Universität Münster

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

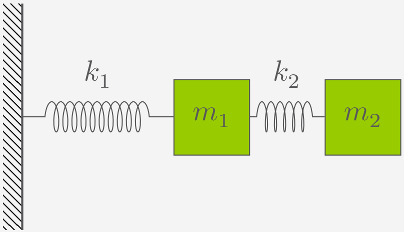
Hendrik Kleikamp, Martin Lazar (Dubrovnik), Cesare Molinari (Genova)

October 18, 2023

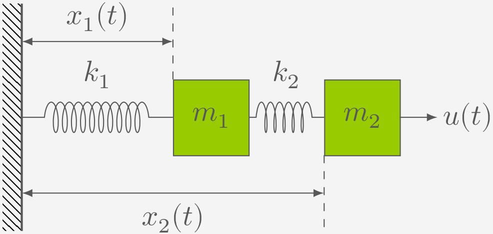
living.knowledge

MM Mathematics Münster Cluster of Excellence

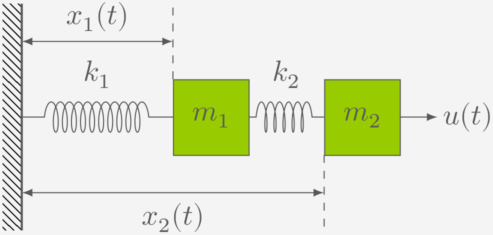
Optimal control of parametrized linear control systems



Optimal control of parametrized linear control systems



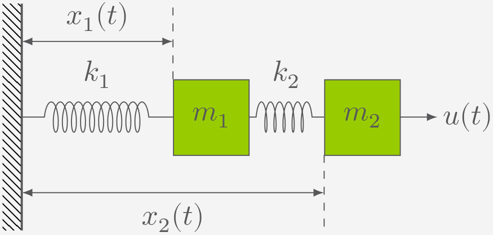
Optimal control of parametrized linear control systems



Goal: Given a parameter value,

- try to steer the system state close to a prescribed target state
- without using too much control energy.

Optimal control of parametrized linear control systems



Goal: Given a parameter value,

- try to steer the system state close to a prescribed target state
- without using too much control energy.

Issue: We would like to solve this problem for many different values of the parameter!

Parametrized linear control systems

General setting:

- State space X (Hilbert space)
- Control space U (Hilbert space)
- Parameter set $\mathcal{P}$ (compact subset of some Banach space)
- Final time $T > 0$

- $H := C^1([0,T];X)$
- $G := C^0([0,T];U)$

Parametrized linear control systems

General setting:

- State space X (Hilbert space)
- Control space U (Hilbert space)
- Parameter set $\mathcal{P}$ (compact subset of some Banach space)
- Final time $T > 0$

- $H := C^1([0,T];X)$
- $G := C^0([0,T];U)$

Parameter dependent quantities ($\mu \in \mathcal{P}$):

- State operator $A_\mu \in \mathcal{L}(X,X)$
- Control operator $B_\mu \in \mathcal{L}(U,X)$
- Initial state $x_\mu^0 \in X$

Parametrized linear control system (LTI)

derivative of the state

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

state — control

Parameter dependent cost functional

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

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

Parameter dependent cost functional

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

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

Assumptions on weighting operators

- $M \in \mathcal{L}(X,X)$ is self-adjoint and positive-semidefinite
- $R \in \mathcal{L}(U,U)$ is self-adjoint and strictly positive-definite, i.e. $R \geq \alpha I$ for some $\alpha > 0$

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, $x_\mu^* \in H$ 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 $u_\mu = u_\mu^*$.

Solving the optimality system

Optimality system

$$\begin{aligned} \dot{x}_\mu^*(t) &= A_\mu x_\mu^*(t) + B_\mu u_\mu^*(t), \\ -\dot{\varphi}_\mu^*(t) &= A_\mu^* \varphi_\mu^*(t), \\ u_\mu^*(t) &= -R^{-1} B_\mu^* \varphi_\mu^*(t), \\ x_\mu^*(0) &= x_\mu^0 \end{aligned}$$

$$\begin{cases} \varphi_\mu^*(t) = e^{A_\mu^*(T-t)} \varphi_\mu^*(T), \\ u_\mu^*(t) = -R^{-1} B_\mu^* \varphi_\mu^*(t), \\ x_\mu^*(t) = \underbrace{e^{A_\mu t} x_\mu^0}_{\text{free dynamics}} + \underbrace{\int_0^t e^{A_\mu (t-s)} B_\mu u_\mu^*(s) \,\mathrm{d}s}_{\text{contribution by the control}} \end{cases}$$

Solving the optimality system

Optimality system

$$
\begin{aligned}
\dot{x}_\mu^*(t) &= A_\mu x_\mu^*(t) + B_\mu u_\mu^*(t),\\
-\dot{\varphi}_\mu^*(t) &= A_\mu^* \varphi_\mu^*(t),\\
u_\mu^*(t) &= -R^{-1} B_\mu^* \varphi_\mu^*(t),\\
x_\mu^*(0) &= x_\mu^0
\end{aligned}
$$

$$
\begin{cases}
\varphi_\mu^*(t) = e^{A_\mu^*(T-t)} \varphi_\mu^*(T),\\
u_\mu^*(t) = -R^{-1} B_\mu^* e^{A_\mu^*(T-t)} \varphi_\mu^*(T),\\
x_\mu^*(t) = e^{A_\mu t} x_\mu^0 - \int_0^t e^{A_\mu(t-s)} B_\mu R^{-1} B_\mu^* e^{A_\mu^*(T-s)} \varphi_\mu^*(T) \,\mathrm{d}s
\end{cases}
$$

State, control and adjoint already 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

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

Weighted controllability Gramian

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

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

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

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

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

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

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.

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

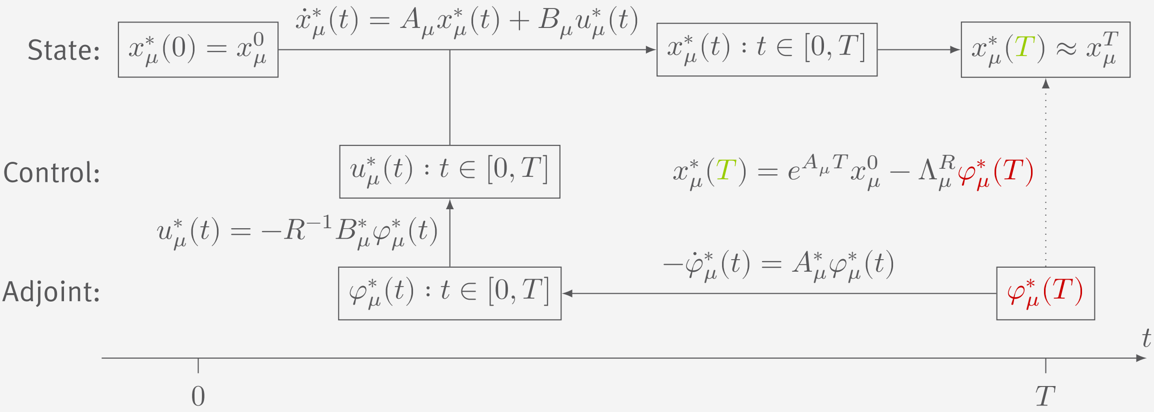
$$\left(I + M\Lambda_\mu^R\right) \varphi_\mu^*(T) = M\left(e^{A_\mu T} x_\mu^0 - x_\mu^T\right),$$

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

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

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

Visualization



Main ideas of the reduced order model

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

Main ideas of the reduced order model

- Approximate the optimal final time adjoint state $\varphi_\mu^*(T)$.
- 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 $\varphi_\mu^*(T)$.
- 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 $\varphi_\mu^*(T)$.
- 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 $\varphi_\mu^*(T)$.
- 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:

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

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:

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

Approximating the solution manifold by linear subspaces

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

Goal of the greedy algorithm

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

$$\operatorname{dist}(X^N, \mathcal{M}) \leq \varepsilon.$$

General (weak) greedy algorithm

Input: Manifold $\mathcal{M}$, tolerance $\varepsilon > 0$, greedy constant $\gamma \in (0, 1]$

Output: Reduced basis $\Phi^N \subset X$, reduced space $X^N = \operatorname{span}(\Phi^N) \subset X$

$N \leftarrow 0, \quad \Phi^N = \emptyset, \quad X^0 = \{0\} \subset X$

while $\operatorname{dist}(X^N, \mathcal{M}) > \varepsilon$ **do**

choose next element $x_{N+1} \in \mathcal{M}$ such that it holds

$$\operatorname{dist}(X^N, \{x_{N+1}\}) \geq \gamma \cdot \operatorname{dist}(X^N, \mathcal{M})$$

$\Phi^{N+1} \leftarrow \Phi^N \cup \{x_{N+1}\}$

$X^{N+1} \leftarrow \operatorname{span}(\Phi^{N+1})$

$N \leftarrow N + 1$

return Φ^N, X^N

Greedy procedure for the optimal control problem I

- 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 $Y_\mu^N = (I + M\Lambda_\mu^R)X^N$:

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

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

- 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 $Y_\mu^N = (I + M\Lambda_\mu^R)X^N$:

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

  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 $X^N \ni \tilde{\varphi}_\mu^N \approx \varphi_\mu^*(T) \in X$:

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

Greedy procedure for the optimal control problem II

- We therefore have

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

and

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

Greedy procedure for the optimal control problem II

- We therefore have

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

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:

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

Greedy procedure for the optimal control problem III

- Choose a finite training set $\mathcal{P}_{\text{train}} \subset \mathcal{P}$.

Greedy procedure for the optimal control problem III

- Choose a finite training set $\mathcal{P}_{\text{train}} \subset \mathcal{P}$.
- Select next parameter $\mu_{N+1} \in \mathcal{P}_{\text{train}}$ as

$$\mu_{N+1} = \operatorname{argmax}_{\mu \in \mathcal{P}_{\text{train}}} \eta_\mu(\tilde{\varphi}_\mu^N).$$

Greedy procedure for the optimal control problem III

- Choose a finite training set $\mathcal{P}_{\text{train}} \subset \mathcal{P}$.
- Select next parameter $\mu_{N+1} \in \mathcal{P}_{\text{train}}$ as

$$\mu_{N+1} = \operatorname{argmax}_{\mu \in \mathcal{P}_{\text{train}}} \eta_\mu(\tilde{\varphi}_\mu^N).$$

- Replace $\operatorname{dist}(X^N, \mathcal{M})$ by $\eta_{\mu_{N+1}}(\tilde{\varphi}^N_{\mu_{N+1}})$.

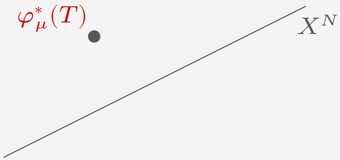
Summary of the notation

- $\varphi_\mu^*(T) \in X$: optimal final time adjoint
- $X^N \subset X$: reduced space
- $P_{X^N}(\varphi)$: projection of $\varphi \in X$ onto X^N
- $\tilde{\varphi}_\mu^N \in X^N$: approximate final time adjoint
- $Y_\mu^N = (I + M\Lambda_\mu^R)X^N$: space of final time states reachable from X^N
- $P_{Y_\mu^N}(x)$: projection of $x \in X$ onto Y_μ^N

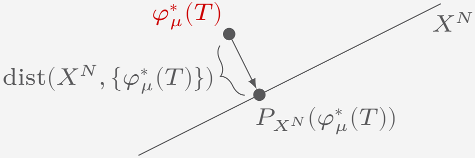
Visualization

$\varphi_\mu^*(T)$

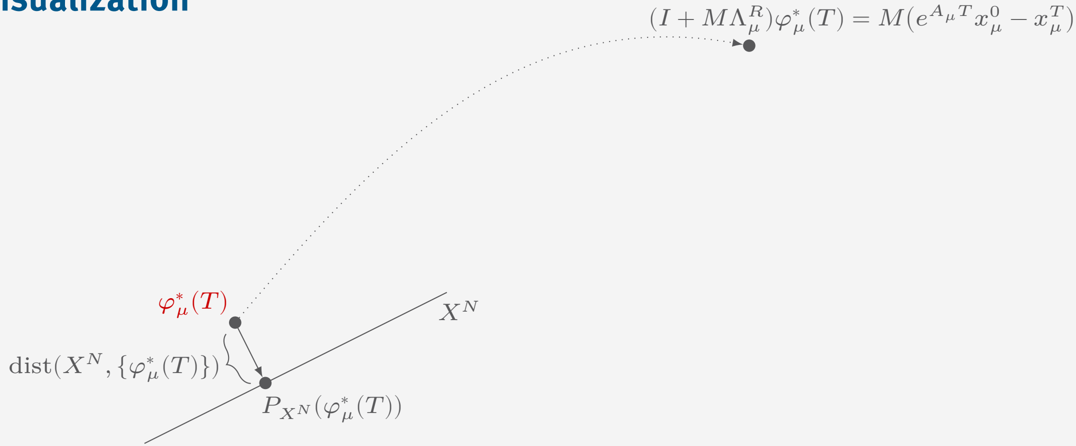
Visualization



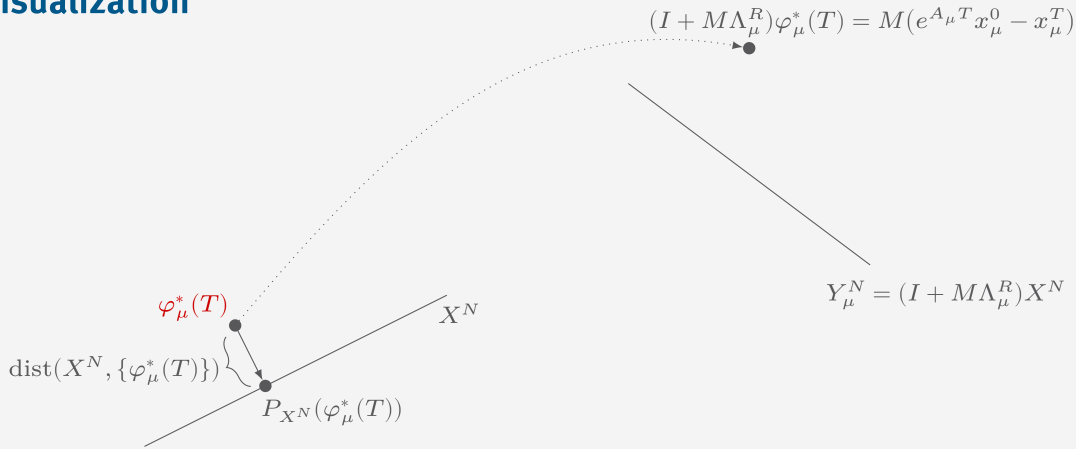
Visualization



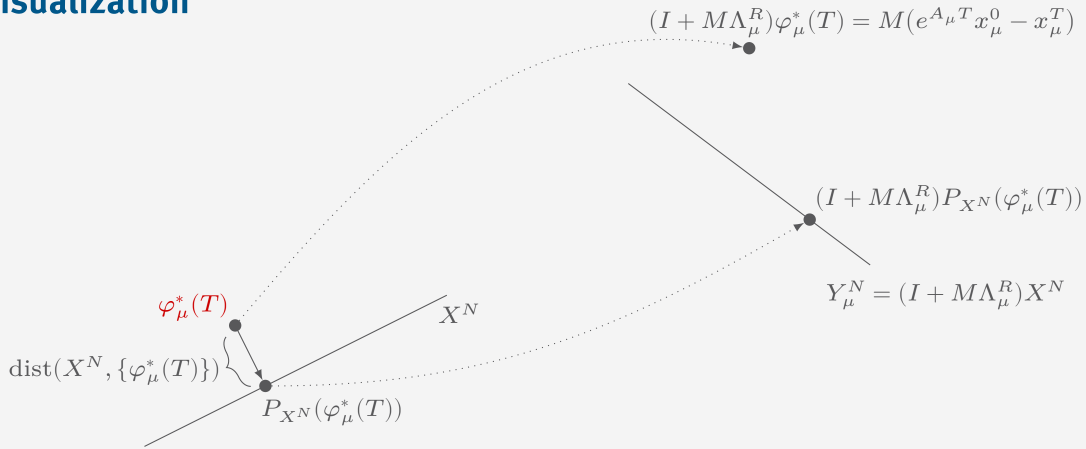
Visualization



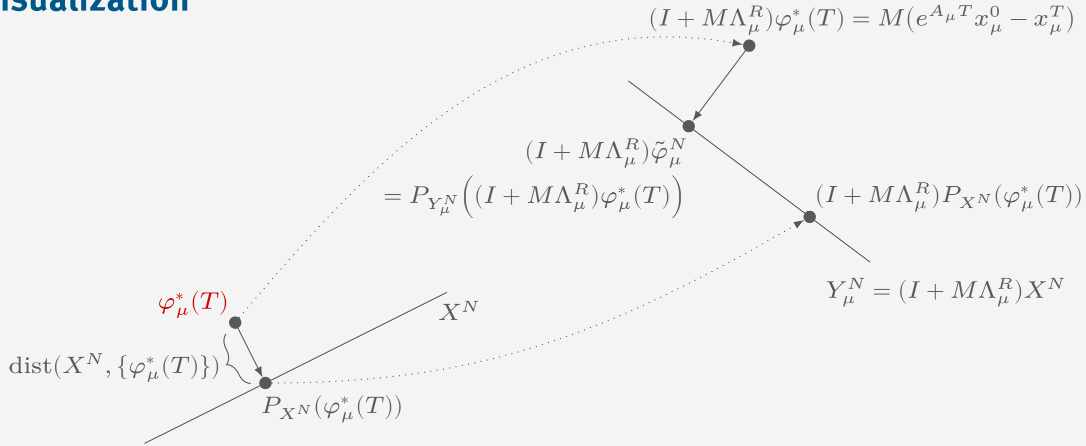
Visualization



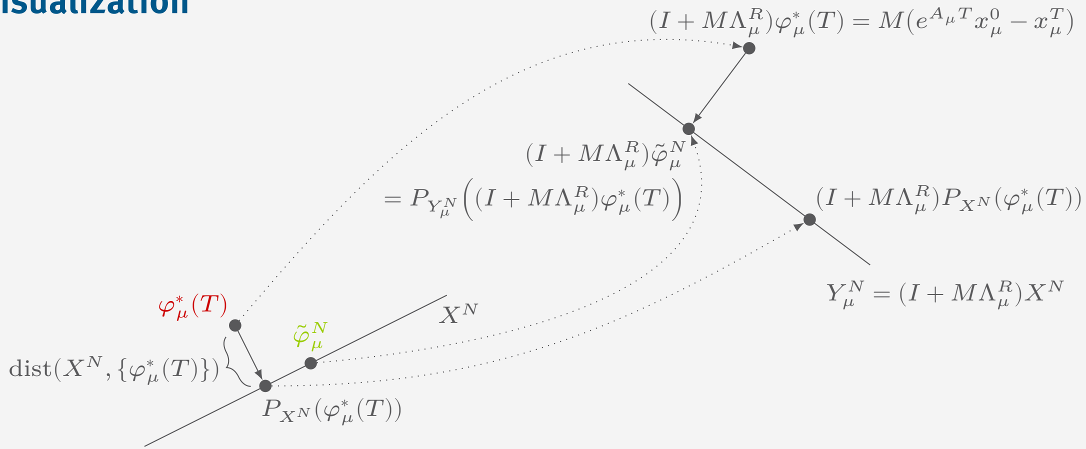
Visualization



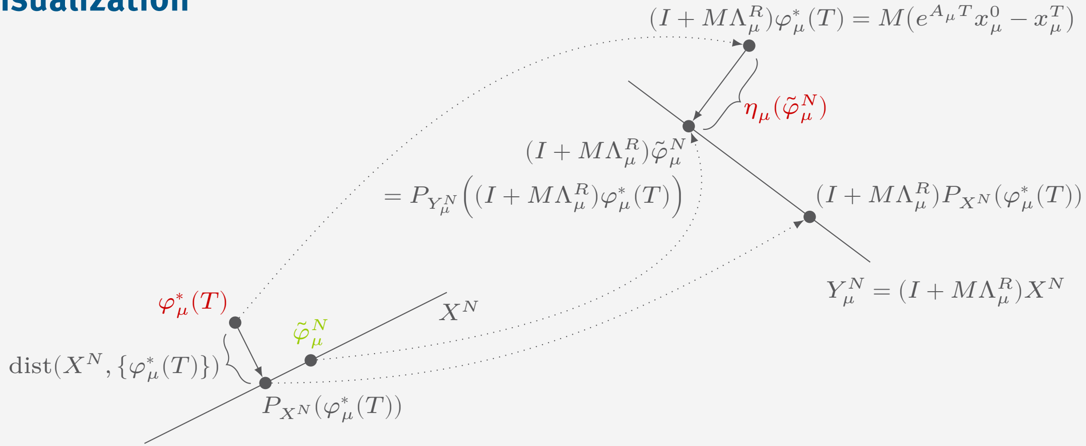
Visualization



Visualization



Visualization



Analysis of the greedy algorithm

Theorem 3 (Weak greedy algorithm and approximation error)

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

$$\gamma := \frac{1}{C_{\varphi^*} + C_\Lambda} \leq 1,$$

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

It further holds for all $\mu \in \mathcal{P}$ that

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

Online computations of the reduced order model (RB ROM)

Given a new parameter $\mu \in \mathcal{P}$:

- compute $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i$ for $i = 1, \dots, N$

Online computations of the reduced order model (RB ROM)

Given a new parameter $\mu \in \mathcal{P}$:

- compute $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i$ for $i = 1, \dots, N$
- assemble operator $\bar{X}_\mu = [x_1^\mu \cdots x_N^\mu] \in \mathcal{L}(\mathbb{R}^N, X)$ for projection onto Y_μ^N

Online computations of the reduced order model (RB ROM)

Given a new parameter $\mu \in \mathcal{P}$:

- compute $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i$ for $i = 1, \dots, N$
- assemble operator $\bar{X}_\mu = [x_1^\mu \cdots x_N^\mu] \in \mathcal{L}(\mathbb{R}^N, X)$ for projection onto Y_μ^N
- compute coefficients $\alpha^\mu = (\alpha_1^\mu, \dots, \alpha_N^\mu)^\top \in \mathbb{R}^N$ as solution of the $N \times N$ linear system

$$\bar{X}_\mu^* \bar{X}_\mu \alpha^\mu = \bar{X}_\mu^* M \left(e^{A_\mu T} x_\mu^0 - x_\mu^T\right)$$

Online computations of the reduced order model (RB ROM)

Given a new parameter $\mu \in \mathcal{P}$:

- compute $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i$ for $i = 1, \dots, N$
- assemble operator $\bar{X}_\mu = [x_1^\mu \cdots x_N^\mu] \in \mathcal{L}(\mathbb{R}^N, X)$ for projection onto Y_μ^N
- compute coefficients $\alpha^\mu = (\alpha_1^\mu, \dots, \alpha_N^\mu)^\top \in \mathbb{R}^N$ as solution of the $N \times N$ linear system

$$\bar{X}_\mu^* \bar{X}_\mu \alpha^\mu = \bar{X}_\mu^* M \left(e^{A_\mu T} x_\mu^0 - x_\mu^T \right)$$

- compute final time adjoint $\tilde{\varphi}_\mu^N = \sum_{i=1}^N \alpha_i^\mu \varphi_i$

Online computations of the reduced order model (RB ROM)

Given a new parameter $\mu \in \mathcal{P}$:

- compute $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i$ for $i = 1, \dots, N$
- assemble operator $\bar{X}_\mu = [x_1^\mu \ \cdots \ x_N^\mu] \in \mathcal{L}(\mathbb{R}^N, X)$ for projection onto Y_μ^N
- compute coefficients $\alpha^\mu = (\alpha_1^\mu, \dots, \alpha_N^\mu)^\top \in \mathbb{R}^N$ as solution of the $N \times N$ linear system

$$\bar{X}_\mu^* \bar{X}_\mu \alpha^\mu = \bar{X}_\mu^* M \left(e^{A_\mu T} x_\mu^0 - x_\mu^T \right)$$

- compute final time adjoint $\tilde{\varphi}_\mu^N = \sum_{i=1}^N \alpha_i^\mu \varphi_i$
- solve $-\dot{\tilde{\varphi}}_\mu(t) = A_\mu^* \tilde{\varphi}_\mu(t)$ for $t \in [0, T]$, $\quad \tilde{\varphi}_\mu(T) = \tilde{\varphi}_\mu^N$ backwards in time

Online computations of the reduced order model (RB ROM)

Given a new parameter $\mu \in \mathcal{P}$:

- compute $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i$ for $i = 1, \dots, N$
- assemble operator $\bar{X}_\mu = [x_1^\mu \; \cdots \; x_N^\mu] \in \mathcal{L}(\mathbb{R}^N, X)$ for projection onto Y_μ^N
- compute coefficients $\alpha^\mu = (\alpha_1^\mu, \dots, \alpha_N^\mu)^\top \in \mathbb{R}^N$ as solution of the $N \times N$ linear system

$$\bar{X}_\mu^* \bar{X}_\mu \alpha^\mu = \bar{X}_\mu^* M \left(e^{A_\mu T} x_\mu^0 - x_\mu^T \right)$$

- compute final time adjoint $\tilde{\varphi}_\mu^N = \sum_{i=1}^N \alpha_i^\mu \varphi_i$
- solve $-\dot{\tilde{\varphi}}_\mu(t) = A_\mu^* \tilde{\varphi}_\mu(t)$ for $t \in [0, T]$, $\quad \tilde{\varphi}_\mu(T) = \tilde{\varphi}_\mu^N$ backwards in time
- compute associated control $\tilde{u}_\mu^N(t) = -R^{-1} B_\mu^* \tilde{\varphi}_\mu(t)$

Machine learning of reduced coefficients

- Costly part in the online phase of the reduced order model:
  Computation of $x_i^{\mu} = (I + M\Lambda_{\mu}^{R})\varphi_i$ for $i = 1, \dots, N$.

Machine learning of reduced coefficients

- Costly part in the online phase of the reduced order model:
  Computation of $x_i^{\mu} = (I + M\Lambda_{\mu}^R)\varphi_i$ for $i = 1, \dots, N$.
- Instead: Learn the map from parameters to coefficients, i.e. approximate the map

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

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

Machine learning of reduced coefficients

- Costly part in the online phase of the reduced order model:
  Computation of $x_i^\mu = (I + M\Lambda_\mu^R)\varphi_i$ for $i = 1, \dots, N$.
- Instead: Learn the map from parameters to coefficients, i.e. approximate the map

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

  by machine learning surrogate $\hat{\pi}_N \colon \mathcal{P} \to \mathbb{R}^N$.
- Approximate the final time adjoint as

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

Training procedure for the machine learning

- Training data is automatically generated during the greedy algorithm.

Training procedure for the machine learning

- Training data is automatically generated during the greedy algorithm.
- Additional training data can be generated by solving the reduced order model.

Training procedure for the machine learning

- Training data is automatically generated during the greedy algorithm.
- Additional training data can be generated by solving the reduced order model.
- Machine learning surrogate is trained on the training data during the offline phase.

Training procedure for the machine learning

- Training data is automatically generated during the greedy algorithm.
- Additional training data can be generated by solving the reduced order model.
- Machine learning surrogate is trained on the training data during the offline phase.
- Improvements of the reduced order model and the machine learning surrogate during the online phase are possible as well, see also [Haasdonk et al'23].

Error estimates for machine learning approximation

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

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

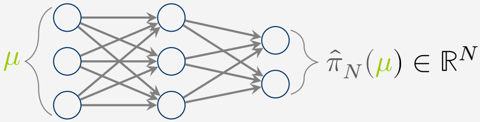
- A posteriori bound:

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

Machine learning approaches

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

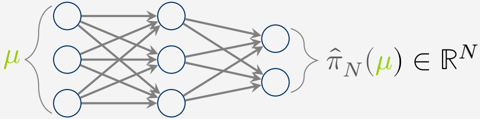
- Deep neural networks (DNN), see for instance [Petersen, Voigtlaender'18].



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].

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

kernel

subset of selected centers

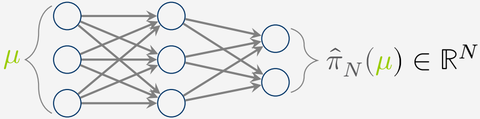
coefficients

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

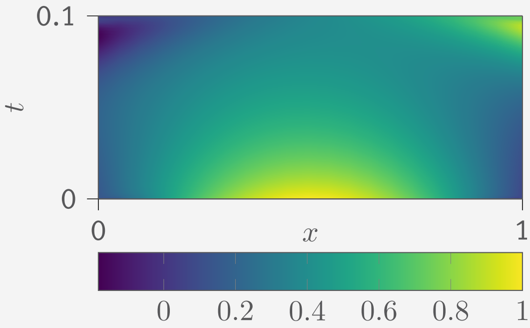
- Spatial discretization: Second order central finite difference scheme

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

- Temporal discretization: Crank-Nicolson scheme (implicit)
- 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



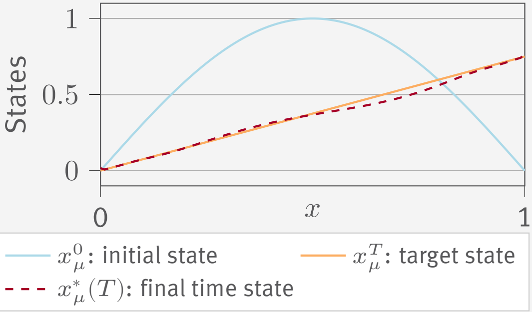


Figure: Optimal state x_μ^* in a space-time plot (left) and initial x_μ^0, final $x_\mu^*(T)$ and target x_μ^T states (right) for the parameter $\mu = (1.5, 0.75)$ in the heat equation example.

Numerical example: Parametrized heat equation

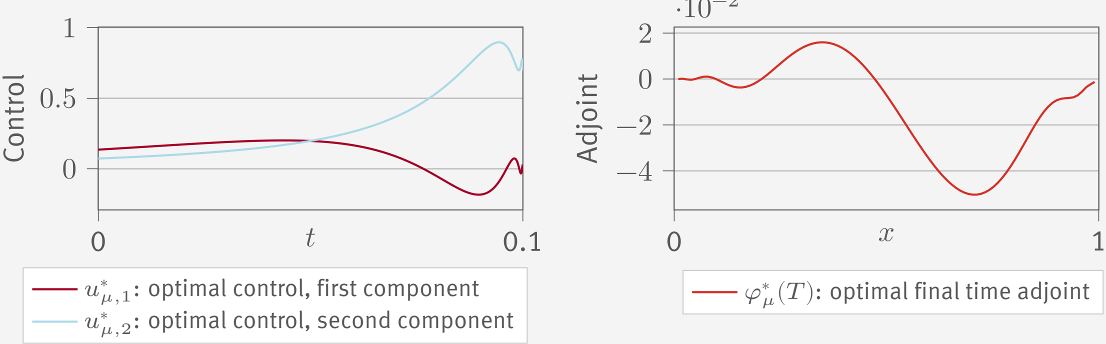
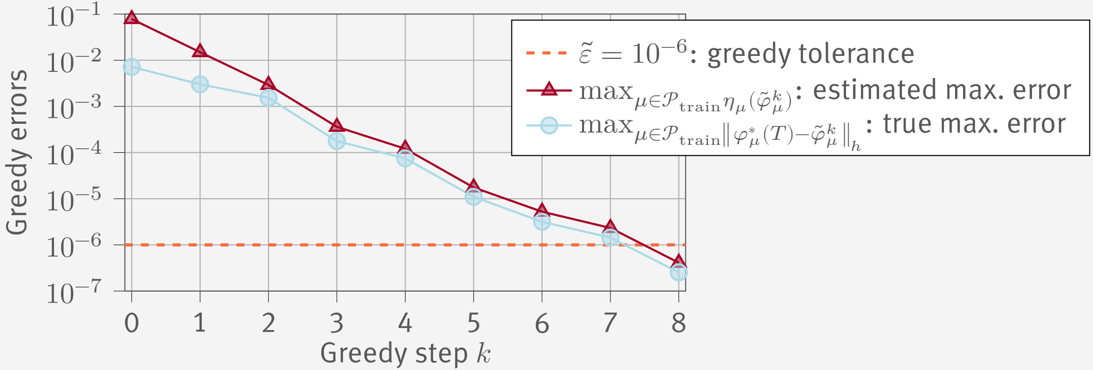


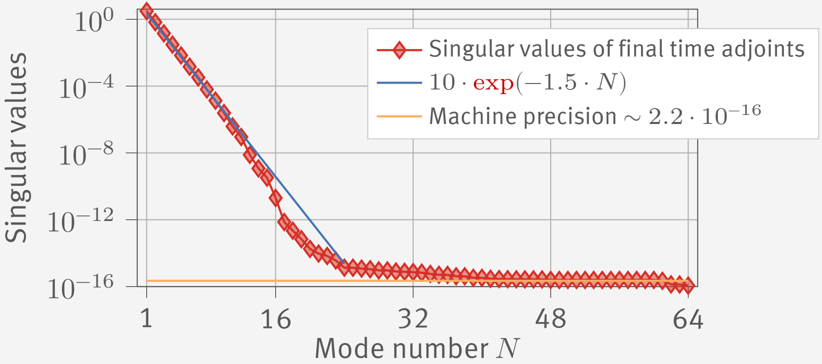
Figure: Optimal control u^*_μ (left) and optimal final time adjoint $\varphi^*_\mu(T)$ (right) for the parameter $\mu = (1.5, 0.75)$ in the heat equation example.

Numerical example: Parametrized heat equation

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



Singular value decay of optimal final time adjoints

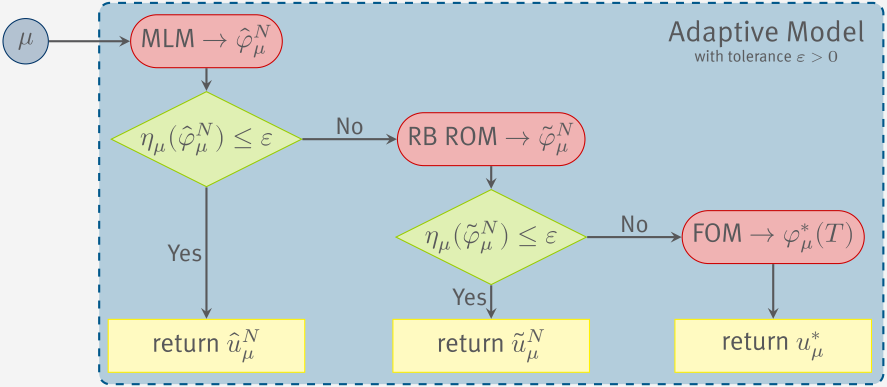


Numerical example: Parametrized heat equation

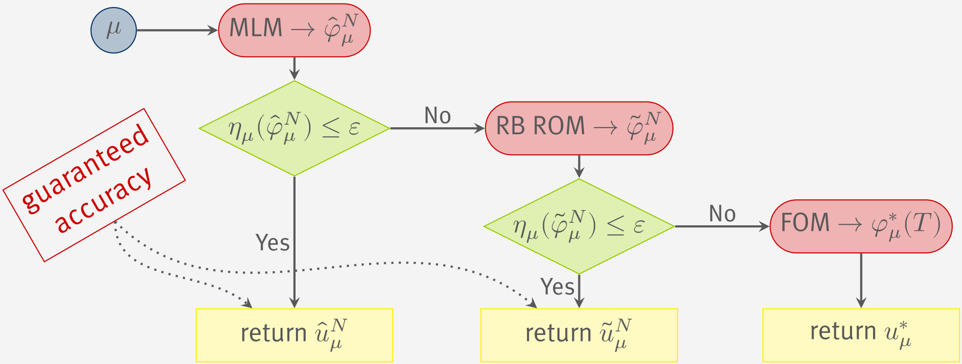
Results on a set of 100 randomly drawn test parameters:

Method	Avg. error adjoint	Avg. error control	Avg. runtime (s)	Avg. speedup
Exact solution	—	—	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

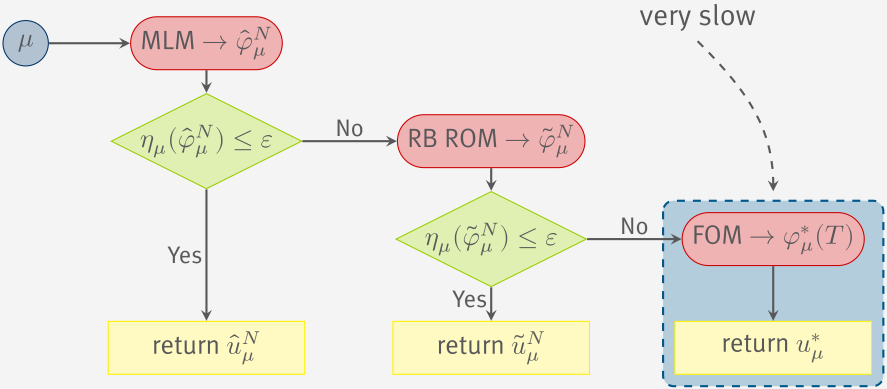
Outlook: Adaptive model hierarchy [Haasdonk et al'23]



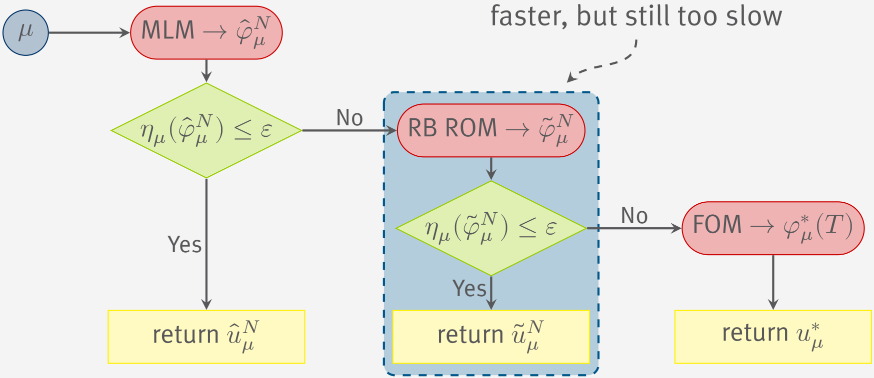
Outlook: Adaptive model hierarchy [Haasdonk et al'23]



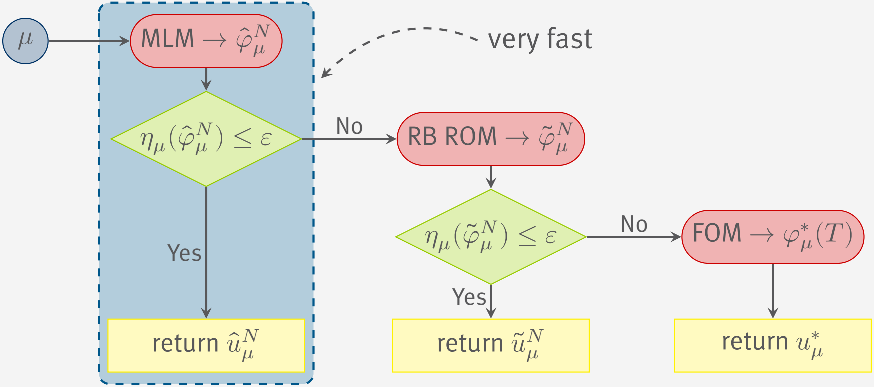
Outlook: Adaptive model hierarchy [Haasdonk et al'23]



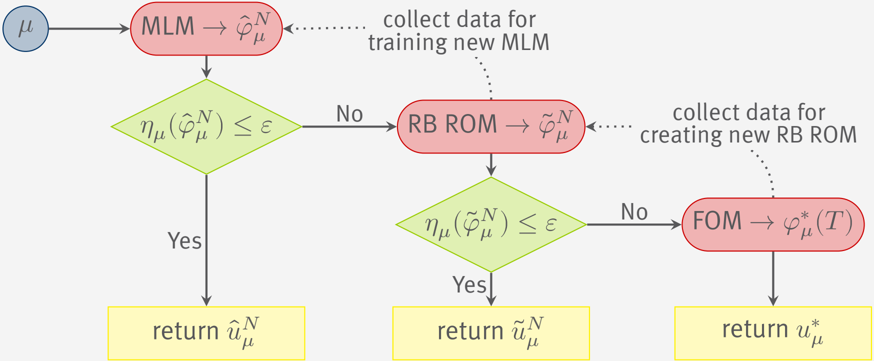
Outlook: Adaptive model hierarchy [Haasdonk et al'23]



Outlook: Adaptive model hierarchy [Haasdonk et al'23]



Outlook: Adaptive model hierarchy [Haasdonk et al'23]



Future research

- Application of the adaptive model hierarchy from [Haasdonk et al'23].

Future research

- Application of the adaptive model hierarchy from [Haasdonk et al'23].
- Advection-diffusion-reaction equations in higher space dimensions as test cases.

Future research

- Application of the adaptive model hierarchy from [Haasdonk et al'23].
- Advection-diffusion-reaction equations in higher space dimensions as test cases.
- Incorporation of theoretical approximation results for the machine learning methods.

Future research

- Application of the adaptive model hierarchy from [Haasdonk et al'23].
- 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.

Future research

- Application of the adaptive model hierarchy from [Haasdonk et al'23].
- 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.
- Extension to other classes of optimal control problems, such as linear time-varying systems or problems with constraints on the control.

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.
https://arxiv.org/abs/2307.15590

The source code for the paper is available open source:

- https://github.com/HenKlei/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

References II

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

- 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