Received: 7 October 2022 | Accepted: 4 January 2023

DOI: 10.1002/pamm.202200201

System Order Reduction for Gas and Energy Networks

Christian Himpe[1,*] and **Sara Grundel**[2]

[1] University of Münster, Orléans-Ring 10, 48149 Münster
[2] Max-Planck-Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg

No matter if natural gas, biogas or hydrogen, gas transport needs to be simulated ahead of dispatch to account for volatilities in demand and supply, so denominations are delivered reliably. The emancipation from producing countries alongside the renewable energy transition increases the number of scenarios to be simulated manifold, which in turn requires the acceleration of computational models to ensure completion of computer simulations before deadlines.

Gas is transported through a network of pipelines which can be mathematically modeled as large-scale nonlinear port-Hamiltonian input-output systems. To reduce computational complexity we propose unsupervised learning via synthetic data of the model's system-theoretic properties which then enables data-driven control or model reduction.

We summarize the aspects of nonlinear model reduction techniques adapted to gas pipeline networks and orchestrated to reduce the order of this challenging class of systems originating from hyperbolic systems of partial differential-algebraic equations, and demonstrate the applicability of our approach numerically.

1 Model Order Reduction for Gas and Energy Networks

The current challenges in gas transport, such as the green energy transition, sudden gas shortages, and hydrogen admixture, can only be overcome by elaborate gas network simulations. For short-term day-ahead planning, all simulations have to be completed before dispatch to ensure delivery of denominations. Hence, the faster simulations can be performed, the more scenarios can be tested before the daily dispatch deadline. To accelerate simulation of gas transport in pipeline networks for this many-query setting, we propose model reduction. Due to the manifold mathematical and numerical challenges of modeling and simulating gas networks, such as dimensionality, nonlinearity, parametricity, and hyperbolicity, we developed and documented a software platform named `morgen` – **M**odel **O**rder **R**eduction for **G**as and **E**nergy **N**etworks, to compare and test models, solvers, and model reduction ensembles heuristically [5], of which we describe and compare new features and extensions as the main contribution of this work. While the (about 30) testable projection-based model reduction algorithms vary in their practical computation, their commonly underlying system-theoretic operators are obtained in a data-driven manner, via synthetic-data unsupervised-learning. Since, this approach is not narrowed to the applicability of a single model order reduction algorithm for a class of models, but rather tuned towards ensembles of models, solvers and reductors, we use the term *system order reduction* here.

2 From PDAE Model to Input-Output System

Starting from the one-dimensional, isothermal Euler equations for a pipeline, and applying Kirchhoff laws for networks, results in a two dimensional system of partial differential-algebraic equations (PDAE). With boundary values constituting the controls and quantities of interest, employing analytic index reduction as well as spatial discretization and refinement [5, Sec. 2], yields an ordinary differential equation (ODE) input-output system:

$$\overbrace{\begin{pmatrix} E_p(\theta) & 0 \\ 0 & E_q \end{pmatrix}}^{E(\theta)} \overbrace{\begin{pmatrix} \dot{p}(t) \\ \dot{q}(t) \end{pmatrix}}^{\dot{x}(t)} = \overbrace{\begin{pmatrix} 0 & A_{pq} \\ \hat{A}_{qp} & 0 \end{pmatrix}}^{A} \overbrace{\begin{pmatrix} p(t) \\ q(t) \end{pmatrix}}^{x(t)} + \overbrace{\begin{pmatrix} 0 & B_{pd} \\ B_{qs} & 0 \end{pmatrix}}^{B} \overbrace{\begin{pmatrix} s_p \\ d_q \end{pmatrix}}^{u(t)} + \overbrace{\begin{pmatrix} 0 \\ F_C \end{pmatrix} + \begin{pmatrix} 0 \\ f_q(p,q,\theta) \end{pmatrix}}^{f(x(t),\theta)},$$

$$\underbrace{\begin{pmatrix} s_q(t) \\ d_p(t) \end{pmatrix}}_{y(t)} = \underbrace{\begin{pmatrix} 0 & C_{sq} \\ C_{dp} & 0 \end{pmatrix}}_{C} \underbrace{\begin{pmatrix} p(t) \\ q(t) \end{pmatrix}}_{x(t)}.$$

Here, the state consists of pressure and mass-flux $x(t) = \begin{pmatrix} p(t) & q(t) \end{pmatrix}^\mathsf{T}$, the input represents the pressure at inflow and mass-flux at outflow nodes $u(t) = \begin{pmatrix} s_p(t) & d_q(t) \end{pmatrix}^\mathsf{T}$, while the output is given by the mass-flux at inflows and pressure at demands $y(t) = \begin{pmatrix} s_q(t) & d_p(t) \end{pmatrix}^\mathsf{T}$. Furthermore, the parametric mass matrix is positive definite $E(\theta) > 0$, the system matrix is skew-symmetric $A = -A^\mathsf{T}$ (ignoring compressors), the output matrix relates to the input matrix by $C = B^\mathsf{T} Q$ (for a diagonal matrix $Q > 0$), while compressors are specified by F_C and the retarding gravity and friction forces are jointly given by f_q. Depending on the modeling of f (and ignoring compressors), this system is of port-Hamiltonian structure [5, Sec. 2.9].

* Corresponding author: e-mail christian.himpe@wwu.de

Model	Class	Port-Hamiltonian?	Reference
Midpoint Discretization	ODE	No	[5, Sec. 2.4.1]
Endpoint Discretization	ODE	Yes	[5, Sec. 2.4.2]

Solver	Type	Order	Properties	Reference
IMEX1	Implicit-Explicit	1	Minimum Numerical Complexity	[5, Sec. 5.3.3]
IMEX2	Implicit-Explicit	2	-	[5, Sec. 5.3.4]
CNAB2	Implicit-Explicit	2	-	[1, Sec. 3.2]
RK2hyp	Explicit	2	Maximum Hyperbolic Stability	[11] & Sec. 4.1
RK4hyp	Explicit	4	Low Dispersion, Low Dissipation	[6, Sec. 2.2]
RK4	Explicit	4	-	[5, Sec. 5.3.2]
Rosenbrock	Implicit	2	Time Adaptive	[5, Sec. 5.3.1]

Reductor	Variants	Projection	Reference
Structured Proper Orthogonal Decomposition (POD)	• reachability	Orthogonal	[5, Sec. 4.2]
Structured Goal-Oriented POD	• reachability	Orthogonal	[5, Sec. 4.5.1]
Structured Empirical Dominant Subspaces (DSPMR)	• reachability & observability • minimality • average minimality	Orthogonal	[5, Sec. 4.3]
Structured Modified POD	• reachability & observability • minimality • average minimality	Oblique	[13, Sec. III.D] & Sec. 4.2
Structured Balanced POD	• reachability & observability	Bi-Orthogonal	[5, Sec. 4.4.3]
Structured Empirical Balanced Truncation	• reachability & observability • minimality • average minimality	Bi-Orthogonal	[5, Sec. 4.4]
Structured Empirical Balanced Gains	• reachability & observability • minimality • average minimality	Bi-Orthogonal	[5, Sec 4.5]
Structured Dynamic Mode Decomposition Galerkin	• reachability	Orthogonal	[5, Sec. 4.6]

Table 1: Summary of available model, solver, and reductor implementations in `morgen` 1.2.

3 The `morgen` Platform

The `morgen` platform is an open-source software package, compatible with MATLAB and Octave, for testing, comparing and benchmarking gas network simulation stacks, particularly semi-discrete models, time stepping solvers, and model reductors, as well as ensembles of such models, solvers, and reductors.

`morgen` is modular, extensible, and configurable; this means the source code is organized into the modules: models, solvers, reductors (and a data module: networks), with each of these modules consisting of collections of independent implementations fulfilling a common interface. These interfaces allow users to extend the modules with new implementations (or data-sets), simply by adding their source or data files. Therein, not only the base functionality, but also the (user) modules can be configured externally. The currently included module contents are listed in Table 1.

`morgen` is designed for projection-based model reduction methods, hence a reductor consumes the model's spatial and temporal discretizations as well as a generic training scenario to compute a set of projection matrices. These projections are applied to the model components, once, before a simulation of a reduced order model (ROM), to the linear components E, A, B, C, and via composition to the nonlinear component f. For details on model reduction and in particular projection-based methods for gas network models, we refer to [5, Sec. 4].

To illustrate the extensibility via modules, we briefly lay out in the following how a new reductor can be added. A reductor is capsuled in a MATLAB function file (`.m`), where the main function has the signature:

```
function [proj,name] = my_reductor(solver,discrete,scenario,config)
```

It receives four arguments: the `solver` function (handle), and the structures `discrete`, `scenario`, `config`, while it returns a cell-array `proj` of projections, and a string `name` identifying the reductor. The `solver` function allows to compute output trajectories, given the three structure arguments `discrete`, `scenario`, and `config`, where the latter is supposed to be the sub-structure `config.solver` passed to reductor. The `discrete` structure, holding system dimensions and system components, can be manipulated, for example by changing the output function to an identity function to obtain state, instead of output trajectories from the solver, and the `scenario` structure, holding input functions, parameters and constants, can be manipulated to create perturbations to the loaded scenario. For details on these structures or components, see the documentation or included reductors.

A reductor returns structured projectors in a cell-array. For an orthogonal projection, this cell-array has two rows and one column, for an oblique or bi-orthogonal projection, this cell-array has two rows and two columns. The first row holds $\dim(p(t)) \times n$ matrices, and the second row holds $\dim(q(t)) \times n$-matrices, with n being the maximum reduced order, and the first column holding left-side projectors, while the second column holding right-side projectors, if different.

A boilerplate reductor is included under `reductors/template_reductor.m`, based on which, a custom reductor can then be implemented in the body of this function, that computes projection matrices from the `discrete` model and `scenario` data. Every function m-file in the `reductors` folder is assumed to be a valid reductor, further function files required by a reductor can be placed in sub-folder `private`.

4 New Features of `morgen` 1.2

In the recent version 1.2 of `morgen` two major features were added: A purely explicit solver with a similar space-time discretization as the implicit-explicit solver(s), and an oblique model reduction method. Furthermore, the model reduction back-end `emgr` – **Em**pirical **Gr**amian framework [3], computing the empirical system Gramians (via unsupervised learning from a machine learning point of view) used by all reductors, has been updated to the most recent version 5.99 [4].

4.1 Hyperbolic Runge-Kutta Methods

Following `morgen` 1.1 and [6], the solver library has been extended with explicit Runge-Kutta solvers, with the aim to find an efficient solver for model variants with more severe nonlinearities, for which the currently preferred implicit-explicit solver loses its advantages. Based on [11], explicit Runge-Kutta methods with maximum hyperbolic stability limit (hyperbolic stability radius) of number of stages minus one, for a given number of intermediate stages were added. These methods are of first order for an even number of stages, of second order for an odd number of stages, and are a generalization of the 5-stage method tested in [6] up to twelve stages. The following algorithm implements this family of methods for an ODE $\dot{x}(t) = f(t, x)$, and $x_k = x(t_k)$, $t_k = k\Delta t$:

$$\begin{aligned} \hat{x}_0 &:= x_k, \\ \hat{x}_i &:= x_k + a_i \Delta t f(t_k + a_i \Delta t, \hat{x}_{i-1}), \quad i = 1 \ldots S, \\ x_{k+1} &:= \hat{x}_S, \end{aligned}$$

and requires only two registers (vectors), one for x_k and one for $\hat{x}_i$, that may be overwritten in each iteration of i. This makes it a memory efficient low-storage method [9], of in fact minimum storage. The coefficients a_i are given in Table 2.

Stages	Order	Limit	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}
5	2	4	$1/4$	$1/6$	$3/8$	$1/2$	1							
6	1	5	$1/5$	$4/35$	$7/25$	$4/13$	$13/25$	1						
7	2	6	$1/6$	$1/12$	$2/9$	$4/19$	$19/54$	$1/2$	1					
8	1	7	$1/7$	$4/63$	$9/49$	$2/13$	$13/49$	$8/25$	$25/49$	1				
9	2	8	$1/8$	$1/20$	$5/32$	$2/17$	$17/80$	$5/22$	$11/32$	$1/2$	1			
10	1	9	$1/9$	$4/99$	$11/81$	$4/43$	$43/243$	$6/35$	$7/25$	$40/123$	$41/81$	1		
11	2	10	$1/10$	$1/30$	$3/25$	$4/53$	$53/350$	$7/52$	$26/152$	$4/17$	$17/50$	$1/2$	1	
12	1	11	$1/11$	$4/143$	$13/121$	$1/16$	$16/121$	$16/147$	$21/121$	$28/155$	$31/121$	$20/61$	$61/121$	1

Table 2: Order, imaginary stability limit, and coefficients for the hyperbolic Runge-Kutta methods with five to twelve stages.

To test the capabilities of this solver, we repeat the experiment from [5, Sec. 6.2] (Yamal-Europe pipeline), yet only for the port-Hamiltonian endpoint discretization [5, Sec. 2.4.2], in combination with the 11-stage hyperbolic Runge-Kutta method, and the Galerkin reductors, see Fig. 1. The eleven-stage variant is the only tested explicit time stepping integrator producing workable simulations and reduced order models, with similar space-time discretization as the implicit-explicit methods. However, compared to the first-order implicit-explicit method, the model reduction errors are much higher, and the associated MORscores $\mu \in [0, 1]$ (see legend), a benchmark score, which assigns a higher score to faster decay of, and lower overall model reduction errors [5, Sec. 6.1], are much lower. Practically, even this "best" explicit Runge-Kutta method is no match for the implicit-explicit method, not only in terms of model reduction error, but particularly in computational complexity. Nonetheless, the 11-stage hyperbolic Runge-Kutta method allows explicit integration of gas network models.

We note that Runge-Kutta solvers of this structure are limited to second order accuracy for nonlinear systems. Hence, the family of methods from [10] with third/fourth order methods for linear systems, would still be of second order for gas networks, while featuring a lower hyperbolic stability limit. These methods were also tested, but performed worse than the first/second order methods from [11]. Lastly, an alternative second-order implicit-explicit method, the Crank-Nicolson-Adams-Bashforth (CNAB) [1] solver, was tested, which even to a lesser degree did not reach the accuracy of the first order implicit-explicit solver.

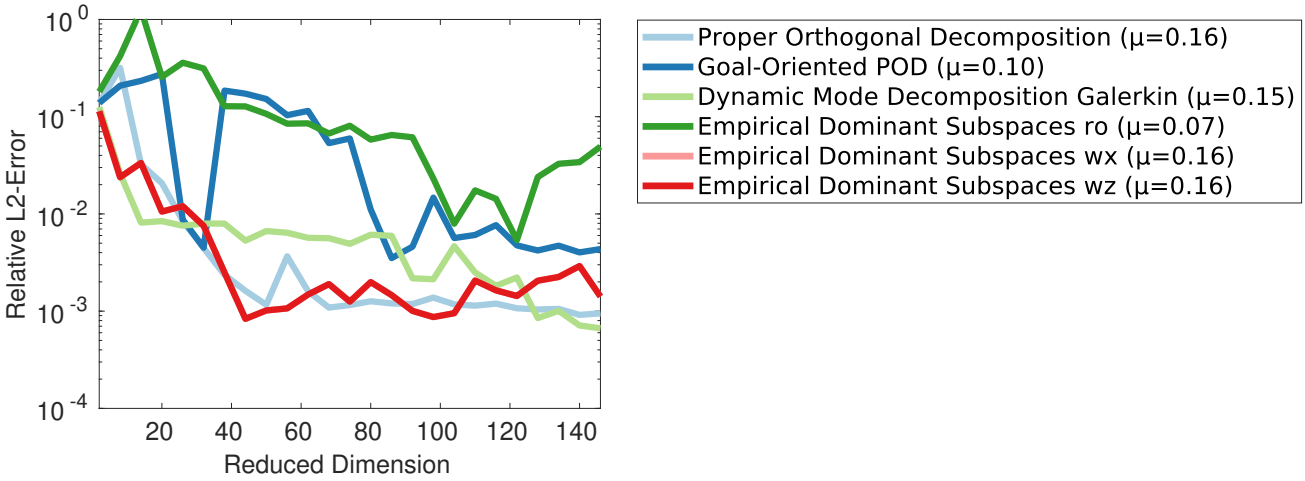

Fig. 1: Relative $L_2 \otimes L_2$ model reduction output error of the Yamal-Europe pipeline model/scenario for increasing reduced order using the explicit, 11-stage, hyperbolic Runge-Kutta solver. MORscores $0 \leq \mu < 1$ (higher is better) are given in the legend.

4.2 Structured Modified POD

In terms of reductors, variants of the *Modified Proper Orthogonal Decomposition* (Modified POD) were newly included. This model reduction algorithm from [13] was designed for non-normal systems, such as gas network models, which are derived from a hyperbolic partial differential equation and hence feature a non-normal system matrix (A). Furthermore, in [2], this method was also tested on a parabolic system. However, there is no error bound available, and particularly no stability guarantee, due to the oblique projection. Here a structured variant is employed, which means the pressure and mass-flux components are projected separately. Notationwise, we follow [5], where $_*$ means "for either": pressure $_p$ and mass-flux $_q$.

In the following three variants of modified POD are summarized.

4.2.1 Reachability- and observability-Gramian-based

The first modified POD variant is based on the empirical reachability and observability Gramians $\widehat{W}_R$ and $\widehat{W}_O$. Given the empirical pressure and mass-flux reachability and observability Gramians [5, Sec. 4.1.1, 4.1.2], their dominant subspaces (of same dimension) directly make up the reducing and reconstructing projections V_* and U_*:

$$\widehat{W}_{R,*} \overset{tSVD}{=} U_* D_* U_*^\intercal,$$
$$\widehat{W}_{O,*} \overset{tSVD}{=} V_*^\intercal D_* V_*.$$

4.2.2 Cross-Gramian-Based

The second modified POD variant is based on the empirical cross Gramians $\widehat{W}_X$ and thus minimality. For the empirical pressure and mass-flux cross Gramians [5, Sec. 4.1.3], the respective orthogonalized right and left dominant singular spaces are the reducing and reconstructing projections:

$$\widehat{W}_{X,*} \overset{tSVD}{=} U_* D_* V_*,$$

which directly extends to the empirical non-symmetric cross Gramians $\widehat{W}_{Z,*}$ [5, Sec. 4.1.4] (average minimality), the third modified POD variant. This method is also closely related to empirical *approximate balancing* [2, Sec. 3.2].

To evaluate the performance of this reductor in relation to others, we again repeat the pipeline experiment from [5, Sec. 6.2], but only compare the new (oblique) *structured modified POD* variants with the (Galerkin) structured empirical dominant subspaces [5, Sec. 4.3] and the (Petrov-Galerkin) structured empirical balanced truncation [5, Sec. 4.4] variants. We also limit this comparison to the port-Hamiltonian endpoint discretization [5, Sec. 2.4.2], and first order implicit-explicit solver [5, Sec. 5.3.3], as this model-solver combo seems to be theoretically and heuristically the best reducible. Practically, this experiment simulates a pipeline flow and thus allows an initial assessment of applicability.

In Fig. 2, the reduced-order-vs-error graph is plotted and the MORscores μ are listed inside the legend for this experiment. The MORscores show that the modified POD variants perform better than the balanced truncation variants. Also, modified POD reduced order models (ROMs) are more often stable compared to the balanced truncation ROMs. But still, the dominant subspaces variants achieve about twice the MORscore, while not producing any unstable ROM.

Thus, in line with [13], the modified POD (and thus approximate balancing) seems more suitable than balanced truncation for this non-normal system, yet it still does not outperform the dominant subspace method.

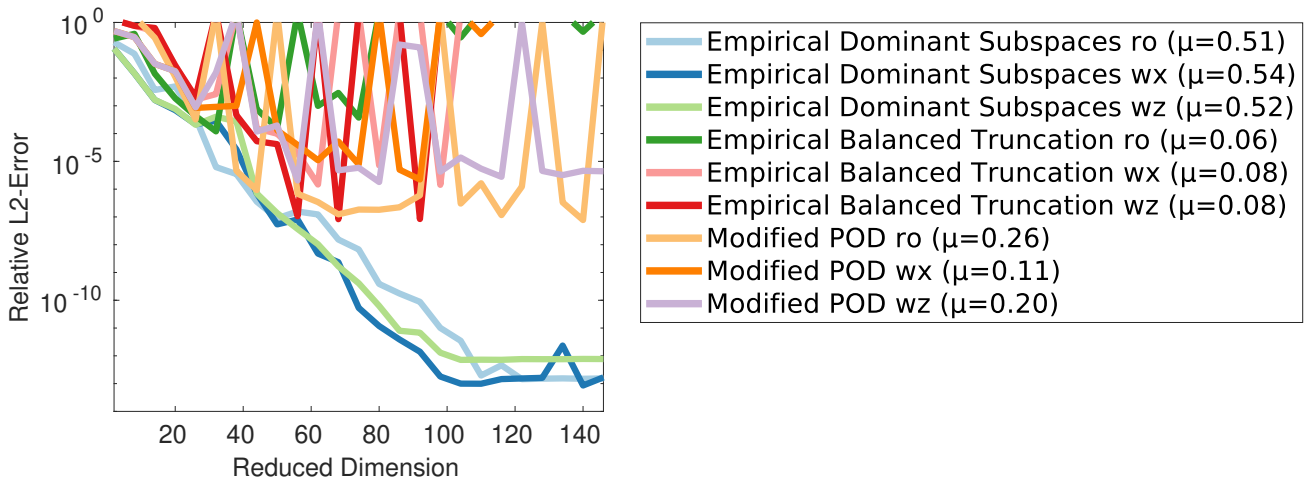

Fig. 2: Relative $L_2 \otimes L_2$ model reduction output error of the Yamal-Europe pipeline model/scenario for variants of dominant subspaces, balanced truncation, and modified POD. MORscores $0 \leq \mu < 1$ (higher is better) are given in the legend.

5 Chinese Gas Transport Network

Besides additions and improvements to the solver and reductor modules, a larger realistic network has been added, too, namely a part of the Chinese transport network for natural gas. This network's topology, pipeline specifications, and compressors locations were taken from [7, 8], and resulted in a data-set, included in `morgen`, comprising a network with a total length of about 8600km, 38 compressor stations, and five inflow as well as three outflow nodes. The network's expanse together with the number compressors constitute the complexities of this dataset.

The ensembles tested in the following consist of the port-Hamiltonian end-point discretization model, the first-order implicit-explicit solver, and each of the six Galerkin reductors: Proper orthogonal decomposition (POD) [5, Sec. 4.2], empirical dominant subspace projection model reduction (DSPMR) in three variants [5, Sec. 4.3], goal-oriented POD (GOPOD) [5, Sec. 4.5.1], and dynamic mode decomposition (DMD) Galerkin [5, Sec. 4.6]. As noted before, the newly added explicit hyperbolic Runge-Kutta solver is a potential alternative to the first-order implicit-explicit (IMEX) solver, but still not on par, hence the IMEX method is employed; also the non-Galerkin modified POD reductor outperforms the Petrov-Galerkin reductors, but not the Galerkin reductors, hence only the latter are compared.

In semi-discrete form, the network's associated input-output system has 11219 states and 8 boundary ports, while a time-step width of $40s$ is used. For the training, generic per port step perturbations over a time interval of $12h$ are used. Here, a longer training phase than in [5, Sec. 6.1] has to be prescribed, due to expanse of network, so input perturbations can arrive at the outputs. For testing, a randomly generated $24h$ scenario is generated and employed. The model reduction methods are compared via the MORscore, measuring performance in model reduction experiments based upon the area above the error graphs [2]. Practically, the numerical experiments were performed in MATLAB 2022a on an AMD Ryzen 5 4500U with 16GiB RAM.

The results plotted in Fig. 3 show the $L_2 \otimes L_2$ output model reduction error, as well as the associated MORscores μ in the legend. This experiment underscores previous results, where the empirical dominant subspaces method (reachability & observability variant) leads in reduction performance. The second best method, POD, achieves a similar MORscore, yet also produces some unstable reduced order models (ROMs), while the empirical dominant subspaces method only provides stable ROMs. The GOPOD reductor achieves a comparable model reduction error for the largest reduced order models, yet the error decays only from a certain reduced order. The minimality-, and average-minimality-based dominant subspaces methods, and the DMD Galerkin reductors achieve basically no reduction up to the maximum order ROM.

6 Summary and Outlook

This work documents the recent advancements of the `morgen` platform, and tests its capabilities on a large-scale network. The numerical results confirm previous findings for a heuristically best reductor: the structured, reachability-observability-based empirical dominant subspaces method, here with a realistic gas network topology.

Furthermore, this software platform can also extend to power networks. Given modeling from [12], two component power network models are available, which, when included into `morgen` together with a suitable solver, can use the simulation and model reduction machinery of the `morgen` platform.

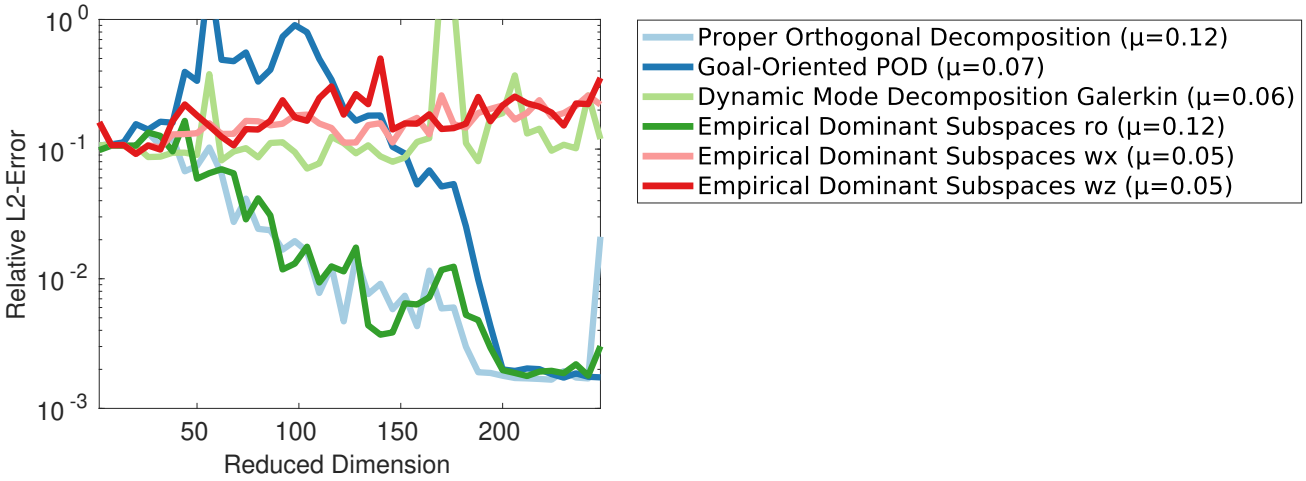

Fig. 3: Relative $L_2 \otimes L_2$ model reduction output error of the Chinese transport network using the Galerkin reductors. MORscores $0 \leq \mu < 1$ (higher is better) are given in the legend.

Lastly, download and documentation of `morgen` can be found at:

`https://git.io/morgen`

Code Availability

The source code of the numerical experiments is licensed under BSD-2-CLAUSE LICENSE, can be obtained from:

`doi:10.5281/zenodo.7157808`

and is authored by: C. HIMPE and S. GRUNDEL.

Acknowledgements Open access funding enabled and organized by Projekt DEAL.

References

[1] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995. doi:10.1137/0732037.

[2] C. Himpe. Comparing (empirical-Gramian-based) model order reduction algorithms. In P. Benner, T. Breiten, H. Faßbender, M. Hinze, T. Stykel, and R. Zimmermann, editors, *Model Reduction of Complex Dynamical Systems*, volume 171 of *International Series of Numerical Mathematics*, pages 141–164. Birkhäuser, Cham, 2021. doi:10.1007/978-3-030-72983-7_7.

[3] C. Himpe. emgr – EMpirical GRamian framework (version 5.99). https://gramian.de, 2022. doi:10.5281/zenodo.6457616.

[4] C. Himpe. emgr – EMpirical GRamian framework version 5.99. e-print 2209.03833, arXiv, 2022. URL: https://arxiv.org/abs/2209.03833.

[5] C. Himpe, S. Grundel, and P. Benner. Model order reduction for gas and energy networks. *Journal of Mathematics in Industry*, 11:13, 2021. doi:10.1186/s13362-021-00109-4.

[6] C. Himpe, S. Grundel, and P. Benner. Next-gen gas network simulation. In M. Ehrhardt, M. Günther, editors, *Progress in Industrial Mathematics at ECMI 2021*, volume 39 of *Mathematics in Industry*, pages 107–113, 2022. doi:10.1007/978-3-031-11818-0_15.

[7] L. Jin and A. K. Wojtanowicz. Optimization of large gas pipeline network in China – a feasibility study. In *Proceedings of the Canadian International Petroleum Conference*, page 174. OnePetro, 2008. doi:10.2118/2008-174.

[8] L. Jin and A. K. Wojtanowicz. Optimization of large gas pipeline network – a case study in China. *Journal of Canadian Petroleum Technology*, 49(4):36–43, 2010. doi:10.2118/136345-PA.

[9] D. I. Ketcheson. Runge-Kutta methods with minimum storage implementations. *Journal of Computational Physics*, 229(5):1763–1773, 2010. doi:10.1016/j.jcp.2009.11.006.

[10] I. P. E. Kinnmark and W. G. Gray. One step integration methods of third-fourth order accuracy with large hyperbolic stability limits. *Mathematics and Computers in Simulation*, 26(3):181–188, 1984. doi:10.1016/0378-4754(84)90056-9.

[11] I. P. E. Kinnmark and W. G. Gray. One step integration methods with maximum stability regions. *Mathematics and Computers in Simulation*, 26(2):87–92, 1984. doi:10.1016/0378-4754(84)90039-9.

[12] T. Nishikawa and A. E. Motter. Comparative analysis of existing models for power-grid synchronization. *New J. Phys.*, 17:015012, January 2015. doi:10.1088/1367-2630/17/1/015012.

[13] A. C. Or, J. L. Speyer, and J. Kim. Reduced balancing transformations for large nonnormal state-space systems. *J. Guid. Control Dyn.*, 35(1):129–137, 2012. doi:10.2514/1.53777.