Universität Münster

Be greedy and learn: efficient and certified algorithms for parametrized optimal control problems

ALGORITMY 2024 – Minisymposium on “New Trends in Model Order Reduction and Learning”

Hendrik Kleikamp (Münster), Martin Lazar (Dubrovnik), Cesare Molinari (Genova)

March 19, 2024

living.knowledge

MM Mathematics Münster Cluster of Excellence

Parametrized linear time-invariant control system

$$\dot{x}_\mu(t) = A_\mu x_\mu(t) + B_\mu u_\mu(t), \qquad t \in [0, T],$$
$$x_\mu(0) = x_\mu^0$$

Parameter dependent dynamical system and cost functional

Parametrized linear time-invariant control system

$$\begin{aligned}\dot{x}_\mu(t) &= A_\mu x_\mu(t) + B_\mu u_\mu(t), \qquad t \in [0,T],\\ x_\mu(0) &= x_\mu^0\end{aligned}$$

Cost functional $\mathcal{J}_\mu \colon G \to \mathbb{R}_+$

$$\mathcal{J}_\mu(u) \coloneqq \frac{1}{2}\left[\underbrace{\left\langle x_\mu(T) - x_\mu^T, M\left(x_\mu(T) - x_\mu^T\right)\right\rangle}_{\text{deviation from the target state } x_\mu^T \in X} + \underbrace{\int_0^T \langle u(t), Ru(t)\rangle \, dt}_{\text{energy of the control}}\right]$$

hendrik.kleikamp@uni-muenster.de

Parametrized optimal control problem

Given a parameter $\mu \in \mathcal{P}$, solve the following optimization problem:

$$\min_{u \in G} \mathcal{J}_\mu(u), \qquad \text{s.t. } \dot{x}_\mu(t) = A_\mu x_\mu(t) + B_\mu u(t) \text{ for } t \in [0, T], \quad x_\mu(0) = x_\mu^0.$$

Optimality system for the optimal control problem

Theorem 1 (Optimality system)

Let $\mu \in \mathcal{P}$ be a parameter, $u_\mu^* \in G$ an optimal control and $x_\mu^* \in H$ the associated state trajectory. Then there exists an adjoint solution $\varphi_\mu^* \in H$, such that the problem

$$\begin{aligned} \dot{x}_\mu(t) &= A_\mu x_\mu(t) + B_\mu u_\mu(t), \\ -\dot{\varphi}_\mu(t) &= A_\mu^* \varphi_\mu(t), \\ u_\mu(t) &= -R^{-1} B_\mu^* \varphi_\mu(t), \end{aligned}$$

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

$$x_\mu(0) = x_\mu^0, \qquad \varphi_\mu(T) = M\left(x_\mu(T) - x_\mu^T\right),$$

is solved by $x_\mu = x_\mu^*$, $\varphi_\mu = \varphi_\mu^*$ and $u_\mu = u_\mu^*$.

Optimality system for the optimal control problem

Theorem 1 (Optimality system)

Let $\mu \in \mathcal{P}$ be a parameter, $u_\mu^* \in G$ an optimal control and $x_\mu^* \in H$ the associated state trajectory. Then there exists an adjoint solution $\varphi_\mu^* \in H$, such that the problem

$$\begin{aligned} \dot{x}_\mu(t) &= A_\mu x_\mu(t) + B_\mu u_\mu(t), \\ -\dot{\varphi}_\mu(t) &= A_\mu^* \varphi_\mu(t), \\ u_\mu(t) &= -R^{-1} B_\mu^* \varphi_\mu(t), \end{aligned}$$

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

$$x_\mu(0) = x_\mu^0, \qquad \varphi_\mu(T) = M\left(x_\mu(T) - x_\mu^T\right),$$

is solved by $x_\mu = x_\mu^*$, $\varphi_\mu = \varphi_\mu^*$ and $u_\mu = u_\mu^*$.

State, control and adjoint are uniquely determined by optimal final time adjoint $\varphi_\mu^*(T)$!

Weighted controllability Gramian

Define the weighted controllability Gramian $\Lambda_\mu^R \in \mathcal{L}(X, X)$ as

$$\Lambda_\mu^R := \int_0^T e^{A_\mu(T-s)} B_\mu R^{-1} B_\mu^* e^{A_\mu^*(T-s)} \, ds.$$

Weighted controllability Gramian

Define the weighted controllability Gramian $\Lambda_\mu^R \in \mathcal{L}(X, X)$ as

$$\Lambda_\mu^R := \int_0^T e^{A_\mu(T-s)} B_\mu R^{-1} B_\mu^* e^{A_\mu^*(T-s)}\, ds.$$

Then we have

$$x_\mu^*(T) = e^{A_\mu T} x_\mu^0 - \Lambda_\mu^R \varphi_\mu^*(T).$$

Weighted controllability Gramian

Define the weighted controllability Gramian $\Lambda_\mu^R \in \mathcal{L}(X, X)$ as

$$\Lambda_\mu^R := \int_0^T e^{A_\mu(T-s)} B_\mu R^{-1} B_\mu^* e^{A_\mu^*(T-s)} \, ds.$$

Then we have

$$x_\mu^*(T) = e^{A_\mu T} x_\mu^0 - \Lambda_\mu^R \varphi_\mu^*(T).$$

Combining this with the terminal condition

$$\varphi_\mu^*(T) = M\left(x_\mu^*(T) - x_\mu^T\right)$$

from the optimality system gives the following linear system for $\varphi_\mu^*(T)$.

Linear system for the optimal final time adjoint

Lemma 1 (Linear system)

Let $\varphi_\mu^*(T)$ denote the optimal adjoint state at time T that determines the solution of the optimality system. Then it holds

$$\left(I + M\Lambda_\mu^R\right) \varphi_\mu^*(T) = M\left(e^{A_\mu T}x_\mu^0 - x_\mu^T\right),$$

where $I \in \mathcal{L}(X, X)$ denotes the identity.

hendrik.kleikamp@uni-muenster.de

Lemma 1 (Linear system)

Let $\varphi_\mu^*(T)$ denote the optimal adjoint state at time T that determines the solution of the optimality system. Then it holds

$$\left(I + M\Lambda_\mu^R\right) \varphi_\mu^*(T) = M \left(e^{A_\mu T} x_\mu^0 - x_\mu^T\right),$$

where $I \in \mathcal{L}(X, X)$ denotes the identity.

$\Longrightarrow$ We have to solve a linear system with $I + M\Lambda_\mu^R$ for every new parameter μ!

Assumption: Let $M\Lambda_\mu^R$ be positive-semidefinite for all parameters $\mu \in \mathcal{P}$.

hendrik.kleikamp@uni-muenster.de

Main ideas of the reduced order model

- Approximate the optimal final time adjoint state $\varphi_\mu^*(T)$ in a reduced space.

hendrik.kleikamp@uni-muenster.de

Main ideas of the reduced order model

- Approximate the optimal final time adjoint state $\varphi_\mu^*(T)$ in a reduced space.
- Define an error estimator as the norm of the residual of the linear system.

Main ideas of the reduced order model

- Approximate the optimal final time adjoint state $\varphi_\mu^*(T)$ in a reduced space.
- Define an error estimator as the norm of the residual of the linear system.
- Apply a greedy algorithm to construct a reduced basis for the optimal final time adjoint states (by means of an efficient error estimator for the reduced space).

Main ideas of the reduced order model

- Approximate the optimal final time adjoint state $\varphi_\mu^*(T)$ in a reduced space.
- Define an error estimator as the norm of the residual of the linear system.
- Apply a greedy algorithm to construct a reduced basis for the optimal final time adjoint states (by means of an efficient error estimator for the reduced space).
- Compute reduced solution by projecting the right hand side of the linear system onto the subspace of states reachable from the reduced space of final time adjoints.

Main ideas of the reduced order model

- Approximate the optimal final time adjoint state $\varphi_\mu^*(T)$ in a reduced space.
- Define an error estimator as the norm of the residual of the linear system.
- Apply a greedy algorithm to construct a reduced basis for the optimal final time adjoint states (by means of an efficient error estimator for the reduced space).
- Compute reduced solution by projecting the right hand side of the linear system onto the subspace of states reachable from the reduced space of final time adjoints.
- Later: Accelerate online phase using machine learning with error certification.

Error estimation using the residual

Given an approximate final time adjoint p, consider the residual of the linear system as error estimator:

$$\eta_\mu(p) \coloneqq \left\| M\left(e^{A_\mu T}x_\mu^0 - x_\mu^T\right) - (I + M\Lambda_\mu^R)p \right\|_X .$$

Error estimation using the residual

Given an approximate final time adjoint p, consider the residual of the linear system as error estimator:

$$\eta_\mu(p) := \left\| M \left(e^{A_\mu T} x_\mu^0 - x_\mu^T\right) - (I + M\Lambda_\mu^R)p \right\|_X .$$

Theorem 2 (Error estimator for an approximate final time adjoint)

Then it holds:

$$\left\| \varphi_\mu^*(T) - p \right\|_X \leqslant \eta_\mu(p) \leqslant \left\| I + M\Lambda_\mu^R \right\|_{\mathcal{L}(X,X)} \left\| \varphi_\mu^*(T) - p \right\|_X .$$

hendrik.kleikamp@uni-muenster.de

Approximating the solution manifold by linear subspaces

- Manifold $\mathcal{M} \coloneqq \{\varphi_\mu^*(T) : \mu \in \mathcal{P}\} \subset X$
- Approximation tolerance $\varepsilon > 0$

Goal of the greedy algorithm

Find a reduced space $X^N \subset X$ of (small) dimension N such that

$$\mathrm{dist}(X^N, \mathcal{M}) \leqslant \varepsilon.$$

Greedy procedure for the optimal control problem I

Universität Münster MM

- Compute reduced approximation $X^N \ni \tilde{\varphi}_\mu^N \approx \varphi_\mu^*(T) \in X$ as the least squares solution to the linear system in the space X^N, i.e.

$$\tilde{\varphi}_\mu^N = \arg\min_{\varphi\in X^N} \eta_\mu(\varphi) = \arg\min_{\varphi\in X^N} \left\| M\left(e^{A_\mu T}x_\mu^0 - x_\mu^T\right) - (I + M\Lambda_\mu^R)\varphi \right\|_X^2 .$$

Greedy procedure for the optimal control problem I

- Compute reduced approximation $X^N \ni \tilde{\varphi}_\mu^N \approx \varphi_\mu^*(T) \in X$ as the least squares solution to the linear system in the space X^N, i.e.

$$\tilde{\varphi}_\mu^N = \operatorname*{arg\,min}_{\varphi \in X^N} \eta_\mu(\varphi) = \operatorname*{arg\,min}_{\varphi \in X^N} \left\| M\left(e^{A_\mu T} x_\mu^0 - x_\mu^T\right) - (I + M\Lambda_\mu^R)\varphi \right\|_X^2 .$$

- Thus, given a reduced space $X^N = \operatorname{span}\{\varphi_1, \dots, \varphi_N\} \subset X$ and a parameter $\mu \in \mathcal{P}$, minimize the residual by projecting onto $Y_\mu^N = (I + M\Lambda_\mu^R)X^N$:

$$P_{Y_\mu^N}\left(M(e^{A_\mu T} x_\mu^0 - x_\mu^T)\right) = \sum_{i=1}^N \alpha_i^\mu x_i^\mu,$$

where $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i$, i.e. $Y_\mu^N = \operatorname{span}\{x_1^\mu, \dots, x_N^\mu\}$.

Greedy procedure for the optimal control problem I

- Compute reduced approximation $X^N \ni \tilde{\varphi}_\mu^N \approx \varphi_\mu^*(T) \in X$ as the least squares solution to the linear system in the space X^N, i.e.

$$\tilde{\varphi}_\mu^N = \underset{\varphi \in X^N}{\arg\min}\, \eta_\mu(\varphi) = \underset{\varphi \in X^N}{\arg\min} \left\| M\left(e^{A_\mu T} x_\mu^0 - x_\mu^T\right) - (I + M\Lambda_\mu^R)\varphi \right\|_X^2.$$

- Thus, given a reduced space $X^N = \operatorname{span}\{\varphi_1, \dots, \varphi_N\} \subset X$ and a parameter $\mu \in \mathcal{P}$, minimize the residual by projecting onto $Y_\mu^N = (I + M\Lambda_\mu^R)X^N$:

$$P_{Y_\mu^N}\left(M(e^{A_\mu T} x_\mu^0 - x_\mu^T)\right) = \sum_{i=1}^N \alpha_i^\mu x_i^\mu,$$

  where $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i$, i.e. $Y_\mu^N = \operatorname{span}\{x_1^\mu, \dots, x_N^\mu\}$.
- Set for the reduced approximation $\tilde{\varphi}_\mu^N$:

$$\tilde{\varphi}_\mu^N = \sum_{i=1}^N \alpha_i^\mu \varphi_i.$$

- We therefore have

$$(I + M\Lambda_\mu^R)\tilde{\varphi}_\mu^N = P_{Y_\mu^N}\left(M(e^{A_\mu T}x_\mu^0 - x_\mu^T)\right)$$

and

$$\eta_\mu(\tilde{\varphi}_\mu^N) = \left\| M(e^{A_\mu T}x_\mu^0 - x_\mu^T) - (I + M\Lambda_\mu^R)\tilde{\varphi}_\mu^N \right\| = \operatorname{dist}\left(Y_\mu^N, \{M(e^{A_\mu T}x_\mu^0 - x_\mu^T)\}\right).$$

- We therefore have

$$(I+M\Lambda_\mu^R)\tilde{\varphi}_\mu^N = P_{Y_\mu^N}\left(M(e^{A_\mu T}x_\mu^0 - x_\mu^T)\right)$$

and

$$\eta_\mu(\tilde{\varphi}_\mu^N) = \left\| M(e^{A_\mu T}x_\mu^0 - x_\mu^T) - (I+M\Lambda_\mu^R)\tilde{\varphi}_\mu^N \right\| = \operatorname{dist}\left(Y_\mu^N, \{M(e^{A_\mu T}x_\mu^0 - x_\mu^T)\}\right).$$

Theorem 3 (Error estimator for a reduced space)

Then it holds:

$$\operatorname{dist}(X^N, \{\varphi_\mu^*(T)\}) \quad \leqslant \quad \eta_\mu(\tilde{\varphi}_\mu^N) \quad \leqslant \quad \left\| I + M\Lambda_\mu^R \right\| \cdot \operatorname{dist}(X^N, \{\varphi_\mu^*(T)\}).$$

Universität Münster MM

Theorem 3 (Weak greedy algorithm and approximation error)

The greedy procedure presented above is a weak greedy algorithm with constant

$$\gamma \coloneqq \frac{1}{C_{\varphi^*} + C_\Lambda} \leqslant 1,$$

where C_{φ^*} is the Lipschitz constant of the mapping $\mu \mapsto \varphi_\mu^*(T)$ and $C_\Lambda \coloneqq \sup_{\mu \in \mathcal{P}} \left\| I + M\Lambda_\mu^R \right\|$. It further holds for all $\mu \in \mathcal{P}$ that

$$\operatorname{dist}(X^N, \{\varphi_\mu^*(T)\}) \leqslant \varepsilon.$$

Machine learning of reduced coefficients

- Costly part in the online phase of the reduced order model:

  Computation of $$x_i^{\mu} = (I + M\Lambda_{\mu}^{R})\varphi_i \qquad \text{for } i = 1, \dots, N.$$

  This requires solving the primal and adjoint systems N times!

Machine learning of reduced coefficients

- Costly part in the online phase of the reduced order model:

  Computation of $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i \qquad \text{for } i = 1, \dots, N.$

  This requires solving the primal and adjoint systems N times!
- Instead: Learn the map from parameters to coefficients, i.e. approximate the map

$$\mu \quad \mapsto \quad \pi_N(\mu) \coloneqq [\alpha_i^\mu]_{i=1}^N$$

  by a machine learning surrogate $\hat{\pi}_N : \mathcal{P} \to \mathbb{R}^N$, see also [Hesthaven, Ubbiali'18].

Machine learning of reduced coefficients

- Costly part in the online phase of the reduced order model:

  Computation of $\quad x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i \quad$ for $i = 1, \dots, N$.

  This requires solving the primal and adjoint systems N times!

- Instead: Learn the map from parameters to coefficients, i.e. approximate the map

$$\mu \quad \mapsto \quad \pi_N(\mu) \coloneqq [\alpha_i^\mu]_{i=1}^N$$

  by a machine learning surrogate $\hat{\pi}_N \colon \mathcal{P} \to \mathbb{R}^N$, see also [Hesthaven, Ubbiali'18].

- Approximate the final time adjoint as

$$\hat{\varphi}_\mu^N = \sum_{i=1}^N [\hat{\pi}_N(\mu)]_i \varphi_i.$$

Error estimates for machine learning approximation

- A priori bound (assuming that the reduced basis is orthonormal):

$$\left\| \varphi_\mu^*(T) - \hat{\varphi}_\mu^N \right\| \leqslant C_\Lambda \underbrace{\varepsilon}_{\substack{\text{greedy}\\\text{tolerance}}} + \underbrace{\left\| \pi_N(\mu) - \hat{\pi}_N(\mu) \right\|}_{\substack{\text{approximation error}\\\text{of machine learning}}}.$$

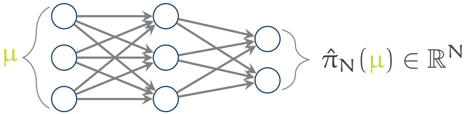
- A posteriori bound:

$$\left\| \varphi_\mu^*(T) - \hat{\varphi}_\mu^N \right\| \leqslant \eta_\mu(\hat{\varphi}_\mu^N) \leqslant \left\| I + M\Lambda_\mu \right\| \left\| \varphi_\mu^*(T) - \hat{\varphi}_\mu^N \right\|.$$

Machine learning approaches

Various machine learning methods can be applied here, we considered:

- Deep neural networks (DNN), see for instance [Petersen, Voigtlaender'18].

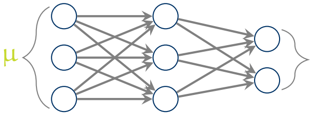


hendrik.kleikamp@uni-muenster.de

Machine learning approaches

Various machine learning methods can be applied here, we considered:

- Deep neural networks (DNN), see for instance [Petersen, Voigtlaender'18].



$\hat{\pi}_N(\mu) \in \mathbb{R}^N$

- Kernel methods (VKOGA), see for instance [Santin, Haasdonk'21].

$$\hat{\pi}_N(\mu) = \sum_{i \in \Xi} \alpha_i k_N(\mu, x_i)$$

kernel

centers

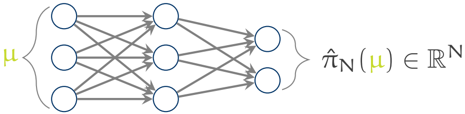
coefficients

subset of selected centers

Machine learning approaches

Various machine learning methods can be applied here, we considered:

- Deep neural networks (DNN), see for instance [Petersen, Voigtlaender'18].



- Kernel methods (VKOGA), see for instance [Santin, Haasdonk'21].

$$\hat{\pi}_N(\mu) = \sum_{i \in \Xi} \alpha_i k_N(\mu, x_i)$$

- Gaussian process regression (GPR), see for instance [Rasmussen, Williams'06].

$$\hat{\pi}_N(\mu) = \mathbb{E}_y[P(y|\mu, X, Y)]$$

input data (X)

output data (Y)

Numerical example: Parametrized heat equation

- Problem definition:

$$\begin{aligned} \partial_t v_\mu(t,y) - \mu_1 \Delta v_\mu(t,y) &= 0 && \text{for } t \in [0,T], y \in \Omega, \\ v_\mu(t,0) &= u_{\mu,1}(t) && \text{for } t \in [0,T], \\ v_\mu(t,1) &= u_{\mu,2}(t) && \text{for } t \in [0,T], \\ v_\mu(0,y) = v_\mu^0(y) &= \sin(\pi y) && \text{for } y \in \Omega. \end{aligned}$$

- Target state:

$$v_\mu^T(y) = \mu_2 y$$

- Details: $\Omega = [0,1]$, $T = 0.1$, $\mathcal{P} = [1,2] \times [0.5, 1.5]$

Numerical example: Details

- Spatial discretization: Second order central finite difference scheme with $n = 100$ grid points

$$A_\mu = \frac{\mu_1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{n\times n} \quad \text{and} \quad B_\mu = \frac{\mu_1}{h^2}\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{n\times 2}$$

- Temporal discretization: Crank-Nicolson scheme (implicit) with $n_t = 300$ time steps
- Weighting matrices:

$$M = I \in \mathbb{R}^{n\times n} \quad \text{and} \quad R = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.25 \end{bmatrix}$$

Numerical example: Parametrized heat equation

Optimal, initial and target state

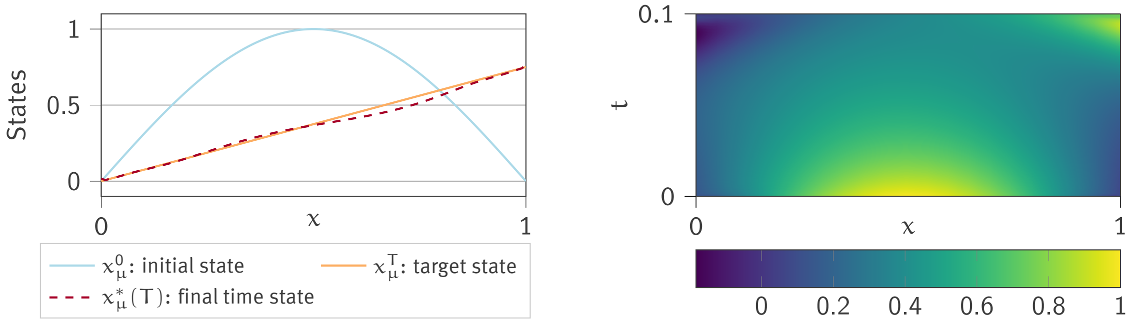


Figure: Optimal state x_μ^* in a space-time plot (left) and initial x_μ^0, final $x_\mu^*(T)$ and target x_μ^T states (right) for the parameter $\mu = (1.5, 0.75)$ in the heat equation example.

Numerical example: Parametrized heat equation
Optimal control and adjoint state

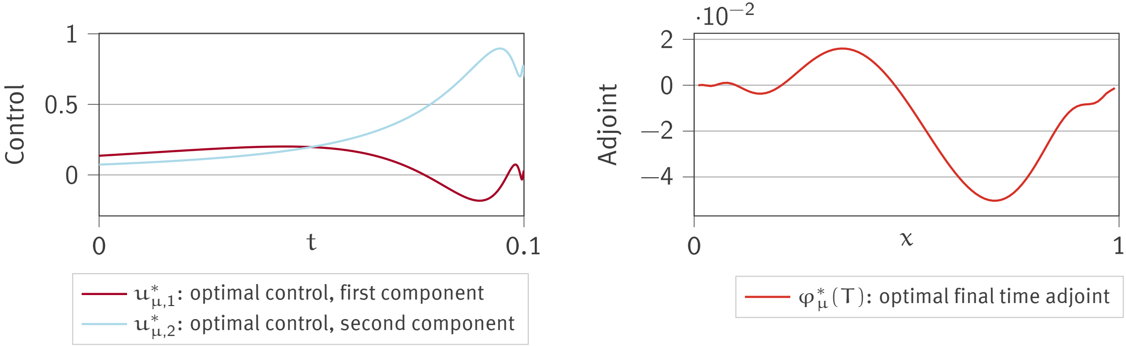
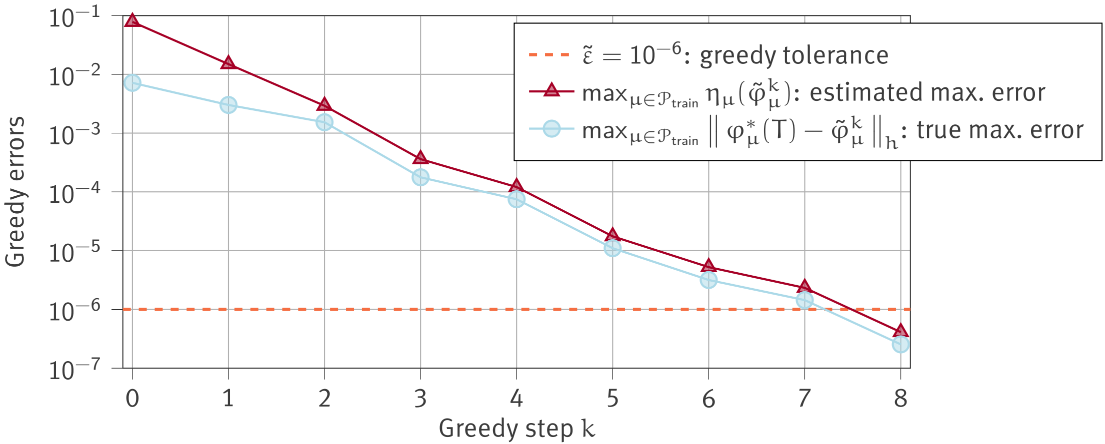


Figure: Optimal control u^*_μ (left) and optimal final time adjoint $\varphi^*_\mu(T)$ (right) for the parameter $\mu = (1.5, 0.75)$ in the heat equation example.

Numerical example: Parametrized heat equation

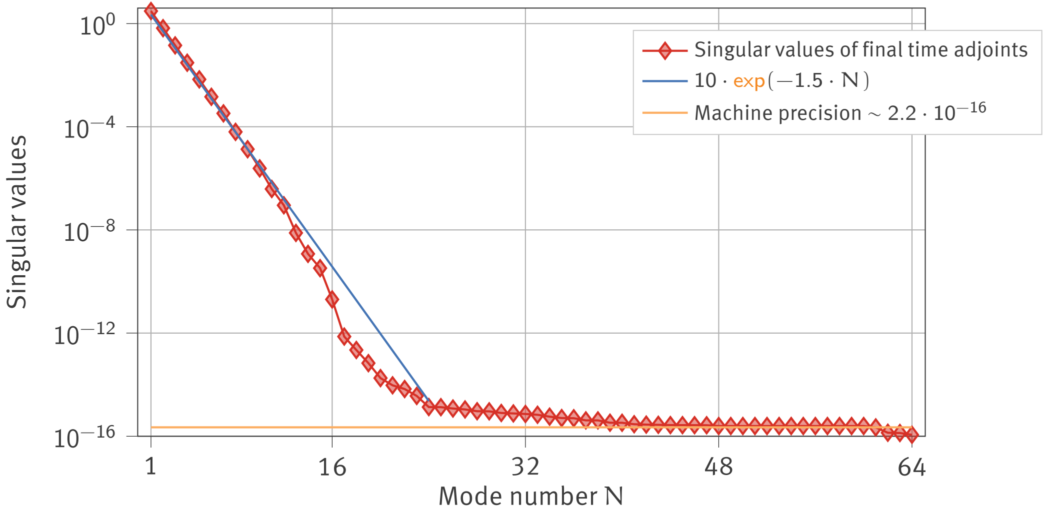
Greedy algorithm

Results of running the greedy algorithm with 64 uniformly distributed training parameters:



Numerical example: Parametrized heat equation

Singular value decay of optimal final time adjoints



Numerical example: Parametrized heat equation
Machine learning approaches

Results on a set of 100 randomly drawn test parameters:

Method	Avg. error adjoint	Avg. error control	Avg. runtime (s)	Avg. speedup
FOM	—	—	6.2760	—
RB-ROM	$5.3 \cdot 10^{-8}$	$5.4 \cdot 10^{-9}$	2.6526	2.37
DNN-ROM	$5.8 \cdot 10^{-6}$	$2.0 \cdot 10^{-6}$	0.1623	40.33
VKOGA-ROM	$1.8 \cdot 10^{-5}$	$6.9 \cdot 10^{-6}$	0.1580	41.03
GPR-ROM	$2.2 \cdot 10^{-6}$	$7.6 \cdot 10^{-7}$	0.1572	41.40

Adaptive model hierarchy [Haasdonk et al'23, Kleikamp'24]

Main ideas

- Start with an empty RB-ROM, that is $X^N = \{0\}$, and empty ML-ROM.

Adaptive model hierarchy [Haasdonk et al'23, Kleikamp'24]
Main ideas

- Start with an empty RB-ROM, that is $X^N = \{0\}$, and empty ML-ROM.
- Query cheapest models first and evaluate the error estimator.

hendrik.kleikamp@uni-muenster.de

Adaptive model hierarchy [Haasdonk et al'23, Kleikamp'24]
Main ideas

- Start with an empty RB-ROM, that is $X^N = \{0\}$, and empty ML-ROM.
- Query cheapest models first and evaluate the error estimator.
- Return result if it is accurate enough, otherwise fall back to more costly models.

Adaptive model hierarchy [Haasdonk et al'23, Kleikamp'24]
Main ideas

- Start with an empty RB-ROM, that is $X^N = \{0\}$, and empty ML-ROM.
- Query cheapest models first and evaluate the error estimator.
- Return result if it is accurate enough, otherwise fall back to more costly models.
- Build RB-ROM using data from FOM calls.

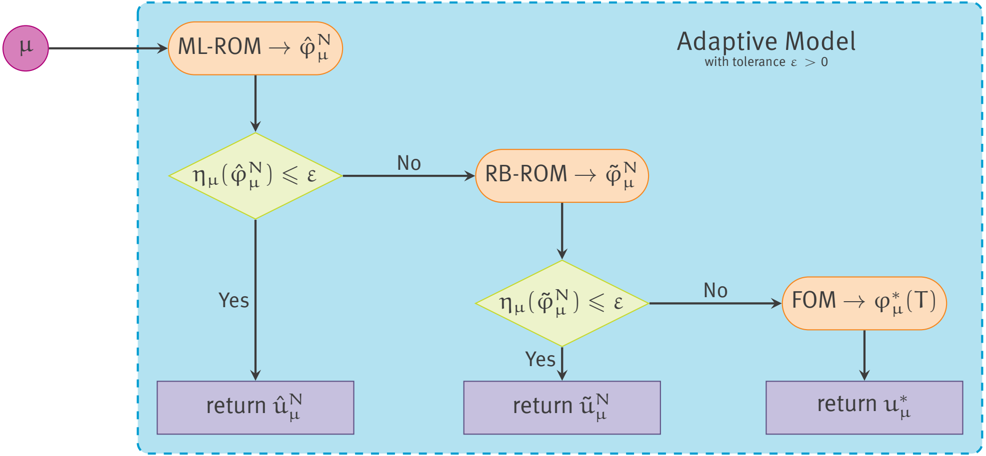
hendrik.kleikamp@uni-muenster.de

Adaptive model hierarchy [Haasdonk et al'23, Kleikamp'24]
Main ideas

- Start with an empty RB-ROM, that is $X^N = \{0\}$, and empty ML-ROM.
- Query cheapest models first and evaluate the error estimator.
- Return result if it is accurate enough, otherwise fall back to more costly models.
- Build RB-ROM using data from FOM calls.
- Train ML-ROM with data from RB-ROM.

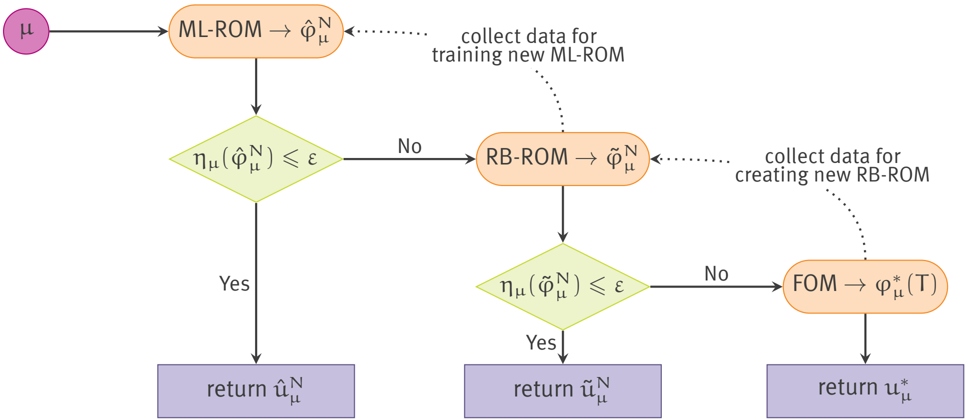
Adaptive model hierarchy [Haasdonk et al'23, Kleikamp'24]

Sketch of evaluating the hierarchy for a parameter



Adaptive model hierarchy [Haasdonk et al'23, Kleikamp'24]

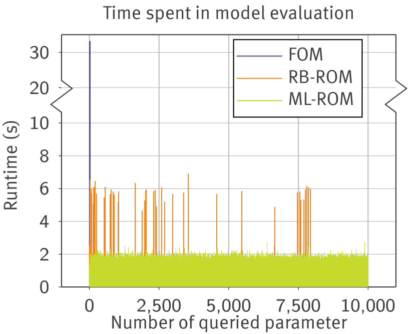
Sketch of evaluating the hierarchy for a parameter

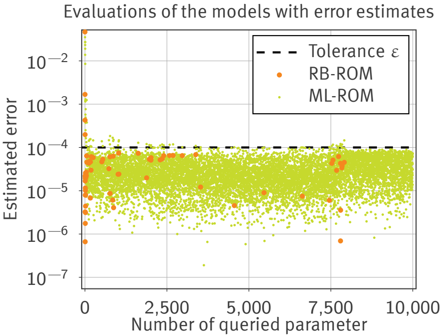


hendrik.kleikamp@uni-muenster.de

Numerical example: Adaptive model hierarchy

Parametrized heat equation (tolerance $\varepsilon = 10^{-4}$)





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	—	112.24	28.06
RB-ROM	65	69	299.26	4.60
ML-ROM	9,931	10,000	16,655.78	1.68

hendrik.kleikamp@uni-muenster.de

Future research

- Advection-diffusion-reaction equations in higher space dimensions as test cases.

hendrik.kleikamp@uni-muenster.de

Future research

- Advection-diffusion-reaction equations in higher space dimensions as test cases.
- Incorporation of theoretical approximation results for the machine learning methods.

Future research

- Advection-diffusion-reaction equations in higher space dimensions as test cases.
- Incorporation of theoretical approximation results for the machine learning methods.
- Additional speedup by applying model order reduction to the parametrized control system, i.e. approximation of the state and adjoint dynamics, see [Haasdonk, Ohlberger'11] and [Fabrini, Iapichino, Volkwein'18].

hendrik.kleikamp@uni-muenster.de

Future research

- Advection-diffusion-reaction equations in higher space dimensions as test cases.
- Incorporation of theoretical approximation results for the machine learning methods.
- Additional speedup by applying model order reduction to the parametrized control system, i.e. approximation of the state and adjoint dynamics, see [Haasdonk, Ohlberger'11] and [Fabrini, Iapichino, Volkwein'18].
- Extension to other classes of optimal control problems, such as linear time-varying systems or problems with constraints on the control.

Universität Münster

Thank you for your attention!

For more details, see:

H. Kleikamp, M. Lazar, C. Molinari.
Be greedy and learn: efficient and certified algorithms for parametrized optimal control problems, (2023).
https://arxiv.org/abs/2307.15590

H. Kleikamp.
Application of an adaptive model hierarchy to parametrized optimal control problems, (2024).
https://arxiv.org/abs/2402.10708

The source code for the papers is available open source:

- https://github.com/HenKlei/ML-OPT-CONTROL
- https://github.com/HenKlei/ADAPTIVE-ML-OPT-CONTROL

References I

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

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

P. Petersen, F. Voigtlaender,
Optimal approximation of piecewise smooth functions using deep ReLU neural networks,
Neural Networks, Vol. 108, 296-330 (2018), DOI: 10.1016/j.neunet.2018.08.019

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

References II

C. Rasmussen, C. Williams,
Gaussian processes for machine learning,
Adaptive computation and machine learning, MIT Press (2006), DOI: 10.7551/mitpress/3206.001.0001

B. Haasdonk, H. Kleikamp, M. Ohlberger, F. Schindler, T. Wenzel,
A New Certified Hierarchical and Adaptive RB-ML-ROM Surrogate Model for Parametrized PDEs,
SIAM Journal on Scientific Computing, Vol. 45, 3 (2023), DOI: 10.1137/22M1493318

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

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

hendrik.kleikamp@uni-muenster.de