An Introduction to Model Order Reduction: from linear to nonlinear dynamical systems

Maria Cruz Varona

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

June 5th 2018
Brief personal introduction

Maria Cruz Varona  
M.Sc. Electrical Engineering

University studies (10/08-03/14):  
Electrical Engineering and Information Technology (KIT)  
Study model 8: “Information and Automation”  
Master thesis at IRS (group: “cooperative systems”)

Research assistant (since 08/14):  
Chair of Automatic Control (Prof. Dr.-Ing. habil. B. Lohmann)  
Technical University of Munich

maria.cruz@tum.de
www.rt.mw.tum.de

Research interests:  
Systems theory, model order reduction, nonlinear dynamical systems, Krylov subspace methods
Chair of Automatic Control

Nonlinear Control

Model Order Reduction

Systems & Control Theory

Energy-based Modeling and Control

Automotive (Vibration and Suspension Control)

Modeling and Control of Distributed Parameter Systems

SFB Modeling of innovation processes
Chair of Automatic Control

Chair of Automatic Control
Chair of Automatic Control – MORLab

LTI Systems

Parametric and LTV Systems

Nonlinear Systems

Second Order Systems (linear and nonlinear)

Tools

ss

ssMOR

ss
Agenda

I. Motivation & Linear Model Order Reduction
   - Modeling, Modeling Strategies
   - Large-scale models, Sparsity
   - Reduced order models, Applications
   - Projective MOR, Linear MOR methods
   - Numerical Examples, FEM & MOR software

II. Polynomial & Nonlinear Model Order Reduction
   - Projective NLMOR, Overview NLMOR methods
   - Polynomial Nonlinear Systems, Volterra series representation
   - Nonlinear Systems, Proper Orthogonal Decomposition

III. Summary & Outlook
Motivation & Linear Model Order Reduction
Modeling of complex dynamical systems

- Models described by ODEs:
  \[
  \frac{d}{dt} x(t) = f(x(t), u(t))
  \]
- Models described by PDEs:
  \[
  \frac{\partial T(z, t)}{\partial t} = \frac{\partial^2 T(z, t)}{\partial z^2} + u(z, t)
  \]
Modeling – Strategies

**0D modeling:** lumped-parameter model

\[
\begin{align*}
R & \quad L \\
C
\end{align*}
\]

\[
\begin{align*}
k & \quad m \quad d
\end{align*}
\]

\[
\begin{align*}
& u \quad : \quad \text{voltage} \\
& i \quad : \quad \text{current} \\
& p \quad : \quad \text{pressure} \\
& q \quad : \quad \text{flow rate}
\end{align*}
\]

\[
\begin{align*}
& v \quad : \quad \text{velocity} \\
& F \quad : \quad \text{force} \\
& e \quad : \quad \text{effort} \\
& f \quad : \quad \text{flow}
\end{align*}
\]

**1D, 2D, 3D modeling:** distributed-parameter model

\[
\begin{align*}
p(x, t), \ q(x, t)
\end{align*}
\]

\[
\begin{align*}
p(x, y, t), \ q(x, y, t)
\end{align*}
\]

\[
\begin{align*}
p(x, y, z, t), \ q(x, y, z, t)
\end{align*}
\]

**Data-driven modeling:** identification of model using experimental data
Large-scale models from spatial discretization

Spatial discretization using:
- Finite-Difference-Method (FDM)
- Finite-Element-Method (FEM)
- Finite-Volume-Method (FVM)

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_k(t) \\ \vdots \\ x_n(t) \end{bmatrix} \quad n = 10^3, \ldots, 10^6$$

High dimension complicates:
- numerical simulation
- design optimization
- estimation, prediction & control
Sparsity of matrices

Matrices coming from FEM/FVM discretization are generally *sparse*

Storage requirement: \( A \in \mathbb{R}^{34722 \times 34722} \)

- Sparse: \(~33.2\) MB
- Full / Dense: 9.0 GB required!
Goal of Model Order Reduction (MOR)

Large-scale full order model (FOM)

\[
\begin{align*}
E \dot{x} &= A x + B u \\
 y &= C x \\
\det(E) &\neq 0
\end{align*}
\]

Reduced order model (ROM)

\[
\begin{align*}
E_r \dot{x}_r &= A_r x_r + B_r u \\
y_r &= C_r x_r \\
x_r &\in \mathbb{R}^r, \quad r \ll n
\end{align*}
\]

- Good approximation
- Preservation of properties
- Numerically efficient
Applications of ROMs

**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
- Real-time optimization and control
- 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, …
Reduced Order Modeling – Strategies

**0D modeling**: lumped-parameter / simplified model

Coarse mesh:

Fine mesh & Projection-based MOR:
Projective MOR

**Assumption:** Dynamical system does not transit all regions of the state-space equally often, but mainly stays and evolves in a subspace of lower dimension.

Approximation of the state vector:
\[ x = V x_r + e , \quad V \in \mathbb{R}^{n \times r} \]

Petrov-Galerkin projection: \[ \Pi = EV(W^T EV)^{-1} W^T \]

\[ W^T E \]
\[ V \]
\[ x_r \]
\[ W^T A V \]
\[ x_r \]
\[ W^T B u \]
\[ y \approx y_r = C V x_r \]

How to choose \( V \) and \( W \)?
Linear MOR methods – Overview

1. Modal Reduction
   • Preservation of dominant eigenmodes
   • Frequently used in structural dynamics / second order systems

2. Truncated Balanced Realization / Balanced Truncation
   • 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
   • “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 Krylov algorithm IRKA
   • H2-optimal reduction
   • Adaptive choice of Krylov reduction parameters (e.g. shifts)
Modal Reduction

**Goal:** Preserve dominant eigenmodes of the system

**Procedure:**

1. **Modal transformation:** Bring system into modal coordinates through state-transformation

   \[
   [W, \Lambda, V] = \lambda(A, E)
   \]

   \[
   W^T A V = \Lambda, \quad W^T E V = I_r
   \]

   \[
   \hat{B} = W^T B, \quad \hat{C} = C V
   \]

   \[
   x = V z
   \]

   \[
   \dot{z} = \begin{bmatrix} \Lambda_1 & \Lambda_2 \\ \hat{B}_1 & \hat{B}_2 \end{bmatrix} z + \begin{bmatrix} \hat{B}_1 \end{bmatrix} u
   \]

   \[
   y = \begin{bmatrix} \hat{C}_1 & \hat{C}_2 \end{bmatrix} z
   \]

2. **Truncation:**

   \[
   A_r = \Lambda_1, \quad B_r = \hat{B}_1, \quad C_r = \hat{C}_1, \quad E_r = I_r
   \]

**Practical implementation:**

Entire modal transformation of FOM is expensive!

→ Only a few eigenvalues and left and right eigenvectors are computed via `eigs`
Truncated Balanced Realization (TBR)

**Goal:** Preserve state-space directions with highest energy transfer

Controllability and Observability Gramians:

\[
A \ P \ E^T + E \ P \ A^T + B \ B^T = 0
\]

\[
A^T \ Q \ E + E^T \ Q \ A + C^T \ C = 0
\]

Energy interpretation:

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

\[
\|y(t)\|_2^2 = x_0^T \ Q \ x_0
\]

Procedure:

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

   \[
P = RR^T, \quad Q = SS^T
\]

   \[
S^T ER = [U_1 \ U_2] \begin{bmatrix} \Sigma_1 & \Sigma_2 \\ \Sigma_1 & \Sigma_2
\end{bmatrix} \begin{bmatrix} N_1^T \\ N_2^T
\end{bmatrix}
\]

2. **Truncation step:** \( \sigma_i \gg \sigma_j, \ i = 1, \ldots, r, \ j = r + 1, \ldots, n \)

   \[
W^T = \Sigma_1^{-1/2} U_1^T S^T, \quad V = R \ N_1 \ \Sigma_1^{-1/2}
\]
Rational Interpolation by Krylov subspace methods

Moments of a transfer function

\[ G(s) = C(sE - A)^{-1}B \]

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

\( \sigma \) : interpolation point (shift)

\( M_i(\sigma) \) : i-th moment around \( \sigma \)

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

Bases for input and output Krylov-subspaces:

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

\[ \text{Ran}(W) \supseteq \text{span} \left\{ A_\sigma^{-T}C^T, A_\sigma^{-T}E^TA_\sigma^{-T}C^T, \ldots, (A_\sigma^{-T}E^T)^{r-1}A_\sigma^{-T}C^T \right\} \]

\[ M_i(\sigma) = M_{r,i}(\sigma) \]

for \( i = 0, \ldots, 2r - 1 \)

Moments from full and reduced order model around certain shifts match!
**Goal:** Find ROM that minimizes the $\mathcal{H}_2$-error

$$\|G - G_r\|_{\mathcal{H}_2} = \min_{\dim(G_r) = r} \|G - \tilde{G}_r\|_{\mathcal{H}_2}$$

**$\mathcal{H}_2$-norm:**

$$\|G(s)\|_{\mathcal{H}_2}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(i\omega)|^2 d\omega$$

**Algorithm 1** Iterative Rational Krylov Algorithm (SISO)

**Input:** $\Sigma := (E, A, b, c^T), \sigma_i, \text{tol}$

**Output:** locally $\mathcal{H}_2$-optimal ROM $\Sigma_r^{\text{opt}}, \sigma_i^{\text{opt}}$

1. **while** (relative change of $\sigma_i > \text{tol}$) **do**
   2. $\Sigma_r \leftarrow \text{RK}(\Sigma, \sigma_i)$  \(\triangleright\) Rational Krylov reduction
   3. $\Lambda_r = \lambda(A_r, E_r)$  \(\triangleright\) Compute eigenvalues of ROM
   4. $\sigma_i \leftarrow -\lambda_r, i$  \(\triangleright\) Mirror reduced eigenvalues
   5. **end while**
6. $\Sigma_r^{\text{opt}} \leftarrow \Sigma_r, \sigma_i^{\text{opt}} \leftarrow \sigma_i$  \(\triangleright\) Return optimal ROM and shifts

IRKA achieves multipoint moment matching at optimal shifts!
Comparison: BT vs. Krylov subspace methods

**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
Numerical comparison

fom: \( n = 1006, \ r = 20 \)

steel profile rail_1357: \( n = 1357, \ r = 20 \)

<table>
<thead>
<tr>
<th>Method</th>
<th>red. time [s]</th>
<th>( |G - G_r|_{\mathcal{H}<em>2} / |G|</em>{\mathcal{H}_2} )</th>
<th>( |G - G_r|<em>{\mathcal{H}</em>\infty} / |G|<em>{\mathcal{H}</em>\infty} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>modalMor</td>
<td>0.40</td>
<td>19.40e-02</td>
<td>4.16e-02</td>
</tr>
<tr>
<td>(lr)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tbr</td>
<td>0.20</td>
<td>1.18e-09</td>
<td>5.78e-09</td>
</tr>
<tr>
<td>rk</td>
<td>0.09</td>
<td>81.47e-02</td>
<td>96.73e-02</td>
</tr>
<tr>
<td>irka</td>
<td>0.60</td>
<td>8.56e-08</td>
<td>5.80e-09</td>
</tr>
<tr>
<td>modalMor</td>
<td>1.21</td>
<td>4.61e-02</td>
<td>3.76e-03</td>
</tr>
<tr>
<td>(lr)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tbr</td>
<td>0.49</td>
<td>3.47e-05</td>
<td>2.65e-06</td>
</tr>
<tr>
<td>rk</td>
<td>0.10</td>
<td>1.34e-07</td>
<td>3.36e-07</td>
</tr>
<tr>
<td>irka</td>
<td>1.32</td>
<td>2.38e-12</td>
<td>9.61e-11</td>
</tr>
</tbody>
</table>
Toolboxes for sparse, large-scale models in

```
s = sss(A,B,C,D,E);
```

```
sysr = tbr(sys,r)
sysr = rk(sys,s0)
sysr = irka(sys,s0)
sysr = cure(sys)
sysr = cirka(sys,s0)
```

```
bode(sys), sigma(sys)
step(sys), impulse(sys)
norm(sys,2), norm(sys,inf)
c2d, lsim, eigs, connect, ...
```

Powered by: **M-M.E.S.S. toolbox** [Saak, Köhler, Benner] for Lyapunov equations
Available at [www.rt.mw.tum.de/?sssMOR](http://www.rt.mw.tum.de/?sssMOR) and [https://github.com/MORLab](https://github.com/MORLab).
[Castagnotto/Cruz Varona/Jeschek/Lohmann '17]: „sss & sssMOR: Analysis and Reduction of Large-Scale Dynamic Systems in MATLAB“, at-Automatisierungstechnik]
Main characteristics

✅ **State-space models** of very high order on a standard computer \((n \approx 10^8)\)

✅ Many Control System Toolbox functions, revisited to **exploit sparsity**

✅ Allows system analysis in **frequency** \((\text{bode, sigma,} \ldots)\) and **time domain** \((\text{step, norm, lsim,} \ldots)\), as well as **manipulations** \((\text{connect, truncate,} \ldots)\)

✅ **Is compatible** with the built-in syntax

✅ **New functionality:** `eigs`, `residue`, `pzmap`, ...

✅ **Classical** \((\text{modalMor, tbr, rk,} \ldots)\) and **state-of-the-art** \((\text{isrk, irka, cirka, cure,} \ldots)\) **reduction methods**

✅ **Both highly-automatized**

\[
\text{sysr} = \text{irka}(\text{sys}, n)
\]

and **highly-customizable**

\[
\begin{align*}
\text{Opts.maxiter} &= 100 \\
\text{Opts.tol} &= 1e-6 \\
\text{Opts.stopcrit} &= \text{‘combAll’} \\
\text{Opts.verbose} &= \text{true} \\
\text{sysr} &= \text{irka}(\text{sys}, n, \text{Opts})
\end{align*}
\]

✅ **solveLse** and **lyapchol** as **core functions**
Comprehensive **documentation** with examples and references

**ssssMOR App**
 graphical user interface

completely **free**
and **open source**
(contributions welcome)
Comprehensive documentation with examples and references

sssMOR App graphical user interface

completely free and open source (contributions welcome)
Comprehensive documentation with examples and references

**sssMOR App**

graphical user interface

completely free and open source

(contributions welcome)
FEM & MOR software

Commercial FEM software:
ANSYS, Abaqus, COMSOL Multiphysics, LS-DYNA, Nastran, …

Open-source FEM software:
AMfe, CalculiX, FEniCS Project, FreeFEM++, JuliaFEM, KRATOS, OOFEM, OpenFOAM, …

Open-source Pre-/Post-Processing tools:
Gmsh, ParaView, …

Open-source MOR software:
pyMOR, sss, sssMOR, pssMOR, emgr, M.E.S.S., MOREMBS, MORE, RBmatlab, …
Polynomial & Nonlinear Model Order Reduction
Projective MOR for Nonlinear Systems

Given a large-scale nonlinear control system of the form

\[ E \dot{x} = f(x, u) \]
\[ y = h(x) \]

\[ x(t) \in \mathbb{R}^n \]

with \( f(x, u) : \mathbb{R}^n \times \mathbb{R}^1 \to \mathbb{R}^n \) and \( h(x) : \mathbb{R}^n \to \mathbb{R}^1 \)

Simulation, design, control and optimization cannot be done efficiently!

Reduced order model (ROM)

\[
\begin{align*}
E_r & \quad V^T \\
\dot{x}_r & = V^T f(V x_r, u) \\
y_r & = h(V x_r) \\
x_r(t) & \in \mathbb{R}^r, \quad r \ll n
\end{align*}
\]

with \( f_r(x_r, u) : \mathbb{R}^r \times \mathbb{R}^1 \to \mathbb{R}^r \) and \( h(x_r) : \mathbb{R}^r \to \mathbb{R}^1 \)

Goal:
\[ y_r(t) \approx y(t) \]
Challenges of Nonlinear MOR

Nonlinear systems can exhibit complex behaviours
  • Strong nonlinearities
  • Multiple equilibrium points
  • Limit cycles
  • Chaotic behaviours

Input-output behaviour of nonlinear systems cannot be described with transfer functions, the state-transition matrix or the convolution integral (only possible for special cases)

Choice of the reduced order basis
  • Projection basis should comprise most dominant directions of the state-space
  • Different existing approaches:
    ▪ Simulation-based methods
    ▪ System-theoretic techniques

Expensive evaluation of \( f(Vx_r) \)
  • Vector of nonlinearities \( f \) still has to be evaluated in full dimension
  • Approximation of \( f \) by so-called hyper-reduction techniques:
    \( \rightarrow \) EIM, DEIM, GNAT, ECSW…
Nonlinear MOR methods – Overview

**Polynomial nonlinear systems**

**Reduction of bilinear systems**

\[ E\dot{x} = Ax + N xu + bu \]
\[ y = c^T x \]

- Transfer of system-theoretic concepts
- Generalization of linear MOR methods:
  - Balanced truncation
  - **Krylov / \( H_2 \)-optimal approach**

**Reduction of quadratic-bilinear systems**

\[ E\dot{x} = Ax + H(x \otimes x) + N xu + bu \]
\[ y = c^T x \]

- Reduction methods for **MIMO** models
- **Input-awareness:**
  - signal generators
  - eigenfunctions

**Nonlinear systems**

**Reduction of nonlinear (parametric) systems**

\[ E\dot{x} = f(x, u) \]
\[ y = h(x) \]

- Simulation-based:
  - POD, TPWL
  - Reduced Basis, Empirical Gramians
- **Simulation-free / System-theoretic**
Polynomial Nonlinear Systems

Polynomialization / Carleman linearization

Starting point:  
\[ E \dot{x} = f(x) + g(x) u \]
\[ y = c^T x \]

Assumptions:
- \( x_S = 0 \)
- \( f(x_S) = 0 \)

\[ E \dot{x} = A^{(1)} x + A^{(2)} (x \otimes x) + \cdots + N^{(1)} x u + \cdots + b u \]
\[ y = c^T x \]

Bilinear dynamical systems

- Result from direct modeling or Carleman (bi)linearization
- Linear in input and linear in state, but not jointly linear in both
- Interface between fully nonlinear and linear systems

\[ E \dot{x} = Ax + Nxu + bu \]
\[ y = c^T x \]
Volterra series representation

\[
\dot{x}(t) = A x(t) + N x(t) u(t) + b u(t), \quad x(0) = x_0,
\]
\[
y(t) = c^T x(t).
\]

Picard fixed-point iteration (successive approximation)

Approximate solution of the bilinear system

\[
x_1(t) = \int_{\tau=0}^{t} e^{A(t-\tau)} b u(\tau) \, d\tau + e^{At} x_0,
\]
\[
x_k(t) = \int_{\tau=0}^{t} e^{A(t-\tau)} N u(\tau) x_{k-1}(\tau) \, d\tau, \quad k \geq 2.
\]

Variational equations (subsystems)

Interpretation as a series of homogenous, cascaded subsystems:

\[
\dot{x}_1(t) = A x_1(t) + b u(t), \quad x_1(0) = x_0,
\]
\[
\dot{x}_k(t) = A x_k(t) + N x_{k-1}(t) u(t), \quad x_k(0) = 0, \quad k \geq 2.
\]
Systems Theory for Volterra systems (1) [Rugh ’81]

Input-Output behavior

\[ y(t) = \sum_{k=1}^{\infty} y_k(t) \]

\[ y(t) = \sum_{k=1}^{\infty} \prod_{\tau_1=-\infty}^{\infty} \prod_{\tau_k=-\infty}^{\infty} \left( c^T e^{A_{\tau_k}} N \cdots N e^{A_{\tau_2}} N e^{A_{\tau_1}} b \right) \]

\[ \times u(t - \tau_k) \cdots u(t - \tau_k - \cdots - \tau_1) \, d\tau_k \cdots d\tau_1 \]

Kernels

\[ k = 1 : \quad g_1(\tau_1) = c^T e^{A_{\tau_1}} b \]

\[ k = 2 : \quad g_2(\tau_1, \tau_2) = c^T e^{A_{\tau_2}} N e^{A_{\tau_1}} b \]

\[ k = 3 : \quad g_3(\tau_1, \tau_2, \tau_3) = c^T e^{A_{\tau_3}} N e^{A_{\tau_2}} N e^{A_{\tau_1}} b \]

Transfer functions

\[ k = 1 : \quad G_1(s_1) = c^T (s_1 I - A)^{-1} b \]

\[ k = 2 : \quad G_2(s_1, s_2) = c^T (s_2 I - A)^{-1} N (s_1 I - A)^{-1} b \]

\[ k = 3 : \quad G_3(s_1, s_2, s_3) = c^T (s_3 I - A)^{-1} N (s_2 I - A)^{-1} N (s_1 I - A)^{-1} b \]
Systems Theory for Volterra systems (2) [Rugh ’81]

Gramians

\[ P = \sum_{k=1}^{\infty} P_k, \quad Q = \sum_{k=1}^{\infty} Q_k, \]

\[ g_k(\tau_1, \ldots, \tau_k) = c^T e^{A\tau_k} N \cdots N e^{A\tau_2} N e^{A\tau_1} b \]

\[ \bar{p}_k(\tau_1, \ldots, \tau_k) \]

\[ \bar{q}_k(\tau_1, \ldots, \tau_k)^T \]

\[ P_k = \int_{\tau_1=0}^{\infty} \cdots \int_{\tau_k=0}^{\infty} \bar{p}_k(\tau_1, \ldots, \tau_k)\bar{p}_k(\tau_1, \ldots, \tau_k)^T d\tau_1 \cdots d\tau_k \]

\[ Q_k = \int_{\tau_1=0}^{\infty} \cdots \int_{\tau_k=0}^{\infty} \bar{q}_k(\tau_1, \ldots, \tau_k)\bar{q}_k(\tau_1, \ldots, \tau_k)^T d\tau_1 \cdots d\tau_k \]

H2-norm

\[ \|\zeta\|_{H_2}^2 = \sum_{k=1}^{\infty} \int_{\tau_1=0}^{\infty} \cdots \int_{\tau_k=0}^{\infty} g_k(\tau_1, \ldots, \tau_k)g_k(\tau_1, \ldots, \tau_k)^T d\tau_1 \cdots d\tau_k \]

\[ = \sum_{k=1}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \frac{1}{(2\pi)^k} G_k(i\omega_1, \ldots, i\omega_k) G_k(-i\omega_1, \ldots, -i\omega_k)^T d\omega_1 \cdots d\omega_k \]

\[ = c^T P c = b^T Q b \]
Multipoint Volterra Series Interpolation

**Goal:** Enforcing multipoint interpolation of the underlying Volterra series

Multipoint Volterra series interpolation

Set of interpolation points: \( S = \{\sigma_1, \ldots, \sigma_r\} \), \( i = 1, \ldots, r \)

\[
\sum_{k=1}^{\infty} \sum_{l_1=1}^{r} \cdots \sum_{l_{k-1}=1}^{r} \eta_{l_1,\ldots,l_{k-1},i} G_k(\sigma_{l_1}, \ldots, \sigma_{l_{k-1}}, \sigma_i) = \sum_{k=1}^{\infty} \sum_{l_1=1}^{r} \cdots \sum_{l_{k-1}=1}^{r} \eta_{l_1,\ldots,l_{k-1},i} G_{k,r}(\sigma_{l_1}, \ldots, \sigma_{l_{k-1}}, \sigma_i)
\]

This approach interpolates the weighted series at the interpolation points \( \sigma_1, \ldots, \sigma_r \)

Projection matrices for Volterra series interpolation

\[
v_i = \sum_{k=1}^{\infty} \sum_{l_1=1}^{r} \cdots \sum_{l_{k-1}=1}^{r} \eta_{l_1,\ldots,l_{k-1},i} (\sigma_i E - A)^{-1} N (\sigma_{l_{k-1}} E - A)^{-1} N \cdots N (\sigma_{l_1} E - A)^{-1} b
\]

\[
w_i = \sum_{k=1}^{\infty} \sum_{l_1=1}^{r} \cdots \sum_{l_{k-1}=1}^{r} \nu_{l_1,\ldots,l_{k-1},i} (\mu_{l_1} E - A)^{-T} N^T (\mu_{l_2} E - A)^{-T} N^T \cdots N^T (\mu_i E - A)^{-T} c
\]
Proper Orthogonal Decomposition (POD)

Starting point: \( E \dot{x} = f(x, u) \)
\[ y = h(x) \]

1. Choose suitable training input signals \( u_1(t), u_2(t), \ldots, u_t(t) \)
2. Take snapshots from simulated full order state trajectories
\[ X_{(n,n_s)} = [x^{u_1}(t_1), x^{u_1}(t_2), \ldots, x^{u_1}(t_N) x^{u_2}(t_1), x^{u_2}(t_2), \ldots] \]
3. Perform singular value decomposition (SVD) of snapshot matrix \( X \)
\[ X \approx M_{(n,r)} \Sigma_{(r,n_s)} N_{(n_s,n_s)}^T \]
4. Reduced order basis: \( V = M_r \in \mathbb{R}^{n \times r} \)

Advantages:
- Straightforward data-driven method
- Choice of reduced order from singular values / error bound for approx. error
- Optimal in least squares sense:
\[ \min_{\text{rank}(X_r)=r} \| X - X_r \|_2 \]

Disadvantages:
- Simulation of full order model for different input signals required
- SVD of large snapshot matrix required
- Training input dependency
Summary & Outlook

Take-Home Messages:

- Modeling via FEM/FVM is becoming more and more important!
- Applicable for several physical domains and many technical applications!
- Model Order Reduction is indispensable to reduce the computational effort
- Reduction is done via projection
- Linear MOR is well developed
- Generalization of system-theoretic concepts and MOR methods to polynomial systems
- POD is still the most employed nonlinear MOR method
- Simulation-free / System-theoretic nonlinear MOR techniques are aimed

Ongoing work:

- Polynomial nonlinear systems
- Simulation-free / System-theoretic NLMOR
Thank you for your attention!
References

[Antoulas ’05] Approximation of Large-Scale Dynamical Systems. SIAM.

[Astolfi ’10] Model reduction by moment matching for linear and nonlinear systems. IEEE TAC.


[Flagg/Gugercin ’15] Multipoint Volterra series interpolation and H2 optimal model reduction of bilinear systems, SIAM Journal on Matrix...


[Rugh ’81] Nonlinear system theory. The Volterra/Wiener Approach. The Johns Hopkins University Press