

Next-Gen Gas Network Simulation



Christian Himpe, Sara Grundel, and Peter Benner

Abstract To overcome many-query optimization, control, or uncertainty quantification work loads in reliable gas and energy network operations, model order reduction is the mathematical technology of choice. To this end, we enhance the model, solver and reductor components of the `morgen` platform, introduced in Himpe et al. [J. Math. Ind. 11:13, 2021], and conclude with a mathematically, numerically and computationally favorable model-solver-reductor ensemble.

1 Model Order Reduction for Gas and Energy Networks

Computer-based simulation of gas transport in pipeline networks has been an industrial as well as academic field of interest since the earliest scientific computing systems [5]. Especially, the transient simulation of gas flow and the dynamic gas network behavior are the pinnacle discipline in this regard. The MATLAB-based `morgen`—Model Order Reduction for Gas and Energy Networks—platform¹ continues this research by providing a modular open-source software simulation stack for the comparison and benchmarking of models (discretizations), solvers (time steppers), and reductors (model reduction algorithms) [3]. Beyond selecting apposite simulator components or ranking model reduction methods, an overall goal is the acceleration of forward simulations, so that many-query tasks relying thereon, such as optimization, control or uncertainty quantification, benefit in terms

¹ See: <https://git.io/morgen>

C. Himpe (✉) · S. Grundel · P. Benner
Computational Methods in Systems and Control Theory Group, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany
e-mail: himpe@mpi-magdeburg.mpg.de; grundel@mpi-magdeburg.mpg.de;
benner@mpi-magdeburg.mpg.de

Table 1 Available models in `morgen` in version 1.1

Name	Identifier	port-Hamiltonian?	Reference
Midpoint discretization	<code>ode_mid</code>	No	[3, Sec. 2.4.1]
Endpoint discretization	<code>ode_end</code>	Yes	[3, Sec. 2.4.2]

Table 2 Available solvers in `morgen` in version 1.1

Name	Identifier	Comment	Reference
Adaptive second order Rosenbrock	<code>generic</code>	Uses <code>ode23s</code>	[3, Sec. 5.3.1]
First order implicit-explicit	<code>imex1</code>	Non-Runge-Kutta	[3, Sec. 5.3.3]
Second order implicit-explicit	<code>imex2</code>	Runge-Kutta	[3, Sec. 5.3.4]
Explicit fourth order Runge-Kutta	<code>rk4</code>		[3, Sec. 5.3.2]
Explicit second order Runge-Kutta	<code>rk2hyp</code>	Increased stability	[9]
Explicit fourth order Runge-Kutta	<code>rk4hyp</code>	Increased stability	[6]

of performance. In this work, we summarize and enhance the foundational work of [3] with additional details, and accompany version 1.1 of `morgen`.

1.1 Modules Overview

The `morgen` platform is organized into modules: *models*, *solvers*, *reductors*, *networks* and *tests*. The *networks* module holds topology and scenario data, and the *tests* module defines the simulation and model reduction experiments, thus, we summarize the currently available core modules: *models*, *solvers*, and *reductors*. The *models* module assembles a semi-discrete input-output system from a network topology. Currently, two spatially discrete models are included (Table 1). The *solvers* module computes a time-discrete output trajectory from a model and a scenario. Six solvers are provided in the current version (Table 2). The *reductors* module compresses a model given a solver and (generic training) scenario. All in all, 23 reductors organized in four classes are available (Table 3).

2 Enhanced Functionality

In this section, we discuss some of the new properties of the `morgen` 1.1 platform. Specifically, one aspect of each core module (*model*, *solver*, *reductor*) is addressed.

Table 3 Available reducers in `morgen` in version 1.1

Name	Identifier	Linear variant	Reference
Structured proper orthogonal decomposition	<code>pod_r</code>	–	[3, Sec. 4.2]
Structured empirical dominant subspaces	<code>eds_ro</code>	<code>eds_ro_l</code>	[3, Sec. 4.3]
Structured empirical dominant subspaces	<code>eds_wx</code>	<code>eds_wx_l</code>	[3, Sec. 4.3]
Structured empirical dominant subspaces	<code>eds_wz</code>	<code>eds_wz_l</code>	[3, Sec. 4.3]
Structured balanced POD	<code>bpod_ro</code>	<code>bpod_ro_l</code>	[3, Sec. 4.4.3]
Structured balanced truncation	<code>ebt_ro</code>	<code>ebt_ro_l</code>	[3, Sec. 4.4]
Structured balanced truncation	<code>ebt_wx</code>	<code>ebt_wx_l</code>	[3, Sec. 4.4]
Structured balanced truncation	<code>ebt_wz</code>	<code>ebt_wz_l</code>	[3, Sec. 4.4]
Structured goal-oriented POD	<code>gopod_r</code>	–	[3, Sec. 4.5.1]
Structured balanced gains	<code>ebg_ro</code>	<code>ebg_ro_l</code>	[3, Sec. 4.5]
Structured balanced gains	<code>ebg_wx</code>	<code>ebg_wx_l</code>	[3, Sec. 4.5]
Structured balanced gains	<code>ebg_wz</code>	<code>ebg_wz_l</code>	[3, Sec. 4.5]
Structured DMD Galerkin	<code>dmd_r</code>	–	[3, Sec. 4.6]

2.1 Gravity Term

One component of the gas pipeline model, particularly of the retarding forces in the mass-flux equation, is the gravity term, which accounts for increase or decrease in momentum due to an incline in a pipeline section. In [2], this gravity term is modeled in great detail, as it does not only consider a height difference between the pipe's end points, as `morgen` does, but also the height profile for the full run of the pipe (see [2, Fig. 11]). Both approaches are justified, depending on the aimed accuracy of the model, as discussed in [1]. Such pipeline height profiles can be included into `morgen` by supplying a pipe as sequence of virtual pipes, each connecting two subsequent local height extrema. In `morgen` 1.1, the gravity term is configurable so it is computable based on the dynamic pressure, static pressure or not at all, whereas the static gravity term, based on the steady-state was newly added.

2.2 Explicit Solvers

In [3], the classic explicit 4th order Runge-Kutta method `rk4` was tested, as it was employed in earlier works. Yet we found it to be *not* suitable for gas network simulations. In [4], an explicit Runge-Kutta method from [9, Sec. 4] was suggested for this application. The Butcher tableau for this explicit 5-stage, 2nd order low-storage scheme with increased stability, is given by:

0					
$\frac{1}{4}$	$\frac{1}{4}$				
$\frac{1}{6}$	0	$\frac{1}{6}$			
$\frac{3}{8}$	0	0	$\frac{3}{8}$		
$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	
	0	0	0	0	1

This additional solver `rk2hyp`, as well as a 4th order Runge-Kutta method with increased *hyperbolic stability limit* from [6, Sec. 4.1] (`rk4hyp`), were added to `morgen` 1.1 and tested against various test problems. Both increased-stability solvers allow larger time steps than `rk4`, specifically in conjunction with the `ode_end` model, but compared to the implicit-explicit solvers `imex1` and `imex2`, they are still not fully competitive. However, these explicit methods could be interesting for new implicit-explicit or predictor-corrector methods.

2.3 Gain Matching

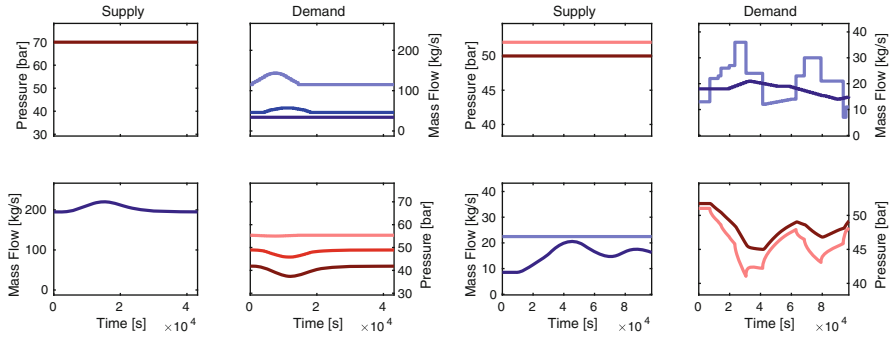
An important quality for certain applications of model reduction, such as electrical circuits, is the preservation of the steady-state gain (also known as DC gain), which is the output for zero frequency input. First, we clarify that we are not discussing the actual steady-state gain of the reduced order model, due to the centering around the steady-state and hence, the steady-state gain match [3, Sec. 3]. Yet, there can still be an output error for a constant input on top of the steady-state input, which is relevant due to the assumed low-frequency boundary values. Since there is an interpretation of gas networks as circuits [8], we consider this reduced model property, which induces two questions: How to compute the steady-state gain, and how to correct a gain mismatch? The former is answered by [10], stating that for a linear port-Hamiltonian model, with components as in [3, Sec. 2.9], the gain S is computable by:

$$S = CQ^{-1}B,$$

with input matrix B , output matrix C , and energy storage matrix Q . Since the models are nonlinear and do not have to be port-Hamiltonian, but comprise the same model components, the above formula can still be applied albeit yielding only an approximation. The per-port gain mismatch D_* is then computed by the difference of full and reduced-order model (reduced-order quantities are denoted by \cdot_r) gain:

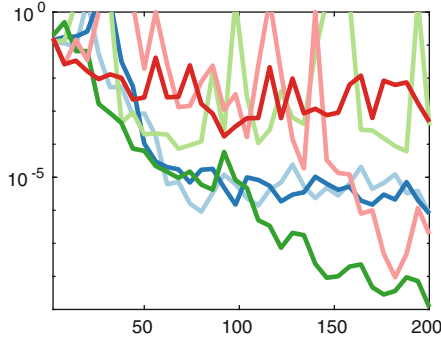
$$D_* := (CQ^{-1}B) - (C_rQ_r^{-1}B_r),$$

which can then be used to correct the reduced-order model gain by adding D_* as a feedthrough matrix to the output function, as described in the gain matching procedure in [7]. We included this approximate gain matching test to `morgen` 1.1.

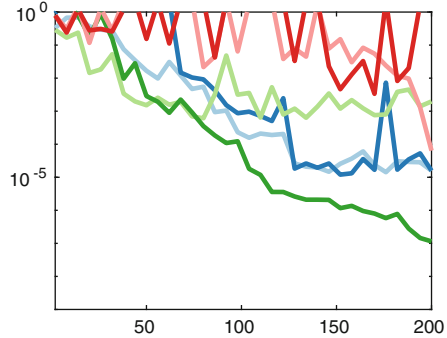


(a) Hypothetical network's test scenario.

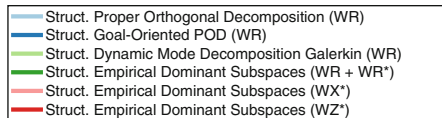
(b) Actual network's test scenario.



(c) Relative $L_2 \otimes L_2$ error between ROM and FOM for the hypothetical network.



(d) Relative $L_2 \otimes L_2$ error between ROM and FOM for the actual network.



(e) Common legend for the model reduction error plots.

Reductor	MORSCORE	Avg. Gain Error
pod_r	0.27	$6 \cdot 10^{-6}$
gopod_r	0.26	$6 \cdot 10^{-6}$
dmd_r	0.18	$8 \cdot 10^{-6}$
eds_ro_1	0.30	$8 \cdot 10^{-6}$
eds_wx_1	0.18	$8 \cdot 10^{-6}$
eds_wz_1	0.15	$8 \cdot 10^{-6}$

(f) MORSCORE $\mu(200, \epsilon_{\text{mach}(16)}) \in (0, 1]$ in the $L_2 \otimes L_2$ error norm (higher means more accurate ROM), and mean steady-state gain error for the hypothetical network.

Reductor	MORSCORE	Avg. Gain Error
pod_r	0.19	$2 \cdot 10^{-5}$
gopod_r	0.15	$1 \cdot 10^{-5}$
dmd_r	0.15	$2 \cdot 10^{-5}$
eds_ro_1	0.24	$2 \cdot 10^{-5}$
eds_wx_1	0.04	$2 \cdot 10^{-5}$
eds_wz_1	0.03	$2 \cdot 10^{-5}$

(g) MORSCORE $\mu(200, \epsilon_{\text{mach}(16)}) \in (0, 1]$ in the $L_2 \otimes L_2$ error norm (higher means more accurate ROM), and mean steady-state gain error for the actual network.

Fig. 1 Visualization of the test scenario, model reduction errors between FOM (full-order model) and ROM (reduced-order model), MORSCORE, and gain errors of the tested ROMs for the hypothetical network [5, Part 2] (left side) and actual network [5, Part 3] (right side). Computed with MATLAB 2021a. See [3, Sec. 6] for a description of the plot presentation

The gain correction was tested with all reducers (Table 3). For all reducers, the correction was about the level of 10^{-5} , see Tables f and g in Fig. 1, except for the `bpod_ro` method, for which the gain correction fully deteriorates the reduced model. Thus, the improvement of reduced-order models is small at best. This is not unexpected, considering the gas network model is hyperbolic: A single pipeline, or more generally an input-output system based on a first-order hyperbolic partial differential equation, has the transport property which expresses as a delay in observable outputs of controllable inputs. Hence, an immediate effect of inputs to outputs (circumventing the system dynamics), i.e. by a feedthrough term, is typically not needed.

3 Numerical Experiments

We extend the numerical experiments in [3], by reimplementing the results from [5], specifically, we test the hypothetical network [5, Part 2], and the actual network [5, Part 3], which are both tree networks, on their associated scenarios.

Six structured empirical-Gramian-based Galerkin reducers are tested on the port-Hamiltonian endpoint model and the first order implicit-explicit solver. The results are presented in Fig. 1. In line with other experiments, the `eds_ro_1` reducer yields the most accurate results.

4 Next-Gen Gas Network Simulation

For the newly tested features we conclude that currently, explicit solvers do not seem a viable option to simulate gas networks, while gain matching offers only minor accuracy improvements; yet, the new static gravity term is more robust with respect to model reduction and is henceforth the default setting in `morgen`.

Overall, based on the comparisons in [3] and this work's numerical results, we currently recommend a port-Hamiltonian model, an implicit-explicit solver, and a Galerkin reducer. Thus, the endpoint discretization, first order IMEX time stepper, and the structured empirical dominant subspaces reducer make a promising model-solver-reducer ensemble for the next generation of transient gas network simulators. Future extensions of the `morgen` platform will refine this recommendation.

References

1. G. Bachman and M. Goodreau. Less is more accuracy versus precision in modeling. In PSIG Annual Meeting 2000, pp. PSIG-0009, 2000.

2. M. Behbahani-Nejad, A. Bermúdez, and M. Shabani. Finite element solution of a new formulation for gas flow in a pipe with source terms. *J. Natural Gas Sci. Engrg.* 61:237–250, 2019.
3. C. Himpe, S. Grundel, and P. Benner. Model order reduction for gas and energy networks. *J. Math. Ind.*, 11:13, 2021.
4. A. Lewandowski. New numerical methods for transient modeling of gas pipeline networks. In *PSIG Annual Meeting*, pp. PSIG–9510, 1995.
5. L.A. Lotito and P.W. Halbert. Computer simulation of gas flow dynamics. *Pipeline Engineer*, 39:31–33, 29–31, 45–47, 1967.
6. J.L. Mead and R.A. Renaut. Optimal Runge-Kutta methods for first order pseudospectral operators. *J. Comput. Phys.* 152(1):404–419, 1999.
7. R. Samar, I. Postlewaite, and D.W. Gu. Model reduction with balanced realizations. *Int. J. Control*, 62(1):33–64, 1995.
8. W.Q. Tao and H.C. Ti. Transient analysis of gas pipeline network. *Chem. Engrg. J.*, 69(1):47–52, 1998.
9. P.J. van der Houwen. Explicit Runge-Kutta formulas with increased stability boundaries. *Numer. Math.* 20:149–164, 1972.
10. A. van der Schaft. Interconnections of input-output Hamiltonian systems with dissipation. In *Proceedings of the 55th IEEE Conference on Decision and Control*, pp. 4886–4691, 2016.