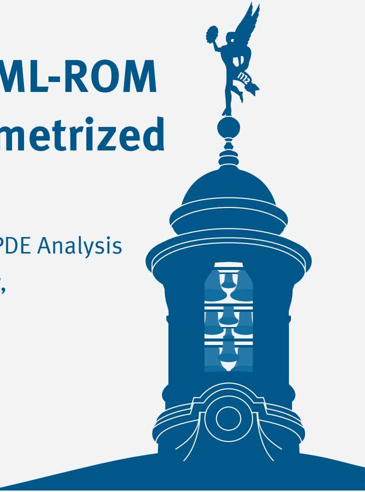
WWU
MÜNSTER

A certified and adaptive RB-ML-ROM surrogate approach for parametrized PDEs

HCM Workshop: Synergies between Data Science and PDE Analysis

Bernard Haasdonk, Hendrik Kleikamp, Mario Ohlberger,
Felix Schindler, Tizian Wenzel

June 13 – 17, 2022



living.knowledge

MM
Mathematics
Münster
Cluster of Excellence

Motivation and setting

Parametrized parabolic PDE in weak form: Find $u(\mu) \in L^2(0,T;V)$ s.t.

$$\begin{aligned} \langle \partial_t u(\mu), v \rangle + a(u(\mu), v; \mu) &= l(v; \mu), && \text{for all } v \in V, \\ u(0;\mu) &= u_0(\mu), && \text{for } u_0(\mu) \in V, \end{aligned}$$

with coercive bilinear form a, linear functional l and initial condition u_0.

Motivation and setting

Parametrized parabolic PDE in weak form: Find $u(\mu) \in L^2(0,T;V)$ s.t.

$$\begin{aligned} \langle \partial_t u(\mu), v \rangle + a(u(\mu), v; \mu) &= l(v; \mu), && \text{for all } v \in V, \\ u(0;\mu) &= u_0(\mu), && \text{for } u_0(\mu) \in V, \end{aligned}$$

with coercive bilinear form a, linear functional l and initial condition u_0.

Applications:

- Many-query scenarios, i.e. solve for many parameters $\mu \in \mathcal{P}$.
- Real-time solution, i.e. solve in real-time for new parameter $\mu \in \mathcal{P}$.

Motivation and setting

Parametrized parabolic PDE in weak form: Find $u(\mu) \in L^2(0,T;V)$ s.t.

$$\begin{aligned} \langle \partial_t u(\mu), v \rangle + a(u(\mu), v; \mu) &= l(v; \mu), && \text{for all } v \in V, \\ u(0;\mu) &= u_0(\mu), && \text{for } u_0(\mu) \in V, \end{aligned}$$

with coercive bilinear form a, linear functional l and initial condition u_0.

Applications:

- Many-query scenarios, i.e. solve for many parameters $\mu \in \mathcal{P}$.
- Real-time solution, i.e. solve in real-time for new parameter $\mu \in \mathcal{P}$.

Need for Model Order Reduction!

Reduced basis (RB) reduced order models

- Build reduced space $V_N \subset V_h$ with $N := \dim V_N \ll \dim V_h$ (e.g. using Greedy algorithm, POD, HaPOD,...).
- Perform Galerkin projection onto V_N instead of V_h.
- Full offline-online-decomposition under certain assumptions on parameter dependence of a and l.
- Solve small linear system in each time step t_k for the coefficients $\underline{u_{\text{RB}}(\mu; t_k)} \in \mathbb{R}^N$ with respect to the basis $\varphi_1, \dots, \varphi_N$ of V_N.
- Reconstruct approximate solution by linear combination of $\varphi_1, \dots, \varphi_N$.

Reduced basis (RB) reduced order models

- Build reduced space $V_N \subset V_h$ with $N := \dim V_N \ll \dim V_h$ (e.g. using Greedy algorithm, POD, HaPOD,...).
- Perform Galerkin projection onto V_N instead of V_h.
- Full offline-online-decomposition under certain assumptions on parameter dependence of a and l.
- Solve small linear system in each time step t_k for the coefficients $u_{\text{RB}}(\mu; t_k) \in \mathbb{R}^N$ with respect to the basis $\varphi_1, \dots, \varphi_N$ of V_N.
- Reconstruct approximate solution by linear combination of $\varphi_1, \dots, \varphi_N$.

Reduced basis (RB) reduced order models

- Build reduced space $V_N \subset V_h$ with $N := \dim V_N \ll \dim V_h$ (e.g. using Greedy algorithm, POD, HaPOD,...).
- Perform Galerkin projection onto V_N instead of V_h.
- Full offline-online-decomposition under certain assumptions on parameter dependence of a and l.
- Solve small linear system in each time step t_k for the coefficients $u_{\text{RB}}(\mu; t_k) \in \mathbb{R}^N$ with respect to the basis $\varphi_1, \dots, \varphi_N$ of V_N.
- Reconstruct approximate solution by linear combination of $\varphi_1, \dots, \varphi_N$.

Reduced basis (RB) reduced order models

- Build reduced space $V_N \subset V_h$ with $N := \dim V_N \ll \dim V_h$ (e.g. using Greedy algorithm, POD, HaPOD,...).
- Perform Galerkin projection onto V_N instead of V_h.
- Full offline-online-decomposition under certain assumptions on parameter dependence of a and l.
- Solve small linear system in each time step t_k for the coefficients $\underline{u_{\mathrm{RB}}(\mu; t_k)} \in \mathbb{R}^N$ with respect to the basis $\varphi_1, \dots, \varphi_N$ of V_N.
- Reconstruct approximate solution by linear combination of $\varphi_1, \dots, \varphi_N$.

Reduced basis (RB) reduced order models

- Build reduced space $V_N \subset V_h$ with $N := \dim V_N \ll \dim V_h$ (e.g. using Greedy algorithm, POD, HaPOD,...).
- Perform Galerkin projection onto V_N instead of V_h.
- Full offline-online-decomposition under certain assumptions on parameter dependence of a and l.
- Solve small linear system in each time step t_k for the coefficients $\underline{u_{\mathrm{RB}}(\mu; t_k)} \in \mathbb{R}^N$ with respect to the basis $\varphi_1, \ldots, \varphi_N$ of V_N.
- Reconstruct approximate solution by linear combination of $\varphi_1, \ldots, \varphi_N$.

Machine learning surrogate models

- Learn map from parameter $\mu \in \mathcal{P}$ to coefficients $\underline{u_{\mathsf{RB}}(\mu;t_k)} \in \mathbb{R}^N$ using machine learning, e.g. kernel models or neural networks.

Machine learning surrogate models

- Learn map from parameter $\mu \in \mathcal{P}$ to coefficients $\underline{u_{\mathrm{RB}}(\mu; t_k)} \in \mathbb{R}^N$ using machine learning, e.g. kernel models or neural networks.
- Two different approaches:
  1. *Time-"vectorized":* Train $\Phi\colon \mathbb{R}^p \to \mathbb{R}^{K \times N}$, with $p := \dim \mathcal{P}$ and K the number of time steps, to predict the ROM coefficients for all time steps at once.
  2. *"Random-access"-in-time:* Train $\Phi\colon \mathbb{R}^{p+1} \to \mathbb{R}^N$ to predict the ROM coefficients at any point in time.

Machine learning surrogate models

- Learn map from parameter $\mu \in \mathcal{P}$ to coefficients $\underline{u_{\mathsf{RB}}(\mu; t_k)} \in \mathbb{R}^N$ using machine learning, e.g. kernel models or neural networks.
- Two different approaches:
    1. *Time-"vectorized":* Train $\Phi\colon \mathbb{R}^p \to \mathbb{R}^{K\times N}$, with $p := \dim \mathcal{P}$ and K the number of time steps, to predict the ROM coefficients for all time steps at once.
    2. *"Random-access"-in-time:* Train $\Phi\colon \mathbb{R}^{p+1} \to \mathbb{R}^N$ to predict the ROM coefficients at any point in time.
- Obtain DoF-vector $\underline{u_{\mathsf{ML}}(\mu; t_k)} \in \mathbb{R}^N$ and corresponding solution $u_{\mathsf{ML}}(\mu; t_k) \in V_N$ by evaluating Φ for parameter $\mu \in \mathcal{P}$.

Comparison of the models

Full-order/high-fidelity model (FOM)

Pros:

- arbitrarily accurate solutions

Cons:

- very slow when considering fine discretizations and many time steps

Comparison of the models

Full-order/high-fidelity model (FOM)

Pros:

- arbitrarily accurate solutions

Cons:

- very slow when considering fine discretizations and many time steps

Reduced order model (ROM)

Pros:

- faster than FOM
- error estimator

Cons:

- slow iterative time stepping
- solve dense linear system in each time step

Comparison of the models

Full-order/high-fidelity model (FOM)

Pros:

- arbitrarily accurate solutions

Cons:

- very slow when considering fine discretizations and many time steps

Reduced order model (ROM)

Pros:

- faster than FOM
- error estimator

Cons:

- slow iterative time stepping
- solve dense linear system in each time step

Machine learning model (MLM)

Pros:

- faster than ROM
- parallel evaluation for different time instances

Cons:

- no certification available (so far)

Parabolic RB a posteriori error estimator

Offline-online-decomposable error estimate for difference between FOM-solution $u_h(\mu) \in V_h$ and ROM-solution $u_{\mathsf{RB}}(\mu) \in V_N$:

$$\|u_h(\mu) - u_{\mathsf{RB}}(\mu)\|_{L^2(0,T;V_h)} \leq E(u_{\mathsf{RB}}(\mu);\mu)$$

$$:= \underbrace{\alpha(\mu)}_{\text{coercivity constant}}^{-1} \left(\sum_{k=1}^{K-1} \Delta t \| \underbrace{R_k(u_{\mathsf{RB}}(t_k;\mu);\mu)}_{\text{residuum in the } k\text{-th time step}} \|_{V_h'}^2 \right)^{1/2}$$

Parabolic RB a posteriori error estimator

Offline-online-decomposable error estimate for difference between FOM-solution $u_h(\mu) \in V_h$ and ROM-solution $u_{\mathsf{RB}}(\mu) \in V_N$:

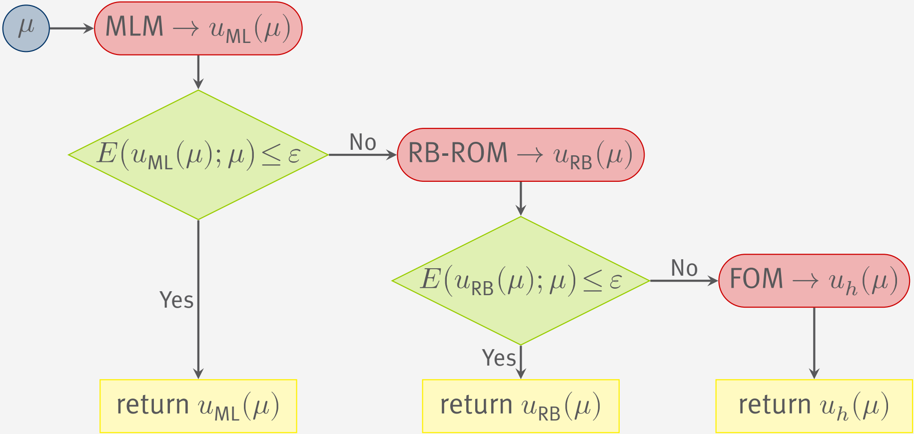
$$\|u_h(\mu) - u_{\mathsf{RB}}(\mu)\|_{L^2(0,T;V_h)} \leq E(u_{\mathsf{RB}}(\mu);\mu)$$

$$:= \underbrace{\alpha(\mu)}_{\text{coercivity constant}}{}^{-1} \left(\sum_{k=1}^{K-1} \Delta t \| \underbrace{R_k(u_{\mathsf{RB}}(t_k;\mu);\mu)}_{\text{residuum in the } k\text{-th time step}} \|_{V_h'}^2 \right)^{1/2}$$

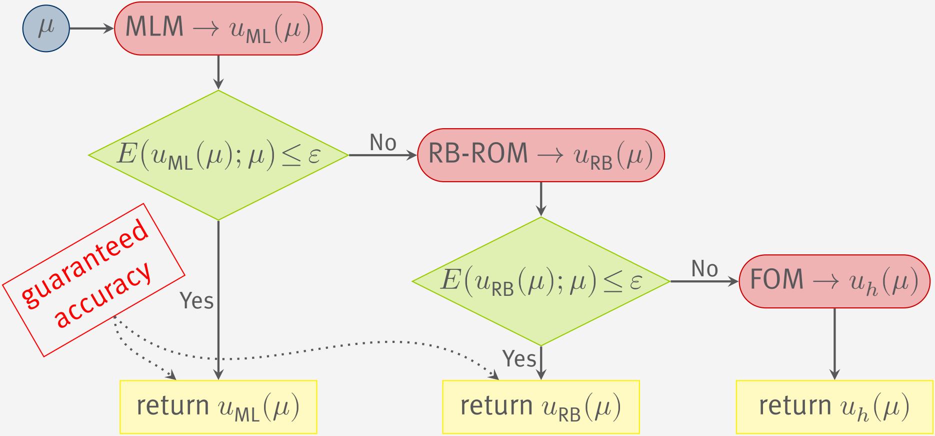
Idea: Apply error estimator E to machine learning prediction $u_{\mathsf{ML}}(\mu)$!

$\Longrightarrow$ A posteriori certification of machine learning results!

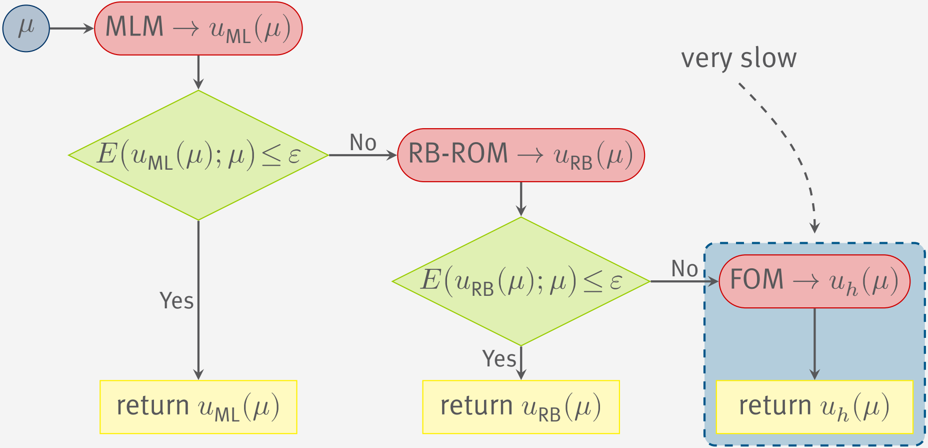
Adaptive model hierarchy



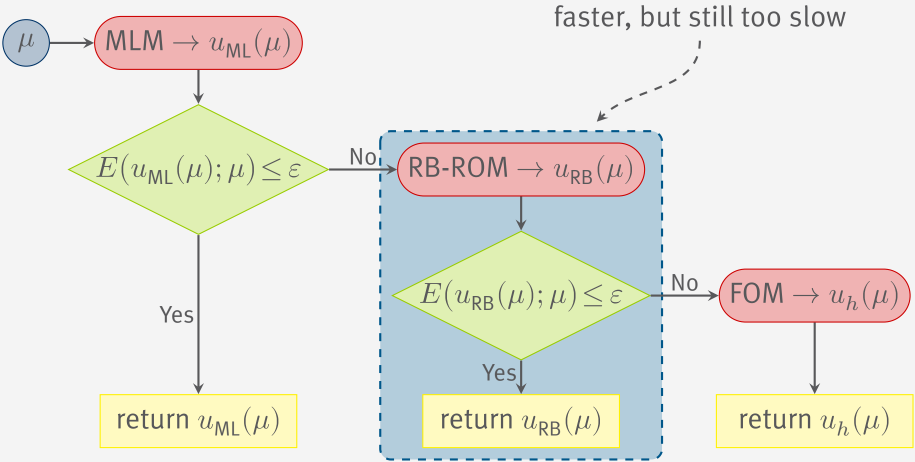
Adaptive model hierarchy



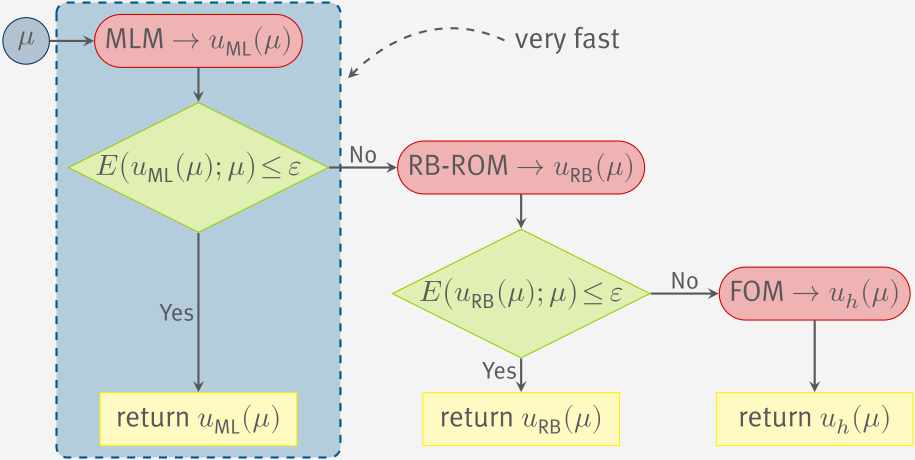
Adaptive model hierarchy



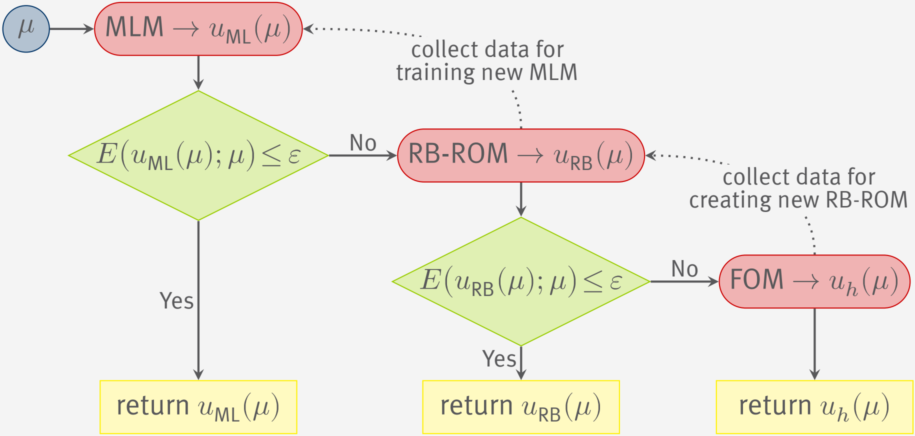
Adaptive model hierarchy



Adaptive model hierarchy



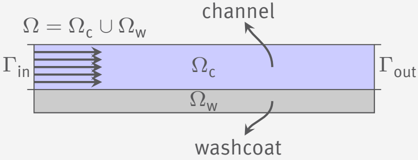
Adaptive model hierarchy



Numerical experiments I – PDE-constr. optimization

Advection-diffusion-reaction with $\mu = (\mathrm{Da}, \mathrm{Pe}) \in \mathcal{P} = [0.01, 10] \times [9, 11]$:

$$\begin{aligned} \partial_t u(\mu) - \nabla \cdot (\kappa \nabla u(\mu)) + \mathrm{Pe}\, \nabla \cdot (\vec{v}\, u(\mu)) + \mathrm{Da}\, \chi_{\Omega_{\mathrm{w}}} u(\mu) &= 0 && \text{in } \Omega \times (0,T), \\ (\kappa \nabla u(\mu)) \cdot n_{\partial\Omega} &= 0 && \text{on } \Gamma_{\mathrm{out}}, \\ u(\mu) &= \chi_{\Gamma_{\mathrm{in}}} && \text{on } \partial\Omega \setminus \Gamma_{\mathrm{out}}, \\ u(t=0;\mu) &= \chi_{\Gamma_{\mathrm{in}}} && \text{in } \Omega \end{aligned}$$



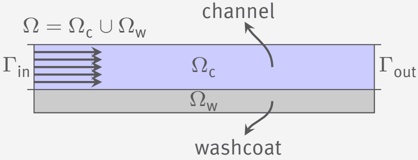
Numerical experiments I – PDE-constr. optimization

Advection-diffusion-reaction with $\mu = (\mathsf{Da}, \mathsf{Pe}) \in \mathcal{P} = [0.01, 10] \times [9, 11]$:

$$
\begin{aligned}
\partial_t u(\mu) - \nabla\cdot(\kappa \nabla u(\mu)) + \mathsf{Pe}\, \nabla\cdot(\vec{v}\, u(\mu)) + \mathsf{Da}\, \chi_{\Omega_{\mathsf{w}}} u(\mu) &= 0 && \text{in } \Omega \times (0,T),\\
(\kappa \nabla u(\mu)) \cdot n_{\partial\Omega} &= 0 && \text{on } \Gamma_{\mathsf{out}},\\
u(\mu) &= \chi_{\Gamma_{\mathsf{in}}} && \text{on } \partial\Omega \backslash \Gamma_{\mathsf{out}},\\
u(t=0;\mu) &= \chi_{\Gamma_{\mathsf{in}}} && \text{in } \Omega
\end{aligned}
$$

Break-through curve as output functional:

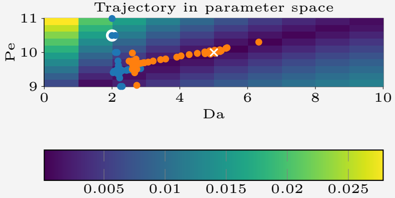
$$
f(t;\mu) = \frac{1}{|\Gamma_{\mathsf{out}}|} \int_{\Gamma_{\mathsf{out}}} u(t;\mu)\, \mathrm{d}s
$$

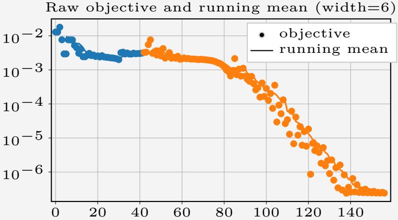


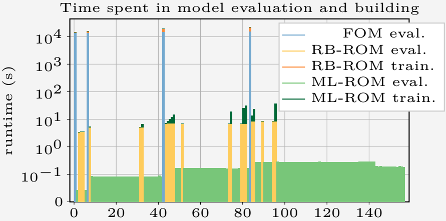
PDE-constrained optimization problem:

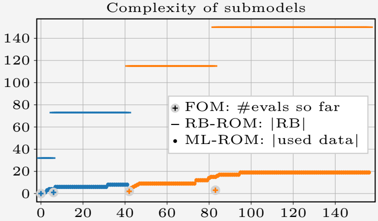
$$
\min_{\mu \in \mathcal{P}} \mathcal{J}(\mu) := \| \hat{f}_h - f_{\mathsf{adapt}}(\mu) \|_{L^\infty([0,T])}
$$

Numerical experiments I – PDE-constr. optimization







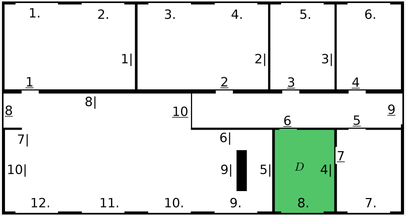


Numerical experiments II – Monte Carlo

Parameter-dependent bilinear form a and source term l:

$$a(u, v; \mu) = \int_\Omega \kappa(\mu)\, \nabla u(x) \cdot \nabla v(x)\, \mathrm{d}x,$$

$$l(v; \mu; t) = \int_\Omega v\, \ell(\mu; t)\, \mathrm{d}x.$$



Numerical experiments II – Monte Carlo

Parameter-dependent bilinear form a and source term l:

$$a(u, v; \mu) = \int_\Omega \kappa(\mu)\, \nabla u(x) \cdot \nabla v(x)\, \mathrm{d}x,$$

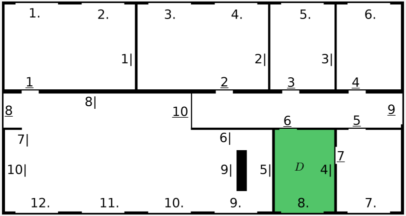
$$l(v; \mu; t) = \int_\Omega v\, \ell(\mu; t)\, \mathrm{d}x.$$

Quantity of interest at time $t \in [0, T]$:

$$f(\mu; t) := \frac{1}{|D|} \int_D u(t, x; \mu)\, \mathrm{d}x$$

Average over time interval $\tau \subset [0, T]$:

$$\bar{f}(\mu) := \frac{1}{|\tau|} \int_\tau f(\mu; t)\, \mathrm{d}t$$



Numerical experiments II – Monte Carlo

Expectation and variance of $\bar{f}$ with respect to density $\rho\colon \mathcal{P} \to [0,\infty)$:

$$\mathrm{E}[\bar{f}] := \int_{\mathcal{P}} \rho(\mu)\bar{f}(\mu)\,\mathrm{d}\mu, \qquad \mathrm{Var}[\bar{f}] := \int_{\mathcal{P}} \rho(\mu)\left(\bar{f}(\mu) - \mathrm{E}[\bar{f}]\right)^2\,\mathrm{d}\mu$$

Numerical experiments II – Monte Carlo

Expectation and variance of $\bar{f}$ with respect to density $\rho\colon \mathcal{P} \to [0,\infty)$:

$$\mathrm{E}[\bar{f}] := \int_{\mathcal{P}} \rho(\mu)\bar{f}(\mu)\,\mathrm{d}\mu, \qquad \mathrm{Var}[\bar{f}] := \int_{\mathcal{P}} \rho(\mu)\left(\bar{f}(\mu) - \mathrm{E}[\bar{f}]\right)^2\,\mathrm{d}\mu$$

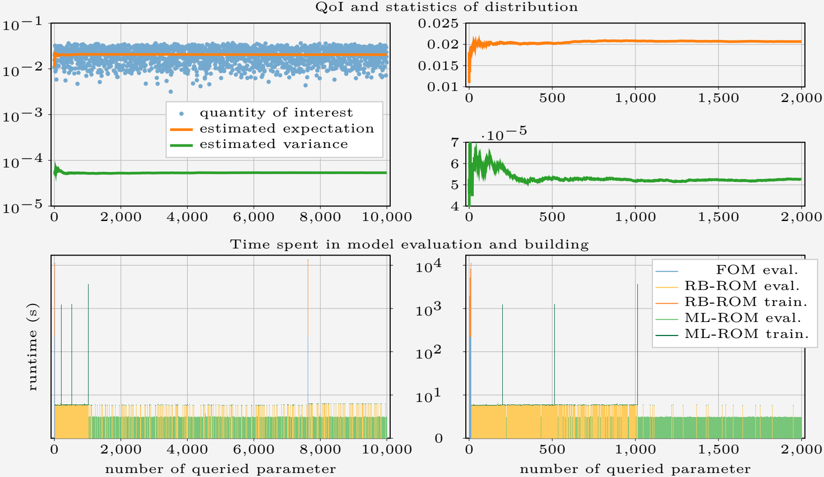
Monte Carlo estimation of $\mathrm{E}[\bar{f}]$ and $\mathrm{Var}[\bar{f}]$:

$$\mathrm{E}[\bar{f}] \approx \hat{\theta}_{\bar{f}} := \frac{1}{N_{\mathsf{mc}}} \sum_{\mu\in\mathcal{P}_{\mathsf{mc}}} \bar{f}(\mu),$$

$$\mathrm{Var}[\bar{f}] \approx \frac{1}{(N_{\mathsf{mc}}-1)} \sum_{\mu\in\mathcal{P}_{\mathsf{mc}}} \left(\bar{f}(\mu) - \hat{\theta}_{\bar{f}}\right)^2,$$

where $\mathcal{P}_{\mathsf{mc}} \subset \mathcal{P}$ is randomly sampled according to ρ.

Numerical experiments II – Monte Carlo



References I

B. Haasdonk, H. Kleikamp, M. Ohlberger, F. Schindler, T. Wenzel, *A new certified hierarchical and adaptive RB-ML-ROM surrogate model for parametrized PDEs*, arXiv, (2022), https://arxiv.org/abs/2204.13454.

M. Grepl and A. Patera, *A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations*, ESAIM: Mathematical Modelling and Numerical Analysis, 39 (2005), pp. 157–181, https://doi.org/10.1051/m2an:2005006.

References II

- Q. WANG, J. S. HESTHAVEN, AND D. RAY, *Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem*, J. Comput. Phys., 384 (2019), pp. 289–307, https://doi.org/10.1016/j.jcp.2019.01.031.

- G. SANTIN AND B. HAASDONK, *Kernel methods for surrogate modeling*, in Model Order Reduction, P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, eds., vol. 2, De Gruyter, 2021, pp. 311–353.