MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Empirical Gramians and Friends

Christian Himpe

2016-10-11

CSC

Outline

1. My Research Interests
2. A Tour Through my Scientific Deeds
3. Lift-Off for MathEnergy
4. Scientific Sideshows

Research Interests

in a Nutshell:

- Combined State and Parameter Reduction
- for Nonlinear Input-Output Systems
- Modeling (Complex) Networks

Notation

Nonlinear Input-Output System:

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t), \theta) \\ y(t) &= g(t, x(t), u(t), \theta) \\ x(0) &= x_0 \end{aligned}$$

Input Function:	$u : \mathbb{R}_{\geq 0} \to \mathbb{R}^M$
State Trajectory:	$x : \mathbb{R}_{\geq 0} \to \mathbb{R}^N$
Output Trajectory:	$y : \mathbb{R}_{\geq 0} \to \mathbb{R}^Q$
Parameter:	$\theta \in \mathbb{R}^P$
Vector-Field:	$f : \mathbb{R}_{\geq 0} \times \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^N$
Output Functional:	$g : \mathbb{R}_{\geq 0} \times \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \to \mathbb{R}^Q$

Combined Reduction

Reduced Order Model:

$$
\begin{aligned}
\dot{x}_r(t) &= f_r(t, x_r(t), u(t), \theta_r) \\
y_r(t) &= g_r(t, x_r(t), u(t), \theta_r) \\
x_r(0) &= x_{r,0}
\end{aligned}
$$

State Trajectory:	$x_r : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$
Output Trajectory:	$y_r : \mathbb{R}_{\geq 0} \to \mathbb{R}^Q$
Parameter:	$\theta_r \in \mathbb{R}^p$
Vector-Field:	$f_r : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^M \times \mathbb{R}^p \to \mathbb{R}^n$
Output Functional:	$g_r : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \times \mathbb{R}^M \times \mathbb{R}^p \to \mathbb{R}^Q$
Reduced State Dimension:	$n \ll N$
Reduced Parameter Dimension:	$p \ll P$
Accuracy:	$\|y(\theta) - y_r(\theta_r)\| \ll 1$

Projection-Based Reduction

Projection-Based Reduced Order Model:

$$\begin{aligned}
\dot{x}_r(t) &= Vf(t, Ux_r(t), u(t), \Pi\theta_r) \\
y_r(t) &= g(t, Ux_r(t), u(t), \Pi\theta_r) \\
x_r(0) &= Vx_0 \\
\theta_r &= \Lambda\theta
\end{aligned}$$

Reducing Truncated State Projection: $V \in \mathbb{R}^{n \times N}$

Reconstructing Truncated State Projection: $U \in \mathbb{R}^{N \times n}$

Bi-Orthogonality: $VU = \mathbb{1}$

Reducing Truncated Parameter Projection: $\Lambda \in \mathbb{R}^{p \times P}$

Reconstructing Truncated Parameter Projection: $\Pi \in \mathbb{R}^{P \times p}$

Bi-Orthogonality: $\Lambda\Pi = \mathbb{1}$

Previously ...

- Dynamic Causal Modelling
- Combined State and Parameter Reduction
  - Empirical-Cross-Gramian-Based
  - Greedy-Optimization-Based
- Hierarchical Approximate POD

Dynamic Causal Modelling (DCM)

Dynamic Causal Modelling [Friston et al'03]

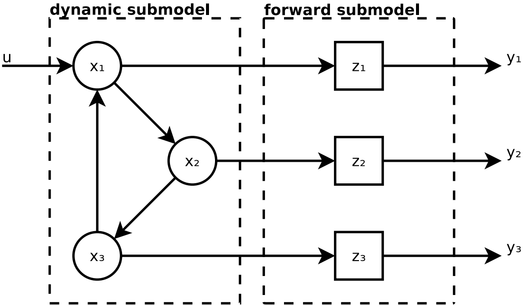
Aim:

- Reconstruction of brain connectivity
- from functional neuroimaging data (fMRI/fNIRS, EEG/MEG)

Concept:

- Hierarchical Model
  1. Network Submodel (Dynamic Submodel)
  2. Observation Submodel (Forward Submodel)
- Connectivity Parametrization
- Effective Connectivity (& Lateral Connectivity)
- SIMO Models (& MIMO Models)
- Bayesian Inference

fMRI & fNIRS Model



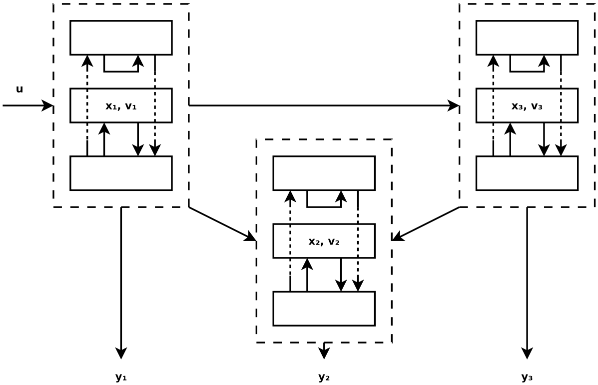
Dynamic Submodel (Taylor):

$$
\begin{aligned}
\dot{x}(t) &= f(x, u, \theta) \\
&\approx f(0, 0, \theta) + \frac{\partial f}{\partial x} x(t) + \frac{\partial f}{\partial u} u(t) \\
&= A(\theta) x(t) + B u(t)
\end{aligned}
$$

$$A_{ij}(\theta) := \theta_{ik+j} \rightarrow x \in \mathbb{R}^k \theta \in \mathbb{R}^{k^2}$$

Forward Submodel (Balloon):

$$
\begin{aligned}
\dot{s}_i(t) &= x_i(t) - \kappa s_i(t) - \gamma(f_i(t) - 1) \\
\dot{f}_i(t) &= s_i(t) \\
\dot{v}_i(t) &= \frac{1}{\tau}(f_i(t) - v_i(t)^{\frac{1}{\alpha}}) \\
\dot{q}_i(t) &= \frac{1}{\tau}(\frac{1}{\rho} f_i(t)(1 - ((1 - \rho))^{\frac{1}{f_i(t)}}) - v_i(t)^{\frac{1}{\alpha} - 1} q_i(t)) \\
y_i(t) &= V_0(k_1(1 - q_i(t)) + k_2(1 - v_i(t)))
\end{aligned}
$$



Neural Mass Model:

$$\begin{pmatrix} \dot{v}(t) \\ \dot{x}(t) \end{pmatrix} = \begin{pmatrix} 0 & \mathbb{1} \\ -T & -T^2 \end{pmatrix} \begin{pmatrix} v(t) \\ x(t) \end{pmatrix} + \begin{pmatrix} 0 \\ A(\theta) \end{pmatrix} \varsigma(Kx(t)) + \begin{pmatrix} 0 \\ B \end{pmatrix} u(t)$$

$$y(t) = Lx(t)$$

Efficient DCM Implemention

Bayesian Inference using:

- EM Algorithm
  - **E** xpectation
    - Weighted
    - Tikhonov regularized
    - Least-squares
    - Temporal Correlations
  - **M** aximization
- Various additional modifications

See my diploma thesis: `doi:10.6084/m9.figshare.1027354.v1`

Now, what if one wants to infer on large networks ($N > 6 \to P > 36$) ?

Combined State and Parameter Reduction

Combined Reduction

(Together with M. Ohlberger)

1. **Empirical-Cross-Gramian-Based**
2. Greedy-Optimization-Based

Cross Gramian [Fernando & Nicholson'83]

Linear time-invariant system:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

Hankel operator:

$$H := \mathcal{O}\mathcal{C}$$

Cross Gramian (assume a square system: $M = Q$):

$$W_X := \mathcal{C}\mathcal{O} = \int_0^\infty \mathrm{e}^{At} BC \, \mathrm{e}^{At} \, \mathrm{d}t \in \mathbb{R}^{N \times N}$$
$$\Rightarrow AW_X + W_X A = -BC$$

Cross Gramian's central property:

$$CA^{-1}B = (CA^{-1}B)^\intercal \Rightarrow \sigma_i(H) = |\lambda_i(W_X)|$$

Cross-Gramian-Based Reduction

Balancing Transformation (for a symmetric system) [Aldhaheri'91]:

$$W_X \stackrel{EVD}{=} T \Lambda T^{-1}$$

Approximate Balancing Transformation [Sorensen & Antoulas'02]:

$$W_X \stackrel{SVD}{=} UDV$$

Direct Truncation [Himpe & Ohlberger'14]:

$$W_X \stackrel{\text{SVD}}{=} UDV \to U = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \to V_1 := U_1^\intercal$$

Empirical Cross Gramian [Streif et al'06]

Empirical linear cross Gramian:

$$W_X = \int_0^\infty \mathrm{e}^{At} BC \, \mathrm{e}^{At} \, \mathrm{d}t = \int_0^\infty \left(\mathrm{e}^{At} B \right) \left(\mathrm{e}^{A^\intercal t} C^\intercal \right)^\intercal \mathrm{d}t$$

Empirical cross Gramian:

$$\widehat{W}_X := \frac{1}{KLM} \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{m=1}^{M} \frac{1}{c_k d_l} \int_0^\infty \Psi^{klm}(t) \, \mathrm{d}t \in \mathbb{R}^{N \times N}$$

$$\Psi_{ij}^{klm}(t) = \langle x_i^{km}(t) - \bar{x}_i, y_m^{lj}(t) - \bar{y}_m \rangle$$

- State trajectory $x^{km}(t)$ for the impulse input $u(t) = c_k e_m \delta(t)$.
- Output trajectory $y^{lj}(t)$ for the initial state $x_0 = d_l e_j$.

Joint Gramian [Himpe & Ohlberger'14]

Augmented system (treat parameters as constant states):

$$\begin{pmatrix} \dot{x}(t) \\ \dot{\theta}(t) \end{pmatrix} = \begin{pmatrix} f(t, x(t), u(t), \theta(t)) \\ 0 \end{pmatrix}$$

$$y(t) = g(t, x(t), u(t), \theta(t))$$

$$\begin{pmatrix} x(0) \\ \theta(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ \theta_0 \end{pmatrix}$$

Joint Gramian (empirical cross Gramian of the augmented system):

$$W_J = \begin{pmatrix} W_X & W_m \\ 0 & 0 \end{pmatrix}$$

- Lower blocks are zero, as parameter-states are uncontrollable.

Cross-Identifiability Gramian

Cross-Identifiability Gramian (Schur complement of symmetric part of W_J):

$$W_{\ddot{I}} = -\frac{1}{2} W_m^\intercal (W_X + W_X^\intercal)^{-1} W_m$$

- $W_{\ddot{I}}$ encodes the **observability** of parameters.
- Schur-complement can be approximated efficiently.

Direct Truncation:

$$W_{\ddot{I}} \overset{\text{SVD}}{=} \Pi \Delta \Lambda \rightarrow \Pi = \begin{pmatrix} \Pi_1 & \Pi_2 \end{pmatrix} \rightarrow \Lambda_1 := \Pi_1^\intercal$$

Cross-Gramian-Based Combined Reduction

1. Compute Empirical Joint Gramian W_J.
2. State-Space Direct Truncation:
$$W_X \overset{\text{SVD}}{=} UDV \rightarrow U = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \rightarrow V_1 := U_1^\intercal$$
3. Parameter-Space Direct Truncation:
$$W_{\ddot{I}} \overset{\text{SVD}}{=} \Pi\Delta\Lambda \rightarrow \Pi = \begin{pmatrix} \Pi_1 & \Pi_2 \end{pmatrix} \rightarrow \Lambda_1 := \Pi_1^\intercal$$

Empirical-Cross-Gramian-Based Reduced Order Model:

$$\begin{aligned} \dot{x}_r(t) &= V_1 f(t, U_1 x_r(t), u(t), \Pi_1 \theta_r) \\ y_r(t) &= g(t, U_1 x_r(t), u(t), \Pi_1 \theta_r) \\ x_r(0) &= V_1 x_0 \\ \theta_r &= \Lambda_1 \theta \end{aligned}$$

Non-Symmetric Cross Gramian [Himpe & Ohlberger'16]

A cross Gramian for non-square and non-symmetric systems:

Cross Gramian Superposition ($B = \begin{pmatrix} b_1 \dots b_M \end{pmatrix}$, $C = \begin{pmatrix} c_1 \dots c_Q \end{pmatrix}^\intercal$):

$$W_X = \sum_{i=1}^{M=Q} \int_0^\infty \mathrm{e}^{At}\, b_i c_i\, \mathrm{e}^{At}\, \mathrm{d}t$$

Non-Symmetric Cross Gramian:

$$\begin{aligned} W_Z &:= \sum_{i=1}^{M} \sum_{j=1}^{Q} \int_0^\infty \mathrm{e}^{At}\, b_i c_j\, \mathrm{e}^{At}\, \mathrm{d}t \\ &= \int_0^\infty \mathrm{e}^{At} \big(\sum_{i=1}^{M} b_i \big) \big(\sum_{j=1}^{Q} c_j \big)\, \mathrm{e}^{At}\, \mathrm{d}t \end{aligned}$$

Interestingly, there exists a connection to tangential interpolation.

Empirical Gramians:

- Empirical Controllability Gramian
- Empirical Observability Gramian
- Empirical Linear Cross Gramian
- Empirical Cross Gramian
- Empirical Sensitivity Gramian
- Empirical Identifiability Gramian
- Empirical Joint Gramian

Features:

- Interfaces for: Solver, Kernel, Distributed Memory
- Non-Symmetric option for all cross Gramians
- Compatible with OCTAVE and MATLAB
- Vectorized and parallelizable
- Open-source licensed

More info at: http://gramian.de

Combined Reduction

(Together with M. Ohlberger)

1. Empirical-Cross-Gramian-Based
2. **Greedy-Optimization-Based**

Idea: Alternatingly extend parameter and state projection

1. Select parameter base vector θ_{I+1} for parameter projection:
   $\Pi_{I+1} = \{\Pi_I \cup (\theta_{I+1} - \Pi_I \Pi_I^T \theta_{I+1})\}$

2. Select state base vector $\bar{x}(\theta_{I+1})$ based on parameter θ_{I+1}:
   $U_{I+1} = \{U_I \cup (\bar{x}(\theta_{I+1}) - U_I U_I^T \bar{x}(\theta_{I+1}))\}$

3. Go to 1.

- Parameter selection: Greedy method
- State selection: Energy-based

Parameter Reduction [Bui-Thanh et al'08]

Greedy Sampling Strategy:

- Locate currently worst approximated parameter
- Formulation and solution as optimization problem:

$$\theta_{I+1} = \text{arg max}_{\theta \in \Theta} \|y(\theta) - y(\Pi_I \Pi_I^\intercal \theta)\|_{L_2}^2 - \beta_2 \|\theta\|_2^2$$

subject to:

$$\dot{x}(t) = f(t, x(t), u(t), \theta)$$
$$y(t) = g(t, x(t), u(t), \theta)$$
$$x(0) = x_0$$

- Adaptive sampling (no pre-defined grid)
- Tikhonov regularization

State Reduction

In each iteration, after selection of θ_{I+1}:

1. Simulate a state trajectory $x(\theta_{I+1})$
2. Select principal component $\bar{x}(\theta_{I+1})$

- Based on input-to-state mapping $u \mapsto x$
- $U_1 = \arg\min_{U \in \mathbb{R}^{N \times n}, U^\intercal U = \mathbb{1}} \|x(\theta_{I+1}) - U x_r(\theta_{I+1})\|_{L_2}^2$
- POD model reduction
- proven method for nonlinear systems

Data-Driven Regularization [Himpe & Ohlberger'15]

An inverse problem provides data y_d.

Extended cost function with data mismatch as regularization:

$$J_d = \|y(\theta) - y_r(\theta_r)\|_{L_2}^2 - \beta_2\|\theta\|_2^2 - \beta_d\|y(\theta) - y_d\|_{L_2}^2$$

- Reduced order model is specific to data
- Partial inversion during reduction
- Accelerates projection assembly
- Determining weighting coefficient more complicated

Optimization-Based Combined Reduction

Truncated parameter-space projection:

$$\Pi_{I+1} \to \Lambda_{I+1} := \Pi_{I+1}^\intercal$$

Truncated state-space projection:

$$U_{I+1} \to V_{I+1} := U_{I+1}^\intercal$$

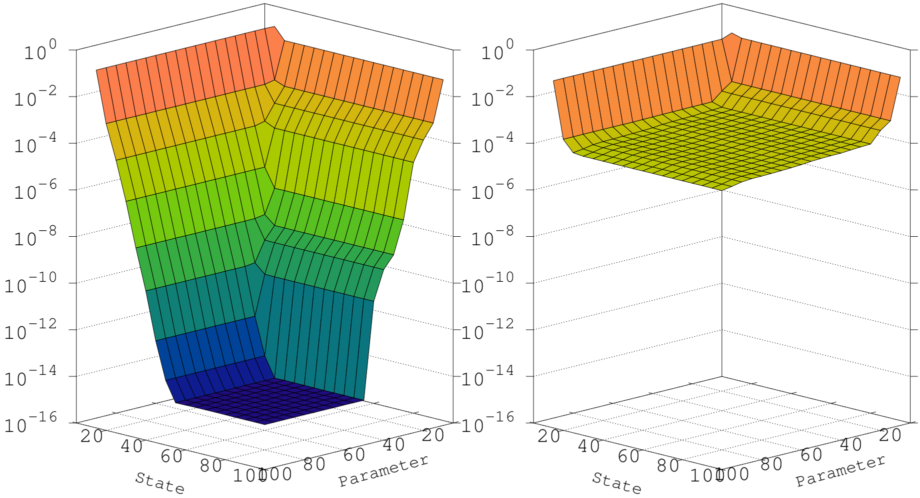
Greedy-optimization-based reduced order model:

$$\begin{aligned}
\dot{x}_r(t) &= V_{I+1} f(t, U_{I+1} x_r(t), u(t), \Pi_{I+1}\theta_r) \\
y_r(t) &= g(t, U_{I+1} x_r(t), u(t), \Pi_{I+1}\theta_r) \\
x_r(0) &= V_{I+1} x_0 \\
\theta_r &= \Lambda_{I+1}\theta
\end{aligned}$$

Software Implementation: `http://github.com/gramian/optmor`

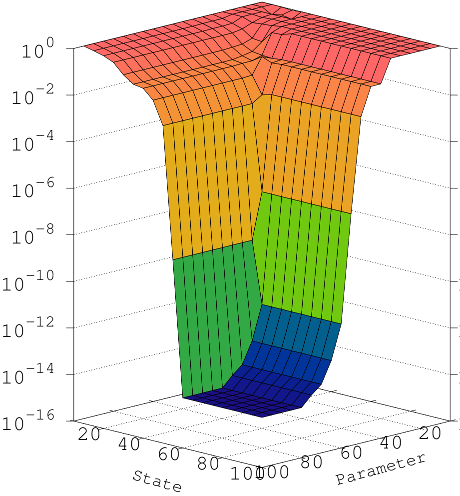
Combined Reduction I

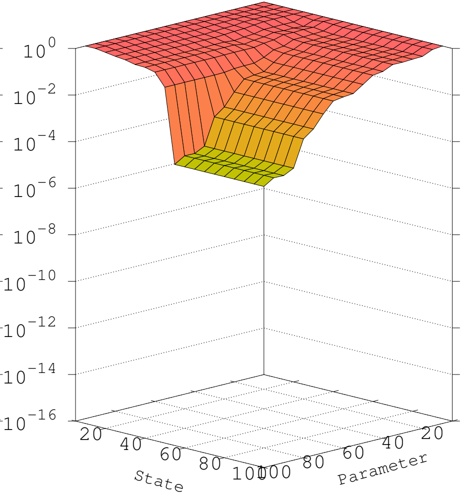
Nonlinear Resistor-Capacitor Ladder: Gramian- vs Optimization-Based



$L_2 \otimes L_2$-output error for varying reduced state and parameter dimensions.

Hyperbolic Network Model: Gramian- vs Optimization-Based

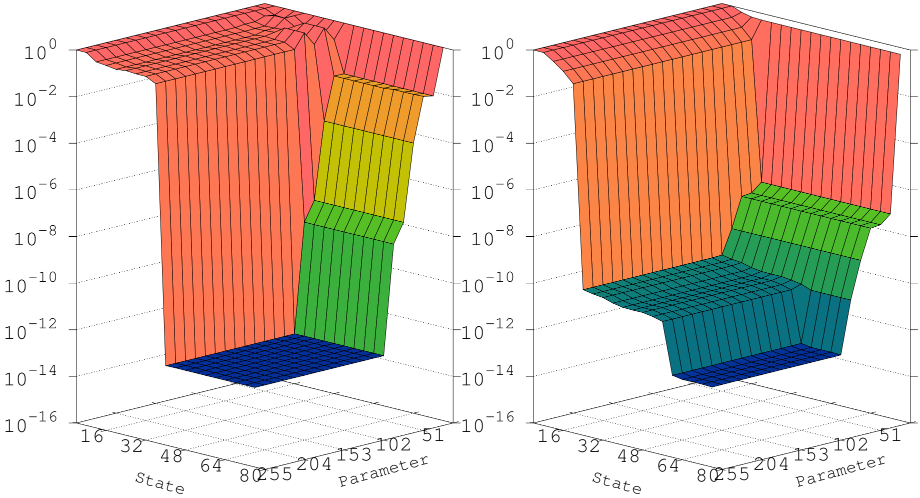




$L_2 \otimes L_2$-output error for varying reduced state and parameter dimensions.

Combined Reduction III

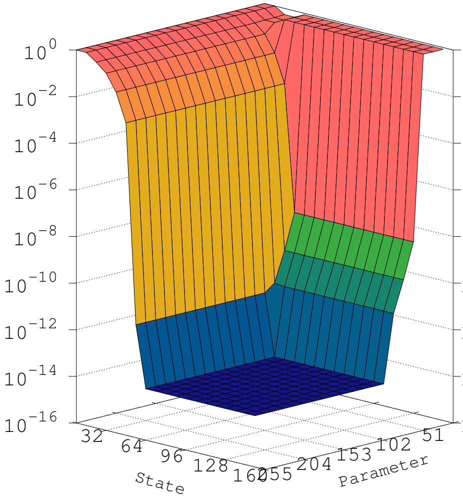
fMRI & fNIRS Dynamic Causal Model: Gramian- vs Optimization-Based



$L_2 \otimes L_2$-output error for varying reduced state and parameter dimensions.

Combined Reduction IIII

EEG & MEG Dynamic Causal Model: Gramian- vs Optimization-Based



$L_2 \otimes L_2$-output error for varying reduced state and parameter dimensions.

Hierarchical Approximate POD

Hierarchical Approximate POD

(Together with T. Leibner & S. Rave)

Why HAPOD? Memory-bound or compute-bound POD applications.

Properties:

- Column-wise slicing of snapshots
- Mean L_2 projection error bound
- Maximum mode bound
- Rooted tree organization
- Modular POD backend

Special Cases:

- Distributed HAPOD (Minimum communication parallel)
- Rolling HAPOD (Minimum memory footprint)

See our preprint: `http://arxiv.org/abs/1607.05210`

HAPOD

Definition:
Let $S \subset V$ be a finite multiset of snapshot vectors in a Hilbert space V. Given a rooted tree $\mathcal{T}$ and mappings:

$$\mathcal{D} : \mathcal{S} \to \mathcal{L}_\mathcal{T},$$
$$\epsilon_\mathcal{T} : \mathcal{N}_\mathcal{T} \to \mathbb{R}_{\geq 0},$$

define recursively for each $\alpha \in \mathcal{N}_\mathcal{T}$

$$\text{HAPOD}(\mathcal{S}, \mathcal{T}, \mathcal{D}, \epsilon_\mathcal{T})(\alpha) := \text{POD}(\mathcal{I}_\alpha, \epsilon_\mathcal{T}(\alpha))$$

where the local input data multiset $\mathcal{I}_\alpha$ is given by

$$\mathcal{I}_\alpha := \begin{cases} \mathcal{D}^{-1}(\alpha) & \alpha \in \mathcal{L}_\mathcal{T} \\ \bigcup_{\beta \in \mathcal{C}_\mathcal{T}(\alpha)} \{\sigma_n \cdot \varphi_n | (\sigma_n, \varphi_n) \in \text{HAPOD}(\mathcal{S}, \mathcal{T}, \mathcal{D}, \epsilon_\mathcal{T})(\beta)\} & \text{otherwise.} \end{cases}$$

We call $\text{HAPOD}(\mathcal{S}, \mathcal{T}, \mathcal{D}, \epsilon_\mathcal{T})$ the **hierarchical approximate POD** (HAPOD) of $\mathcal{S}$ for the rooted tree $\mathcal{T}$, the snapshot mapping $\mathcal{D}$ and the local tolerances $\epsilon_\mathcal{T}$.

Faster Empirical Cross Gramians

Simplified Description of the Empirical Cross Gramian:

$$\widehat{W}_X = \sum_{m=1}^{M} \int_0^\infty \Psi^m(t)\,\mathrm{d}t \in \mathbb{R}^{N \times N}$$

$$\Psi_{ij}^m(t) := \langle x_i^m(t), y_m^j(t) \rangle$$

Column-Wise Compution of the Empirical Cross Gramian (k-th column):

$$\Rightarrow \widehat{W}_{X,*k} = \sum_{m=1}^{M} \int_0^\infty \psi^{mk}(t)\,\mathrm{d}t \in \mathbb{R}^{N \times 1}$$

$$\psi_i^{mk}(t) := \langle x_i^m(t), y_m^k(t) \rangle$$

Low-Rank Empircal Cross Gramian via HAPOD!

MathEnergy Project

(Together with P. Benner and S. Grundel)

MathEnergy – Mathematical Key Technologies for Evolving Energy Grids

Subproject: Model Order Reduction

- Model Reduction for Gas Networks
- Software implementation
- Power-to-Gas integration
- BMWi funded

Federal Ministry
for Economic Affairs
and Energy

Model [Grundel et al'14]

Hierarchical Model:

1. Network
2. Pipes (Euler equation):

$$\frac{\partial p(x,t)}{\partial t} = -\frac{c^2}{A}\frac{\partial q(x,t)}{\partial x}$$
$$\frac{\partial q(x,t)}{\partial t} = -A\frac{\partial p(x,t)}{\partial x} + \frac{\lambda c^2}{2DA}\frac{q(x,t)|q(x,t)|}{p(x,t)}$$

Challenges:

- Semi-discretized $\rightarrow$ DAE System
- Nonlinear
- **Hyperbolic**

Perspective Extensions: Compressibility, Elevation, Temperature

Reduction

- Empirical Gramians
- Modifications for
  - Hyperbolic systems
  - DAE systems
- Alternative methods
- Efficient simulation
  - FOM
  - ROM
- Implementation
- Comparison of
  - Models
  - Reduction approaches

Scientific Sideshows

- Replicability, Reproducibility & Reusability (doi:10.3934/Math.2016.3.261)
- MORwiki (http://modelreduction.org)
- MOR Benchmark Framework
- Stability Analysis on ROMs (with: A. Petersen)
- RKHS Method for Empirical Gramians (with: B. Hamzi)
- Scientific Computing on Single-Board Computers
- FlexiBLAS (http://www.mpi-magdeburg.mpg.de/projects/flexiblas)
- Explicit Runge-Kutta Methods
- Efficient Octave / MATLAB Code (http://git.io/mtips)
- "Good" Colormaps

tl;dl

- Combined State and Parameter Reduction
- for Nonlinear Network Systems
- using Empirical Cross Gramians

http://himpe.science

References

- C. Himpe and T. Leibner and S. Rave. **Hierarchical Approximate Proper Orthogonal Decomposition**. Preprint arXiv math.NA: 1–24, 2016.
- J. Fehr, J. Heiland, C. Himpe and J. Saak. **Best Practices for Replicability, Reproducibility and Reusability of Computer-Based Experiments Exemplified by Model Reduction Software**. AIMS Mathematics 1(3): 262–281, 2016.
- C. Himpe and M. Ohlberger. **A Note on the Non-Symmetric Cross Gramian**. System Science and Control Engineering 4(1): 199–208, 2016.
- U. Baur, P. Benner, B. Haasdonk, C. Himpe, I. Martini and M. Ohlberger. **Comparison of methods for parametric model order reduction of instationary problems**. To appear in "Model Reduction and Approximation: Theory and Algorithms", 2016.
- C. Himpe and M. Ohlberger. **Data-driven combined state and parameter reduction for inverse problems**. Advances in Computational Mathematics (MoRePaS Special Issue) 41(5):1343–1364, 2015.
- C. Himpe and M. Ohlberger. **The Empirical Cross Gramian for Parametrized Nonlinear Systems**. In MATHMOD 2015, vol 48(1), pages 727–728, 2015.
- C. Himpe and M. Ohlberger. **Cross-Gramian Based Combined State and Parameter Reduction for Large-Scale Control Systems**. Mathematical Problems in Engineering, 2014:1–13, 2014.
- C. Himpe and M. Ohlberger. **Model Reduction for Complex Hyperbolic Networks**. In Proceedings of the ECC, pages 2739–2743, 2014.
- C. Himpe and M. Ohlberger. **A Unified Software Framework for Empirical Gramians**. Journal of Mathematics, 2013:1–6, 2013.