

The Distributed Empirical Cross Gramian

Christian Himpe (christian.himpe@uni-muenster.de)
Mario Ohlberger (mario.ohlberger@uni-muenster.de)
Stephan Rave (stephan.rave@uni-muenster.de)

WWU Münster
Institute for Computational and Applied Mathematics

Data-Driven Model Order Reduction and Machine Learning (MORML)
31.03.2016

- Cross Gramian** a tool for model reduction
- Empirical** based on experiments, data-driven
- Distributed** a parallel approach

Motivation

Application:

- Neurophysiological
- Parameter Inference (ie.: Network Connectivity)
- Functional Neuroimaging Data

Model:

- Nonlinear
- Parametric
- High-Dimensional State- **and** Parameter-Space

Notation (I)

Linear (Time-Invariant) System:

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

$$y(t) = Cx(t)$$

$$x(0) = x_0$$

- Input: $u(t)$
- State: $x(t)$
- Output: $y(t)$
- $M := \dim(u(t))$
- $N := \dim(x(t))$
- $Q := \dim(y(t))$
- System Matrix: $A \in \mathbb{R}^{N \times N}$
- Input Matrix: $B \in \mathbb{R}^{N \times M}$
- Output Matrix: $C \in \mathbb{R}^{Q \times N}$

Notation (II)

General Input-Output System:

$$\dot{x}(t) = f(x(t), u(t), \theta)$$

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

$$x(0) = x_0$$

- Parameter: $\theta \in \mathbb{R}^P$
- Vector Field: $f : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \rightarrow \mathbb{R}^N$
- Output Functional: $g : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}^P \rightarrow \mathbb{R}^Q$

Model Order Reduction (MOR)

Relevant Input-Output Mapping:

$$u \mapsto y$$

Actual Input-Output Mapping:

$$u \mapsto x \mapsto y$$

Reduction Rationale:

- $N \gg 1$
- $M \ll N$
- $Q \ll N$

Reduced Order Model (ROM)

Reduced Order System:

$$\dot{x}_r(t) = f_r(x_r(t), u(t), \theta)$$

$$y_r(t) = g_r(x_r(t), u(t), \theta)$$

$$x_r(0) = x_{0,r}$$

- $n := \dim(x_r(t))$
- $n \ll N$
- $\|y(\theta) - y_r(\theta)\| \ll 1$

Projection-Based Model Reduction

Galerkin Projection U :

$$\text{rank}(U) = n$$

$$UU = U$$

$$U^T U = \mathbb{1}$$

Projection-Based ROM:

$$\dot{x}_r(t) = U^T f(Ux_r(t), u(t), \theta)$$

$$y_r(t) = g(Ux_r(t), u(t), \theta)$$

$$x_r(0) = U^T x_0$$

Cross Gramian¹ [Fernando & Nicholson'83, Laub et al.'83]

For square systems ($M = Q$):

$$W_X := C \circ O$$

$$= \int_0^{\infty} e^{At} BC e^{At} dt$$

$$\stackrel{\text{part. Int.}}{\Rightarrow} AW_X + W_X A = -BC$$

Special properties:

- state-space symmetric systems
- (orthogonally) symmetric systems

¹The cross Gramian is not a Gramian matrix!

Direct Truncation

Hankel operator:

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

For symmetric systems:

$$\begin{aligned} H &= H^* \\ \Rightarrow \mathcal{O}\mathcal{C} &= (\mathcal{O}\mathcal{C})^* \\ \Rightarrow \sigma_i(H) &= \sqrt{\lambda_i((\mathcal{O}\mathcal{C})^*\mathcal{O}\mathcal{C})} \\ &= \sqrt{\lambda_i(\mathcal{O}\mathcal{C}\mathcal{O}\mathcal{C})} \\ &= \sqrt{\lambda_i(\mathcal{C}\mathcal{O}\mathcal{C}\mathcal{O})} \\ &= \sqrt{\lambda_i(W_X W_X)} = |\lambda_i(W_X)| \end{aligned}$$

Approximate Hankel Singular Value Decomposition:

$$W_X \stackrel{SVD}{=} UDV \rightarrow U = (U_1 \quad U_2)$$

Domains of Application

- Model Reduction [Sorensen & Antoulas'02]
- System Indices:
 - Minimality Test [Fernando & Nicholson'82]
 - System Gain [Gheondea & Ober'99]
 - Singularity Index [Mironovskii & Soloveva'15]
 - Cauchy Index [Fernando & Nicholson'83]
 - Minimum Information Loss Index [Fu et al.'09]
- Sensitivity Analysis [Streif et al.'06, Streif et al.'09]
- Parameter Identification [H. & Ohlberger'14]
- Decentralized Control [Moaveni & Khaki-Sedigh'06]
- and more . . .

Empirical Linear Cross Gramian² [Fernando & Nicholson'85, Shaker'12]

$$\begin{aligned}W_X &= \int_0^{\infty} e^{At} BC e^{At} dt \\ &= \int_0^{\infty} (e^{At} B)(e^{A^T t} C^T)^T dt \\ &= \int_0^{\infty} x(t)z(t)^T dt\end{aligned}$$

- $x(t)$ Impulse Response
- $z(t)$ Adjoint Impulse Response

²See also: [Moore'81, Lall et al.'99, Lall et al.'02]

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

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

- $x_i^m(t)$ – i -th state component for m -th perturbed input
- $y_m^j(t)$ – m -th output component for j -th perturbed initial state
- For linear systems: $\widehat{W}_X = W_X$
- For parametric systems³: $\widehat{W}_X = \sum_{\theta_i \in \Theta_h} \widehat{W}_X(\theta_i)$

³ Based on [Sun & Hahn'06]

Combined State and Parameter Reduction

Reduced Order System:

$$\dot{x}_r(t) = f_r(x_r(t), u(t), \theta_r)$$

$$y_r(t) = g_r(x_r(t), u(t), \theta_r)$$

$$x_r(0) = x_{0,r}$$

- $p := \dim(\theta_r)$
- $p \ll P$
- $\|y(\theta) - y_r(\theta_r)\| \ll 1$

Empirical Joint Gramian [Geffen et al.'08, H. & Ohlberger'14]

Augmented System:

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

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

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

Joint Gramian (Cross Gramian of the Augmented System):

$$W_J = \begin{pmatrix} W_X & W_M \\ W_m & W_\theta \end{pmatrix}$$

Uncontrollable Parameters:

$$W_m = 0$$

$$W_\theta = 0$$

Empirical Cross-Identifiability Gramian [H. & Ohlberger'14]

Schur-Complement of W_θ (Cross-Identifiability Gramian):

$$W_j := 0 - \frac{1}{2} W_M^T (W_X + W_X^T)^+ W_M$$

W_j encodes the “observability” of parameters.

Parameter Projection as Principal Components:

$$W_j \stackrel{SVD}{=} \Pi \Delta \Lambda \rightarrow \Pi = (\Pi_1 \quad \Pi_2)$$

Cross-Gramian-Based Combined Reduction

State-space projection:

$$W_X \stackrel{TSVD}{=} U_1 D_1 V_1$$

Parameter-space projection:

$$W_j \stackrel{TSVD}{=} \Pi_1 \Delta_1 \Lambda_1$$

Combined state and parameter ROM:

$$\dot{x}_r(t) = U_1^T f(U_1 x_r(t), u(t), \Pi_1 \theta_r),$$

$$y_r(t) = g(U_1 x_r(t), u(t), \Pi_1 \theta_r),$$

$$x_r(0) = U_1^T x_0,$$

$$\theta_r = \Pi_1^T \theta$$

Nonlinear Test System⁴

Weakly Nonlinear SISO System:

$$\dot{x}(t) = A \tanh(K(\theta)x(t)) + Bu(t)$$

$$y(t) = Cx(t)$$

$$x(0) = 0$$

System Dimensions:

- $\dim(u(t)) = 1$
- $\dim(x(t)) = 10^2$
- $\dim(y(t)) = 1$
- $\dim(\theta) = 10^2$

⁴ See: Hyperbolic Network Model [Quan et al.'01]

Empirical Gramians:

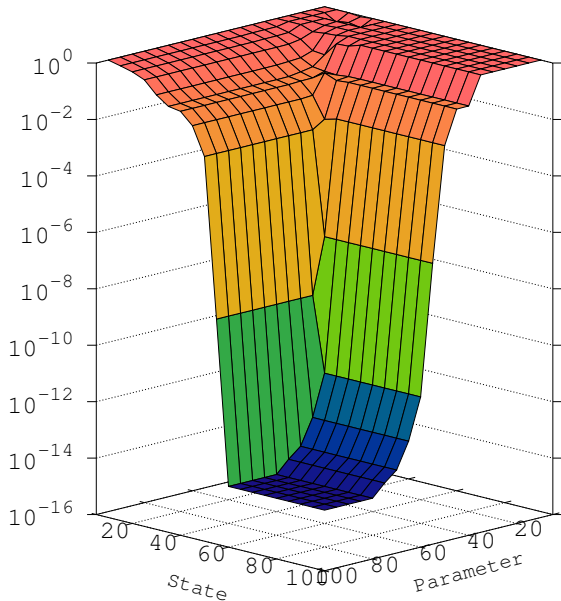
- Empirical Controllability Gramian
- Empirical Observability Gramian
- Empirical Linear Cross Gramian
- Empirical Cross Gramian
- Empirical Sensitivity Gramian
- Empirical Identifiability Gramian
- Empirical Joint Gramians

Features:

- Custom Solver Interface
- Non-Symmetric Cross Gramian
- Compatible with OCTAVE and MATLAB
- Vectorized and Parallelizable
- Open Source License (BSD 2-Clause)

More info at: gramian.de

Nonlinear Numerical Results ($L_2 \otimes L_2$ Error)



Distributed Empirical Cross Gramian

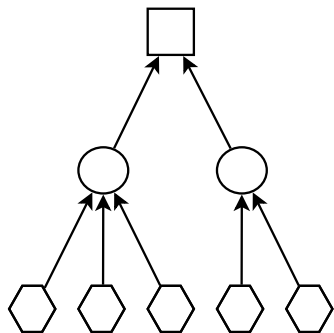
Empirical Cross Gramian:

$$\widehat{W}_X = \sum_{m=1}^M \int_0^{\infty} \Psi^m(t) dt \in \mathbb{R}^{N \times N}$$
$$\Psi_{ij}^m(t) := \langle x_i^m(t), y_m^j(t) \rangle$$

Empirical Cross Gramian's k -th column:

$$\Rightarrow \widehat{W}_{X,*k} = \sum_{m=1}^M \int_0^{\infty} \psi^{mk}(t) dt \in \mathbb{R}^{N \times 1}$$
$$\psi_i^{mk}(t) := \langle x_i^m(t), y_m^k(t) \rangle$$

HAPOD – Hierarchical Approximate POD




HAPOD:

- Rooted tree with ...
- nodes representing PODs of ...
- child nodes' PODs and ...
- leafs of snapshots.


Provides:

- Approximation error bound
- Mode bound

Check Out Stephan's HAPOD Poster



HAPOD – HIERARCHICAL APPROXIMATE POD
Stephan Rave and Christian Himpe



RWTH AACHEN UNIVERSITY
Institute for Data Science in Mechanical Engineering

Abstract

Proper orthogonal decomposition (POD) is a widely-used model order reduction technique for the computation of surrogate reduced state spaces from given solution snapshot data. However, performing a POD is often a computationally demanding task since the complexity scales quadratically to the number of snapshots. A ready-to-use solution to overcome this limitation is to compute, when and where available, PODs of subsets of the global snapshot set, and then to use the resulting POD modes as input for an additional POD. We formalize this approach as "hierarchical approximate POD" (HAPOD), allowing arbitrary trees of localized PODs, making HAPOD suitable for distributed, heterogeneous compute environments. As special cases of the HAPOD we consider a "distributed approximate POD" (DAPOD) and a "rolling approximate POD" (RAPOD), for which we present numerical results for a model reduction benchmark problem.

HAPOD – Hierarchical Approximate POD

HAPOD can be easily implemented on top of any existing POD code. Given a set \mathcal{J} of snapshot vectors, we assume that $\text{POD}(\mathcal{J}, \epsilon)$ computes the first N POD mode / singular value pairs such that:

$$X = \text{POD}(\mathcal{J}, \epsilon) = \arg \min_{\substack{U \in \mathbb{R}^{n \times N} \\ U^T U = I}} \left(\sum_{i=1}^N \sigma_i^2 \right) \leq \sqrt{\epsilon} + \epsilon$$

$$X = \text{POD}(\mathcal{J}, \epsilon) = \arg \min_{\substack{U \in \mathbb{R}^{n \times N} \\ U^T U = I}} \left(\sum_{i=1}^N \sigma_i^2 \right) \leq \sqrt{\epsilon} + \epsilon$$

HAPOD input:

- \mathcal{J} : snapshot of vectors \mathcal{J}
- ϵ : accuracy of global modes
- μ : accuracy of child modes

Abstract Error Bound:

$$\left(\sum_{i=1}^N \sum_{j=1}^L \|\sigma_j - \mu_j\|^2 \right)^{1/2} \leq \sum_{i=1}^L \sum_{j=1}^N \frac{\sigma_j}{\sqrt{\mu_j}} \epsilon(\mu_j)$$

HAPOD output:

- \mathcal{J}_1 : $\text{POD}(\text{Input}, \epsilon(\mu_1))$ scaled by singular values
- \mathcal{J}_2 : $\text{POD}(\text{Input}, \epsilon(\mu_2))$ scaled by singular values
- μ : $\text{POD}(\text{Input}, \epsilon(\mu))$

Abstract Mode Bound:

$$\|\text{HAPOD}(\mathcal{J}, \mu, \epsilon)\| \leq \text{POD} \left(\mu, \sqrt{\epsilon} \left(\sum_{i=1}^{L-1} \sum_{j=1}^N \frac{\sigma_j}{\sqrt{\mu_j}} \right) \right)$$

Choice of Error Tolerances:

$$\epsilon(\mu_1) = \frac{\sqrt{\epsilon}}{\sqrt{L-1}}, \quad \epsilon(\mu_2) = \frac{\sqrt{\epsilon}}{\sqrt{L-1}} \mu^{-1}, \quad \epsilon(\mu) = \frac{\sqrt{\epsilon}}{\sqrt{L-1}} \mu^{-1}$$

Error Bound:

$$\left(\sum_{i=1}^N \sum_{j=1}^L \|\sigma_j - \mu_j\|^2 \right)^{1/2} \leq \epsilon$$

Mode Bound:

$$\|\text{HAPOD}(\mathcal{J}, \mu, \epsilon)\| \leq \text{POD}(\mathcal{J}, (1 - \mu) \sqrt{\epsilon})$$

Notation:

- \mathcal{J} : Snapshot matrix
- σ_i : SVD values
- μ_j : SVD values
- ϵ : Error tolerance
- μ : Accuracy of child modes
- μ_j : Accuracy of child modes
- $\epsilon(\mu_j)$: Accuracy of child modes
- $\epsilon(\mu)$: Accuracy of child modes
- $\epsilon(\mu_1)$: Accuracy of child modes
- $\epsilon(\mu_2)$: Accuracy of child modes

DAPOD – Distributed Approximate POD



Choice of Error Tolerances:

$$\epsilon(\mu_1) = \frac{\sqrt{\epsilon}}{\sqrt{L-1}}, \quad \epsilon(\mu_2) = \mu^{-1}, \quad \epsilon(\mu) = \mu^{-1}$$

RAPOD – Rolling Approximate POD



Choice of Error Tolerances:

$$\epsilon(\mu_1) = \frac{\sqrt{\epsilon}}{\sqrt{L-1}} (1 - \mu)^{-1}, \quad \epsilon(\mu_2) = \frac{\sqrt{\epsilon}}{\sqrt{L-1}} \mu^{-1}, \quad \epsilon(\mu) = \mu^{-1}$$

Benchmark Problem

Synthetic Parameter Model (See <https://www.rwth-aachen.de/en/institute-for-data-science-in-mechanical-engineering/our-work/our-works>)

- Linear time-invariant
- Offline parameters
- Single input
- $\sigma^2 = 1$
- $\mu \in (0, 1)$
- $\mu \in (0, 1)$
- Impacts 80%
- Space = $\{0, 1\} \times \{0, 1, \dots, 8, 9, 1, \dots, 8, 9\}$
- Time = $\{0, 1, 2\} \times \{0, 1\}$

Get the Code



<https://github.com/stephanrave/haPod>

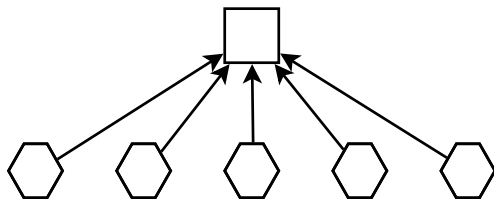
DAPOD Numerical Experiment



RAPOD Numerical Experiment



DAPOD – Distributed Approximate POD



DAPOD:

- Flat tree
- Tall and skinny partitioning

Distributed Direct Truncation

Distributed Empirical Cross Gramian:

$$\widetilde{W}_X = \{\widehat{W}_{X,*K_i} \mid K_i \subset \mathbb{N}, K_i \cap K_j = \emptyset, \bigcup_i K_i = 1 \dots N\}$$

Distributed Approximate POD:

$$\text{DAPOD}(\widetilde{W}_X)$$

Linear Model Reduction Benchmark⁵

Affine Parametric SISO System:

$$\dot{x}(t) = (A_0 + A_\theta\theta)x(t) + Bu(t)$$

$$y(t) = Cx(t)$$

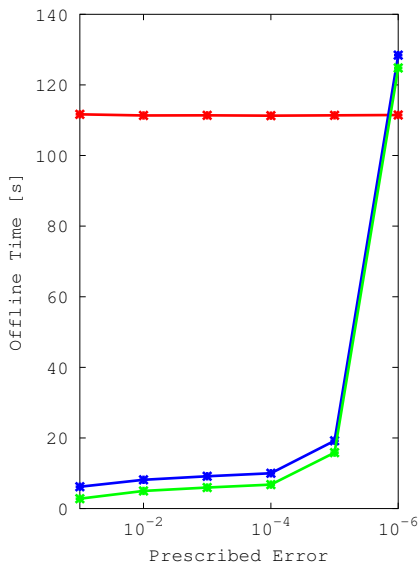
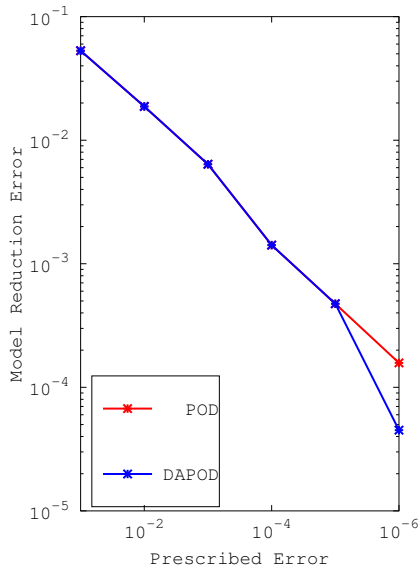
$$x(0) = 0$$

System Dimensions:

- $\dim(u(t)) = 1$
- $\dim(x(t)) = 10^4$
- $\dim(y(t)) = 1$
- $\dim(\theta) = 1$

⁵See: MORwiki modelreduction.org/index.php/Synthetic_parametric_model

Linear Numerical Results (L_2 Error & POD Time)



3 Levels:

- Distributed Memory:
 - Distributed Empirical **Cross** Gramian
- Explicit Shared Memory:
 - “Observability”-Snapshots
- Implicit Shared Memory / SIMD / GPU Offload:
 - “Gramian” Computation

- **Non-Symmetric Cross Gramian**⁶
- Distributed Empirical **Joint Gramian**
- **Dynamic Mode Decomposition**⁷ Hyperreduction
- **Balanced Gains**⁸

⁶We have a preprint: [H. & Ohlberger'15]

⁷[Proctor et al.'15]

⁸[Davidson'86]

Summary:

- Combined State and Parameter Reduction
for Nonlinear Parametric Systems
- Distributed Empirical Cross Gramian
utilizing the DAPOD

wwwmath.uni-muenster.de/u/himpe

Thanks!