MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC

COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

# Accelerating Gas Network Simulations

**C. Himpe, S. Grundel, P. Benner**

CSC Seminar

2020–10–27

**Supported by:**

Federal Ministry
for Economic Affairs
and Energy

# Outline

**We will do some gas network ...**

# Outline

**We will do some gas network ...**

- Modeling

# Outline

**We will do some gas network ...**

- Modeling
- Model simplification

# Outline

**We will do some gas network ...**

- Modeling
- Model simplification
- Model discretization

# Outline

**We will do some gas network ...**

- Modeling
- Model simplification
- Model discretization
- Model reduction

# Outline

**We will do some gas network ...**

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

# About

**Why accelerate gas network simulations?**

# About

**Why accelerate gas network simulations?**

- MathEnergy project

# About

**Why accelerate gas network simulations?**

- MathEnergy project
- Renewable energy transition

# About

**Why accelerate gas network simulations?**

- MathEnergy project
- Renewable energy transition
- Modernization of control

# About

**Why accelerate gas network simulations?**

- MathEnergy project
- Renewable energy transition
- Modernization of control
- Modern numerics

# About

**Why accelerate gas network simulations?**

- MathEnergy project
- Renewable energy transition
- Modernization of control
- Modern numerics
- It's a challenge

# Give Me Numbers

**Some gas network properties:**

1
2
3

# Give Me Numbers

**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

# Give Me Numbers

**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

# Give Me Numbers

**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

# Give Me Numbers

**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.

---

[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

# Give Me Numbers

**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 $24$h.

[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

# 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.: $g$

# Modeling II: Network

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

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

# Modeling II: Network

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

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

**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.

# Modeling III: Gas Network

**Vectorized PDAE gas network model:**

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

- $p^*$ 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 $q^*$:**

[4] S. Grundel, L. Jansen, N. Hornung, T. Clees, C. Tischendorf, P. Benner. **Model order reduction of differential algebraic equations arising from the simulation of gas transport networks**. In: Progress in Differential-Algebraic Equations, Differential-Algebraic Equations Forum: 183–205, 2014. 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[4].
  - Boundary values affect friction term.

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

# Simplification I: Index

**The choice of $p^*$ and $q^*$:**

- Pipe midpoints:
  - (P)DAE tractability index bounded $\tau \leq 2$.
  - Given some weak topology constraints, PDAE becomes PDE[4].
  - 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.

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

# Simplification II: Model

**You were correct to assume:**

# Simplification II: Model

**You were correct to assume:**

- Only cylindrical pipes.

# Simplification II: Model

**You were correct to assume:**

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

## Simplification II: Model

**You were correct to assume:**

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

# Simplification II: Model

**You were correct to assume:**

- 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: $-\frac{\gamma_0}{S^2}\left(\frac{q^2}{p}\right)_x \approx 0$.

# Simplification II: Model

**You were correct to assume:**

- 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: $\gamma_0 = (T_0 \;\; R_S)$.

# Simplification II: Model

**You were correct to assume:**

- 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

**You were correct to assume:**

- 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

**Simplified edge-based compressor models:**

**Simplified edge-based compressor models:**

- Energy-based:

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

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

# Simplification III: Compressors

**Simplified edge-based compressor models:**

- Energy-based:

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

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

- Multiplicative:

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

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

# Simplification III: Compressors

**Simplified edge-based compressor models:**

- Energy-based:

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

- 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 Finite Differences:**

# Discretization I: Space

**1D First-Order Finite Differences:**

- Axis-symmetric domain.

**1D First-Order Finite Differences:**

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

# Discretization I: Space

**1D First-Order Finite Differences:**

- Axis-symmetric domain.
- Pipelines length exceeds diameter by orders of magnitude.
- (Very) turbulent flow, $\mathrm{Re} > 100,000$.

# Discretization I: Space

**1D First-Order Finite Differences:**

- Axis-symmetric domain.
- Pipelines length exceeds diameter by orders of magnitude.
- (Very) turbulent flow, $\mathrm{Re} > 100,000$.
- Stable under CLF condition.

**Time-aware spatial discretization:**

# Discretization II: Space-Time

**Time-aware spatial discretization:**

- Set unit pipeline length based on CLF condition.

# Discretization II: Space-Time

**Time-aware spatial discretization:**

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

# Discretization II: Space-Time

**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 II: Space-Time

**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.

# Discretization III: Time

**If you only have a hammer ...**

# Discretization III: Time

**If you only have a hammer ...**

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

# Discretization III: Time

**If you only have a hammer ...**

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

# Discretization III: Time

**If you only have a hammer ...**

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

# Discretization III: Time

**If you only have a hammer ...**

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

# The Input-Output Model

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

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

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

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

**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: $A_{qp} p_{k+1} = -\Big(B_{qs}\bar{s}_p + F_c + f_q(p_k, \bar{q}, \bar{s}_p, \theta)\Big)$

# Sidebar: Steady-State

**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!

# Sidebar: Steady-State

**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 QR-least-norm (in parallel).

# Sidebar: Steady-State

**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 QR-least-norm (in parallel).
- Repeat Step 2 until happy (reuse QR of Step 1b).

# Sidebar: Steady-State

**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 QR-least-norm (in parallel).
- Repeat Step 2 until happy (reuse QR of Step 1b).
- Repeating Step 2 is a special case of an IMEX solver.

# Sidebar: Steady-State

**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 QR-least-norm (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.

# Sidebar: Steady-State

**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 QR-least-norm (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.
- Practically, $z_0$ is also computed in Step 2.

**Recap:**

From: Hyperbolic 2D PDAE

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

**Shopping list:**

**Recap:**

From: Hyperbolic 2D PDAE

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

**Shopping list:**

- Perturbation system $\rightarrow$ Deviation from steady state

**Recap:**

From: Hyperbolic 2D PDAE

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

**Shopping 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

**Shopping list:**

- Perturbation system $\rightarrow$ Deviation from steady state
- Input-output system $\rightarrow$ System-theoretic methods
- Coupled system $\rightarrow$ Structure-preserving methods

**Recap:**

From: Hyperbolic 2D PDAE
To: Non-normal, coupled, nonlinear, parametric ODE

**Shopping 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

**Shopping 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
- Large-scale $\rightarrow$ Low-rank computable methods*

# Model Reduction II: Tested Methods

**The reducing dozen:**

# Model Reduction II: Tested Methods

## The reducing dozen:

| | |
|---|---|
| Structured POD, via: | empirical reachability Gramian |

## The reducing dozen:

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

## The reducing dozen:

| | |
|---|---|
| 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 |

## The reducing dozen:

| | |
|---|---|
| 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 |

# Model Reduction II: Tested Methods

## The reducing dozen:

| | |
|---|---|
| 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 |

# Model Reduction II: Tested Methods

## The reducing dozen:

| | |
|---|---|
| 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 |

# Spotlight: DMD-Galerkin

**Plain Vanilla DMD:**

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

5

6

**Plain Vanilla DMD:**

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

**Secret Sauce: Centering**[5]

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

[5] 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

[6]

# Spotlight: DMD-Galerkin

**Plain Vanilla DMD:**

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

**Secret Sauce: Centering**[5]

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

**Magic Dust: (Almost) DMD-Galerkin**[6]

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

[5] 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

[6] 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

# Spotlight: DMD-Galerkin

**Plain Vanilla DMD:**

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

**Secret Sauce: Centering**[5]

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

**Magic Dust: (Almost) DMD-Galerkin**[6]

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

**Cherry on top: Exact-DMD-Pseudo-Kernel**

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

[5] 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

[6] 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:**

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

**Secret Sauce: Centering**[5]

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

**Magic Dust: (Almost) DMD-Galerkin**[6]

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

**Cherry on top: Exact-DMD-Pseudo-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!**

[5] 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

[6] 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

**Yes, yes, yes:**

# Model Reduction III: Hyper Reduction

**Yes, yes, yes:**

- First, what is the best reduced order linear subspace?

# Model Reduction III: Hyper Reduction

**Yes, yes, yes:**

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

**Yes, yes, yes:**

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

**Yes, yes, yes:**

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

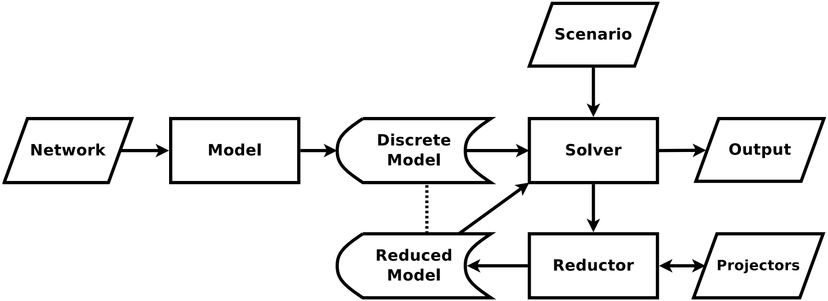
**Yes, yes, yes:**

- First, what is the best reduced order linear subspace?
- 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?

**Yes, yes, yes:**

- First, what is the best reduced order linear subspace?
- 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 (yet).




## Major modules:

- `networks`
- `models`
- `solvers`
- `reductors`
- `tests`

## Minor modules:

- `utils`
- `tools`

# Workflow

## Setup.exe

7

# Workflow

## Setup.exe

- Short training, long testing

7

# Workflow

**Setup.exe**

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

7

# Workflow

## Setup.exe

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

7

# Workflow

**Setup.exe**

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

7

# Workflow

## Setup.exe

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

7

# Workflow

## Setup.exe

- 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`,

---

7

# Workflow

## Setup.exe

- 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

7

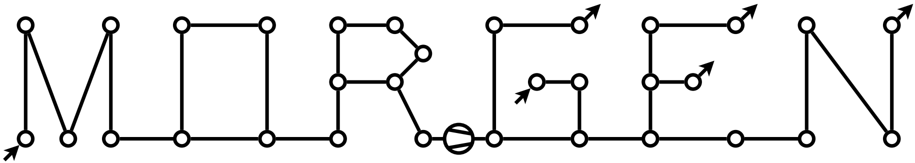
# Workflow

## Setup.exe

- 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[7]

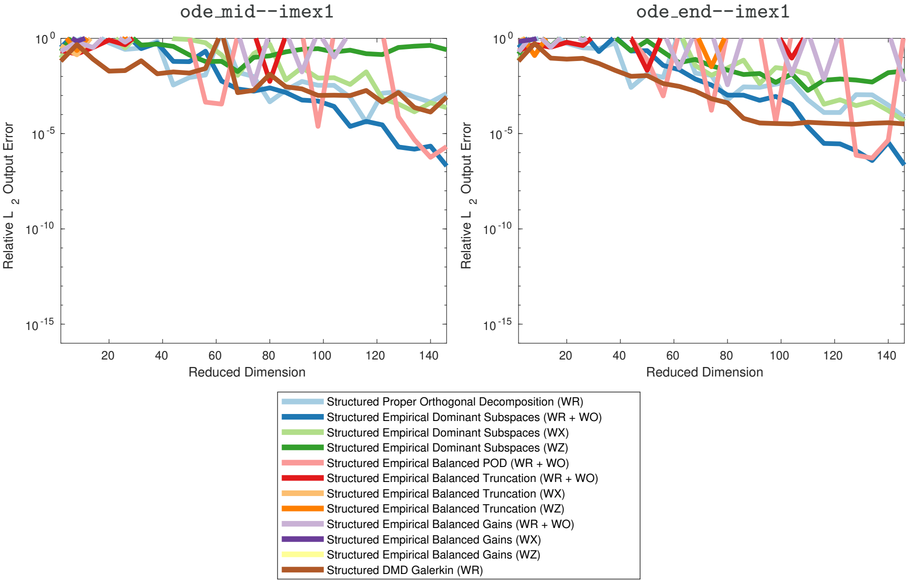
[7] C. Himpe. **Comparing (empirical-Gramian-based) model order reduction algorithms**. arXiv, math.OC: 2002.12226, 2020. https://arxiv.org/abs/2002.12226

# Experiment I: MORGEN Network



- 2 Cycles
- 1 Compressor
- 2 Supply nodes
- 4 Demand nodes
- Pipe length $[20, 60]$km
- Time resolution $60$s
- 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: $1$h
- Test horizon: $24$h
- Perturbed steady-state training
- Standard load profiles testing

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



# Experiment III: 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.

# Outlook

**Open problems and future work:**

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

# Summary

**How to accelerate gas network simulations:**

- Prefer the endpoint model.
- Prefer the first-order IMEX solver.
- Prefer Galerkin model reduction methods.
- Check out `morgen` (Model Order Reduction for Gas and Energy Networks).

https://himpe.science

**Acknowledgment:**
Supported by the German Federal Ministry for Economic Affairs and Energy, in the joint project: "**MathEnergy** – Mathematical Key Technologies for Evolving Energy Grids", sub-project: Model Order Reduction (Grant number: 0324019**B**).

# Selected Articles


1. Z. Vostrý, J. Záworka. **Simulation and control of gas transport and distribution by large-scale pipeline networks**. In: Mutual Impact of Computing Power and Control Theory: 65–75, 1993. doi:10.1007/978-1-4615-2968-2_4
2. T.P. Azevedo-Perdicoúlis, G. Jank. **Modelling aspects of describing gas networks through a DAE system**. IFAC Proceedings Volume (3rd IFAC Symposium on Structure and Control), 40(20): 40–45, 2007. doi:10.3182/20071017-3-BR-2923.00007
3. S. Grundel, L. Jansen, N. Hornung, T. Clees, C. Tischendorf, P. Benner. **Model order reduction of differential algebraic equations arising from the simulation of gas transport networks**. In: Progress in Differential-Algebraic Equations, Differential-Algebraic Equations Forum: 183–205, 2014. doi:10.1007/978-3-662-44926-4_9
4. 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