MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC

COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

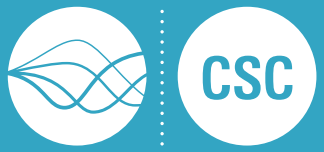
MOR Software

(Co-Developed in the CSC Group)

C. Himpe, P. Mlinarić, J. Saak, S.W.R. Werner

CSC Seminar

2021–05–18



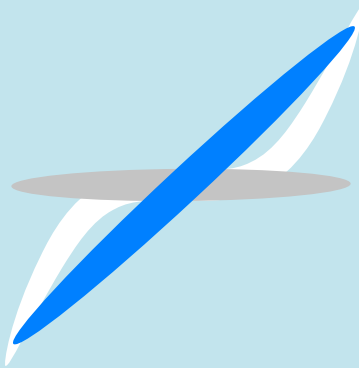
Overview

1. `emgr`
2. `hapod`
3. `morgen`
4. M-M.E.S.S.
5. MORLAB
6. SOLBT
7. SOMDDPA
8. pyMOR

emgr

emgr – EMpirical GRamian Framework

- Authors: C. Himpe
- Version: 5.9 (2021-01-21)
- License: BSD-2-Clause
- Website: https://gramian.de



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

- Language: MATLAB & Octave (600 LoC)
- Type: back-end library
- Started: 2012
- Paper: C. Himpe: **emgr – The Empirical Gramian Framework**; Algorithms 11(7): 91, 2018. doi:10.3390/a11070091

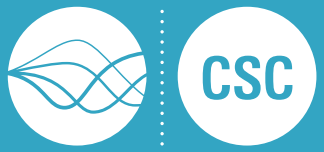
emgr: Features

Empirical Gramians:

- Empirical Controllability Gramian
- Empirical Observability Gramian
- Empirical Cross Gramian
- Empirical Linear Cross Gramian
- Empirical Sensitivity Gramian (Parameter Controllability)
- Empirical Identifiability Gramian (Parameter Observability)
- Empirical Joint Gramian (Parameter Observability & State Minimality)

Advanced Features:

- Custom integrator / solver
- Custom inner product / kernel
- Partitioned cross Gramian / joint Gramian



emgr: Applications

- Model Reduction
  - Nonlinear Model Order Reduction
  - Parametric Model Order Reduction
  - Combined State and Parameter Reduction
  - Dynamic Mode Decomposition
- Decentralized Control
- Sensitivity Analysis
- Parameter Identification
- Nonlinearity Quantification
- Uncertainty Quantification
- Optimal Placement
- System Norms & Indices
- Matrix Equations
- Tau Functions

hapod

- Authors: C. Himpe & S. Rave
- Version: 3.2 (2021-05-05)
- License: BSD-2-Clause
- Website: https://git.io/hapod

$$X = U\Sigma\left(V\right)$$

- Language: MATLAB & Octave (300 LoC)
- Type: drop-in function
- Started: 2016
- Paper: C. Himpe, T. Leibner, S. Rave: **Hierarchical Approximate Proper Orthogonal Decomposition**; SIAM Journal on Scientific Computing 40(5): A3267–A3292, 2018. doi:10.1137/16M1085413

hapod: Features

Algorithm Features:

- Error-driven computation
- Column-wise partitioned data
- Incremental and distributed topology
- Rigorous POD-of-PODs
- Single-pass algorithm
- Minimal Communication
- Custom SVD back-ends
- MapReduce ready

BTW: Combined, `emgr` and `hapod` enable low-rank empirical cross Gramians.

hapod: Applications

Large-Scale ...

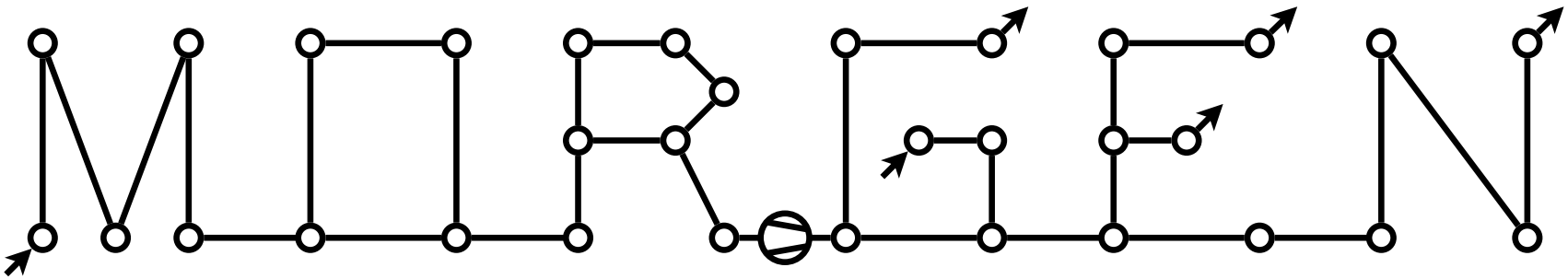
- Proper Orthogonal Decomposition (POD)
- Singular Value Decomposition (SVD)
- Principal Component Analysis (PCA)
- Empirical Orthogonal Functions (EOF)
- Karhunen-Loeve Transformation (KLT)
- Empirical Eigenfunctions (EEF)
- Unsupervised Learning (UL)
- Data Compression / Dimension Reduction (in general)

In ...

- Memory-limited settings (single board computers)
- Distributed memory settings (super computers)

morgen

- Authors: C. Himpe & S. Grundel
- Version: 0.99 (2021-04-12)
- License: BSD-2-Clause
- Website: https://git.io/morgen



- Language: MATLAB & Octave (6000 LoC)
- Type: test platform for gas, water, and district-heating networks
- Started: 2016
- Paper: C. Himpe, S. Grundel, P. Benner: **Model Order Reduction for Gas and Energy Networks**; arXiv math.OC: 2011.12099, 2021. arXiv:2011.12099

morgen: Features

Design Principles:

- Modular
- Configurable
- Extensible

Modules:

2 Models (discretizations)

4 Solvers (time steppers)

23 Reductors (structured & projection-based, `emgr` back-end included)

>20 Networks (topologies and scenarios)

>20 Tests (simulation and reduction)

morgen: Applications

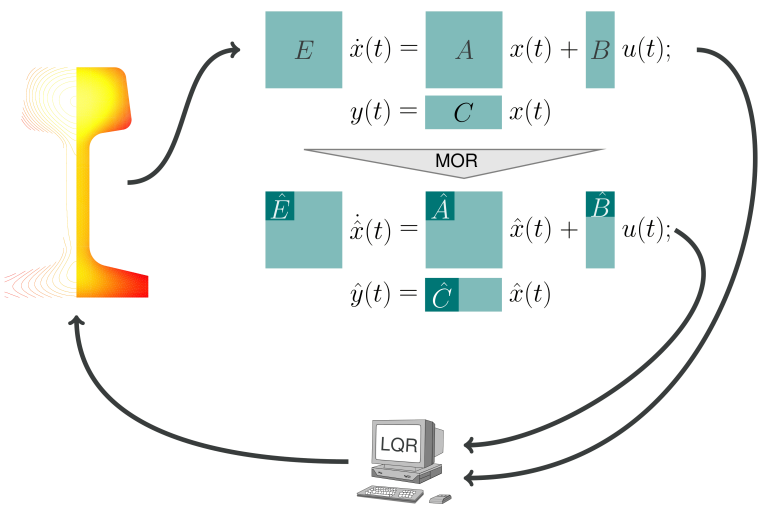
- Testing model-solver-reductor combinations
- Comparing models, solvers and reductors
- Benchmarking reductors
- Uncertainty quantification
- Prototyping algorithms

Gas networks are ...

- high-dimensional, hyperbolic, parametric, and nonlinear;
- and thus a real-life model reduction challenge.

M-M.E.S.S.

M-M.E.S.S. — Matrix Equation Sparse Solvers



- All equations are associated to a possibly abstract **standard state-space system** $\boldsymbol{\Sigma(\mathbf{E}; \mathbf{A}, \mathbf{B}, \mathbf{C})}$ or its dual.
- Proper differential-algebraic equations (DAE) or second-order (SO) systems use **implicit index-reduction** or **implicit linearization** to achieve this.
- The transformed implicit systems are (preferably) never formed explicitly.
- Operations with systems are abstracted into **user-supplied functions sets** (USFS):

USFS	`default`	`so_1 / so_2`	`dae_1`	`dae_2`	`dae_1/2/3_so`
System	standard / generalized state-space form	second-order 1st/2nd companion form	index-1 DAE	index-2 Stokes-type DAEs	second-order index-1/2/3 using companion form
Demos	FDM, Rail	TripleChain	DAE1 (BIPS Power-systems model)	DAE2 Stokes, Kárman vortex shedding	constrained TripleChain

$$0 = \mathbf{A}^\mathsf{T}\mathbf{X} + \mathbf{X}\mathbf{A} + \mathbf{C}^\mathsf{T}\mathbf{C} - \mathbf{X}\mathbf{B}\mathbf{B}^\mathsf{T}\mathbf{X}$$

$$0 = \mathbf{A}^\mathsf{T}\mathbf{X}\mathbf{E} + \mathbf{E}^\mathsf{T}\mathbf{X}\mathbf{A} + \mathbf{C}^\mathsf{T}\mathbf{C}$$

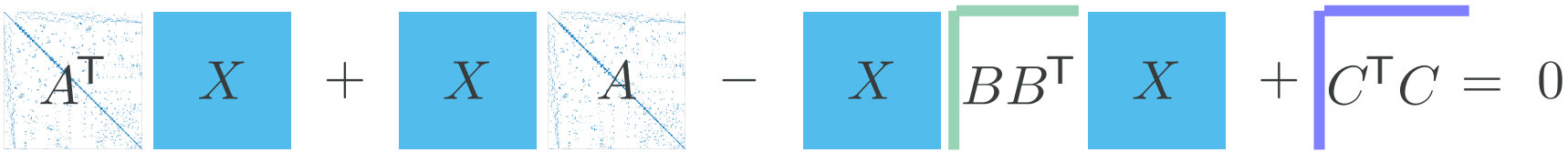
$$0 = \mathbf{A}^\mathsf{T}\mathbf{X}\mathbf{E} + \mathbf{E}^\mathsf{T}\mathbf{X}\mathbf{A} + \mathbf{C}^\mathsf{T}\mathbf{C} - \mathbf{E}^\mathsf{T}\mathbf{X}\left(\mathbf{B}_1\mathbf{B}_1^\mathsf{T} - \mathbf{B}_2\mathbf{B}_2^\mathsf{T}\right)\mathbf{X}\mathbf{E}$$

$$\mathbf{E}^\mathsf{T}\dot{\mathbf{X}}(t)\mathbf{E} = \mathbf{A}^\mathsf{T}\mathbf{X}(t)\mathbf{E} + \mathbf{E}^\mathsf{T}\mathbf{X}(t)\mathbf{A} + \mathbf{C}^\mathsf{T}\mathbf{C} - \mathbf{E}^\mathsf{T}\mathbf{X}(t)\mathbf{B}\mathbf{B}^\mathsf{T}\mathbf{X}(t)\mathbf{E}$$

$$\mathbf{E}(t)^\mathsf{T}\dot{\mathbf{X}}(t)\mathbf{E}(t) = \left(\mathbf{A}(t) + \dot{\mathbf{E}}(t)\right)^\mathsf{T}\mathbf{X}(t)\mathbf{E}(t) + \mathbf{E}(t)^\mathsf{T}\mathbf{X}(t)\left(\mathbf{A}(t) + \dot{\mathbf{E}}(t)\right) + \mathbf{C}(t)^\mathsf{T}\mathbf{C}(t) - \mathbf{E}(t)^\mathsf{T}\mathbf{X}(t)\mathbf{B}(t)\mathbf{B}(t)^\mathsf{T}\mathbf{X}(t)\mathbf{E}(t)$$

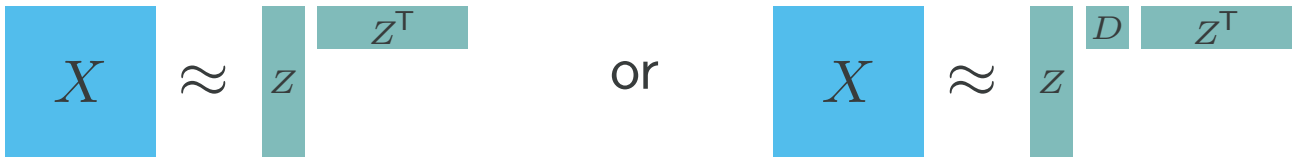
$$0 = \mathbf{A}^\mathsf{T}\mathbf{X}\mathbf{E} + \mathbf{E}^\mathsf{T}\mathbf{X}\mathbf{A} + \sum_k \mathbf{N}_k^\mathsf{T}\mathbf{X}\mathbf{N}_k + \mathbf{C}^\mathsf{T}\mathbf{C}$$

Algebraic Riccati equations $\mathcal{R}(X) = 0$:



where $A \in \mathbb{R}^{n\times n}$ (sparse), $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{n\times p}$ with $m, p \ll n$ and n "large".

Common solution paradigm: compute low-rank approximation:



Balanced Truncation

- `mess_balanced_truncation`
- `mess_balanced_truncation_bilinear`
- BT Demos in `Rail` and `FDM` folders

Balanced Truncation for structured (DAE) systems

- Second order: BT Demos in `Triplechain` folder
- First order DAEs: BT Demos in `DAE_1` and `DAE_2` folders
- Second order DAEs: BT Demos in `DAE_2_SO` and `DAE_3_SO` folders (`DAE_1_SO` following as soon as moderate size model is found)

Interpolatory MOR (theoretically all supported system classes)

- `mess_tangential_irka`

For Riccati-based BT methods: see ModRed 2019 book chapter

M-M.E.S.S. — Matrix Equation Sparse Solvers

M-M.E.S.S.-2.1 — facts & figures:

# contributors:	≈ 15
# code lines:	≈ 18 500 (≈ 26 000 including CI tests)
# comment and help lines:	≈ 12 000 (≈ 13 000 including CI tests)
history:	more than 10 years of active development
license:	BSD 2-Clause

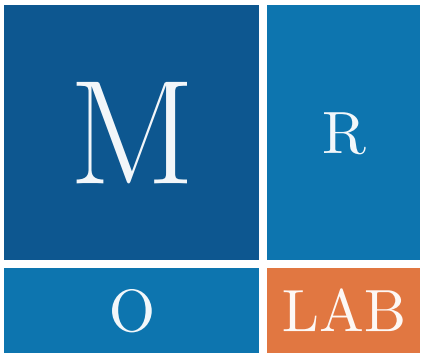
Benchmarks

Fishtail	BT, second-order, ≈780k DoFs
Adaptive Spindle support	BT, second-order index-1, ≈250k DoFs
Bilinear Rail	BT, bilinear, ≈ 5k DoFs

E-Mail	mess@mpi-magdeburg.mpg.de
WWW	https://www.mpi-magdeburg.mpg.de/projects/mess
MPI GITLab	https://gitlab.mpi-magdeburg.mpg.de/mess/mmess-releases
Github	https://github.com/mpimd-csc/mmess
Zenodo	https://doi.org/10.5281/zenodo.3368844

MORLAB

MORLAB (Model Order Reduction LABoratory)

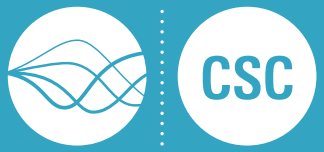


MORLAB (**M**odel **O**rder **R**eduction **LAB**oratory)

- (dense) model reduction in MATLAB/Octave
- based on spectral projection methods

https://www.mpi-magdeburg.mpg.de/projects/morlab

- Authors: P. Benner, S. W. R. Werner
- Version: 5.0 (2019-08-23)
- License: GNU AGPL v3.0+
- Started: 2006
- Dependencies: MATLAB ($\geq$ 2012b); Octave ($\geq$ 4.0.0)
- Paper: P. Benner, and S. W. R. Werner:
  MORLAB – The Model Order Reduction LABoratory,
  e-print 2002.12682, arXiv, 2020, https://arxiv.org/abs/2002.12682



MORLAB: Model reduction features

- modal truncation
- balanced truncation
- limited balanced truncation
- LQG balanced truncation
- $\mathcal{H}_\infty$ balanced truncation
- balanced stochastic truncation
- positive-real balanced truncation
- bounded-real balanced truncation
- Hankel-norm approximation
- second-order balanced truncation
- limited second-order balanced truncation

MORLAB: Model reduction features

- modal truncation
- balanced truncation
- limited balanced truncation
- LQG balanced truncation
- $\mathcal{H}_\infty$ balanced truncation
- balanced stochastic truncation
- positive-real balanced truncation
- bounded-real balanced truncation
- Hankel-norm approximation
- second-order balanced truncation
- limited second-order balanced truncation

Standard systems

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

Descriptor systems

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

Second-order systems

$$\begin{aligned} M\ddot{x}(t) &= -E\dot{x}(t) - Kx(t) + B_{\mathrm{u}}u(t), \\ y(t) &= C_{\mathrm{p}}x(t) + C_{\mathrm{v}}\dot{x}(t) + Du(t) \end{aligned}$$

Matrix equation solvers

Lyapunov equations

$$AXE^\mathsf{T} + EXA^\mathsf{T} + BB^\mathsf{T} = 0,$$
$$AXA^\mathsf{T} - EXE^\mathsf{T} + BB^\mathsf{T} = 0$$

Sylvester equations

$$AXE + FXB + C = 0$$

Bernoulli equations

$$A^\mathsf{T}XE + E^\mathsf{T}XA - E^\mathsf{T}XBB^\mathsf{T}XE = 0$$

Riccati equations

$$A^\mathsf{T}XE + E^\mathsf{T}XA + C^\mathsf{T}C - E^\mathsf{T}XBB^\mathsf{T}XE = 0$$

...

MORLAB: Further features

Matrix equation solvers

Lyapunov equations

$$AXE^\mathsf{T} + EXA^\mathsf{T} + BB^\mathsf{T} = 0,$$
$$AXA^\mathsf{T} - EXE^\mathsf{T} + BB^\mathsf{T} = 0$$

Sylvester equations

$$AXE + FXB + C = 0$$

Bernoulli equations

$$A^\mathsf{T}XE + E^\mathsf{T}XA - E^\mathsf{T}XBB^\mathsf{T}XE = 0$$

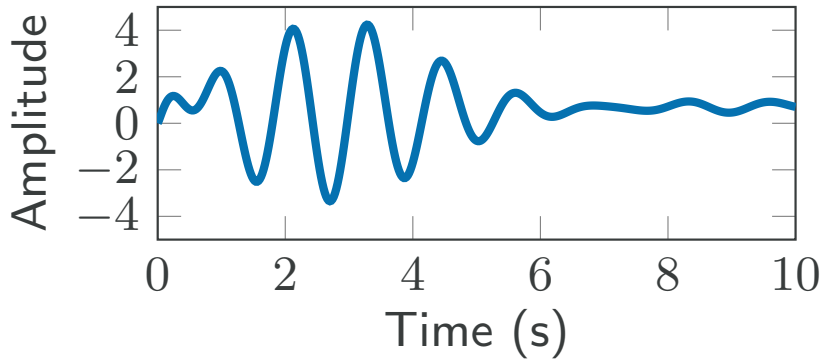
Riccati equations

$$A^\mathsf{T}XE + E^\mathsf{T}XA + C^\mathsf{T}C - E^\mathsf{T}XBB^\mathsf{T}XE = 0$$

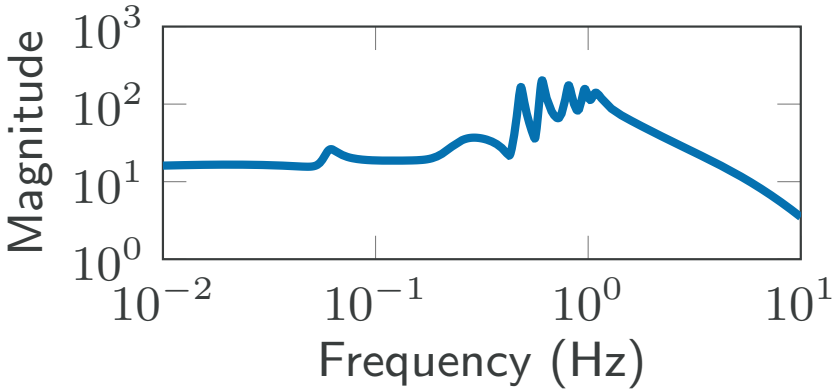
...

System evaluation

Time domain



Frequency domain



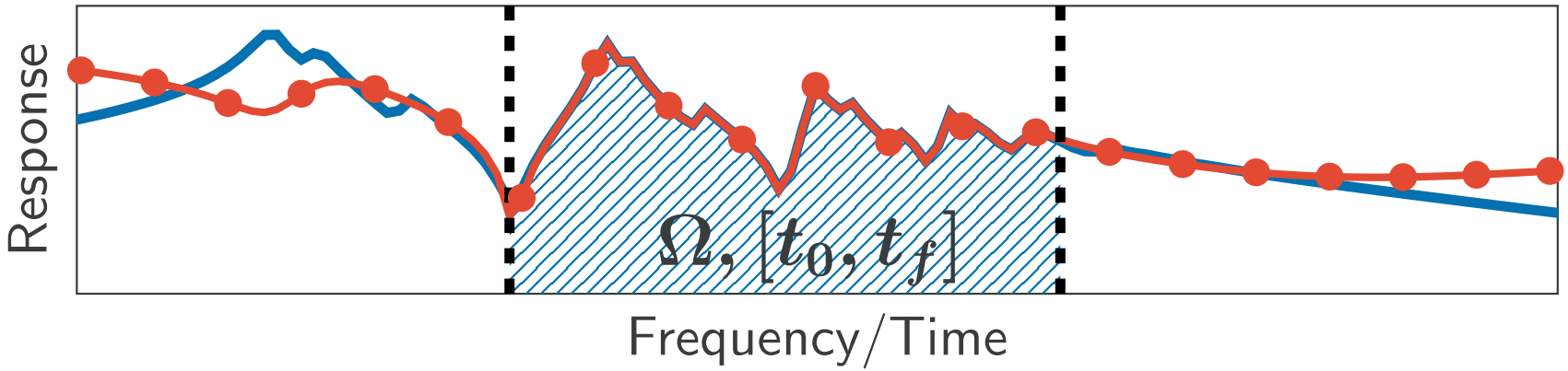
...

SOLBT

SOLBT (Limited BT for Sparse Second-Order Systems)

- Author: S. W. R. Werner
- Academic supervisor: P. Benner
- Version: 3.0 (2021-04-27)
- License: BSD-2-Clause
- Started: 2019
- Dependencies: MATLAB ($\geq$ 2012b) with Control System Toolbox; Octave ($\geq$ 4.0.0) with Control Package
- Paper: P. Benner, and S. W. R. Werner:
  Frequency- and time-limited balanced truncation for large-scale second-order systems,
  Linear Algebra Appl., 623:68–103, 2021, doi:10.1016/j.laa.2020.06.024

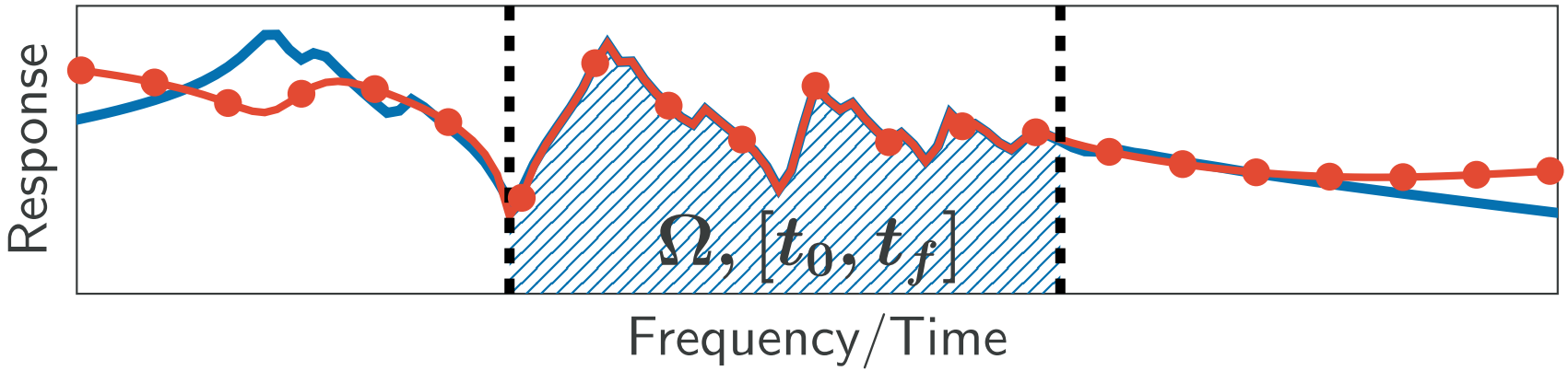
SOLBT: Features



- limited model reduction for large-scale sparse second-order systems

$$M\ddot{x}(t) + E\dot{x}(t) + Kx(t) = B_{\mathrm{u}}u(t),$$
$$y(t) = C_{\mathrm{p}}x(t) + C_{\mathrm{v}}\dot{x}(t)$$

SOLBT: Features



- limited model reduction for large-scale sparse second-order systems

$$M\ddot{x}(t) + E\dot{x}(t) + Kx(t) = B_{\mathrm{u}}u(t),$$
$$y(t) = C_{\mathrm{p}}x(t) + C_{\mathrm{v}}\dot{x}(t)$$

- projection-based solvers for limited Lyapunov equations

$$AX_{\Omega}E^{\mathsf{T}} + EX_{\Omega}A^{\mathsf{T}} + B_{\Omega}B^{\mathsf{T}} + BB_{\Omega}^{\mathsf{T}} = 0,$$
$$AX_{\Theta}E^{\mathsf{T}} + EX_{\Theta}A^{\mathsf{T}} + B_{t_{\mathrm{f}}}B_{t_{\mathrm{f}}}^{\mathsf{T}} - B_{t_0}B_{t_0}^{\mathsf{T}} = 0,$$

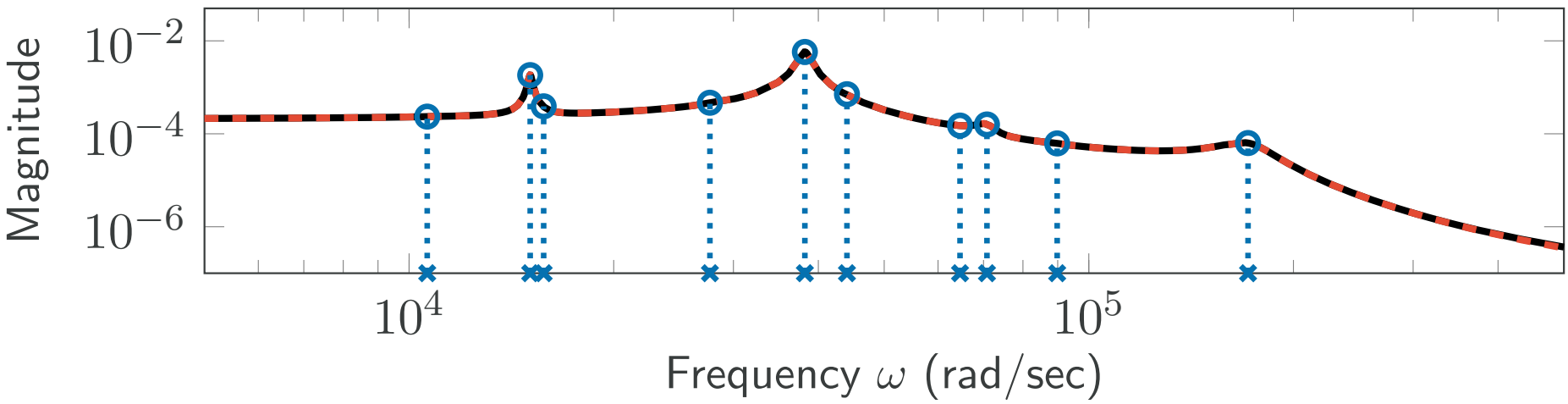
with $B_{\Omega} = \mathrm{Re}\big(\frac{i}{\pi}\ln((A + i\omega_2 E)(A + i\omega_1 E)^{-1})\big)B$, and $B_{t_{0,\mathrm{f}}} = e^{AE^{-1}t_{0,\mathrm{f}}}B$

SOMDDPA

SOMDDPA (Second-Order Modally-Damped Dominant Pole Algorithm)

- Author: S. W. R. Werner
- Academic supervisor: P. Benner
- Version: 2.0 (2021-04-27)
- License: BSD-2-Clause
- Started: 2019
- Dependencies: MATLAB ($\geq$ 2012b); Octave ($\geq$ 4.0.0)
- Paper: J. Saak, D. Siebelts, and S. W. R. Werner:
  A comparison of second-order model order reduction methods for an artificial fishtail,
  at-Automatisierungstechnik, 67 (2019), pp. 648–667, doi:10.1515/auto-2019-0027

SOMDDPA: Features

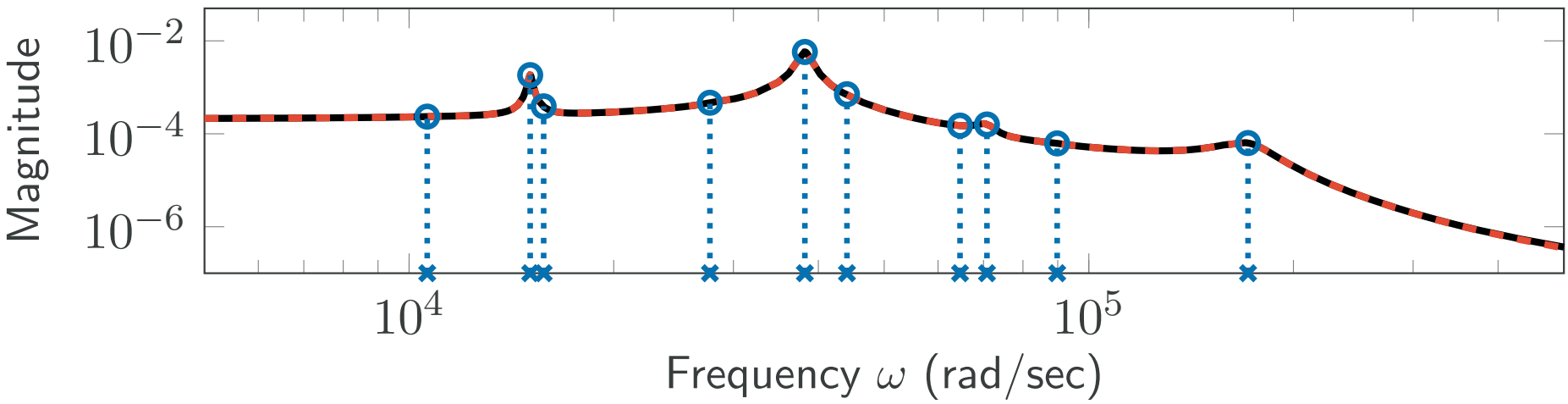


- dominant pole computation for modally-damped second-order systems

$$M\ddot{q}(t) + E\dot{q}(t) + Kq(t) = B_{\mathrm{u}},$$
$$y(t) = C_{\mathrm{p}}q(t),$$

with $M, E, K > 0$ and $EM^{-1}K = KM^{-1}E$

SOMDDPA: Features



- dominant pole computation for modally-damped second-order systems

$$M\ddot{q}(t) + E\dot{q}(t) + Kq(t) = B_{\mathrm{u}},$$
$$y(t) = C_{\mathrm{p}}q(t),$$

  with $M, E, K > 0$ and $EM^{-1}K = KM^{-1}E$

- eigenvalue computation based on Newton scheme and subspace acceleration

pyMOR

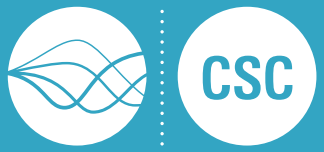
People

- maintainers: L. Balicki, R. Fritze, P. Mlinarić, S. Rave, F. Schindler
- # code contributors: 18

pyMOR — Model Order Reduction with Python

Info

- version: 2020.2.0 (2020-12-10)
- started: 2012
- license: BSD-2-Clause
- website: `pymor.org`



pyMOR — Model Order Reduction with Python

Info

- version: 2020.2.0 (2020-12-10)
- started: 2012
- license: BSD-2-Clause
- website: `pymor.org`

Size

- # Python code lines: 19 615 (28 731 with tests, demos, . . .)
- # documentation lines: 12 603 (17 739 with rst, tests, . . .)

Algorithms	Data structures
MOR methods	vectors and matrices

Algorithms	Data structures
MOR methods	vectors and matrices

Possible sources of high-dimensional data (backends)

- NumPy, SciPy
- PDE solver libraries (deal.II, DUNE, FEniCS, NGSolve, ...)
- proprietary software

Algorithms	Data structures
MOR methods	vectors and matrices

Possible sources of high-dimensional data (backends)

- NumPy, SciPy
- PDE solver libraries (deal.II, DUNE, FEniCS, NGSolve, ...)
- proprietary software

Fundamental pyMOR classes

1. `VectorSpace`
2. `VectorArray`
3. `Operator`

Three types of algorithms/implementations

1. abstract vectors, abstract operators ("good")
2. concrete vectors, abstract operators ("bad")
3. concrete vectors, concrete operators ("ugly")

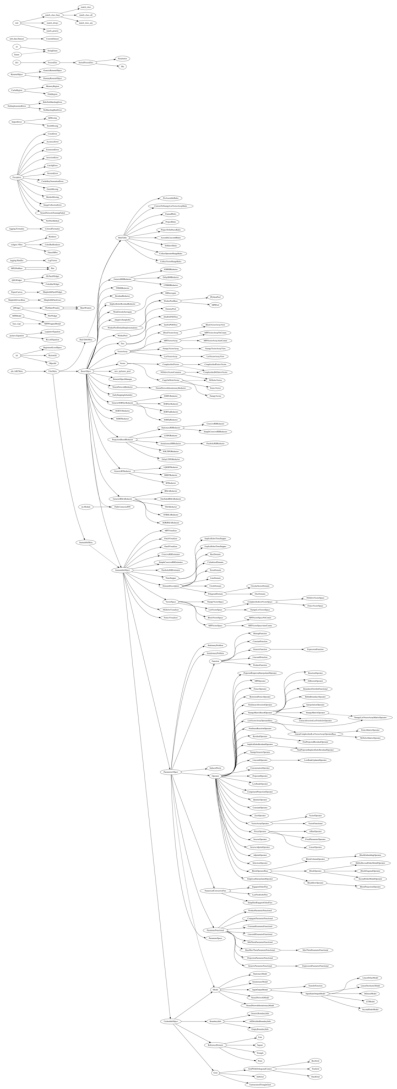
Three types of algorithms/implementations

1. abstract vectors, abstract operators ("good")
2. concrete vectors, abstract operators ("bad")
3. concrete vectors, concrete operators ("ugly")

Example (linear system solvers)

- "ugly": `scipy.linalg.solve`
- "bad": `scipy.sparse.linalg.lgmres`
- "good": `pymor.algorithms.genericsolvers.lgmres`

- pymor.algorithms
- pymor.analyticalproblems
- pymor.bindings
- pymor.core
- pymor.discretizers
- pymor.models
- pymor.operators
- pymor.parallel
- pymor.parameters
- pymor.reductors
- pymor.tools
- pymor.vectorarrays



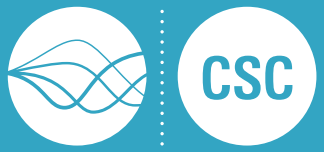
Algorithms and bindings

- (non)linear system solvers (SciPy, `lgmres`, `newton`)
- empirical interpolation (EIM, DEIM)
- eigenvalue solvers (`eigs`, SAMDP)
- Gram-Schmidt (bi)orthogonalization
- POD/SVD, HAPOD, randomized SVD
- matrix equation solvers (SciPy, Slycot, Py-M.E.S.S., LR-ADI, LR-RADI)
- time-stepping (explicit and implicit Euler)
- PDE solver library bindings (FEniCS, NGSolve)

MOR methods

- reduced basis methods for parametric (in)stationary problems
- system-theoretic methods:
  - BT, LQGBT, BRBT, SOBT(p/v)
  - interpolation, IRKA, one-sided IRKA, TSIA, TF-IRKA, SOR-IRKA
  - MT (2021.1)
- neural network method for parametric problems

Summary



Summary

https://www.mpi-magdeburg.mpg.de/csc/software

PS: More MOR software is in the MORwiki:

http://modelreduction.org/index.php/Comparison_of_Software
http://modelreduction.org/index.php/Further_Software

PPS: How to write your own (MOR) Software:

J. Fehr, J. Heiland, C. Himpe, J. Saak: **Best Practices for Replicability, Reproducibility and Reusability of Computer-Based Experiments Exemplified by Model Reduction Software**; AIMS Mathematics 1(3): 261–281, 2016. doi:bsb2

J. Fehr, C. Himpe, S. Rave, J. Saak: **Sustainable Research Software Hand-Over**; Journal of Open Research Software 9(1): 5, 2021. doi:10.5334/jors.307