MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC
COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Reduced-order Modelling and Simulation of Gas Transportation Networks

Peter Benner
Joint work with Sara Grundel and Christian Himpe

Trends in Mathematical Modelling, Simulation and Optimisation: Theory and Applications

Virtual, 2–3 March 2021

Supported by:

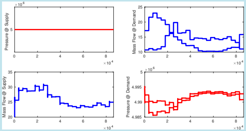
Federal Ministry
for Economic Affairs
and Energy

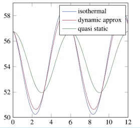
Math
Energy

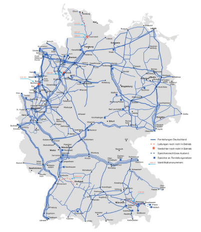
Simulation of German energy transportation networks

- **Goals:**
  - hierarchical modeling of transport and distribution networks
  - fast simulation on all levels
  - real-time scenario analysis for network operators
  - coupling of power and gas networks
- **Results:** New **discretization** and **model order reduction** methods for
  - isothermal Euler equations on network graph
  - with nonsmooth nonlinearity
  - leading to coupled system of differential-algebraic equations (DAEs)
  - with uncertain parameters

  Implemented in **morgen** — **M**odel **O**rder **R**eduction of **G**as and **E**nergy **N**etworks.







The German natural gas transportation network

Partners:

Fraunhofer SCAI
Fraunhofer ITWM
MPI Magdeburg
TU Berlin
HU Berlin
TU Dortmund
U Trier
PSI AG

Funded by:

Federal Ministry for Economic Affairs and Energy

We will do some gas network ...

We will do some gas network ...

- Modeling

We will do some gas network ...

- Modeling
- Model simplification

We will do some gas network ...

- Modeling
- Model simplification
- Model discretization

We will do some gas network ...

- Modeling
- Model simplification
- Model discretization
- Model reduction

We will do some gas network ...

- Modeling
- Model simplification
- Model discretization
- Model reduction
- Simulation experiments

Outline

1. Introduction

2. Modeling

3. Model Order Reduction

4. Outlook, Summary, Details

Outline

1. Introduction

2. Modeling

3. Model Order Reduction

4. Outlook, Summary, Details

Why accelerate gas network simulations?

Why accelerate gas network simulations?

- Transition to renewable and green energies.

Why accelerate gas network simulations?

- Transition to renewable and green energies.
- Regulatory requirements, real-time (15min decision horizon) control.

Why accelerate gas network simulations?

- Transition to renewable and green energies.
- Regulatory requirements, real-time (15min decision horizon) control.
- Employ modern developments in numerics and reduced-order modeling.

Why accelerate gas network simulations?

- Transition to renewable and green energies.
- Regulatory requirements, real-time (15min decision horizon) control.
- Employ modern developments in numerics and reduced-order modeling.
- It remains a challenge!

Some gas network properties:

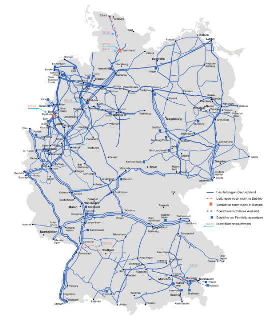
- $> 500,000$km gas pipelines in Germany[1] (earth-moon $< 400,000$km).

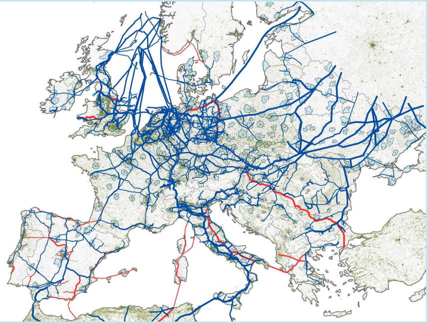
1 https://www.bmwi.de/Redaktion/EN/Artikel/Energy/gas-natural-gas-supply-in-germany.html
2
3

Some gas network properties:

- $> 500,000$km gas pipelines in Germany[1] (earth-moon $< 400,000$km).

German gas transportation network **. . . embedded into European network.**





Some gas network properties:

- $> 500,000$km gas pipelines in Germany[1] (earth-moon $< 400,000$km).
- $> 240,000,000\text{m}^3$ natural gas consumed per day.[2].

1 https://www.bmwi.de/Redaktion/EN/Artikel/Energy/gas-natural-gas-supply-in-germany.html
2 https://www.eia.gov/international/analysis/country/DEU
3

Some gas network properties:

- $> 500,000$km gas pipelines in Germany[1] (earth-moon $< 400,000$km).
- $> 240,000,000\text{m}^3$ natural gas consumed per day.[2].
- Gas and power become (critically) interlinked due to renewables.[3]

[1] https://www.bmwi.de/Redaktion/EN/Artikel/Energy/gas-natural-gas-supply-in-germany.html

[2] https://www.eia.gov/international/analysis/country/DEU

[3] http://www.bmwi-energiewende.de/EWD/Redaktion/EN/Newsletter/2017/07/Meldung/direkt-account.html

Some gas network properties:

- $> 500,000$km gas pipelines in Germany[1] (earth-moon $< 400,000$km).
- $> 240,000,000$m^3 natural gas consumed per day.[2].
- Gas and power become (critically) interlinked due to renewables.[3]
- Weather has effect on consumption **and** production.

[1] https://www.bmwi.de/Redaktion/EN/Artikel/Energy/gas-natural-gas-supply-in-germany.html
[2] https://www.eia.gov/international/analysis/country/DEU
[3] http://www.bmwi-energiewende.de/EWD/Redaktion/EN/Newsletter/2017/07/Meldung/direkt-account.html

Some gas network properties:

- $> 500,000$km gas pipelines in Germany[1] (earth-moon $< 400,000$km).
- $> 240,000,000\text{m}^3$ natural gas consumed per day.[2].
- Gas and power become (critically) interlinked due to renewables.[3]
- Weather has effect on consumption **and** production.
- Planning horizon is 24h, operator decision horizon is 15min.

[1] https://www.bmwi.de/Redaktion/EN/Artikel/Energy/gas-natural-gas-supply-in-germany.html
[2] https://www.eia.gov/international/analysis/country/DEU
[3] http://www.bmwi-energiewende.de/EWD/Redaktion/EN/Newsletter/2017/07/Meldung/direkt-account.html

Outline

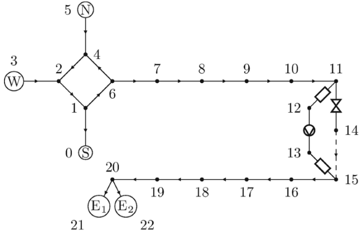
1. Introduction
2. Modeling
3. Model Order Reduction
4. Outlook, Summary, Details

Modeling I: Pipeline

Friction-dominated isothermal Euler equations for 1D pipes:

$$
\begin{aligned}
\frac{1}{\gamma_0\, z_0}\, \partial_t p &= -\frac{1}{S}\, \partial_x q \\
\partial_t q &= -S\, \partial_x p - \Big(\underbrace{\frac{S\, g\, \partial_x h}{\gamma_0\, z_0} p}_{\text{Gravity}} + \underbrace{\frac{\gamma_0\, z_0\, \lambda_0}{2\, d\, S} \frac{q\, |q|}{p}}_{\text{Friction}} \Big)
\end{aligned}
$$

- Pressure: $p(x,t)$
- Mass-flux: $q(x,t)$
- Height: $h(x)$
- Temperature: T_0
- Diameter: d
- Cross-section: S
- Roughness: k
- Gas Const.: R_S
- Gas state: $\gamma_0(T_0, R_S)$
- Compress.: $z_0(T_0, p)$
- Friction: $\lambda_0(k, d)$
- Grav. accel.: g

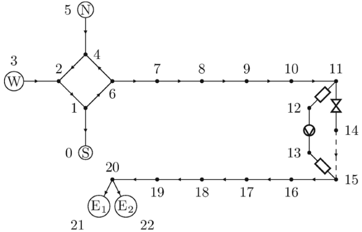


gas_N23_A24 from [Benner et al, 2019], modified from *GasLib-134*.

P. Benner, S. Grundel, C. Himpe, C. Huck, T. Streubel, C. Tischendorf (2019). Gas Network Benchmark Models. Applications of Differential-Algebraic Equations: Examples and Benchmarks, pp. 171-197, Springer, Cham.

Graph $(\mathcal{N}, \mathcal{E})$ incidence matrix $\mathcal{A}$:

$$\mathcal{A}_{ij} = \begin{cases} -1 & \mathcal{E}_j \text{ connects } \textbf{from } \mathcal{N}_i, \\ 0 & \mathcal{E}_j \text{ connects } \textbf{not } \mathcal{N}_i, \\ 1 & \mathcal{E}_j \text{ connects } \textbf{to } \mathcal{N}_i. \end{cases}$$

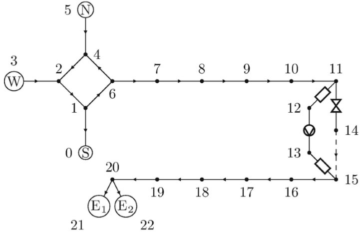


gas_N23_A24 from [Benner et al, 2019], modified from *GasLib-134*.

P. Benner, S. Grundel, C. Himpe, C. Huck, T. Streubel, C. Tischendorf (2019). Gas Network Benchmark Models. Applications of Differential-Algebraic Equations: Examples and Benchmarks, pp. 171-197, Springer, Cham.

Graph $(\mathcal{N}, \mathcal{E})$ incidence matrix $\mathcal{A}$:

$$\mathcal{A}_{ij} = \begin{cases} -1 & \mathcal{E}_j \text{ connects } \textbf{from } \mathcal{N}_i, \\ 0 & \mathcal{E}_j \text{ connects } \textbf{not } \mathcal{N}_i, \\ 1 & \mathcal{E}_j \text{ connects } \textbf{to } \mathcal{N}_i. \end{cases}$$



gas_N23_A24 from [Benner et al, 2019], modified from *GasLib-134*.

Kirchhoff's laws:

1. The net mass-flux at every node is zero.
2. The sum of directed pressure drops in every loop is zero.

P. Benner, S. Grundel, C. Himpe, C. Huck, T. Streubel, C. Tischendorf (2019). Gas Network Benchmark Models. Applications of Differential-Algebraic Equations: Examples and Benchmarks, pp. 171-197, Springer, Cham.

Vectorized PDAE gas network model:

$$
\begin{aligned}
D_d\,\partial_t p^* &= D_q \partial_x q, \\
\partial_t q^* &= D_p \partial_x p - \left(D_g\, p^* + D_f\, \frac{q^*\,|q^*|}{p^*} \right), \\
\mathcal{A}_0\, q^* &= \mathcal{B}_d\, d_q, \\
\mathcal{A}_0^\intercal\, p^* &= \mathcal{B}_s\, s_p,
\end{aligned}
$$

- p^* is the pressure at a t.b.d. pipe location.
- q^* is the mass-flux at a t.b.d. pipe location.
- D_* are diagonal matrices.
- $\mathcal{A}_0$ is the incidence matrix without supply node rows.
- $\mathcal{B}_s$ is the incidence matrix of supply node rows.
- $\mathcal{B}_d$ is the incidence matrix of demand node columns.

The choice of p^* and q^*:

The choice of p^* and q^*:

- Pipe midpoints:
  - (P)DAE tractability index bounded $\tau \leq 2$.
  - Given some weak topology constraints, PDAE becomes PDE [Grundel et al, 2014].
  - Boundary values affect friction term.

S. Grundel, L. Jansen, N. Hornung, T. Clees, C. Tischendorf, P. Benner (2014). Model order reduction of differential algebraic equations arising from the simulation of gas transport networks. In: Progress in Differential-Algebraic Equations, 183–205, Springer, Cham. doi:10.1007/978-3-662-44926-4_9.

The choice of p^* and q^*:

- Pipe midpoints:
  - (P)DAE tractability index bounded $\tau \leq 2$.
  - Given some weak topology constraints, PDAE becomes PDE [Grundel et al, 2014].
  - Boundary values affect friction term.
- Pipe endpoints:
  - (P)DAE tractability index bounded $\tau < 2$.
  - Given some weak topology constraints, PDAE becomes PDE.
  - Less oscillatory behaviour.

Hidden assumptions in this model:

Hidden assumptions in this model:

- Only cylindrical pipes.

Hidden assumptions in this model:

- Only cylindrical pipes.
- No temperature or pressure influence on pipe diameter: d const.

Hidden assumptions in this model:

- Only cylindrical pipes.
- No temperature or pressure influence on pipe diameter: d const.
- No variability or wear on pipe roughness: k const.

Hidden assumptions in this model:

- Only cylindrical pipes.
- No temperature or pressure influence on pipe diameter: d const.
- No variability or wear on pipe roughness: k const.
- No inertia term due to slow (sub-sonic) gas velocity: $-\frac{\gamma_0}{S^2}\left(\frac{q^2}{p}\right)_x \approx 0$.

Hidden assumptions in this model:

- Only cylindrical pipes.
- No temperature or pressure influence on pipe diameter: d const.
- No variability or wear on pipe roughness: k const.
- No inertia term due to slow (sub-sonic) gas velocity: $-\frac{\gamma_0}{S^2}\left(\frac{q^2}{p}\right)_x \approx 0$.
- Parametrization of averaged temperature and gas mix: $\gamma_0 = (T_0 \;\; R_S)$.

Hidden assumptions in this model:

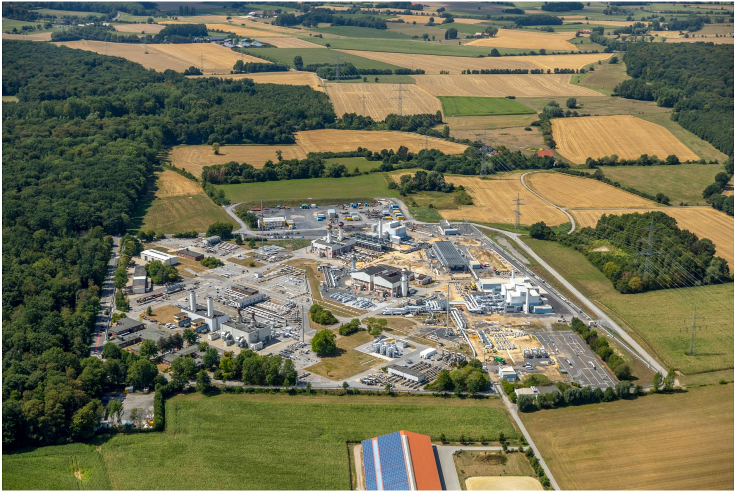
- Only cylindrical pipes.
- No temperature or pressure influence on pipe diameter: d const.
- No variability or wear on pipe roughness: k const.
- No inertia term due to slow (sub-sonic) gas velocity: $-\frac{\gamma_0}{S^2}\left(\frac{q^2}{p}\right)_x \approx 0$.
- Parametrization of averaged temperature and gas mix: $\gamma_0 = (T_0 \;\; R_S)$.
- Averaged compressibility based on steady-state: $z(p, T, x, t) \to z_0$.

Simplification II: Model

Hidden assumptions in this model:

- Only cylindrical pipes.
- No temperature or pressure influence on pipe diameter: d const.
- No variability or wear on pipe roughness: k const.
- No inertia term due to slow (sub-sonic) gas velocity: $-\frac{\gamma_0}{S^2}\left(\frac{q^2}{p}\right)_x \approx 0$.
- Parametrization of averaged temperature and gas mix: $\gamma_0 = (T_0 \;\; R_S)$.
- Averaged compressibility based on steady-state: $z(p, T, x, t) \to z_0$.
- Only step function boundary values.

Simplification III: Compressors



Natural gas compressor station in Werne, Germany, operated by Open Grid Europe.

Simplification III: Compressors

Simplified edge-based compressor models:

- Energy-based:

$$q_{\text{out}} = q_{\text{in}}$$

$$p_{\text{out}} = p_{\text{in}} \left(\frac{P_{\text{max}}}{p\gamma_0 z_0} \frac{\nu - 1}{\nu} + 1 \right)^{\frac{\nu}{\nu - 1}}$$

T.W.K. Mak, P. Van Hentenryck, A. Zlotnik, R. Bent (2019). Dynamic compressor optimization in natural gas pipeline systems. INFORMS Journal on Computing, 31(1):1–26. doi:10.1287/ijoc.2018.0821.

Simplified edge-based compressor models:

- Energy-based:

$$q_{\text{out}} = q_{\text{in}}$$
$$p_{\text{out}} = p_{\text{in}} \left(\frac{P_{\text{max}}}{p \gamma_0 z_0} \frac{\nu - 1}{\nu} + 1 \right)^{\frac{\nu}{\nu - 1}}$$

- Multiplicative:

$$q_{\text{out}} = q_{\text{in}}$$
$$p_{\text{out}} = p_{\text{in}} m_c$$

T.W.K. Mak, P. Van Hentenryck, A. Zlotnik, R. Bent (2019). Dynamic compressor optimization in natural gas pipeline systems. INFORMS Journal on Computing, 31(1):1–26. doi:10.1287/ijoc.2018.0821.

Simplified edge-based compressor models:

- Energy-based:

$$q_{\text{out}} = q_{\text{in}}$$
$$p_{\text{out}} = p_{\text{in}}\left(\frac{P_{\text{max}}}{p\gamma_0 z_0}\frac{\nu-1}{\nu}+1\right)^{\frac{\nu}{\nu-1}}$$

- Multiplicative:

$$q_{\text{out}} = q_{\text{in}}$$
$$p_{\text{out}} = p_{\text{in}} m_c$$

- Affine*:

$$q_{\text{out}} = q_{\text{in}}$$
$$p_{\text{out}} = p_c$$

T.W.K. Mak, P. Van Hentenryck, A. Zlotnik, R. Bent (2019). Dynamic compressor optimization in natural gas pipeline systems. INFORMS Journal on Computing, 31(1):1–26. doi:10.1287/ijoc.2018.0821.

1D First-Order Upwind Finite Differences:

1D First-Order Upwind Finite Differences:

- Axis-symmetric domain.

1D First-Order Upwind Finite Differences:

- Axis-symmetric domain.
- Pipelines length exceeds diameter by orders of magnitude.

1D First-Order Upwind Finite Differences:

- Axis-symmetric domain.
- Pipelines length exceeds diameter by orders of magnitude.
- Stable under CLF condition.

Discretization I: Space

1D First-Order Upwind Finite Differences:

- Axis-symmetric domain.
- Pipelines length exceeds diameter by orders of magnitude.
- Stable under CLF condition.

Time-aware spatial discretization:

- Set unit pipeline length based on CLF condition.

1D First-Order Upwind Finite Differences:

- Axis-symmetric domain.
- Pipelines length exceeds diameter by orders of magnitude.
- Stable under CLF condition.

Time-aware spatial discretization:

- Set unit pipeline length based on CLF condition.
- Treat too short pipes as short-cuts (instant and friction-free).

1D First-Order Upwind Finite Differences:

- Axis-symmetric domain.
- Pipelines length exceeds diameter by orders of magnitude.
- Stable under CLF condition.

Time-aware spatial discretization:

- Set unit pipeline length based on CLF condition.
- ~~Treat too short pipes as short-cuts (instant and friction-free).~~

1D First-Order Upwind Finite Differences:

- Axis-symmetric domain.
- Pipelines length exceeds diameter by orders of magnitude.
- Stable under CLF condition.

Time-aware spatial discretization:

- Set unit pipeline length based on CLF condition.
- ~~Treat too short pipes as short-cuts (instant and friction-free).~~
- Treat too short pipes as unit-length pipe with scaled friction.

1D First-Order Upwind Finite Differences:

- Axis-symmetric domain.
- Pipelines length exceeds diameter by orders of magnitude.
- Stable under CLF condition.

Time-aware spatial discretization:

- Set unit pipeline length based on CLF condition.
- ~~Treat too short pipes as short-cuts (instant and friction-free).~~
- Treat too short pipes as unit-length pipe with scaled friction.
- Sub-divide too long pipes to set of unit-length pipes.

Discussion of time-stepping:

Discussion of time-stepping:

- Adaptive methods (i.e. `ode45`, `ode23s`) are problematic.

Discussion of time-stepping:

- Adaptive methods (i.e. `ode45`, `ode23s`) are problematic.
- Implicit Runge-Kutta is problematic due to nonlinearity.

Discussion of time-stepping:

- Adaptive methods (i.e. `ode45`, `ode23s`) are problematic.
- Implicit Runge-Kutta is problematic due to nonlinearity.
- Implicit-Explicit (IMEX) methods are an appropriate tool.

Discussion of time-stepping:

- Adaptive methods (i.e. `ode45`, `ode23s`) are problematic.
- Implicit Runge-Kutta is problematic due to nonlinearity.
- Implicit-Explicit (IMEX) methods are an appropriate tool.
- Consider: SSP optimality, stiff accuracy, passivity, efficiency.

Discussion of time-stepping:

- Adaptive methods (i.e. `ode45`, `ode23s`) are problematic.
- Implicit Runge-Kutta is problematic due to nonlinearity.
- Implicit-Explicit (IMEX) methods are an appropriate tool.
- Consider: SSP optimality, stiff accuracy, passivity, efficiency.
- We recommend first order IMplicit-EXplicit method (i.e., combination of forward/backward Euler), providing often the best compromise between efficiency and accuracy, but other solvers are available in `morgen`, e.g. second-order IMEX (trapezoidal rule + SDIRK) with parametric Butcher tableau:

Explicit:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Implicit:

$$\begin{array}{c|cc} \lambda & \lambda & 0 \\ 1-\lambda & 1-2\lambda & \lambda \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

The Input-Output Model

Parametric, Structured, Nonlinear, Non-Normal, Square:

$$\begin{bmatrix} E_p(\theta) & 0 \\ 0 & I_{N_q} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & A_{pq} \\ A_{qp} & 0 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} + \begin{bmatrix} 0 & B_{pd} \\ B_{qs} & 0 \end{bmatrix} \begin{bmatrix} s_p \\ d_q \end{bmatrix} + \begin{bmatrix} 0 \\ F_c + f_q(p, q, s_p, \theta) \end{bmatrix}$$

$$\begin{bmatrix} s_q \\ d_p \end{bmatrix} = \begin{bmatrix} 0 & C_{sq} \\ C_{dp} & 0 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix}$$

$$\begin{bmatrix} p_0 \\ q_0 \end{bmatrix} = \begin{bmatrix} \bar{p}(\bar{s}_p, \bar{d}_q) \\ \bar{q}(\bar{s}_p, \bar{d}_q) \end{bmatrix}$$

Input:

- Pressure at supply: s_p
- Mass-Flux at demand: d_q

State:

- Pressure: p
- Mass-Flux: q

Output:

- Mass-Flux at supply: s_q
- Pressure at demand: d_p

Two-step steady state algorithm:

1a. Linear mass-flux steady-state: $A_{pq}\,\bar{q} = -B_{pd}\,\bar{d}_q$

1b. Linear pressure steady-state: $A_{qp}\,\bar{p} = -\Big(B_{qs}\,\bar{s}_p + F_c\Big)$

2. Corrected pressure steady-state: $A_{qp} p_{k+1} = -\Big(B_{qs}\bar{s}_p + F_c + f_q(p_k, \bar{q}, \bar{s}_p, \theta)\Big)$

Two-step steady state algorithm:

1a. Linear mass-flux steady-state: $A_{pq}\,\bar{q} = -B_{pd}\,\bar{d}_q$

1b. Linear pressure steady-state: $A_{qp}\,\bar{p} = -\Big(B_{qs}\,\bar{s}_p + F_c\Big)$

2. Corrected pressure steady-state: $A_{qp} p_{k+1} = -\Big(B_{qs}\bar{s}_p + F_c + f_q(p_k, \bar{q}, \bar{s}_p, \theta)\Big)$

- Note, A and B do not depend on the parameter!

Two-step steady state algorithm:

1a. Linear mass-flux steady-state: $A_{pq}\,\bar{q} = -B_{pd}\,\bar{d}_q$

1b. Linear pressure steady-state: $A_{qp}\,\bar{p} = -\Big(B_{qs}\,\bar{s}_p + F_c\Big)$

2. Corrected pressure steady-state: $A_{qp} p_{k+1} = -\Big(B_{qs}\bar{s}_p + F_c + f_q(p_k, \bar{q}, \bar{s}_p, \theta)\Big)$

- Note, A and B do not depend on the parameter!
- Step 1a and Step 1b via least squares (in parallel).

Two-step steady state algorithm:

1a. Linear mass-flux steady-state: $A_{pq}\,\bar{q} = -B_{pd}\,\bar{d}_q$

1b. Linear pressure steady-state: $A_{qp}\,\bar{p} = -\Big(B_{qs}\,\bar{s}_p + F_c\Big)$

2. Corrected pressure steady-state: $A_{qp}p_{k+1} = -\Big(B_{qs}\bar{s}_p + F_c + f_q(p_k, \bar{q}, \bar{s}_p, \theta)\Big)$

- Note, A and B do not depend on the parameter!
- Step 1a and Step 1b via least squares (in parallel).
- Repeat Step 2 until happy (reuse QR of Step 1b).

Two-step steady state algorithm:

1a. Linear mass-flux steady-state: $A_{pq}\,\bar{q} = -B_{pd}\,\bar{d}_q$

1b. Linear pressure steady-state: $A_{qp}\,\bar{p} = -\Big(B_{qs}\,\bar{s}_p + F_c\Big)$

2. Corrected pressure steady-state: $A_{qp}p_{k+1} = -\Big(B_{qs}\bar{s}_p + F_c + f_q(p_k, \bar{q}, \bar{s}_p, \theta)\Big)$

- Note, A and B do not depend on the parameter!
- Step 1a and Step 1b via least squares (in parallel).
- Repeat Step 2 until happy (reuse QR of Step 1b).
- Repeating Step 2 is a special case of an IMEX solver.

Two-step steady state algorithm:

1a. Linear mass-flux steady-state: $A_{pq}\,\bar{q} = -B_{pd}\,\bar{d}_q$

1b. Linear pressure steady-state: $A_{qp}\,\bar{p} = -\Big(B_{qs}\,\bar{s}_p + F_c\Big)$

2. Corrected pressure steady-state: $A_{qp} p_{k+1} = -\Big(B_{qs}\bar{s}_p + F_c + f_q(p_k, \bar{q}, \bar{s}_p, \theta)\Big)$

- Note, A and B do not depend on the parameter!
- Step 1a and Step 1b via least squares (in parallel).
- Repeat Step 2 until happy (reuse QR of Step 1b).
- Repeating Step 2 is a special case of an IMEX solver.
- If more accuracy is needed, iterate with 1st order IMEX solver.

Outline

1. Introduction

2. Modeling

3. Model Order Reduction

4. Outlook, Summary, Details

Recap:

From: Hyperbolic 2D PDAE

To: Non-normal, coupled, nonlinear, parametric ODE

Wish list:

Recap:

From: Hyperbolic 2D PDAE

To: Non-normal, coupled, nonlinear, parametric ODE

Wish list:

- Perturbation system $\rightarrow$ Deviation from steady state

Recap:

From: Hyperbolic 2D PDAE

To: Non-normal, coupled, nonlinear, parametric ODE

Wish list:

- Perturbation system → Deviation from steady state
- Input-output system → System-theoretic methods

Recap:

From: Hyperbolic 2D PDAE

To: Non-normal, coupled, nonlinear, parametric ODE

Wish list:

- Perturbation system → Deviation from steady state
- Input-output system → System-theoretic methods
- Coupled system → Structure-preserving methods

Recap:

From: Hyperbolic 2D PDAE

To: Non-normal, coupled, nonlinear, parametric ODE

Wish list:

- Perturbation system → Deviation from steady state
- Input-output system → System-theoretic methods
- Coupled system → Structure-preserving methods
- Nonlinearity and 2D parametrization → Data-driven methods

Recap:

From: Hyperbolic 2D PDAE

To: Non-normal, coupled, nonlinear, parametric ODE

Wish list:

- Perturbation system → Deviation from steady state
- Input-output system → System-theoretic methods
- Coupled system → Structure-preserving methods
- Nonlinearity and 2D parametrization → Data-driven methods
- Large-scale → Low-rank computable methods*

Split reduction operators

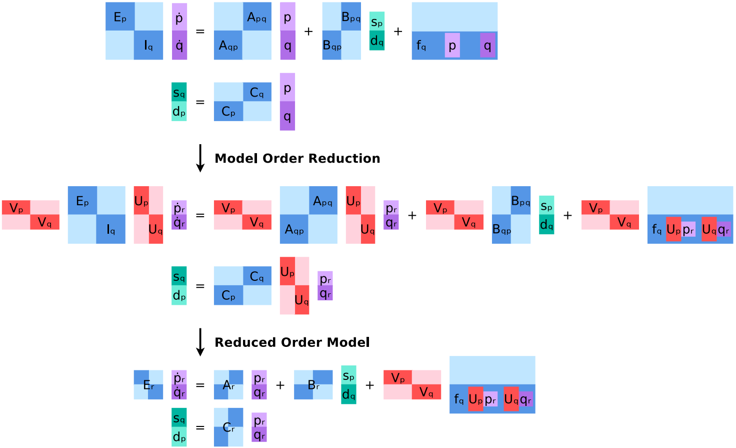
$$W = \begin{bmatrix} W_p \\ W_q \end{bmatrix} \in \mathbb{R}^{(N_p+N_q)\times r}$$

into structure-preserving reduction operator

$$\begin{bmatrix} W_p & \\ & W_q \end{bmatrix} \in \mathbb{R}^{(N_p+N_q)\times 2r},$$

where $W \in \{V, U\}$.

K. Kerns, A. Yang (1998). Preservation of passivity during RLC network reduction via split congruence transformations. IEEE Trans. CAD Integr. Circuits Syst. 17(7):582–591.



K. Kerns, A. Yang (1998). Preservation of passivity during RLC network reduction via split congruence transformations. IEEE Trans. CAD Integr. Circuits Syst. 17(7):582–591.

The tested model reduction methods:

The tested model reduction methods:

Structured POD, via:	empirical reachability Gramian

All implemented via `emgr` software platform [Himpe 2018].

The tested model reduction methods:

Structured POD, via:	empirical reachability Gramian
Structured Dominant Subspaces, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian

All implemented via `emgr` software platform [Himpe 2018].

The tested model reduction methods:

Structured POD, via:	empirical reachability Gramian
Structured Dominant Subspaces, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian
Structured Balanced POD, via:	empirical reachability & observability Gramian

All implemented via `emgr` software platform [Himpe 2018].

The tested model reduction methods:

Structured POD, via:	empirical reachability Gramian
Structured Dominant Subspaces, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian
Structured Balanced POD, via:	empirical reachability & observability Gramian
Structured Balanced Truncation, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian

All implemented via `emgr` software platform [Himpe 2018].

The tested model reduction methods:

Structured POD, via:	empirical reachability Gramian
Structured Dominant Subspaces, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian
Structured Balanced POD, via:	empirical reachability & observability Gramian
Structured Balanced Truncation, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian
Structured Balanced Gains, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian

All implemented via `emgr` software platform [Himpe 2018].

The tested model reduction methods:

Structured POD, via:	empirical reachability Gramian
Structured Dominant Subspaces, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian
Structured Balanced POD, via:	empirical reachability & observability Gramian
Structured Balanced Truncation, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian
Structured Balanced Gains, via:	empirical reachability & observability Gramian
	empirical cross Gramian
	empirical non-symmetric cross Gramian
Structured DMD Galerkin, via:	empirical reachability Gramian

All implemented via `emgr` software platform [Himpe 2018].

Spotlight: DMD-Galerkin

Plain Vanilla DMD:

$$X = [x_0 \;\; x_1 \;\; \dots \;\; x_T] \rightarrow \left\{ \begin{matrix} X_0 := [x_0 \;\; x_1 \;\; \dots \;\; x_{T-1}] \\ X_1 := [x_1 \;\; x_2 \;\; \dots \;\; x_T] \end{matrix} \right\} \rightarrow X_1 \overset{!}{\approx} \mathcal{A} X_0 \Rightarrow \mathcal{A} \approx X_1 X_0^+$$

4

5

Spotlight: DMD-Galerkin

Plain Vanilla DMD:

$$X = [x_0 \;\; x_1 \;\; \dots \;\; x_T] \to \begin{Bmatrix} X_0 := [x_0 \;\; x_1 \;\; \dots \;\; x_{T-1}] \\ X_1 := [x_1 \;\; x_2 \;\; \dots \;\; x_T] \end{Bmatrix} \to X_1 \overset{!}{\approx} \mathcal{A} X_0 \Rightarrow \mathcal{A} \approx X_1 X_0^+$$

. . . with centering[4]

$$X \to \overline{X} := \big[(x_0 - \bar{x}) \quad (x_1 - \bar{x}) \quad \dots \quad (x_T - \bar{x})\big]$$

[4] S.M. Hirsh, K.D. Harris, J.N. Kutz, B.W. Brunton. Centering Data Improves the Dynamic Mode Decomposition. SIAM J. Appl. Dyn. Syst., 19(3): 1920–1955, 2020. doi:10.1137/19M1289881

[5]

Plain Vanilla DMD:

$$X = [x_0 \ \ x_1 \ \ \ldots \ \ x_T] \to \begin{Bmatrix} X_0 := [x_0 \ \ x_1 \ \ \ldots \ \ x_{T-1}] \\ X_1 := [x_1 \ \ x_2 \ \ \ldots \ \ x_T] \end{Bmatrix} \to X_1 \overset{!}{\approx} \mathcal{A} X_0 \Rightarrow \mathcal{A} \approx X_1 X_0^+$$

. . . with centering[4]

$$X \to \overline{X} := \left[(x_0 - \bar{x}) \quad (x_1 - \bar{x}) \quad \ldots \quad (x_T - \bar{x})\right]$$

. . . used as Model reduction method:DMD-Galerkin[5]

$$\mathcal{A} \overset{\text{tSVD}}{=} U_1 D_1 V_1$$

[4] S.M. Hirsh, K.D. Harris, J.N. Kutz, B.W. Brunton. Centering Data Improves the Dynamic Mode Decomposition. SIAM J. Appl. Dyn. Syst., 19(3): 1920–1955, 2020. doi:10.1137/19M1289881

[5] A. Alla, J.N. Kutz. Nonlinear model order reduction via dynamic mode decomposition. SIAM J. Sci. Comput., 39(5):B778–B796, 2017. doi:10.1137/16M1059308

Plain Vanilla DMD:

$$X = [x_0 \ \ x_1 \ \ \dots \ \ x_T] \rightarrow \begin{Bmatrix} X_0 := [x_0 \ \ x_1 \ \ \dots \ \ x_{T-1}] \\ X_1 := [x_1 \ \ x_2 \ \ \dots \ \ x_T] \end{Bmatrix} \rightarrow X_1 \overset{!}{\approx} \mathcal{A} X_0 \Rightarrow \mathcal{A} \approx X_1 X_0^+$$

. . . with centering[4]

$$X \rightarrow \overline{X} := \left[(x_0 - \bar{x}) \quad (x_1 - \bar{x}) \quad \dots \quad (x_T - \bar{x}) \right]$$

. . . used as Model reduction method:DMD-Galerkin[5]

$$\mathcal{A} \overset{\text{tSVD}}{=} U_1 D_1 V_1$$

. . . can be computed via empirical Gramian (exact-DMD "kernel"):

$$W_R = \sum_m^M \kappa(\overline{X}_m, \overline{X}_m) \quad \begin{cases} \kappa_{\text{Linear}}(X, Y) := X\, Y^\intercal \\ \kappa_{\text{DMD}}(X, Y) := X_1 Y_0^+ \end{cases}$$

---

[4] S.M. Hirsh, K.D. Harris, J.N. Kutz, B.W. Brunton. Centering Data Improves the Dynamic Mode Decomposition. SIAM J. Appl. Dyn. Syst., 19(3): 1920–1955, 2020. doi:10.1137/19M1289881

[5] A. Alla, J.N. Kutz. Nonlinear model order reduction via dynamic mode decomposition. SIAM J. Sci. Comput., 39(5):B778–B796, 2017. doi:10.1137/16M1059308

Plain Vanilla DMD:

$$X = [x_0 \ x_1 \ \dots \ x_T] \rightarrow \begin{Bmatrix} X_0 := [x_0 \ x_1 \ \dots \ x_{T-1}] \\ X_1 := [x_1 \ x_2 \ \dots \ x_T] \end{Bmatrix} \rightarrow X_1 \overset{!}{\approx} \mathcal{A} X_0 \Rightarrow \mathcal{A} \approx X_1 X_0^+$$

. . . with centering[4]

$$X \rightarrow \overline{X} := \left[(x_0 - \bar{x}) \quad (x_1 - \bar{x}) \quad \dots \quad (x_T - \bar{x}) \right]$$

. . . used as Model reduction method:DMD-Galerkin[5]

$$\mathcal{A} \overset{\text{tSVD}}{=} U_1 D_1 V_1$$

. . . can be computed via empirical Gramian (exact-DMD "kernel"):

$$W_R = \sum_m^M \kappa(\overline{X}_m, \overline{X}_m) \quad \begin{cases} \kappa_{\text{Linear}}(X, Y) := X\,Y^\intercal \\ \kappa_{\text{DMD}}(X, Y) := X_1 Y_0^+ \end{cases}$$

→ (Centered) DMD-Galerkin via (Discrete) Empirical Reachability Gramian!

[4] S.M. Hirsh, K.D. Harris, J.N. Kutz, B.W. Brunton. Centering Data Improves the Dynamic Mode Decomposition. SIAM J. Appl. Dyn. Syst., 19(3): 1920–1955, 2020. doi:10.1137/19M1289881

[5] A. Alla, J.N. Kutz. Nonlinear model order reduction via dynamic mode decomposition. SIAM J. Sci. Comput., 39(5):B778–B796, 2017. doi:10.1137/16M1059308

Disclaimer:

Disclaimer:

- First, what is the best linear subspace for model order reduction?

Disclaimer:

- First, what is the best linear subspace for model order reduction?
- What hyper-reduction should be used (DEIM, DMD, NL, etc.)?

Disclaimer:

- First, what is the best linear subspace for model order reduction?
- What hyper-reduction should be used (DEIM, DMD, NL, etc.)?
- How do model reduction and hyper-reduction interact?

Disclaimer:

- First, what is the best linear subspace for model order reduction?
- What hyper-reduction should be used (DEIM, DMD, NL, etc.)?
- How do model reduction and hyper-reduction interact?
- How to recycle simulations (efficiently)?

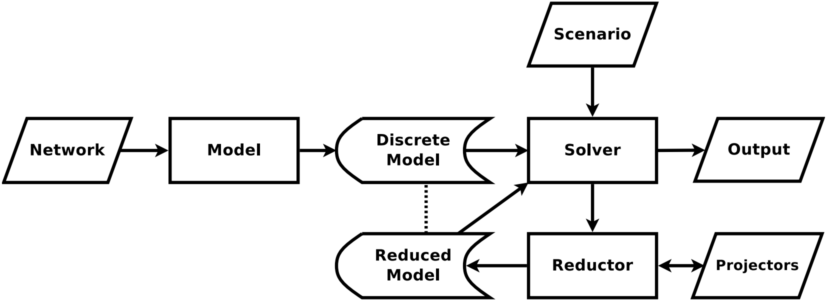
Disclaimer:

- First, what is the best linear subspace for model order reduction?
- What hyper-reduction should be used (DEIM, DMD, NL, etc.)?
- How do model reduction and hyper-reduction interact?
- How to recycle simulations (efficiently)?
- Is hyper-reduction avoidable due to repeated scalar nonlinearites?

Disclaimer:

- First, what is the best linear subspace for model order reduction?
- What hyper-reduction should be used (DEIM, DMD, NL, etc.)?
- How do model reduction and hyper-reduction interact?
- How to recycle simulations (efficiently)?
- Is hyper-reduction avoidable due to repeated scalar nonlinearites?

$\rightarrow$ No hyper-reduction implemented (yet).



Major modules:

- networks
- models
- solvers
- reductors
- tests

Minor modules:

- utils
- tools

Set-up

6

Set-up

- Short training, long testing

6

Set-up

- Short training, long testing
- Generic training scenario (constant input)

6

Set-up

- Short training, long testing
- Generic training scenario (constant input)
- Disjoint training and test parameters

6

Set-up

- Short training, long testing
- Generic training scenario (constant input)
- Disjoint training and test parameters
- Tested models: `ode_mid`, `ode_end`

6

Set-up

- Short training, long testing
- Generic training scenario (constant input)
- Disjoint training and test parameters
- Tested models: `ode_mid`, `ode_end`
- Tested solvers: `imex1`, `imex2`

6

Set-up

- Short training, long testing
- Generic training scenario (constant input)
- Disjoint training and test parameters
- Tested models: `ode_mid`, `ode_end`
- Tested solvers: `imex1`, `imex2`
- Tested reductors:
  - `pod_r`
  - `eds_ro`, `eds_wx`, `eds_wz`
  - `bpod_ro`,
  - `ebt_ro`, `ebt_wx`, `ebt_wz`
  - `ebg_ro`, `ebg_wx`, `ebg_wz`
  - `dmd_r`,

6

Set-up

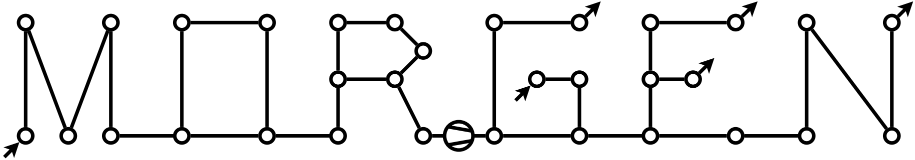
- Short training, long testing
- Generic training scenario (constant input)
- Disjoint training and test parameters
- Tested models: `ode_mid`, `ode_end`
- Tested solvers: `imex1`, `imex2`
- Tested reductors:
  - `pod_r`
  - `eds_ro`, `eds_wx`, `eds_wz`
  - `bpod_ro`,
  - `ebt_ro`, `ebt_wx`, `ebt_wz`
  - `ebg_ro`, `ebg_wx`, `ebg_wz`
  - `dmd_r`,
- Heuristic $L_{i\in\{1,2,\infty\}} \otimes L_{j\in\{1,2,\infty\}}$ error norm computation

6

Set-up

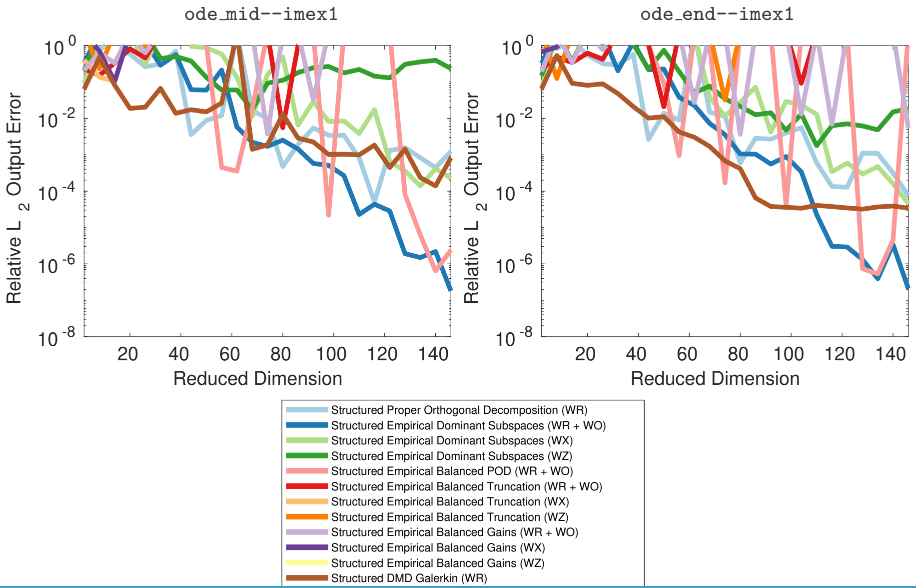
- Short training, long testing
- Generic training scenario (constant input)
- Disjoint training and test parameters
- Tested models: `ode_mid`, `ode_end`
- Tested solvers: `imex1`, `imex2`
- Tested reductors:
  - `pod_r`
  - `eds_ro`, `eds_wx`, `eds_wz`
  - `bpod_ro`,
  - `ebt_ro`, `ebt_wx`, `ebt_wz`
  - `ebg_ro`, `ebg_wx`, `ebg_wz`
  - `dmd_r`,
- Heuristic $L_{i\in\{1,2,\infty\}} \otimes L_{j\in\{1,2,\infty\}}$ error norm computation
- Compare MORSCORE[6]

[6] C. Himpe (2020). Comparing (empirical-Gramian-based) model order reduction algorithms. arXiv [math.OC], arXiv:2002.12226.



- 2 Cycles
- 1 Compressor
- 2 Supply nodes
- 4 Demand nodes
- Pipe length $[20, 60]$km
- Time resolution 60s
- Temperature: $[0, 15]$°C
- Gas constant: $[500, 600]\frac{\text{J}}{\text{kg K}}$
- *Schifrinson* friction factor
- *AGA88* compressibility factor
- 900 States
- 6 Inputs & Outputs
- Training horizon: 1h
- Test horizon: 24h
- Perturbed steady-state training
- Standard load profiles testing

Experiment II: $L_2 \otimes L_2$ Model Reduction Error



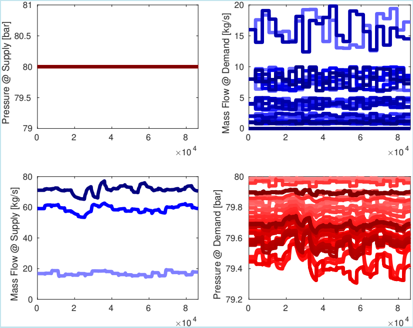
Experiment II: Evaluation

	ode_mid imex_1	ode_end imex_1	ode_mid imex_2	ode_end imex_2
pod_r	0.12	0.12	0.04	0.05
eds_ro	0.16	0.16	0.05	0.06
eds_wx	0.08	0.08	0.02	0.02
eds_wz	0.03	0.07	0.02	0.04
bpod_ro	0.07	0.07	0.02	0.02
ebt_ro	0.00	0.00	0.03	0.03
ebt_wx	0.00	0.00	0.00	0.00
ebt_wz	0.00	0.00	0.00	0.00
ebg_ro	0.00	0.01	0.02	0.02
ebg_wx	0.00	0.00	0.00	0.00
ebg_wz	0.00	0.00	0.00	0.00
dmd_r	0.14	0.18	0.03	0.04

MORscores $\mu(150, \epsilon_{\text{mach}(16)})$ in the $L_2 \otimes L_2$ norm for the "MORGEN" network.

The Network

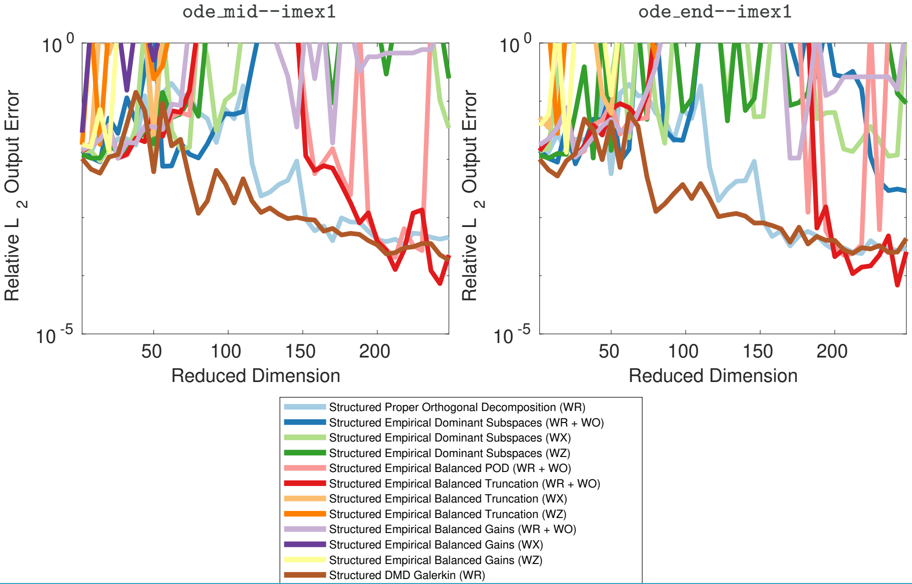
The Scenarios



- total length: 1412km
- 1 compressor

- steady-state, used as initial state:
  - pressure of 80bar at supply nodes and compressor;
  - demand mass-fluxes up to 16kg.

- 3886 states
- 48 inputs and outputs
- 20sec time steps


ode_mid--imex1
ode_end--imex1
10^0
10^-5
Relative L 2 Output Error
50
100
150
200
Reduced Dimension
Structured Proper Orthogonal Decomposition (WR)
Structured Empirical Dominant Subspaces (WR + WO)
Structured Empirical Dominant Subspaces (WX)
Structured Empirical Dominant Subspaces (WZ)
Structured Empirical Balanced POD (WR + WO)
Structured Empirical Balanced Truncation (WR + WO)
Structured Empirical Balanced Truncation (WX)
Structured Empirical Balanced Truncation (WZ)
Structured Empirical Balanced Gains (WR + WO)
Structured Empirical Balanced Gains (WX)
Structured Empirical Balanced Gains (WZ)
Structured DMD Galerkin (WR)

1. Introduction

2. Modeling

3. Model Order Reduction

4. Outlook, Summary, Details

Some open problems and future work:

- Port-Hamiltonian model
- Parametric pipe roughness
- Intraday switchable valves
- Minimal training horizon
- SciGRID_gas network
- OGE partDE network

Conclusions from computational experiments:

- Prefer the endpoint model.
- Prefer the first-order IMEX solver.
- Prefer Galerkin model reduction methods.

Conclusions from computational experiments:

- Prefer the endpoint model.
- Prefer the first-order IMEX solver.
- Prefer Galerkin model reduction methods.

The Paper

Christian Himpe, Sara Grundel, and Peter Benner.
Model Order Reduction for Gas and Energy Networks.
arXiv Preprint [math.OC] arXiv:2011.12099, November 2020.

Summary and Details

Conclusions from computational experiments:

- Prefer the endpoint model.
- Prefer the first-order IMEX solver.
- Prefer Galerkin model reduction methods.

The Paper

Christian Himpe, Sara Grundel, and Peter Benner.
Model Order Reduction for Gas and Energy Networks.
arXiv Preprint [math.OC] arXiv:2011.12099, November 2020.

The Software: `morgen` (Model Order Reduction for Gas and Energy Networks)

MATLAB code (Octave-compatible), under BSD 2-Clause License, available at:

`doi:10.5281/zenodo.4288510`