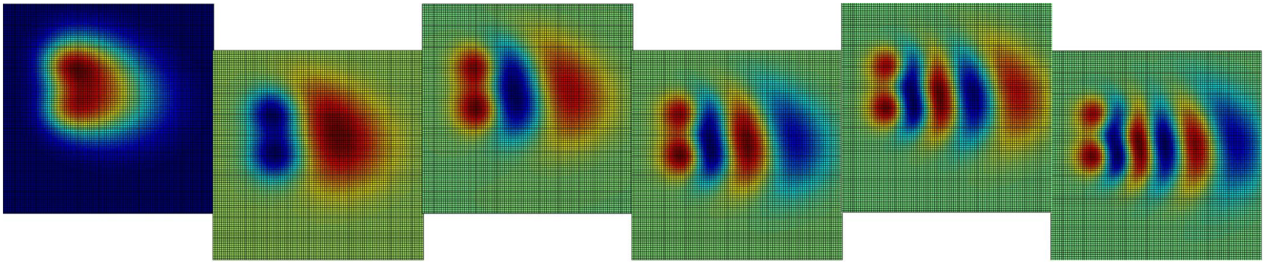
Westfälische Wilhelms-Universität Münster
CeNoS
Center for Nonlinear Science
Institut für Numerische und Angewandte Mathematik

Error Control and Adaptivity for Reduced Basis Approximations of Parametrized Evolution Equations



Mario Ohlberger

In cooperation with: M. Dihlmann, M. Drohmann, B. Haasdonk, G. Rozza

Workshop on *A posteriori error estimates and mesh adaptivity for evolutionary nonlinear problems*, Paris, July 7, 2010

mario.ohlberger@uni-muenster.de www.uni-muenster.de/math/num/ohlberger

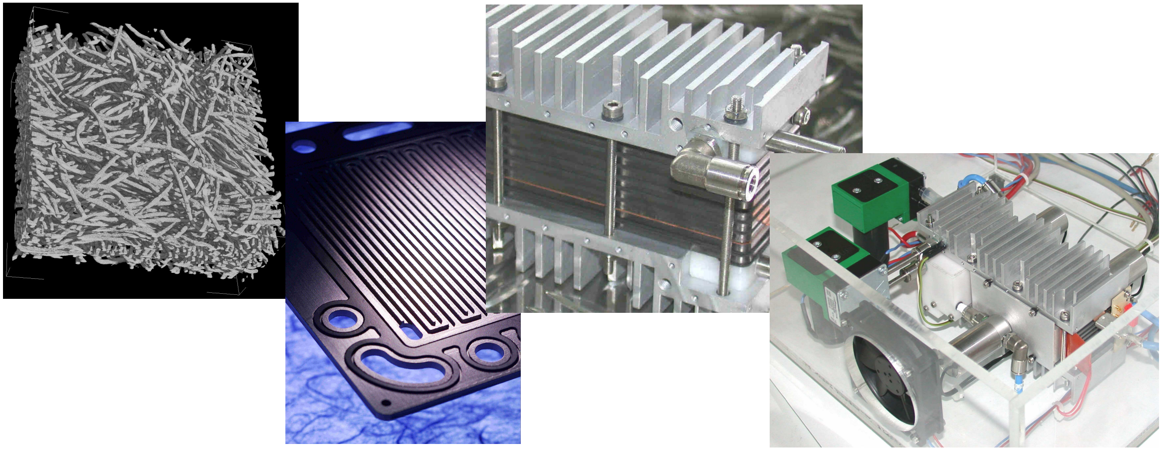
Outline

- Background: Multi-scale and multi-physics problems
  - Mathematical challenges
- Model reduction for parameterized evolution equations
  - Reduced basis methods for linear problems
  - Adaptive basis enrichment
  - Generalization to non-linear problems
- Current work and outlook

Multi-Scale- and Multi-Physics Problems

Example: PEM fuel cells

Pore **Cell** **Stack** **System**



[BMBF-Project **PEMDesign**: Fraunhofer ITWM and Fraunhofer ISE]

Multi-Scale- and Multi-Physics Problems

Example: PEM fuel cells

Porous layers:
- Two phase flow with phase transition
- Species transport with reaction for O_2, H_2, H_2O
- Potential flow for electrons
- Energy balance

Membrane:
- Two phase water transport
- Potential flow for protons
- Energy balance

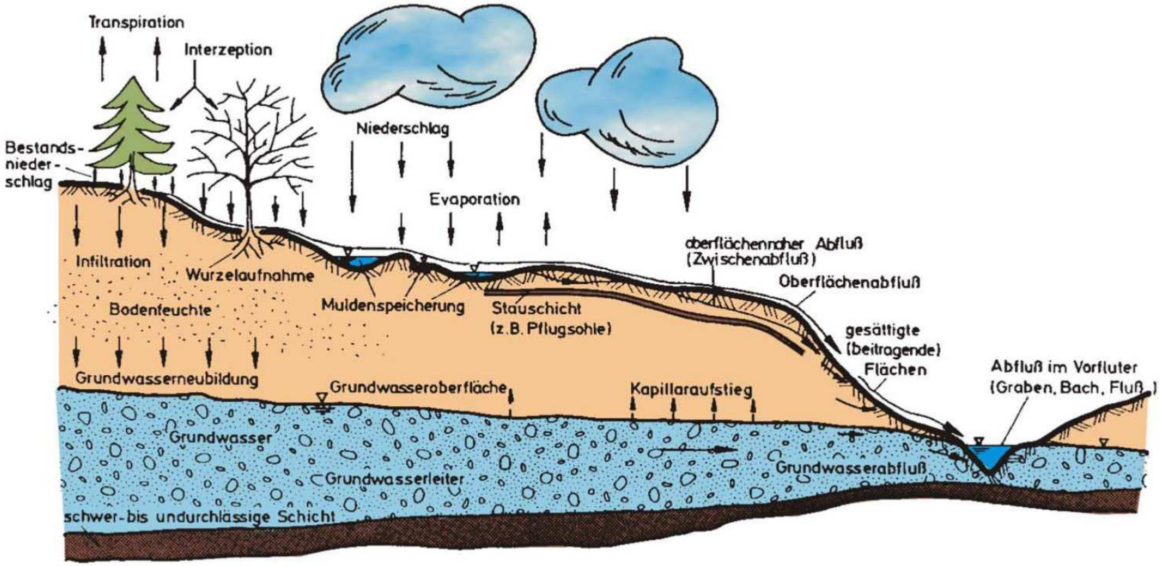
Gas channels:
- flow, species transport and energy balance

Bipolar plates:
- electron flow and energy balance

Coupling through interface conditions

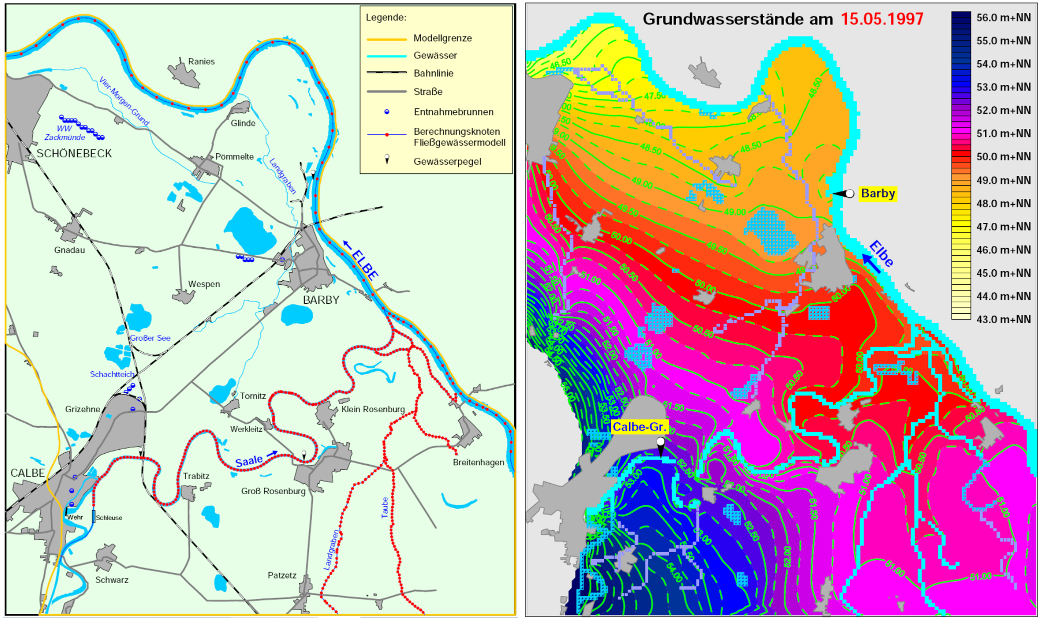
Multi-Scale- and Multi-Physics Problems

Example: Hydrological Modeling



[BMBF-Project **AdaptHydroMod**: Bronstert et al., Potsdam]

Multi-Scale- and Multi-Physics Problems



[BMBF-Project **AdaptHydroMod**: Wald & Corbe, Hügelsheim]

Mathematical Challenges and Possible Solutions

Multiscale Problem

Homogenization

Numerical Multiscale Methods

Efficient Numerical Methods

Higher Order Discretization

Adaptive Schemes

Parallelization

Model Reduction

Adaptive Modelling

Dimension Reductions

Reduced Basis Methods

Qualitative Analyse

Bifurcation

Optimization

Parameter Identification/ Inverse Modelling

System Simulation

Mathematical Challenges and Possible Solutions

Multiscale Problem

Homogenization

Numerical Multiscale Methods

Efficient Numerical Methods

Higher Order Discretization

Adaptive Schemes

Parallelization

Model Reduction

Adaptive Modelling

Dimension Reductions

Reduced Basis Methods

Qualitative Analyse

Bifurcation

Optimization

Parameter Identification/ Inverse Modelling

System Simulation

Reduced Basis Method for Linear Parameterized Evolution Equations

with B. Haasdonk and G. Rozza

Goal: Fast “Online”-Simulation of Complex Evolution Systems for

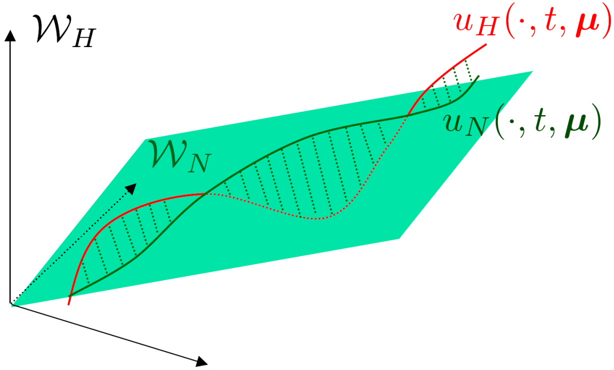
- Parameter Optimization
- Design Optimization
- Optimal Control
- Integration into System Simulation

Reduced Basis Method for Linear Parameterized Evolution Equations

with B. Haasdonk and G. Rozza

Goal: Fast "Online"-Simulation of Complex Evolution Systems for

- Parameter Optimization
- Design Optimization
- Optimal Control
- Integration into System Simulation



Ansatz:

- Reduced Basis Method (RB)

  $\dim(W_N) \ll \dim(W_H)$!

Some references:

notation RB [Noor, Peters '80], initial value problems [Porsching, Lee '87], method [Nguyen et al. '05], book [Patera, Rozza '07], http://augustine.mit.edu

Example: Convection-Diffusion Problem

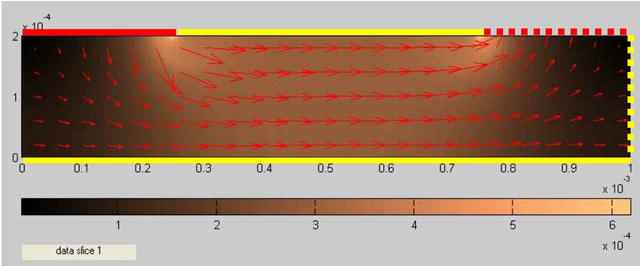
$$\partial_t c(\boldsymbol{\mu}) + \nabla \cdot (\mathbf{v}(\boldsymbol{\mu})c(\boldsymbol{\mu}) - d(\boldsymbol{\mu})\nabla u(\boldsymbol{\mu})) = 0 \;\text{ in } \Omega \times [0, T_{\max}],$$

$$\begin{aligned} c(\cdot, 0; \boldsymbol{\mu}) &= u_0(\boldsymbol{\mu}) \;\text{ in } \Omega, \\ c(\boldsymbol{\mu}) &= b_{\text{dir}}(\boldsymbol{\mu}) \;\text{ in } \Gamma_{\text{dir}} \times [0, T_{\max}], \\ (\mathbf{v}(\boldsymbol{\mu})c(\boldsymbol{\mu}) - d(\boldsymbol{\mu})\nabla c(\boldsymbol{\mu})) \cdot \mathbf{n} &= b_{\text{neu}}(u; \boldsymbol{\mu}) \;\text{ in } \Gamma_{\text{neu}} \times [0, T_{\max}]. \end{aligned}$$

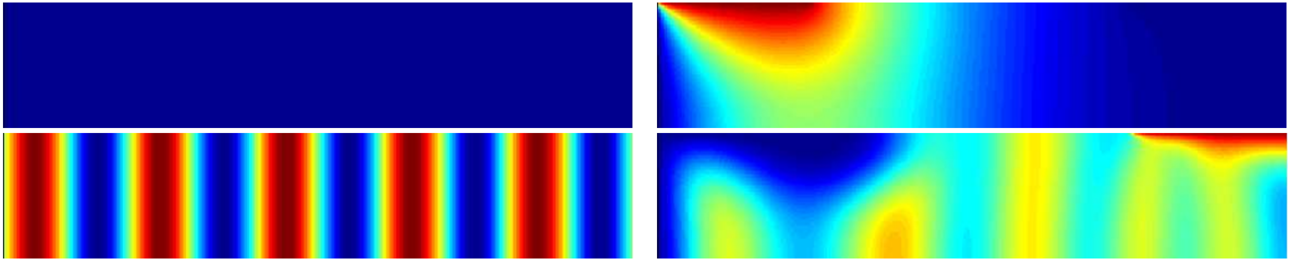
Discretization by Finite Volumes $\Longrightarrow$ $\mathbf{c_H}(\boldsymbol{\mu}) \in \mathbf{W_H}$.

Example: Convection-Diffusion Problem

Parameter:
- Initial Data
- Boundary Values
- Diffusion Parameter



Possible Variations of the Solution:



Model Reduction: Reduced Basis Method

Goal: Find $\mathbf{c}(\cdot, t; \mu) \in L^2(\Omega)$ for $t \in [0, T]$, $\mu \in P \subset \mathbb{R}^p$ with

$$\partial_t \mathbf{c}(\mu) + \mathbf{L}_{\mu}(\mathbf{c}(\mu)) = 0 \qquad \text{in } \Omega \times [0, T],$$

plus suitable Initial and Boundary Conditions.

Model Reduction: Reduced Basis Method

Goal: Find $\mathbf{c}(\cdot, t; \mu) \in L^2(\Omega)$ for $t \in [0, T]$, $\mu \in P \subset \mathbb{R}^p$ with

$$\partial_t \mathbf{c}(\mu) + L_{\mu}(\mathbf{c}(\mu)) = 0 \qquad \text{in } \Omega \times [0, T],$$

plus suitable Initial and Boundary Conditions.

Assumption: FV/DG Approximation $c_H(\mu) \in W_H$ for given Parameter μ

Model Reduction: Reduced Basis Method

Goal: Find $\mathbf{c}(\cdot, t; \boldsymbol{\mu}) \in L^2(\Omega)$ for $t \in [0, T]$, $\boldsymbol{\mu} \in P \subset \mathbb{R}^p$ with

$$\partial_t \mathbf{c}(\boldsymbol{\mu}) + \mathbf{L}_{\boldsymbol{\mu}}(\mathbf{c}(\boldsymbol{\mu})) = \mathbf{0} \qquad \text{in } \Omega \times [0, T],$$

plus suitable Initial and Boundary Conditions.

Assumption: FV/DG Approximation $c_H(\boldsymbol{\mu}) \in W_H$ for given Parameter $\boldsymbol{\mu}$

Ansatz (RB): Define low dimensional Subspace $W_N \subset W_H$
and project FV/DG Scheme onto the Subspace
$\Longrightarrow$ RB Approximation $c_N(\boldsymbol{\mu}) \in W_N$.

Model Reduction: Reduced Basis Method

Goal: Find $c(\cdot, t; \mu) \in L^2(\Omega)$ for $t \in [0, T]$, $\mu \in P \subset \mathbb{R}^p$ with

$$\partial_t c(\mu) + L_\mu(c(\mu)) = 0 \qquad \text{in } \Omega \times [0, T],$$

plus suitable Initial and Boundary Conditions.

Assumption: FV/DG Approximation $c_H(\mu) \in W_H$ for given Parameter μ

Ansatz (RB): Define low dimensional Subspace $W_N \subset W_H$
and project FV/DG Scheme onto the Subspace
$\Longrightarrow$ RB Approximation $c_N(\mu) \in W_N$.

Requirement:
- Efficient Choice of W_N (Exponential Convergence in N)
- Offline–Online Decomposition for fast Calculation of $c_N(\mu)$
- Error Control for $||c_H(\mu) - c_N(\mu)||$

Model Reduction: Reduced Basis Method

Assumption: **FV/DG Scheme for Evolution Equations of the Form**

$$c_H^0 = P[c_0(\boldsymbol{\mu})], \qquad L_I^k(\boldsymbol{\mu})[c_H^{k+1}(\boldsymbol{\mu})] = L_E^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})] + b^k(\boldsymbol{\mu}).$$

with time step counter k and $\mathbf{c_H^k}(\boldsymbol{\mu}) \in \mathbf{W_H}$.

Model Reduction: Reduced Basis Method

Assumption: **FV/DG Scheme for Evolution Equations of the Form**

$$c_H^0 = P[c_0(\boldsymbol{\mu})], \qquad L_I^k(\boldsymbol{\mu})[c_H^{k+1}(\boldsymbol{\mu})] = L_E^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})] + b^k(\boldsymbol{\mu}).$$

with time step counter k and $\mathbf{c_H^k}(\boldsymbol{\mu}) \in \mathbf{W_H}$.

RB Method: **Let $W_N \subset W_H$ be given, $\{\varphi_1, ..., \varphi_N\}$ a ONB of W_N.**
Sought: $\mathbf{c_N^k}(\boldsymbol{\mu}) = \sum_{n=1}^{N} \mathbf{a_n^k}(\boldsymbol{\mu})\varphi_n$ with

$$\mathbf{L}_I^k(\boldsymbol{\mu})\mathbf{a}^{k+1} \;=\; \mathbf{L}_E^k(\boldsymbol{\mu})\mathbf{a}^k + \mathbf{b}^k(\boldsymbol{\mu}).$$

We then have

$$(\mathbf{L}_I^k(\boldsymbol{\mu}))_{nm} := \int_\Omega \varphi_n L_I^k(\boldsymbol{\mu})[\varphi_m], \qquad (\mathbf{L}_E^k(\boldsymbol{\mu}))_{nm} := \int_\Omega \varphi_n L_E^k(\boldsymbol{\mu})[\varphi_m],$$

$$(\mathbf{a}^0(\boldsymbol{\mu}))_n = \int_\Omega P[c_0(\boldsymbol{\mu})]\varphi_n, \qquad (\mathbf{b}^k(\boldsymbol{\mu}))_n := \int_\Omega \varphi_n b^k(\boldsymbol{\mu}).$$

A Posteriori Error Estimates

Definition: **Residual of the FV/DG Method at Time t^k**

$$R^{k+1}(\boldsymbol{\mu})[c_N] := \frac{1}{\Delta t}\Big(L_I^k(\boldsymbol{\mu})[c_N^{k+1}(\boldsymbol{\mu})] - L_E^k(\boldsymbol{\mu})[c_N^k(\boldsymbol{\mu})] - b^k(\boldsymbol{\mu})\Big)$$

A Posteriori Error Estimates

Definition: **Residual of the FV/DG Method at Time t^k**

$$R^{k+1}(\boldsymbol{\mu})[c_N] := \frac{1}{\Delta t}\left(L_I^k(\boldsymbol{\mu})[c_N^{k+1}(\boldsymbol{\mu})] - L_E^k(\boldsymbol{\mu})[c_N^k(\boldsymbol{\mu})] - b^k(\boldsymbol{\mu})\right)$$

Theorem: **A Posteriori Error Estimate in $L^\infty L^2$**

$$\left\|c_N^k(\boldsymbol{\mu}) - c_H^k(\boldsymbol{\mu})\right\|_{L^2(\Omega)} \leq \sum_{l=0}^{k-1} \Delta t \, (C_E)^{k-1-l} \; \left\|R^{l+1}(\boldsymbol{\mu})[c_N(\boldsymbol{\mu})]\right\|_{L^2(\Omega)}$$

A Posteriori Error Estimates

Theorem: A Posteriori Error Estimate in a Weighted Energy Norm

$$||c_N^k(\boldsymbol{\mu}) - c_H^k(\boldsymbol{\mu})||_\gamma^2 \leq \frac{1}{4\alpha C(1-\gamma C)} \left(\sum_{l=0}^{k-1} \Delta t \left\| R^{l+1}[c_N(\boldsymbol{\mu})] \right\|_{L^2(\Omega)}^2 \right)$$

with weighted energy norm

$$||v||_\gamma^2 := \left\| v^k \right\|_{L^2(\Omega)}^2 + \gamma \left(\sum_{l=1}^{k} \Delta t \langle v^l, L_I^l[v^l] \rangle \right)$$

Offline–Online Decomposition

Goal: **All Steps for the Calculation of $c_N(\boldsymbol{\mu})$ and for the Calculation of the Error Estimator are split into Two Parts:**

- **Offline–Step: Complexity depending on dim(W_H)**
- **Online–Step: Complexity independent of dim(W_H)**

Offline–Online Decomposition

Goal: All Steps for the Calculation of $c_N(\boldsymbol{\mu})$ and for the Calculation of the Error Estimator are split into Two Parts:

- **Offline**–Step: Complexity **depending** on $\dim(W_H)$
- **Online**–Step: Complexity **independent** of $\dim(W_H)$

Constrained: Affine Parameter Dependency of the Evolution Scheme

$$L_I^k(\boldsymbol{\mu})[\cdot] = \sum_{q=1}^{Q} \underbrace{L_I^{k,q}[\cdot]}_{\text{depending on } x} \underbrace{\sigma_{L_I}^q(\boldsymbol{\mu})}_{\text{depending on } \mu}$$

$\Longrightarrow$ **Precompute offline:** $(L_I^{k,q})_{nm} := \int_\Omega \varphi_n L_I^{k,q}[\varphi_m]$

$\Longrightarrow$ **Assemble online:** $(L_I^k(\boldsymbol{\mu}))_{nm} := \sum_{q=1}^{Q} (L_I^{k,q})_{nm} \sigma_{L_I}^q(\boldsymbol{\mu})$

Offline–Online Decomposition

Goal: All Steps for the Calculation of $c_N(\boldsymbol{\mu})$ and for the Calculation of the Error Estimator are split into Two Parts:

- **Offline**–Step: Complexity **depending** on $\dim(W_H)$
- **Online**–Step: Complexity **independent** of $\dim(W_H)$

Constrained: Affine Parameter Dependency of the Evolution Scheme

$$L_I^k(\boldsymbol{\mu})[\cdot] \;=\; \sum_{q=1}^{Q} \underbrace{L_I^{k,q}[\cdot]}_{\text{depending on } x} \quad \underbrace{\sigma_{L_I}^q(\boldsymbol{\mu})}_{\text{depending on } \mu}$$

$\Longrightarrow$ Precompute **offline**: $(L_I^{k,q})_{nm} := \int_\Omega \varphi_n L_I^{k,q}[\varphi_m]$

$\Longrightarrow$ Assemble **online**: $(L_I^k(\boldsymbol{\mu}))_{nm} := \sum_{q=1}^{Q} (L_I^{k,q})_{nm} \sigma_{L_I}^q(\boldsymbol{\mu})$

Numerical Results

CPU-Time Comparison for the Convection-Diffusion Problem:

Discretization: 40×200 **Elements,** $K = 200$ **time steps**

	time dependent data			constant data		
	Reference	RB online	RB offline	Reference	RB online	RB offline
implicit	155.94s	16.67s	447.16s	45.67s	1.02s	2.41s
Factor		9.44			44.77	
explicit	105.97s	16.53s	437.20s	1.51s	0.79s	2.31s
Factor		6.41			1.91	

Discretization: 80×400 **Elements,** $K = 1000$ **time steps**

	time dependent data			constant data		
	Reference	RB online	RB offline	Reference	RB online	RB offline
implicit	4043.18s	143.57s	8693.90s	924.91s	6.18s	9.22s
Factor		28.27			149.66	
explicit	2758.20s	134.00s	8506.60s	17.41s	3.64s	8.83s
Factor		20.58			4.78	

Efficient Choice of W_N

Idea:

- Construct $\tilde{W}_{\tilde{N}}$ as the span of Snapshots $c_H(\mu)$, $\mu \in D \subset P$.
- Use Error Estimator for an efficient Choice of the Snapshots with Guaranteed Error Control on a Training Set.
- Reduce $\tilde{W}_{\tilde{N}}$ to W_N with Principal Component Analysis (PCA).

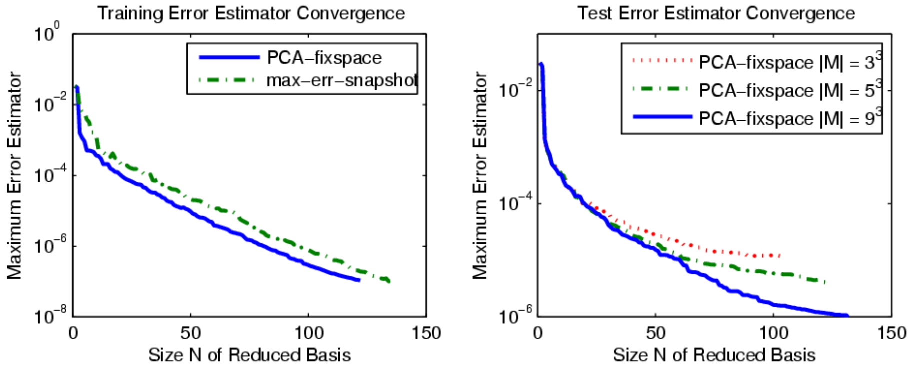
Goal: Exponential Convergence in N!

Efficient Choice of W_N

Idea:
- Construct $\tilde{W}_{\tilde{N}}$ as the span of Snapshots $c_H(\mu)$, $\mu \in D \subset P$.
- Use Error Estimator for an efficient Choice of the Snapshots with Guaranteed Error Control on a Training Set.
- Reduce $\tilde{W}_{\tilde{N}}$ to W_N with Principal Component Analysis (PCA).

Goal: Exponential Convergence in N!

Preliminary Result: Convergence in N for Training and Test Sets



Improvement through Adaptive Basis Enrichment

Algorithms for Basis Enrichment: Fixed / Adaptive Training Sets

$\text{ESGREEDY}(\Phi_0, M_{train}, \varepsilon_{tol}, M_{val}, \rho_{tol})$

$\Phi := \Phi_0$
repeat
  $\boldsymbol{\mu}^* := \arg\max_{\boldsymbol{\mu} \in M_{train}} \Delta(\boldsymbol{\mu}, \Phi)$
  if $\Delta(\boldsymbol{\mu}^*) > \varepsilon_{tol}$
   then
    $\varphi := \text{ONBASISEXT}(u_H(\boldsymbol{\mu}^*), \Phi)$
    $\Phi := \Phi \cup \{\varphi\}$
  $\varepsilon := \max_{\boldsymbol{\mu} \in M_{train}} \Delta(\boldsymbol{\mu}, \Phi)$
  $\rho := \max_{\boldsymbol{\mu} \in M_{val}} \Delta(\boldsymbol{\mu}, \Phi)/\varepsilon$
 until $\varepsilon \leq \varepsilon_{tol}$ **or** $\rho \geq \rho_{tol}$
return Φ, ε

Here, $\mathcal{M}$ denotes a Partition of the Parameter Space,
and $V(\mathcal{M})$ are the Vertices of $\mathcal{M}$.

mario.ohlberger@uni-muenster.de www.uni-muenster.de/math/num/ohlberger

Improvement through Adaptive Basis Enrichment

Algorithms for Basis Enrichment: Fixed / Adaptive Training Sets

$\text{ESGREEDY}(\Phi_0, M_{train}, \varepsilon_{tol}, M_{val}, \rho_{tol})$

$\Phi := \Phi_0$
repeat
    $\boldsymbol{\mu}^* := \arg\max_{\boldsymbol{\mu} \in M_{train}} \Delta(\boldsymbol{\mu}, \Phi)$
    if $\Delta(\boldsymbol{\mu}^*) > \varepsilon_{tol}$
      then
        $\varphi := \text{ONBASISEXT}(u_H(\boldsymbol{\mu}^*), \Phi)$
        $\Phi := \Phi \cup \{\varphi\}$
    $\varepsilon := \max_{\boldsymbol{\mu} \in M_{train}} \Delta(\boldsymbol{\mu}, \Phi)$
    $\rho := \max_{\boldsymbol{\mu} \in M_{val}} \Delta(\boldsymbol{\mu}, \Phi)/\varepsilon$
  until $\varepsilon \leq \varepsilon_{tol}$ **or** $\rho \geq \rho_{tol}$
return Φ, ε

$\text{RBADAPTIVE}(\Phi_0, \mathcal{M}_0, \varepsilon_{tol}, M_{val}, \rho_{tol})$

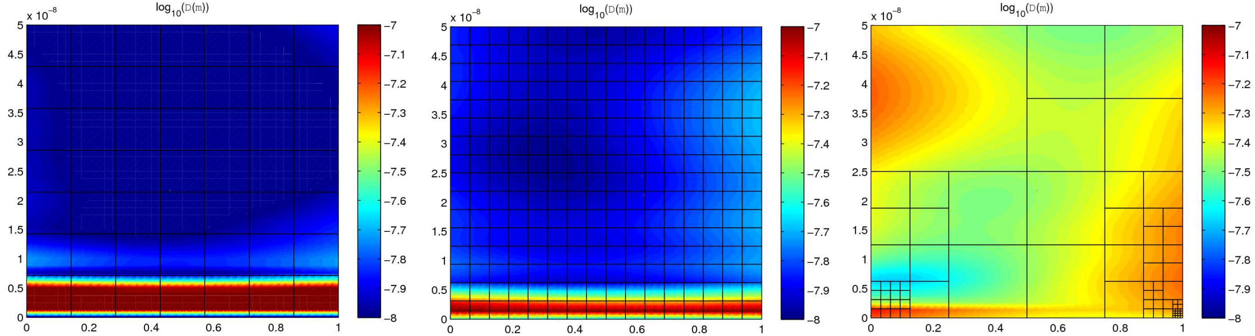
$\Phi := \Phi_0, \mathcal{M} := \mathcal{M}_0$
repeat
    $M_{train} := V(\mathcal{M})$
    $[\Phi, \varepsilon] := \text{ESGREEDY}(\Phi, M_{train}, \varepsilon_{tol},$
      $M_{val}, \rho_{tol})$
    if $\varepsilon > \varepsilon_{tol}$
      then
        $\boldsymbol{\eta} = \text{ELEMENTINDICATORS}(\mathcal{M}, \Phi, \varepsilon)$
        $\mathcal{M} := \text{MARK}(\mathcal{M}, \boldsymbol{\eta})$
        $\mathcal{M} := \text{REFINE}(\mathcal{M})$
  until $\varepsilon \leq \varepsilon_{tol}$
return Φ

Here, $\mathcal{M}$ denotes a Partition of the Parameter Space,
and $V(\mathcal{M})$ are the Vertices of $\mathcal{M}$.

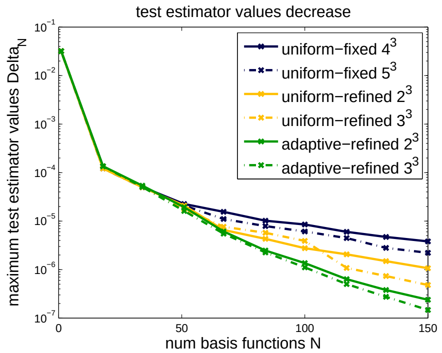
mario.ohlberger@uni-muenster.de
www.uni-muenster.de/math/num/ohlberger

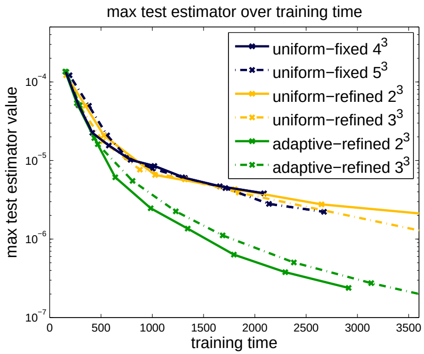
Improvement through Adaptive Basis Enrichment

Error Distribution for Uniform / Adaptive Training Sets



Exponential Convergence and CPU-Efficiency





mario.ohlberger@uni-muenster.de www.uni-muenster.de/math/num/ohlberger

Adaptive Parameter Domain Partition

[Dihlmann, Haasdonk, Ohlberger '10]

Idea: **Construct a reduced model with prescribed error tolerance and an upper bound for the dimension of the reduced space.**

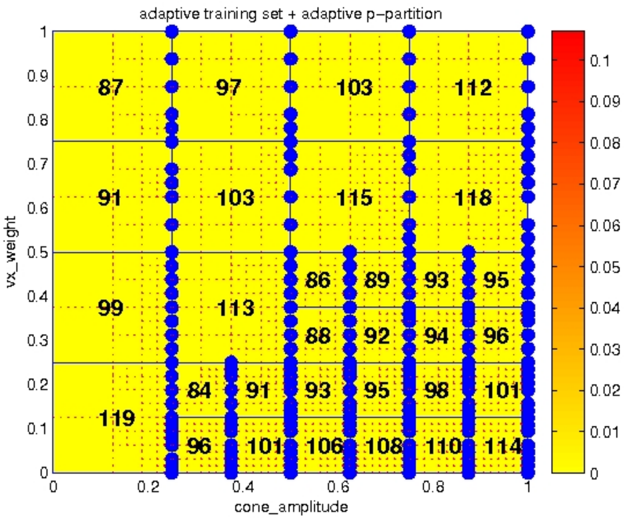
Ansatz: **Parameter domain partition and construction of independent reduced spaces in the parameter sub-domains.**

Adaptive Parameter Domain Partition

[Dihlmann, Haasdonk, Ohlberger '10]

Idea: **Construct a reduced model with prescribed error tolerance and an upper bound for the dimension of the reduced space.**

Ansatz: **Parameter domain partition and construction of independent reduced spaces in the parameter sub-domains.**

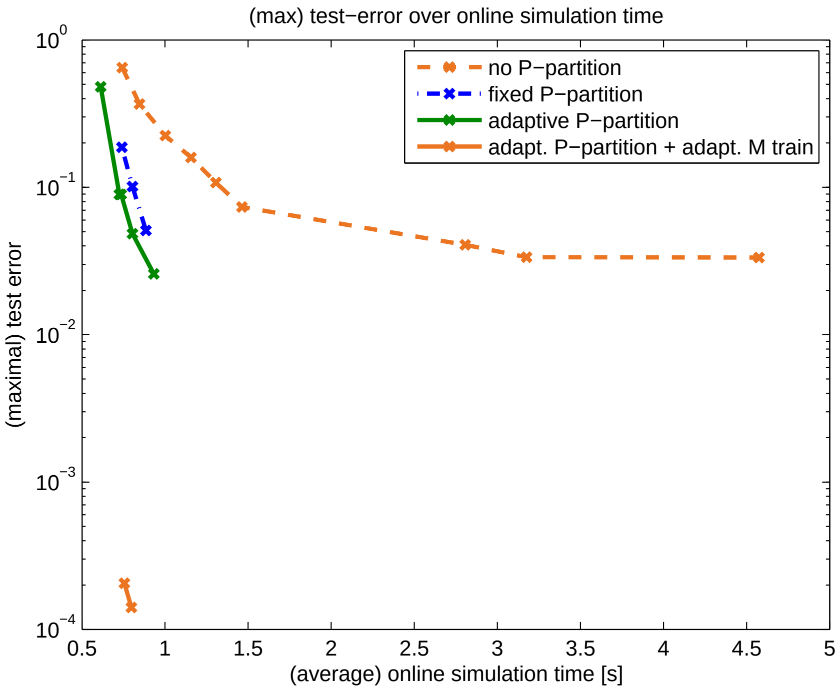


Adaptive Parameter Domain Partition

AdaptiveParamPartition($\mathcal{M}_0, \varepsilon_{tol}, N_{max}$)

$\mathcal{M} := \mathcal{M}_0, \Phi(e) := \emptyset$ **for** $e \in \mathcal{E}(\mathcal{M})$
repeat
    for $e \in \mathcal{E}(\mathcal{M})$ **with** $\Phi(e) = \emptyset$
    do $\Phi_0 :=$ InitBasis(e)
        $M_{train} :=$ MTrain(e)
        $\eta(e) := 0$
        $[\Phi(e), \varepsilon(e)] :=$ EarlyStoppingGreedy($\Phi, M_{train}, \varepsilon_{tol}, \emptyset, \infty, N_{max}$)
        if $\varepsilon(e) > \varepsilon_{tol}$
            then $\eta(e) := 1, \Phi(e) := \emptyset$
    $\eta_{max} := \max_{e \in \mathcal{E}(\mathcal{M})} \eta(e)$
    if $\eta_{max} > 0$
        then $\mathcal{M} :=$ Mark($\mathcal{M}, \boldsymbol{\eta}$)
            $\mathcal{M} :=$ Refine($\mathcal{M}$)
  until $\eta_{max} = 0$
return $\mathcal{M}, \{\Phi(e), \varepsilon(e)\}_{e \in \mathcal{E}(\mathcal{M})}$

Evaluation of the Online Efficiency



Extension to Nonlinear Explicit Operators

Reduced Basis Method for Explicit Finite Volume Approximations of Nonlinear Conservation Laws

[Haasdonk, Ohlberger '08]

A Simple Model Problem

$$\partial_t c(\mu) + \nabla \cdot (\mathbf{v}c(\mu)^{\mu}) = 0 \qquad \text{in } \Omega \times [0, T], \ \mu \in [1, 2]$$

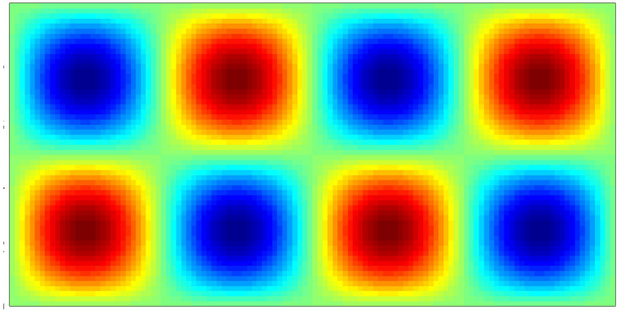
plus suitable Initial and Boundary Conditions.

$\mu = 1 \implies$ **Linear Transport**

$\mu = 2 \implies$ **Burgers Equation**

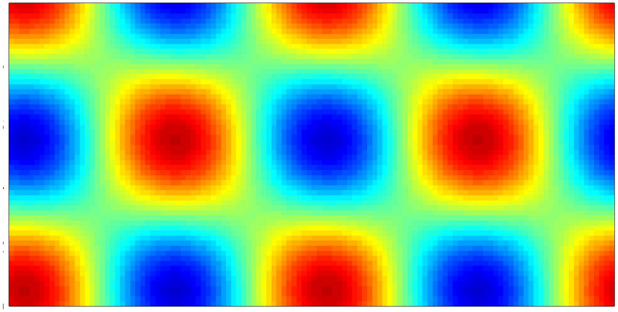
Numerical Results

Initial values: $c_0(x) = 1/2(1 + \sin(2\pi x_1)\sin(2\pi x_2))$

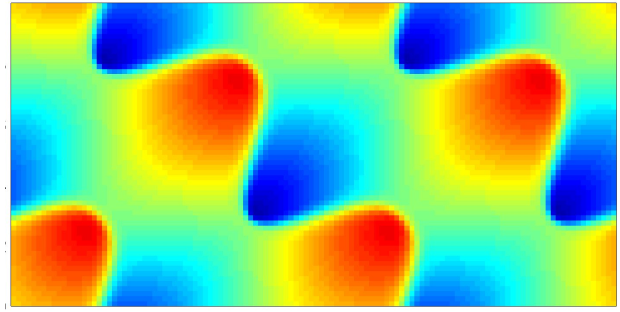


Solution at t = 0.3

Linear Transport



Burgers Equation



General Framework

Nonlinear Equation:

$$\partial_t c(\boldsymbol{\mu}) + L_{\boldsymbol{\mu}}[c(\boldsymbol{\mu})] = 0 \qquad \textbf{in } \Omega \times [0,T],$$

Explicit Discretization:

$$c_H^{k+1}(\boldsymbol{\mu}) = c_H^k(\boldsymbol{\mu}) - \Delta t L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})].$$

Problem: **Non-Affine Parameter Dependency**
Non-Linear Evolution Operator

General Framework

Nonlinear Equation:

$$\partial_t c(\boldsymbol{\mu}) + L_{\boldsymbol{\mu}}[c(\boldsymbol{\mu})] = 0 \qquad \textbf{in } \Omega \times [0,T],$$

Explicit Discretization:

$$c_H^{k+1}(\boldsymbol{\mu}) = c_H^k(\boldsymbol{\mu}) - \Delta t L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})].$$

Problem: **Non-Affine Parameter Dependency**
Non-Linear Evolution Operator

Idea: **Linear Affine Approximation through Empirical Interpolation**

$$L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})](x) \approx \sum_{m=1}^{M} y_m(c, \boldsymbol{\mu}, t^k)\xi_m(x)$$

General Framework

Nonlinear Equation:

$$\partial_t c(\boldsymbol{\mu}) + L_{\boldsymbol{\mu}}[c(\boldsymbol{\mu})] = 0 \qquad \textbf{in } \Omega \times [0,T],$$

Explicit Discretization:

$$c_H^{k+1}(\boldsymbol{\mu}) = c_H^k(\boldsymbol{\mu}) - \Delta t L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})].$$

Problem: **Non-Affine Parameter Dependency**
Non-Linear Evolution Operator

Idea: **Linear Affine Approximation through Empirical Interpolation**

$$L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})](x) \approx \sum_{m=1}^{M} y_m(c,\boldsymbol{\mu},t^k)\xi_m(x)$$

$$y_m(c,\boldsymbol{\mu},t^k) := L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})](x_m)$$

Empirical Interpolation of Localized Operators

Idea: Construct a **Collateral Reduced Basis Space W_M** that approximates the space spanned by $L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})]$

Empirical Interpolation of Localized Operators

Idea: Construct a Collateral Reduced Basis Space W_M that approximates the space spanned by $L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})]$

Ingredients: Collateral Reduced Basis Space:

$$W_M := \mathbf{span}\{L_H^{k_m}(\boldsymbol{\mu}_m)[c_H^{k_m}(\boldsymbol{\mu}_m)] | m = 1, \ldots, M\}$$

Empirical Interpolation of Localized Operators

Idea: Construct a **Collateral Reduced Basis Space W_M** that approximates the space spanned by $L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})]$

Ingredients:

Collateral Reduced Basis Space:

$$W_M := \textbf{span}\{L_H^{k_m}(\boldsymbol{\mu}_m)[c_H^{k_m}(\boldsymbol{\mu}_m)] | m = 1, \ldots, M\}$$

Nodal Collateral Reduced Basis:

$$\{\xi_m\}_{m=1}^M \Longrightarrow W_M = \textbf{span}\{\xi_m | m = 1, \ldots, M\}$$

Interpolation Points:

$$\{x_k\}_{k=1}^M \textbf{ with } \xi_m(x_k) = \delta_{mk}$$

Empirical Interpolation of Localized Operators

Idea: Construct a **Collateral Reduced Basis Space W_M** that approximates the space spanned by $L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})]$

Ingredients:

Collateral Reduced Basis Space:

$$W_M := \mathbf{span}\{L_H^{k_m}(\boldsymbol{\mu}_m)[c_H^{k_m}(\boldsymbol{\mu}_m)] | m = 1, \dots, M\}$$

Nodal Collateral Reduced Basis:

$$\{\xi_m\}_{m=1}^M \Longrightarrow W_M = \mathbf{span}\{\xi_m | m = 1, \dots, M\}$$

Interpolation Points:

$$\{x_k\}_{k=1}^M \text{ with } \xi_m(x_k) = \delta_{mk}$$

Empirical Interpolation:

$$\mathcal{I}_M[L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})]] := \sum_{m=1}^M y_m(c, \boldsymbol{\mu}, t^k)\xi_m(x)$$

mario.ohlberger@uni-muenster.de

Empirical Interpolation of Localized Operators

Idea: Construct a Collateral Reduced Basis Space $\mathbf{W_M}$ that approximates the space spanned by $L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})]$

Ingredients: **Collateral Reduced Basis Space:**

$$W_M := \mathbf{span}\{L_H^{k_m}(\boldsymbol{\mu}_m)[c_H^{k_m}(\boldsymbol{\mu}_m)] | m = 1, \ldots, M\}$$

Nodal Collateral Reduced Basis:

$$\{\xi_m\}_{m=1}^M \Longrightarrow W_M = \mathbf{span}\{\xi_m | m = 1, \ldots, M\}$$

Interpolation Points:

$$\{x_k\}_{k=1}^M \textbf{ with } \xi_m(x_k) = \delta_{mk}$$

Empirical Interpolation:

$$\mathcal{I}_M[L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})]] := \sum_{m=1}^M y_m(c, \boldsymbol{\mu}, t^k)\xi_m(x)$$

Offline: Collateral Basis $\{\xi_m\}_{m=1}^M$ and Interpolation Points $\{x_m\}_{m=1}^M$

Online: Calculate Coefficients $y_m = L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})](x_m)$

Empirical Interpolation of Localized Operators

Idea: Construct a **Collateral Reduced Basis Space W_M** that approximates the space spanned by $L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})]$

Ingredients: **Collateral Reduced Basis Space:**

$$W_M := \textbf{span}\{L_H^{k_m}(\boldsymbol{\mu}_m)[c_H^{k_m}(\boldsymbol{\mu}_m)] | m = 1, \ldots, M\}$$

Nodal Collateral Reduced Basis:

$$\{\xi_m\}_{m=1}^M \Longrightarrow W_M = \textbf{span}\{\xi_m | m = 1, \ldots, M\}$$

Interpolation Points:

$$\{x_k\}_{k=1}^M \textbf{ with } \xi_m(x_k) = \delta_{mk}$$

Empirical Interpolation:

$$\mathcal{I}_M[L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})]] := \sum_{m=1}^M y_m(c, \boldsymbol{\mu}, t^k)\xi_m(x)$$

Offline: Collateral Basis $\{\xi_m\}_{m=1}^M$ and Interpolation Points $\{x_m\}_{m=1}^M$

Online: Calculate Coefficients $y_m = L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})](x_m)$

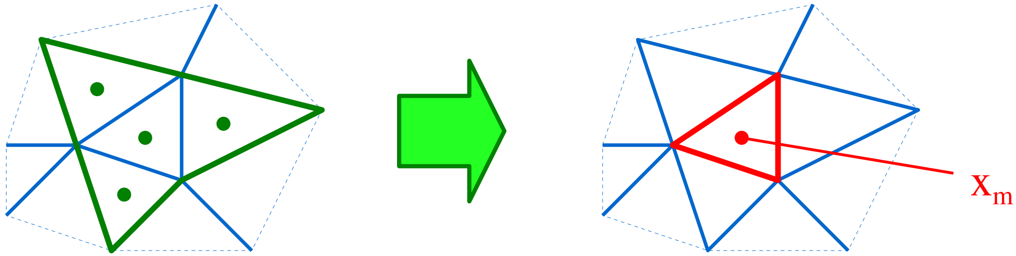
$\Longrightarrow$ **Localized operators for H-independent point evaluations**

Local Operator Evaluations and Reduced Basis Scheme

Local Operator Evaluations in the Online-Phase require:

1.) **Local reconstruction of c_N^k from coefficients a^k**

2.) **Local operator evaluation:** $y_m = L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})](x_m)$



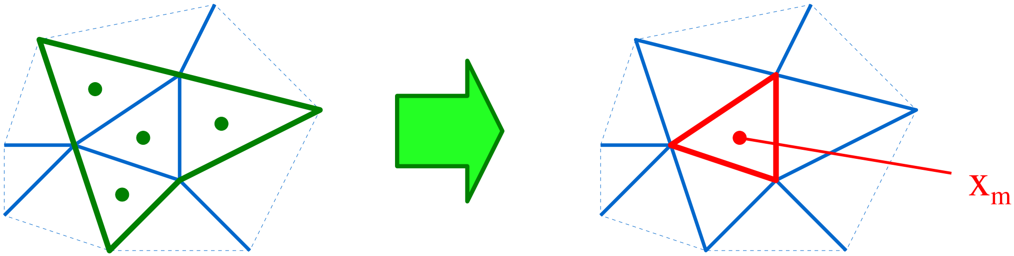
Requires Offline: **Numerical subgrids, local representation of** $\varphi_n \in W_N$

Local Operator Evaluations and Reduced Basis Scheme

Local Operator Evaluations in the Online-Phase require:

1.) **Local reconstruction of** c_N^k **from coefficients** a^k

2.) **Local operator evaluation:** $y_m = L_H^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})](x_m)$



Requires Offline: **Numerical subgrids, local representation of** $\varphi_n \in W_N$

RB Method: Galerkin projection of interpolated evolution scheme

$$\int_\Omega \left(c_N^{k+1}(\boldsymbol{\mu}) = c_N^k(\boldsymbol{\mu}) - \Delta t \mathcal{I}_M [L_H^k(\boldsymbol{\mu})[c_N^k(\boldsymbol{\mu})]] \right) \varphi, \quad \forall \varphi \in W_N.$$

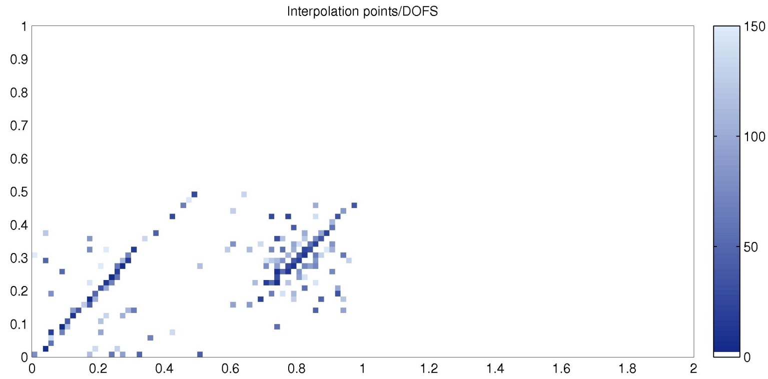
Offline-Online decomposition analog to the linear and affine case!!

Numerical Experiment

Empirical Interpolation:

$M_{\mathbf{max}} = 150$ **interpolation points**

Translation symmetry detected

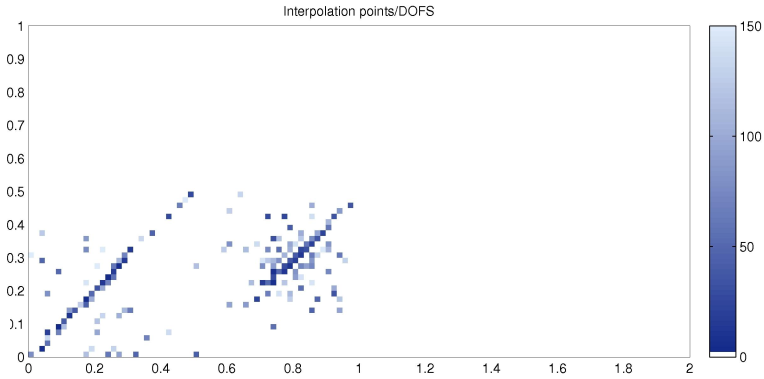


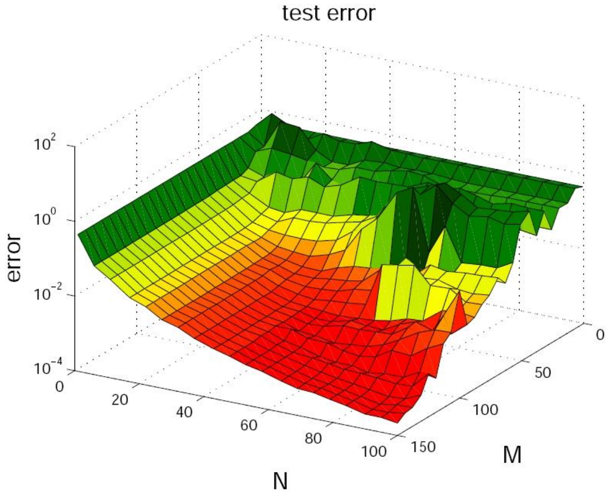
Numerical Experiment

Empirical Interpolation:

$M_{\mathbf{max}} = 150$ **interpolation points**

Translation symmetry detected





Test error convergence:

Exponential convergence for simultaneous increase of N and M

Numerical Experiment

Comparison of Online-Runtimes

Simulation	Dimension	Runtime $[s]$	Gain Factor
detailed	H = 7200	20.22	
reduced	N=20, M=30	0.91	22.2
reduced	N=40, M=60	1.22	16.6
reduced	N=60, M=90	1.55	13.0
reduced	N=80, M=120	1.77	11.4
reduced	N=100, M=150	2.06	9.8

Numerical Experiment

Comparison of Online-Runtimes

Simulation	Dimension	Runtime $[s]$	Gain Factor
detailed	H = 7200	20.22	
reduced	N=20, M=30	0.91	22.2
reduced	N=40, M=60	1.22	16.6
reduced	N=60, M=90	1.55	13.0
reduced	N=80, M=120	1.77	11.4
reduced	N=100, M=150	2.06	9.8

$<$ **Demonstration** $>$

Extension to Nonlinear Implicit Operators

with M. Drohmann and B. Haasdonk

$$\partial_t c(\boldsymbol{\mu}) + L_{\boldsymbol{\mu}}[c(\boldsymbol{\mu})] = 0 \qquad \textbf{in } \Omega \times [0,T],$$

Mixed Implicit - Explicit Discretization:

$$L_I^k(\boldsymbol{\mu})[c_H^{k+1}(\boldsymbol{\mu})] = L_E^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})].$$

Problem: **Non-Affine Parameter Dependency**
Non-Linear Evolution Operators
L_I^k **involves the solution of a non-linear System**

Extension to Nonlinear Implicit Operators

with M. Drohmann and B. Haasdonk

$$\partial_t c(\boldsymbol{\mu}) + L_{\boldsymbol{\mu}}[c(\boldsymbol{\mu})] = 0 \qquad \textbf{in } \Omega \times [0,T],$$

Mixed Implicit - Explicit Discretization:

$$L_I^k(\boldsymbol{\mu})[c_H^{k+1}(\boldsymbol{\mu})] = L_E^k(\boldsymbol{\mu})[c_H^k(\boldsymbol{\mu})].$$

Problem: Non-Affine Parameter Dependency
Non-Linear Evolution Operators
L_I^k involves the solution of a non-linear System

Ansatz: Newton's Method and
Empirical interpolation for the linearized defect equation

Newton's Method and Empirical Interpolation

Define the defect

$$d_H^{k+1,\nu+1} := c_H^{k+1,\nu+1} - c_H^{k+1,\nu}.$$

Solve in each Newton step ν for the defect

$$F_I^k(\boldsymbol{\mu})[c_H^{k+1,\nu}]d_H^{k+1,\nu+1} = -L_I^k(\boldsymbol{\mu})[c_H^{k+1,\nu}] + L_E^k(\boldsymbol{\mu})[c_H^k],$$

and update

$$c_H^{k+1,\nu+1} = c_H^{k+1,\nu} + d_H^{k+1,\nu+1}.$$

Here F_I^k is the Frechet derivative of L_I^k.

Problem: F_I^k **has Non-Affine Parameter Dependency**
L_I^k **and** L_E^k **can be treated as before!**

Empirical Interpolation for the Frechet Derivative

Starting point: Empirical interpolation for L_I^k

$$\mathcal{I}_M[L_I^k(\boldsymbol{\mu})[c_H]] = \sum_{m=1}^{M} y_m^I(c_H^k, \boldsymbol{\mu})\, \xi_m.$$

Empirical Interpolation for F_I^k

$$\mathcal{I}_M[F_I^k(\boldsymbol{\mu})[c_H]v_H] := \sum_{i=1}^{H}\sum_{m=1}^{M} \partial_i y_m^I(c_H^k, \boldsymbol{\mu}) v_i\, \xi_m \overset{!}{=} \sum_{i\in\tau}\sum_{m=1}^{M} \partial_i y_m^I(c_H^k, \boldsymbol{\mu}) v_i\, \xi_m.$$

Properties:

- Here $\tau \subset \{1, \dots, H\}$ is the smallest subset, such that the equality holds $\Longrightarrow$ card$(\tau) = \mathcal{O}(M)$, since L_I^k is supposed to be localized!
- $(v_i)_{i\in\tau}$ can be evaluated efficiently in case of a nodal basis of W_H.

Resulting Reduced Basis Formulation of one Newton Step

Ansatz: $c_N^{k,\nu}(x) = \sum_{n=1}^{N} a_n^{k,\nu}\phi_n(x)$, **($\mathbf{a}^{k,\nu}$ denotes the coefficient vector)**

$$G\; A[c_N^{k+1,\nu}]\underbrace{(\mathbf{a}^{k+1,\nu+1} - \mathbf{a}^{k+1,\nu})}_{=:\mathbf{d}^{\mathbf{k+1},\nu+1}} = RHS(\mathbf{a}^{k+1,\nu}, \mathbf{a}^k).$$

Thereby the matrices $A[c_N], G$ are given as

$$(A[c_N])_{m,n} := \sum_{i=1}^{M} \partial_i y_m^I(c_N, \boldsymbol{\mu})\varphi_n(x_i), \qquad G_{n,m} := \int_\Omega \xi_m \varphi_n$$

with a corresponding offline-online splitting.

A Posteriori Error Estimate

Definition: Residual of the approximated FV/DG Method

$$R^{k+1}(\boldsymbol{\mu})\left[c_N\right] = \mathcal{I}_M\left[L_I(\boldsymbol{\mu})\left[{c_N}^{k+1}\right]\right] - \mathcal{I}_M\left[L_E(\boldsymbol{\mu})\left[{c_N}^{k}\right]\right]$$

Theorem: A Posteriori Error Estimate in $L^\infty L^2$

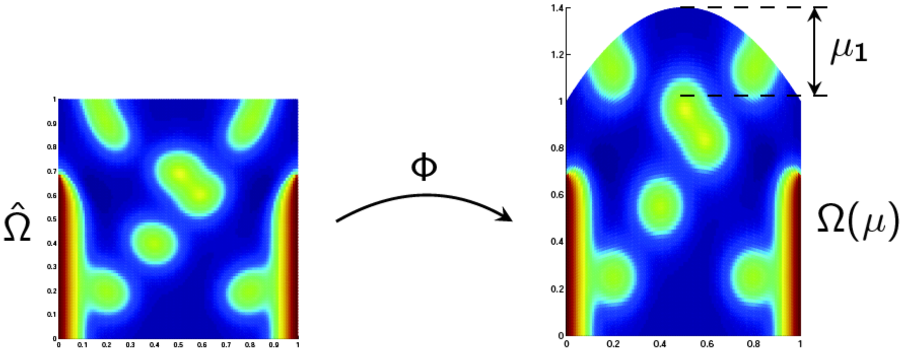
$$\left\|c_N^k(\boldsymbol{\mu}) - c_H^k(\boldsymbol{\mu})\right\|_{L^2(\Omega)} \leq \sum_{l=0}^{k-1} C_I^{k-l} C_E^{k-1-l} \left(\left\| \sum_{m=M}^{M+M'} \left(y_m^I\left({c_N}^{l+1}, \boldsymbol{\mu}\right) - y_m^E\left({c_N}^{l}, \boldsymbol{\mu}\right)\right)\xi_m \right\|_{L^2(\Omega)} + 2\varepsilon^{Newton} + \left\|R^{l+1}(\boldsymbol{\mu})\left[c_N\right]\right\|_{L^2(\Omega)} \right)$$

Reduced Basis Approximation for the Heat Equation on Parameterized Geometries

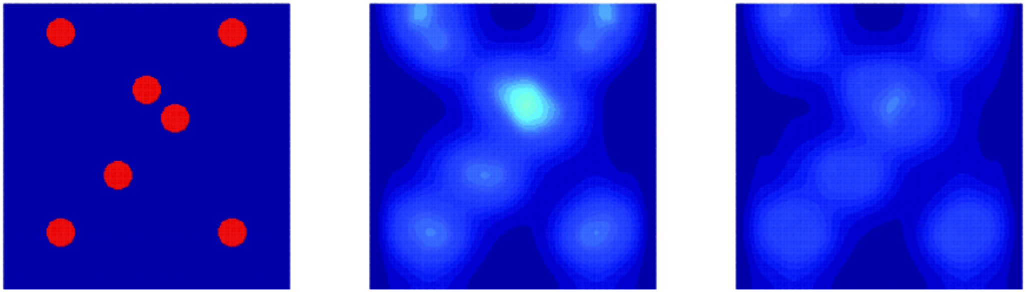
with M. Drohmann and B. Haasdonk

Transformed heat equation on reference domain $\hat{\Omega}$

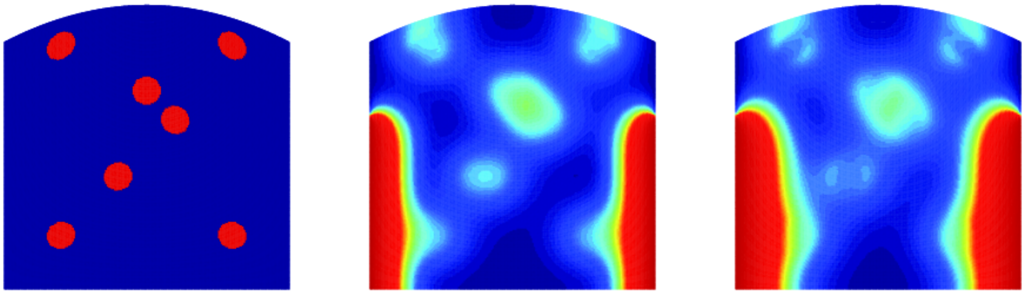
$$\partial_t \hat{u} + \kappa(\mu)\nabla \cdot (GG^\top \nabla \hat{u}) + \kappa(\mu)\nabla \cdot (v\hat{u}) + \kappa(\mu)\nabla \cdot v\hat{u} = f$$



Preliminary experimental results



Results for $\mu = (0,0)$ and $t = 0, t = 0.75$, and $t = 1.5$.



Results for $\mu = (0.2, 0.2)$ and $t = 0, t = 0.75$, and $t = 1.5$.

Preliminary experimental results

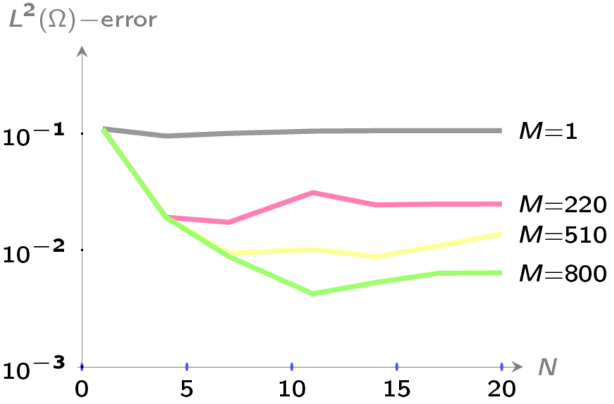


Fig. L^2 convergence

Dimension	CPU time
H = 40.000	24.37
N = 7, M= 267	1.22
N = 7, M= 800	2.05
N = 14, M= 267	1.25
N = 14, M= 800	2.10
N = 20, M= 267	1.27
N = 20, M= 800	2.11

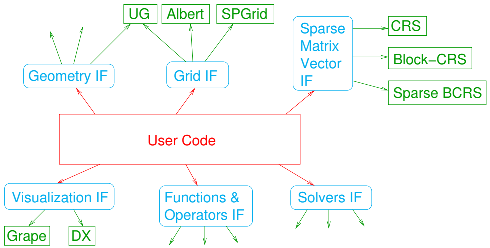
Tab. Computing times

Gain factor: $\approx$ 10

Further work on reduced basis techniques

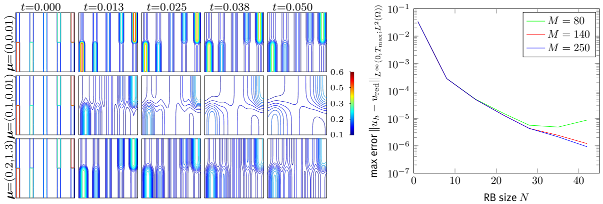
Coupling with DUNE

[www.dune-project.org]

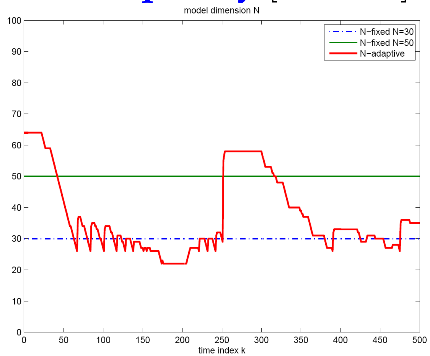


Non-linear evolution equations

[DHO ’10]

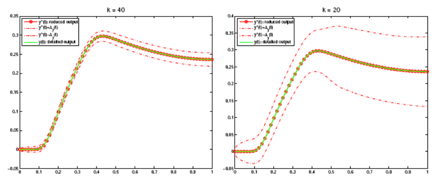


N-Adaptivity [HO ’09]



Reduced basis medthods for dynamical systems [H0 ’09]

$$\frac{d}{dt}x(t) = \mathbf{A}(t,\boldsymbol{\mu})x(t) + \mathbf{B}(t,\boldsymbol{\mu})u(t)$$
$$y(t) = \mathbf{C}(t,\boldsymbol{\mu})x(t) + \mathbf{D}(t,\boldsymbol{\mu})u(t)$$



RB with output estimation

Thank you for your attention!

www.uni-muenster.de/math/num/ohlberger

Software: **DUNE ALUGrid GRAPE RBmatlab**