Model reduction for multiscale problems

Mario Ohlberger

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

M. Ohlberger Model reduction for multiscale problems
Example: Hydrological Modeling

[BMBF-Project AdaptHydroMod: Wald & Corbe, Hügelsheim]
Mathematical Modelling and Model Reduction

Increasing Efficiency

Real World Problem

Continuous Mathematical Model
- Here: system of partial differential equations
- Problem: infinite dimensional solution space
- No solutions in closed form

M. Ohlberger
Model reduction for multiscale problems
Mathematical Modelling and Model Reduction

Increasing Efficiency

Continuous Mathematical Model

- fine grid resolution
- Discretization!!
- Classical Paradigm
- coarse grid resolution

Model reduction for multiscale problems
Mathematical Modelling and Model Reduction
Increasing Efficiency

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 \( X_h \)

\[ \implies \text{Very general approach, but in particular cases not very efficient!!} \]

M. Ohlberger
Model reduction for multiscale problems
Mathematical Modelling and Model Reduction

Increasing Efficiency

Continuous Mathematical Model

Increasing Error

Problem Specific Adaptivity

M. Ohlberger

Model reduction for multiscale problems
Continuous Mathematical Model

Problem specific: Adaptive Mesh Refinement

- Typical error estimates:
  \[ ||u - u_h|| \leq c \eta(u_h) \]

- Error related to approximate solution!
- 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

HMM for nonlinear monotone elliptic problems
[Henning, Ohlberger 2011]

⇒ see poster (8) at this workshop
Continuous Mathematical Model

Problem class specific: Reduced Basis Method

- Typical error estimates:
  \[ \| (u - u_N)(\mu) \| \leq c \eta(u_N(\mu)) \]

- Error related to reduced solution!

- \(\Rightarrow\) 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],
> 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

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

plus suitable Initial and Boundary Conditions.

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

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

plus suitable Initial and Boundary Conditions.

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

Ansatz (RB): Define low dimensional Subspace \( W_N \subset W_H \) and project FV/DG Scheme onto the Subspace

\[\Rightarrow \quad \text{RB Approximation } c_N(\mu) \in W_N.\]
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

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

plus suitable Initial and Boundary Conditions.

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

Ansatz (RB): Define low dimensional Subspace $W_N \subset W_H$ and project FV/DG Scheme onto the Subspace

$\implies$ RB Approximation $c_N(\mu) \in W_N$.

Requirement:
- Efficient Choice of $W_N$ (Exponential Convergence in N)
- Offline–Online Decomposition for all Calculations
- Error Control for $\|c_H(\mu) - c_N(\mu)\|$
Assumption: FV/DG Scheme for Evolution Equations

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

with time step counter \( k \) and \( c_H^k(\mu) \in \mathcal{W}_H \).
Model Reduction: Reduced Basis Method

Assumption: FV/DG Scheme for Evolution Equations

\[ c^0_H = P[c_0(\mu)], \quad L^k_I(\mu)[c_H^{k+1}(\mu)] = L^k_E(\mu)[c_H^k(\mu)] + b^k(\mu). \]

with time step counter \( k \) and \( c_H^k(\mu) \in W_H \).

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

Sought: \( c_N^k(\mu) = \sum_{n=1}^N a_n^k(\mu)\varphi_n \) with \( L^k_I(\mu)a^{k+1} = L^k_E(\mu)a^k + b^k(\mu) \)

where

\[ (L^k_I(\mu))_{nm} := \int_\Omega \varphi_n L^k_I(\mu)[\varphi_m], \quad (L^k_E(\mu))_{nm} := \int_\Omega \varphi_n L^k_E(\mu)[\varphi_m], \]

\[ (a^0(\mu))_n = \int_\Omega P[c_0(\mu)]\varphi_n, \quad (b^k(\mu))_n := \int_\Omega \varphi_n b^k(\mu). \]
Offline–Online Decomposition

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

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

Goal: All Steps for the Calculation of \( c_N(\mu) \) 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

\[
L_i^k(\mu)[\cdot] = \sum_{q=1}^{Q} L_{i}^{k,q}[\cdot] \quad \sigma_{L_i}^q(\mu)
\]

depending on \( x \) \quad depending on \( \mu \)
Offline–Online Decomposition

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

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

Constrained: Affine Parameter Dependency of the Evolution Scheme

\[
L^k_i(\mu)[\cdot] = \sum_{q=1}^{Q} L^k_i,q[\cdot], \quad \sigma^q_{L_i}(\mu)
\]

depending on $x$ depending on $\mu$

$\implies$ Precompute offline: $$(L^k_i,q)_{nm} := \int_{\Omega} \varphi_n L^k_i,q[\varphi_m]$$

$\implies$ Assemble online: $$(L^k_i(\mu))_{nm} := \sum_{q=1}^{Q} (L^k_i,q)_{nm}\sigma^q_{L_i}(\mu)$$
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:** \(40 \times 200\) Elements, \(K = 200\) time steps

<table>
<thead>
<tr>
<th></th>
<th>time dependent data</th>
<th>constant data</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reference</td>
<td>RB online</td>
<td>RB offline</td>
<td>Reference</td>
<td>RB online</td>
</tr>
<tr>
<td>implicit Factor</td>
<td>155.94s</td>
<td>16.67s</td>
<td>447.16s</td>
<td>45.67s</td>
<td>1.02s</td>
</tr>
<tr>
<td>explicit Factor</td>
<td>105.97s</td>
<td>16.53s</td>
<td>437.20s</td>
<td>1.51s</td>
<td>0.79s</td>
</tr>
</tbody>
</table>

**Discretization:** \(80 \times 400\) Elements, \(K = 1000\) time steps

<table>
<thead>
<tr>
<th></th>
<th>time dependent data</th>
<th>constant data</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reference</td>
<td>RB online</td>
<td>RB offline</td>
<td>Reference</td>
<td>RB online</td>
</tr>
<tr>
<td>implicit Factor</td>
<td>4043.18s</td>
<td>143.57s</td>
<td>8693.90s</td>
<td>924.91s</td>
<td>6.18s</td>
</tr>
<tr>
<td>explicit Factor</td>
<td>2758.20s</td>
<td>134.00s</td>
<td>8506.60s</td>
<td>17.41s</td>
<td>3.64s</td>
</tr>
</tbody>
</table>

M. Ohlberger

Model reduction for multiscale problems
A Posteriori Error Estimates [Haasdonk, Ohlberger ’08]

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

$$R^{k+1}(\mu)[c_N] := \frac{1}{\Delta t} \left( L^k_i(\mu)[c^k_{N+1}(\mu)] - L^k_E(\mu)[c^k_N(\mu)] - b^k(\mu) \right)$$
\( R^{k+1}(\mu)[c_N] := \frac{1}{\Delta t} \left( L^k_i(\mu)[c_{N}^{k+1}(\mu)] - L^k_E(\mu)[c_{N}^k(\mu)] - b^k(\mu) \right) \)

\[
\| c_N^k(\mu) - c_H^k(\mu) \|_{L^2(\Omega)} \leq \sum_{l=0}^{k-1} \Delta t \ (C_E)^{k-1-l} \ \| R^{l+1}(\mu)[c_N(\mu)] \|_{L^2(\Omega)}
\]
Efficient Choice of $W_N$: POD-Greedy [Haasdonk, O. ’08]

General Idea: • Construct $W_N$ from snapshots $c^l_H(\mu)$. 
Efficient Choice of $W_N$: POD-Greedy [Haasdonk, O. ’08]

General Idea:  
• Construct $W_N$ from snapshots $c_l^H(\mu)$.

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 $\mathcal{W}_N$: POD-Greedy [Haasdonk, O. ’08]

General Idea:
- Construct $\mathcal{W}_N$ from snapshots $c^l_H(\mu)$.

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 $\mathcal{W}_N$: POD-Greedy [Haasdonk, O. ’08]

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

![Training Error Estimator Convergence](image1)

![Test Error Estimator Convergence](image2)
Adaptive Basis Enrichment  [Haasdonk, Ohlberger ’08]

Error Distribution for Uniform / Adaptive Training Sets

Exponential Convergence and CPU-Efficiency

M. Ohlberger  Model reduction for multiscale problems
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]

- Non-affine parameter dependence: Use classical empirical interpolation of functions

- 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
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

- 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

- **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

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

plus suitable Initial and Boundary Conditions.

\[ \mu = 1 \implies \text{Linear Transport} \]

\[ \mu = 2 \implies \text{Burgers Equation} \]
Numerical Results

Initial values: \( c_0(x) = \frac{1}{2}(1 + \sin(2\pi x_1) \sin(2\pi x_2)) \)

Solution at \( t = 0.3 \)

Linear Transport  
Burgers Equation

M. Ohlberger  
Model reduction for multiscale problems
General Framework

Nonlinear Equation

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

Explicit Discretization

\[ c_{H}^{k+1}(\mu) = c_{H}^{k}(\mu) - \Delta t L_{H}^{k}(\mu)[c_{H}^{k}(\mu)]. \]

Problem: Non-Affine Parameter Dependency
Non-Linear Evolution Operator
Idea: Linear Affine Approximation through Empirical Interpolation

\[ L_{H}^{k}(\mu)[c_{H}^{k}(\mu)](x) \approx \sum_{m=1}^{M} y_m(c, \mu, t^k) \xi_m(x) \]

M. Ohlberger Model reduction for multiscale problems
General Framework

Nonlinear Equation

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

Explicit Discretization

$$c_{H}^{k+1}(\mu) = c_{H}^{k}(\mu) - \Delta tL_{H}^{k}(\mu)[c_{H}^{k}(\mu)].$$

Problem: Non-Affine Parameter Dependency

Non-Linear Evolution Operator

Idea: Linear Affine Approximation through Empirical Interpolation

$$L_{H}^{k}(\mu)[c_{H}^{k}(\mu)](x) \approx \sum_{m=1}^{M} y_{m}(c, \mu, t^{k})\xi_{m}(x)$$

$$y_{m}(c, \mu, t^{k}) := L_{H}^{k}(\mu)[c_{H}^{k}(\mu)](x_{m})$$
Empirical Interpolation of Localized Operators

Idea: Construct a Collateral Reduced Basis Space $W_M$ that approximates the space spanned by $L^k_H(\mu)[c^k_H(\mu)]$
Empirical Interpolation of Localized Operators

Idea: Construct a Collateral Reduced Basis Space $W_M$ that approximates the space spanned by $L^k_H(\mu)[c^k_H(\mu)]$

Ingredients: Collateral Reduced Basis Space:

$$W_M := \text{span}\{L^k_H(\mu_m)[c^k_H(\mu_m)] | m = 1, \ldots, M\}$$
Empirical Interpolation of Localized Operators

Idea: Construct a Collateral Reduced Basis Space \( W_M \) that approximates the space spanned by \( L^k_H(\mu)[c^k_H(\mu)] \)

Ingredients: Collateral Reduced Basis Space:
\[
W_M := \text{span}\{L^k_H(\mu_m)[c^k_H(\mu_m)] | m = 1, \ldots, M\}
\]

Nodal Collateral Reduced Basis:
\[
\{\xi_m\}_{m=1}^M \Rightarrow W_M = \text{span}\{\xi_m | m = 1, \ldots, M\}
\]

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 $L^k_H(\mu)[c^k_H(\mu)]$

Ingredients: Collateral Reduced Basis Space:

$$W_M := \text{span}\{L^m_H(\mu)[c^m_H(\mu)] | m = 1, \ldots, M\}$$

Nodal Collateral Reduced Basis:

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

Interpolation Points:

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

Empirical Interpolation:

$$\mathcal{I}_M[L^k_H(\mu)[c^k_H(\mu)]] := \sum_{m=1}^M y_m(c, \mu, t^k)\xi_m(x)$$
Empirical Interpolation of Localized Operators

Idea: Construct a Collateral Reduced Basis Space $W_M$ that approximates the space spanned by $L^k_H(\mu)[c^k_H(\mu)]$

Ingredients: Collateral Reduced Basis Space:

$W_M := \text{span}\{L^k_m(\mu_m)[c^k_m(\mu_m)]| m = 1, \ldots, M\}$

Nodal Collateral Reduced Basis:

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

Interpolation Points:

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

Empirical Interpolation:

$I_M[L^k_H(\mu)[c^k_H(\mu)]] := \sum_{m=1}^M y_m(c, \mu, t^k)\xi_m(x)$

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

Online: Calculate Coefficients $y_m = L^k_H(\mu)[c^k_H(\mu)](x_m)$
Empirical Interpolation of Localized Operators

Idea: Construct a Collateral Reduced Basis Space $W_M$ that approximates the space spanned by $L^k_H(\mu)[c^k_H(\mu)]$

Ingredients: Collateral Reduced Basis Space:
$$W_M := \text{span}\{L^k_H(\mu_m)[c^k_H(\mu_m)]| m = 1, \ldots, M\}$$

Nodal Collateral Reduced Basis:
$$\{\xi_m\}_{m=1}^M \implies W_M = \text{span}\{\xi_m| m = 1, \ldots, M\}$$

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

Empirical Interpolation:
$$\mathcal{I}_M[L^k_H(\mu)[c^k_H(\mu)]] := \sum_{m=1}^M y_m(c, \mu, t^k)\xi_m(x)$$

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

Online: Calculate Coefficients $y_m = L^k_H(\mu)[c^k_H(\mu)](x_m)$
$$\implies$$ 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 $c^k_N$ from coefficients $a^k$
2.) Local operator evaluation: $y_m = L^k_H(\mu)[c^k_H(\mu)](x_m)$

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 $c^k_N$ from coefficients $a^k$
2.) Local operator evaluation: $y_m = L^k_H(\mu)[c^k_H(\mu)](x_m)$

Requires Offline: Numerical subgrids, local basis representation

RB Method: Galerkin projection of interpolated scheme
$$\int_\Omega \left( c^{k+1}_N(\mu) - c^k_N(\mu) - \Delta t I_M[L^k_H(\mu)[c^k_H(\mu)]] \right) \varphi, \quad \forall \varphi \in W_N.$$
Numerical Experiment

Empirical Interpolation:

\[ M_{\text{max}} = 150 \text{ interpolation points} \]

Translation symmetry detected
Numerical Experiment

Empirical Interpolation:

\[ M_{\text{max}} = 150 \text{ interpolation points} \]

Translation symmetry detected

Test error convergence:

Exponential convergence for simultaneous increase of \( N \) and \( M \)
## Numerical Experiment

### Comparison of Online-Runtimes

<table>
<thead>
<tr>
<th>Simulation</th>
<th>Dimension</th>
<th>Runtime [s]</th>
<th>Gain Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>detailed</td>
<td>H = 7200</td>
<td>20.22</td>
<td></td>
</tr>
<tr>
<td>reduced</td>
<td>N=20, M=30</td>
<td>0.91</td>
<td>22.2</td>
</tr>
<tr>
<td>reduced</td>
<td>N=40, M=60</td>
<td>1.22</td>
<td>16.6</td>
</tr>
<tr>
<td>reduced</td>
<td>N=60, M=90</td>
<td>1.55</td>
<td>13.0</td>
</tr>
<tr>
<td>reduced</td>
<td>N=80, M=120</td>
<td>1.77</td>
<td>11.4</td>
</tr>
<tr>
<td>reduced</td>
<td>N=100, M=150</td>
<td>2.06</td>
<td>9.8</td>
</tr>
</tbody>
</table>
Extension to Nonlinear Implicit Operators

[Drohmann, Haasdonk, Ohlberger 2010]

Evolution Equation

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

Mixed Implicit - Explicit Discretization

\[(Id + \Delta t L_i^k(\mu))[c_h^{k+1}(\mu)] = (Id - \Delta t L_E^k(\mu))[c_h^k(\mu)].\]

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

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

Mixed Implicit - Explicit Discretization

$$(I_d + \Delta t L^{k}_I(\mu))[c^{k+1}_H(\mu)] = (I_d - \Delta t L^{k}_E(\mu))[c^{k}_H(\mu)].$$

Problem: Non-Affine Parameter Dependency

Non-Linear Evolution Operators

$L^k_I$ 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 \( \nu \) for the defect
\[
(Id + \Delta t F_{I}^{k}(\mu))[c_{H}^{k+1,\nu}]d_{H}^{k+1,\nu+1} = (Id - \Delta t L_{I}^{k}(\mu))[c_{H}^{k+1,\nu}] + (Id - \Delta t L_{E}^{k}(\mu))[c_{H}^{k}],
\]
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 $L^k_i$

\[
\mathcal{I}_M[L^k_i(\mu)[c_H]] = \sum_{m=1}^{M} y^l_m(c^k_H, \mu) \xi_m.
\]

Empirical Interpolation for $F^k_i$

\[
\mathcal{I}_M[F^k_i(\mu)[c_H]v_H] := \sum_{i=1}^{H} \sum_{m=1}^{M} \partial_i y^l_m(c^k_H, \mu)v_i \xi_m = \sum_{i \in \tau} \sum_{m=1}^{M} \partial_i y^l_m(c^k_H, \mu)v_i \xi_m.
\]

Properties:

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

M. Ohlberger
Resulting RB Formulation of one Newton Step

Ansatz: \( c_{N}^{k,\nu}(x) = \sum_{n=1}^{N} a_{n}^{k,\nu} \phi_{n}(x), \) (\( a^{k,\nu} \): coefficient vector)

\[
(Id + \Delta t G A[c_{N}^{k+1,\nu}]) \begin{pmatrix} a_{k+1,\nu+1}^{k+1} - a_{k+1,\nu}^{k+1} \end{pmatrix} = RHS(a^{k+1,\nu}, a^{k}).
\]

Thereby the matrices \( A[c_{N}], G \) are given as

\[
(A[c_{N}])_{m,n} := \sum_{i=1}^{M} \partial_{i} y_{m}^{l}(c_{N}, \mu) \phi_{n}(x_{i}), \quad G_{n,m} := \int_{\Omega} \xi_{m} \phi_{n}
\]

with a corresponding offline-online splitting.
A Posteriori Error Estimate

Definition: Residual of the approximated FV/DG Method

\[ \Delta t R^{k+1}(\mu)[c_N] = (\text{Id} + \Delta t I_M[L_I(\mu)]) \left[ c_N^{k+1} \right] - (\text{Id} - \Delta t I_M[L_E(\mu)]) \left[ c_N^k \right] \]

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

\[ \left\| c_N^k(\mu) - c_H^k(\mu) \right\|_{L^2(\Omega)} \leq \sum_{i=0}^{k-1} c_{i+1}^{k-i} c_{E}^{k-1} \left( \left\| \sum_{m=M}^{M+M'} \Delta t \left( y_m^i \left( c_N^{i+1}, \mu \right) - y_m^E \left( c_N^i, \mu \right) \right) \right\|_{L^2(\Omega)} \right) \]

\[ + \varepsilon^{\text{New}} + \left\| R^{l+1}(\mu)[c_N] \right\|_{L^2(\Omega)} \]

M. Ohlberger

Model reduction for multiscale problems
Numerical Experiments

Model Problem: Porous Medium Equation

\[ \partial_t c(\mu) + \mu_2 \Delta(c^{\mu_1}(\mu)) = 0 \quad \text{in } \Omega \times [0, T], \quad \mu \in [1, 5] \times [0, 0.001] \times [0, 0.2] \]

plus suitable initial and boundary conditions.

Nonlinearity:

- \( \mu_1 > 2 \) \( \Rightarrow \) adiabatic flow
- \( \mu_1 = 2 \) \( \Rightarrow \) isothermal case
- \( \mu_1 = 1 \) \( \Rightarrow \) linear diffusion

\( \mu_3 \) dependent initial data
Reduced solutions for various parameters

M. Ohlberger

Model reduction for multiscale problems
## Averaged Runtime Comparison

<table>
<thead>
<tr>
<th>Simulation</th>
<th>Dimensionality</th>
<th>Runtime [s]</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detailed</td>
<td>H=22500</td>
<td>605.66</td>
<td>—</td>
</tr>
<tr>
<td>Reduced</td>
<td>N=15, M=75</td>
<td>5.01</td>
<td>$4.93 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>Reduced</td>
<td>N=30, M=150</td>
<td>7.14</td>
<td>$1.73 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>Reduced</td>
<td>N=40, M=200</td>
<td>8.27</td>
<td>$8.53 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>Reduced</td>
<td>N=50, M=250</td>
<td>9.78</td>
<td>$7.59 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

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:

\[-\nabla \cdot (\lambda(s^\varepsilon)k^\varepsilon \nabla p^\varepsilon) = q,\]

\[\partial_t s^\varepsilon - \nabla \cdot A^\varepsilon(u^\varepsilon, s^\varepsilon, \nabla s^\varepsilon) = f.\]
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 \( \lambda = \lambda(x, t) \):

\[-\nabla \cdot (\lambda k^\epsilon \nabla p^\epsilon(\lambda)) = q,\]
A new localized RB-DG multiscale method

[Kaulmann, Ohlberger, Haasdonk 2011]

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

\[-\nabla \cdot (\lambda(s^\varepsilon)k^\varepsilon \nabla p^\varepsilon) = q,\]

\[\partial_t s^\varepsilon - \nabla \cdot A^\varepsilon(u^\varepsilon, s^\varepsilon, \nabla s^\varepsilon) = f.\]

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

\[-\nabla \cdot (\lambda k^\varepsilon \nabla p^\varepsilon(\lambda)) = q,\]

\[\Rightarrow \text{Apply ideas from the RB-framework!!}\]
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 \( \lambda \) there exists a smooth, non-linear mapping \( S \) with

\[
\| p(\lambda(x); x) - S(p_1, \ldots, p_N)(x) \| \leq \text{TOL},
\]
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 \( \lambda \) 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.
\( \Phi^F := \{ \varphi^1_F, \ldots, \varphi^N_F \} \) with \( \varphi^i_F \in S_h, k(F) \)

\( \mathcal{W}^N = \{ v^N \in L^2(\Omega) : v^N|_F \in \text{span}(\Phi^F), \forall F \in \mathcal{Z} \} \).

Given \( \lambda \), we define \( p^\lambda_N \in \mathcal{W}^N \) as solution of the RB-DG multiscale method

\( \mathcal{B}^\text{DG}(\lambda; p^\lambda_N, v^N) = \mathcal{L}(\lambda; v^N) \forall v^N \in \mathcal{W}^N \).

with

\[
\mathcal{B}^\text{DG}(\lambda; v, w) = \sum_{F \in \mathcal{Z}} \int_F \lambda k \nabla v \cdot \nabla w - \sum_{e \in \mathcal{E}} \int_e \{ \lambda k \nabla v \cdot n_e \} \llbracket w \rrbracket - \sum_{e \in \mathcal{E}} \int_e \{ \lambda k \nabla w \cdot n_e \} \llbracket v \rrbracket + \sum_{e \in \mathcal{E}} \sigma|_e |\beta \int_e \llbracket v \rrbracket \llbracket w \rrbracket,
\]

\( \mathcal{L}(\lambda; v) = \sum_{F \in \mathcal{Z}} \int_F fv + \sum_{e \in \mathcal{E}} \mathcal{B} \int_e (\sigma|_e |\beta v - \lambda k \nabla v \cdot n_e) g_D. \)
RB-DG multiscale method

\[ \Phi_F := \{ \varphi^1_F, \ldots, \varphi^{N_F}_F \}, \varphi^i_F \in S_{h,k}(F), \]
\[ W_N = \{ v_N \in L^2(\Omega) \mid v_N|_F \in \text{span}(\Phi_F), \quad \forall F \in \mathcal{Z}_H \}. \]

Given \( \lambda \), we define \( p^\lambda_N \in W_N \) as solution of the RB-DG multiscale method

\[ B_{DG}(\lambda; p^\lambda_N, v_N) = L(\lambda; v_N) \quad \forall v_N \in W_N. \]

with

\[ B_{DG}(\lambda; v, w) = \sum_{F \in \mathcal{Z}_H} \int_F \lambda k \nabla v \cdot \nabla w - \sum_{e \in \Xi} \int_e \{ \lambda k \nabla v \cdot n_e \}[w] - \sum_{e \in \Xi} \int_e \{ \lambda k \nabla w \cdot n_e \}[v] + \sum_{e \in \Xi} \frac{\sigma}{|e|^{\beta}} \int_e [v][w], \]

\[ L(\lambda; v) = \sum_{F \in \mathcal{Z}_H} \int_F f v + \sum_{e \in \Xi_B} \int_e \left( \frac{\sigma}{|e|^{\beta}} v - \lambda k \nabla v \cdot n \right) g_D. \]
Theorem: A posteriori error estimate

\[ \| p^\lambda - p^\lambda_N \|_{0, \Omega} \leq \| R(p^\lambda_N) - p^\lambda_N \|_{0, \Omega} + \sum_{F \in \mathcal{Z}_H} \eta^F_1(R(p^\lambda_N)) \]

\[ + \sum_{e \in \Gamma_I} \eta^e_2(R(p^\lambda_N)) + \sum_{e \in \Xi_B} \eta^e_3(R(p^\lambda_N)) \]

where \( R(p^\lambda_N) \) denotes a higher order reconstruction of \( p^\lambda_N \) and the indicators are given as

\[ \eta^F_1(\xi) = \frac{C_0^2}{k_1} \| f + \nabla \cdot (\lambda k \nabla \xi) \|_{0,F} + C_r \left( \frac{C_0 k_2}{k_1} + h_e \right) \sum_{e \subset \partial F} \| r_e(\xi) \|_{0,\Omega}, \]

\[ \eta^e_2(\xi) = (C_0 + h_e) \frac{C_r C_0}{k_1} \| r_e(\lambda k \nabla \xi \cdot \mathbf{n}) \|_{0,\Omega}, \]

\[ \eta^e_3(\xi) = C_r \left( \frac{C_0 k_2}{k_1} + h_e \right) \| r_e(\xi - g_D) \|_{0,\Omega}. \]
Adaptive basis construction for $W_N$

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

Generate basis $\Phi$ of $W_N$:

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

2. Let a basis $\tilde{\Phi}_{k}^{-1} = \bigcup_{F \in Z_H} \tilde{\Phi}, F$ and a parameter function $\lambda_k$ be given. Perform detailed simulation to obtain $p_{\lambda_k}$ and define preliminary basis extension $\tilde{\phi}_F, F \in Z_H$ by $\tilde{\phi}_F := p_{\lambda_k}|_F, \forall F \in Z_H$. Add $\tilde{\phi}_F, F \in Z_H$ to the basis $\tilde{\Phi}_{k}^{-1}$ and obtain $\tilde{\Phi}_{k} = \bigcup_{F \in Z_H} \tilde{\Phi}_{k}, F$.

3. Compute offline-parts of the DG scheme and of the error estimator for the current basis $\tilde{\Phi}_{k}$.

4. Compute reduced solutions $p_{\lambda}$ 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.

5. If $N < N_{\text{max}}$ and if the error bound for the reduced solution $p_{\lambda_{k+1}}$ is larger than $\Delta$, continue with Step (1) with $\lambda_{k+1}$ from Step (3). Else apply POD with accuracy $\Delta_{\text{POD}}$ to $\tilde{\Phi}_{k}, F$ on each coarse cell $F \in Z_H$ and obtain the reduced orthogonalized local bases $\Phi_F$ and the global basis $\Phi = \bigcup_{F \in Z_H} \Phi_F$. 

M. Ohlberger Model reduction for multiscale problems
Adaptive basis construction for $W_N$

**Given:** $M_{\text{train}} := \{\lambda^i, i \in I_{\text{train}}\}$, a tolerance $\Delta$, a maximum basis size $N_{\text{max}}$ and a POD-tolerance $\Delta_{\text{POD}}$.

**Generate basis $\Phi$ of $W_N$:**

1. Set $\Phi_{-1}, \Phi_{-1,F} := \emptyset$ for all $F \in Z_H$ and choose $\lambda_0 \in M_{\text{train}}$ 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 $\Delta$, a maximum basis size $N_{\text{max}}$ and a POD-tolerance $\Delta_{\text{POD}}$.

**Generate basis $\Phi$ of $W_N$:**

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

1. Let a basis $\tilde{\Phi}_{k-1} = \bigcup_{F \in Z_H} \tilde{\Phi}_{k-1,F}$ and a parameter function $\lambda_k$ be given. Perform detailed simulation to obtain $p^\lambda_h$ and define preliminary basis extension $\tilde{\varphi}_F, F \in Z_H$ by $\tilde{\varphi}_F := p^\lambda_h|_F, \forall F \in Z_H$. Add $\tilde{\varphi}_F, F \in Z_H$ to the basis $\tilde{\Phi}_{k-1,F}$ and obtain $\tilde{\Phi}_k, F, \tilde{\Phi}_k = \bigcup_{F \in Z_H} \tilde{\Phi}_{k,F}$.
Adaptive basis construction for \( W_N \)

**Given:** \( \mathcal{M}_{\text{train}} := \{ \lambda^i, i \in I_{\text{train}} \} \), a tolerance \( \Delta \), a maximum basis size \( N_{\text{max}} \) and a POD-tolerance \( \Delta_{\text{POD}} \).

**Generate basis \( \Phi \) of \( W_N \):**

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

1. Let a basis \( \tilde{\Phi}_{k-1} = \bigcup_{F \in Z_H} \tilde{\Phi}_{k-1,F} \) and a parameter function \( \lambda_k \) be given. Perform detailed simulation to obtain \( p^\lambda_h \) and define preliminary basis extension \( \tilde{\varphi}_F, F \in Z_H \) by \( \tilde{\varphi}_F := p^\lambda_h|_F, \forall F \in Z_H \). Add \( \tilde{\varphi}_F, F \in Z_H \) to the basis \( \tilde{\Phi}_{k-1,F} \) and obtain \( \tilde{\Phi}_k,F \), \( \tilde{\Phi}_k = \bigcup_{F \in Z_H} \tilde{\Phi}_{k,F} \).

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 $\Delta$, a maximum basis size $N_{\text{max}}$ and a POD-tolerance $\Delta_{\text{POD}}$.

**Generate basis $\Phi$ of $\mathcal{W}_N$:**

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 $\lambda_k$ be given. Perform detailed simulation to obtain $p_{\lambda_k}^{\lambda_k}$ and define preliminary basis extension by $\tilde{\varphi}_F := p_{\lambda_k}^{\lambda_k}|_F, \forall F \in \mathcal{Z}_H$. 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 $p_{\lambda}^\lambda$ 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 $\Delta$, a maximum basis size $N_{\text{max}}$ and a POD-tolerance $\Delta_{\text{POD}}$.

Generate basis $\Phi$ of $W_N$:

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

1. Let a basis $\tilde{\Phi}_{k-1} = \bigcup_{F \in Z_H} \tilde{\Phi}_{k-1,F}$ and a parameter function $\lambda_k$ be given. Perform detailed simulation to obtain $p_{h}^{\lambda_k}$ and define preliminary basis extension $\tilde{\varphi}_F, F \in Z_H$ by $\tilde{\varphi}_F := p_{h}^{\lambda_k} |_{F}, \forall F \in Z_H$. Add $\tilde{\varphi}_F, F \in Z_H$ to the basis $\tilde{\Phi}_{k-1,F}$ and obtain $\tilde{\Phi}_{k,F}$, $\tilde{\Phi}_k = \bigcup_{F \in Z_H} \tilde{\Phi}_{k,F}$.

2. Compute offline-parts of the DG scheme and of the error estimator for the current basis $\tilde{\Phi}_k$.

3. Compute reduced solutions $p_N^{\lambda}$ 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 $N < N_{\text{max}}$ and if the error bound for the reduced solution $p_N^{\lambda_{k+1}}$ is larger than $\Delta$, continue with Step (1) with $\lambda_{k+1}$ from Step (3).
Adaptive basis construction for $W_N$

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

Generate basis $\Phi$ of $W_N$:

0. Set $\Phi_{-1}, \Phi_{-1,F} := \emptyset$ 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 $\Phi_{k-1} = \bigcup_{F \in \mathcal{Z}_H} \Phi_{k-1,F}$ and a parameter function $\lambda_k$ 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 $\Phi_{k-1,F}$ and obtain $\Phi_{k,F}, \tilde{\Phi}_k = \bigcup_{F \in \mathcal{Z}_H} \Phi_{k,F}$.

2. Compute offline-parts of the DG scheme and of the error estimator for the current basis $\tilde{\Phi}_k$.

3. Compute reduced solutions $p_N^{\varphi}$ for all $\varphi \in \mathcal{M}_{\text{train}}$ 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 $N < N_{\text{max}}$ and if the error bound for the reduced solution $p_N^{\lambda_{k+1}}$ is larger than $\Delta$, continue with Step (1) with $\lambda_{k+1}$ from Step (3).

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

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

with

\[k^\varepsilon(x) := \frac{2}{3}(1 + x_1)(1 + \cos^2(2\pi \frac{x_1}{\varepsilon})) ,\]

\[\lambda(x) := \frac{1}{\eta_o} - \frac{2}{\eta_o} S(x) + \frac{\eta_o + \eta_w^2}{\eta_w \eta_o} \sum_{m,n=1}^{N_S} \mu_n \mu_m S_n(x) S_m(x),\]

\[S(x) := \sum_{n=1}^{N_S} \mu_n S_n(x) \text{ with } N_S = 3 \text{ and } S_n(x) \text{ given.}\]

+ suitable Dirichlet boundary conditions.

M. Ohlberger

Model reduction for multiscale problems
Simulation results

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

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

M. Ohlberger

Model reduction for multiscale problems
CPU times for the new method

| $|\mathcal{T}_h|$ | N | $t_{\text{highdim}}$ (s) | $t_{\text{lowdim}}$ (ms) | $t_{\text{recons}}$ (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 ($t_{\text{highdim}}$ and $t_{\text{lowdim}}$); the reconstruction ($t_{\text{recons}}$) 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/