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 — Model Order Reduction of Gas and Energy Networks.

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
1. Introduction

2. Modeling

3. Model Order Reduction

4. Outlook, Summary, Details
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.
Introduction

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).

---


\(^2\)

\(^3\)
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:

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

---

2. [https://www.eia.gov/international/analysis/country/DEU](https://www.eia.gov/international/analysis/country/DEU)
**Facts**

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

---

\(^2\) [https://www.eia.gov/international/analysis/country/DEU](https://www.eia.gov/international/analysis/country/DEU)
Facts

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.
- Planning horizon is 24h, operator decision horizon is 15min.

\(^2\) [https://www.eia.gov/international/analysis/country/DEU](https://www.eia.gov/international/analysis/country/DEU)
1. Introduction

2. Modeling

3. Model Order Reduction

4. Outlook, Summary, Details
Friction-dominated isothermal Euler equations for 1D pipes:

\[
\frac{1}{\gamma_0 z_0} \partial_t p = - \frac{1}{S} \partial_x q \\
\partial_t q = -S \partial_x p - \left( \frac{S g \partial_x h}{\gamma_0 z_0} p + \frac{\gamma_0 z_0 \lambda_0}{2 d S} \frac{|q|}{p} \right) 
\]

- 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.: \( g \)
Graph-based modeling of transportation networks:

\[ \text{Graph} (N, E) \]

Incidence matrix \( A \):

\[ A_{ij} = \begin{cases} -1 & \text{if } E_j \text{ connects from } N_i, \\ 0 & \text{if } E_j \text{ connects not } N_i, \\ 1 & \text{if } E_j \text{ connects to } N_i. \end{cases} \]

Gas network benchmark models: 

\textit{gas} \_N23\_A24 from \cite{Benner2019}, modified from \textit{GasLib-134}.

Graph-based modeling of transportation networks:

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

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

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

---

**References**

Graph-based modeling of transportation networks:

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

\[
A_{ij} = \begin{cases} 
-1 & \mathcal{E}_j \text{ connects from } \mathcal{N}_i, \\
0 & \mathcal{E}_j \text{ connects not } \mathcal{N}_i, \\
1 & \mathcal{E}_j \text{ connects to } \mathcal{N}_i. 
\end{cases}
\]

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.

---

Vectorized PDAE gas network model:

\[ D_d \partial_t p^* = D_q \partial_x q, \]
\[ \partial_t q^* = D_p \partial_x p - \left( D_g p^* + D_f \frac{q^* |q^*|}{p^*} \right), \]
\[ A_0 q^* = B_d d_q, \]
\[ A_0^T p^* = B_s s_p, \]

- \( p^* \) 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.
- \( A_0 \) is the incidence matrix without supply node rows.
- \( B_s \) is the incidence matrix of supply node rows.
- \( B_d \) 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.

---

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: $d \text{ const.}$
- No variability or wear on pipe roughness: $k \text{ const.}$
- 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: $z(p, T, x, t) \to z_0.$
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: \(-\gamma_0\frac{q^2}{S^2}\left(\frac{q^2}{p}\right)_x \approx 0\).
- Parametrization of averaged temperature and gas mix: \( \gamma_0 = (T_0 \ R_S) \).
- Averaged compressibility based on steady-state: \( z(p, T, x, t) \approx z_0 \).
- Only step function boundary values.
Natural gas compressor station in Werne, Germany, operated by Open Grid Europe.

Simplification III: Compressors

Energy-based:
\[ q_{out} = q_{in} \]
\[ p_{out} = p_{in} (P_{max} p^{\gamma_0} z_0 - 1 + 1)^{\frac{1}{\gamma_0}} \]

Multiplicative:
\[ q_{out} = q_{in} \]
\[ p_{out} = p_{in} m_c \]

Affine\*:
\[ q_{out} = q_{in} \]
\[ p_{out} = p_c \]


© benner@mpi-magdeburg.mpg.de

Reduced-order Modelling and Simulation of Gas Transportation Networks 14/34
Simplified edge-based compressor models:

- **Energy-based:**
  \[
  q_{\text{out}} = q_{\text{in}} \\
  p_{\text{out}} = p_{\text{in}} \left( \frac{P_{\text{max}}}{p_{\gamma_0 z_0}} \frac{\nu - 1}{\nu} + 1 \right)^{\frac{1}{\nu - 1}}
  \]

- **Multiplicative:**
  \[
  q_{\text{out}} = q_{\text{in}} \\
  p_{\text{out}} = p_{\text{in}} \cdot m
  \]

- **Affine:**
  \[
  q_{\text{out}} = q_{\text{in}} \\
  p_{\text{out}} = p_{\text{in}} + c
  \]

---

Simplified edge-based compressor models:

- **Energy-based:**
  \[ q_{out} = q_{in} \]
  \[ p_{out} = p_{in} \left( \frac{P_{max}}{p_{\gamma_0} z_0} \frac{\nu - 1}{\nu} + 1 \right)^{\frac{\nu}{\nu-1}} \]

- **Multiplicative:**
  \[ q_{out} = q_{in} \]
  \[ p_{out} = p_{in} m_c \]

---

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*:**
  \[ q_{\text{out}} = q_{\text{in}} \]
  \[ p_{\text{out}} = p_c \]

---

1D First-Order Upwind Finite Differences:
Discretization I: Space

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.
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.
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.
- 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.
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.
- 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:

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:

\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 1 & 0 \\
1 & 2 & 1 \\
2 & 2 & 1 \\
\end{array}
\]

Implicit:

\[
\begin{array}{ccc}
\lambda & \lambda & 0 \\
1 & -\lambda & 1 \\
-2 & \lambda & \lambda \\
1 & 2 & 1 \\
\end{array}
\]
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:

\[
\begin{align*}
\text{Explicit:} & & \text{Implicit:} \\
0 & 0 & 0 & \lambda & \lambda & 0 \\
1 & 1 & 0 & 1 - \lambda & 1 - 2\lambda & \lambda \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{align*}
\]
The Input-Output Model

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

\[
\begin{bmatrix}
E_p(\theta) & 0 \\
0 & I_{N_q}
\end{bmatrix}
\begin{bmatrix}
\dot{p} \\
\dot{q}
\end{bmatrix}
= \begin{bmatrix}
0 & A_{pq} \\
A_{qp} & 0
\end{bmatrix}
\begin{bmatrix}
p \\
q
\end{bmatrix}
+ \begin{bmatrix}
0 & B_{ps} \\
B_{qp} & 0
\end{bmatrix}
\begin{bmatrix}
s_p \\
d_q
\end{bmatrix}
+ \begin{bmatrix}
F + f_q(p, q, s_p, \theta) \\
0
\end{bmatrix}
\]

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

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

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: \( A_{qp} \bar{p} = -\left( B_{qs} \bar{s}_p + F_c \right) \)

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

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

- 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: $A_{qp} \bar{p} = -\left( B_{qs} \bar{s}_p + F_c \right)$

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

- 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: \( A_{qp} \bar{p} = - \left( B_{qs} \bar{s}_p + F_c \right) \)

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

- 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: \( A_{qp} \bar{p} = -\left( B_{qs} \bar{s}_p + F_c \right) \)

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

- 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: \[ A_{qp} \bar{p} = -\left( B_{qs} \bar{s}_p + F_c \right) \]

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

- 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.
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 $\rightarrow$ Deviation from steady state
- Input-output system $\rightarrow$ 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 $\rightarrow$ Deviation from steady state
- Input-output system $\rightarrow$ System-theoretic methods
- Coupled system $\rightarrow$ Structure-preserving methods
- Nonlinearity and 2D parametrization $\rightarrow$ 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

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

where \( W \in \{V, U\} \).

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 \( \text{emgr} \) software platform [Himpe 2018].
The tested model reduction methods:

Structured POD, via: empirical reachability Gramian

All implemented via emgr software platform [HIMPE 2018].
## The tested model reduction methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured POD, via:</td>
<td>empirical reachability Gramian</td>
</tr>
<tr>
<td>Structured Dominant Subspaces, via:</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
</tbody>
</table>

All implemented via *emgr* software platform [Himpe 2018].
The tested model reduction methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Via:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured POD</td>
<td>empirical reachability Gramian</td>
</tr>
<tr>
<td>Structured Dominant Subspaces</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
<tr>
<td>Structured Balanced POD</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
</tbody>
</table>

All implemented via `emgr` software platform [Himpe 2018].
The tested model reduction methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Gramians</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured POD, via:</td>
<td>empirical reachability Gramian</td>
</tr>
<tr>
<td>Structured Dominant Subspaces, via:</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
<tr>
<td>Structured Balanced POD, via:</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td>Structured Balanced Truncation, via:</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
</tbody>
</table>

All implemented via emgr software platform [HIMPE 2018].
The tested model reduction methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Via</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured POD</td>
<td>empirical reachability Gramian</td>
</tr>
<tr>
<td>Structured Dominant Subspaces</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
<tr>
<td>Structured Balanced POD</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td>Structured Balanced Truncation</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
<tr>
<td>Structured Balanced Gains</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
</tbody>
</table>

All implemented via emgr software platform \[HIMPE 2018\].
The **tested model reduction methods**:

<table>
<thead>
<tr>
<th>Method</th>
<th>Gramians</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured POD, via:</td>
<td>empirical reachability Gramian</td>
</tr>
<tr>
<td>Structured Dominant Subspaces, via:</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
<tr>
<td>Structured Balanced POD, via:</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td>Structured Balanced Truncation, via:</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
<tr>
<td>Structured Balanced Gains, via:</td>
<td>empirical reachability &amp; observability Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical cross Gramian</td>
</tr>
<tr>
<td></td>
<td>empirical non-symmetric cross Gramian</td>
</tr>
<tr>
<td>Structured DMD Galerkin, via:</td>
<td>empirical reachability Gramian</td>
</tr>
</tbody>
</table>

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

© benner@mpi-magdeburg.mpg.de

Reduced-order Modelling and Simulation of Gas Transportation Networks 22/34
Plain Vanilla DMD:

\[
X = \begin{bmatrix} x_0 & x_1 & \ldots & x_T \end{bmatrix} \rightarrow \begin{cases} X_0 := \begin{bmatrix} x_0 & x_1 & \ldots & x_{T-1} \end{bmatrix} \\ X_1 := \begin{bmatrix} x_1 & x_2 & \ldots & x_T \end{bmatrix} \end{cases} \rightarrow X_1 \approx AX_0 \Rightarrow A \approx X_1X_0^+ 
\]
Plain Vanilla DMD:

\[ X = \begin{bmatrix} x_0 & x_1 & \ldots & x_T \end{bmatrix} \rightarrow \left\{ \begin{aligned}
X_0 &:= \begin{bmatrix} x_0 & x_1 & \ldots & x_{T-1} \end{bmatrix} \\
X_1 &:= \begin{bmatrix} x_1 & x_2 & \ldots & x_T \end{bmatrix}
\end{aligned} \right\} \rightarrow X_1 \approx AX_0 \Rightarrow A \approx X_1 X_0^+ \]

\ldots \text{with centering}^4

\[ X \rightarrow \overline{X} := \begin{bmatrix} (x_0 - \bar{x}) & (x_1 - \bar{x}) & \ldots & (x_T - \bar{x}) \end{bmatrix} \]

---

Plain Vanilla DMD:

\[
X = [x_0 \ x_1 \ \ldots \ x_T] \rightarrow \left\{ X_0 := [x_0 \ x_1 \ \ldots \ x_{T-1}] \right\} \rightarrow X_1 \overset{!}{=} AX_0 \Rightarrow A \approx X_1X_0^+
\]

\ldots \text{with centering}^4

\[
X \rightarrow \overline{X} := [(x_0 - \bar{x}) \ (x_1 - \bar{x}) \ \ldots \ (x_T - \bar{x})]
\]

\ldots \text{used as Model reduction method: DMD-Galerkin}^5

\[
A_{tSVD}^D \coloneqq U_1D_1V_1
\]

---


Plain Vanilla DMD:

\[ X = [x_0 \ x_1 \ \ldots \ x_T] \rightarrow \left\{ \begin{array}{l} X_0 := [x_0 \ x_1 \ \ldots \ x_{T-1}] \\ X_1 := [x_1 \ x_2 \ \ldots \ x_T] \end{array} \right\} \rightarrow X_1 \approx AX_0 \Rightarrow A \approx X_1X_0^+ \]

...with centering

\[ X \rightarrow \overline{X} := [(x_0 - \bar{x}) \ (x_1 - \bar{x}) \ \ldots \ (x_T - \bar{x})] \]

...used as Model reduction method: DMD-Galerkin

\[ A^{tSVD} = U_1 D_1 V_1 \]

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

\[ W_R = \sum_{m=1}^{M} \kappa(\overline{X}_m, \overline{X}_m) \left\{ \begin{array}{l} \kappa_{\text{Linear}}(X, Y) := XY^T \\ \kappa_{\text{DMD}}(X, Y) := X_1Y_0^+ \end{array} \right\} \]

---


Plain Vanilla DMD:

\[ X = [x_0 \ x_1 \ \ldots \ \ x_T] \rightarrow \begin{cases} 
X_0 := [x_0 \ x_1 \ \ldots \ \ x_{T-1}] \\
X_1 := [x_1 \ x_2 \ \ldots \ \ x_T]
\end{cases} \rightarrow X_1 \approx AX_0 \Rightarrow A \approx X_1X_0^+ \\
\ldots \text{with centering}^4 \\
\ldots \text{used as Model reduction method: DMD-Galerkin}^5 \\
A^{tSVD} = U_1D_1V_1 \\
\ldots \text{can be computed via empirical Gramian (exact-DMD ”kernel”):} \\
W_R = \sum_{m}^{M} \kappa(X_m, X_m) \\
\left\{ \begin{array}{l}
\kappa_{\text{Linear}}(X,Y) := XX^T \\
\kappa_{\text{DMD}}(X,Y) := X_1Y_0^+
\end{array} \right. \\
\rightarrow (\text{Centered) DMD-Galerkin via (Discrete) Empirical Reachability Gramian!} \\

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 nonlinearities?
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 nonlinearities?

→ No hyper-reduction implemented (yet).
Major modules:
- networks
- models
- solvers
- reductors
- tests

Minor modules:
- utils
- tools
Set-up
Set-up

- Short training, long testing
Set-up

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

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

Set-up

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

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
Workflow

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
  - pod_r
  - eds_ro, eds.wx, eds.wz
  - bpod_ro,
  - ebt_ro, ebt.wx, ebt.wz
  - ebg_ro, ebg.wx, ebg.wz
  - dmd_r,


© benner@mpi-magdeburg.mpg.de Reduced-order Modelling and Simulation of Gas Transportation Networks 26/34
Workflow

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
  - 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
Workflow

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
  - pod_r
  - eds_ro, eds_wx, eds_wz
  - bpod_ro,
  - ebt_ro, ebt_wx, ebt_wz
  - ebg_ro, ebg_wx, ebg_wz
  - dmd_r,

- Tested reductors:

- Heuristic $L_{i \in \{1,2,\infty\}} \otimes L_{j \in \{1,2,\infty\}}$ error norm computation

- Compare **MORscore**

---

Experiment I: MORGEN Network

- 2 Cycles
- 1 Compressor
- 2 Supply nodes
- 4 Demand nodes
- Pipe length \([20, 60]\text{km}\)
- Time resolution 60s
- Temperature: \([0, 15]^{\circ}\text{C}\)
- 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

 ode_mid--imex1

 ode_end--imex1

Reduced-order Modelling and Simulation of Gas Transportation Networks
## Experiment II: Evaluation

<table>
<thead>
<tr>
<th></th>
<th>ode_mid</th>
<th>ode_end</th>
<th>ode_mid</th>
<th>ode_end</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>imex_1</td>
<td>imex_1</td>
<td>imex_2</td>
<td>imex_2</td>
</tr>
<tr>
<td>pod_r</td>
<td>0.12</td>
<td>0.12</td>
<td>0.04</td>
<td>0.05</td>
</tr>
<tr>
<td>eds_ro</td>
<td>0.16</td>
<td>0.16</td>
<td>0.05</td>
<td>0.06</td>
</tr>
<tr>
<td>eds.wx</td>
<td>0.08</td>
<td>0.08</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>eds.wx</td>
<td>0.03</td>
<td>0.07</td>
<td>0.02</td>
<td>0.04</td>
</tr>
<tr>
<td>bpo_ro</td>
<td>0.07</td>
<td>0.07</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>ebt_ro</td>
<td>0.00</td>
<td>0.00</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>ebt.wx</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>ebt.wz</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>ebg_ro</td>
<td>0.00</td>
<td>0.01</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>ebg.wx</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>ebg.wz</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>dmd_r</td>
<td>0.14</td>
<td>0.18</td>
<td>0.03</td>
<td>0.04</td>
</tr>
</tbody>
</table>

**MORscores** $\mu(150, \epsilon_{\text{mach}(16)})$ in the $L_2 \otimes L_2$ norm for the “MORGEN” network.
Experiment III: GasLib-134v2

The Network
- total length: 1412km
- 1 compressor

The Scenarios
- 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
Experiment III: $L_2 \otimes L_2$ Model Reduction Error

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.
Conclusions from computational experiments:

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

The Paper


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