case, full grid sampling quickly becomes expensive (curse of dimensionality)
- Latin hypercube sampling remains tractable
- Non-uniform sampling, sparse grid sampling

Large number of parameters (d > 10)

- Difficult to balance: number of sample points vs. coverage of the parameter space
- Problem-aware, adaptive sampling schemes required!
- Adaptive greedy search, sensitivity analysis using Taylor series, subspace angles, etc…

Adaptive Sampling

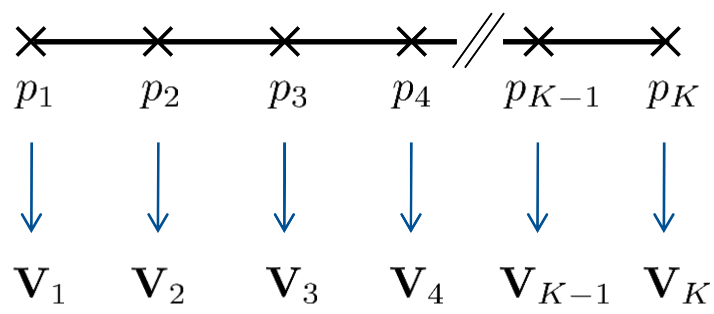
Requirements:

- Parametric space should be adequately sampled
- Avoid undersampling and oversampling
- More parameter samples should be placed in highly sensitive zones

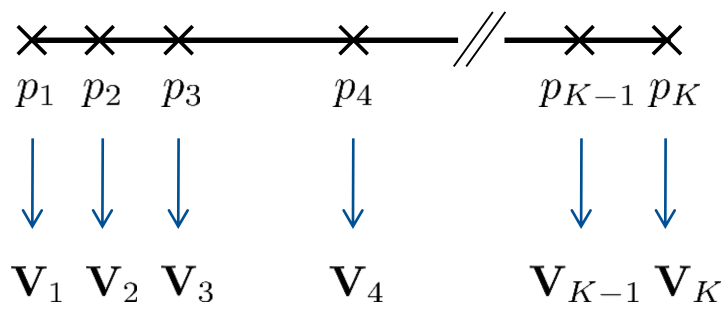
Quantification of parametric sensitivity:

- System-theoretic measure that quantifies the parametric sensitivity is needed in order to guide the adaptive refinement
- Adaptive sampling using angle between subspaces

Uniform Sampling:

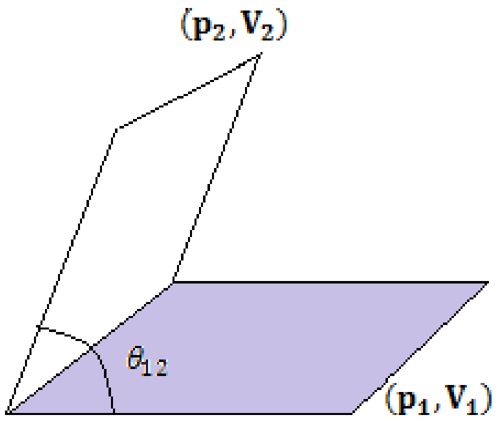


Adaptive Sampling:



Adaptive Sampling via subspace angles

Concept of subspace angles:



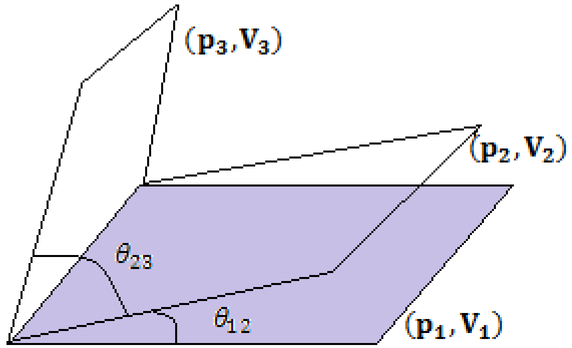
- $\mathbf{V}_1$ and $\mathbf{V}_2$ are orthonormal bases for the subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$
- The largest angle between the subspaces can be determined by

$$\theta_{12} = \arcsin\left(\sqrt{1-\sigma_r^2}\right) = \arccos(\sigma_r)$$

σ_r : smallest singular value of $\mathbf{V}_1^T\mathbf{V}_2$

Usage for adaptive grid refinement:

- The larger the subspace angle, the more different are the projection matrices, and thus:
  - the higher the parametric sensitivity
  - and the more sample points can be introduced in the respective sub-span



Automatic Adaptive Sampling: Pseudo-Code

[Cruz et al. '17]

1) Input $\theta_{\max}$
2) Divide the entire parameter range into a uniform grid, calling it $p_1, p_2, \cdots, p_K$
3) **While** all $l_{i,i+1} > 1$ **do**
   a) Calculate the projection matrices $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_K$ corresponding to each of these values $p_1, p_2, \cdots, p_K$
   b) Compute subspace angles $\theta_{12}, \theta_{23}, \cdots, \theta_{K-1,K}$ between these $\mathbf{V}_i$´s, each taken pairwise
   c) Calculate

$$l_{12} = \left\lceil \frac{\theta_{12}}{\theta_{\max}} \right\rceil, l_{23} = \left\lceil \frac{\theta_{23}}{\theta_{\max}} \right\rceil, \cdots, l_{K-1,K} = \left\lceil \frac{\theta_{K-1,K}}{\theta_{\max}} \right\rceil$$

   d) Divide the interval between p_1 and p_2 into l_{12} further intervals. Likewise, do the same for all the other intervals.
   e) Obtain new grid points $p_1, p_2, \cdots, p_N$, whereas $N > K$

   End While

Local reduction at sample points possible using any preferred MOR technique

```
theta(i) = subspace(Vp{i},Vp{i+1})
```

Quantitative indicator of how many pieces each parameter interval is to be further broken

Stopping criterion:
1. All ratios are equal to 1
2. Specified maximum number of samples points reached

Next iteration: local reduction, etc. <u>only</u> at points that got added in the last while-loop iteration (efficient!)

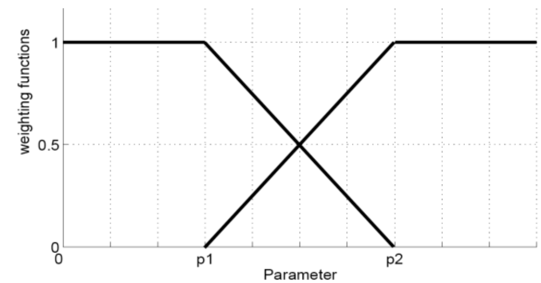
Interpolation method – Weighting functions

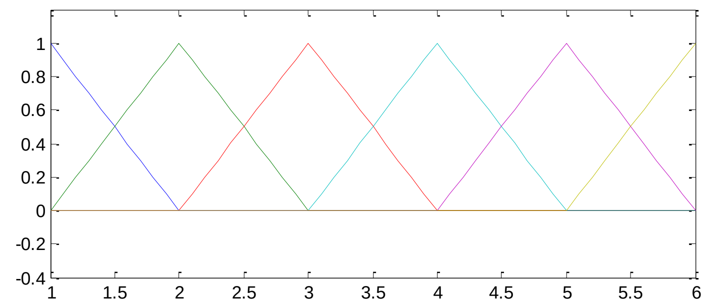
[Benner et al. '15]

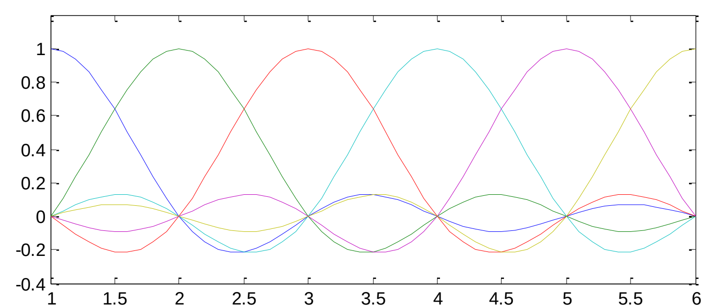
For the weighting or the interpolation, appropriate weighting functions should be selected!

Basically, any multivariate interpolation method could be used for this purpose:

- Polynomial interpolation (Lagrange polynomials)
- Piecewise linear interpolation
- Piecewise polynomial interpolation (e.g. bi-/trilinear, cubic (splines), …)
- Radial basis functions (RBF)
- Kriging interpolation (Gaussian regression)
- Inverse distance weighting (IDW) based on nearest-neighbor interpolation
- Sparse grid interpolation







pMOR Software

Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

psssMOR Toolbox – Analysis and Reduction of Parametric Models in MATLAB

- ✓ Definition of **parametric sparse state-space models**

```
psys = psss(func,userData,names)
psys = loadFemBeam3D(Opts)
psys = loadAnemometer3parameter
```

- ✓ Manipulation of **psss-class objects**

```
psys = fixParameter(psys,2,1.7)
psys = unfixParameter(psys,3)
```

- ✓ **Compatible** with the sss & sssMOR toolboxes

```
param = [p1, p2, p3, p4]
sys = psys(param)

bode(psys,param); step(sys);
```

- ✓ Different **parametric reduction methods** available (offline- & online-phase)

```
psysr = matrInterpOffline
(psys,param,r,Opts);

psysr = globalPmorOffline
(psys,param,r,Opts)

sysr = psysr(pQuery)
```

- ✓ **`localReduction`** & **`adaptiveSampling`** as **core functions**

```
[sysrp,Vp,Wp] =
localReduction(psys,param,r,Opts)

paramRef =
adaptiveSampling(psys,param,r)
```

www.rt.mw.tum.de/?morlab

www.rt.mw.tum.de/?psssmor

Numerical Examples

pMOR in Applications

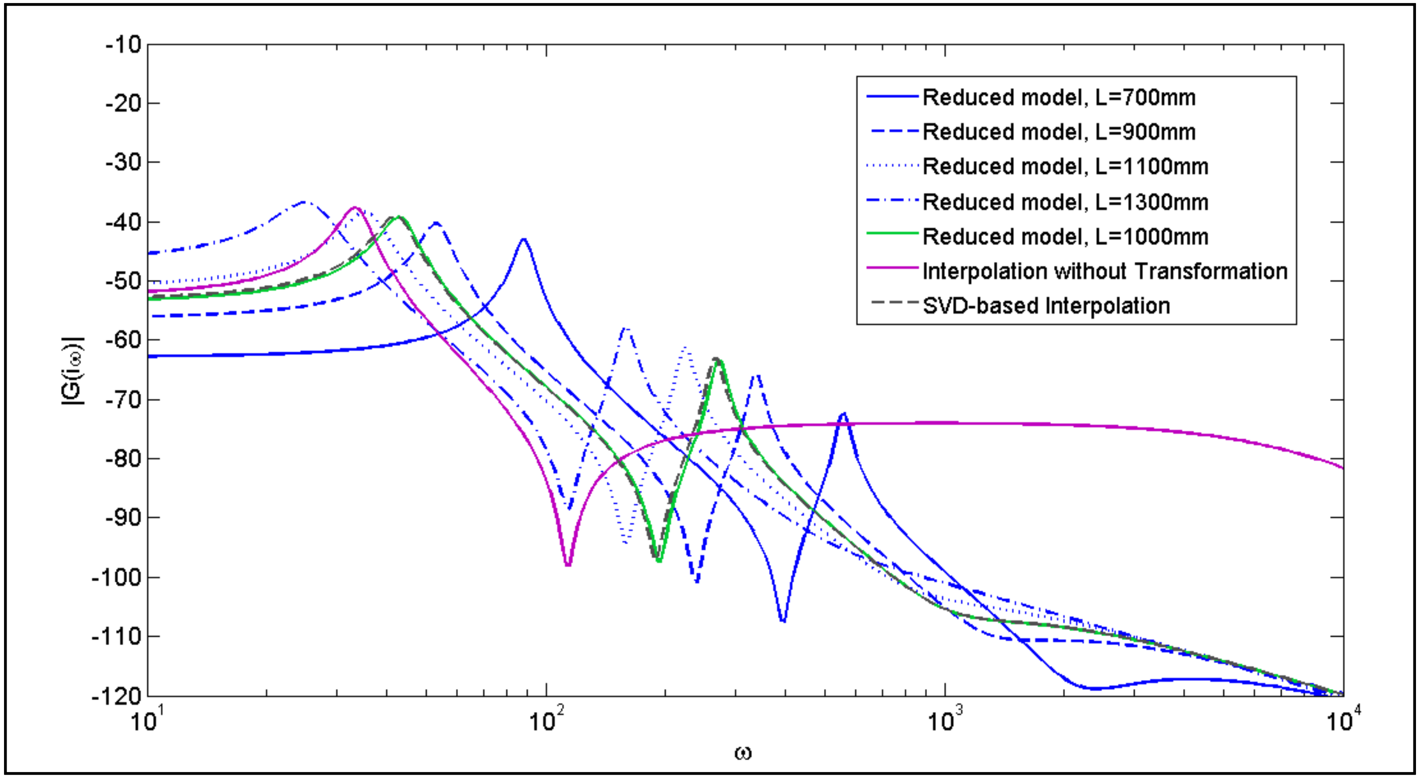
Numerical example: Beam model

Parameter: Length *L*

Thickness and width: 10 mm
Young Modulus: 2.10^5 Pa.
Damping: Proportional/Rayleigh

Order of the original system: **720**
Order of the reduced system: **5**

4 local models; Weights: Lagrange interp.
s_0: ICOP (Eid2009);



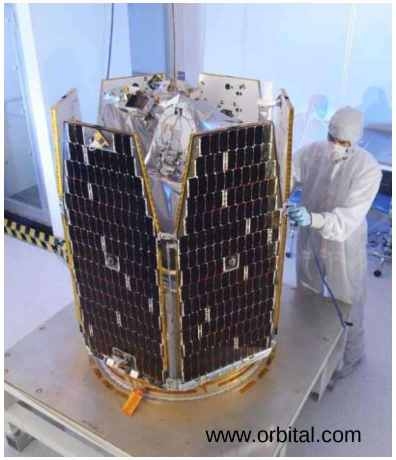
Numerical example: Solar panel model

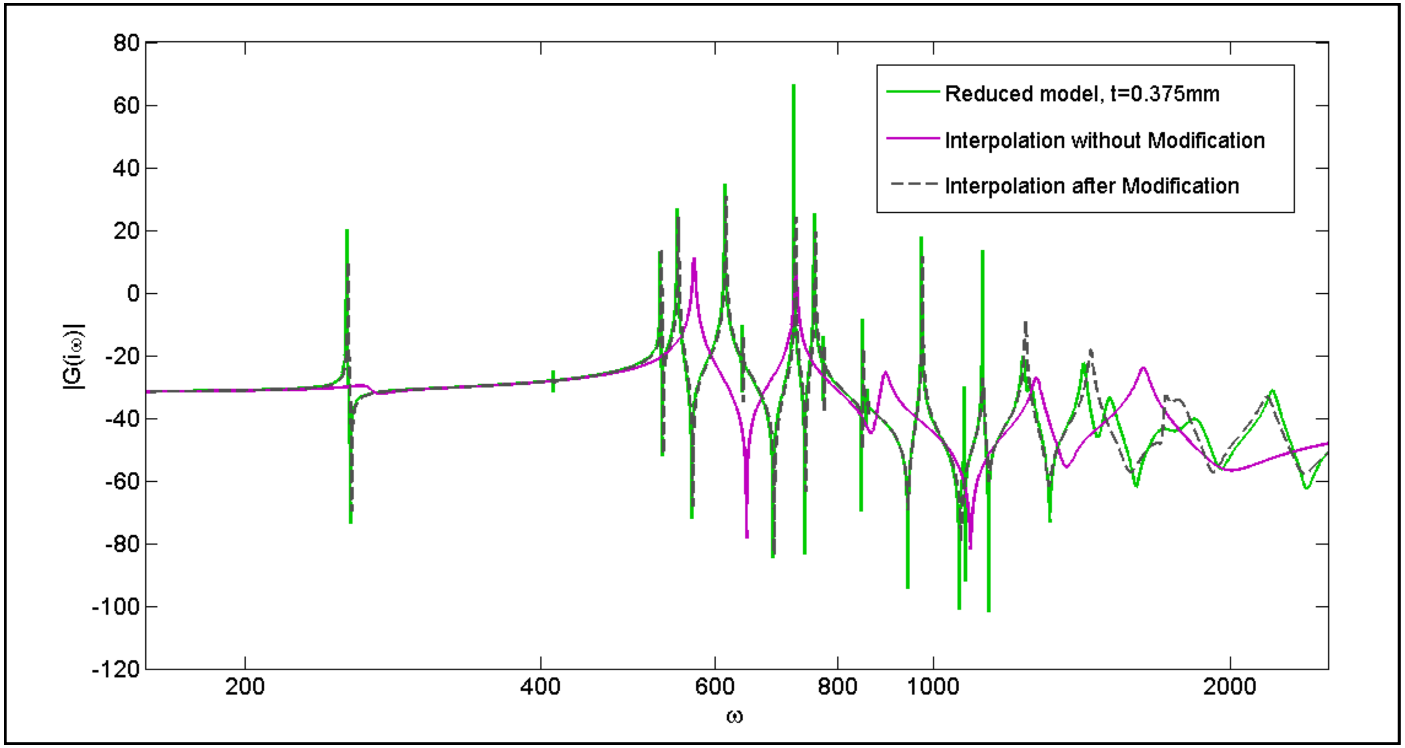
Parameter: Thickness *t* of the panel
(varies between 0.25 and 0.5 mm)

Order of the original system: 5892

Order of the reduced system: 60

2 local models; Weights: Linear interpolation





Numerical example: Timoshenko Beam

[Cruz et al. '17]

- Finite element 3D model of a Timoshenko beam
- Parameter is the length of the beam: $p \equiv L$
- One-sided Krylov reduction with shifts at $s_0 = 0$
- $\theta_{\max} = 10^\circ$ chosen

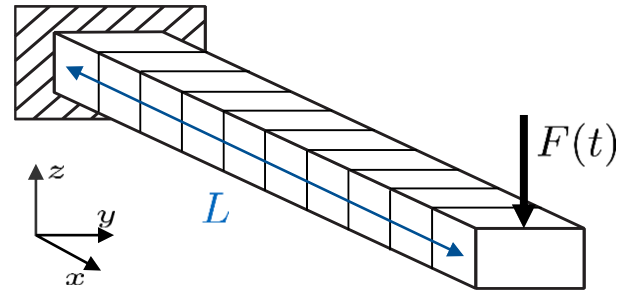


Table 1: Sample points p_i, subspace angles $\theta_{i,i+1}$ and ratios $l_{i,i+1}$

iter 1	p_i[m]	0.5		1.5		2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[^\circ]$		25.79		14.20		8.61		7.06		5.73	
	$l_{i,i+1}$		3		2		1		1		1	

iter 2	p_i[m]	0.5		0.833		1.167		1.5		2		2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[^\circ]$		10.15		8.59		7.05		8.18		6.03		8.61		7.06		5.73	
	$l_{i,i+1}$		2		1		1		1		1		1		1		1	

iter 3	p_i[m]	0.5		0.667		0.833		1.167		1.5		2		2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[^\circ]$		5.26		4.89		8.59		7.05		8.18		6.03		8.61		7.06		5.73	
	$l_{i,i+1}$		1		1		1		1		1		1		1		1		1	

Numerical example: Timoshenko Beam

[Cruz et al. ’17]

Initial uniform grid with $K = 6$

Adaptive Sampling Scheme

Table 1: Sample points p_i, subspace angles $\theta_{i,i+1}$ and ratios $l_{i,i+1}$

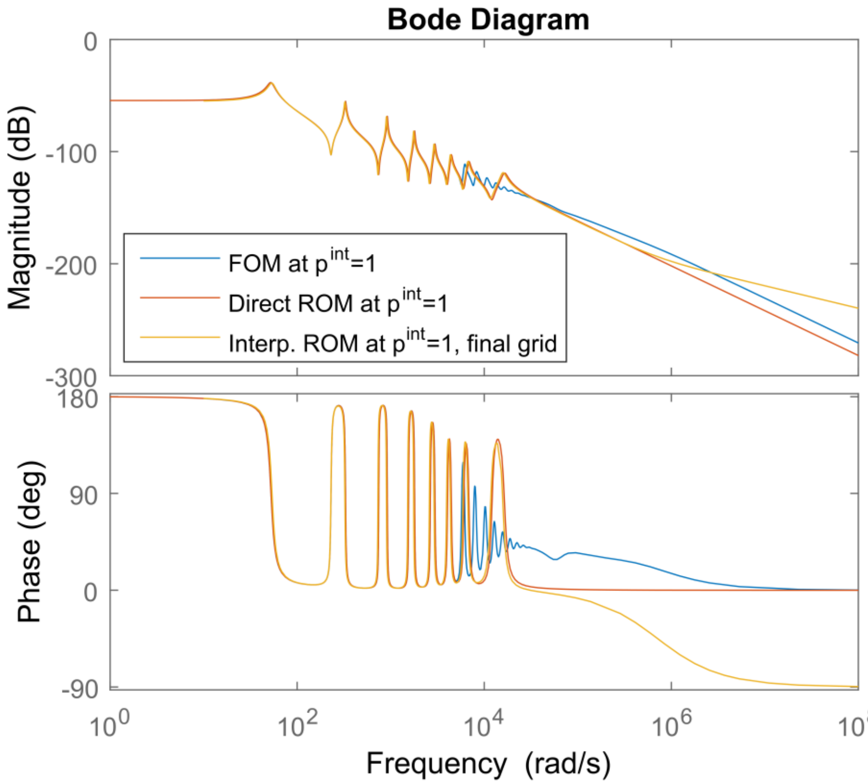
iter 1	$p_i[\mathrm{m}]$	0.5								1.5				2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[^\circ]$								25.79				14.20		8.61		7.06		5.73	
	$l_{i,i+1}$								3				2		1		1		1	
iter 2	$p_i[\mathrm{m}]$	0.5				0.833		1.167		1.5		2		2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[^\circ]$				10.15		8.59		7.05		8.18		6.03		8.61		7.06		5.73	
	$l_{i,i+1}$				2		1		1		1		1		1		1		1	
iter 3	$p_i[\mathrm{m}]$	0.5		0.667		0.833		1.167		1.5		2		2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[^\circ]$		5.26		4.89		8.59		7.05		8.18		6.03		8.61		7.06		5.73	
	$l_{i,i+1}$		1		1		1		1		1		1		1		1		1	

Interpolation point $p^{\mathrm{int}} = 1.0$
between ROM 3 & ROM 4

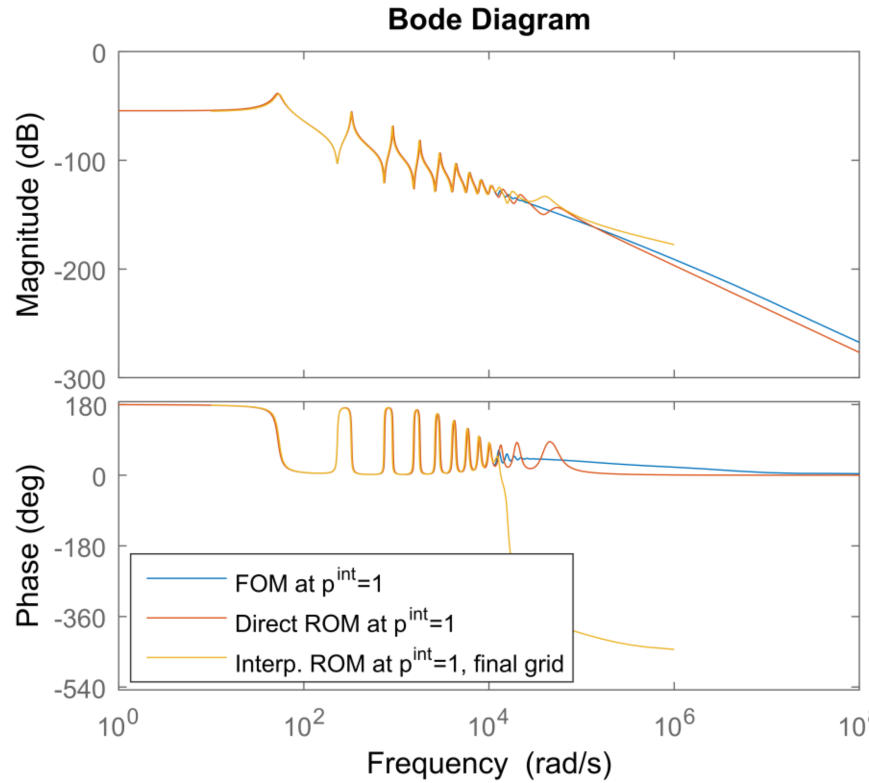
Final refined grid with $N = 10$

Numerical results – Direct vs. Interpolated ROM

[Cruz et al. ’17]



FOM size n = 240, ROMs size r = 17



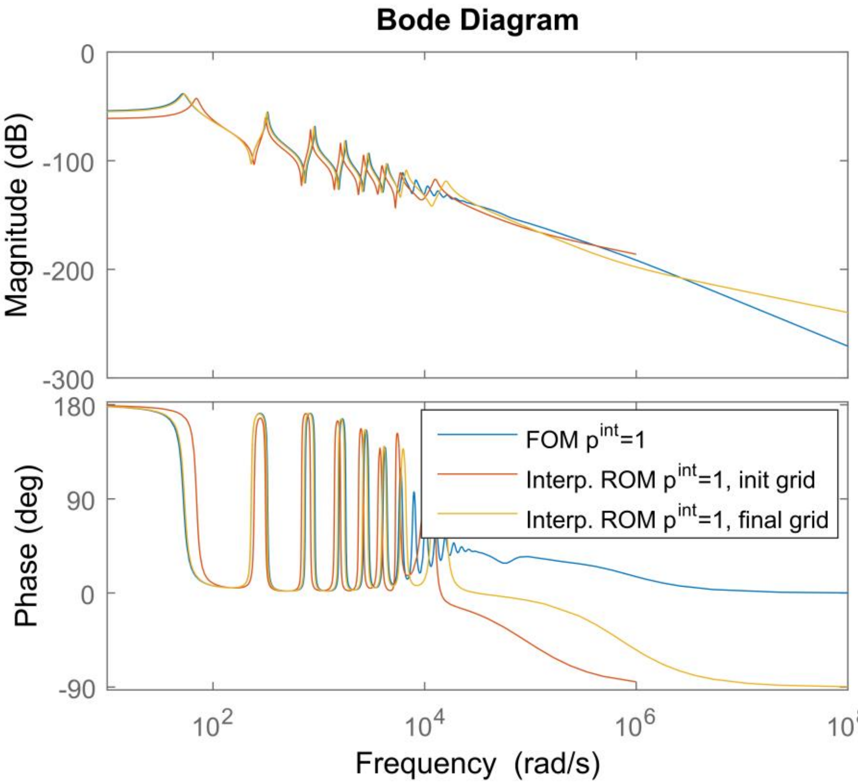
FOM size n = 2400, ROMs size r = 25

Two errors: model reduction error + interpolation error

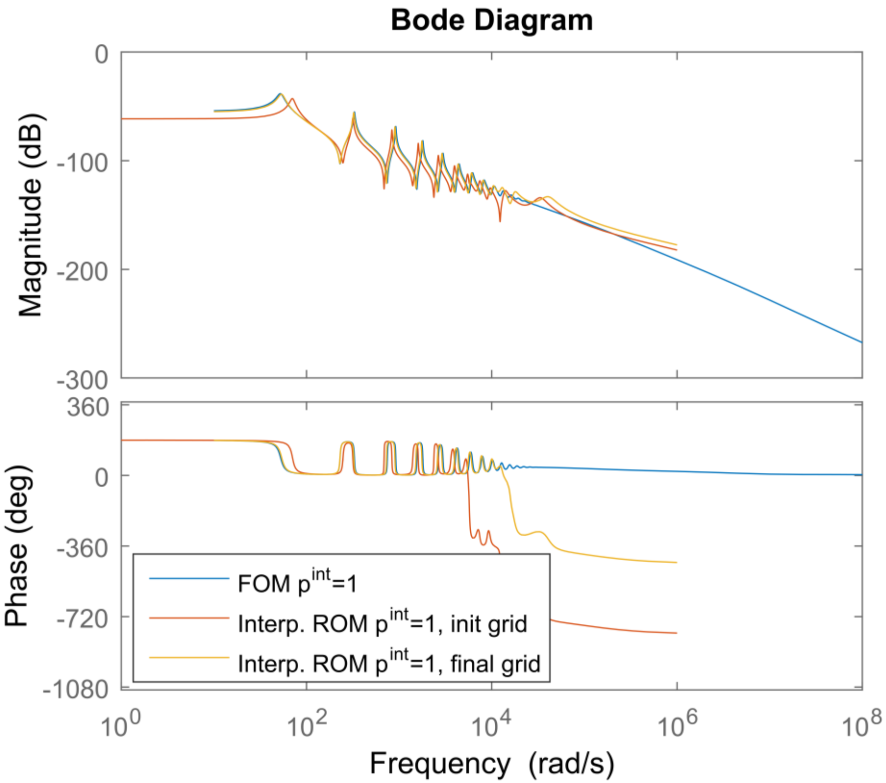
With MatrInterp: no need to reduce the model for every new parameter value

Numerical results – Initial vs. Final Grid

[Cruz et al. '17]



FOM size n = 240, ROMs size r = 17

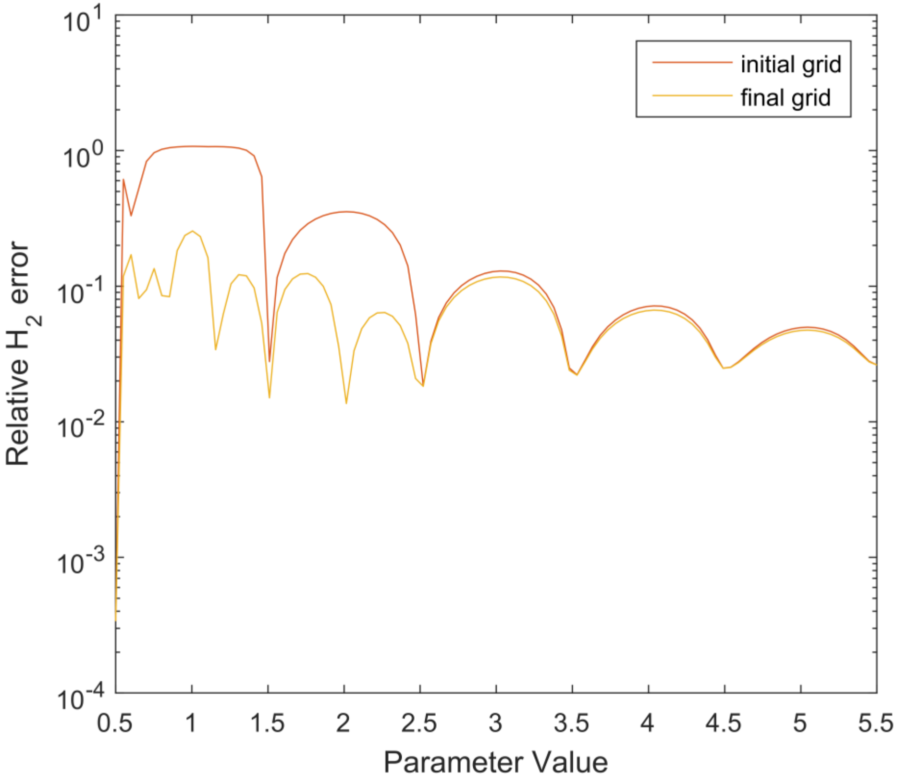


FOM size n = 2400, ROMs size r = 25

ROMs calculated with the final grid yield better approximations

Numerical results – Initial vs. Final Grid

[Cruz et al. ’17]



Relative H2 error between FOMs and interpolated ROMs for different parameter values and grids: FOM size n = 240, ROM size r = 17

- **Quantitative evaluation of the approximation**
- **Relative H2 error for** `nP=100` **different query points** p^{int}
- **Errors particularly small in the proximity of the sample points**
- **Final grid yields smaller errors for smaller beam lengths due to the adaptive refinement in this region**

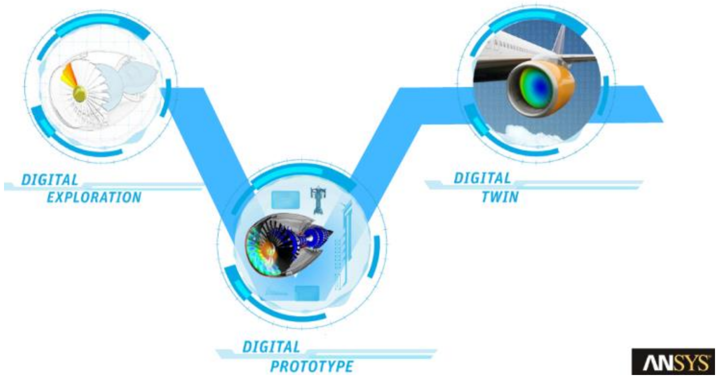
pMOR in Applications

Off-line applications:

- Efficient numerical simulation – “solves in seconds vs. hours”
- **Design optimization – analysis for different parameters and “what if” scenarios**
- Computer-aided failure mode and effects analysis (FMEA) – validation

On-line applications:

- **Parameter estimation, Uncertainty Quantification**
- **Inverse problems, Real-time optimization**
- Digital Twin, Predictive Maintenance



Physical domains:

mechanical, electrical, thermal, fluid, acoustics, electromagnetism, …

Application areas:

CSD, CFD, FSI, EMBS, MEMS, crash simulation, vibroacoustics, civil & geo, biomedical, …

pMOR in Applications – Some success stories

[Baur et al. '16] U. Baur, P. Benner, B. Haasdonk, C. Himpe, I. Martini and M. Ohlberger. Comparison of methods for parametric model order reduction of time-dependent problems.

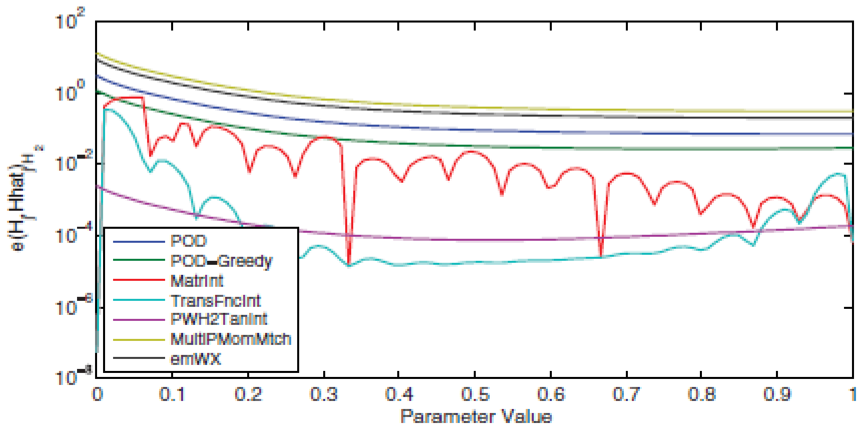


Figure 9.24. *Relative $\mathcal{H}_2$-error for the anemometer.*

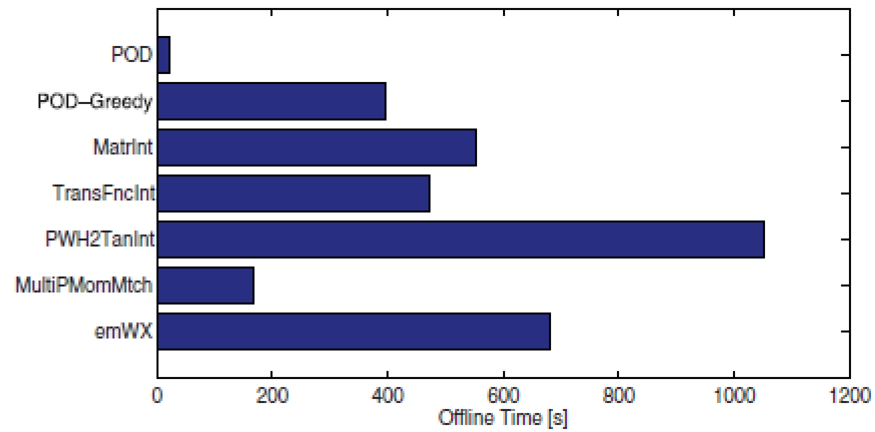
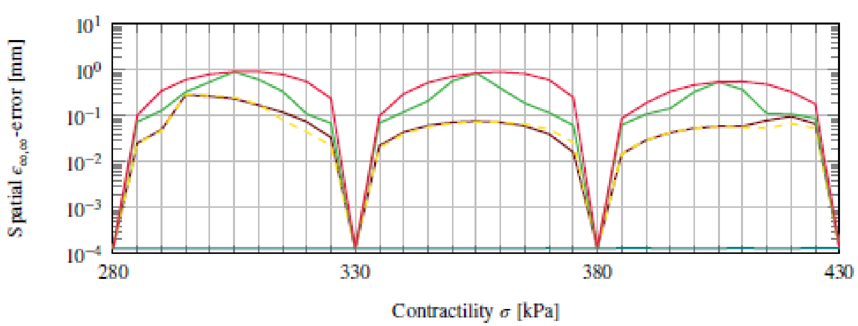
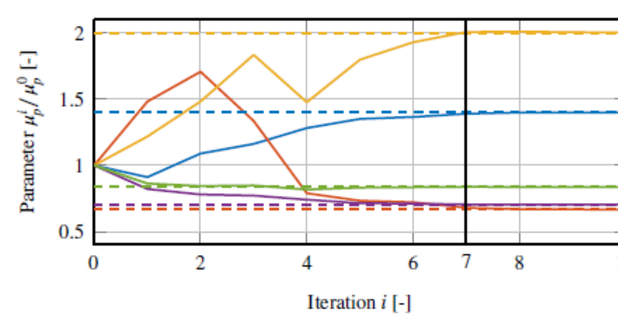


Figure 9.25. *Offline times for the anemometer.*

[Pfaller et al. '19] M. R. Pfaller, M. Cruz Varona, J. Lang, C. Bertoglio and W. A. Wall. Parametric model order reduction and its application to inverse analysis of large nonlinear coupled cardiac problems. (on arxiv)



(c) Four sample points $\sigma_k = \{280, 330, 380, 430\}$ [kPa], increment $\Delta\sigma = 5$ kPa.



(b) pROM300 parameters.

ROM direct — ROM constant — pROM Grassmann — pROM CoB — pROM direct interpolation — pROM CoS

σ — α_{max} — α_{min} — t_{syst} — t_{dias}

FIGURE 12 Convergence of parameters in inverse analysis.

Summary & Outlook

References

Summary & Outlook

Takehome Messages:

- Large, *parametric* FEM/FVM models (linear/nonlinear) arise in many technical applications!
- Parametric MOR (pMOR) is indispensable to reduce the computational effort!
- Global and local pMOR approaches exist: e.g. concatenation of bases, interpolation of bases and matrix interpolation
- Offline/online decomposition of the methods
- Efficient sampling of the parameter space is crucial, especially for many parameters (d>10)
- Different interpolation methods and weighting functions available (linear, splines, RBF, …)
- pROMs can be applied for an efficient design optimization, inverse analysis, uncertainty quantification, etc.

Challenges / Outlook:

- High-dimensional parameter spaces:
  - ➢ Adaptive sampling schemes
  - ➢ Avoiding the curse of dimensionality (tensor techniques!?)
- pMOR for systems with time-dependent parameters: p(t)MOR

References (I)

[Amsallem ’08] D. Amsallem and C. Farhat. An interpolation method for adapting reduced-order models and application to aeroelasticity. AIAA Journal, 46(7):1803-1813, 2008.

[Amsallem ’11] D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. SIAM Journal on Scientific Computing, 33(5):2169-2198, 2011.

[Baur ’09] U. Baur and P. Benner. Model reduction for parametric systems using balanced truncation and interpolation. at-Automatisierungstechnik, 57(8):411-419, 2009.

[Baur et al. ’11] U. Baur, C. Beattie, P. Benner and S. Gugercin. Interpolatory projection methods for parameterized model reduction. SIAM Journal on Scientific Computing, 33(5):2489-2518, 2011.

[Baur et al. ’16] U. Baur, P. Benner, B. Haasdonk, C. Himpe, I. Martini and M. Ohlberger. Comparison of methods for parametric model order reduction of time-dependent problems. Model reduction and approximation: Theory and Algorithms, SIAM, Chapter 9, 377-407, 2016.

[Benner et al. ’15] P. Benner, S. Gugercin and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Review, 55(4):483-531, 2015.

[Cruz et al. ’17] M. Cruz Varona, M. Nabi and B. Lohmann. Automatic Adaptive Sampling in Parametric Model Order Reduction by Matrix Interpolation. In: IEEE Advanced Intelligent Mechatronics (AIM), 472-477, 2017.

[Geuss et al. ’13] M. Geuss, H. Panzer and B. Lohmann. On parametric model order reduction by matrix interpolation. In: IEEE European Control Conference (ECC), 3433-3438, 2013.

[Geuss et al. ’14a] M. Geuss, H. Panzer, T. Wolf and B. Lohmann. Stability preservation for parametric model order reduction by matrix interpolation. In: IEEE European Control Conference (ECC), 1098-1103, 2014.

References (II)

[Geuss et al. ’14b] M. Geuss, H. K. F. Panzer, I. D. Clifford and B. Lohmann. Parametric Model Order Reduction Using Pseudoinverses for the Matrix Interpolation of Differently Sized Reduced Models. IFAC Proceedings Volumes, 47(3):9468-9473, Elsevier, 2014.

[Geuss et al. ’15] M. Geuss, B. Lohmann, B. Peherstorfer and K. Willcox. A black-box method for parametric model order reduction. In F. Breitenecker, A. Kugi and I. Troch (eds.), 8th MATHMOD, 127-128, 2015.

[Lohmann/Eid ’09] B. Lohmann and R. Eid. Efficient order reduction of parametric and nonlinear models by superposition of locally reduced models. In Methoden und Anwendungen der Regelungstechnik – Erlangen-Münchener Workshops 2007 und 2008, 2009.

[Panzer et al. ’09] H. Panzer, J. Hubele, R. Eid and B. Lohmann. Generating a parametric finite element model of a 3D cantilever Timoshenko beam using MATLAB. Technical reports on automatic control (Vol. TRAC-4), Lehrstuhl für Regelungstechnik, Technische Universität München, 2009.

[Panzer et al. ’10] H. Panzer, J. Mohring, R. Eid and B. Lohmann. Parametric model order reduction by matrix interpolation. at-Automatisierungstechnik, 58(8):475-484, 2010.

[Pfaller et al. ’19] M. R. Pfaller, M. Cruz Varona, J. Lang, C. Bertoglio and W. A. Wall. Parametric model order reduction and its application to inverse analysis of large nonlinear coupled cardiac problems. Submitted to International Journal for Numerical Methods in Biomedical Engineering (https://arxiv.org/abs/1810.12033).

Backup

pMOR by Matrix Interpolation – Features

Properties:

- Local pMOR approach
- Analytical expression of the parameter-dependency in general not available
- Model only available at certain parameter sample points

Main idea:

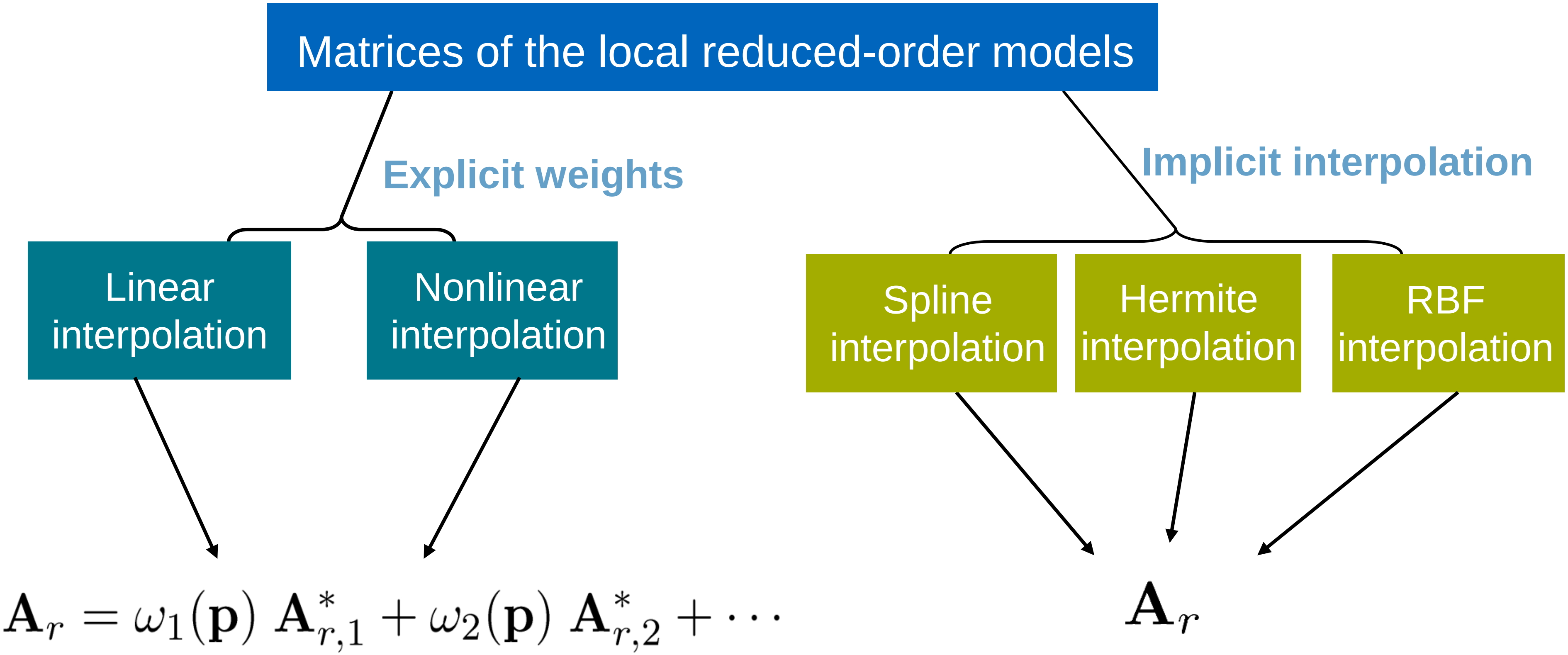
1. Individual reduction of each local model
2. Transformation of the local reduced models
3. Interpolation of the reduced matrices

Advantages
• No analytically expressed parameter-dependency required • Any desired MOR technique applicable for the local reduction • Offline/Online decomposition • Reduced order independent of the number of local models

Drawbacks

• Choice of degrees of freedom
  – Parameter sample points
  – Interpolation method
• Stability preservation
• Error bounds

Types of weighting functions / Interpolations



pMOR by Matrix Interpolation

Evaluation of the method according to different criteria

Criterion	Evaluation
Structure preservation	
Reduced order	
Storage effort	
Computational cost	
Offline/Online decomposition	
Stability preservation	
Error bounds	

Extensions for the Matrix Interpolation

Vereinheitlichendes Framework
[Geuss et al. '13]

Framework mit folgenden Schritten:
1.) Wahl der Parameterstützstellen
2.) Reduktion der lokalen Modelle
3.) Anpassung der lokalen Basen
4.) Wahl der Interpolationsmannigfaltigkeit
5.) Wahl der Interpolationsmethode

Stabilitätserhaltung
[Geuss et al. '14a]

- Interpolation (selbst stabiler) reduzierter Modelle garantiert i.A. keine Stabilität
- **Idee:** Stabile reduzierte Modelle auf dissipative Form bringen, damit ein stabiles interpoliertes System resultiert
  → Lösung von Lyapunov-Gleichungen

Interpolation zwischen Modellen verschiedener reduzierter Ordnung
[Geuss et al. '14b]

- Interpolation zwischen Modellen mit unterschiedlicher reduzierter Ordnung r_i nicht möglich
- **Idee:** Basen $\mathbf{V}_i, \mathbf{W}_i$ auf dieselbe Größe r_0 bringen durch die Berechnung von $\mathbf{T}_i, \mathbf{M}_i$ mittels Pseudoinversen

Black-Box Methode
[Geuss et al. '15]

- **Ziel:** Automatisierte pMOR-Methode
- **Idee:** Kreuzvalidierungsfehler für die iterative Ermittlung von Stützstellen und die optimale Wahl der Interpolations-mannigfaltigkeit und Interpolations-methode verwenden