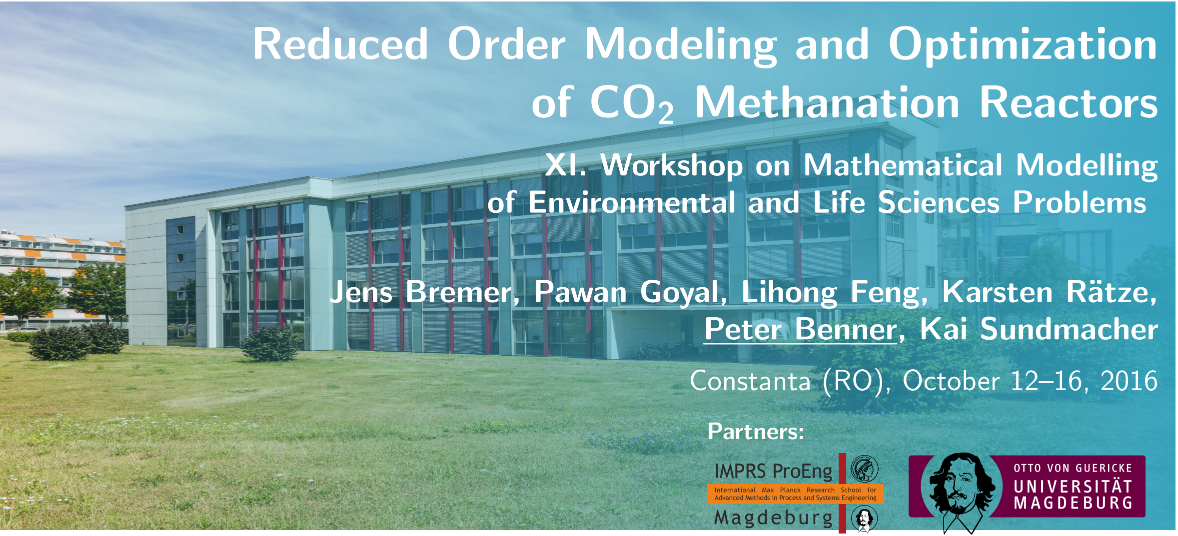
MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC COMPUTATIONAL METHODS IN SYSTEMS AND CONTROL THEORY

PSE PROCESS SYSTEMS ENGINEERING



Reduced Order Modeling and Optimization of CO_2 Methanation Reactors

XI. Workshop on Mathematical Modelling of Environmental and Life Sciences Problems

Jens Bremer, Pawan Goyal, Lihong Feng, Karsten Rätze, Peter Benner, Kai Sundmacher

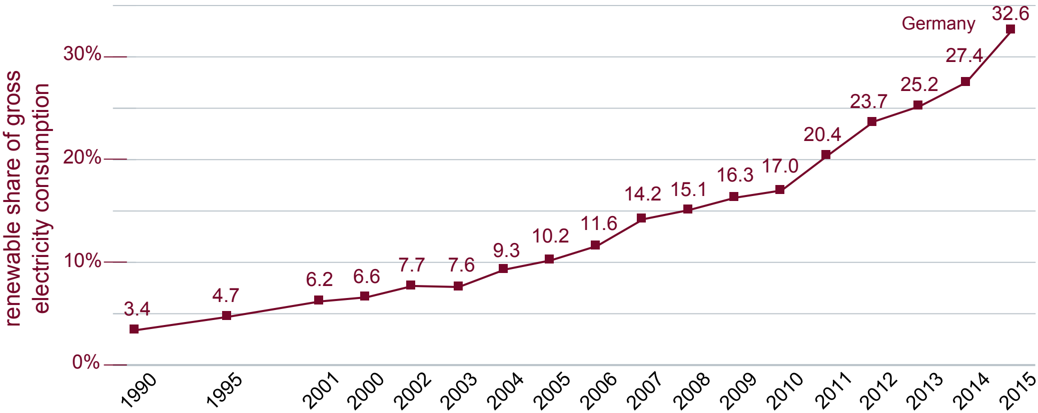
Constanta (RO), October 12–16, 2016

Partners:

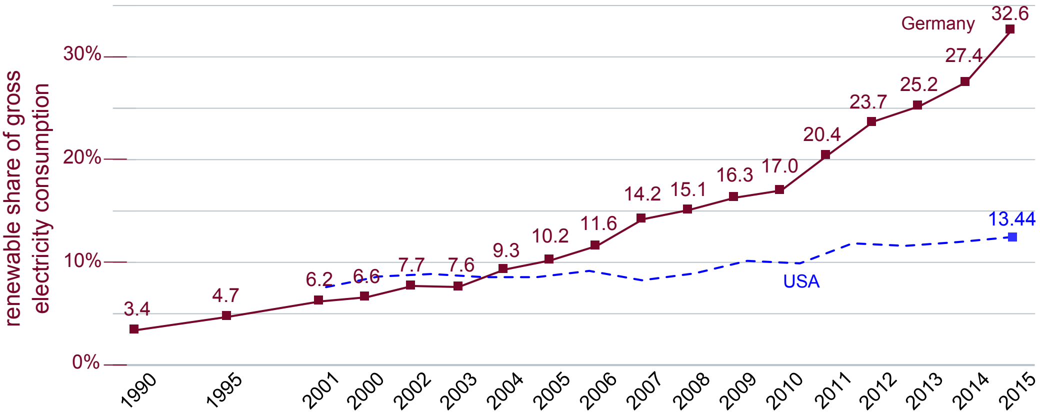
IMPRS ProEng
International Max Planck Research School for Advanced Methods in Process and Systems Engineering
Magdeburg

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

Towards 100% Renewable Electricity

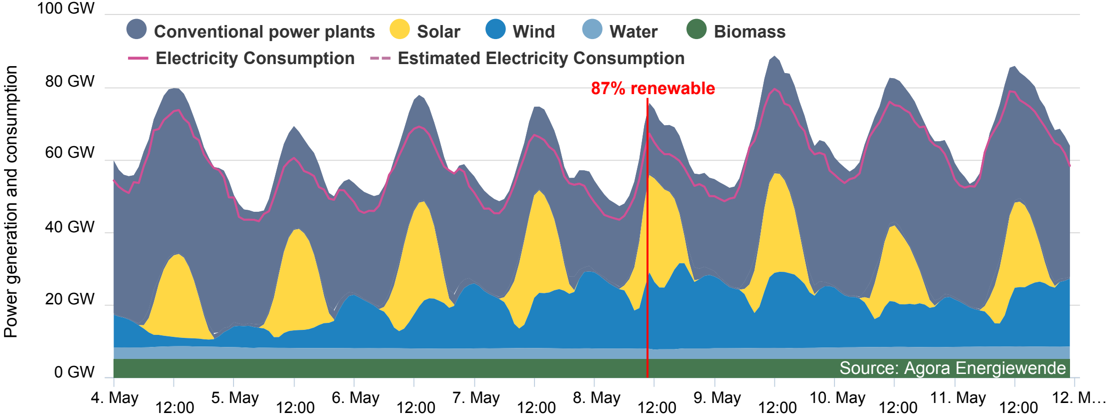


Towards 100% Renewable Electricity

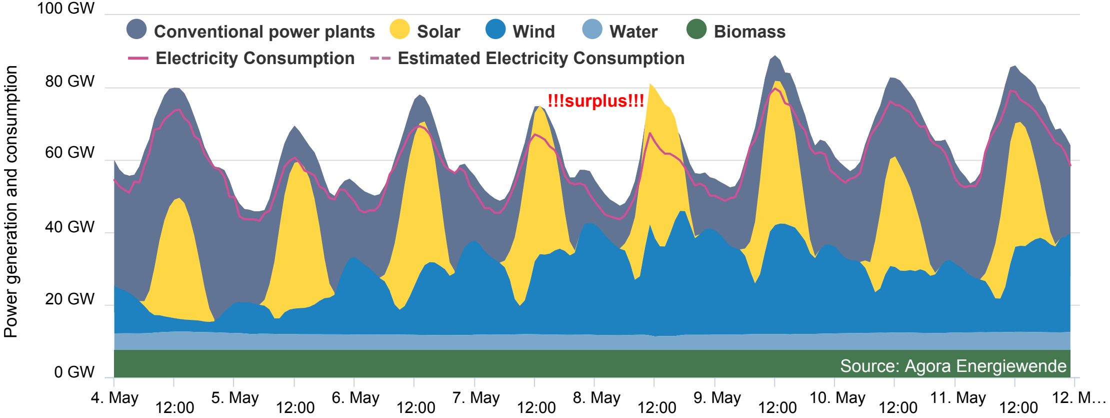


The Big Picture

Dynamics of Power Generation and Consumption in Germany (4.May - 12.May 2016)

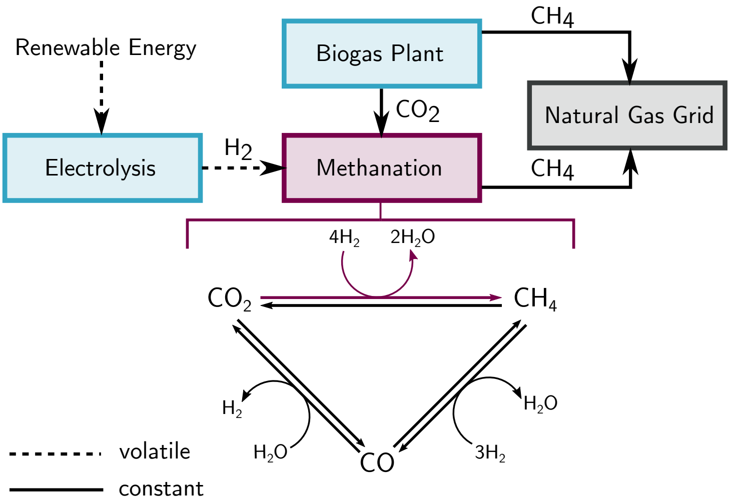


Dynamics of Power Generation and Consumption in Germany (Possible Future Scenario)



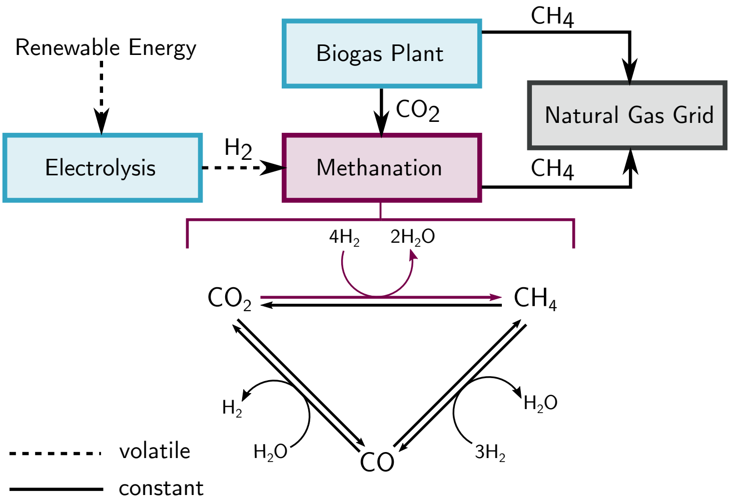
Flexible storage processes to utilize volatile renewable electricity are crucial!

CO_2-Methanation



- High energy density
- Use of available infrastructure
- Flexible

CO_2-Methanation

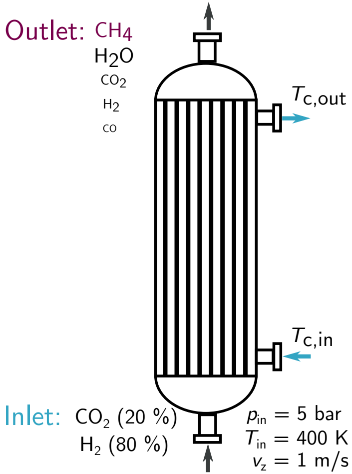


- High energy density
- Use of available infrastructure
- **Flexible**

Dynamic Consideration

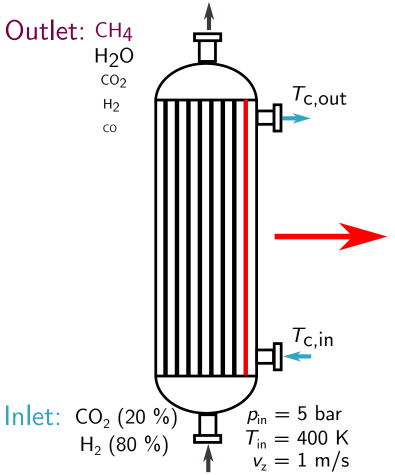
- Steady-state disturbances
- **Start-up** / Shut-down

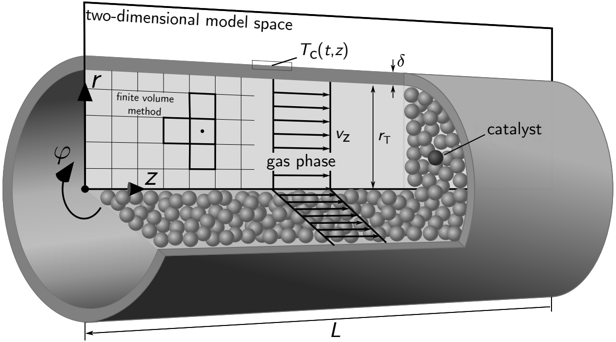
Modeling Approach



2D Reactor Model

Modeling Approach





Bremer, J. (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	

2D Reactor Model

Governing Equations for 2D Pseudo-Homogenous Reactor Model

Mass Balance

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

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)

$$\frac{\partial \rho_\alpha}{\partial r} = 0, \quad \frac{\partial T}{\partial r} = 0 \quad \text{at } r = 0$$

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

2D Reactor Model

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

PDAE:

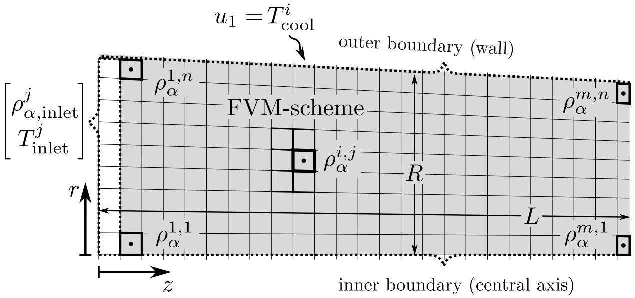
$$\frac{\partial \rho_\alpha}{\partial t} = -\frac{v_z}{\varepsilon}\frac{\partial \rho_\alpha}{\partial z} + \dots$$
$$\frac{\partial T}{\partial t} = \dots$$

$\xRightarrow{\;FVM\;}$

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**



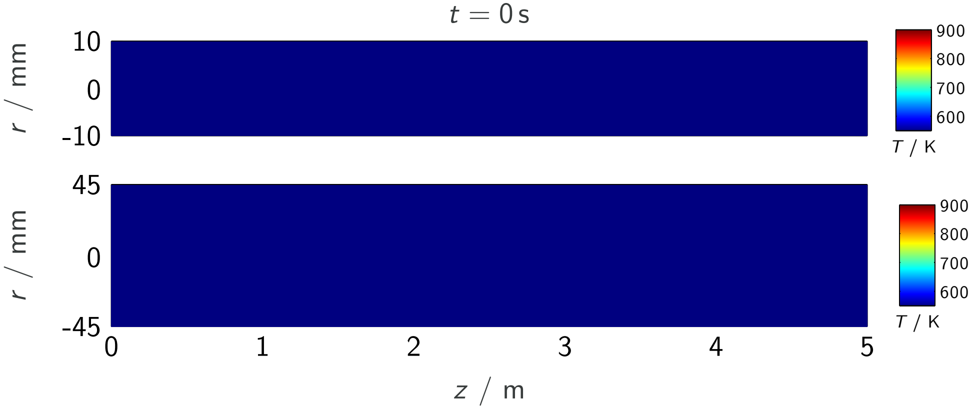
$\dim(\mathbf{x}) = 350$ - 4000
(depending on grid density)

Start-Up Scenario - **Temperature Distribution**

CPU Time:

95.9 s

160.9 s

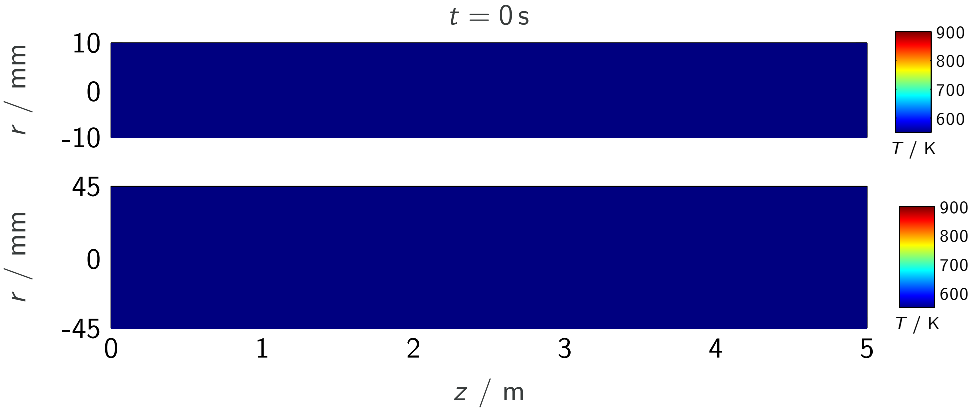


Start-Up Scenario - **Temperature Distribution**

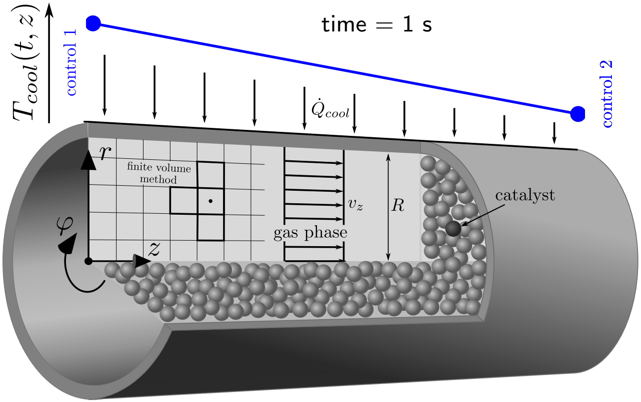
CPU Time:

95.9 s

160.9 s



Exemplary Illustration of Jacket Cooling Approach



$$T_{cool,ub} = 650 \quad K$$
$$T_{cool,lb} = 400 \quad K$$

$$T_{ub} = 750 \quad K$$
$$T_{lb} = 300 \quad K$$

Formulation

Start-Up Optimal Control Problem (OCP)

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

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

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

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

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

Simultaneous optimization approach [Biegler et al.]:
orthogonal collocation on finite elements ⇒ **large scale NLP** (above 100'000 variables)

Computational Implementation of OCP

CasADi

- A minimalistic Computer Algebra System (CAS) written in self-contained C++.
  - MATLAB-like syntax - **"everything is a matrix"**.
  - Use from C++, Python and **MATLAB**.
  - C-code generation from all interfaces - just-in-time compilation.
- "Smart interfaces" to numerical codes.

Computational Implementation of OCP

CasADi

- A minimalistic Computer Algebra System (CAS) written in self-contained C++.
  - MATLAB-like syntax - **"everything is a matrix"**.
  - Use from C++, Python and **MATLAB**.
  - C-code generation from all interfaces - just-in-time compilation.
- "Smart interfaces" to numerical codes.
  - NLP solvers: IPOPT, sIPOPT, KNITRO, ...

    $\rightsquigarrow$ Automatic generation of **exact, sparse Hessians and Jacobians**.
  - Integrators: CVODES, IDAS

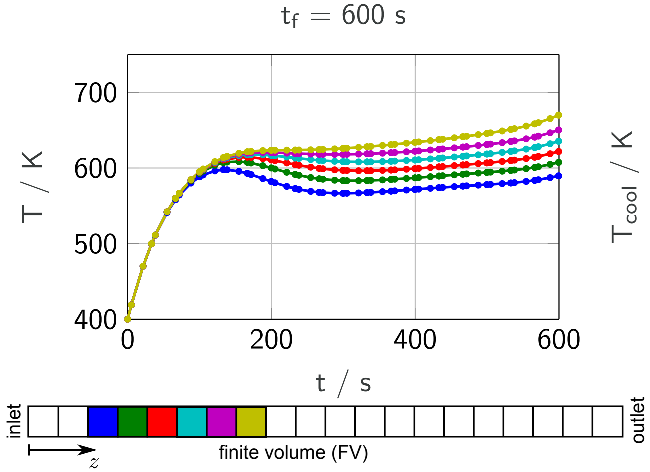
    $\rightsquigarrow$ Access to **shooting methods** with automatic formulation of sensitivities.
  - Symbolic reformulation of DAE's.

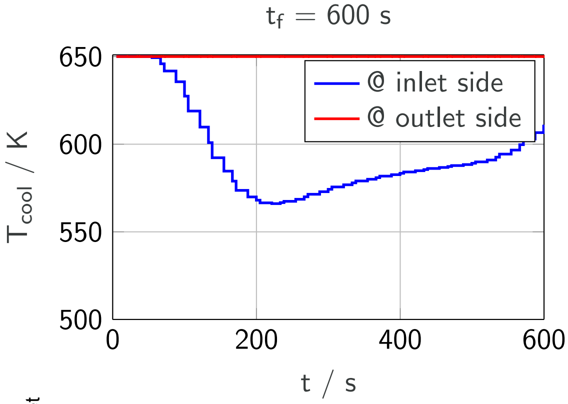
    $\rightsquigarrow$ Sorting/scaling of variables and equations.

    $\rightsquigarrow$ Elimination of some or all algebraic states symbolically.

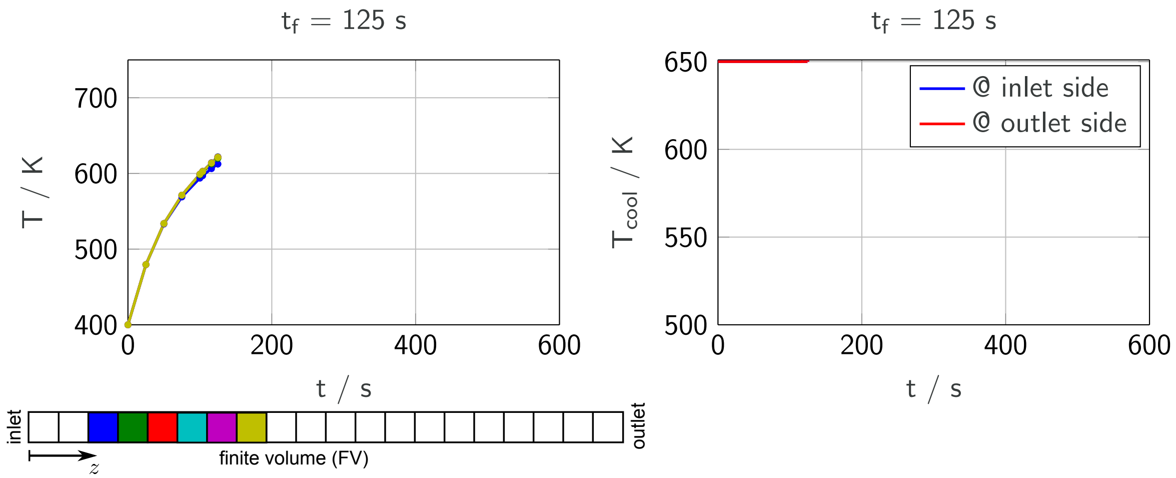
Optimal Control

1D Results - One-Step Optimization with Insufficient Initialization

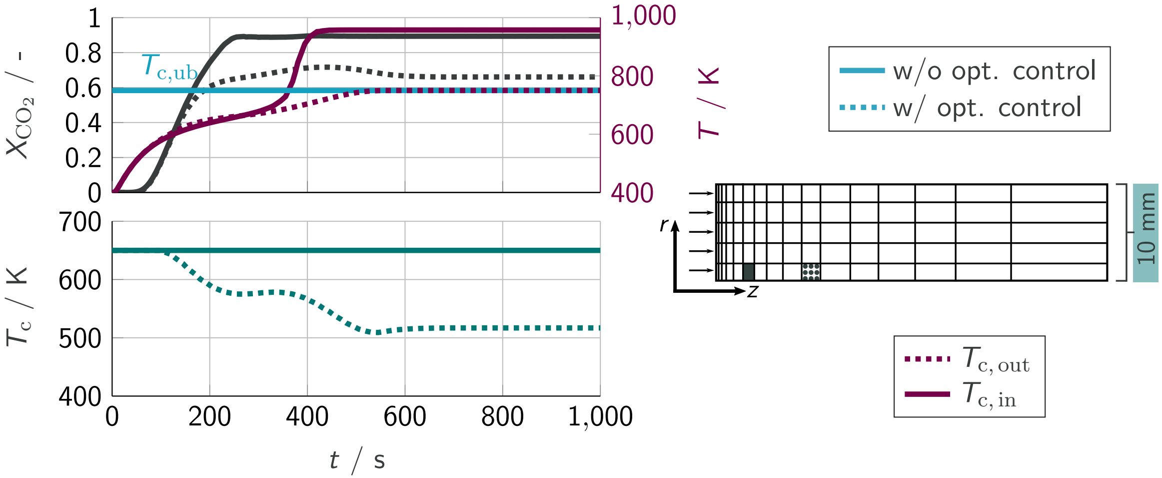




1D Results - Multi-Step Optimization

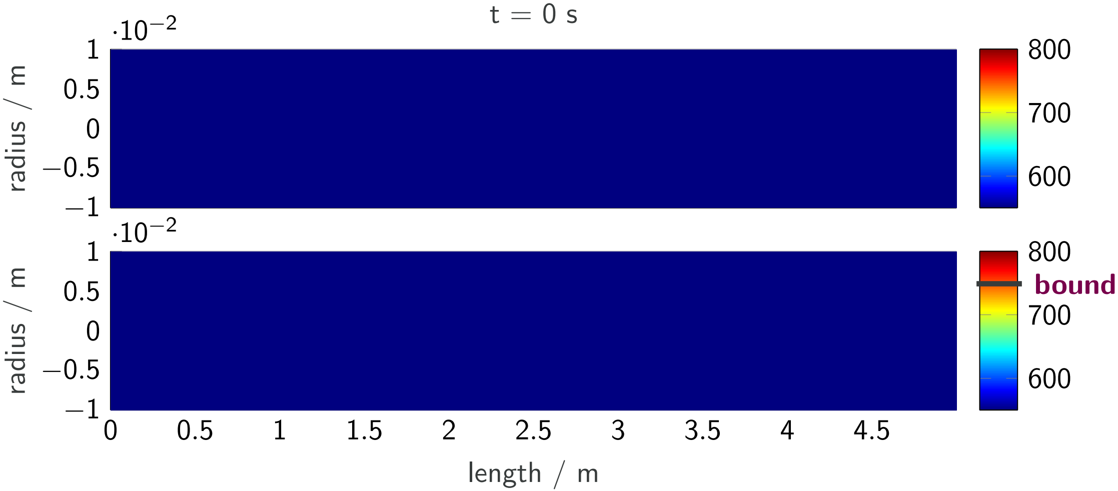


2D Results - Objective and Control



Optimal Control

2D Results - Spatial Temperature Distribution



Model Order Reduction

Background

- Need a **fine spatial** discretization to capture all important system dynamics.

  $\rightsquigarrow$ a large number of ODE equations, e.g., $\mathcal{O}(10^3 - 10^5)$.

Background

- Need a **fine spatial** discretization to capture all important system dynamics.

  $\rightsquigarrow$ a large number of ODE equations, e.g., $\mathcal{O}(10^3 - 10^5)$.

- But with such large-scale systems, these studies are **numerical inefficient**.

Model Order Reduction

Background

- Need a **fine spatial** discretization to capture all important system dynamics.

  $\rightsquigarrow$ a large number of ODE equations, e.g., $\mathcal{O}(10^3 - 10^5)$.

- But with such large-scale systems, these studies are **numerical inefficient**.
- **For example**, practical controllers often require small number of equations (say, $N = 10$) due to
  - real-time constraints,
  - increasing fragility for larger N.

Background

- Thus, we need **surrogate models**, having
  - less number of equations, and
  - desirable accuracy.

Background

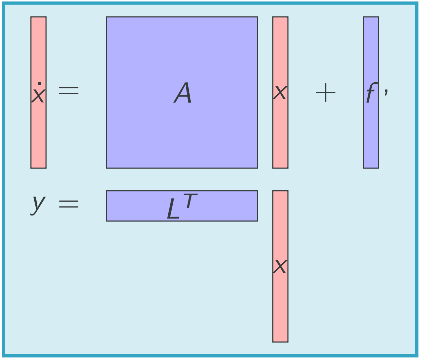
- Thus, we need **surrogate models**, having
  - less number of equations, and
  - desirable accuracy.

- Our approach is **Model Order Reduction** (MOR).

MOR Concept

- Consider the following ODE:

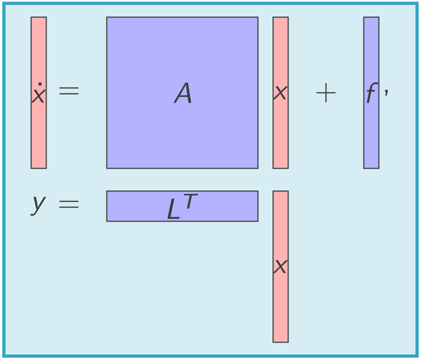
$$\dot{x}(t) = Ax(t) + f(x),$$
$$y(t) = L^T x(t).$$



MOR Concept

- Consider the following ODE:

$$\dot{x}(t) = Ax(t) + f(x),$$
$$y(t) = L^T x(t).$$



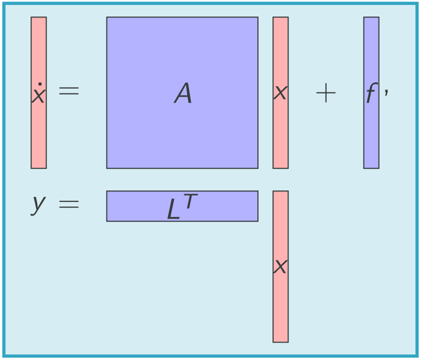
where

- $x \in \mathbb{R}^n$ are the system variables;
- y are the system output;
- A, L^T are constant matrices;
- f is a nonlinear function.

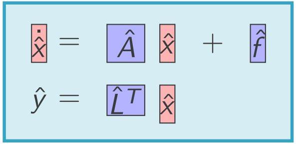
MOR Concept

- Consider the following ODE:

$$\dot{x}(t) = Ax(t) + f(x),$$
$$y(t) = L^T x(t).$$



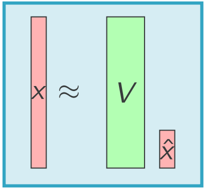
Reduced system (ROM)



- such that $y \approx \hat{y}$.

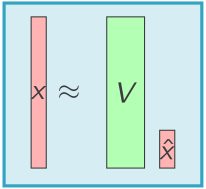
How to Construct ROM

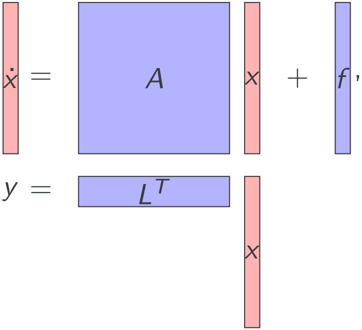
- Construct the projection matrix V, such that



How to Construct ROM

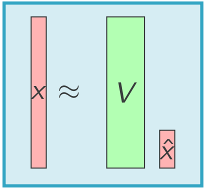
- Construct the projection matrix V, such that

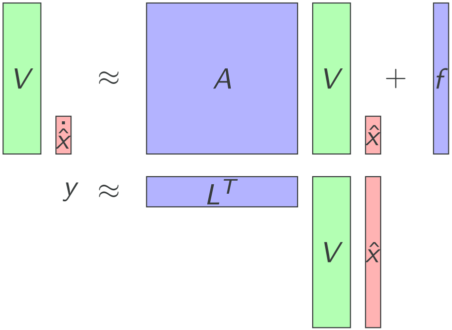




How to Construct ROM

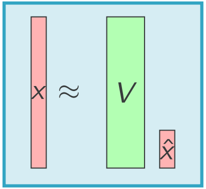
- Construct the projection matrix V, such that

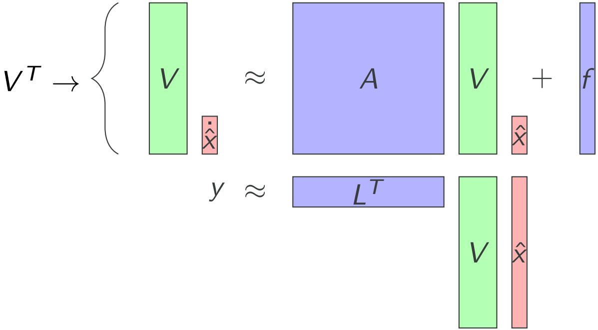




How to Construct ROM

- Construct the projection matrix V, such that





Model Order Reduction

How to Construct ROM

- Construct the projection matrix V, such that

$$x \approx V \hat{x}$$

$$V^T \rightarrow \left\{ \begin{array}{l} V \dot{\hat{x}} \approx A V \hat{x} + f \\ y \approx L^T V \hat{x} \end{array} \right.$$

- **The reduced-order system:**

$$\dot{\hat{x}} = \hat{A} \hat{x} + \hat{f}$$
$$\hat{y} = \hat{L}^T \hat{x}$$

Proper Orthogonal Decomposition

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

Proper Orthogonal Decomposition

- The quality of ROM depends on the choice of V.
- A common technique for nonlinear systems is proper orthogonal decomposition.

Proper Orthogonal Decomposition

- The quality of ROM depends on the choice of V.
- A common technique for nonlinear systems is proper orthogonal decomposition.
- **Proper Orthogonal Decomposition (POD)**
  - Take computed or experimental 'snapshots' of full model: $[x(t_1), x(t_2), \ldots, x(t_N)] := X$.

Proper Orthogonal Decomposition

- The quality of ROM depends on the choice of V.
- A common technique for nonlinear systems is proper orthogonal decomposition.
- **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: $X = U\Sigma W^T$.

Proper Orthogonal Decomposition

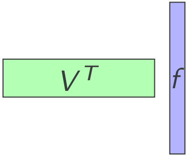
- The quality of ROM depends on the choice of V.
- A common technique for nonlinear systems is proper orthogonal decomposition.
- **Proper Orthogonal Decomposition (POD)**
  - Take computed or experimental 'snapshots' of full model: $[x(t_1), x(t_2), \ldots, x(t_N)] := X$.
  - Perform SVD of snapshot matrix: $X = U\Sigma W^T$.
  - Then, the projection matrix $V = U(:, 1:r)$.

Model Order Reduction

Computational Issue

- Observe the nonlinear term:

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



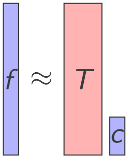
- Still, we need computations of the nonlinear function on the full grid.

  $\rightsquigarrow$ no reduction in computations.

- Therefore, we require **(Discrete) Empirical Interpolation Method** ((D)EIM).

Discrete Empirical Interpolation Method (DEIM)

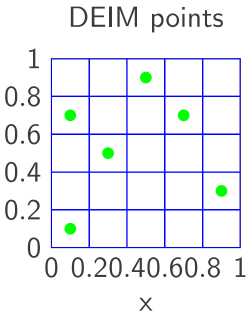
- The idea is:

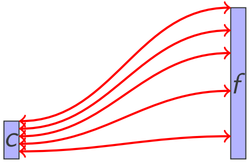


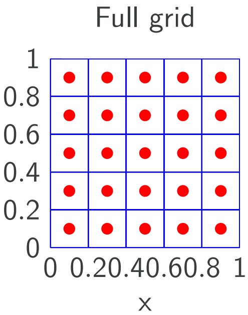
where

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

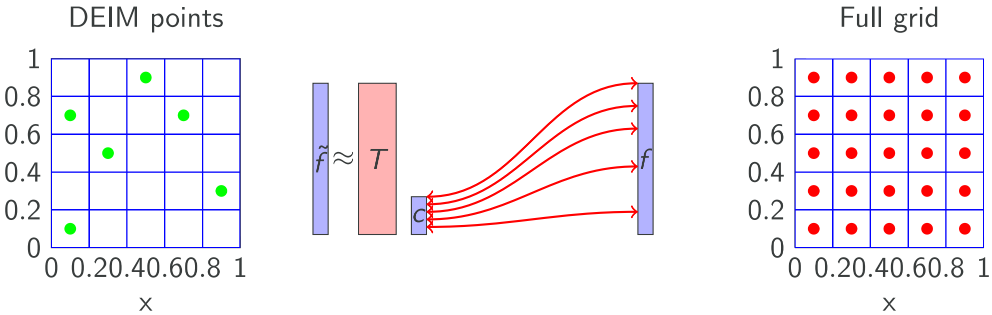
A graphical representation







A graphical representation



- A greedy algorithm to select the grid points (DEIM points) for the nonlinear function.

DEIM - Example

- Consider a nonlinear function:

$$f(x,\mu) = (1-x)\cdot \sin(2\pi\mu(x+1))\cdot e^{-(1+x)\mu}, \quad x\in[-1,1], \quad \mu\in[1,2\pi].$$

DEIM - Example

- Consider a nonlinear function:

$$f(x,\mu) = (1-x)\cdot\sin(2\pi\mu(x+1))\cdot e^{-(1+x)\mu}, \quad x\in[-1,1], \quad \mu\in[1,2\pi].$$

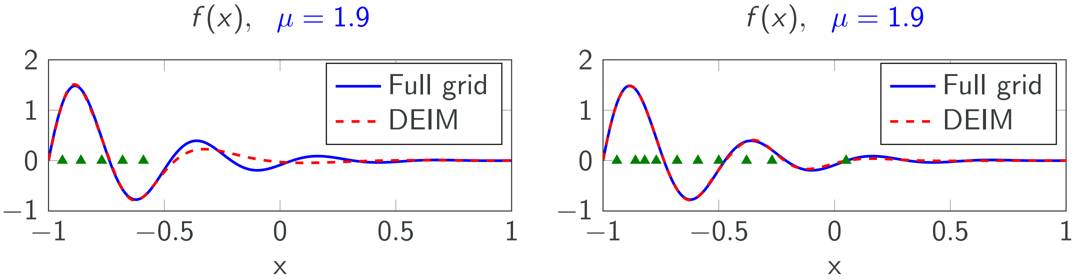
- We take 100 points on the grid and training sample: $\mu = 1 : 0.2 : 6.8$.

DEIM - Example

- Consider a nonlinear function:

$$f(x,\mu) = (1-x)\cdot\sin(2\pi\mu(x+1))\cdot e^{-(1+x)\mu}, \quad x\in[-1,1], \quad \mu\in[1,2\pi].$$

- We take 100 points on the grid and training sample: $\mu = 1 : 0.2 : 6.8$.



For 5 DEIM points. For 10 DEIM points.

Reactor Model State-Space Representaion

- Original system (FOM):

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

$$\Downarrow$$

$$\dot{\mathbf{x}} = \mathbf{A}_1(\mathbf{x})\,\mathbf{x} + \mathbf{A}_2\,\mathbf{x} + \mathbf{B}_1(\mathbf{x})\,\mathbf{u}_1 + \mathbf{B}_2\,\mathbf{u}_2 + \mathbf{f}(\mathbf{x})$$

- $\mathbf{A}_1 \uparrow$: $\mathcal{D}_{r,\alpha}$, a_r
- $\mathbf{x} \uparrow$: ρ_α, T
- $\mathbf{B}_1 \uparrow$: λ
- $\mathbf{u}_1 \uparrow$: T_{cool}
- $\mathbf{u}_2 \uparrow$: $\rho_{\alpha,\text{inlet}}$, T_{inlet}
- $\mathbf{f} \uparrow$: $\tilde{r}r_\beta$, $\mathcal{D}_{r,\alpha}$, a_r λ

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

Reduced Reactor Model

POD-DEIM applied to the reactor model

- POD-DEIM leads to the the following system:

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

$$\mathbf{y} = \mathbf{L}^{\mathrm{T}}\,\mathbf{x}^*. \qquad \text{with: } \mathbf{x}^* = \mathbf{V}\mathbf{x}_r$$

$$dim(\mathbf{x}_r) << dim(\mathbf{x})$$

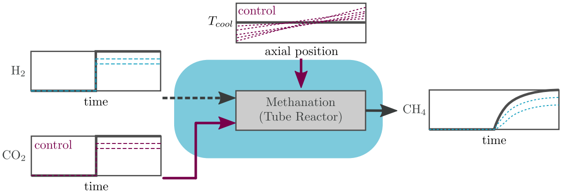
- $\mathbf{V}$ - from SVD of $\mathbf{x}$ snapshots (POD)
- $\mathbf{Q}_A$ - from SVD of $\mathbf{A}_1(\mathbf{x})\,\mathbf{x}$ snapshots (DEIM)
- $\mathbf{Q}_B$ - from SVD of $\mathbf{B}_1(\mathbf{x})\,\mathbf{x}$ snapshots (DEIM)
- $\mathbf{Q}_f$ - from SVD of $\mathbf{f}(\mathbf{x})$ snapshots (DEIM)

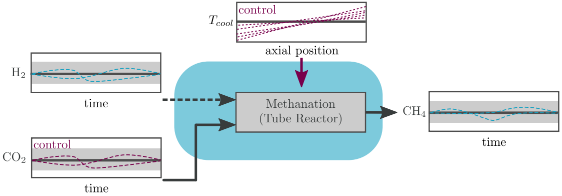
e.g., for $\mathbf{f}(\mathbf{x})$

$$SVD\begin{bmatrix} f_1(\mathbf{x}(t_1)) & f_1(\mathbf{x}(t_2)) & f_1(\mathbf{x}(t_3)) & \cdots \\ f_2(\mathbf{x}(t_1)) & f_1(\mathbf{x}(t_2)) & f_3(\mathbf{x}(t_3)) & \cdots \\ f_3(\mathbf{x}(t_1)) & f_1(\mathbf{x}(t_2)) & f_1(\mathbf{x}(t_3)) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Reduced Reactor Model

POD-DEIM applied to the reactor model



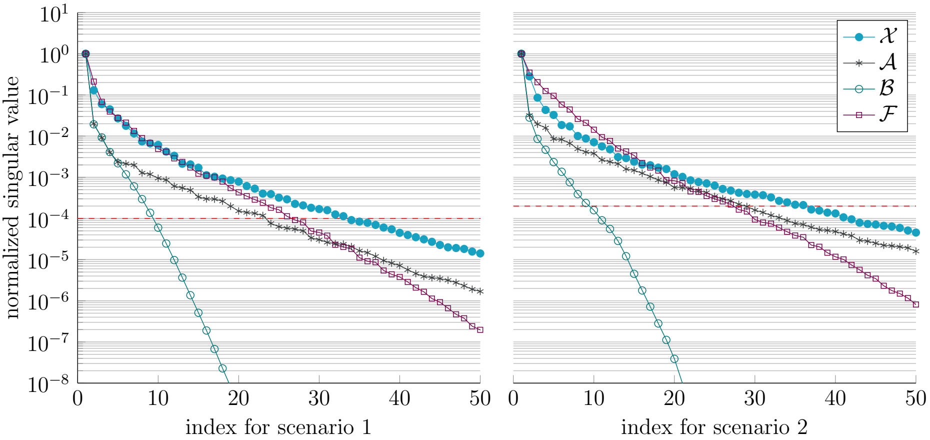


- The range of $x_{CO2} \in [0.7, 0.9]$.
- The range of $x_{H2} \in [0.1, 0.3]$.
- The range of $T_{cool} \in [500K, 700K]$.

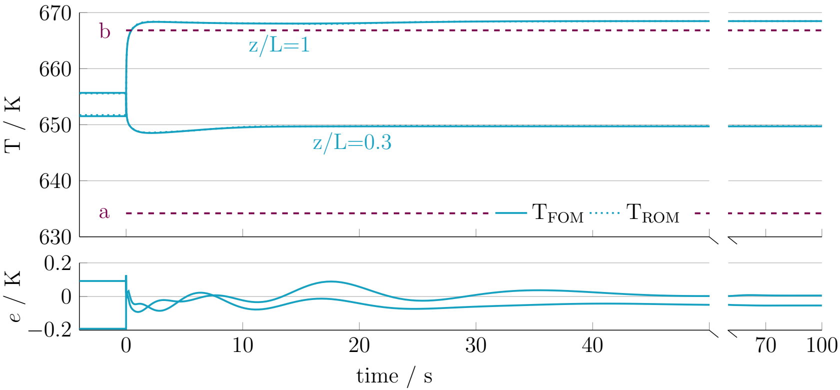
- No. Training cases: 50
- No. Test cases: 20

Reduced Reactor Model

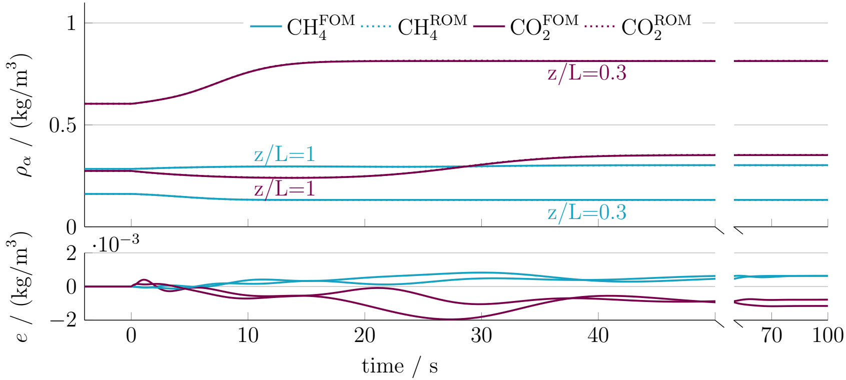
Singular Value Decay



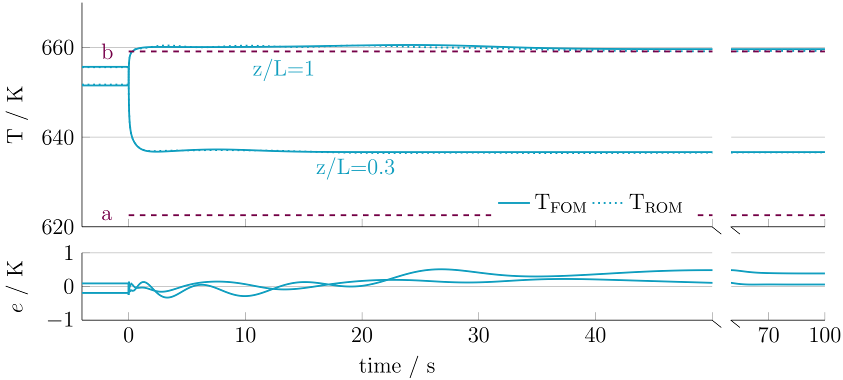
ROM vs. FOM - Continuous Operation Best Case



ROM vs. FOM - Continuous Operation Best Case

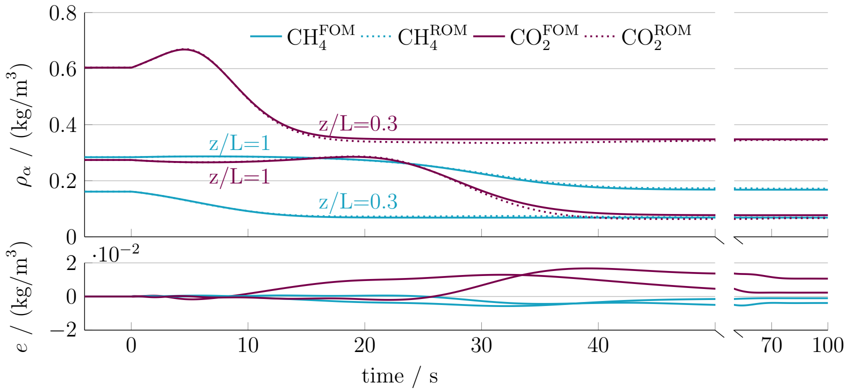


ROM vs. FOM - Continuous Operation Worst Case



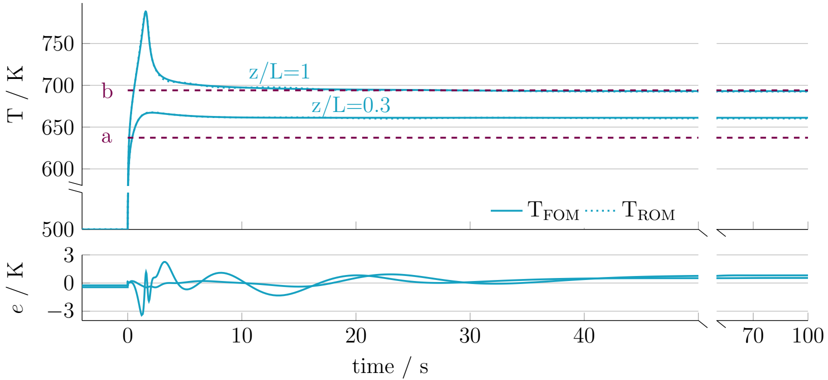
Reduced Reactor Model

ROM vs. FOM - Continuous Operation Worst Case



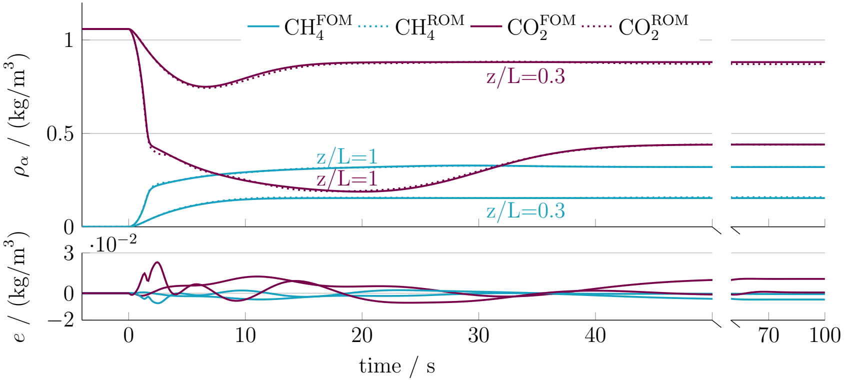
Reduced Reactor Model

ROM vs. FOM - Start-Up Best Case



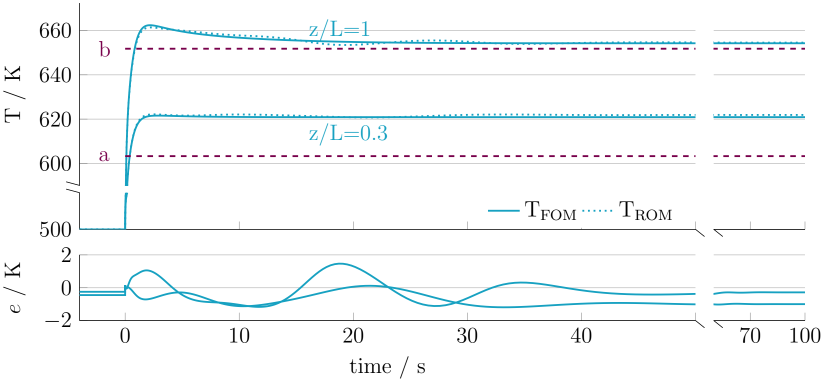
Reduced Reactor Model

ROM vs. FOM - Start-Up Best Case

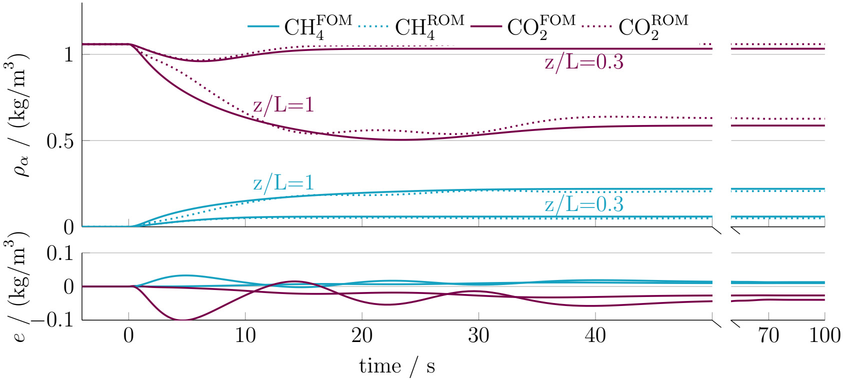


Reduced Reactor Model

ROM vs. FOM - Start-Up Worst Case



ROM vs. FOM - Start-Up Worst Case



Reduced 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

Potential of MOR in Dynamic Optimization

$$\max_{\mathbf{u}(t)} \quad \ldots,$$
$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)),$$
$$\mathbf{x}(t_0) = \mathbf{x}_0,$$
$$\vdots$$

$\xrightarrow{\text{POD DEIM}}$

$$\max_{\mathbf{u}(t)} \quad \ldots,$$
$$\text{s.t.} \quad \dot{\mathbf{x}}_r(t) = \mathbf{f}_r(\mathbf{x}_r(t), \mathbf{u}(t))$$
$$\mathbf{x}_r(t_0) = \mathbf{x}_{r,0}$$
$$\vdots$$

Order of FOM:

$dim(\mathbf{x}) = 350 - 5000$

Order of NLP:

$dim(\mathbf{x}) = 35'000 - 500'000$

CPU time / memory usage:

hours! / - - -

Order of ROM:

$dim(\mathbf{x}_r) \approx 34 - 36$

Order of NLP:

$dim(\mathbf{x}_r) \approx 3'400 - 3'600$

CPU time / memory usage:

??? / +++

Summary and Outlook

- Optimal start-up control of a packed-bed methanation reactor is **feasible**.

Summary and Outlook

- Optimal start-up control of a packed-bed methanation reactor is **feasible**.
- **POD-DEIM** has been successfully applied to construct reduced systems.

Summary and Outlook

- Optimal start-up control of a packed-bed methanation reactor is **feasible**.
- **POD-DEIM** has been successfully applied to construct reduced systems.
- POD-DEIM can be seen as a **multi-purpose tool** for nonlinear-MOR.

Summary and Outlook

- Optimal start-up control of a packed-bed methanation reactor is **feasible**.
- **POD-DEIM** has been successfully applied to construct reduced systems.
- POD-DEIM can be seen as a **multi-purpose tool** for nonlinear-MOR.
- Apply ROM to **optimization** studies and **controller design - (NMPC???)**.

Summary and Outlook

- Optimal start-up control of a packed-bed methanation reactor is **feasible**.
- **POD-DEIM** has been successfully applied to construct reduced systems.
- POD-DEIM can be seen as a **multi-purpose tool** for nonlinear-MOR.
- Apply ROM to **optimization** studies and **controller design - (NMPC???)**.
- Adaptive ROM update within the **optimization** - **(POD - trust region concept???)**.

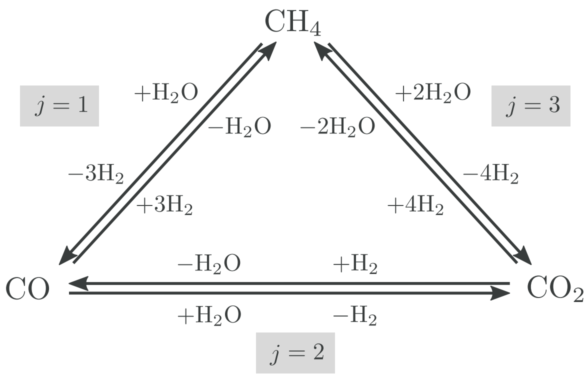
Summary and Outlook

- Optimal start-up control of a packed-bed methanation reactor is **feasible**.
- **POD-DEIM** has been successfully applied to construct reduced systems.
- POD-DEIM can be seen as a **multi-purpose tool** for nonlinear-MOR.
- Apply ROM to **optimization** studies and **controller design - (NMPC???)**.
- Adaptive ROM update within the **optimization** - **(POD - trust region concept???)**.
- Experimental validation of the reactor model.

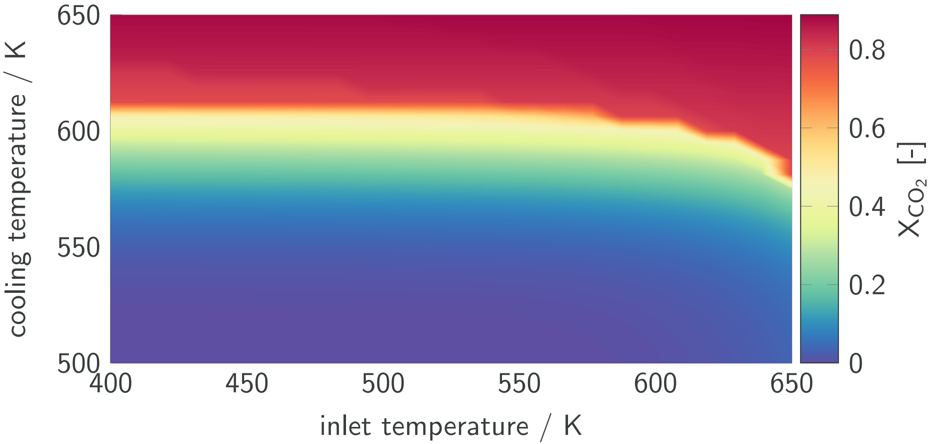
Summary and Outlook

- Optimal start-up control of a packed-bed methanation reactor is **feasible**.
- **POD-DEIM** has been successfully applied to construct reduced systems.
- POD-DEIM can be seen as a **multi-purpose tool** for nonlinear-MOR.
- Apply ROM to **optimization** studies and **controller design - (NMPC???)**.
- Adaptive ROM update within the **optimization** - **(POD - trust region concept???)**.
- Experimental validation of the reactor model.

Thank you for your attention !!!

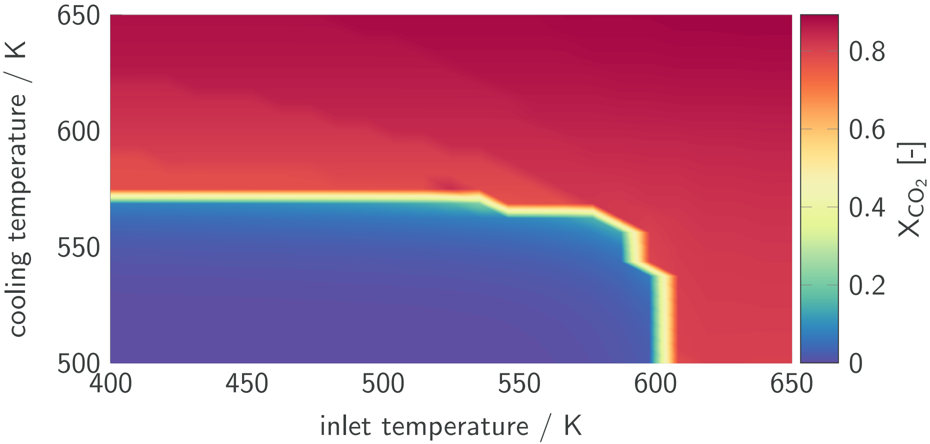

CH_4
$j = 1$
$+H_2O$
$-H_2O$
$-2H_2O$
$+2H_2O$
$j = 3$
$-3H_2$
$+3H_2$
$+4H_2$
$-4H_2$
$-H_2O$
$+H_2$
CO
CO_2
$+H_2O$
$-H_2$
$j = 2$

Appendix



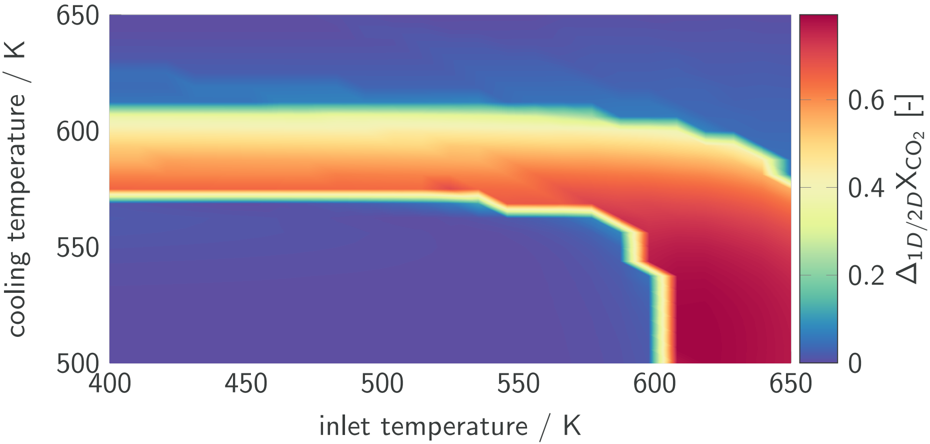
650
600
550
500
cooling temperature / K
400
450
500
550
600
650
inlet temperature / K
0.8
0.6
0.4
0.2
0
X_{CO_2} [-]

Appendix



650
600
550
500
cooling temperature / K
400
450
500
550
600
650
inlet temperature / K
0.8
0.6
0.4
0.2
0
X_{CO_2} [-]

Appendix



650
600
550
500
cooling temperature / K
400
450
500
550
600
650
inlet temperature / K
0.6
0.4
0.2
0
$\Delta_{1D/2D}X_{CO_2}$ [-]