Universität Münster

Adaptive model hierarchies for solving parametrized optimal control problems in multi-query scenarios

Seminar Talk – Università di Genova

Hendrik Kleikamp (University of Münster); based on projects with Bernard Haasdonk, Martin Lazar, Cesare Molinari, Mario Ohlberger, Lukas Renelt, Felix Schindler and Tizian Wenzel.

October 21, 2024

living.knowledge

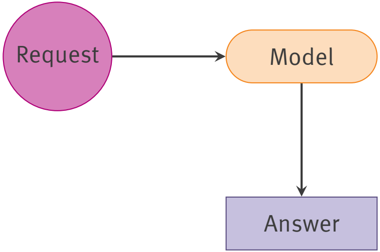
MM Mathematics Münster Cluster of Excellence

Adaptive model hierarchies

What is a model?

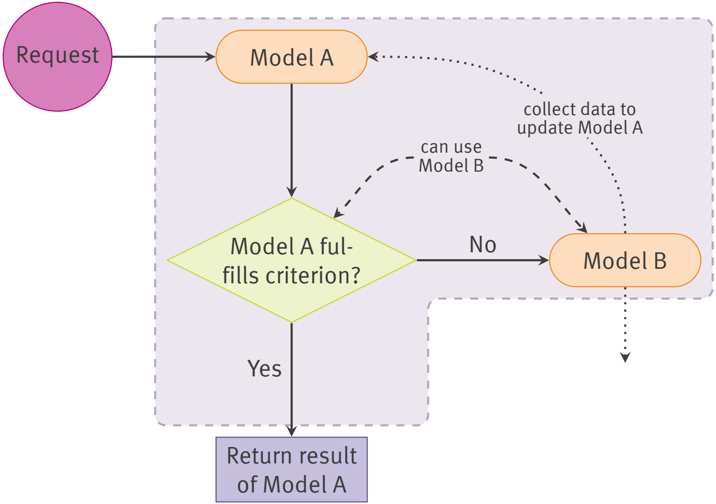
Assumptions:

- A model can handle requests and produces associated outputs.
- There is a notion of accuracy of a model and its output.
- There is a notion of computational effort or speed of a model.



Model hierarchies for multi-query problems [Haasdonk et al.'23]

The main building block

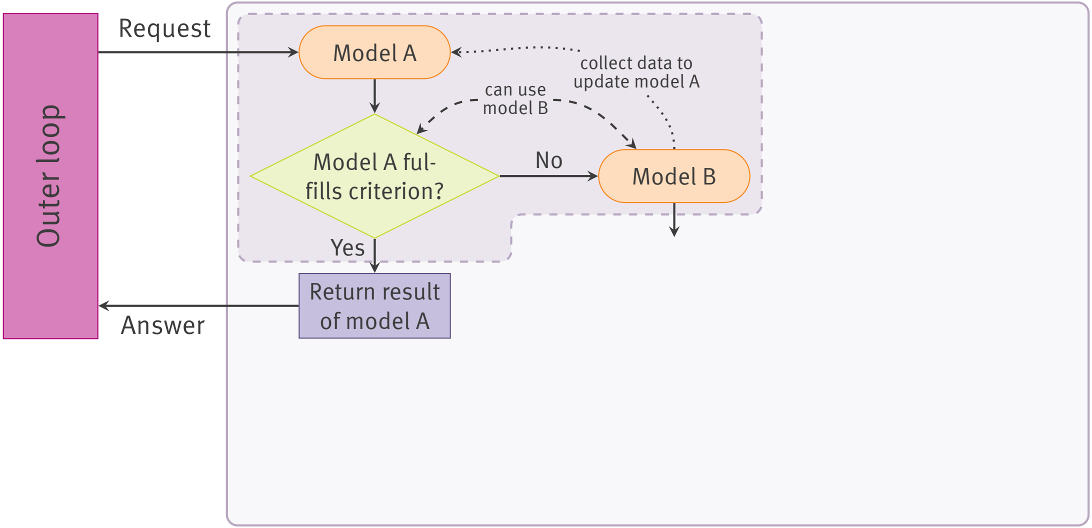


Assumptions:

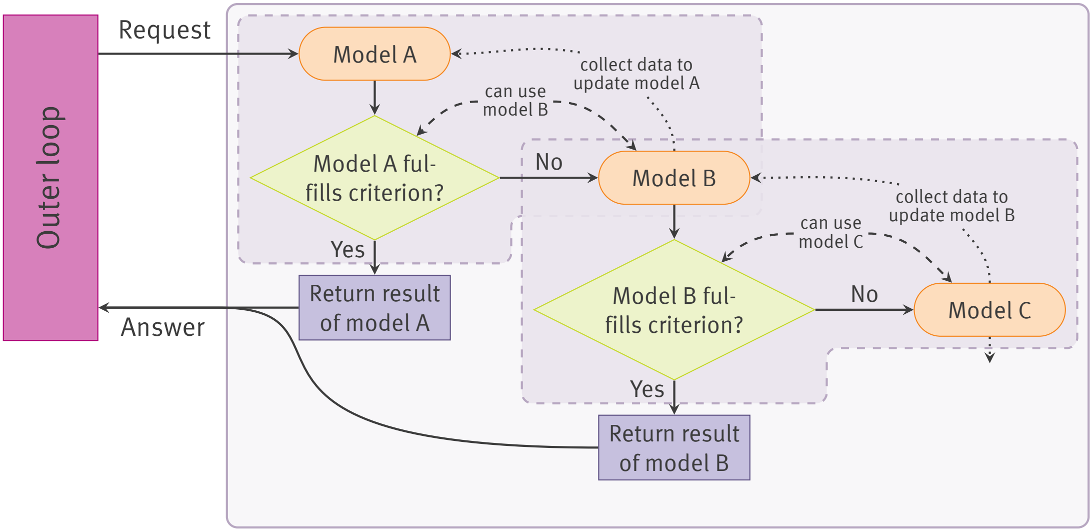
- Model A can be solved faster than Model B.
- Model B is more accurate than Model A.
- Model A can be improved by means of information from Model B.

Model hierarchies for multi-query problems

Application within an outer loop

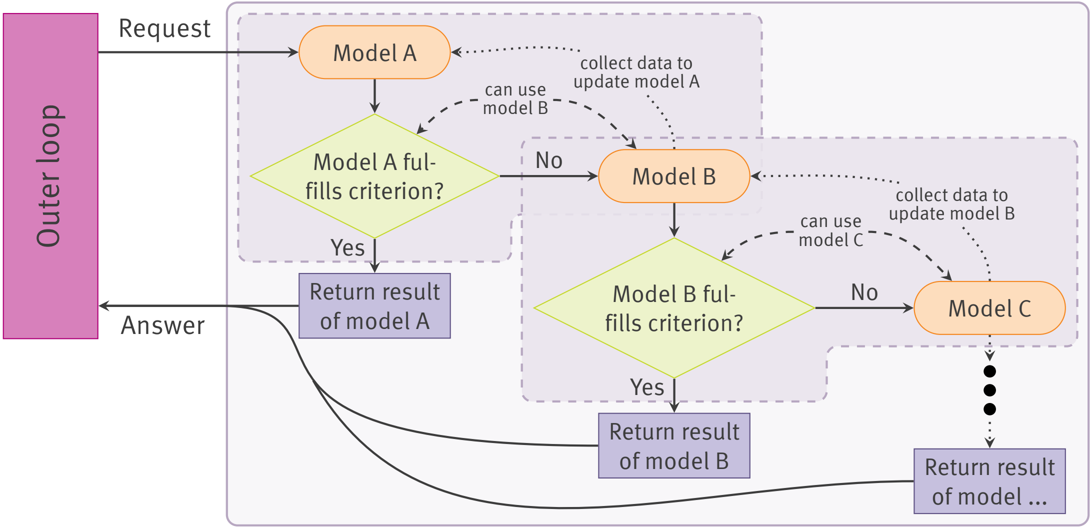


Model hierarchies for multi-query problems
Application within an outer loop



Model hierarchies for multi-query problems

Application within an outer loop



hendrik.kleikamp@uni-muenster.de

Model hierarchies for multi-query problems
Key points and ideas

- No offline training phase required

hendrik.kleikamp@uni-muenster.de

Model hierarchies for multi-query problems
Key points and ideas

- No offline training phase required
- Certified answers to all requests

Model hierarchies for multi-query problems
Key points and ideas

- No offline training phase required
- Certified answers to all requests
- Returns sufficiently accurate results as fast as possible

hendrik.kleikamp@uni-muenster.de

Model hierarchies for multi-query problems
Key points and ideas

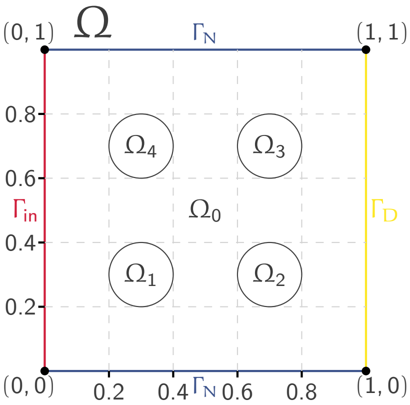
- No offline training phase required
- Certified answers to all requests
- Returns sufficiently accurate results as fast as possible
- Adaptive refinement of models using data that is computed anyways

hendrik.kleikamp@uni-muenster.de

Parametrized optimal control problems

The cookie baking example [Rave/Saak'21]

Problem setting



Computational domain

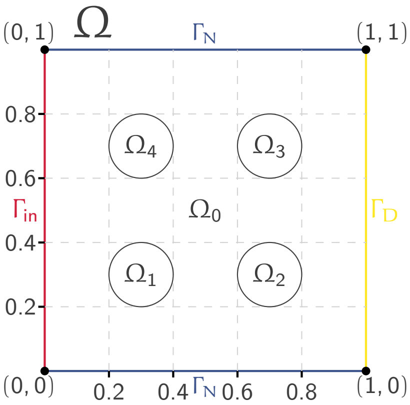
- State equation:

$$\partial_t \theta(t,\xi;\mu) - \nabla \cdot (\sigma(t,\xi;\mu)\nabla\theta(t,\xi;\mu)) = 0$$
$$\sigma(t,\xi;\mu)\nabla\theta(t,\xi;\mu)\cdot\vec{n}(\xi) = u(t), \quad \xi \in \Gamma_{in},$$

\+ homogeneous initial, Dirichlet and Neumann conditions

The cookie baking example [Rave/Saak'21]

Problem setting



Computational domain

- State equation:

$$\partial_t \theta(t,\xi;\mu) - \nabla \cdot (\sigma(t,\xi;\mu)\nabla\theta(t,\xi;\mu)) = 0$$
$$\sigma(t,\xi;\mu)\nabla\theta(t,\xi;\mu)\cdot\vec{n}(\xi) = u(t), \quad \xi \in \Gamma_{in},$$

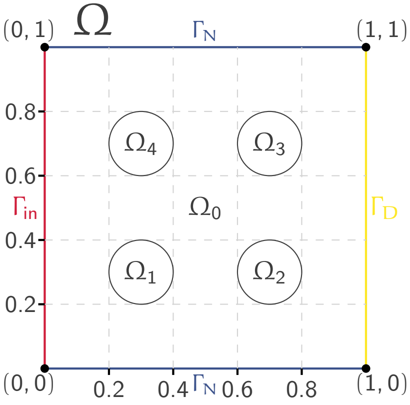
+ homogeneous initial, Dirichlet and Neumann conditions

- Parametric and time-dependent diffusivity:

$$\sigma(t,\xi;\mu) := \begin{cases} 14\cdot(t-0.25)^2 + 0.125, & \text{for } \xi \in \Omega_0, \\ \mu_1, & \text{for } \xi \in \Omega_1 \cup \Omega_3, \\ \mu_2, & \text{for } \xi \in \Omega_2 \cup \Omega_4. \end{cases}$$

The cookie baking example [Rave/Saak'21]

Problem setting



Computational domain

- State equation:

$$\partial_t \theta(t,\xi;\mu) - \nabla\cdot\left(\sigma(t,\xi;\mu)\nabla\theta(t,\xi;\mu)\right) = 0$$
$$\sigma(t,\xi;\mu)\nabla\theta(t,\xi;\mu)\cdot\vec{n}(\xi) = u(t),\quad \xi\in\Gamma_{in},$$

+ homogeneous initial, Dirichlet and Neumann conditions

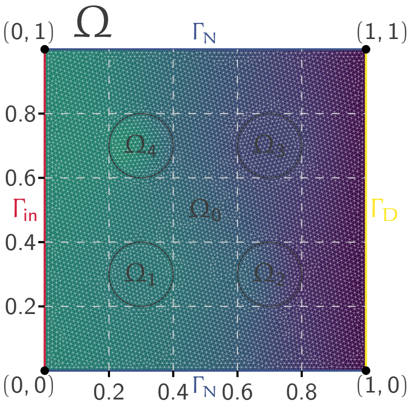
- Parametric and time-dependent diffusivity:

$$\sigma(t,\xi;\mu) := \begin{cases} 14\cdot(t-0.25)^2 + 0.125, & \text{for } \xi\in\Omega_0, \\ \mu_1, & \text{for } \xi\in\Omega_1\cup\Omega_3, \\ \mu_2, & \text{for } \xi\in\Omega_2\cup\Omega_4. \end{cases}$$

- Output quantities:

$$y_i(t;\mu) := \frac{1}{|\Omega_i|}\int_{\Omega_i}\theta(t,\xi;\mu)\,d\xi \qquad \text{for } i=1,\dots,4.$$

The cookie baking example
Problem setting

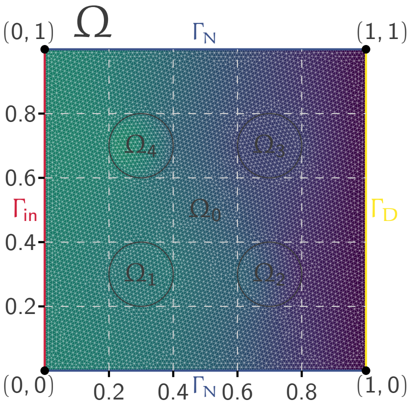


Optimal solution at time $t = 1$ for $\mu = (100, 0.1)$

- Discretized state system:

$$E \frac{d}{dt} x_\mu(t) = A(\mu; t) x_\mu(t) + B u_\mu(t),$$
$$y(t; \mu) = C x_\mu(t).$$

The cookie baking example
Problem setting



Optimal solution at time $t = 1$
for $\mu = (100, 0.1)$

- Discretized state system:

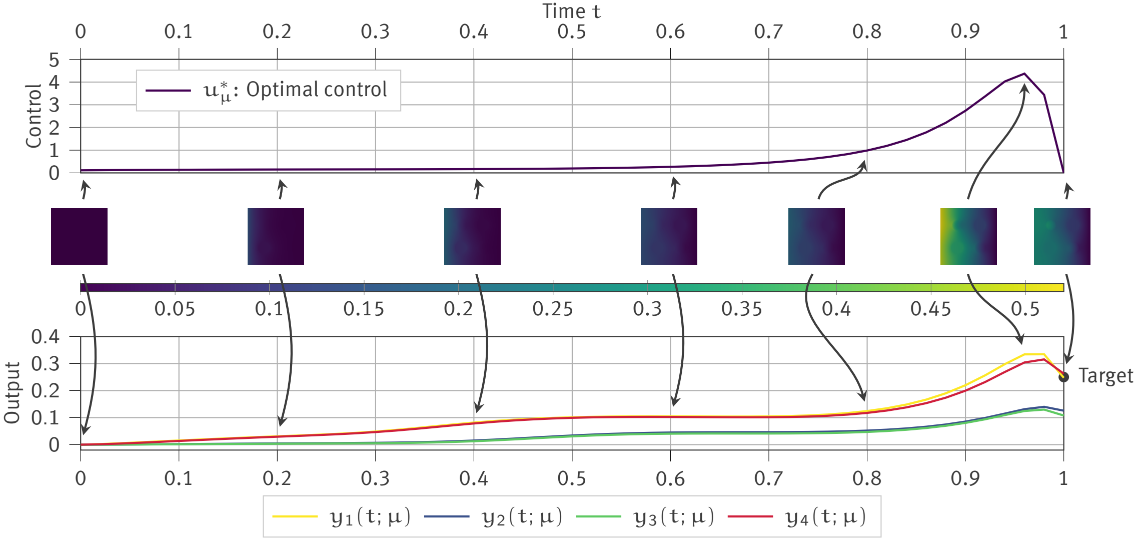
$$E\frac{d}{dt}x_\mu(t) = A(\mu;t)x_\mu(t) + Bu_\mu(t),$$
$$y(t;\mu) = Cx_\mu(t).$$

- Optimal control problem:

$$\min_{u\in G} \frac{1}{2}\left[\underbrace{\left\| C\left(x_\mu(T) - x_\mu^T\right)\right\|^2}_{\text{deviation of output}} + \underbrace{\int_0^T \langle u(t), R(t)u(t)\rangle \, dt}_{\text{control energy}}\right]$$

The cookie baking example

Optimal control and evolution of output for $\mu = (100, 0.1)$



Overview of the involved (reduced) models

Components applied in the hierarchy

- **Main goal:** Solve optimal control problem fast for many parameters!

Overview of the involved (reduced) models

Components applied in the hierarchy

- **Main goal:** Solve optimal control problem fast for many parameters!

Different models available:

- Full order model
- Reduced basis reduced order model [K./Lazar/Molinari'24]
- Machine learning surrogate [Hesthaven/Ubbiali'18]

Details on the machine learning surrogate

- Builds on the reduced basis model

Details on the machine learning surrogate

- Builds on the reduced basis model
- Uses the same underlying reduced space

hendrik.kleikamp@uni-muenster.de

Details on the machine learning surrogate

- Builds on the reduced basis model
- Uses the same underlying reduced space
- Approximates the map from parameter to reduced coefficients

Details on the machine learning surrogate

- Builds on the reduced basis model
- Uses the same underlying reduced space
- Approximates the map from parameter to reduced coefficients
- Connection to reduced basis model allows to apply the same a posteriori error estimator

hendrik.kleikamp@uni-muenster.de

Details on the machine learning surrogate

- Builds on the reduced basis model
- Uses the same underlying reduced space
- Approximates the map from parameter to reduced coefficients
- Connection to reduced basis model allows to apply the same a posteriori error estimator
- Different supervised machine learning methods are applicable, for instance neural networks, kernel methods, Gaussian process regression

Summary and comparison of the models

Full order model (FOM)

Pros:

- arbitrarily accurate solutions (serve as reference)

Cons:

- very slow when dealing with large systems

Summary and comparison of the models

Full order model (FOM)

Pros:

- arbitrarily accurate solutions (serve as reference)

Cons:

- very slow when dealing with large systems

Reduced order model (ROM)

Pros:

- faster than FOM
- reliable a posteriori error estimator available

Cons:

- still relatively slow due to high-dimensional computations

Summary and comparison of the models

Full order model (FOM)

Pros:

- arbitrarily accurate solutions (serve as reference)

Cons:

- very slow when dealing with large systems

Reduced order model (ROM)

Pros:

- faster than FOM
- reliable a posteriori error estimator available

Cons:

- still relatively slow due to high-dimensional computations

Machine learning model (MLM)

Pros:

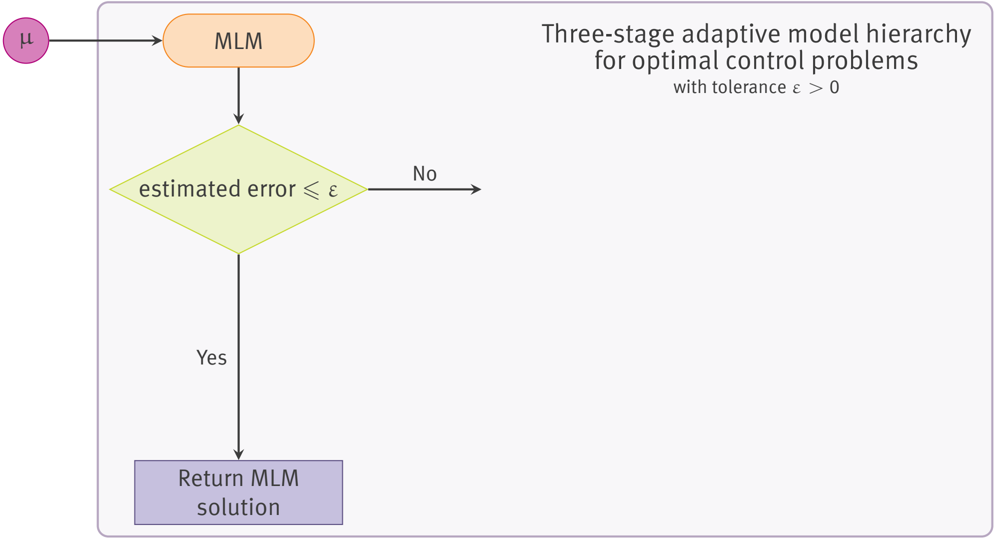
- faster than ROM
- reuses error estimator of the ROM

Cons:

- typically requires lots of training data and hyper-parameter tuning

hendrik.kleikamp@uni-muenster.de

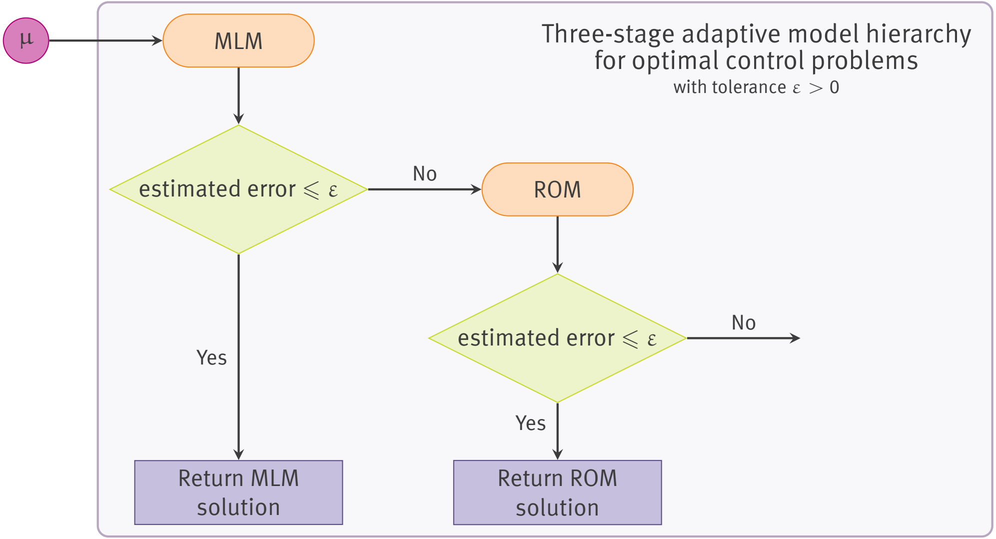
Three-stage model hierarchy for optimal control problems



Three-stage model hierarchy for optimal control problems

Universität Münster

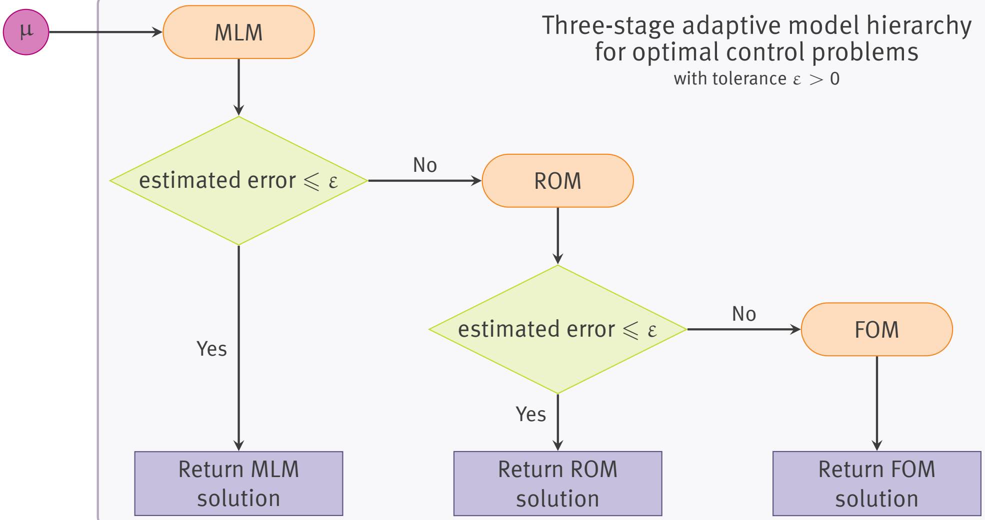
MM



hendrik.kleikamp@uni-muenster.de

Three-stage model hierarchy for optimal control problems

Universität Münster MM



Three-stage adaptive model hierarchy for optimal control problems

with tolerance $\varepsilon > 0$

Numerical experiment

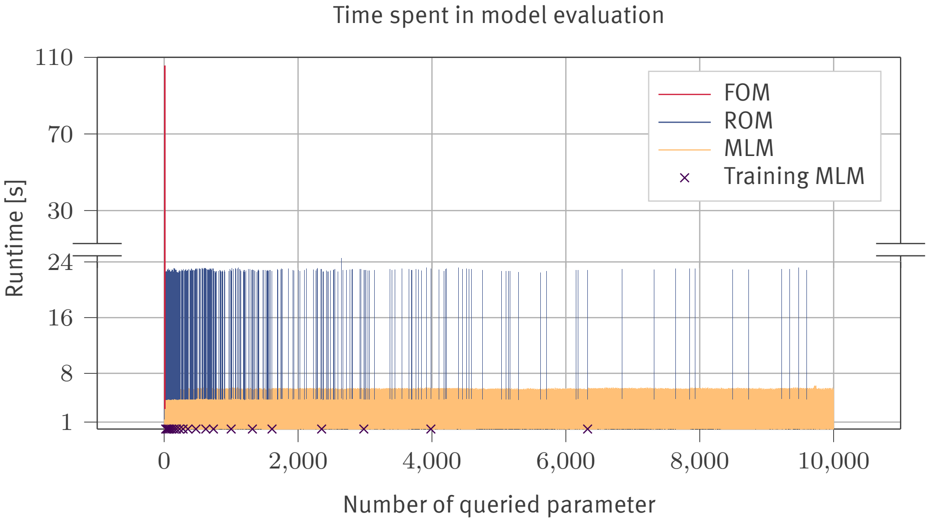
Numerical results: Applying the model hierarchy

The cookie baking example revisited

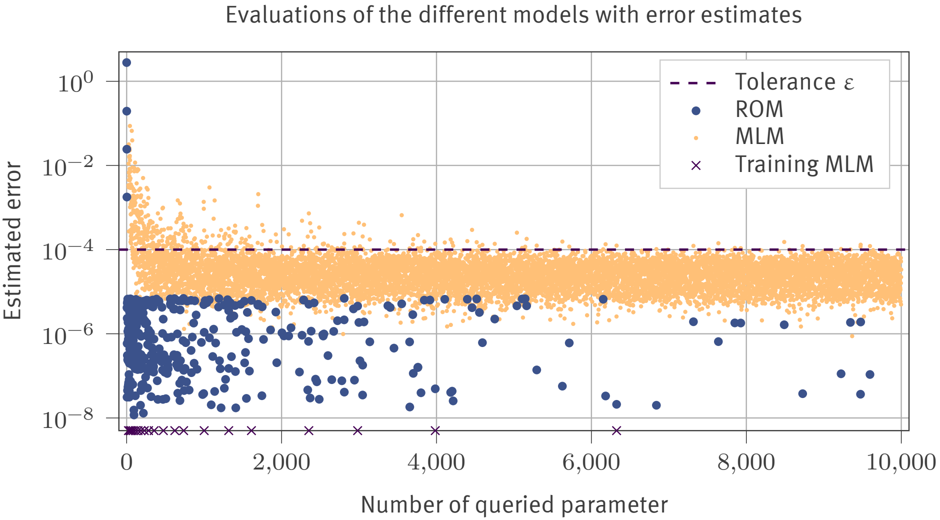
Model	Number of solves	Number of error estimates	Total time for error est. and solving [s]	Average time for error est. and solving per solve [s]
FOM	4	—	330.31	82.58
ROM	412	416	7,653.35	18.58
MLM	9,584	10,000	56,776.25	5.92

Numerical results: Applying the model hierarchy

The cookie baking example revisited



Numerical results: Applying the model hierarchy
The cookie baking example revisited



Outlook

- Extension of the reduced order models to other optimal control problems ($\rightarrow$ ongoing discussions with Cesare)

Outlook

- Extension of the reduced order models to other optimal control problems ($\longrightarrow$ ongoing discussions with Cesare)

- Application of different machine learning techniques

hendrik.kleikamp@uni-muenster.de

Outlook

- Extension of the reduced order models to other optimal control problems ($\rightarrow$ ongoing discussions with Cesare)
- Application of different machine learning techniques
- Construction of adaptive model hierarchies in other contexts

Outlook

- Extension of the reduced order models to other optimal control problems
  ($\longrightarrow$ ongoing discussions with Cesare)
- Application of different machine learning techniques
- Construction of adaptive model hierarchies in other contexts
- Develop strategies for dealing with changing tolerances when querying the model hierarchy

For more details, see:

B. Haasdonk, H. Kleikamp, M. Ohlberger, F. Schindler, and T. Wenzel.
A new certified hierarchical and adaptive RB-ML-ROM surrogate model for parametrized PDEs, (2023).

H. Kleikamp, M. Lazar, and C. Molinari.
Be greedy and learn: efficient and certified algorithms for parametrized optimal control problems, (2024).

H. Kleikamp.
Application of an adaptive model hierarchy to parametrized optimal control problems, (2024).

H. Kleikamp and L. Renelt.
Two-stage model reduction approaches for the efficient and certified solution of parametrized optimal control problems, (2024).

The source code for the papers is available open source:

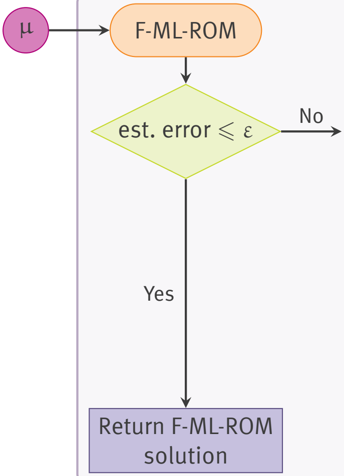
- Greedy and learn: `https://doi.org/10.5281/zenodo.8188417`
- Three-stage hierarchy: `https://doi.org/10.5281/zenodo.10669855`
- Fully reduced models: `https://doi.org/10.5281/zenodo.13382950`
- Four-stage hierarchy: `https://doi.org/10.5281/zenodo.13652744`

hendrik.kleikamp@uni-muenster.de

Thank you for your attention!

hendrik.kleikamp@uni-muenster.de

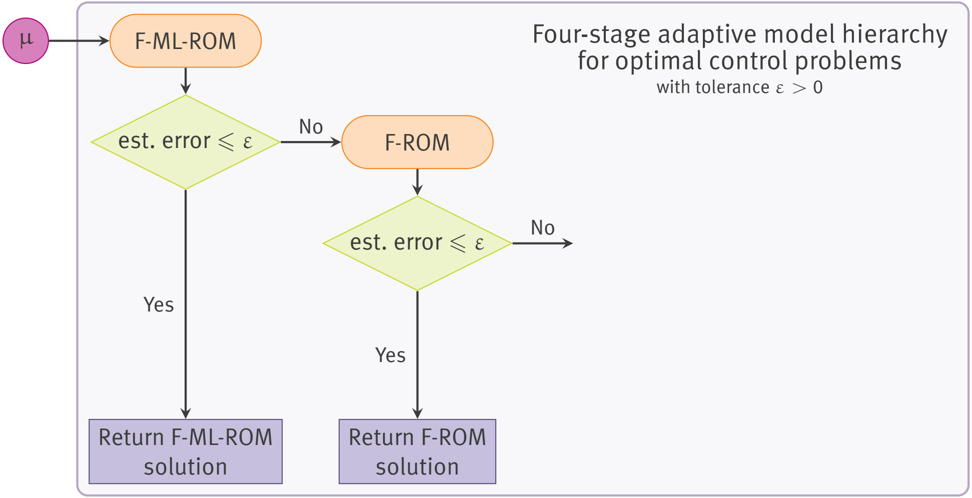
Four-stage model hierarchy for optimal control problems



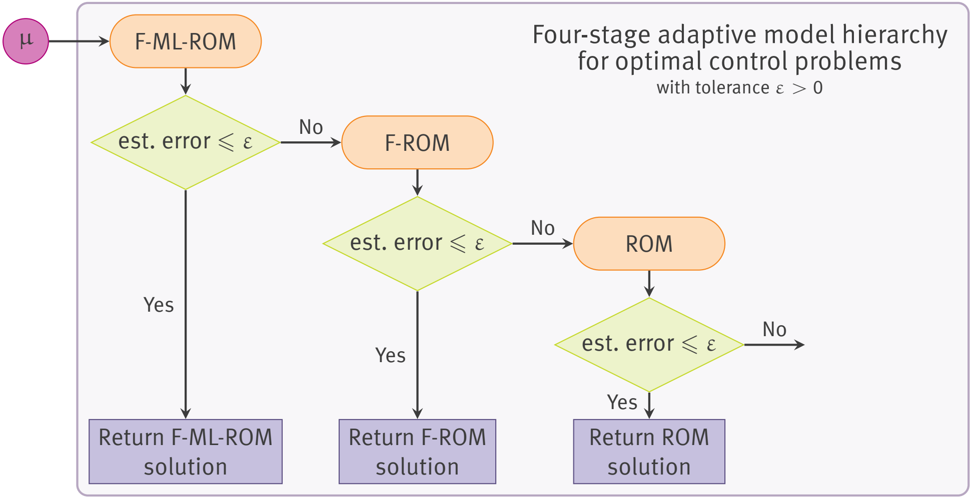
Four-stage adaptive model hierarchy for optimal control problems

with tolerance $\varepsilon > 0$

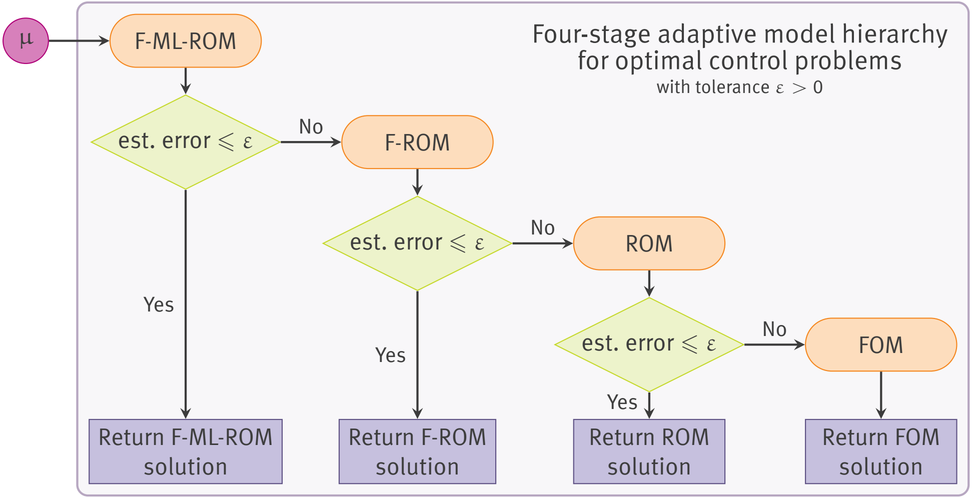
Four-stage model hierarchy for optimal control problems



Four-stage model hierarchy for optimal control problems



Four-stage model hierarchy for optimal control problems



Numerical results: Applying the four-stage model hierarchy

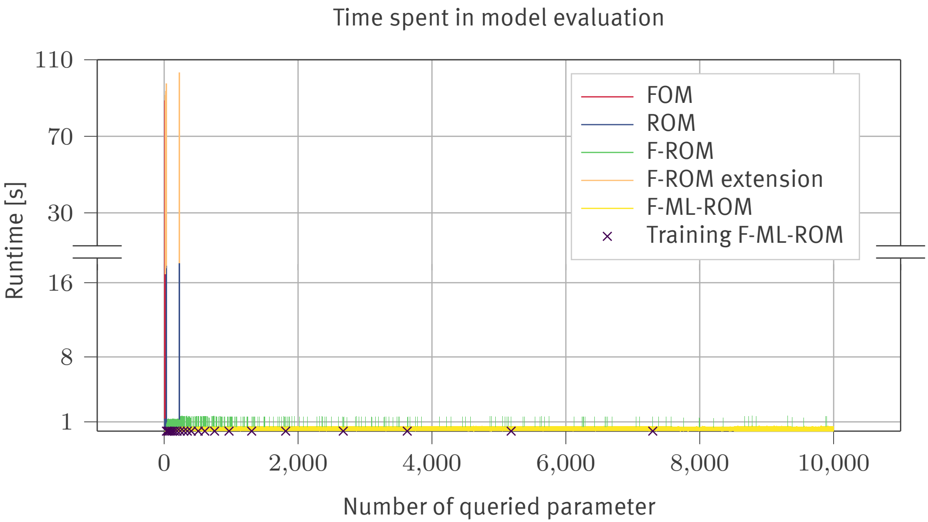
The cookie baking example revisited

Model	Number of solves	Number of error estimates	Total time for error est. and solving [s]	Average time for error est. and solving per solve [s]
FOM	4	—	304.95	76.24
ROM	12	16	234.56	19.55
F-ROM	437	453	450.39	1.03
F-ML-ROM	9,547	10,000	5,136.95	0.54

hendrik.kleikamp@uni-muenster.de

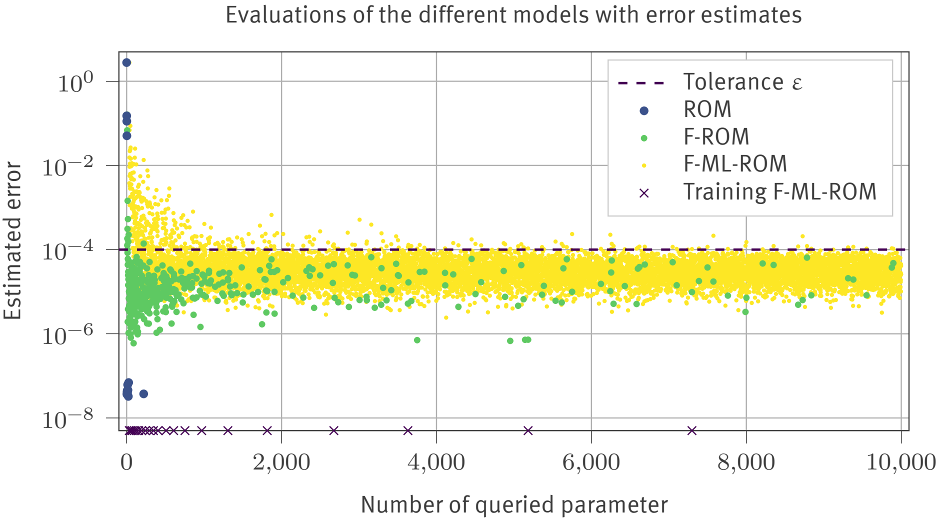
Numerical results: Applying the four-stage model hierarchy

The cookie baking example revisited



Numerical results: Applying the four-stage model hierarchy

The cookie baking example revisited



Parametrized optimal control problems
General setting

For a parameter $\mu \in \mathcal{P}$, solve

$$\min_{u \in G} \quad \frac{1}{2}\Bigg[\underbrace{\| C\left(x_\mu(T) - x_\mu^T\right) \|^2}_{\text{deviation of output}} + \underbrace{\int_0^T \langle u(t), R(t)u(t)\rangle \, dt}_{\text{control energy}}\Bigg],$$

$$\text{such that} \quad E\frac{d}{dt}x_\mu(t) = A(\mu)x_\mu(t) + B(\mu)u(t) \quad \text{for almost all } t \in [0,T],$$
$$x_\mu(0) = x_\mu^0 \in \mathbb{R}^n,$$

where

$$x_\mu \colon [0,T] \to \mathbb{R}^n, \qquad u \colon [0,T] \to \mathbb{R}^m, \qquad C \in \mathbb{R}^{l \times n}$$
$$E \in \mathbb{R}^{n \times n}, \qquad A(\mu) \in \mathbb{R}^{n \times n}, \qquad B(\mu) \in \mathbb{R}^{n \times m}.$$

The matrix $R(t) \in \mathbb{R}^{m \times m}$ is assumed to be positive-definite for almost all $t \in [0,T]$.

Optimality system

Theorem [K./Renelt'24]

The optimal state x_μ^*, control u_μ^* and adjoint φ_μ^* can be characterized as solution to the system:

$$\begin{aligned} E\frac{d}{dt}x_\mu(t) &= A(\mu;t)x_\mu(t) + B(\mu;t)u_\mu(t), \\ u_\mu(t) &= -R(t)^{-1}B(\mu;t)^\top \varphi_\mu(t), \\ -E^\top \frac{d}{dt}\varphi_\mu(t) &= A(\mu;t)^\top \varphi_\mu(t), \end{aligned}$$

for almost all $t \in [0, T]$ with initial respectively terminal conditions

$$x_\mu(0) = x_\mu^0, \qquad E^\top \varphi_\mu(T) = C^\top C\left(x_\mu(T) - x_\mu^T\right).$$

Universität Münster MM

Lemma [K./Renelt'24]

The optimal final time adjoint $\varphi_\mu^*(T) \in \mathbb{R}^n$ is given as the solution to the linear system

$$\boxed{S(\mu)\varphi_\mu^*(T) = g(\mu)}$$

with

$$S(\mu) \coloneqq E^\top + C^\top C\Lambda(\mu) \in \mathbb{R}^{n\times n} \qquad \text{and} \qquad g(\mu) \coloneqq C^\top C\left(\Psi_\mu(T,0)x_\mu^0 - x_\mu^T\right) \in \mathbb{R}^n.$$

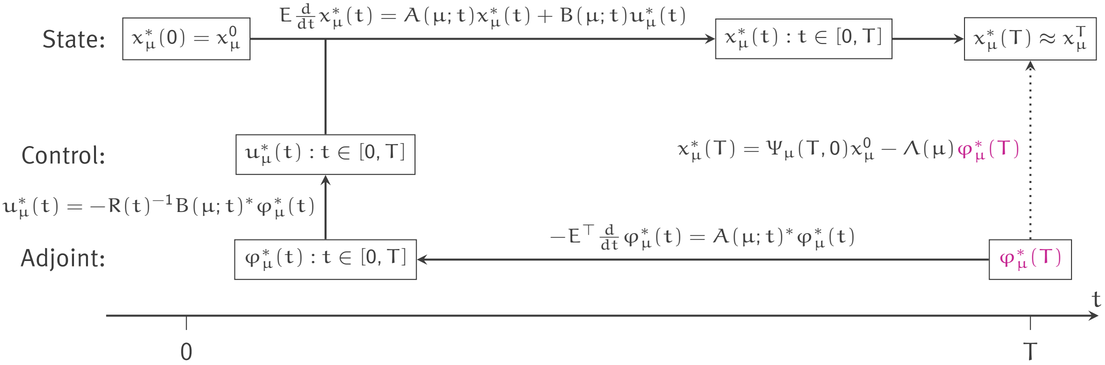
For $p \in \mathbb{R}^n$ we define

$$\Lambda(\mu)p \coloneqq -x_\mu(T)$$

as the solution of the optimality system for final time adjoint $\varphi_\mu(T) = p$ and $x_\mu(0) = 0$.

hendrik.kleikamp@uni-muenster.de

Visualization of the primal, adjoint and control equations



hendrik.kleikamp@uni-muenster.de

Reduced order modeling

Fully reduced model
Overview of the main components

- **Main goal:** Approximate optimal final time adjoints fast!

Fully reduced model
Overview of the main components

- **Main goal:** Approximate optimal final time adjoints fast!
- To this end: Reduced order models...

Fully reduced model
Overview of the main components

- **Main goal:** Approximate optimal final time adjoints fast!

- To this end: Reduced order models...

1. Reduced order model for optimal final time adjoints [K./Lazar/Molinari'24]
   $\rightarrow$ still involves high-dimensional computations

Fully reduced model
Overview of the main components

- **Main goal:** Approximate optimal final time adjoints fast!

- To this end: Reduced order models...

1. Reduced order model for optimal final time adjoints [K./Lazar/Molinari'24]
   $\longrightarrow$ still involves high-dimensional computations
2. Reduced order model for dynamical systems [Haasdonk/Ohlberger'11]
   $\longrightarrow$ approximation of primal and adjoint system

Fully reduced model

Overview of the main components

- **Main goal:** Approximate optimal final time adjoints fast!

- To this end: Reduced order models...

1. Reduced order model for optimal final time adjoints [K./Lazar/Molinari'24]
   $\rightarrow$ still involves high-dimensional computations
2. Reduced order model for dynamical systems [Haasdonk/Ohlberger'11]
   $\rightarrow$ approximation of primal and adjoint system
3. Combination of 1. and 2. to a fully reduced model [Fabrini/Iapichino/Volkwein'18],[K./Renelt'24]
   $\rightarrow$ completely independent of the high-dimensional state space

Motivation for approximation by a low-dimensional subspace

Lemma [K./Renelt'24]

For all $\mu \in \mathcal{P}$ we have

$$\varphi_\mu^*(T) = E^{-\top} C^\top C \left(\Psi_\mu(T, 0) x_\mu^0 - x_\mu^T - \Lambda(\mu) \varphi_\mu^*(T) \right) \in \operatorname{Im}(E^{-\top} C^\top C).$$

There holds in particular

$$\boxed{\dim \left(\operatorname{span} \left(\varphi_\mu^*(T) : \mu \in \mathcal{P} \right) \right) \leqslant \operatorname{rank}(C) \leqslant l} \qquad (\text{output dimension: } C \in \mathbb{R}^{l \times n}).$$

Part 1: Reduced model for final time adjoints (ROM)
Computation of the reduced solution

Given a linear N-dimensional subspace $X^N \subset \mathbb{R}^n$, approximate the optimal final time adjoint by

$$\tilde{\varphi}_\mu^N \coloneqq \operatorname*{arg\,min}_{p \in X^N} \| g(\mu) - S(\mu)p \|^2,$$

i.e. $\tilde{\varphi}_\mu^N$ is the least squares solution of the linear system.

Computation of the reduced solution

Given a linear N-dimensional subspace $X^N \subset \mathbb{R}^n$, approximate the optimal final time adjoint by

$$\boxed{\tilde{\varphi}_\mu^N \coloneqq \arg\min_{p \in X^N} \| g(\mu) - S(\mu)p \|^2,}$$

i.e. $\tilde{\varphi}_\mu^N$ is the least squares solution of the linear system.

Hence, we can compute $\tilde{\varphi}_\mu^N$ as the solution of

$$S(\mu)\tilde{\varphi}_\mu^N = P_{\mathcal{Y}_\mu^N}\big(g(\mu)\big),$$

where $\mathcal{Y}_\mu^N \subset \mathbb{R}^n$ is defined as

$$\mathcal{Y}_\mu^N \coloneqq S(\mu)X^N.$$

hendrik.kleikamp@uni-muenster.de

A posteriori error estimator

For a parameter $\mu \in \mathcal{P}$ and an approximate final time adjoint $p \in \mathbb{R}^n$, define the error estimator as

$$\boxed{\eta_\mu(p) \coloneqq C_{op} \, \| \, g(\mu) - S(\mu)p \, \| \, ,}$$

where C_{op} is a constant such that

$$C_{op} \geqslant \left\| \, S(\mu)^{-1} \, \right\| .$$

For a parameter $\mu \in \mathcal{P}$ and an approximate final time adjoint $p \in \mathbb{R}^n$, define the error estimator as

$$\boxed{\eta_\mu(p) \coloneqq C_{\mathrm{op}} \left\| g(\mu) - S(\mu)p \right\|,}$$

where C_{op} is a constant such that

$$C_{\mathrm{op}} \geqslant \left\| S(\mu)^{-1} \right\|.$$

Then we have

$$\boxed{\left\| \varphi_\mu^*(T) - p \right\| \leqslant \eta_\mu(p).}$$

Part 2: Reduced model for system dynamics

Definition of the reduced systems

Given matrices $V_{pr} \in \mathbb{R}^{n \times k_{pr}}$, $W_{pr} \in \mathbb{R}^{n \times k_{pr}}$, $V_{ad} \in \mathbb{R}^{n \times k_{ad}}$ and $W_{ad} \in \mathbb{R}^{n \times k_{ad}}$, we define the projected optimality system as

$$\begin{aligned} \hat{E}_{pr} \frac{d}{dt} \hat{x}_\mu(t) &= \hat{A}_{pr}(\mu; t) \hat{x}_\mu(t) + \hat{B}_{pr}(\mu; t) u(t), \\ \hat{u}_\mu(t) &= -R(t)^{-1} \hat{B}_{ad}(\mu; t)^\top \hat{\varphi}_\mu(t), \\ -\hat{E}_{ad} \frac{d}{dt} \hat{\varphi}_\mu(t) &= \hat{A}_{ad}(\mu; t)^\top \hat{\varphi}_\mu(t), \end{aligned}$$

with projected initial and terminal conditions

$$\hat{x}_\mu(0) = \hat{\bar{x}} = V_{pr}^\top \bar{x}, \qquad \hat{\varphi}_\mu(T) = \hat{\bar{\varphi}} = V_{ad}^\top \bar{\varphi},$$

where the projected system matrices are defined as

$$\begin{aligned} \hat{E}_{pr} &= W_{pr}^\top E V_{pr} \in \mathbb{R}^{k_{pr} \times k_{pr}}, & \hat{E}_{ad} &= W_{ad}^\top E V_{ad} \in \mathbb{R}^{k_{ad} \times k_{ad}}, \\ \hat{A}_{pr}(\mu; t) &= W_{pr}^\top A(\mu; t) V_{pr} \in \mathbb{R}^{k_{pr} \times k_{pr}}, & \hat{B}_{pr}(\mu; t) &= W_{pr}^\top B(\mu; t) \in \mathbb{R}^{k_{pr} \times m}, \\ \hat{A}_{ad}(\mu; t) &= W_{ad}^\top A(\mu; t) V_{ad} \in \mathbb{R}^{k_{ad} \times k_{ad}}, & \hat{B}_{ad}(\mu; t)^\top &= B(\mu; t)^\top V_{ad} \in \mathbb{R}^{m \times k_{ad}}. \end{aligned}$$

Part 2: Reduced model for system dynamics
Offline-online decomposition

- Parameter-separability:

$$A(\mu;t) = \sum_{q=1}^{Q_A} \theta_A^q(\mu;t)A_q \qquad \text{and} \qquad B(\mu;t) = \sum_{q=1}^{Q_B} \theta_B^q(\mu;t)B_q,$$

Part 2: Reduced model for system dynamics
Offline-online decomposition

- Parameter-separability:

$$A(\mu; t) = \sum_{q=1}^{Q_A} \theta_A^q(\mu; t) A_q \qquad \text{and} \qquad B(\mu; t) = \sum_{q=1}^{Q_B} \theta_B^q(\mu; t) B_q,$$

- Offline precomputations:

$$\hat{A}_{\text{pr}}^q = W_{\text{pr}}^\top A_q V_{\text{pr}} \qquad \text{and} \qquad \hat{B}_{\text{pr}}^q = W_{\text{pr}}^\top B_q$$

Part 2: Reduced model for system dynamics

Offline-online decomposition

- Parameter-separability:

$$A(\mu;t) = \sum_{q=1}^{Q_A} \theta_A^q(\mu;t)A_q \qquad \text{and} \qquad B(\mu;t) = \sum_{q=1}^{Q_B} \theta_B^q(\mu;t)B_q,$$

- Offline precomputations:

$$\hat{A}_{\text{pr}}^q = W_{\text{pr}}^\top A_q V_{\text{pr}} \qquad \text{and} \qquad \hat{B}_{\text{pr}}^q = W_{\text{pr}}^\top B_q$$

- Online phase:

$$\hat{A}_{\text{pr}}(\mu;t) = \sum_{q=1}^{Q_A} \theta_A^q(\mu;t)\hat{A}_{\text{pr}}^q \qquad \text{and} \qquad \hat{B}_{\text{pr}}(\mu;t) = \sum_{q=1}^{Q_B} \theta_B^q(\mu;t)\hat{B}_{\text{pr}}^q.$$

Define the residual for the primal equation as

$$R_{\mu}^{\text{pr}}\left[\hat{x}, u\right](t) \coloneqq A(\mu; t)\hat{x}(t) + B(\mu; t)u(t) - E\frac{d}{dt}\hat{x}(t).$$

Part 2: Reduced model for system dynamics
Residual-based a posteriori estimation

Define the residual for the primal equation as

$$R_\mu^{\mathrm{pr}}[\hat{x}, u](t) := A(\mu; t)\hat{x}(t) + B(\mu; t)u(t) - E\frac{d}{dt}\hat{x}(t).$$

Then we can estimate the error for the reduced primal solution as

$$\boxed{\| x_\mu(t) - \hat{x}_\mu(t) \| \leqslant \Delta_\mu^{\mathrm{pr}}[u](t).}$$

The error estimator $\Delta_\mu^{\mathrm{pr}}[u]$ is given as

$$\Delta_\mu^{\mathrm{pr}}[u](t) := C_1(\mu) \| x_\mu^0 - V_{\mathrm{pr}}V_{\mathrm{pr}}^\top x_\mu^0 \| + C_1(\mu)\int_0^t \| E^{-1}R_\mu^{\mathrm{pr}}[\hat{x}_\mu, u](s) \| \, ds$$

for a suitable constant $C_1(\mu)$.

Computation of the fully reduced solution

The fully reduced solution $\hat{\varphi}_{\mu}^{N,\mathrm{red}} \in X^N$ is defined as

$$\boxed{\hat{\varphi}_{\mu}^{N,\mathrm{red}} \coloneqq \operatorname*{arg\,min}_{p \in X^N} \left\| \hat{g}(\mu) - \hat{S}(\mu)p \right\|_X^2,}$$

where

$$\hat{S}(\mu) \coloneqq E^\top + C^\top C \hat{\Lambda}(\mu) V_{\mathrm{ad}} V_{\mathrm{ad}}^\top \in \mathbb{R}^{n \times n}$$

and

$$\hat{g}(\mu) \coloneqq C^\top C \left(\hat{\Psi}_{\mu}^{\mathrm{pr}}(T,0) V_{\mathrm{pr}} V_{\mathrm{pr}}^\top x_{\mu}^0 - x_{\mu}^T \right) \in \mathbb{R}^n.$$

hendrik.kleikamp@uni-muenster.de

Part 3: Fully reduced model (F-ROM)
Computation of the fully reduced solution

The fully reduced solution $\hat{\varphi}_\mu^{N,\text{red}} \in X^N$ is defined as

$$\boxed{\hat{\varphi}_\mu^{N,\text{red}} \coloneqq \operatorname*{arg\,min}_{p \in X^N} \left\| \hat{g}(\mu) - \hat{S}(\mu)p \right\|_X^2,}$$

where

$$\hat{S}(\mu) \coloneqq E^\top + C^\top C \hat{\Lambda}(\mu) V_{\text{ad}} V_{\text{ad}}^\top \in \mathbb{R}^{n \times n}$$

and

$$\hat{g}(\mu) \coloneqq C^\top C \left(\hat{\Psi}_\mu^{\text{pr}}(T, 0) V_{\text{pr}} V_{\text{pr}}^\top x_\mu^0 - x_\mu^T \right) \in \mathbb{R}^n.$$

This least squares problem can be solved as before by orthogonal projection in $\mathbb{R}^n$, i.e.

$$\hat{S}(\mu) \hat{\varphi}_\mu^{N,\text{red}} = P_{\hat{S}(\mu) X^N} \hat{g}(\mu).$$

For an approximate final time adjoint φ^N we define the error estimator

$$\boxed{\eta_\mu^{\text{red}}(\varphi^N) := C_{\text{op}}\left(\|C^\top C\|\,\Delta_\mu^{\text{pr}}[0](T) + \hat{\eta}_\mu(\varphi^N) + \|C^\top C\|\,\Delta_\mu^{\Lambda}[\varphi^N]\right),}$$

where the reduced error estimator is given as

$$\hat{\eta}_\mu(\varphi^N) := \left\| \hat{g}(\mu) - \hat{S}(\mu)\varphi^N \right\|$$

and the Gramian error estimator is given as

$$\Delta_\mu^{\Lambda}[\varphi^N] := C_1(\mu)C_2(\mu)\int_0^T \Delta_\mu^{\text{ad}}[\varphi^N](s)\,ds + C_1(\mu)\int_0^T \left\| E^{-1}R_\mu^{\text{pr}}[\hat{x}_\mu,\hat{u}_\mu](s)\right\| ds.$$

Part 3: Fully reduced model (F-ROM)

Practical remarks

- Fully reduced model can be decomposed offline-online such that online phase is completely independent of n.

Part 3: Fully reduced model (F-ROM)

Practical remarks

- Fully reduced model can be decomposed offline-online such that online phase is completely independent of n.

- A posteriori error estimator can be evaluated in a complexity independent of n as well.

hendrik.kleikamp@uni-muenster.de

Part 3: Fully reduced model (F-ROM)
Practical remarks

- Fully reduced model can be decomposed offline-online such that online phase is completely independent of n.
- A posteriori error estimator can be evaluated in a complexity independent of n as well.
- Several approaches for the construction of X^N and $V_{\text{pr}}, V_{\text{ad}}, W_{\text{pr}}, W_{\text{ad}}$ are possible, see [K./Renelt'24] for details.

Part 3: Fully reduced model (F-ROM)
Practical remarks

- Fully reduced model can be decomposed offline-online such that online phase is completely independent of n.
- A posteriori error estimator can be evaluated in a complexity independent of n as well.
- Several approaches for the construction of X^N and $V_{\text{pr}}, V_{\text{ad}}, W_{\text{pr}}, W_{\text{ad}}$ are possible, see [K./Renelt'24] for details.
- Here, we construct reduced spaces adaptively within a model hierarchy, see below.

Machine learning surrogate (F-ML-ROM)

Learning the map from parameters to reduced coefficients [Hesthaven/Ubbiali'18]

- Observation: Given a basis $\varphi_1, \dots, \varphi_N \in X^N$, the reduced coefficients $\hat{\alpha}_\mu^{N,\text{red}} \in \mathbb{R}^N$ determine $\hat{\varphi}_\mu^{N,\text{red}} \in X^N$ as

$$\hat{\varphi}_\mu^{N,\text{red}} = \sum_{i=1}^{N} \left[\hat{\alpha}_\mu^{N,\text{red}}\right]_i \cdot \varphi_i.$$

Machine learning surrogate (F-ML-ROM)

Learning the map from parameters to reduced coefficients [Hesthaven/Ubbiali'18]

- Observation: Given a basis $\varphi_1, \dots, \varphi_N \in X^N$, the reduced coefficients $\hat{\alpha}_\mu^{N,\mathrm{red}} \in \mathbb{R}^N$ determine $\hat{\varphi}_\mu^{N,\mathrm{red}} \in X^N$ as

$$\hat{\varphi}_\mu^{N,\mathrm{red}} = \sum_{i=1}^{N} \left[\hat{\alpha}_\mu^{N,\mathrm{red}}\right]_i \cdot \varphi_i.$$

- Idea: Learn the map from parameter to reduced coefficients, i.e. approximate

$$\pi_N \colon \mathcal{P} \ni \mu \mapsto \hat{\alpha}_\mu^{N,\mathrm{red}} \in \mathbb{R}^N$$

by a machine learning surrogate $\hat{\pi} \colon \mathcal{P} \to \mathbb{R}^N$.

Machine learning surrogate (F-ML-ROM)

Learning the map from parameters to reduced coefficients [Hesthaven/Ubbiali'18]

- Observation: Given a basis $\varphi_1, \dots, \varphi_N \in X^N$, the reduced coefficients $\hat{\alpha}_\mu^{N,\text{red}} \in \mathbb{R}^N$ determine $\hat{\varphi}_\mu^{N,\text{red}} \in X^N$ as

$$\hat{\varphi}_\mu^{N,\text{red}} = \sum_{i=1}^{N} \left[\hat{\alpha}_\mu^{N,\text{red}}\right]_i \cdot \varphi_i.$$

- Idea: Learn the map from parameter to reduced coefficients, i.e. approximate

$$\pi_N \colon \mathcal{P} \ni \mu \mapsto \hat{\alpha}_\mu^{N,\text{red}} \in \mathbb{R}^N$$

  by a machine learning surrogate $\hat{\pi} \colon \mathcal{P} \to \mathbb{R}^N$.
- Compute machine learning reduced solution as

$$\hat{\varphi}_\mu^{N,\text{red,ml}} = \sum_{i=1}^{N} [\hat{\pi}(\mu)]_i \cdot \varphi_i.$$

Machine learning surrogate (F-ML-ROM)

Learning the map from parameters to reduced coefficients [Hesthaven/Ubbiali'18]

- Observation: Given a basis $\varphi_1, \dots, \varphi_N \in X^N$, the reduced coefficients $\hat{\alpha}_\mu^{N,\text{red}} \in \mathbb{R}^N$ determine $\hat{\varphi}_\mu^{N,\text{red}} \in X^N$ as

$$\hat{\varphi}_\mu^{N,\text{red}} = \sum_{i=1}^{N} \left[\hat{\alpha}_\mu^{N,\text{red}}\right]_i \cdot \varphi_i.$$

- Idea: Learn the map from parameter to reduced coefficients, i.e. approximate

$$\pi_N \colon \mathcal{P} \ni \mu \mapsto \hat{\alpha}_\mu^{N,\text{red}} \in \mathbb{R}^N$$

  by a machine learning surrogate $\hat{\pi} \colon \mathcal{P} \to \mathbb{R}^N$.
- Compute machine learning reduced solution as

$$\hat{\varphi}_\mu^{N,\text{red,ml}} = \sum_{i=1}^{N} [\hat{\pi}(\mu)]_i \cdot \varphi_i.$$

- Solve reduced dynamical systems from the F-ROM to compute the approximate control using $\hat{\varphi}_\mu^{N,\text{red,ml}}$ as final time adjoint.

References

S. Rave, J. Saak,
A Non-stationary Thermal-Block Benchmark Model for Parametric Model Order Reduction,
International Series of Numerical Mathematics, 349–356 (2021), DOI: 10.1007/978-3-030-72983-7_16

M. Lazar, E. Zuazua,
Greedy controllability of finite dimensional linear systems,
Automatica, Vol. 74, 327–340 (2016), DOI: 10.1016/j.automatica.2016.08.010

G. Fabrini, L. Iapichino, S. Volkwein,
Reduced-Order Greedy Controllability of Finite Dimensional Linear Systems,
Proceedings of the 9th Vienna International Conference on Mathematical Modelling, 51:2, 296–301 (2018),
DOI:10.1016/j.ifacol.2018.03.051

B. Haasdonk, M. Ohlberger,
Efficient Reduced Models and A-Posteriori Error Estimation for Parametrized Dynamical Systems by Offline/Online Decomposition,
Mathematical and Computer Modelling of Dynamical Systems, 17:2, 145–161 (2011), DOI: 10.1080/13873954.2010.514703

J.S. Hesthaven, S. Ubbiali,
Non-intrusive reduced order modeling of nonlinear problems using neural networks,
Journal of Computational Physics, Vol. 363, 55–78 (2018), DOI: 10.1016/j.jcp.2018.02.037

G. Santin, B. Haasdonk,
Kernel Methods for Surrogate Modeling,
Model Order Reduction, De Gruyter (2021), DOI: 10.1515/9783110498967-009