MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Reduced-order Modeling Enables Digital Twins for Smart Process Engineering

Peter Benner

Smart Process Engineering 2022
Online Symposium
"Towards Sustainable & Circular Production Processes"
2–4 March 2022

Supported by:

DFG-Graduiertenkolleg
MATHEMATISCHE
KOMPLEXITÄTSREDUKTION

SmartProSys

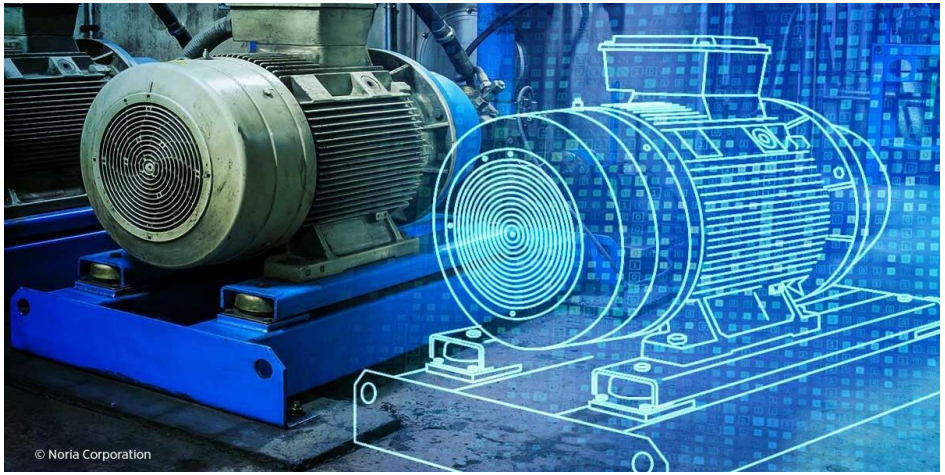
Partners:

Digital Twins

Digital twins provide a virtual model of a real-world object / device / process.

They allow

- the simulation for analyzing behavior,
- optimizing design and control synthesis,
- surveillance and prediction,
- continuous improvement of the plant model and its controller.



©Noria Corporation

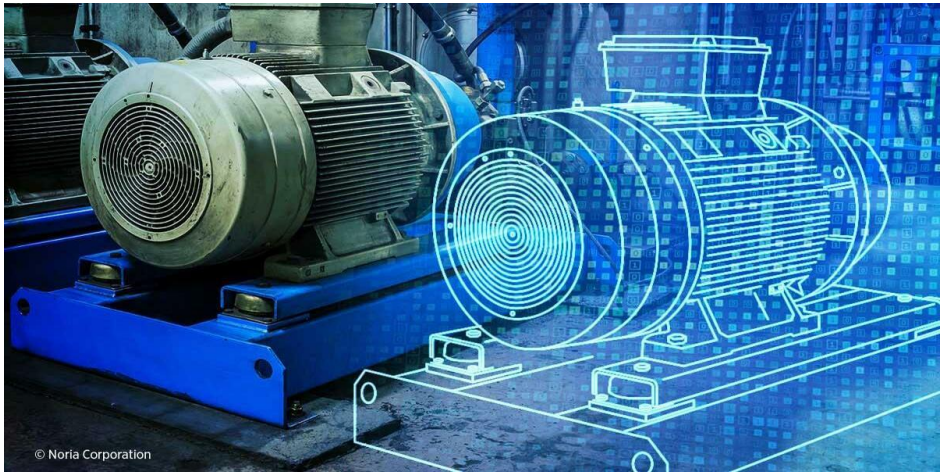
Digital Twins

Digital twins provide a virtual model of a real-world object / device / process.

They allow

- the simulation for analyzing behavior,
- optimizing design and control synthesis,
- surveillance and prediction,
- continuous improvement of the plant model and its controller.

The latter require real-time response times!



©Noria Corporation

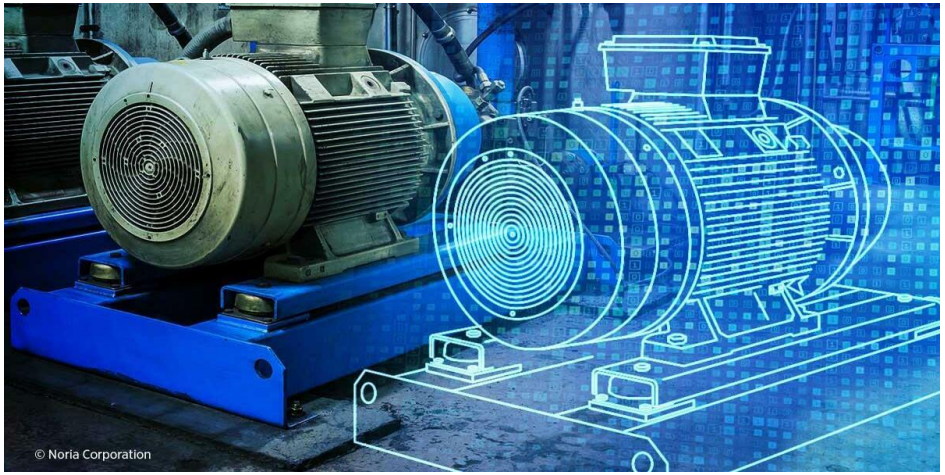
Digital Twins

Digital twins provide a virtual model of a real-world object / device / process.

They allow

- the simulation for analyzing behavior,
- optimizing design and control synthesis,
- surveillance and prediction,
- continuous improvement of the plant model and its controller.

The latter require real-time response times!



©Noria Corporation

Smart process engineering (SmartProSys) requires digital twins of, e.g., chemical reactors.

- This involves mathematical models (mass and energy balances, reaction kinetics,...).
- For high precision, this involves accurate discretizations of systems of nonlinear coupled partial differential equations.
- Real-time demands (but also, optimization and controller design) require fast-to-evaluate surrogate models.

©Emerson Electric Co.

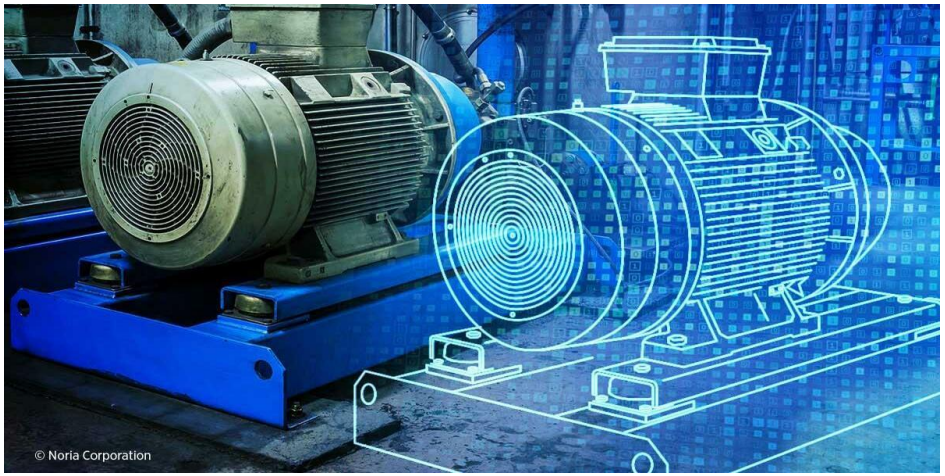
Digital Twins

Digital twins provide a virtual model of a real-world object / device / process.

They allow

- the simulation for analyzing behavior,
- optimizing design and control synthesis,
- surveillance and prediction,
- continuous improvement of the plant model and its controller.

The latter require real-time response times!



©Noria Corporation

Smart process engineering (SmartProSys) requires digital twins of, e.g., chemical reactors.

©Emerson Electric Co.

- This involves mathematical models (mass and energy balances, reaction kinetics,...).
- For high precision, this involves accurate discretizations of systems of nonlinear coupled partial differential equations.
- Real-time demands (but also, optimization and controller design) require fast-to-evaluate surrogate models.

⇝ **MODEL REDUCTION is enabling technology!**

Original System

$$\Sigma : \begin{cases} \dot{x}(t) &= f(t, x(t), u(t)), \\ y(t) &= g(t, x(t), u(t)), \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Model Order Reduction of Dynamical Systems

Original System

$$\Sigma : \begin{cases} \dot{x}(t) &= f(t, x(t), u(t)), \\ y(t) &= g(t, x(t), u(t)), \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Reduced-Order Model (ROM)

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) &= \widehat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) &= \widehat{g}(t, \hat{x}(t), u(t)), \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.

Model Order Reduction of Dynamical Systems

Original System

$$\Sigma : \begin{cases} \dot{x}(t) &= f(t, x(t), u(t)), \\ y(t) &= g(t, x(t), u(t)), \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Reduced-Order Model (ROM)

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) &= \widehat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) &= \widehat{g}(t, \hat{x}(t), u(t)), \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.

Goals:

$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\|$ for all admissible input signals.

Model Order Reduction of Dynamical Systems

Original System

$$\Sigma : \begin{cases} \dot{x}(t) &= f(t, x(t), u(t)), \\ y(t) &= g(t, x(t), u(t)), \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Reduced-Order Model (ROM)

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) &= \widehat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) &= \widehat{g}(t, \hat{x}(t), u(t)), \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.

Goals:

$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\|$ for all admissible input signals.

Secondary goal: reconstruct approximation of x from $\hat{x}$.

Original System

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t). \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Reduced-Order Model (ROM)

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) = \widehat{A}\hat{x}(t) + \widehat{B}u(t), \\ \hat{y}(t) = \widehat{C}\hat{x}(t) + \widehat{D}u(t). \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.

Goals:

$\|y - \hat{y}\| <$ tolerance $\cdot \|u\|$ for all admissible input signals.

Secondary goal: reconstruct approximation of x from $\hat{x}$.

Model Order Reduction of Linear Systems

Model Reduction Schematically

$$E\,\dot{x}(t) = A\,x(t) + B\,u(t)$$

$$y(t) = C\,x(t) + D\,u(t)$$

MOR

$$\hat{E}\,\dot{\hat{x}}(t) = \hat{A}\,\hat{x}(t) + \hat{B}\,u(t)$$

$$\hat{y}(t) = \hat{C}\,\hat{x}(t) + \hat{D}\,u(t)$$

- $E, A \in \mathbb{R}^{n\times n}$
- $B \in \mathbb{R}^{n\times m}$
- $C \in \mathbb{R}^{p\times n}$
- $D \in \mathbb{R}^{p\times m}$
- $x(t) \in \mathbb{R}^{n}$

- $\hat{E}, \hat{A} \in \mathbb{R}^{r\times r}$
- $\hat{B} \in \mathbb{R}^{r\times m}$
- $\hat{C} \in \mathbb{R}^{p\times r}$
- $\hat{D} \in \mathbb{R}^{p\times m}$
- $\hat{x}(t) \in \mathbb{R}^{r}, \quad r \ll n$

MOR Methods Based on Projection

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.

MOR Methods Based on Projection

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

MOR Methods Based on Projection

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

Then, with $\hat{x} = W^T x$, we obtain $x \approx V\hat{x} = VW^T x =: \tilde{x}$ so that

$$\|x - \tilde{x}\| = \|x - V\hat{x}\|.$$

MOR Methods Based on Projection

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

Then, with $\hat{x} = W^T x$, we obtain $x \approx V\hat{x} = VW^T x =: \tilde{x}$ so that

$$\|x - \tilde{x}\| = \|x - V\hat{x}\|.$$

For linear systems, the reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T AV, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

MOR Methods Based on Projection

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

Then, with $\hat{x} = W^T x$, we obtain $x \approx V\hat{x} = VW^T x =: \tilde{x}$ so that

$$\|x - \tilde{x}\| = \|x - V\hat{x}\|.$$

For linear systems, the reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T A V, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

Extends to nonlinear systems with some effort:

$$\begin{aligned} \dot{\hat{x}} &= W^T f(t, V\hat{x}, u), \\ \hat{y} &= g(t, V\hat{x}, u). \end{aligned}$$

Needs hyper-reduction if the cost for evaluation of the functions $W^T f, g$ is not reduced!

MOR Methods Based on Projection

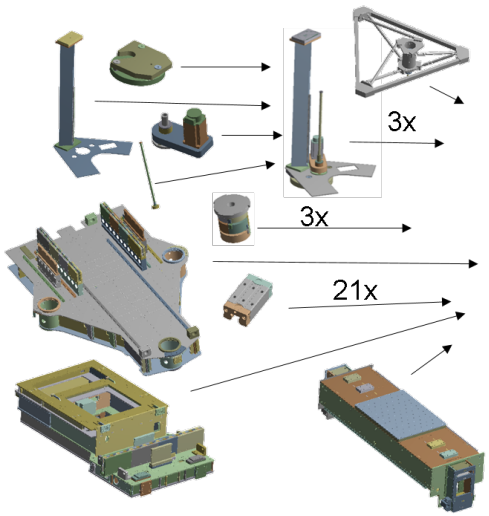
Classes of Projection-based MOR Methods

1. Modal Truncation
2. Rational Interpolation / Moment Matching
   (Padé-Approximation and (rational) Krylov Subspace Methods)
3. Balanced Truncation
4. Proper Orthogonal Decomposition (POD) / Principal Component Analysis (PCA)
5. Reduced Basis Method
6. ...

MOR Methods Based on Projection

Example: Virtual twin of experimental machine tool MAX

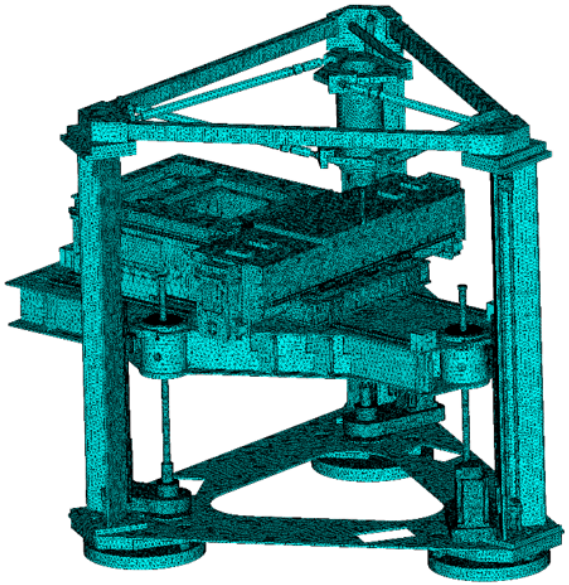
50 subassemblies



CAD model

$\overset{FEM}{\rightsquigarrow}$

FE-Model: 1.2M DOFs



MAX: Results considering an inhomogeneous initial condition $T_0 \neq 0$

Results by Julia Vettermann (MilT/TU Chemnitz)

FE-coupled

method	red. order tol 10^{-3}	t_{red}
2phase	196	6.5h
BTX0	174	4.5h

output-coupled

method	red. order tol 10^{-3}	t_{red}
2phase	3005	2h
BTX0	2515	1.8h

MAX: Results considering an inhomogeneous initial condition $T_0 \neq 0$

Results by Julia Vettermann (MilT/TU Chemnitz)

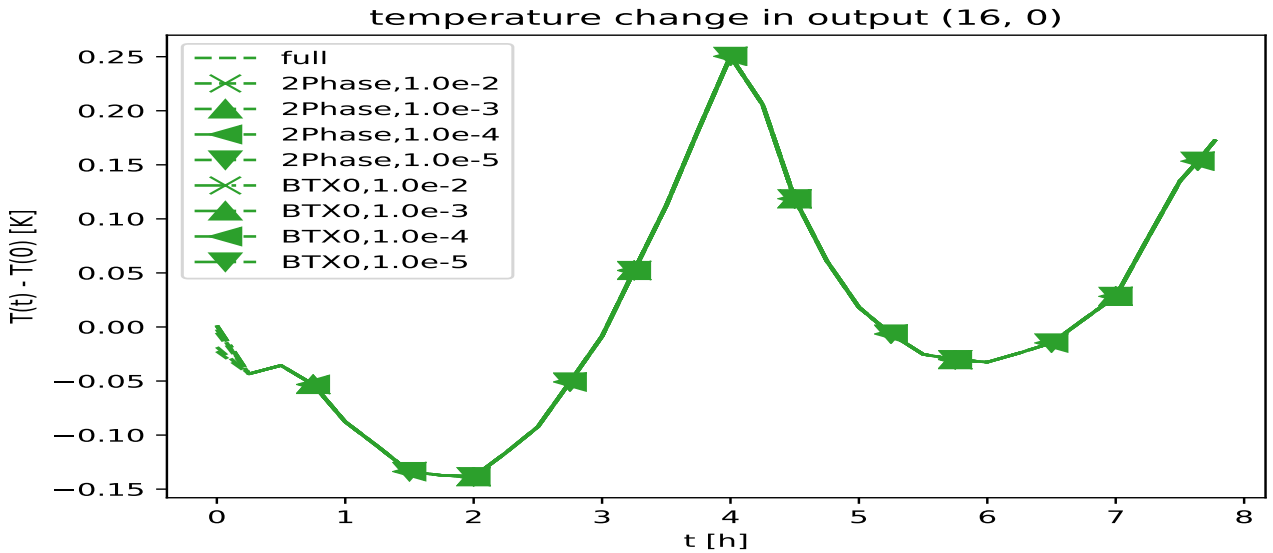
FE-coupled

method	red. order tol 10^{-3}	t_{red}
2phase	196	6.5h
BTX0	174	4.5h

output-coupled

method	red. order tol 10^{-3}	t_{red}
2phase	3005	2h
BTX0	2515	1.8h

→ *Required storage for reduced matrices just 1MB!*
→ *Simulation speed-up factors range from* ≈ 8–$2,000$.



Vettermann, J., Sauerzapf, S., Naumann, A., Beitelschmidt, M., Herzog, R., Benner, P., Saak, J. (2021): Model order reduction methods for coupled machine tool models. MM Science Journal, pp. 4652–4659.

What about the Data?

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

The reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T A V, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

What about the Data?

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

The reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T A V, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

We need the matrices A, B, C, D to compute the reduced-order model!

What about the Data?

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

The reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T A V, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

We need the matrices A, B, C, D to compute the reduced-order model!

Using proprietary simulation software, we would need to **intrude** the software to get the matrices

$\rightsquigarrow$ **intrusive MOR**

$=$ *learning (compact, surrogate) models from (full, detailed) models.*

This is often impossible!

What about the Data?

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

The reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T A V, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

We need the matrices A, B, C, D to compute the reduced-order model!

Using proprietary simulation software, we would need to **intrude** the software to get the matrices

$\rightsquigarrow$ **intrusive MOR**

$=$ *learning (compact, surrogate) models from (full, detailed) models.*

This is often impossible!

$\rightsquigarrow$ **non-intrusive MOR**

$=$ *Learning (compact, surrogate) models from data!*

Now assume we are only given an oracle, allowing us to evaluate Σ, given $u(t)$ or $U(s)$:

Now assume we are only given an oracle, allowing us to evaluate Σ, given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or

Now assume we are only given an oracle, allowing us to evaluate Σ, given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Now assume we are only given an oracle, allowing us to evaluate Σ, given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, ...]

Now assume we are only given an oracle, allowing us to evaluate Σ, given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, ...]
- Neural networks: time domain [Narendra/Parthasarathy 1990; Lee/Carlberg 2019; ...]

Now assume we are only given an oracle, allowing us to evaluate Σ, given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, ...]
- Neural networks: time domain [Narendra/Parthasarathy 1990; Lee/Carlberg 2019; ...]
- Loewner interpolation: frequency and time domain
  [Antoulas/Anderson 1986; Mayo/Antoulas 2007; Gosea, Gugercin, Ionita, Lefteriu, Peherstorfer, ...]

Now assume we are only given an oracle, allowing us to evaluate Σ, given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, ...]
- Neural networks: time domain [Narendra/Parthasarathy 1990; Lee/Carlberg 2019; ...]
- Loewner interpolation: frequency and time domain
  [Antoulas/Anderson 1986; Mayo/Antoulas 2007; Gosea, Gugercin, Ionita, Lefteriu, Peherstorfer, ...]
- Koopman/Dynamic Mode Decomposition (DMD): time domain
  [Mezič 2005; Schmid 2008; Brunton, Kevrekidis, Kutz, Rowley, Noé, Nüske, Schütte, Peitz, ...],
  for control systems [Kaiser/Kutz/Brunton 2017, B./Himpe/Mitchell 2018]

Now assume we are only given an oracle, allowing us to evaluate Σ, given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

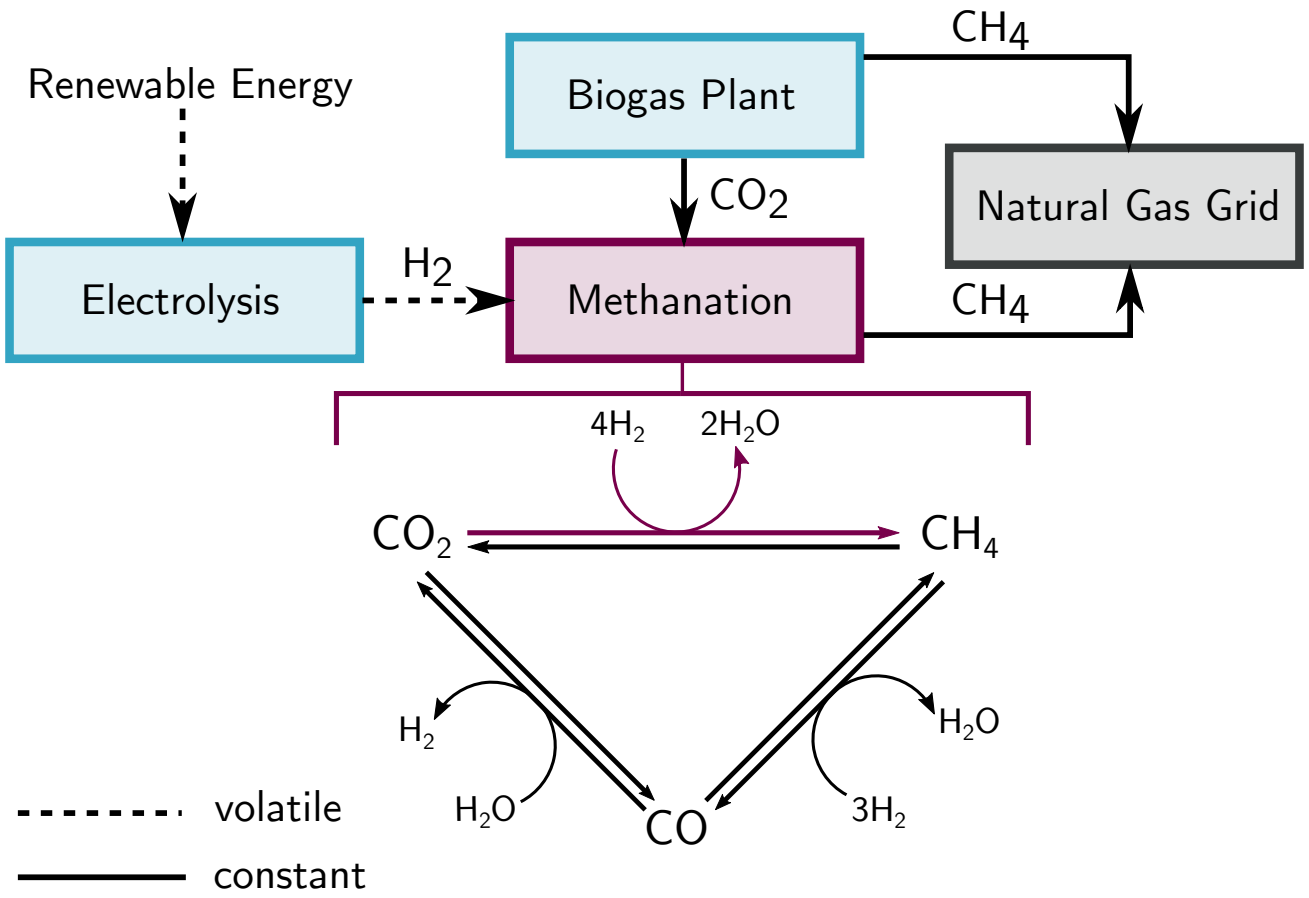
- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, ...]
- Neural networks: time domain [Narendra/Parthasarathy 1990; Lee/Carlberg 2019; ...]
- Loewner interpolation: frequency and time domain
  [Antoulas/Anderson 1986; Mayo/Antoulas 2007; Gosea, Gugercin, Ionita, Lefteriu, Peherstorfer, ...]
- Koopman/Dynamic Mode Decomposition (DMD): time domain
  [Mezič 2005; Schmid 2008; Brunton, Kevrekidis, Kutz, Rowley, Noé, Nüske, Schütte, Peitz, ...],
  for control systems [Kaiser/Kutz/Brunton 2017, B./Himpe/Mitchell 2018]
- Operator inference (OpInf): time domain [Peherstorfer/Willcox 2016; Kramer, Qian, B., Goyal,...]

An Example in Process Engineering

CO_2-Methanation

[Bremer/Goyal/Feng/B./Sundmacher 2017]

Consider Power2Gas process for production of methane employing volatile renewable energy resources:



Source: Jens Bremer (2016)

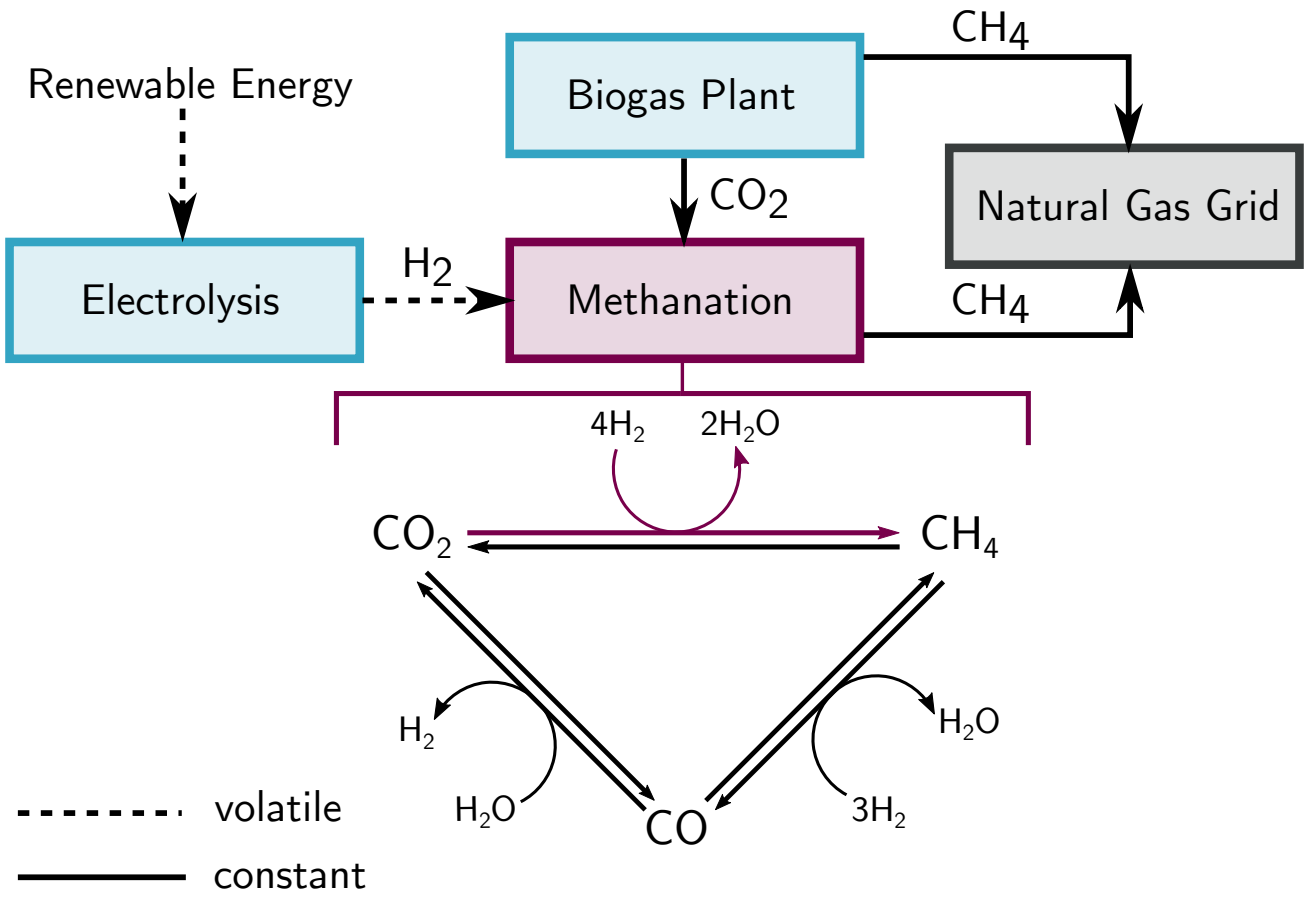
- High energy density
- Use of available infrastructure
- Flexible

An Example in Process Engineering

CO_2-Methanation

[Bremer/Goyal/Feng/B./Sundmacher 2017]

Consider Power2Gas process for production of methane employing volatile renewable energy resources:



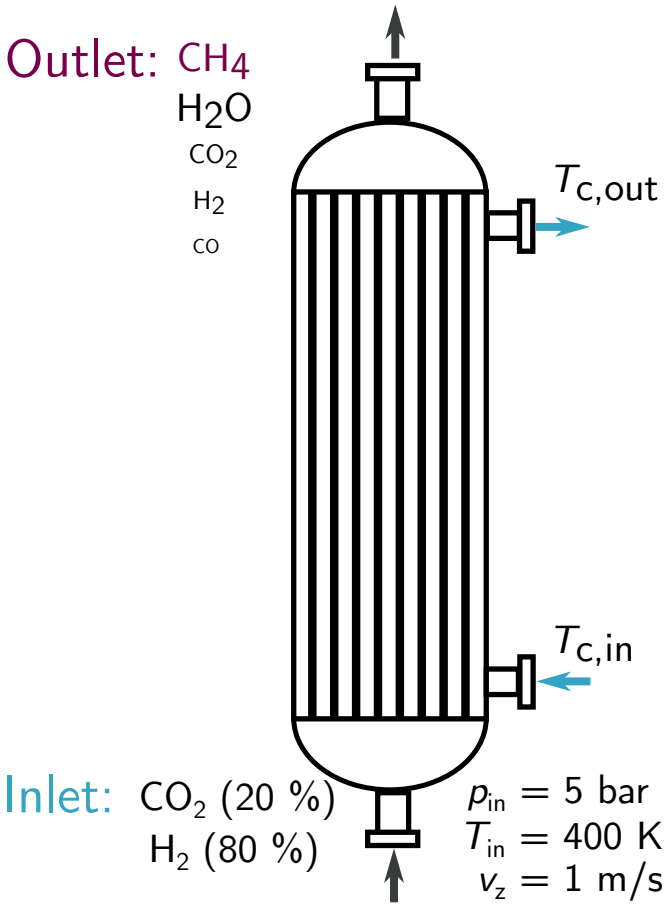
Source: Jens Bremer (2016)

- High energy density
- Use of available infrastructure
- Flexible

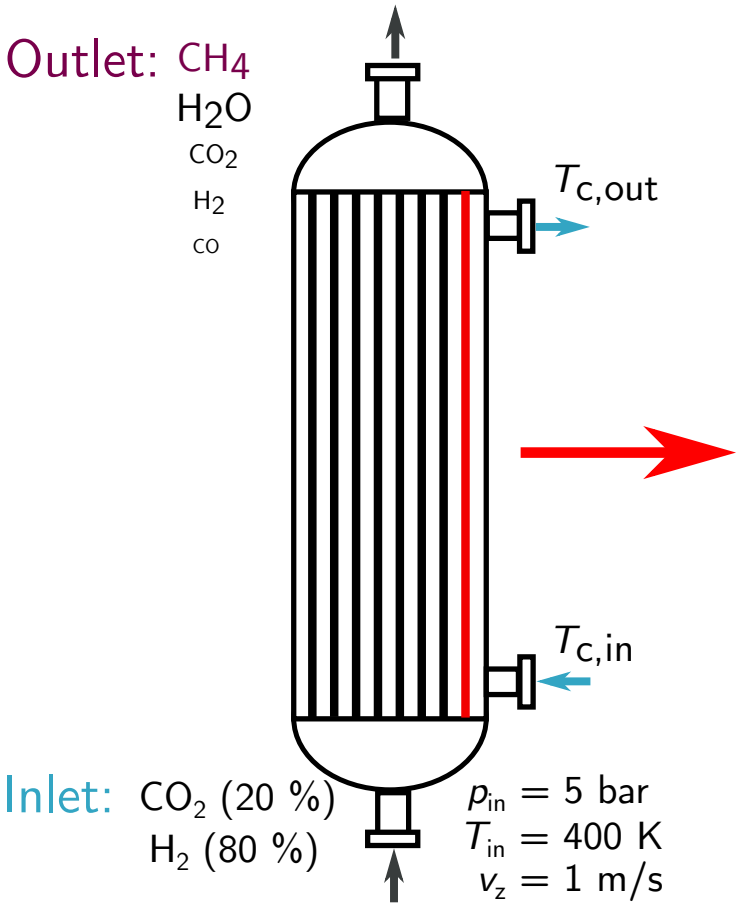
Dynamic Consideration

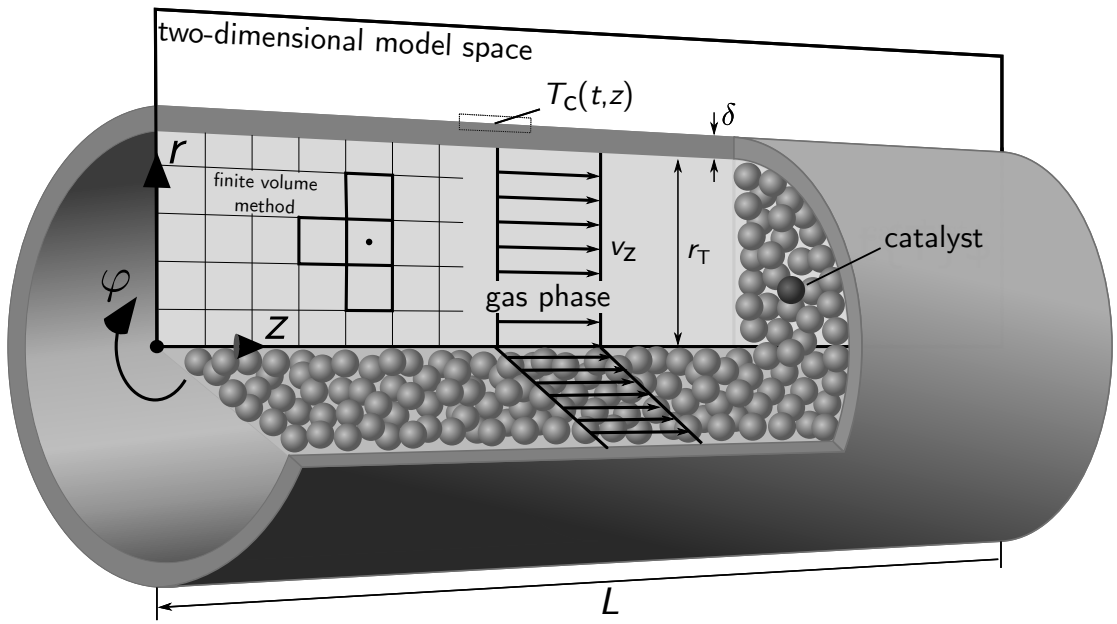
- Steady-state disturbances
- Start-up / Shut-down

Modeling Approach for 2D Methanation Reactor Model



Modeling Approach for 2D Methanation Reactor Model





Source: Jens Bremer (2016)

Reactor tube radius	r_T	=	10 / 45	mm
Reactor tube length	L	=	5	m
Wall thickness	δ	=	20	mm
Catalyst particle diameter	d_{cat}	=	2	mm
Fixed-bed void fraction	ε	=	0.4	

Governing Equations for 2D Pseudo-homogeneous Reactor Model

Mass Balance

$$\frac{\partial \rho_\alpha}{\partial t} = -\frac{v_z}{\varepsilon}\frac{\partial \rho_\alpha}{\partial z} + \frac{\mathcal{D}_{\mathrm{r},i}^{\mathrm{eff}}}{\varepsilon}\left(\frac{\partial^2 \rho_\alpha}{\partial r^2} + \frac{1}{r}\frac{\partial \rho_\alpha}{\partial r}\right) + \frac{1-\varepsilon}{\varepsilon}\tilde{M}_\alpha \sum_{\beta=1}^{3} \nu_{\alpha,\beta}\,\tilde{r}_\beta, \quad \alpha = 1\ldots 6$$

Energy Balance

$$\frac{\partial T}{\partial t} = \frac{1}{(\rho c_\mathrm{p})_\mathrm{eff}}\left[-\rho c_\mathrm{p} v_z \frac{\partial T}{\partial z} + \lambda_{\mathrm{eff,r}}\left(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r}\right) + (1-\varepsilon)\sum_{\beta=1}^{3}\left(-\Delta_\mathrm{R}\tilde{H}_\beta\right)\tilde{r}_j\right]$$

Boundary Conditions (radial)

$$\frac{\partial \rho_\alpha}{\partial r} = 0, \quad \frac{\partial T}{\partial r} = 0 \quad \text{at } r = 0 \qquad \left| \qquad \frac{\partial \rho_\alpha}{\partial r} = 0, \quad \frac{\partial T}{\partial r} = \frac{k_\mathrm{w}}{\lambda_{\mathrm{eff,r}}}(T_\mathrm{c} - T) \quad \text{at } r = r_\mathrm{T}\right.$$

From PDAE to ODE/DAE via Finite Volume Method (FVM)

PDAE:

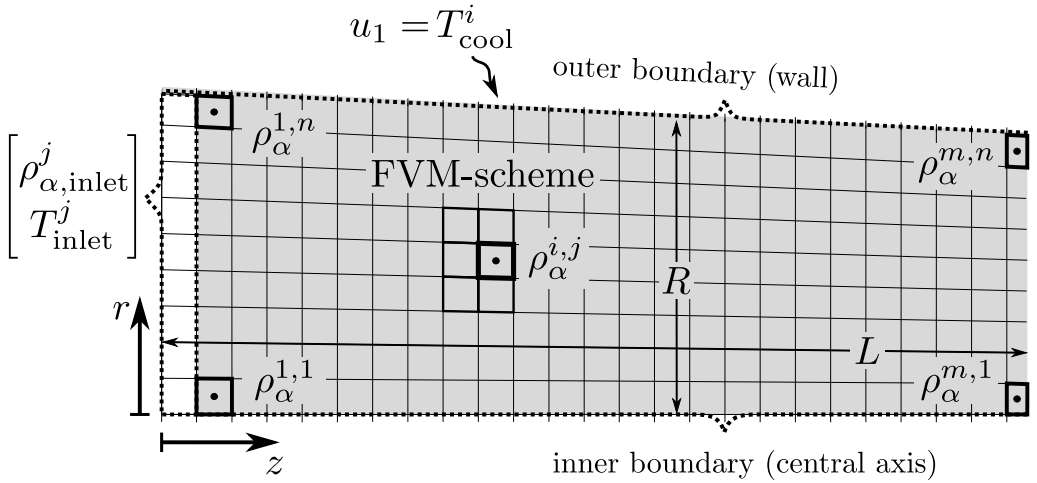
$$\frac{\partial \rho_\alpha}{\partial t} = -\frac{v_z}{\varepsilon}\frac{\partial \rho_\alpha}{\partial z} + \ldots \quad \overset{FVM}{\Longrightarrow}$$

$$\frac{\partial T}{\partial t} = \ldots$$

ODE/DAE:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{w}(t), \mathbf{u}(t))$$
$$0 = \mathbf{w}(t) - \mathbf{d}(\mathbf{x}(t), \mathbf{u}(t))$$

$\mathbf{x}(t)$ - "diff. state vector"
$\mathbf{w}(t)$ - "alg. state vector"
$\mathbf{u}(t)$ - "control vector"
$\mathbf{f}$ - strongly nonlinear RHS

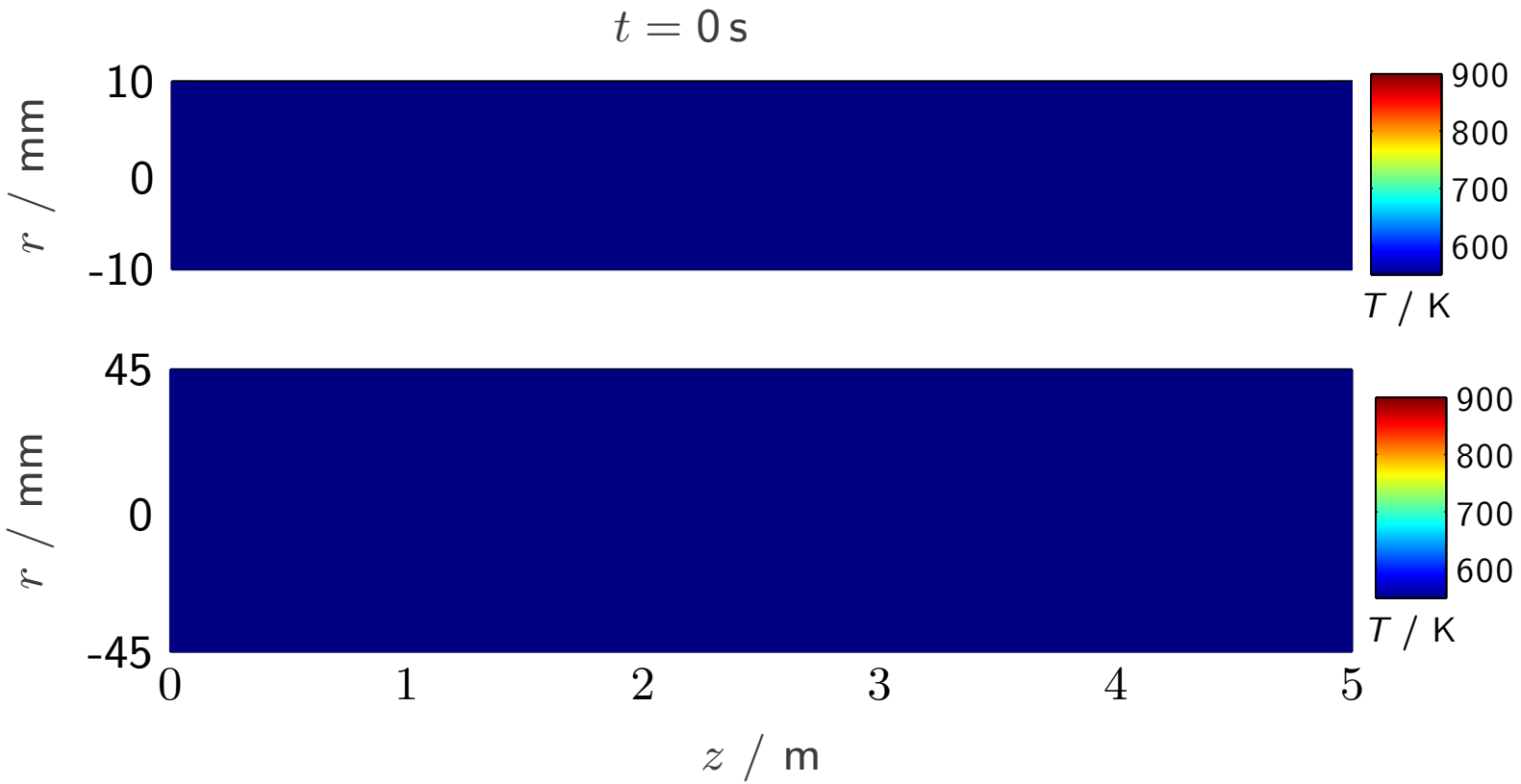


$\dim(\mathbf{x}) = 350 - 5{,}000$ (depending on mesh refinement)

CPU Time:

95.9 s

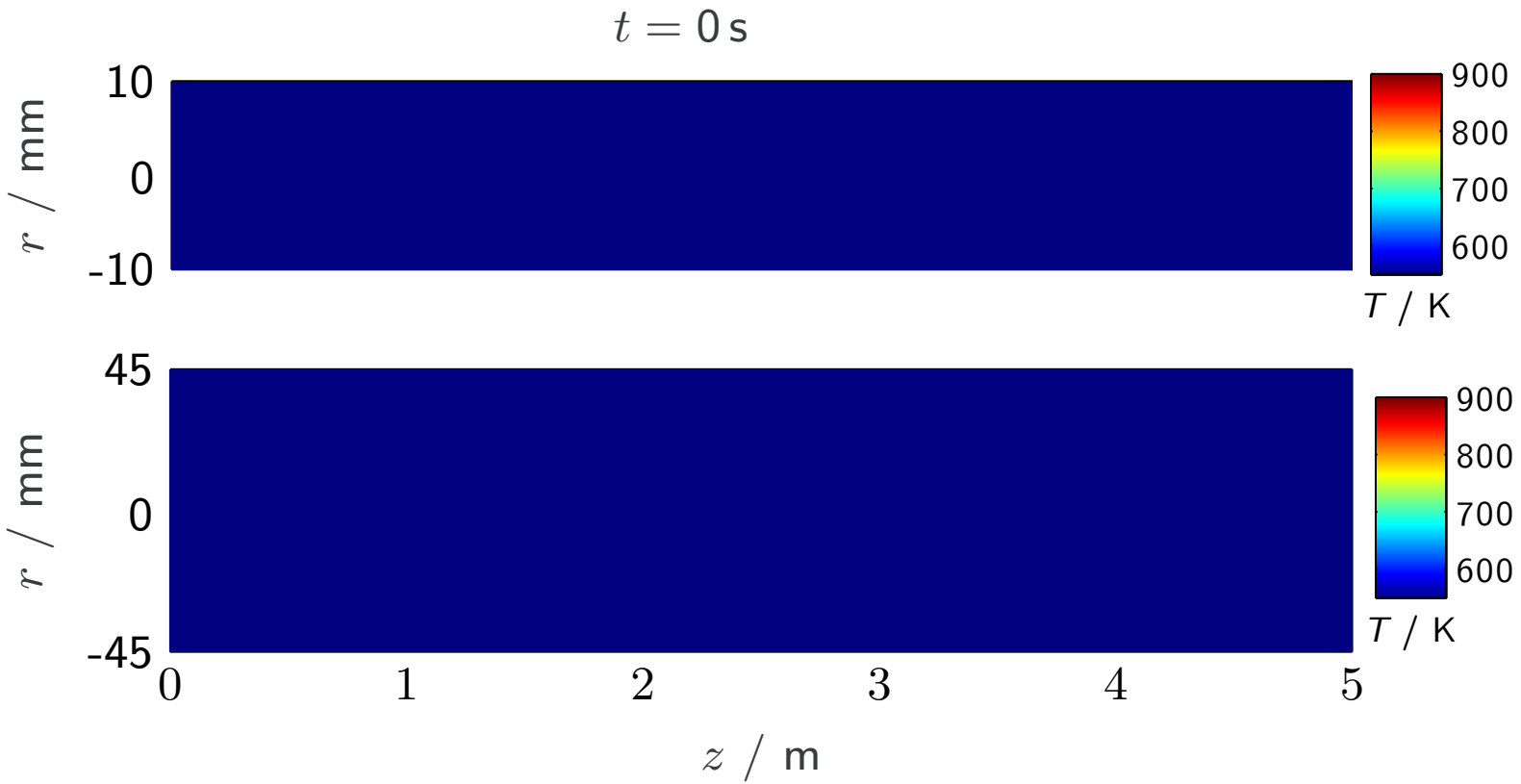
160.9 s



CPU Time:

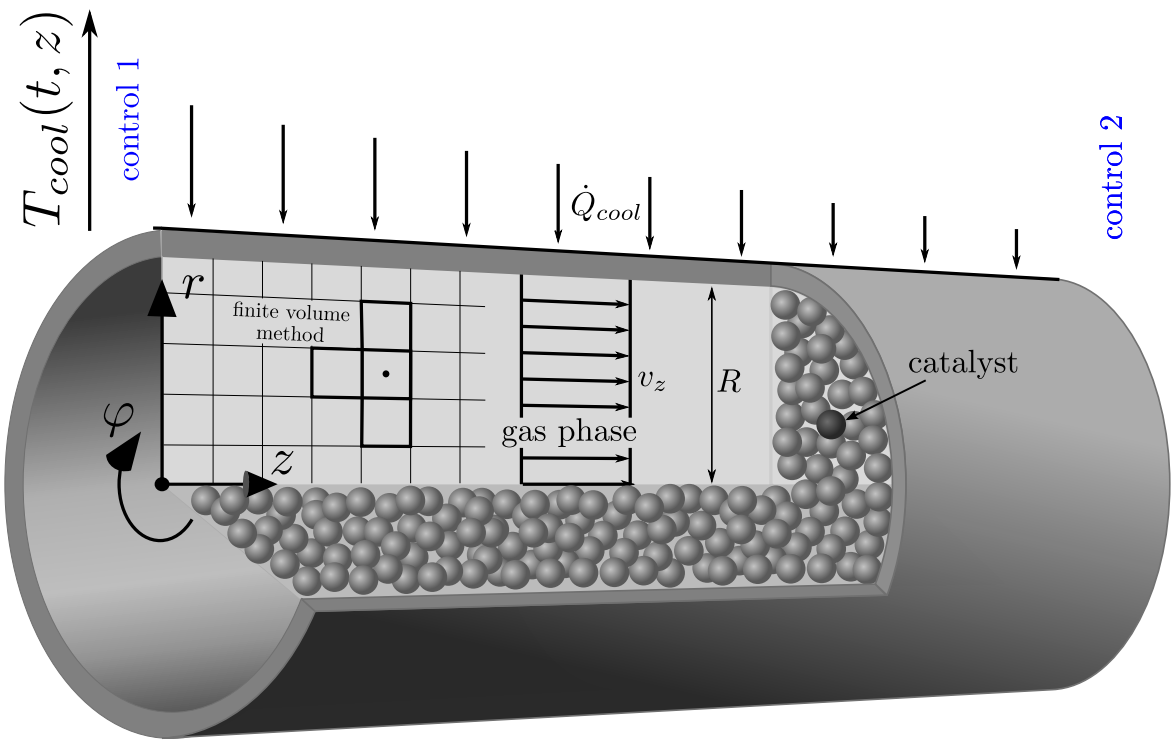
95.9 s

160.9 s



An Example in Process Engineering: 2D Methanation Reactor Model

Illustration of Jacket Cooling Approach



$$
\begin{aligned}
T_{\mathsf{cool,ub}} &= 650 \quad \mathsf{K} \\
T_{\mathsf{cool,lb}} &= 400 \quad \mathsf{K} \\
\\
T_{\mathsf{ub}} &= 750 \quad \mathsf{K} \\
T_{\mathsf{lb}} &= 300 \quad \mathsf{K}
\end{aligned}
$$

Start-Up Optimal Control Problem (OCP)

$$\max_{\mathbf{u}(t)} \quad \int_{t_0}^{t_f} X_{CO_2}(\mathbf{x}(t), \mathbf{u}(t))\, dt \; + \; R(\mathbf{u}(t)) \;\Rightarrow \text{time optimal start-up}$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \forall t \in [t_0 \; t_f] \;\Rightarrow \text{reactor model}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0,$$

$$\mathbf{x}_{\text{ub}} \geq \mathbf{x} \geq \mathbf{x}_{\text{lb}} \;\Rightarrow \text{reactor temperature bounds}$$

$$\mathbf{u}_{\text{ub}} \geq \mathbf{u} \geq \mathbf{u}_{\text{lb}} \;\Rightarrow \text{cooling at reactor jacket}$$

Simultaneous optimization approach [Biegler et al.]:
orthogonal collocation on finite elements $\Rightarrow$ large scale NLP ($> 100,000$ variables)

Recall:

Reduced-order model obtained from projection $(V^TV = I)$

$$\begin{aligned} \dot{\hat{x}} &= V^T f(t, V\hat{x}, u), \\ \hat{y} &= g(t, V\hat{x}, u). \end{aligned}$$

- The quality of ROM depends on the choice of V.

Recall:

Reduced-order model obtained from projection $(V^TV = I)$

$$\begin{aligned} \dot{\hat{x}} &= V^T f(t, V\hat{x}, u), \\ \hat{y} &= g(t, V\hat{x}, u). \end{aligned}$$

- The quality of ROM depends on the choice of V.
- A common technique for nonlinear systems is Proper Orthogonal Decomposition (POD):
  - Take computed or experimental 'snapshots' of full model:

$$[x(t_1), x(t_2), \ldots, x(t_N)] := X.$$

Recall:

Reduced-order model obtained from projection $(V^TV = I)$

$$\begin{aligned} \dot{\hat{x}} &= V^T f(t, V\hat{x}, u), \\ \hat{y} &= g(t, V\hat{x}, u). \end{aligned}$$

- The quality of ROM depends on the choice of V.
- A common technique for nonlinear systems is Proper Orthogonal Decomposition (POD):
  - Take computed or experimental 'snapshots' of full model:

$$[x(t_1), x(t_2), \ldots, x(t_N)] := X.$$

  - Perform SVD of snapshot matrix: $X = U\Sigma W^T$.

Recall:

Reduced-order model obtained from projection $(V^TV = I)$

$$\begin{aligned} \dot{\hat{x}} &= V^T f(t, V\hat{x}, u), \\ \hat{y} &= g(t, V\hat{x}, u). \end{aligned}$$

- The quality of ROM depends on the choice of V.
- A common technique for nonlinear systems is Proper Orthogonal Decomposition (POD):
  - Take computed or experimental 'snapshots' of full model:

$$[x(t_1), x(t_2), \ldots, x(t_N)] := X.$$

  - Perform SVD of snapshot matrix: $X = U\Sigma W^T$.
  - Then, the projection matrix is $V = U(:, 1:r)$.
    (Given the snapshots, this is the optimal choice w.r.t. energy.)

- Consider the nonlinear term:

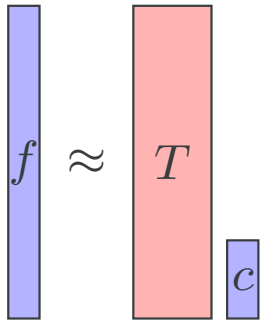
$$\hat{f} = V^T f = \boxed{V^T} \; \boxed{} f$$

- Still, we need computations of the nonlinear function on the full grid.

  $\rightsquigarrow$ no sufficient reduction in computational effort.

- Therefore, we require hyper-reduction; here, Discrete Empirical Interpolation Method (DEIM) [Chaturantabat/Sorensen 2010].

- The idea is:

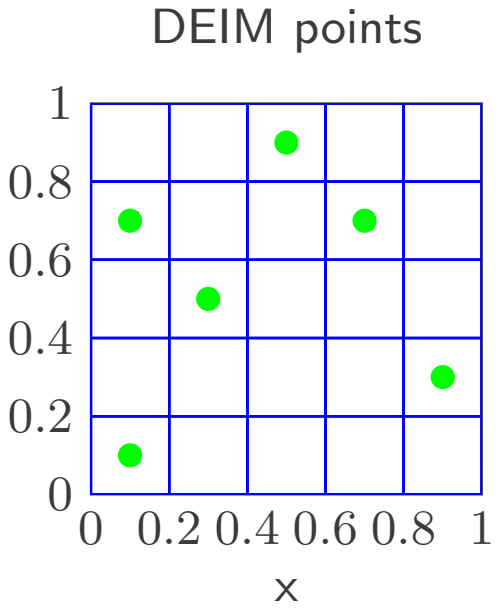


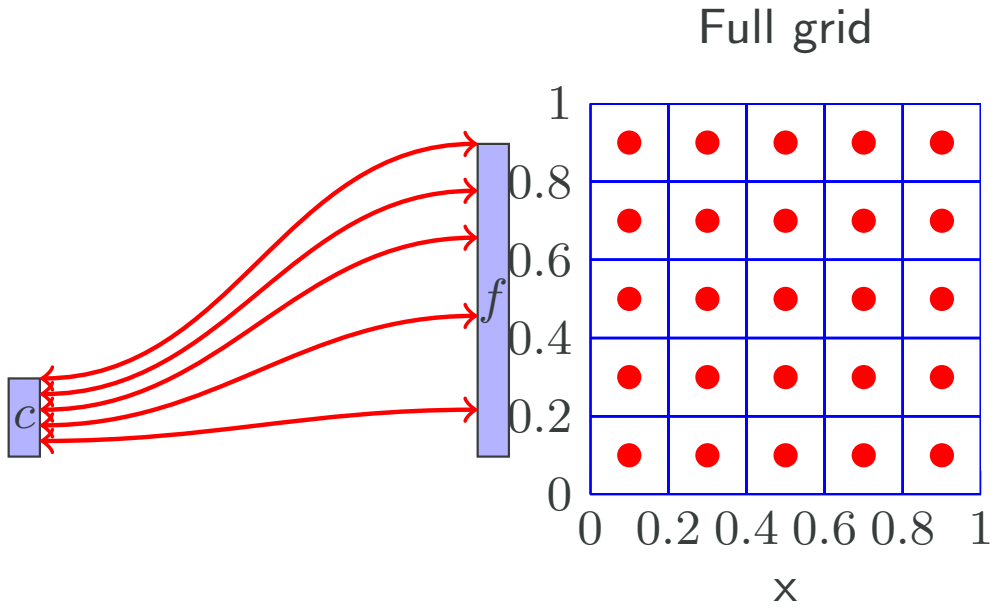
where

- 'T' is a rectangular matrix, and
- the vector 'c' contains the nonlinear function evaluations at specific grid points.

Model Order Reduction of 2D Methanation Reactor Model

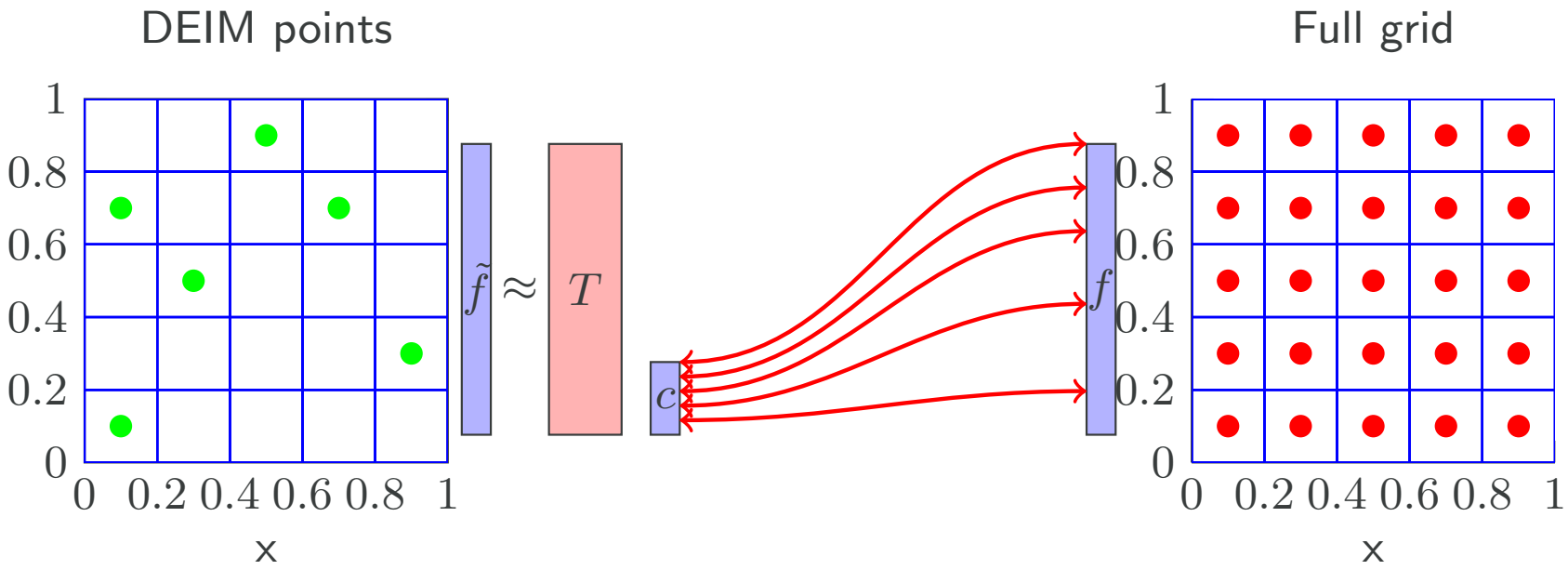
A graphical representation





Model Order Reduction of 2D Methanation Reactor Model

A graphical representation



- Use greedy algorithm to select the important grid points (DEIM points) for approximating the nonlinear function via interpolation.

Original system (full-order model, FOM):

ODE:
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

$$\Downarrow$$

$$\dot{\mathbf{x}} = \underset{\substack{\uparrow \\ \mathcal{D}_{r,\alpha} \\ a_r}}{\mathbf{A}_1(\mathbf{x})} \underset{\substack{\uparrow \\ \rho_\alpha \\ T}}{\mathbf{x}} + \mathbf{A}_2\,\mathbf{x} + \underset{\substack{\uparrow \\ \lambda}}{\mathbf{B}_1(\mathbf{x})} \underset{\substack{\uparrow \\ T_{\text{cool}}}}{\mathbf{u}_1} + \mathbf{B}_2 \underset{\substack{\uparrow \\ \rho_{\alpha,\text{inlet}} \\ T_{\text{inlet}}}}{\mathbf{u}_2} + \underset{\substack{\uparrow \\ \tilde{r}r_\beta \\ \mathcal{D}_{r,\alpha} \\ a_r\ \lambda}}{\mathbf{f}(\mathbf{x})}$$

$$\mathbf{y} = \mathbf{L}^\mathsf{T}\,\mathbf{x}$$

- POD-DEIM leads to the the following system:

$$\dot{\mathbf{x}}_r = \mathbf{Q}_A\mathbf{P}_A\mathbf{A}_1(\mathbf{x}^*)\,\mathbf{x}^* + \mathbf{V}^T\mathbf{A}_2\,\mathbf{V}\mathbf{x}_r + \mathbf{Q}_B\mathbf{P}_B\mathbf{B}_1(\mathbf{x}^*)\,\mathbf{u}_1 + \mathbf{V}^T\mathbf{B}_2\,\mathbf{u}_2 + \mathbf{Q}_f\mathbf{P}_f\mathbf{f}(\mathbf{x}^*),$$

$$\mathbf{y} = \mathbf{L}^\mathsf{T}\,\mathbf{x}^*,$$

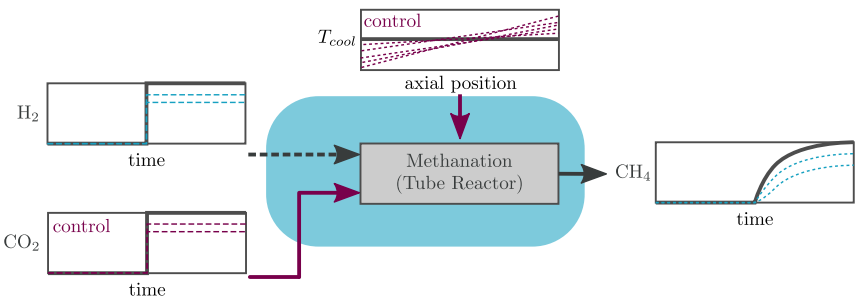
where $\mathbf{x}^* = \mathbf{V}\mathbf{x}_r$, with $\dim(\mathbf{x}_r) \ll \dim(\mathbf{x})$.

- $\mathbf{V}$ — from SVD of $\mathbf{x}$ snapshots (POD)
- $\mathbf{Q}_A$ — from SVD of $\mathbf{A}_1(\mathbf{x})\,\mathbf{x}$ snapshots (DEIM)
- $\mathbf{Q}_B$ — from SVD of $\mathbf{B}_1(\mathbf{x})\,\mathbf{x}$ snapshots (DEIM)
- $\mathbf{Q}_f$ — from SVD of $\mathbf{f}(\mathbf{x})$ snapshots (DEIM)

e.g., for $\mathbf{f}(\mathbf{x})$

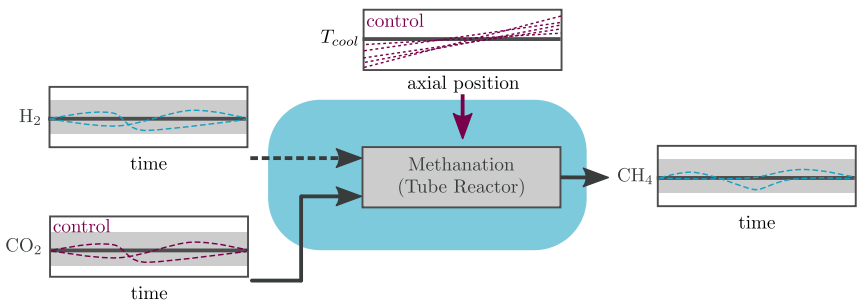
$$SVD\left[\begin{array}{c|c|c|c} f_1(\mathbf{x}(t_1)) & f_1(\mathbf{x}(t_2)) & f_1(\mathbf{x}(t_3)) & \cdots \\ \hline f_2(\mathbf{x}(t_1)) & f_1(\mathbf{x}(t_2)) & f_3(\mathbf{x}(t_3)) & \cdots \\ \hline f_3(\mathbf{x}(t_1)) & f_1(\mathbf{x}(t_2)) & f_1(\mathbf{x}(t_3)) & \cdots \\ \hline \vdots & \vdots & \vdots & \ddots \end{array}\right]$$

Scenario 1: start-up phase



- The range of $x_{CO2} \in [0.7, 0.9]$.
- The range of $x_{H2} \in [0.1, 0.3]$.
- The range of $T_{cool} \in [500K, 700K]$.

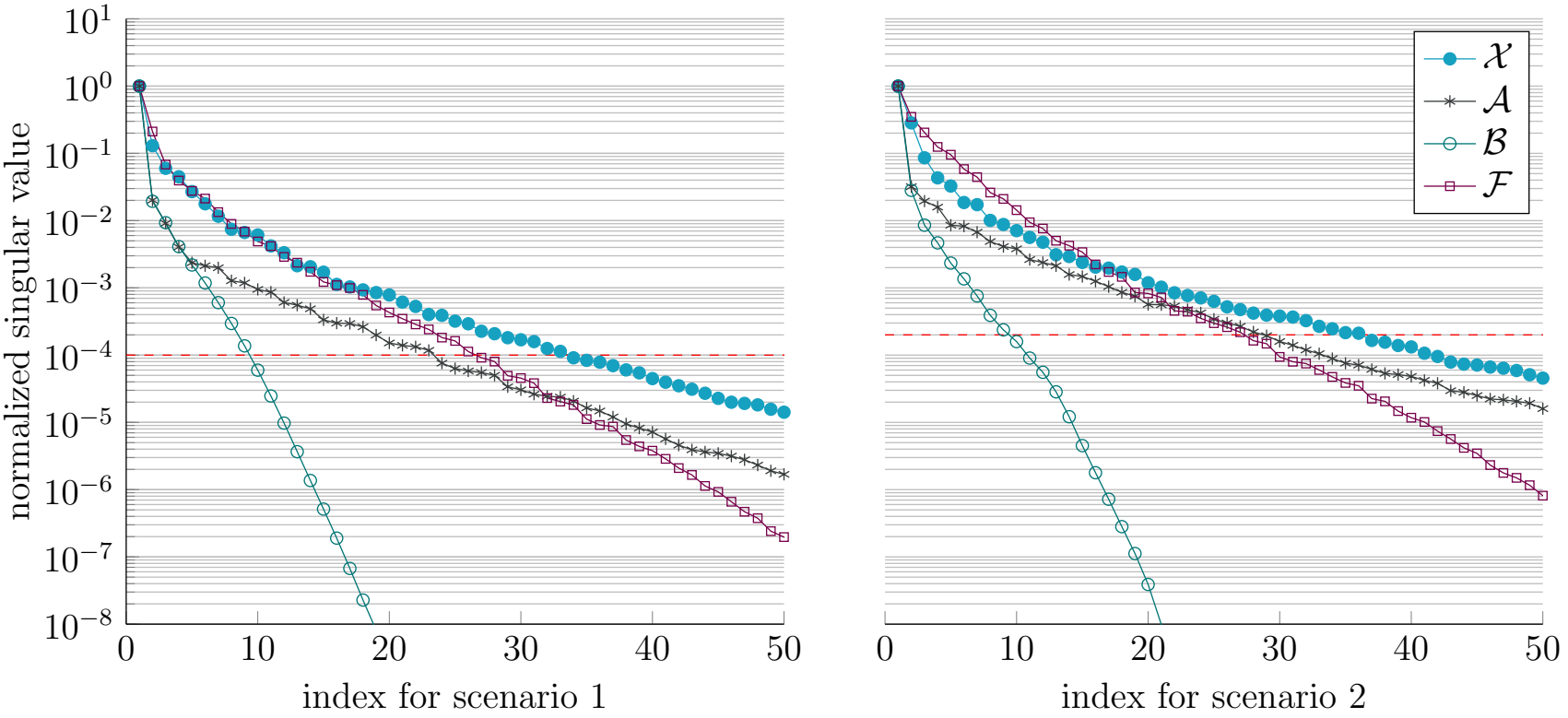
Scenario 2: continuous operation



- No. Training cases: 50
- No. Test cases: 20

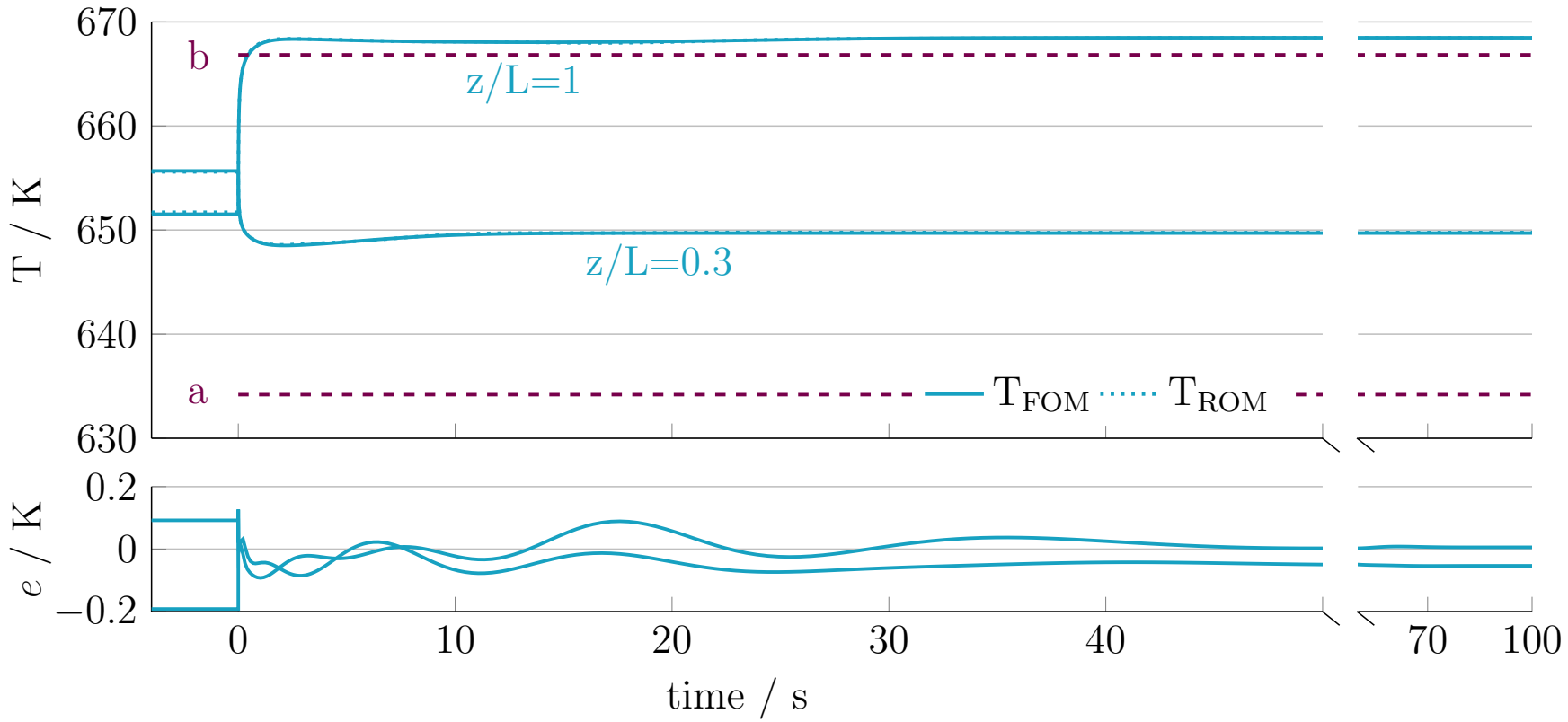
Model Order Reduction of 2D Methanation Reactor Model

Singular Value Decay



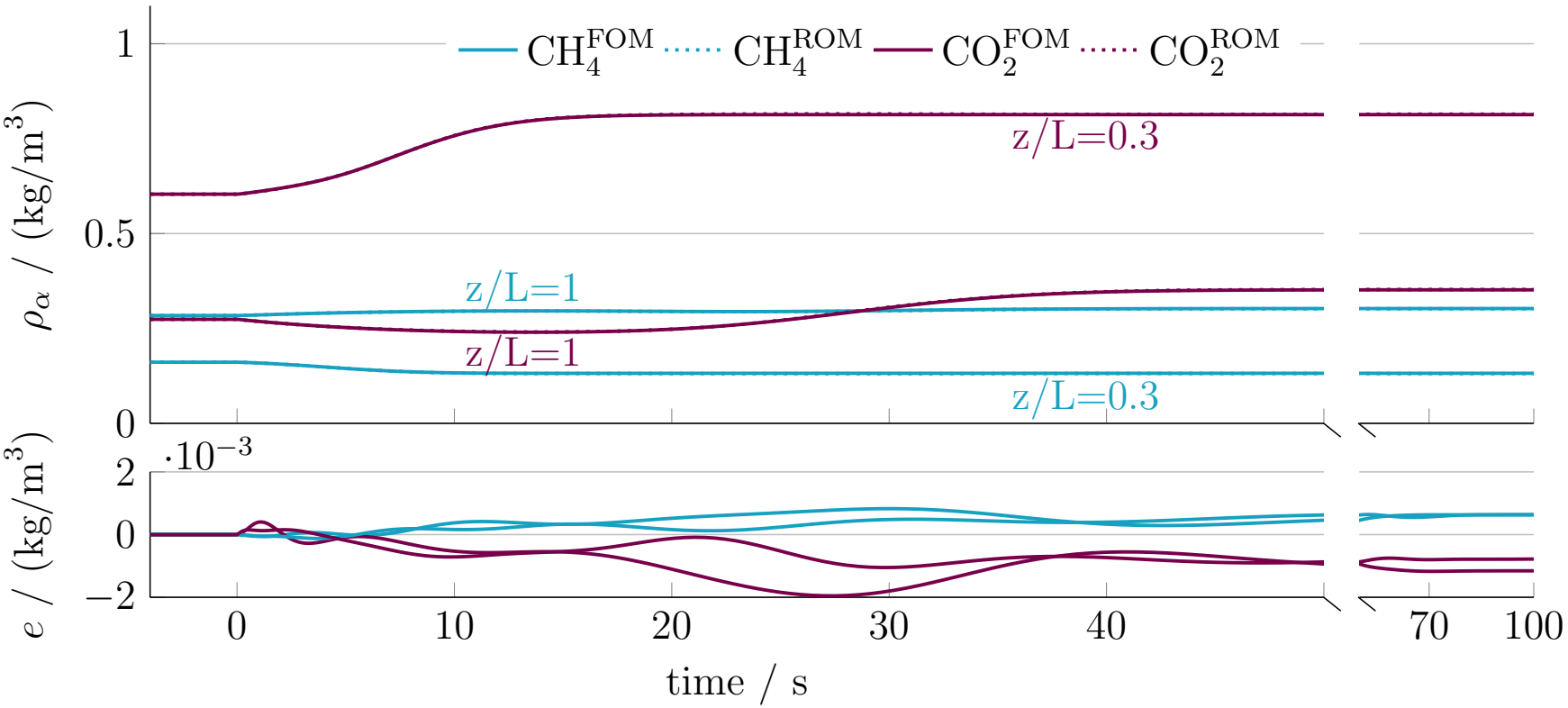
Model Order Reduction of 2D Methanation Reactor Model

ROM vs. FOM — Continuous Operation (Best Case)



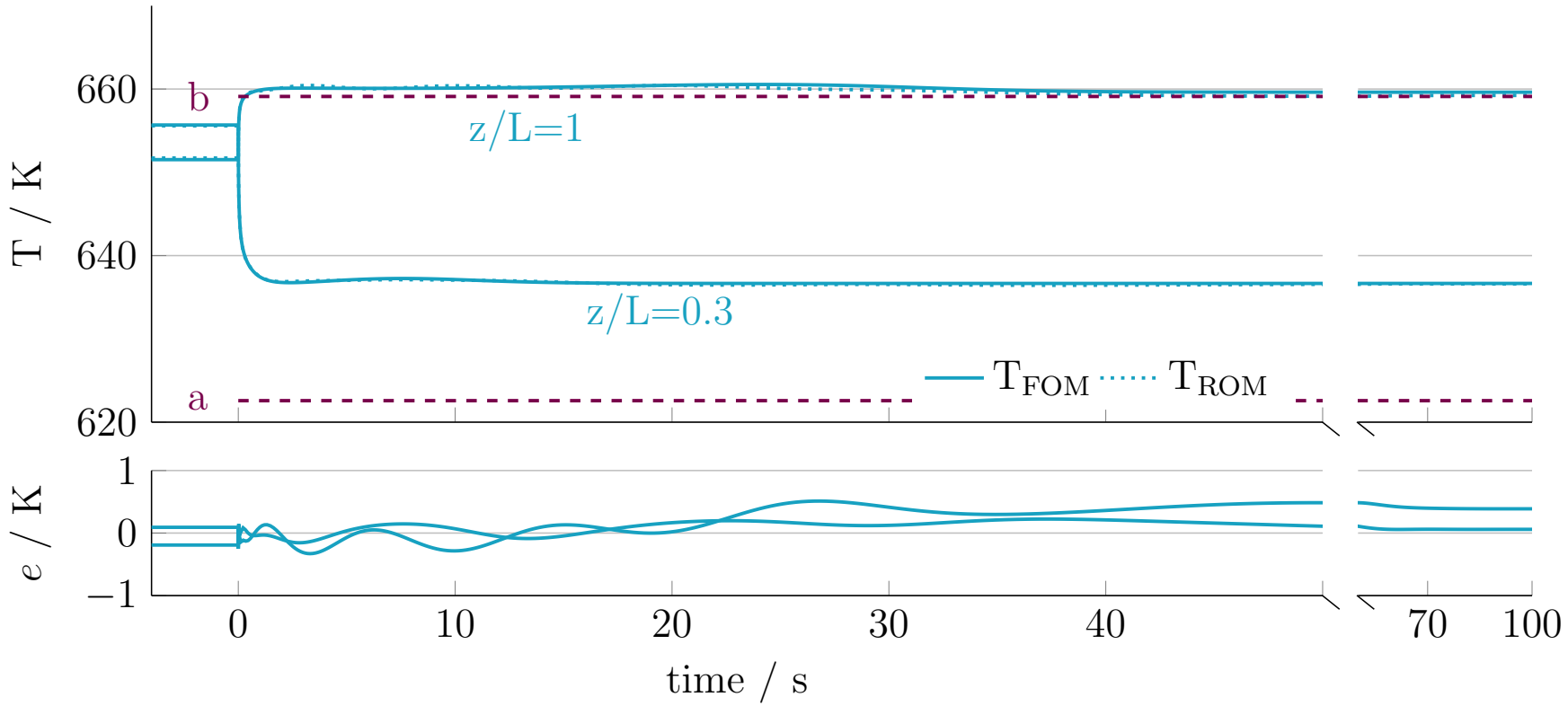
Model Order Reduction of 2D Methanation Reactor Model

ROM vs. FOM — Continuous Operation (Best Case)



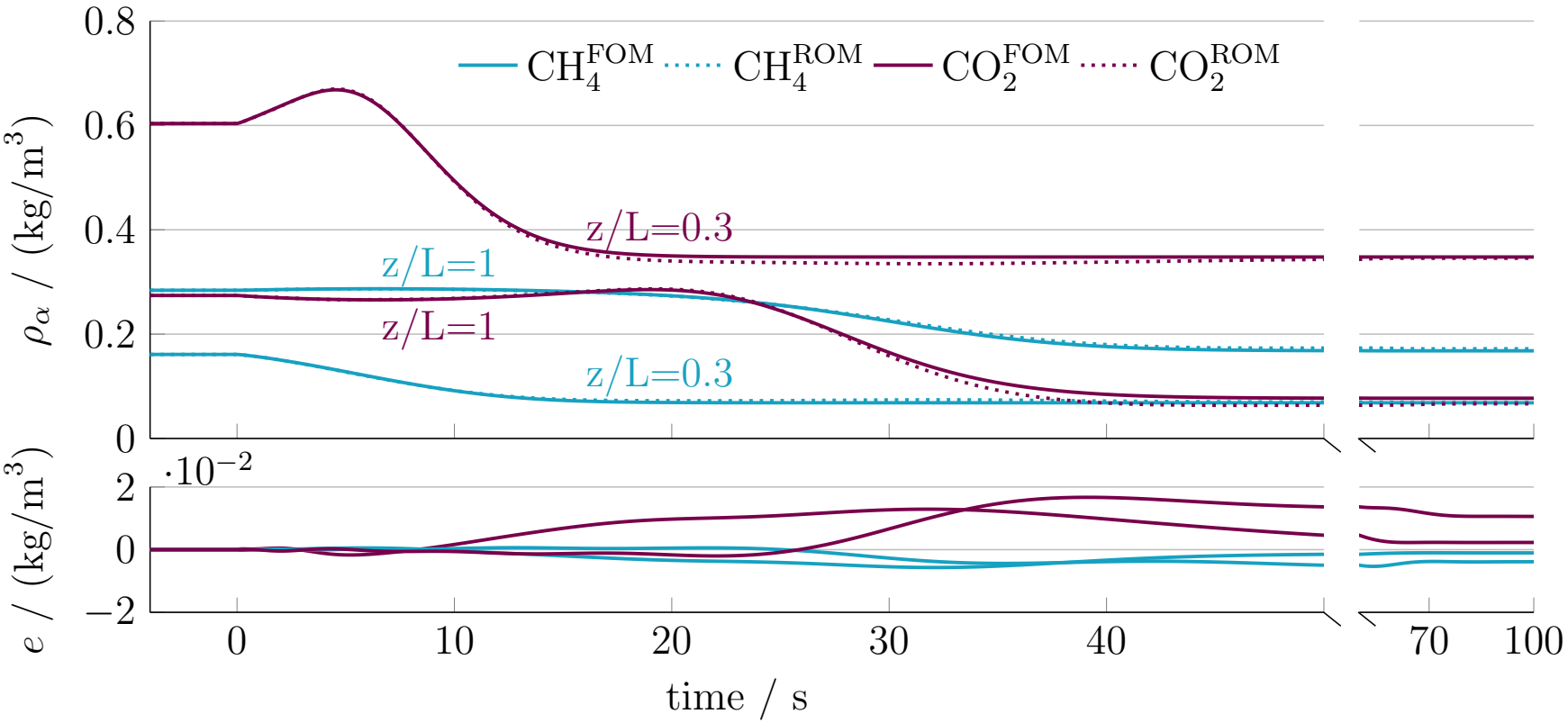
Model Order Reduction of 2D Methanation Reactor Model

ROM vs. FOM — Continuous Operation (Worst Case)



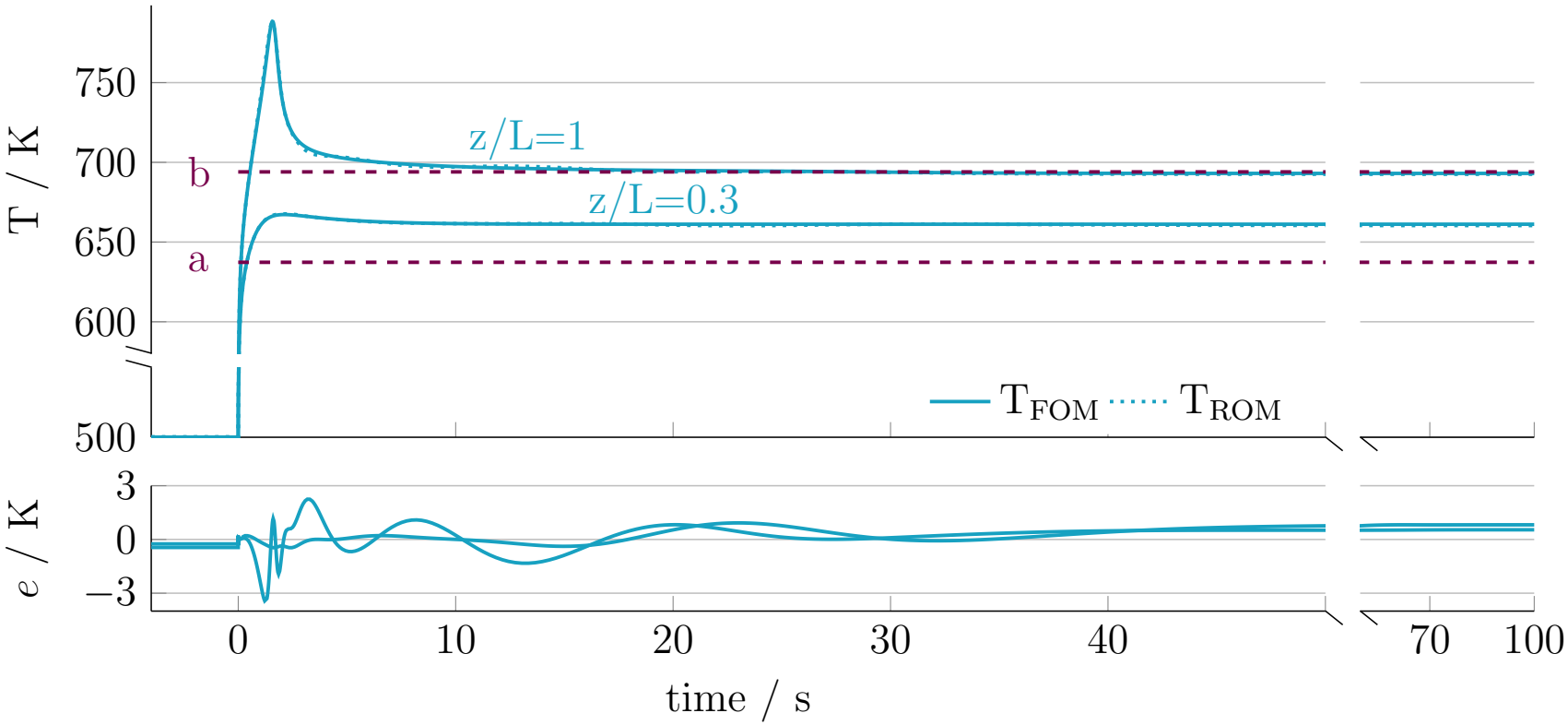
Model Order Reduction of 2D Methanation Reactor Model

ROM vs. FOM — Continuous Operation (Worst Case)



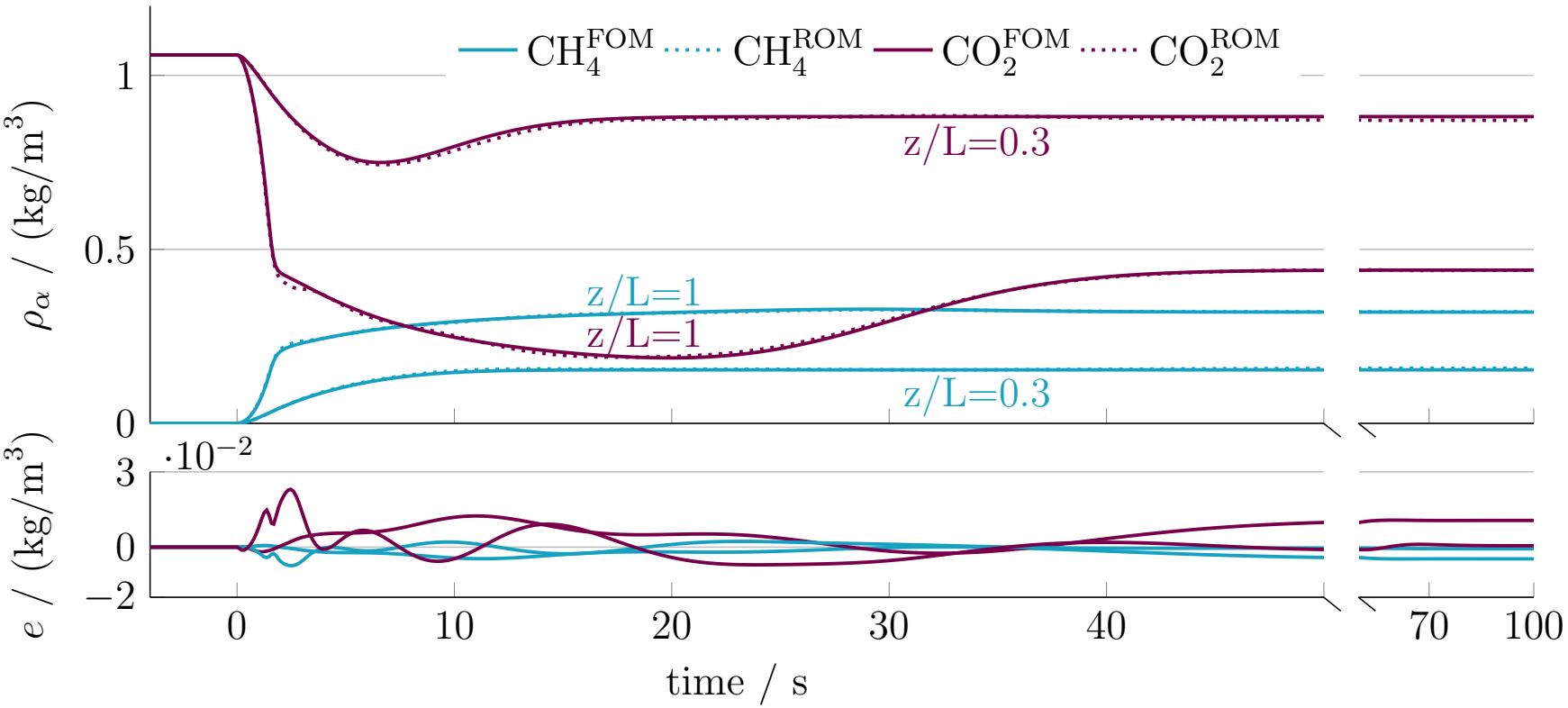
Model Order Reduction of 2D Methanation Reactor Model

ROM vs. FOM — Start-Up (Best Case)



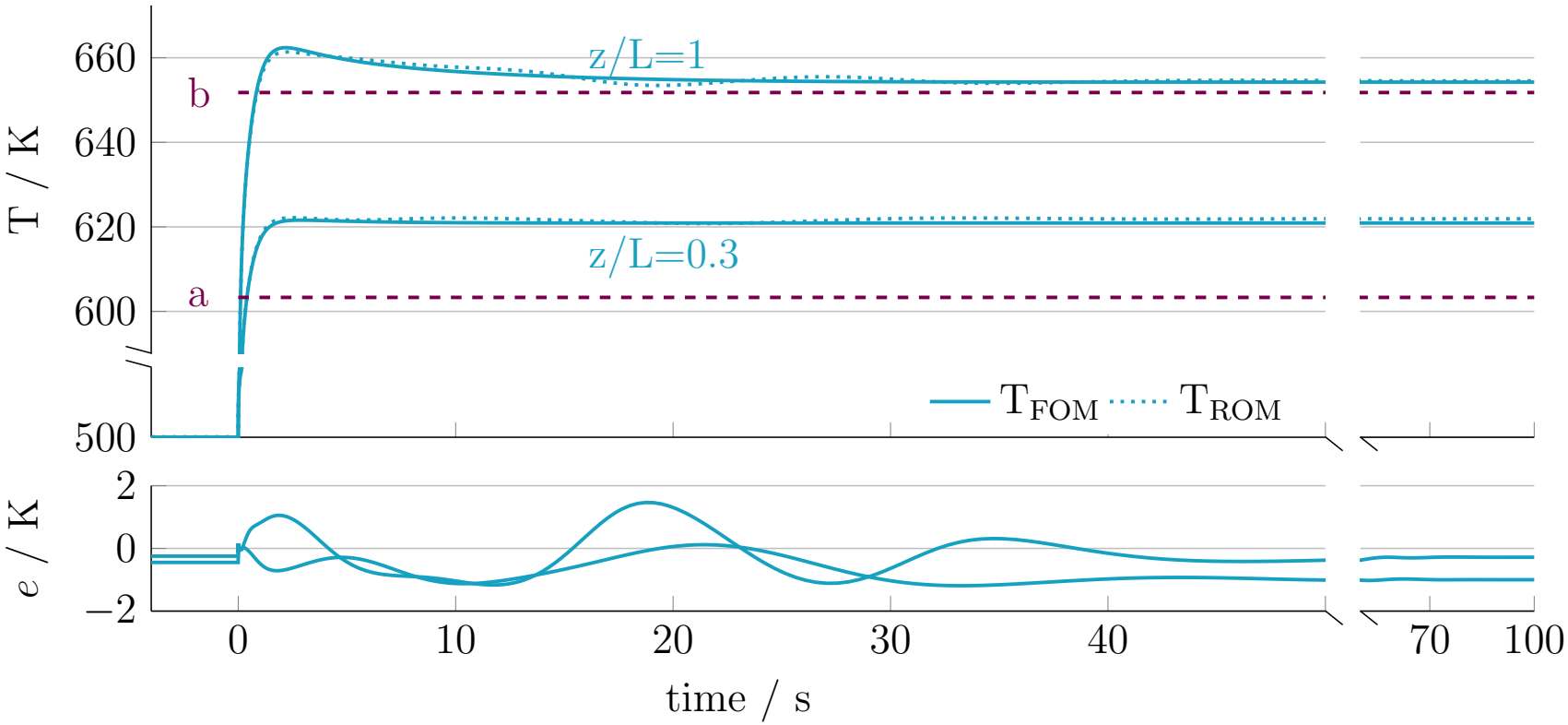
Model Order Reduction of 2D Methanation Reactor Model

ROM vs. FOM — Start-Up (Best Case)



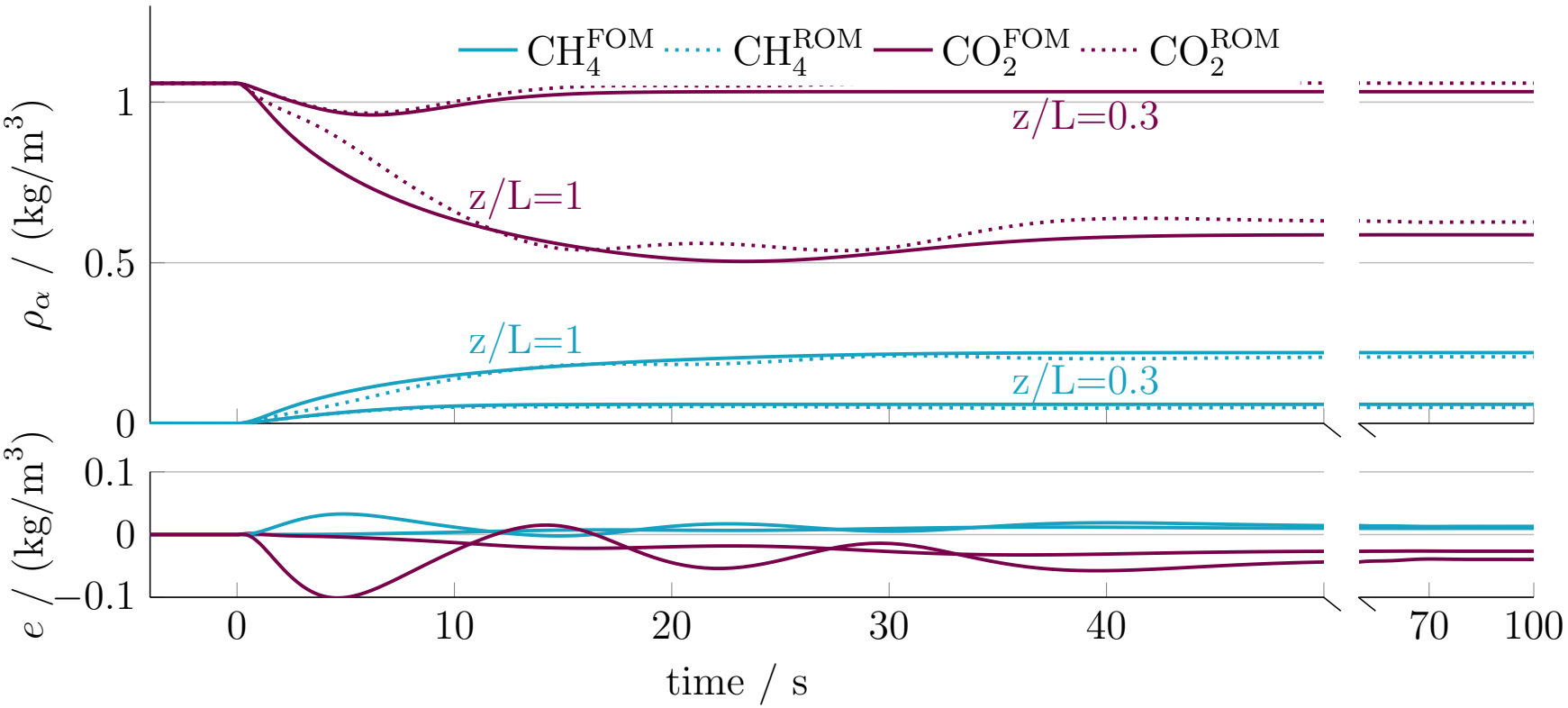
Model Order Reduction of 2D Methanation Reactor Model

ROM vs. FOM — Start-Up (Worst Case)



Model Order Reduction of 2D Methanation Reactor Model

ROM vs. FOM — Start-Up (Worst Case)



Model Order Reduction of 2D Methanation Reactor Model

ROM vs. FOM — Summary

model	no. states	avg. CPU-time	median of ε / %						
			$\bar{\varepsilon}_{\mathrm{CH4}}$	$\bar{\varepsilon}_{\mathrm{CO}}$	$\bar{\varepsilon}_{\mathrm{CO2}}$	$\bar{\varepsilon}_{\mathrm{H2O}}$	$\bar{\varepsilon}_{\mathrm{H2}}$	$\bar{\varepsilon}_{\mathrm{N2}}$	$\bar{\varepsilon}_{\mathrm{T}}$
FOM-S1	4375	19.5 s	-	-	-	-	-	-	-
ROM-S1	34	1.3 s	1.16	2.06	0.84	1.12	0.88	0.22	0.02
FOM-S2	4375	39.8 s	-	-	-	-	-	-	-
ROM-S2	36	2.4 s	1.77	3.27	1.13	1.74	1.29	0.56	0.18

Conclusions and Outlook for SmartProSys

- Modern model order reduction techniques are able to produce surrogate models enabling real-time simulation and control for the nonlinear coupled systems of partial differential equations describing chemical reactions.

Conclusions and Outlook for SmartProSys

- Modern model order reduction techniques are able to produce surrogate models enabling real-time simulation and control for the nonlinear coupled systems of partial differential equations describing chemical reactions.
- Current efforts are mostly directed towards data-driven methods, i.e., to inferring surrogate models directly from data, using machine/deep learning.

Conclusions and Outlook for SmartProSys

- Modern model order reduction techniques are able to produce surrogate models enabling real-time simulation and control for the nonlinear coupled systems of partial differential equations describing chemical reactions.
- Current efforts are mostly directed towards data-driven methods, i.e., to inferring surrogate models directly from data, using machine/deep learning.
- Updating the surrogate models using current measurements/data of a chemical process requires the fusion of model order reduction with data assimilation techniques!

Conclusions and Outlook for SmartProSys

- Modern model order reduction techniques are able to produce surrogate models enabling real-time simulation and control for the nonlinear coupled systems of partial differential equations describing chemical reactions.
- Current efforts are mostly directed towards data-driven methods, i.e., to inferring surrogate models directly from data, using machine/deep learning.
- Updating the surrogate models using current measurements/data of a chemical process requires the fusion of model order reduction with data assimilation techniques!
- Smart process engineering will require digital twins, building on
  - *the accuracy of high-fidelity models inherited by the surrogates obtained from model order reduction;*
  - *calibration of the surrogate models using measurement data;*
  - *efficient data assimilation techniques for updating the surrogates.*

Selected References

S.Li, L. Feng, A. Seidel-Morgenstern, P. Benner.
Using surrogate models for efficient optimization of simulated moving bed chromatography.
Computers & Chemical Engineering 67:121–132, 2014.

S.Li, Y. Yue, L. Feng, A. Seidel-Morgenstern, P. Benner.
Model reduction for linear simulated moving bed chromatography systems using Krylov-subspace methods,
AIChE-Journal 60(11):3773–3783, 2014.

M. Mangold, L. Feng, D. Khlopov, S. Palis, P. Benner, D. Binev, A. Seidel-Morgenstern.
Nonlinear model reduction of a continuous fluidized bed crystallizer.
Journal of Computational and Applied Mathematics 289:253–266, 2015.

J. Bremer, P. Goyal, L. Feng, P. Benner, K. Sundmacher.
Nonlinear model order reduction for catalytic tubular reactors.
In Proc. of the 26th European Symposium on Computer Aided Process Engineering, pp. 2373–2378, 2016.

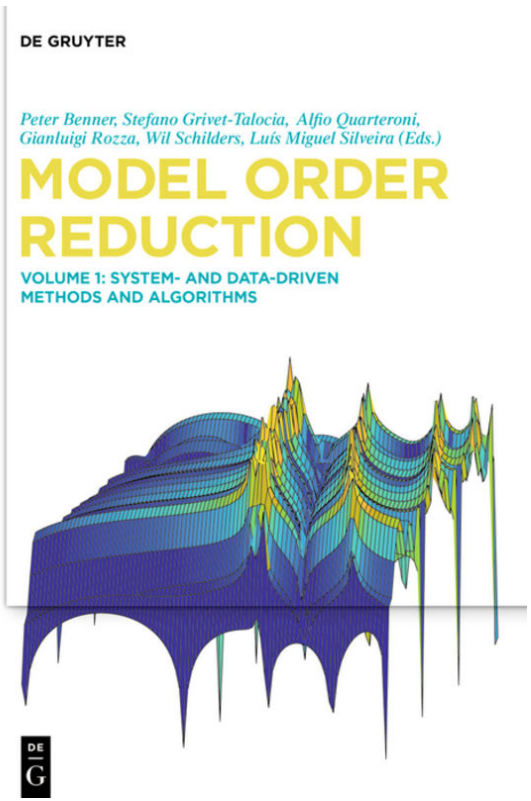
J. Bremer, P. Goyal, L. Feng, P. Benner, K. Sundmacher.
POD-DEIM for efficient reduction of a dynamic 2D catalytic reactor model.
Computers & Chemical Engineering 106:777–784, 2017.

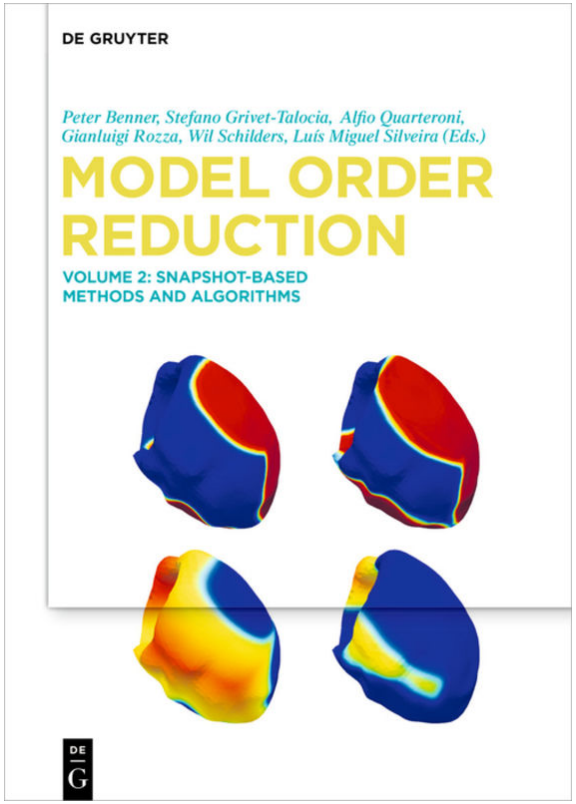
Y. Zhang, L. Feng, A. Seidel-Morgenstern, P. Benner.
Accelerating optimization and uncertainty quantification of nonlinear SMB chromatography using reduced-order models,
Computers & Chemical Engineering 96:237–247, 2017.

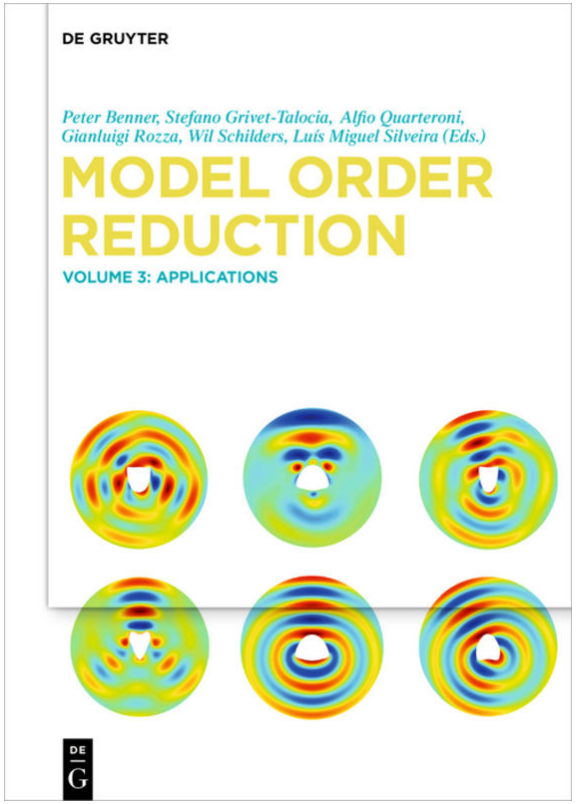
D. Garmatter, A. Maggi, M. Wenzel, S. Monem, M. Hahn, M. Stoll, S. Sager, P. Benner, K. Sundmacher
Power-to-Chemicals: a superstructure problem for sustainable syngas production.
In S. Göttlich, M. Herty, and A. Milde (Eds.), *Mathematical Modeling, Simulation and Optimization for Power Engineering and Management*, Mathematics in Industry, Springer, Vol. 34, pp. 145–168, 2021.

J. Bremer, J. Heiland, P. Benner, K. Sundmacher.
Non-intrusive time Galerkin POD for optimal control of a fixed-bed reactor for CO2 methanation.
IFAC-PapersOnLine 54:122–127, 2021.

Commercial 1: 3-Volume Handbook "Model Order Reduction"





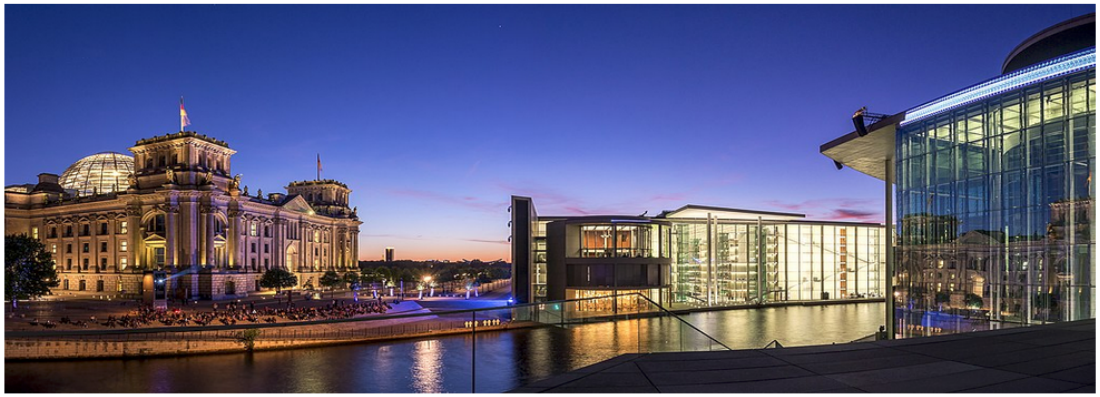


- Edited by Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, and Luís Miguel Silveira,
- contains 30 tutorial chapters on modern model reduction techniques, methods, applications, and software,
- published by DeGruyter in 2021, ebook is fully Open Access!

https://more.sciencesconf.org

Model Reduction and Surrogate Modeling (MORE)

19-23 Sep 2022 Berlin (Germany)



A_Peach, CC BY 2.0, via Wikimedia Commons

Login

MAIN MENU

- Home
- News
- Committees
- Invited speakers
- Important dates
- Call for contributions
- Location

HELP

- @ Contact

OVERVIEW

The conference merges activities of the two independent conference series MoRePaS and MODRED. The goal is to foster an international exchange of new concepts and ideas related to the following topics:

- *Parametric model order reduction*
- *System-theoretic model reduction methods and frequency-domain methods*
- *Data-driven approaches*
- *Non-intrusive model order reduction*
- *Machine learning/deep learning and model order reduction*
- *Tensor methods*
- *Kernel methods for nonlinear MOR*
- *MOR for problems with poor Kolmogorov N-width decay (e.g. transport phenomena)*
- *Structure-preserving and energy-based MOR (e.g. Hamiltonian or port-Hamiltonian systems)*
- *Nonlinear model reduction*
- *Localized MOR and multi-scale problems*
- *Randomized algorithms*
- *High dimensional parameter spaces and reduction in parameter space*
- *Dynamic and adaptive approximations, error estimation*
- *Multifidelity methods*
- *MOR for uncertainty quantification*
- *Model reduction for nonlinear bifurcating PDEs*
- *Model reduction for optimization, control and inverse problems*
- *MOR for multiphysics/multiphase problems*
- *Large-scale applications, digital twins*
- *MOR for industrial applications*
- *Emerging computational technologies based on ROM*
- *Model Reduction Software and Benchmarks*