Westfälische
Wilhelms-Universität
Münster

Model reduction for multiscale problems

Mario Ohlberger

wissen.leben
WWU Münster

Dec. 12-16, 2011

RICAM, Linz

› Outline

Motivation: Multi-Scale and Multi-Physics Problems

Model Reduction: The Reduced Basis Approach

A new Reduced Basis DG Multiscale Method

› Outline

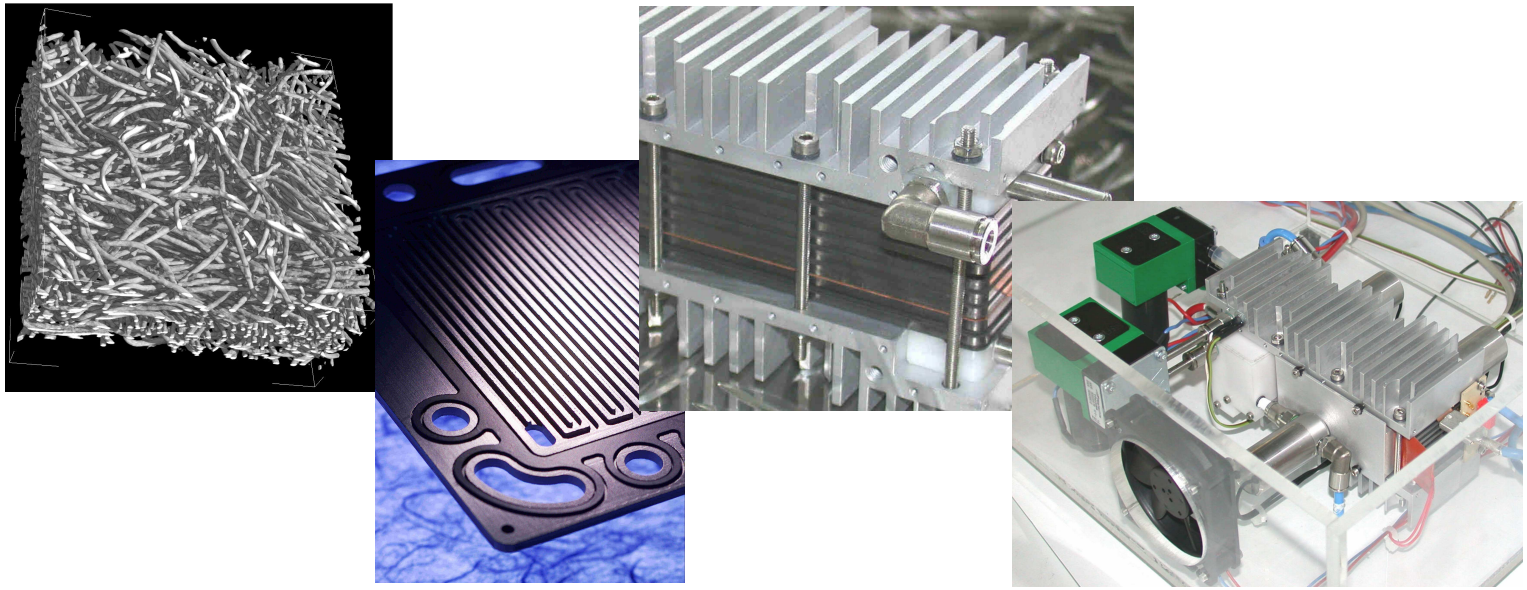
Motivation: Multi-Scale and Multi-Physics Problems

Model Reduction: The Reduced Basis Approach

A new Reduced Basis DG Multiscale Method

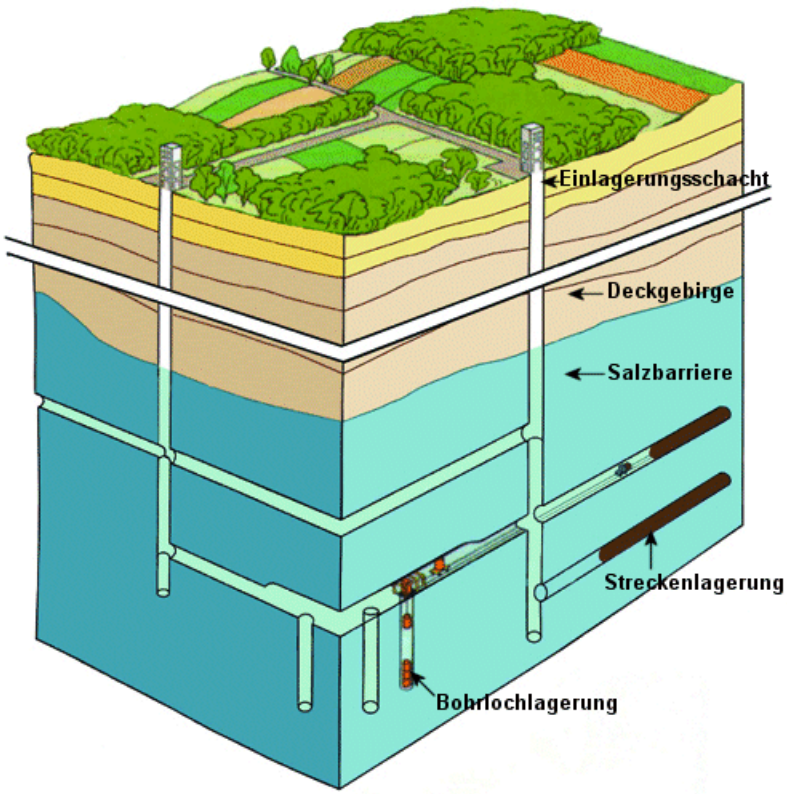
› Example: PEM fuel cells

Pore Cell Stack System

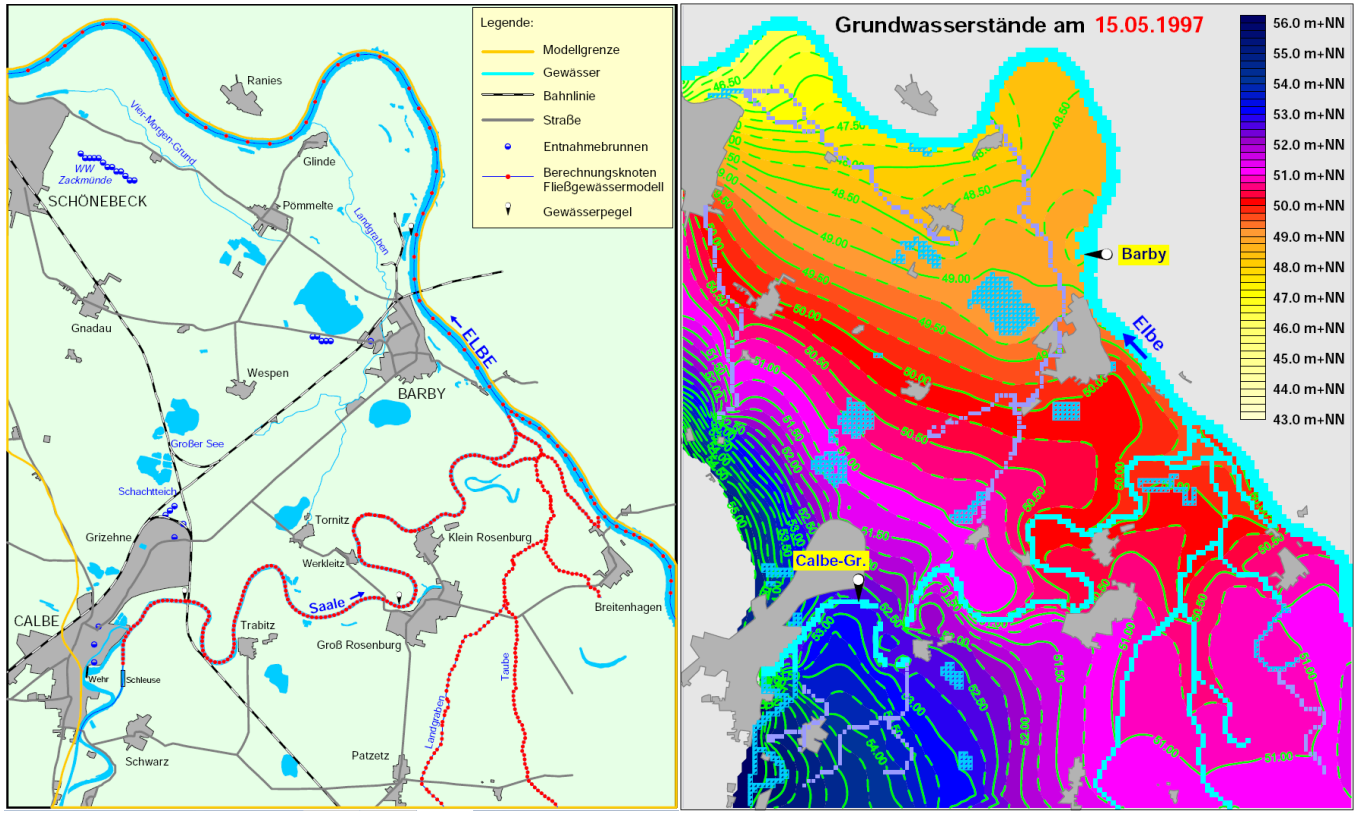


[BMBF-Project PEMDesign: Fraunhofer ITWM and Fraunhofer ISE]

› Security behavior of nuclear waste disposals



› Example: Hydrological Modeling



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

› Mathematical Modelling and Model Reduction

Increasing Efficiency →

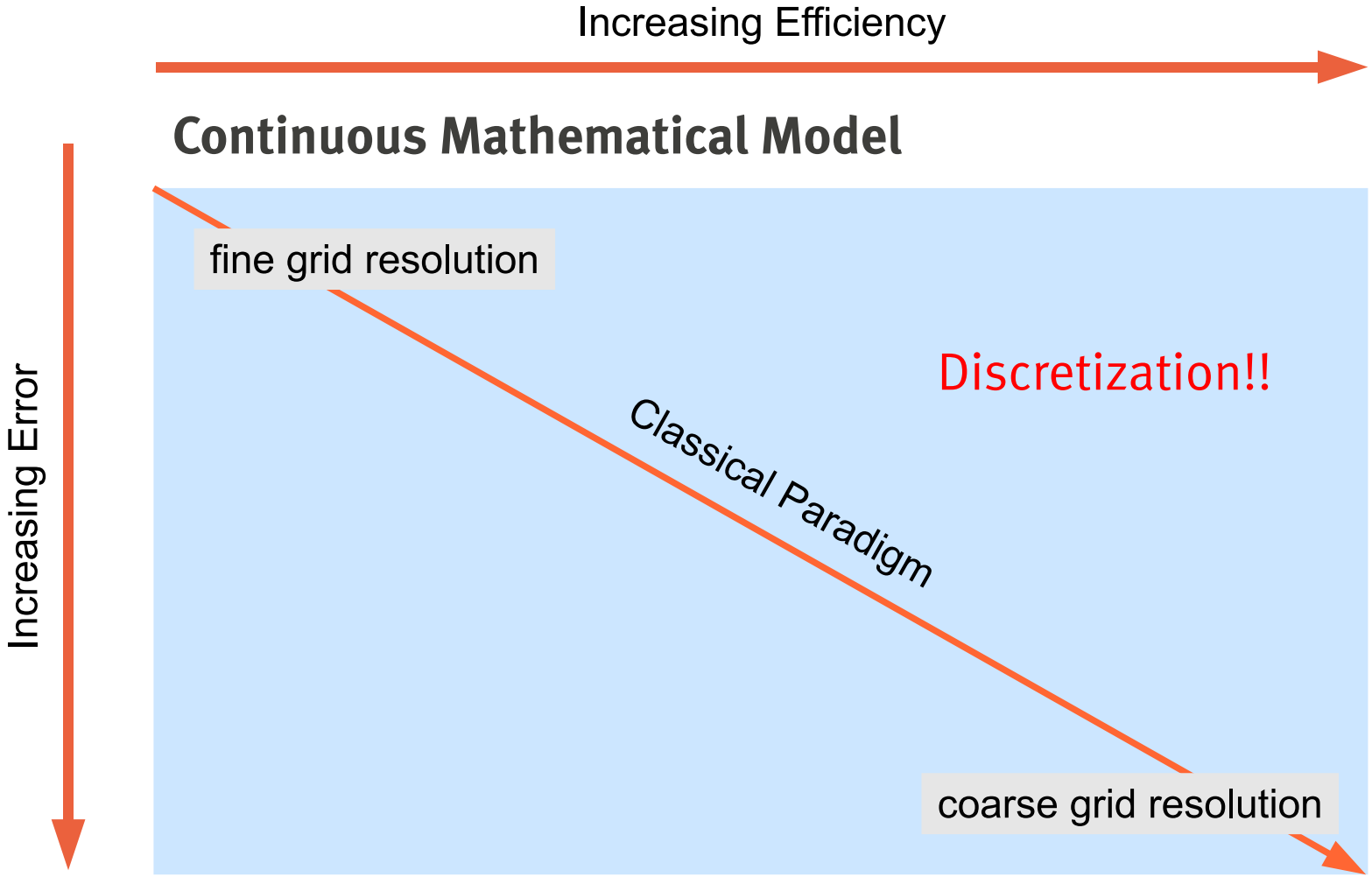
Increasing Error ↓

Real World Problem

Continuous Mathematical Model

- Here: system of partial differential equations
- Problem: infinite dimensional solution space
- no solutions in closed form

› Mathematical Modelling and Model Reduction



› Mathematical Modelling and Model Reduction

Increasing Efficiency →

Increasing Error ↓

Continuous Mathematical Model

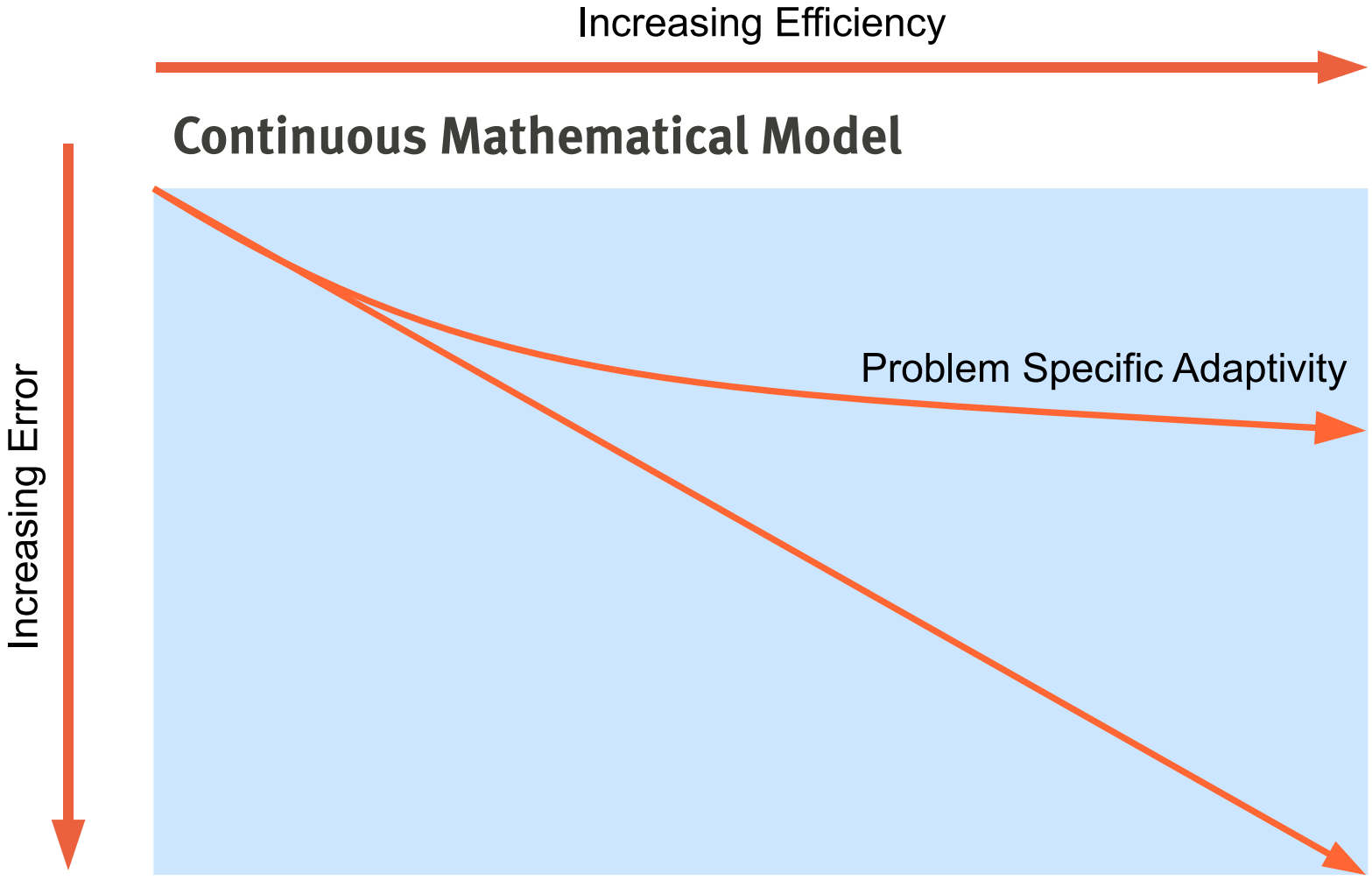
Discrete model on uniform grid (FEM, FV, DG, ...)

- Typical error estimates:

$$||u - u_h|| \leq c \inf_{v_h \in X_h} ||u - v_h||$$

- Error related to approximation property of X_h
- $\Longrightarrow$ Very general approach, but in particular cases not very efficient!!

› Mathematical Modelling and Model Reduction



› Mathematical Modelling and Model Reduction

Increasing Efficiency →

Increasing Error ↓

Continuous Mathematical Model

Problem specific: Adaptive Mesh Refinement

- Typical error estimates:

$$||u - u_h|| \leq c\, \eta(u_h)$$

- Error related to approximate solution!
- $\Longrightarrow$ Construct optimal mesh!
- Problem: Grid construction for every solve! Resulting system is still high-dimensional!

› Error Control and Adaptivity for HMM

HMM for linear elliptic homogenization problems

[Ohlberger: Multiscale Model. Simul., 2005]
[Henning, Ohlberger: Numer. Math., 2009]

HMM for multi-scale transport with large expected drift

[Henning, Ohlberger: Netw. Heterog. Media. 2010]
[Henning, Ohlberger: J. Anal. Appl. 2011]

HMM for nonlinear monotone elliptic problems

[Henning, Ohlberger 2011]

⟹ *see poster (8) at this workshop*

› Mathematical Modelling and Model Reduction

Increasing Efficiency

Increasing Error

Continuous Mathematical Model

Problem class specific: Reduced Basis Method

- Typical error estimates:

$$||(u - u_N)(\boldsymbol{\mu})|| \leq c\, \eta(u_N(\boldsymbol{\mu}))$$

- Error related to reduced solution!
- $\Longrightarrow$ Construct optimal reduced space for problem class!!
  Resulting system is low dimensional!

› Outline

Motivation: Multi-Scale and Multi-Physics Problems

Model Reduction: The Reduced Basis Approach

A new Reduced Basis DG Multiscale Method

› Reduced Basis Method for Evolution Equations

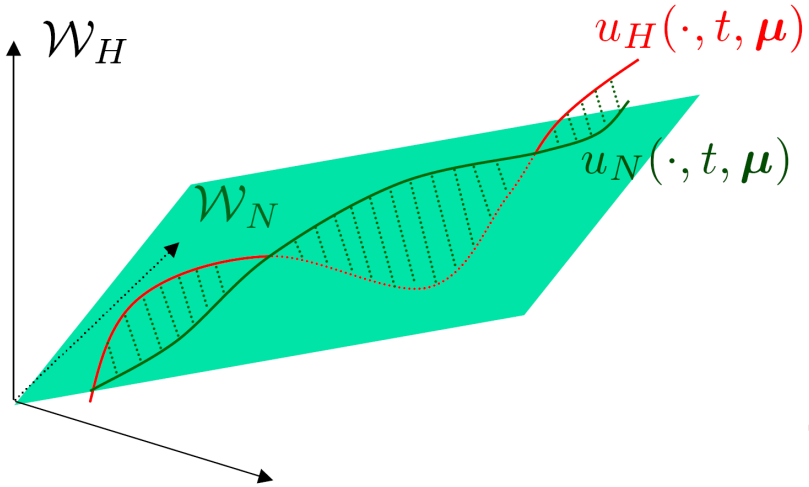
Goal: Fast "Online"-Simulation of Complex Evolution Systems for

- Parameter/Design Optimization
- Optimal Control
- Integration into System Simulation
- Uncertainty Quantification

Ansatz:

- Reduced Basis Method (RB)

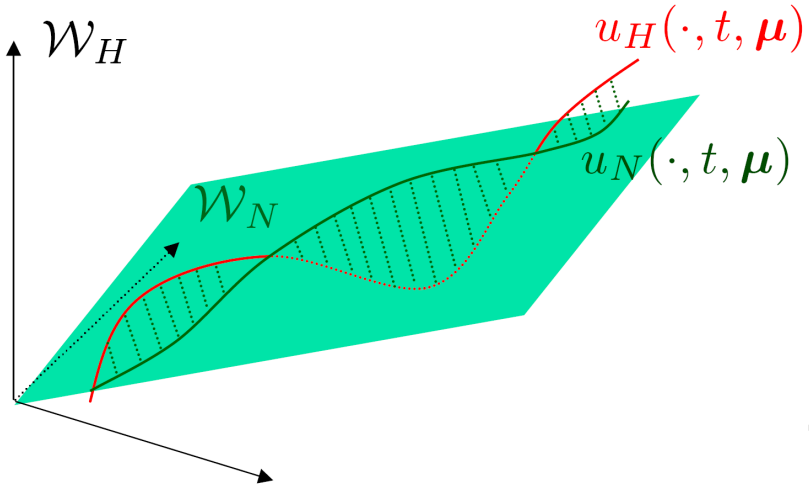
$$\dim(W_N) \ll \dim(W_H)\,!$$



› Reduced Basis Method for Evolution Equations

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

- Parameter/Design Optimization
- Optimal Control
- Integration into System Simulation
- Uncertainty Quantification



Ansatz:

- Reduced Basis Method (RB)

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

Classical 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, http://morepas.org

› Model Reduction: Reduced Basis Method

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

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

plus suitable Initial and Boundary Conditions.

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

› Model Reduction: Reduced Basis Method

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

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

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

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

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 $c_N(\boldsymbol{\mu}) \in W_N$.

Requirement:
- Efficient Choice of W_N (Exponential Convergence in N)
- Offline–Online Decomposition for all Calculations
- Error Control for $||c_H(\boldsymbol{\mu}) - c_N(\boldsymbol{\mu})||$

› Model Reduction: Reduced Basis Method

Assumption: FV/DG Scheme for Evolution Equations

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

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

› Model Reduction: Reduced Basis Method

Assumption: FV/DG Scheme for Evolution Equations

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

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

RB Method: Let $W_N \subset W_H$ be given, $\{\varphi_1, \ldots, \varphi_N\}$ a ONB of W_N.

Sought: $c_N^k(\boldsymbol{\mu}) = \sum_{n=1}^{N} \mathbf{a}_n^k(\boldsymbol{\mu})\varphi_n$ with $\mathbf{L}_I^k(\boldsymbol{\mu})\mathbf{a}^{k+1} = \mathbf{L}_E^k(\boldsymbol{\mu})\mathbf{a}^k + \mathbf{b}^k(\boldsymbol{\mu})$

where

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

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

› 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

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

› 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

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

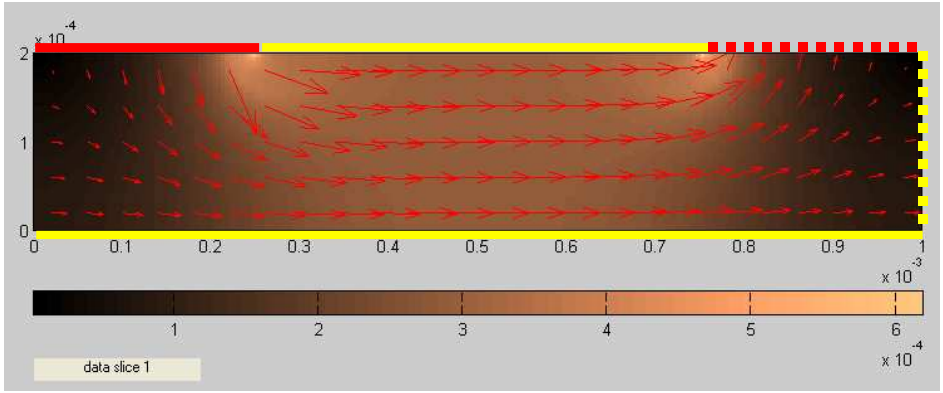
$\Longrightarrow$ Precompute offline: $(L_l^{k,q})_{nm} := \int_\Omega \varphi_n L_l^{k,q}[\varphi_m]$

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

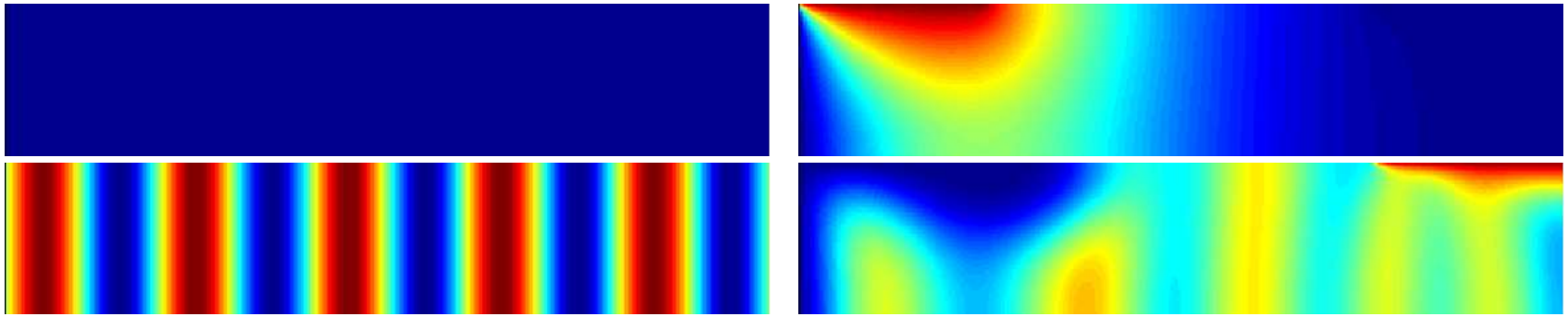
› Example: Convection-Diffusion Problem

Parameter:

- Initial Data
- Boundary Values
- Diffusion Parameter



Possible Variations of the Solution:



› Numerical Experiment

CPU-Time Comparison for a 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**	

› A Posteriori Error Estimates [Haasdonk, Ohlberger '08]

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

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

› A Posteriori Error Estimates [Haasdonk, Ohlberger '08]

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

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

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

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

› Efficient Choice of W_N: POD-Greedy [Haasdonk, O. '08]

General Idea: • Construct W_N from snapshots $c_H^l(\mu)$.

› Efficient Choice of W_N: POD-Greedy [Haasdonk, O. '08]

General Idea:
- Construct W_N from snapshots $c_H^l(\mu)$.

POD-Greedy:
- Use a Greedy algorithm based on the error estimator on a training set for an efficient parameter choice.
- Reduce time trajectory for the selected parameter with a Principal Orthogonal Decomposition (POD).

› Efficient Choice of W_N: POD-Greedy [Haasdonk, O. '08]

General Idea: • Construct W_N from snapshots $c_H^l(\mu)$.

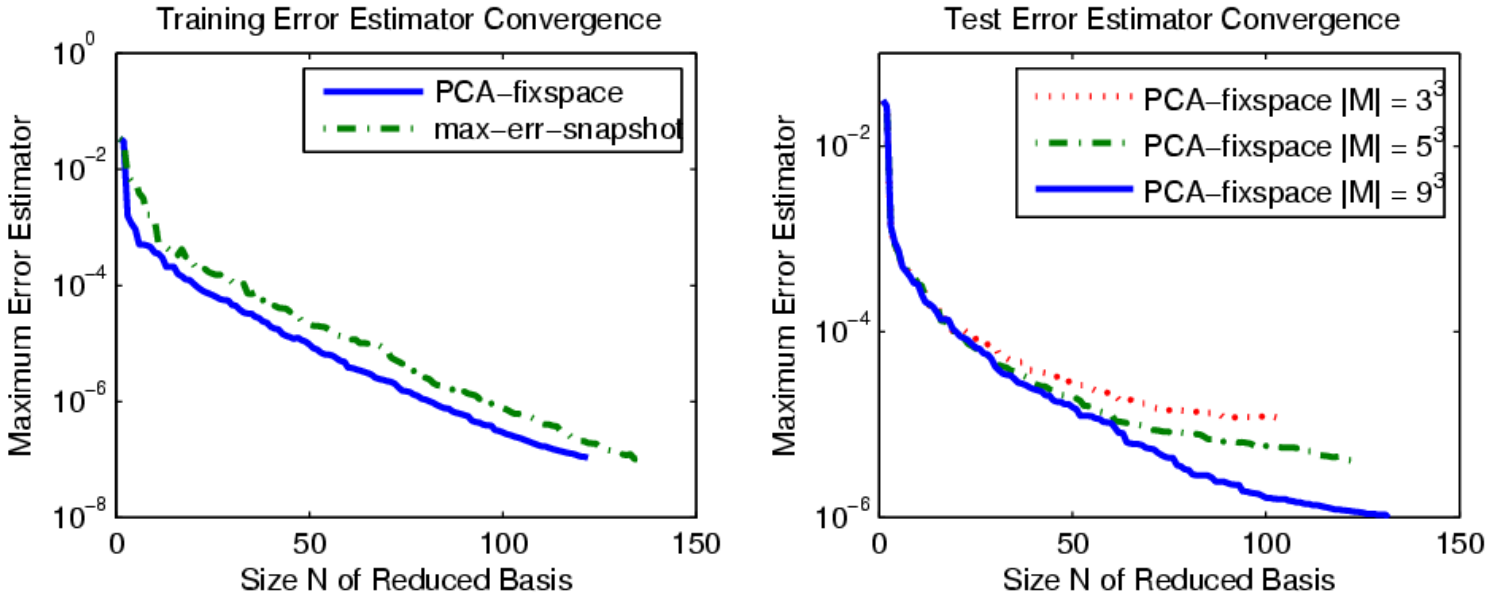
POD-Greedy: • Use a Greedy algorithm based on the error estimator on a training set for an efficient parameter choice.

• Reduce time trajectory for the selected parameter with a Principal Orthogonal Decomposition (POD).

Goal: Exponential Convergence in N !?

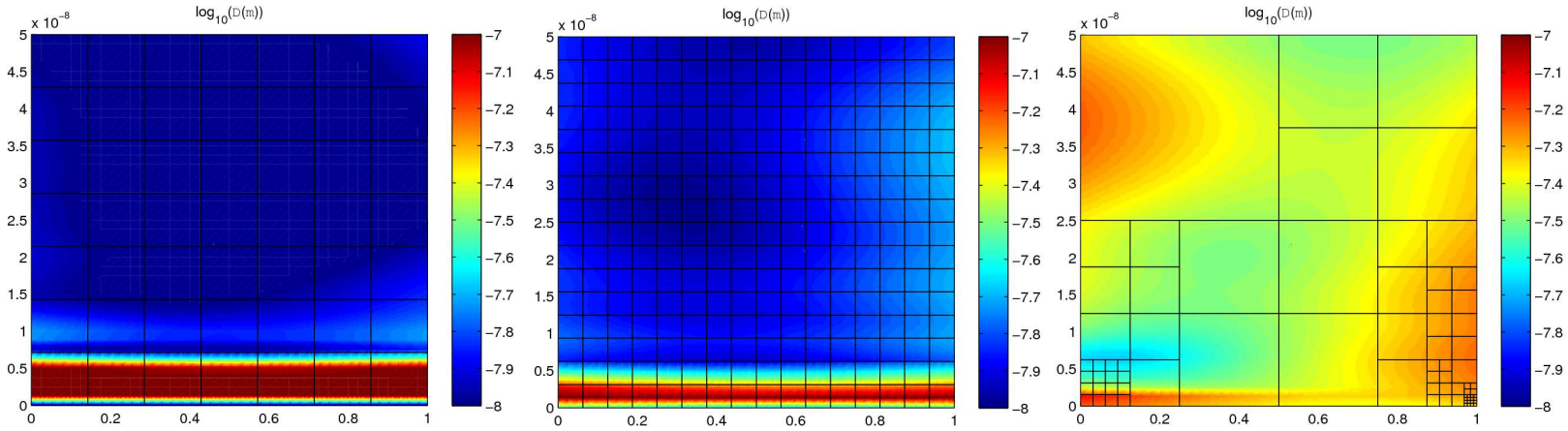
› Efficient Choice of W_N: POD-Greedy [Haasdonk, O. '08]

Preliminary result: convergence in N for fixed training and test sets

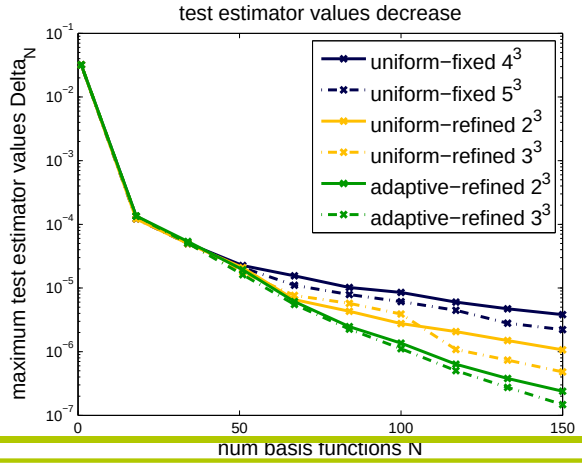


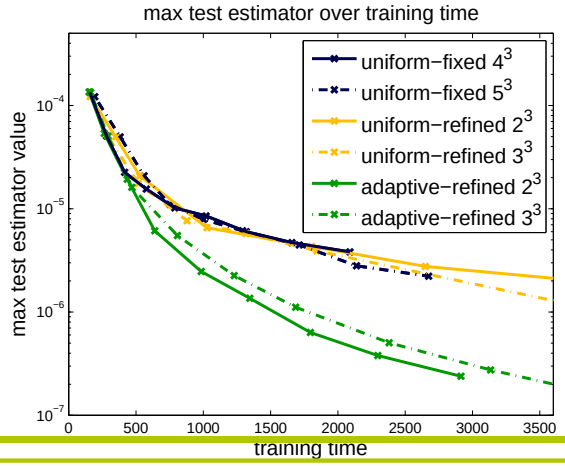
› Adaptive Basis Enrichment [Haasdonk, Ohlberger '08]

Error Distribution for Uniform / Adaptive Training Sets



Exponential Convergence and CPU-Efficiency





› Efficient Choice of W_N: POD-Greedy

Theorem (Haasdonk 2011)

If the Kolmogorov n-width of the compact set of time trajectories decays algebraically (exponentially), then also the POD-Greedy approximation error decays algebraically (exponentially).

The proof extends the arguments from the pure Greedy case presented in [Binev et al. 2010].

› How to treat nonlinear problems?

Current approaches

- Polynomial nonlinearity: Use multi-linear approach
  –› higher order reduced tensors
  [Rozza 05, Jung et al. 09, Nguyen et al. '09]

› How to treat nonlinear problems?

Current approaches

- Polynomial nonlinearity: Use multi-linear approach
  –› higher order reduced tensors
  [Rozza 05, Jung et al. 09, Nguyen et al. '09]
- Non-affine parameter dependence: Use classical empirical interpolation of functions
  [Barrault et al. '04, Grepl et al. '07, Canuto et al. '09]

› How to treat nonlinear problems?

Current approaches

- Polynomial nonlinearity: Use multi-linear approach
  –› higher order reduced tensors
  [Rozza 05, Jung et al. 09, Nguyen et al. '09]
- Non-affine parameter dependence: Use classical empirical interpolation of functions
  [Barrault et al. '04, Grepl et al. '07, Canuto et al. '09]
- Question: How to deal with general nonlinear problems?
  -› Discrete Empirical Interpolation [Chaturantabut, Sorensen '10]
  -› Empirical Operator Interpolation
  [Haasdonk et al. '08, Drohmann et al. '10]

› How to treat nonlinear problems?

Current approaches

- Polynomial nonlinearity: Use multi-linear approach
  –› higher order reduced tensors
  [Rozza 05, Jung et al. 09, Nguyen et al. '09]
- Non-affine parameter dependence: Use classical empirical interpolation of functions
  [Barrault et al. '04, Grepl et al. '07, Canuto et al. '09]
- Question: How to deal with general nonlinear problems?
  -› Discrete Empirical Interpolation [Chaturantabut, Sorensen '10]
  -› Empirical Operator Interpolation
  [Haasdonk et al. '08, Drohmann et al. '10]

› Empirical Interpolation of Explicit Operators

Reduced Basis Method for Explicit Finite Volume Approximations of Nonlinear Conservation Laws
[Haasdonk, Ohlberger, Rozza '08], [Haasdonk, Ohlberger '09]

A Simple Model Problem

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

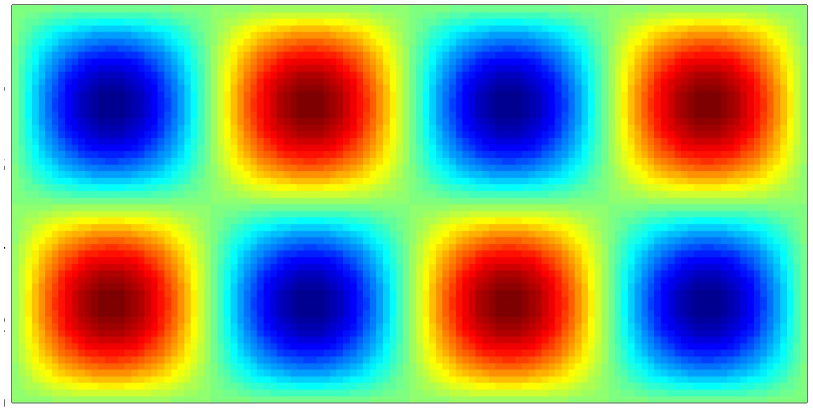
plus suitable Initial and Boundary Conditions.

$\mu = 1 \implies$ Linear Transport

$\mu = 2 \implies$ Burgers Equation

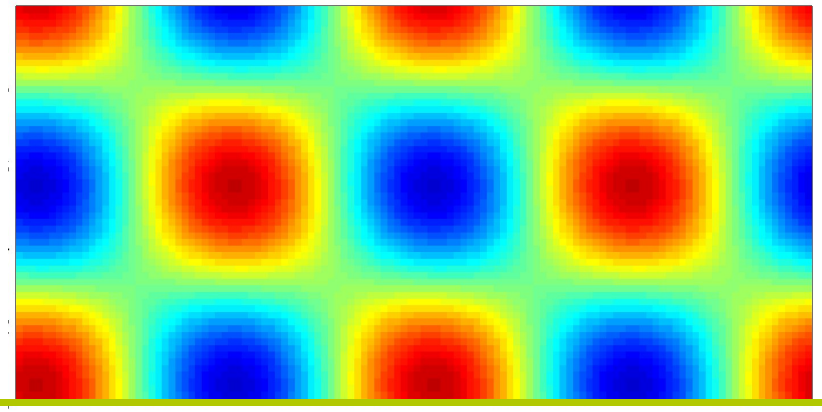
› Numerical Results

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

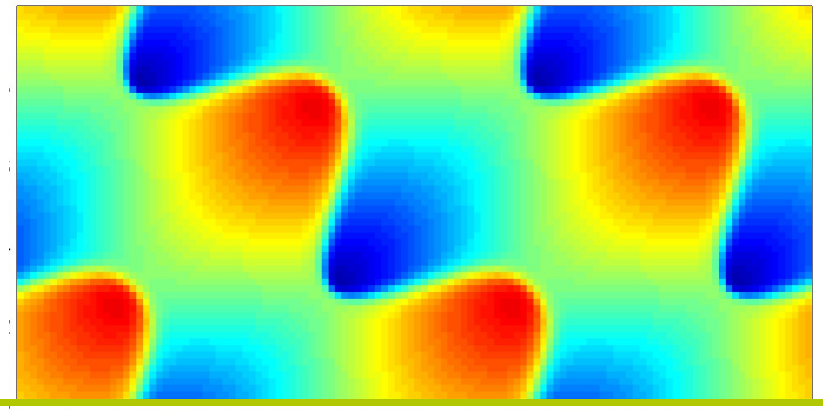


Solution at t = 0.3

Linear Transport



Burgers Equation



› General Framework

Nonlinear Equation

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

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

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

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:

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

› 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:

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

Nodal Collateral Reduced Basis:

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

Interpolation Points:

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

› 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:

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

Nodal Collateral Reduced Basis:

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

Interpolation Points:

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

Empirical Interpolation:

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

› 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:

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

Nodal Collateral Reduced Basis:

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

Interpolation Points:

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

Empirical Interpolation:

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

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:

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

Nodal Collateral Reduced Basis:

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

Interpolation Points:

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

Empirical Interpolation:

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

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

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

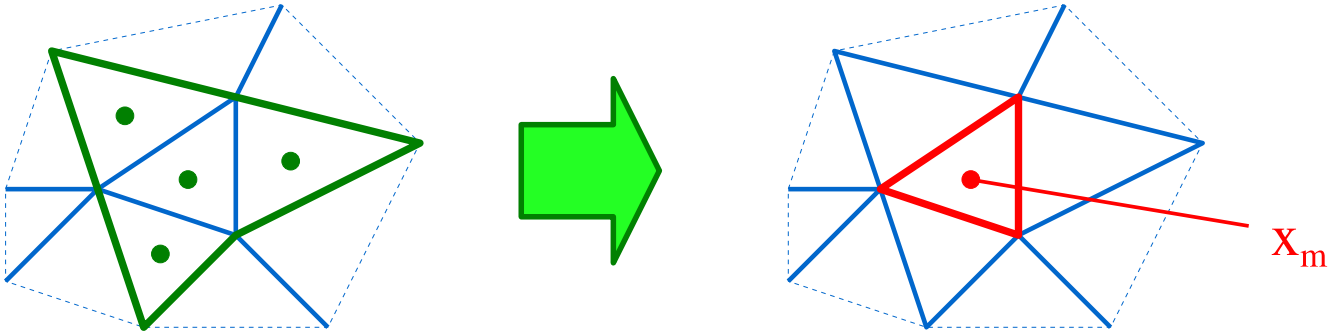
$\Longrightarrow$ Localized operators for H-independent point evaluations

› Local Operator Evaluations and RB Scheme

Local Operator Evaluations in the Online-Phase require:

1.) Local reconstruction of c_N^k from coefficients $\mathbf{a}^k$

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



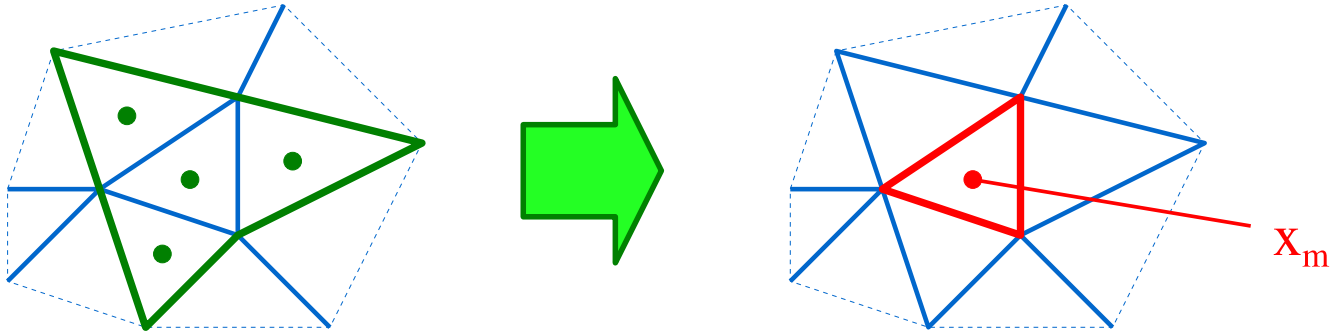
Requires Offline: Numerical subgrids, local basis representation

› Local Operator Evaluations and RB Scheme

Local Operator Evaluations in the Online-Phase require:

1.) Local reconstruction of c_N^k from coefficients $\mathbf{a}^k$

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



Requires Offline: Numerical subgrids, local basis representation

RB Method: Galerkin projection of interpolated scheme

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

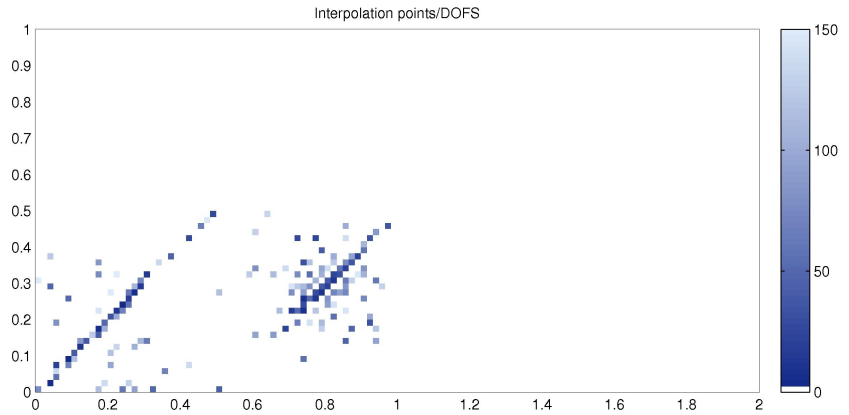
Offline-Online decomposition analog to the linear and affine case!!

› Numerical Experiment

Empirical Interpolation:

$M_{\max} = 150$ interpolation points

Translation symmetry detected

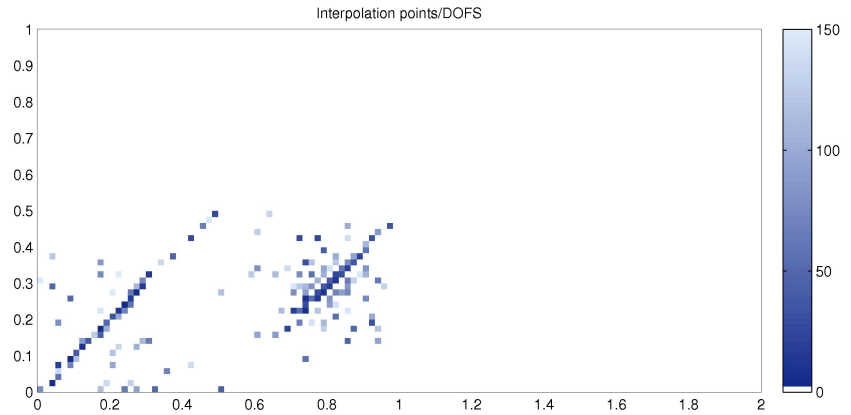


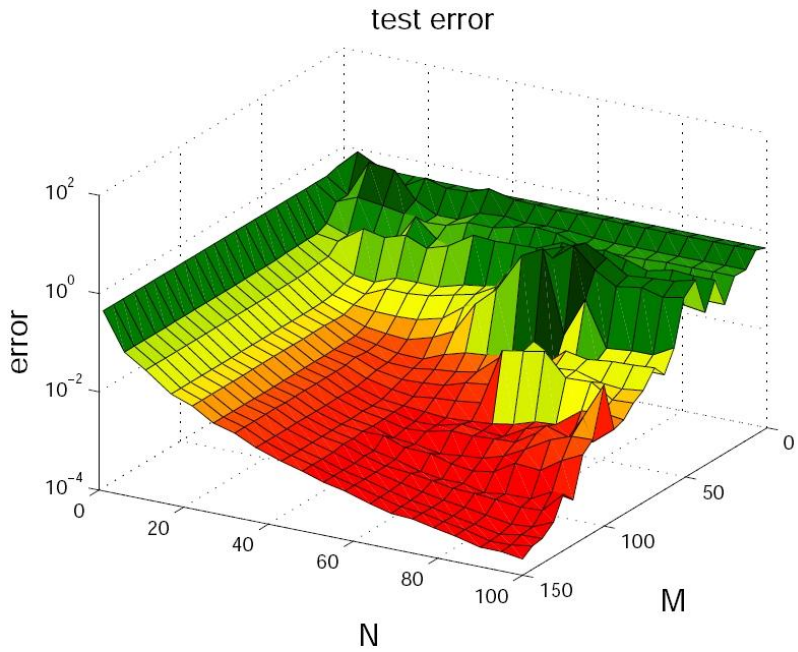
› Numerical Experiment

Empirical Interpolation:

$M_{\max} = 150$ 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**

› Extension to Nonlinear Implicit Operators

[Drohmann, Haasdonk, Ohlberger 2010]

Evolution Equation

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

Mixed Implicit - Explicit Discretization

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

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

[Drohmann, Haasdonk, Ohlberger 2010]

Evolution Equation

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

Mixed Implicit - Explicit Discretization

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

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

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

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

Empirical interpolation for L_l^k

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

Empirical Interpolation for F_l^k

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

Properties:

- $\tau \subset \{1, \dots, H\}$ is the smallest subset, such that equality holds
  $\Longrightarrow \operatorname{card}(\tau) = \mathcal{O}(M)$, since L_l^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 RB Formulation of one Newton Step

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

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

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

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

with a corresponding offline-online splitting.

› A Posteriori Error Estimate

Definition: Residual of the approximated FV/DG Method

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

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

$$\left\|c_N^k(\boldsymbol{\mu}) - c_H^k(\boldsymbol{\mu})\right\|_{L^2(\Omega)} \leq \sum_{i=0}^{k-1} C_I^{k-i+1} C_E^{k-1}\left(\left\|\sum_{m=M}^{M+M'} \Delta t\left(y_m^I\left({c_N}^{i+1}, \boldsymbol{\mu}\right) - y_m^E\left({c_N}^{i}, \boldsymbol{\mu}\right)\right)\xi_m\right\|_{L^2(\Omega)} + \varepsilon^{New} + \left\|R^{l+1}(\boldsymbol{\mu})\left[c_N\right]\right\|_{L^2(\Omega)}\right)$$

› Numerical Experiments

Model Problem: Porous Medium Equation

$$\partial_t c(\boldsymbol{\mu}) + \boldsymbol{\mu}_2 \Delta(c^{\boldsymbol{\mu}_1}(\boldsymbol{\mu})) = 0 \qquad \text{in } \Omega \times [0, T],\ \boldsymbol{\mu} \in [1, 5] \times [0, 0.001] \times [0, 0.2]$$

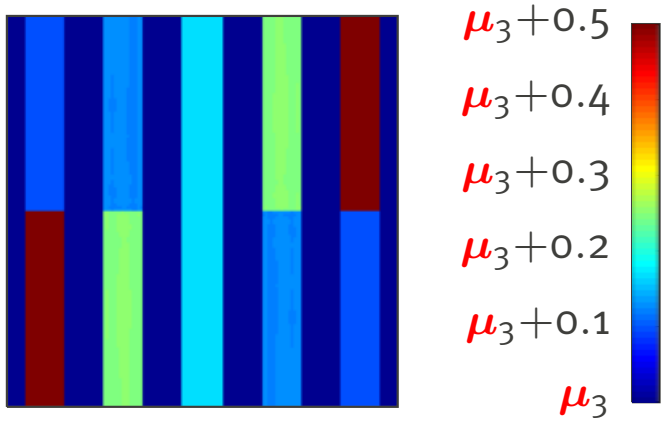
plus suitable initial and boundary conditions.

Nonlinearity:

$\boldsymbol{\mu}_1 > 2 \implies$ adiabatic flow

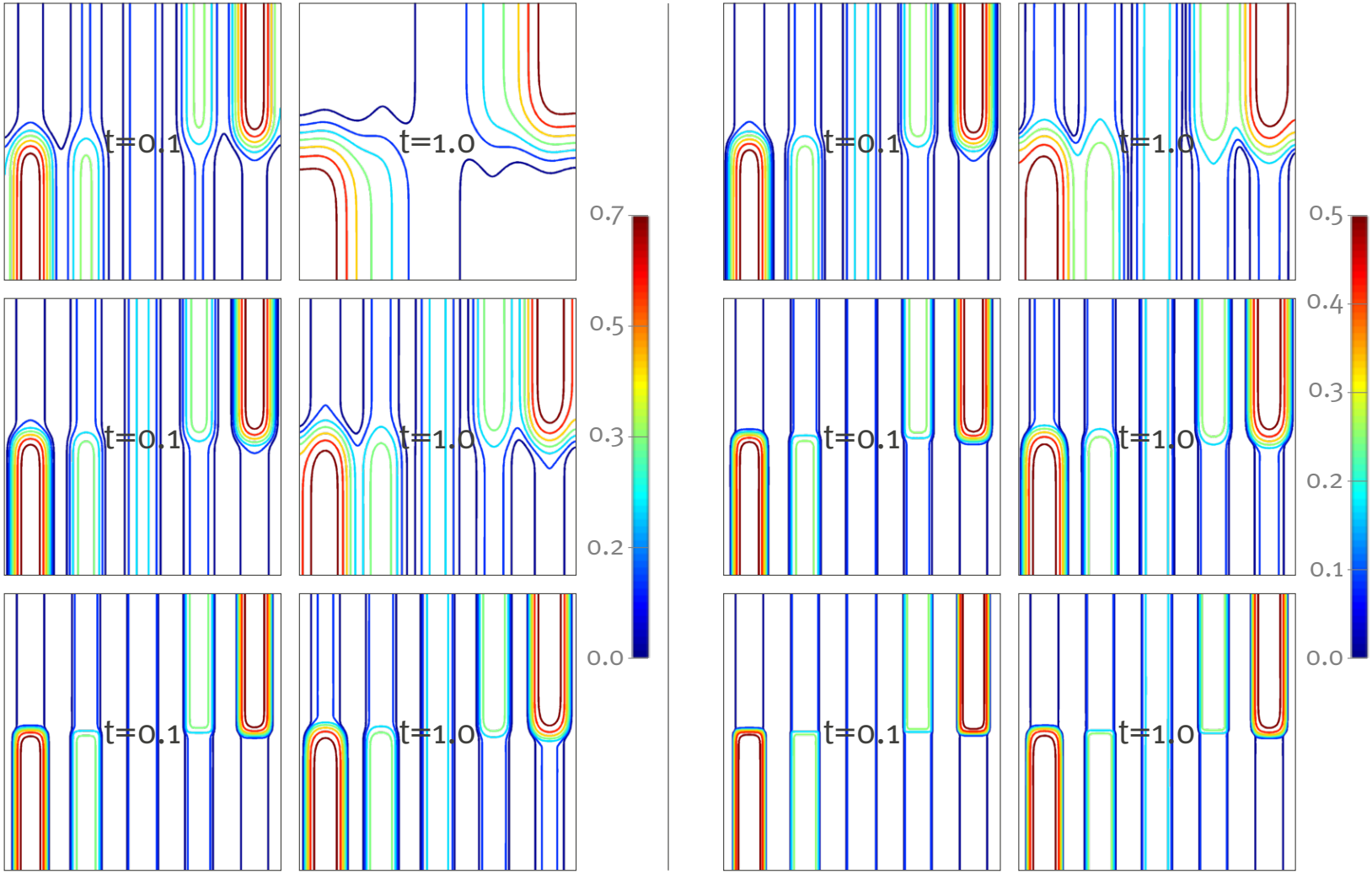
$\boldsymbol{\mu}_1 = 2 \implies$ isothermal case

$\boldsymbol{\mu}_1 = 1 \implies$ linear diffusion



$\boldsymbol{\mu}_3$ dependent initial data

› Reduced solutions for various parameters



› Averaged Runtime Comparison

Simulation	Dimensionality	Runtime[s]	Error
Detailed	H=22500	605.66	–
Reduced	N=15, M=75	5.01	$4.93 \cdot 10^{-3}$
Reduced	N=30, M=150	7.14	$1.73 \cdot 10^{-3}$
Reduced	N=40, M=200	8.27	$8.53 \cdot 10^{-4}$
Reduced	N=50, M=250	9.78	$7.59 \cdot 10^{-4}$

Gain Factor about 60 - 120

› Outline

Motivation: Multi-Scale and Multi-Physics Problems

Model Reduction: The Reduced Basis Approach

A new Reduced Basis DG Multiscale Method

› A new localized RB-DG multiscale method

[Kaulmann, Ohlberger, Haasdonk 2011]

Goal: Multiscale problem for two phase flow in porous media:

$$-\nabla \cdot (\lambda(s^\epsilon) k^\epsilon \nabla p^\epsilon) = q,$$
$$\partial_t s^\epsilon - \nabla \cdot A^\epsilon(u^\epsilon, s^\epsilon, \nabla s^\epsilon) = f.$$

› A new localized RB-DG multiscale method

[Kaulmann, Ohlberger, Haasdonk 2011]

Goal: Multiscale problem for two phase flow in porous media:

$$-\nabla \cdot (\lambda(s^\epsilon) k^\epsilon \nabla p^\epsilon) = q,$$
$$\partial_t s^\epsilon - \nabla \cdot A^\epsilon(u^\epsilon, s^\epsilon, \nabla s^\epsilon) = f.$$

First step: Consider the pressure equation as a problem depending on a parameter function $\boldsymbol{\lambda} = \boldsymbol{\lambda}(x, t)$:

$$-\nabla \cdot (\boldsymbol{\lambda} k^\epsilon \nabla p^\epsilon(\boldsymbol{\lambda}) = q,$$

› A new localized RB-DG multiscale method

[Kaulmann, Ohlberger, Haasdonk 2011]

Goal: Multiscale problem for two phase flow in porous media:

$$-\nabla \cdot (\lambda(s^\epsilon) k^\epsilon \nabla p^\epsilon) = q,$$
$$\partial_t s^\epsilon - \nabla \cdot A^\epsilon(u^\epsilon, s^\epsilon, \nabla s^\epsilon) = f.$$

First step: Consider the pressure equation as a problem depending on a parameter function $\boldsymbol{\lambda} = \boldsymbol{\lambda}(x, t)$:

$$-\nabla \cdot (\boldsymbol{\lambda} k^\epsilon \nabla p^\epsilon(\boldsymbol{\lambda}) = q,$$

$\Longrightarrow$ Apply ideas from the RB-framework!!

› General Idea (see also [Aarnes, Efendiev, Jiang 2008])

Idea: Find a small number of representative fields $\{p_i, i = 1, \ldots, N\}$, such that for all admissible parameter functions $\boldsymbol{\lambda}$ there exists a smooth, non-linear mapping S with

$$||p(\boldsymbol{\lambda}(x); x) - S(p_1, \ldots, p_N)(x)|| \leq \mathrm{TOL},$$

› General Idea (see also [Aarnes, Efendiev, Jiang 2008])

Idea: Find a small number of representative fields $\{p_i, i = 1, \ldots, N\}$, such that for all admissible parameter functions λ there exists a smooth, non-linear mapping S with

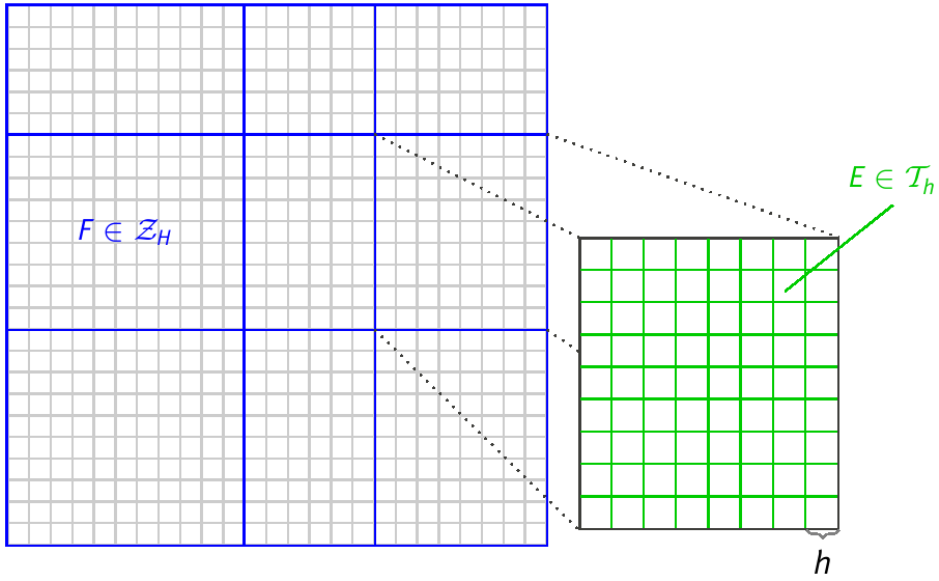
$$||p(\lambda(x); x) - S(p_1, \ldots, p_N)(x)|| \leq \text{TOL},$$

Ansatz: Define mapping S through

$$S(p_1, \ldots, p_N)(x) = \sum_{i=1}^{N} a_i(x) p_i(x)$$

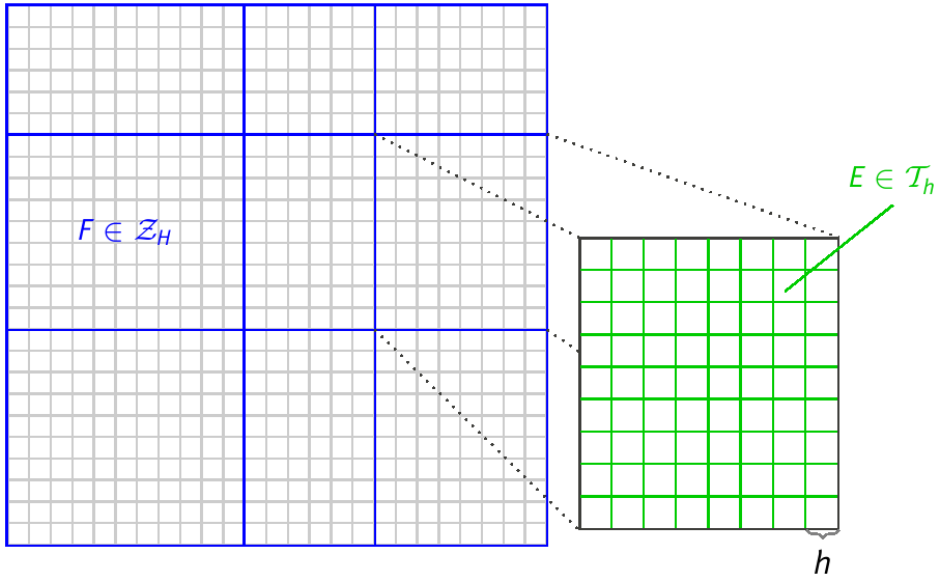
If the coefficient functions $a_i(x)$ are assumed to be piecewise constant on a coarse mesh, this leads to our new method.

› RB-DG multiscale method



› RB-DG multiscale method

$$\begin{aligned} \Phi_F &:= \{\varphi_F^1, \dots, \varphi_F^{N_F}\},\ \varphi_F^i \in S_{h,k}(F), \\ W_N &= \{v_N \in L^2(\Omega) |\ v_N|_F \in \operatorname{span}(\Phi_F), \\ &\qquad \forall F \in \mathcal{Z}_H\}. \end{aligned}$$



Given $\boldsymbol{\lambda}$, we define $p_N^{\boldsymbol{\lambda}} \in W_N$ as solution of the RB-DG multiscale method

$$B_{\mathrm{DG}}(\boldsymbol{\lambda}; p_N^{\boldsymbol{\lambda}}, v_N) = L(\boldsymbol{\lambda}; v_N) \quad \forall v_N \in W_N.$$

with

$$\begin{aligned} B_{\mathrm{DG}}(\lambda; v, w) &= \sum_{F \in \mathcal{Z}_H} \int_F \lambda k \nabla v \cdot \nabla w - \sum_{e \in \Xi} \int_e \{\lambda k \nabla v \cdot \mathbf{n_e}\}[\mathbf{w}] - \sum_{\mathbf{e} \in \cdot\cdot} \int_{\mathbf{e}} \{\lambda \mathbf{k} \nabla \mathbf{w} \cdot \mathbf{n_e}\}[\mathbf{v}] + \sum_{\mathbf{e} \in \cdot\cdot} \frac{\sigma}{|\mathbf{e}|^\beta} \int_{\mathbf{e}} [\mathbf{v}][\mathbf{w}], \\ L(\lambda; v) &= \sum_{F \in \mathcal{Z}_H} \int_F f v + \sum_{e \in \Xi_B} \int_e \left(\frac{\sigma}{|e|^\beta} v - \lambda k \nabla v \cdot \mathbf{n} \right) g_D. \end{aligned}$$

› Theorem: A posteriori error estimate

$$\begin{aligned}\|p^\lambda - p_N^\lambda\|_{0,\Omega} \quad \leq \quad & \|\mathcal{R}(p_N^\lambda) - p_N^\lambda\|_{0,\Omega} + \sum_{F\in\mathcal{Z}_H} \eta_1^F(\mathcal{R}(p_N^\lambda)) \\ & + \sum_{e\in\Gamma_I} \eta_2^e(\mathcal{R}(p_N^\lambda)) + \sum_{e\in\Xi_B} \eta_3^e(\mathcal{R}(p_N^\lambda))\end{aligned}$$

where $\mathcal{R}(p_N^\lambda)$ denotes a higher order reconstruction of p_N^λ and the indicators are given as

$$\begin{aligned}\eta_1^F(\xi) &= \frac{C_o^2}{k_1}\|f + \nabla\cdot(\lambda k\nabla\xi)\|_{0,F} + C_r\left(\frac{C_o k_2}{k_1} + h_e\right)\sum_{e\subset\partial F}\|\mathbf{r}_e(\xi)\|_{0,\Omega}, \\ \eta_2^e(\xi) &= (C_o + h_e)\frac{C_r C_o}{k_1}\|\mathbf{r}_e(\lambda k\nabla\xi\cdot\mathbf{n})\|_{0,\Omega}, \\ \eta_3^e(\xi) &= C_r\left(\frac{C_o k_2}{k_1} + h_e\right)\|\mathbf{r}_e(\xi - g_D)\|_{0,\Omega}.\end{aligned}$$

› Adaptive basis construction for W_N

Given: $\mathcal{M}_{\text{train}} := \{\lambda^i, i \in I_{\text{train}}\}$, a tolerance Δ, a maximum basis size $N_{\max}$ and a POD-tolerance Δ_{POD}.

Generate basis Φ of W_N:

› Adaptive basis construction for W_N

Given: $\mathcal{M}_{\text{train}} := \{\lambda^i, i \in I_{\text{train}}\}$, a tolerance Δ, a maximum basis size $N_{\max}$ and a POD-tolerance Δ_{POD}.

Generate basis Φ of W_N:

0. Set $\tilde{\Phi}_{-1}, \tilde{\Phi}_{-1,F} := \emptyset$ for all $F \in \mathcal{Z}_H$ and choose $\lambda_0 \in \mathcal{M}_{\text{train}}$ for the construction of an initial basis.

› Adaptive basis construction for W_N

Given: $\mathcal{M}_{\text{train}} := \{\lambda^i, i \in I_{\text{train}}\}$, a tolerance Δ, a maximum basis size $N_{\max}$ and a POD-tolerance Δ_{POD}.

Generate basis Φ of W_N:

0. Set $\tilde{\Phi}_{-1}, \tilde{\Phi}_{-1,F} := \emptyset$ for all $F \in \mathcal{Z}_H$ and choose $\lambda_0 \in \mathcal{M}_{\text{train}}$ for the construction of an initial basis.
1. Let a basis $\tilde{\Phi}_{k-1} = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k-1,F}$ and a parameter function λ_k be given. Perform detailed simulation to obtain $p_h^{\lambda_k}$ and define preliminary basis extension $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ by $\tilde{\varphi}_F := p_h^{\lambda_k}|_F, \forall F \in \mathcal{Z}_H$. Add $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ to the basis $\tilde{\Phi}_{k-1,F}$ and obtain $\tilde{\Phi}_{k,F}$, $\tilde{\Phi}_k = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k,F}$.

› Adaptive basis construction for W_N

Given: $\mathcal{M}_{\text{train}} := \{\lambda^i, i \in I_{\text{train}}\}$, a tolerance Δ, a maximum basis size $N_{\max}$ and a POD-tolerance Δ_{POD}.

Generate basis Φ of W_N:

0. Set $\tilde{\Phi}_{-1}, \tilde{\Phi}_{-1,F} := \emptyset$ for all $F \in \mathcal{Z}_H$ and choose $\lambda_0 \in \mathcal{M}_{\text{train}}$ for the construction of an initial basis.
1. Let a basis $\tilde{\Phi}_{k-1} = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k-1,F}$ and a parameter function λ_k be given. Perform detailed simulation to obtain $p_h^{\lambda_k}$ and define preliminary basis extension $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ by $\tilde{\varphi}_F := p_h^{\lambda_k}|_F, \forall F \in \mathcal{Z}_H$. Add $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ to the basis $\tilde{\Phi}_{k-1,F}$ and obtain $\tilde{\Phi}_{k,F}$, $\tilde{\Phi}_k = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k,F}$.
2. Compute offline-parts of the DG scheme and of the error estimator for the current basis $\tilde{\Phi}_k$.

› Adaptive basis construction for W_N

Given: $\mathcal{M}_{\text{train}} := \{\lambda^i, i \in I_{\text{train}}\}$, a tolerance Δ, a maximum basis size $N_{\max}$ and a POD-tolerance Δ_{POD}.

Generate basis Φ of W_N:

0. Set $\tilde{\Phi}_{-1}, \tilde{\Phi}_{-1,F} := \emptyset$ for all $F \in \mathcal{Z}_H$ and choose $\lambda_0 \in \mathcal{M}_{\text{train}}$ for the construction of an initial basis.
1. Let a basis $\tilde{\Phi}_{k-1} = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k-1,F}$ and a parameter function λ_k be given. Perform detailed simulation to obtain $p_h^{\lambda_k}$ and define preliminary basis extension $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ by $\tilde{\varphi}_F := p_h^{\lambda_k}|_F, \forall F \in \mathcal{Z}_H$. Add $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ to the basis $\tilde{\Phi}_{k-1,F}$ and obtain $\tilde{\Phi}_{k,F}$, $\tilde{\Phi}_k = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k,F}$.
2. Compute offline-parts of the DG scheme and of the error estimator for the current basis $\tilde{\Phi}_k$.
3. Compute reduced solutions p_N^λ for all $\lambda \in \mathcal{M}_{\text{train}}$ using the current basis. Then evaluate error estimator for all these solutions and find the parameter function $\lambda_{k+1} \in \mathcal{M}_{\text{train}}$ with largest error.

› Adaptive basis construction for W_N

Given: $\mathcal{M}_{\text{train}} := \{\lambda^i, i \in I_{\text{train}}\}$, a tolerance Δ, a maximum basis size $N_{\max}$ and a POD-tolerance Δ_{POD}.

Generate basis Φ of W_N:

0. Set $\tilde{\Phi}_{-1}, \tilde{\Phi}_{-1,F} := \emptyset$ for all $F \in \mathcal{Z}_H$ and choose $\lambda_0 \in \mathcal{M}_{\text{train}}$ for the construction of an initial basis.
1. Let a basis $\tilde{\Phi}_{k-1} = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k-1,F}$ and a parameter function λ_k be given. Perform detailed simulation to obtain $p_h^{\lambda_k}$ and define preliminary basis extension $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ by $\tilde{\varphi}_F := p_h^{\lambda_k}|_F, \forall F \in \mathcal{Z}_H$. Add $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ to the basis $\tilde{\Phi}_{k-1,F}$ and obtain $\tilde{\Phi}_{k,F}$, $\tilde{\Phi}_k = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k,F}$.
2. Compute offline-parts of the DG scheme and of the error estimator for the current basis $\tilde{\Phi}_k$.
3. Compute reduced solutions p_N^{λ} for all $\lambda \in \mathcal{M}_{\text{train}}$ using the current basis. Then evaluate error estimator for all these solutions and find the parameter function $\lambda_{k+1} \in \mathcal{M}_{\text{train}}$ with largest error.
4. If $N < N_{\max}$ and if the error bound for the reduced solution $p_N^{\lambda_{k+1}}$ is larger than Δ, continue with Step (1) with λ_{k+1} from Step (3).

› Adaptive basis construction for W_N

Given: $\mathcal{M}_{\text{train}} := \{\lambda^i, i \in I_{\text{train}}\}$, a tolerance Δ, a maximum basis size $N_{\max}$ and a POD-tolerance Δ_{POD}.

Generate basis Φ of W_N:

0. Set $\tilde{\Phi}_{-1}, \tilde{\Phi}_{-1,F} := \emptyset$ for all $F \in \mathcal{Z}_H$ and choose $\lambda_0 \in \mathcal{M}_{\text{train}}$ for the construction of an initial basis.
1. Let a basis $\tilde{\Phi}_{k-1} = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k-1,F}$ and a parameter function λ_k be given. Perform detailed simulation to obtain $p_h^{\lambda_k}$ and define preliminary basis extension $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ by $\tilde{\varphi}_F := p_h^{\lambda_k}|_F, \forall F \in \mathcal{Z}_H$. Add $\tilde{\varphi}_F, F \in \mathcal{Z}_H$ to the basis $\tilde{\Phi}_{k-1,F}$ and obtain $\tilde{\Phi}_{k,F}$, $\tilde{\Phi}_k = \bigcup_{F \in \mathcal{Z}_H} \tilde{\Phi}_{k,F}$.
2. Compute offline-parts of the DG scheme and of the error estimator for the current basis $\tilde{\Phi}_k$.
3. Compute reduced solutions p_N^{λ} for all $\lambda \in \mathcal{M}_{\text{train}}$ using the current basis. Then evaluate error estimator for all these solutions and find the parameter function $\lambda_{k+1} \in \mathcal{M}_{\text{train}}$ with largest error.
4. If $N < N_{\max}$ and if the error bound for the reduced solution $p_N^{\lambda_{k+1}}$ is larger than Δ, continue with Step (1) with λ_{k+1} from Step (3).

   Else Apply POD with accuracy Δ_{POD} to $\tilde{\Phi}_{k,F}$ on each coarse cell $F \in \mathcal{Z}_H$ and obtain the reduced orthogonalized local bases Φ_F and the global basis $\Phi = \bigcup_{F \in \mathcal{Z}_H} \Phi_F$.

› Numerical Experiment

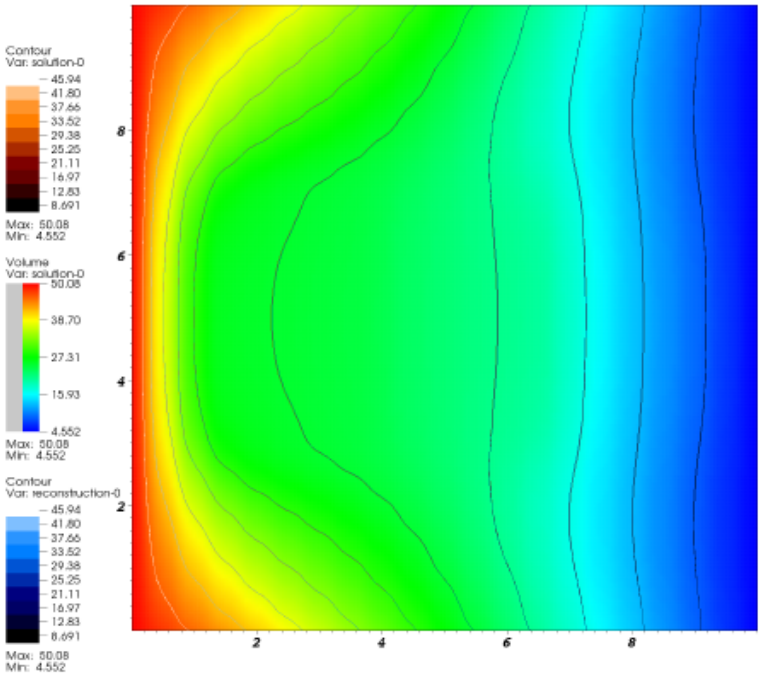
$$-\nabla \cdot (\lambda k^\epsilon \nabla p^\epsilon(\lambda) = 0 \qquad \text{on } \Omega = [0, 10]^2$$

with

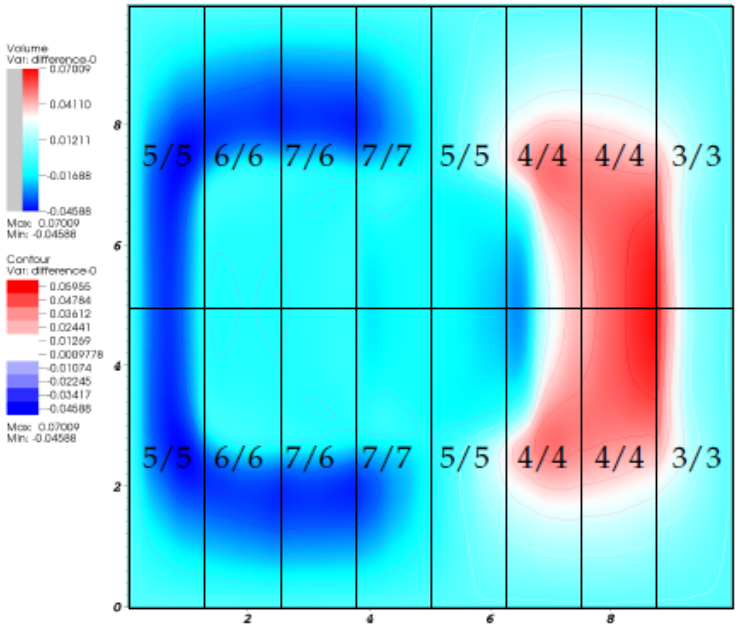
$$\begin{aligned}
k^\varepsilon(x) &:= \frac{2}{3}(1 + x_1)(1 + \cos^2(2\pi \frac{x_1}{\varepsilon}), \\
\lambda(x) &:= \frac{1}{\eta_o} - \frac{2}{\eta_o} S(x) + \frac{\eta_o + \eta_w^2}{\eta_w \eta_o} \sum_{m,n=1}^{N_S} \mu_n \mu_m S_n(x) S_m(x), \\
S(x) &:= \sum_{n=1}^{N_S} \mu_n S_n(x) \text{ with } N_S = 3 \text{ and } S_n(x) \text{ given.}
\end{aligned}$$

$+$ suitable Dirichlet boundary conditions.

› Simulation results



Contour plots of fine scale solution (solid lines) and reconstructed reduced solution (dotted lines) for $\mu_1 = 0.85$, $\mu_2 = 0.5$, $\mu_3 = 0.1$ ($|\mathcal{T}_h| = 32768$).



Difference between fine scale and reduced solution. Coarse triangulation (black) with number of reduced basis functions $|\Phi_F|$ ($|\mathcal{T}_h| = 2048/32768$, respectively).

› CPU times for the new method

$\lvert\mathcal{T}_h\rvert$	N	t_{highdim} (s)	t_{lowdim} (ms)	t_{recons} (ms)	Factor	rel. error
2,048	82	0.19	8.54	36.78	**4**	$4.74e{-4}$
8,192	80	2.59	9.93	151.4	**16**	$6.44e{-4}$
32,768	80	22.58	12.24	545.3	**40**	$7.59e{-4}$

Averaged runtimes over 125 simulations: high and low dimensional algorithms (t_{highdim} and t_{lowdim}); the reconstruction (t_{recons}) and mean relative errors ($\|p_h^\lambda - p_N^\lambda\|_{L^2} / \|p_h^\lambda\|_{L^2}$) for different grid sizes.

Thank you for your attention!

Software: DUNE, DUNE-FEM, RBmatlab, DUNE-RB

www.wwu.de/math/num/ohlberger

Thank you for your attention!

Software: DUNE, DUNE-FEM, RBmatlab, DUNE-RB

www.wwu.de/math/num/ohlberger

PDESoft2012: Workshop on PDE Software Frameworks
10th Anniversary of DUNE
June 18 - 20, 2012, Muenster, Germany.
http://pdesoft2012.uni-muenster.de/

MoRePaS II: Second International Workshop on Model Reduction for Parametrized Systems
Oct 2-5, 2012, Schloss Reisensburg, Guenzburg, Germany.
http://www.morepas.org/workshop2012/