Westfälische
Wilhelms-Universität
Münster

Reduction of Microscale Li-Ion Battery Models

living.knowledge
WWU Münster

M. Ohlberger,
S. Rave, F. Schindler

Coupled Problems
Venice
May 20, 2015

Outline

- The MULTIBAT Project
- The Reduced Basis Method
- The Localized Reduced Basis Multiscale Method
- Software

The MULTIBAT Project

The MULTIBAT Project

ulm university universität uulm

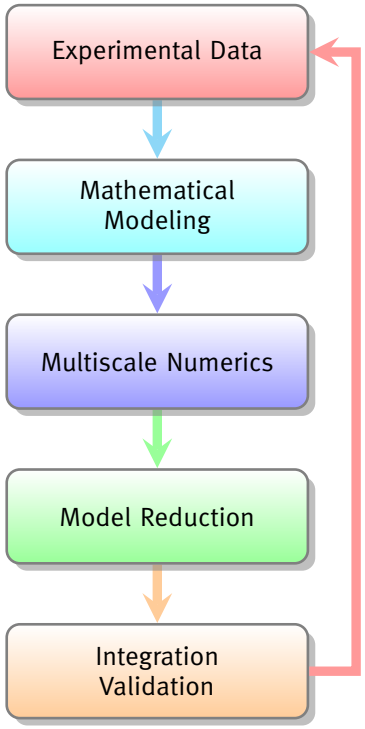
Institute of Technical Thermodynamics
DLR

MULTIBAT

SPONSORED BY THE
Federal Ministry of Education and Research

Fraunhofer ITWM

Westfälische Wilhelms-Universität Münster



- Understand degradation processes in rechargeable Li-Ion Batteries through mathematical modeling and simulation.
- Focus: Li-Plating.

Problem

- Li-plating initiated at interface between active particles and electrolyte.
- Need microscale models which resolve active particle geometry.
- Huge nonlinear discrete models.
  - Cannot be solved at cell scale on current hardware.
  - **Parameter studies extremely expensive, even on small domains.**

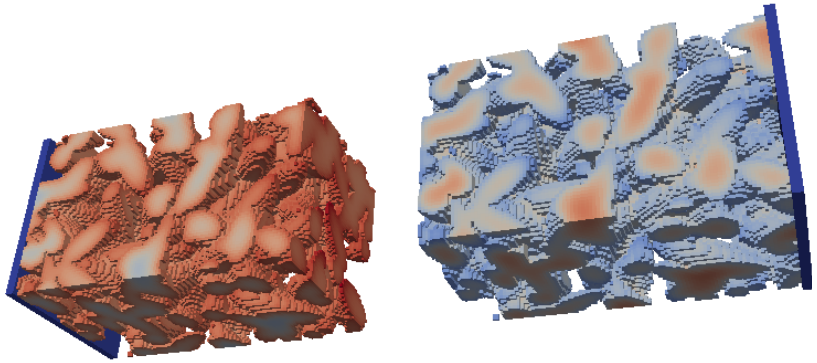


Figure : Simulation of microscale battery model on $246\mu m \times 60\mu m \times 60\mu m$ domain with random electrode geometry.

The Microscale Model

- On each part of domain (electrodes, electrolyte, current collector):

$$\frac{\partial c}{\partial t} - \nabla \cdot (\alpha(c,\phi)\nabla c + \beta(c,\phi)\nabla\phi) = 0 \qquad c : \mathrm{Li}^+ \text{ concentration}$$

$$-\nabla \cdot (\gamma(c,\phi)\nabla c + \delta(c,\phi)\nabla\phi) = 0 \qquad \phi : \text{potential}$$

(α, β, γ, δ constant in first approximation)

- **Coupling:** Normal fluxes at particle/electrolyte interface are given by Butler-Volmer kinetics

$$j_{se} = 2k\sqrt{c_e c_s (c_{max} - c_s)}\sinh\left(\frac{\phi_s - \phi_e - U_0(\frac{c_s}{c_{max}})}{2RT} \cdot F\right)$$

$$N_{se} = \frac{1}{F} \cdot j_{se}.$$

Discretization

- Cell centered finite volume on voxel grid + implicit Euler:

$$\begin{bmatrix} \frac{1}{\Delta t}\left(c_\mu^{(t+1)} - c_\mu^{(t)}\right) \\ 0 \end{bmatrix} + A_\mu\left(\begin{bmatrix} c_\mu^{(t+1)} \\ \phi_\mu^{(t+1)} \end{bmatrix}\right) = 0, \qquad c_\mu^{(t)}, \phi_\mu^{(t)} \in V_h$$

- Numerical fluxes on interfaces = Butler-Volmer fluxes.
- Newton scheme with algebraic multigrid solver.
- Implemented by Fraunhofer ITWM in **BEST**.
- $\mu \in \mathcal{P}$ indicates dependence on model parameters we want to vary (e.g. temperature T, charge rate).

The Reduced Basis Method

Parametric Model Order Reduction

Want to evaluate solution map

$$\Phi : \mathcal{P} \longrightarrow V$$

from compact set $\mathcal{P}$ into normed space V.

Assume we can determine $\Phi(\mu)$ for a **single** μ with lots of effort.

Parametric Model Order Reduction

Want to evaluate solution map

$$\Phi : \mathcal{P} \longrightarrow V$$

from compact set $\mathcal{P}$ into normed space V.

Assume we can determine $\Phi(\mu)$ for a **single** μ with lots of effort.

But we want to

- calculate $\Phi(\mu)$ for **many** $\mu \in \mathcal{P}$.
  (interactive simulation tools, optimization, inverse problems)
- calculate $\Phi(\mu)$ **quickly** for some $\mu \in \mathcal{P}$.
  (embedded systems, Formula 1)

Parametric Model Order Reduction

Want to evaluate solution map

$$\Phi : \mathcal{P} \longrightarrow V$$

from compact set $\mathcal{P}$ into normed space V.

Assume we can determine $\Phi(\mu)$ for a **single** μ with lots of effort.

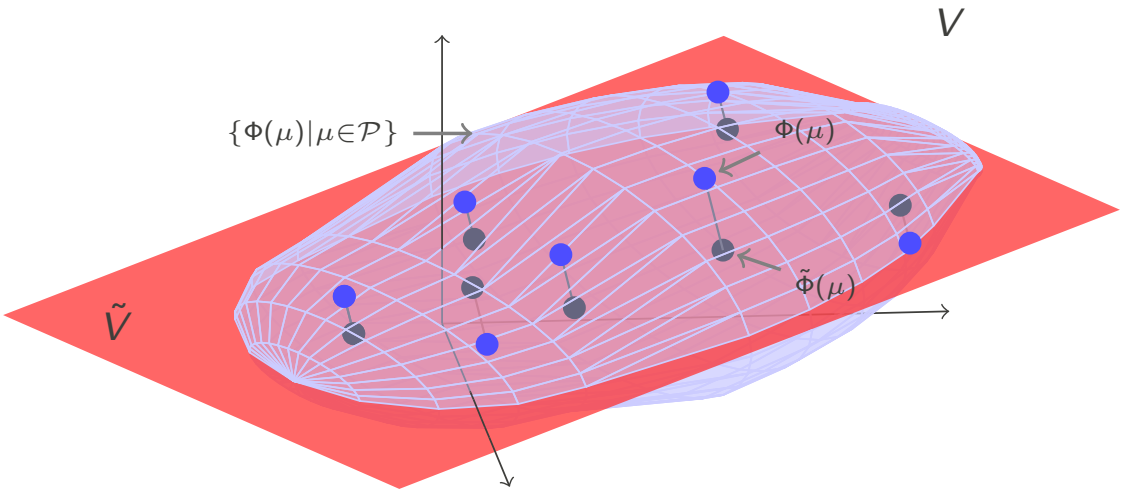
But we want to

- calculate $\Phi(\mu)$ for **many** $\mu \in \mathcal{P}$.
  (interactive simulation tools, optimization, inverse problems)
- calculate $\Phi(\mu)$ **quickly** for some $\mu \in \mathcal{P}$.
  (embedded systems, Formula 1)

Use model order reduction!

Parametric Model Order Reduction

Goal: Quickly evaluate $\Phi : \mathcal{P} \longrightarrow V$



Solution: Build reduced model by finding

1. **problem adapted** low dim. subspace $\tilde{V} \subset V$ for approximating $\Phi(\mathcal{P})$. ($100 \approx \dim \tilde{V} \ll \dim V$)
2. **quickly computable** approximation $\tilde{\Phi} : \mathcal{P} \longrightarrow \tilde{V}$ s.t. $||\Phi(\mu) - \tilde{\Phi}(\mu)|| < \varepsilon$
3. **quickly computable** upper bound $\tilde{\Delta}(\tilde{\Phi}(\mu)) \geq ||\Phi(\mu) - \tilde{\Phi}(\mu)||$.

The Reduced Basis Method

Online phase: Determine reduced solution $\tilde{\Phi}(\mu)$ by solving Galerkin projected equation

$$\begin{bmatrix} \frac{1}{\Delta t}(\tilde{c}_\mu^{(t+1)} - \tilde{c}_\mu^{(t)}) \\ 0 \end{bmatrix} + \{P_{\tilde{V}} \circ A_\mu\}\left(\begin{bmatrix} \tilde{c}_\mu^{(t+1)} \\ \tilde{\phi}_\mu^{(t+1)} \end{bmatrix}\right) = 0, \quad \begin{bmatrix} \tilde{c}_\mu^{(t)} \\ \tilde{\phi}_\mu^{(t)} \end{bmatrix} \in \tilde{V}_c \oplus \tilde{V}_\phi = \tilde{V}.$$

The Reduced Basis Method

Online phase: Determine reduced solution $\tilde{\Phi}(\mu)$ by solving Galerkin projected equation

$$\begin{bmatrix} \frac{1}{\Delta t}(\tilde{c}_\mu^{(t+1)} - \tilde{c}_\mu^{(t)}) \\ 0 \end{bmatrix} + \{P_{\tilde{V}} \circ A_\mu\}\left(\begin{bmatrix} \tilde{c}_\mu^{(t+1)} \\ \tilde{\phi}_\mu^{(t+1)} \end{bmatrix}\right) = 0, \qquad \begin{bmatrix} \tilde{c}_\mu^{(t)} \\ \tilde{\phi}_\mu^{(t)} \end{bmatrix} \in \tilde{V}_c \oplus \tilde{V}_\phi = \tilde{V}.$$

Offline phase: Build $\tilde{V}_c, \tilde{V}_\phi$ using iterative greedy algorithm:

function GREEDY($\mathcal{S}_{train} \subset \mathcal{P}, \varepsilon, \tilde{V}_c^0, \tilde{V}_\phi^0$)
  $\tilde{V}_c, \tilde{V}_\phi \leftarrow \tilde{V}_c^0, \tilde{V}_\phi^0$
  while $\max_{\mu \in \mathcal{S}_{train}}$ ERR-EST(RB-SOLVE(μ), μ) $> \varepsilon$ **do**
    $\mu^* \leftarrow$ arg-max$_{\mu \in \mathcal{S}_{train}}$ ERR-EST(RB-SOLVE(μ), μ)
    $\tilde{V}_c, \tilde{V}_\phi \leftarrow$ BASIS-EXT($\tilde{V}_c, \tilde{V}_\phi$, SOLVE($\mu^*$))
  end while
  return $\tilde{V}_c, \tilde{V}_\phi$
end function

Empirical Interpolation

Problem: Still expensive to evaluate

$$P_{\tilde{V}} \circ A_\mu : \tilde{V}_c \oplus \tilde{V}_\phi \longrightarrow V_h \oplus V_h \longrightarrow \tilde{V}_c \oplus \tilde{V}_\phi .$$

Empirical Interpolation

Problem: Still expensive to evaluate

$$P_{\tilde{V}} \circ A_\mu : \tilde{V}_c \oplus \tilde{V}_\phi \longrightarrow V_h \oplus V_h \longrightarrow \tilde{V}_c \oplus \tilde{V}_\phi .$$

Solution:

- Use locality of finite volume operators: to evaluate M DOFs of $A_\mu(c, \phi)$ need only $M' \leq C \cdot M$ DOFs of (c, ϕ).
- Approximate

$$P_{\tilde{V}} \circ A_\mu \approx P_{\tilde{V}} \circ (I_M \circ \tilde{A}_{M,\mu} \circ R_{M'}) =: P_{\tilde{V}} \circ \mathcal{I}_M[A_\mu]$$

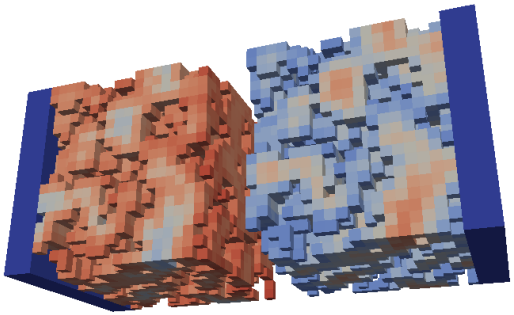
where

$\tilde{A}_{M,\mu}$:	A_μ restricted to M interpolation DOFs
I_M:	Interpolation operator
$R_{M'}$:	Restriction to M' DOFs needed for evaluation

- Use greedy algorithm to determine DOFs and interpolation basis.

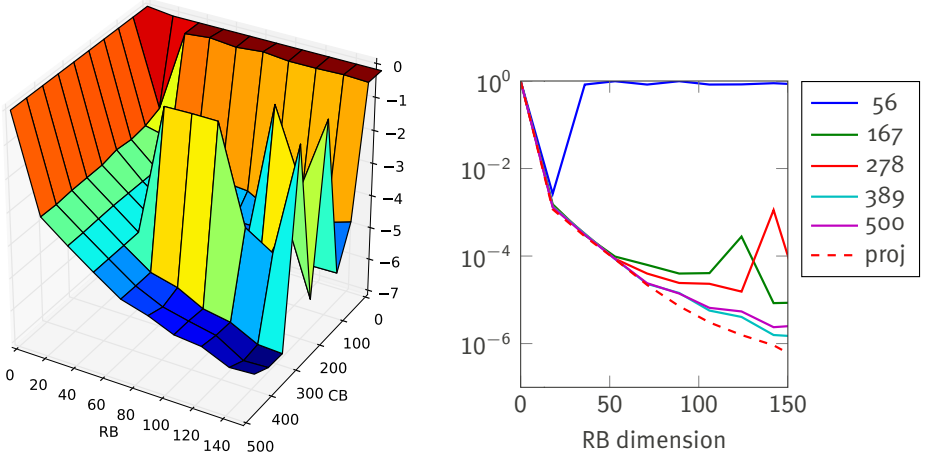
First Results

- Test geometry (36,800 DOFs):

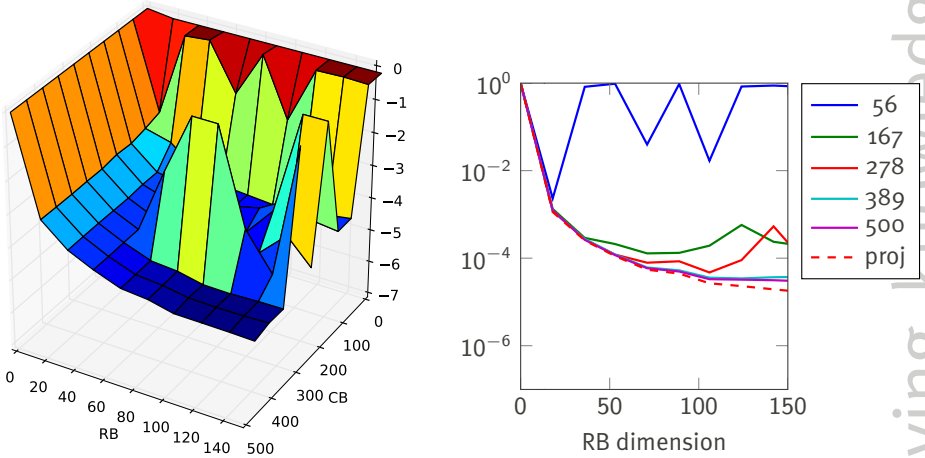


- $\alpha, \beta, \gamma, \delta$ constant.
- Charge rate $\in [0.1C, 1C]$.
- POD for RB generation on 10 snapshots.
- Time for solution $\approx$ 1000s.
- Time for red. solution $\approx$ 40s. (dim $RB = 50$, dim $CB = 278$)

$L^\infty - L^2$ err., concentration, training set



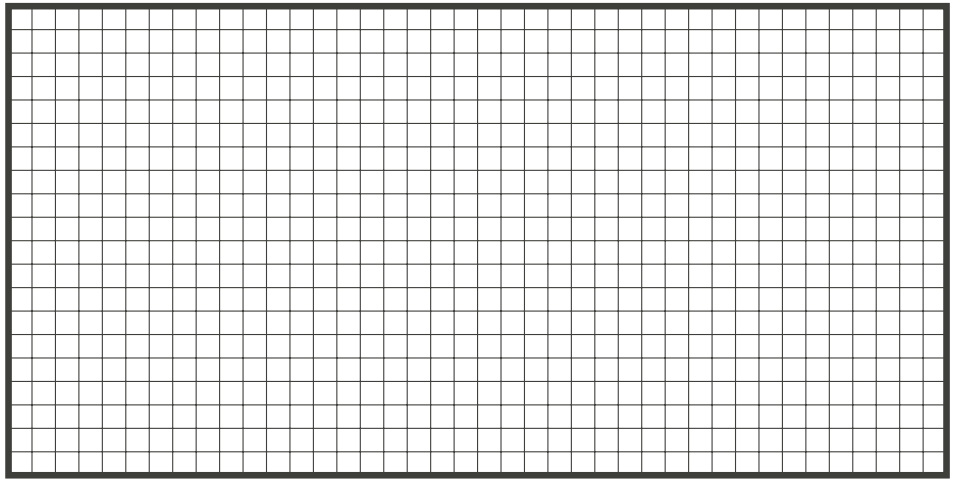
$L^\infty - L^2$ err., concentration, random params.



The Localized Reduced Basis Multiscale Method

The Localized Reduced Basis Multiscale Method

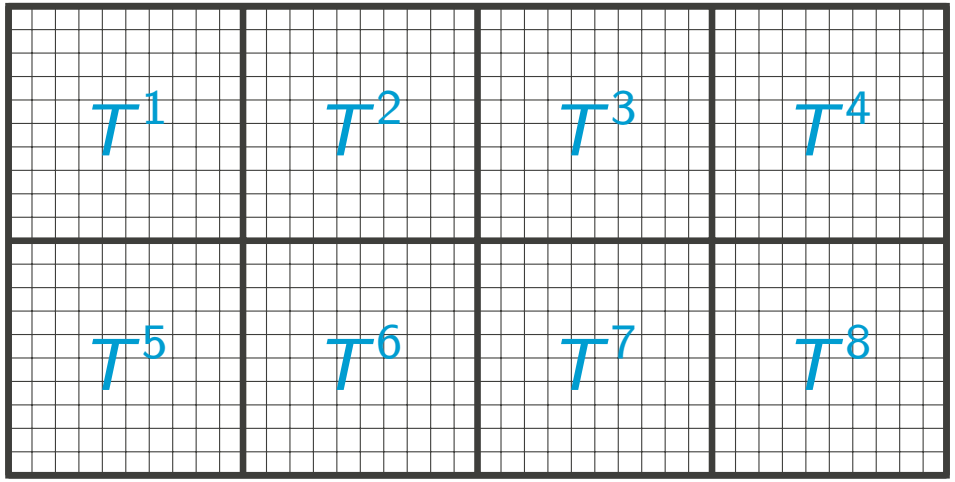
(block) Discontinuous Galerkin discretization (for elliptic problems)



The Localized Reduced Basis Multiscale Method

(block) Discontinuous Galerkin discretization (for elliptic problems)

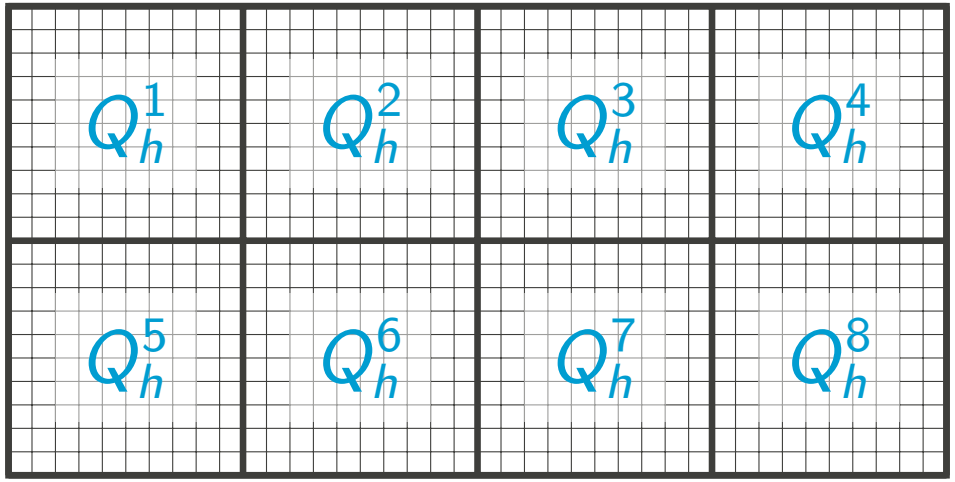
- introduce a coarse triangulation ($\mathcal{T}_H$)



The Localized Reduced Basis Multiscale Method

(block) Discontinuous Galerkin discretization (for elliptic problems)

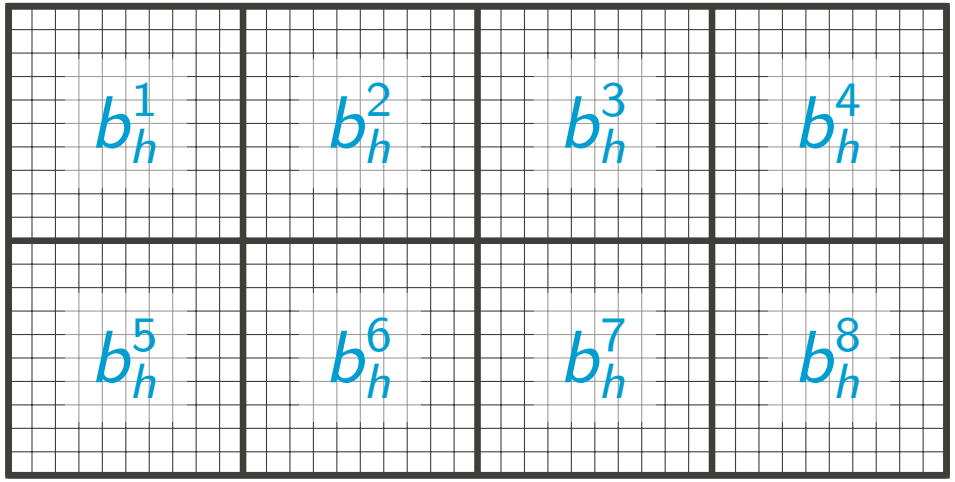
- introduce a coarse triangulation ($\mathcal{T}_H$) $\left(Q_h = \oplus_{T \in \mathcal{T}_H} Q_h^T\right)$
- use your favorite space inside each coarse element



The Localized Reduced Basis Multiscale Method

(block) Discontinuous Galerkin discretization (for elliptic problems)

- introduce a coarse triangulation ($\mathcal{T}_H$) $\left(Q_h = \oplus_{T\in\mathcal{T}_H} Q_h^T\right)$
- use your favorite space and discretization inside each coarse element

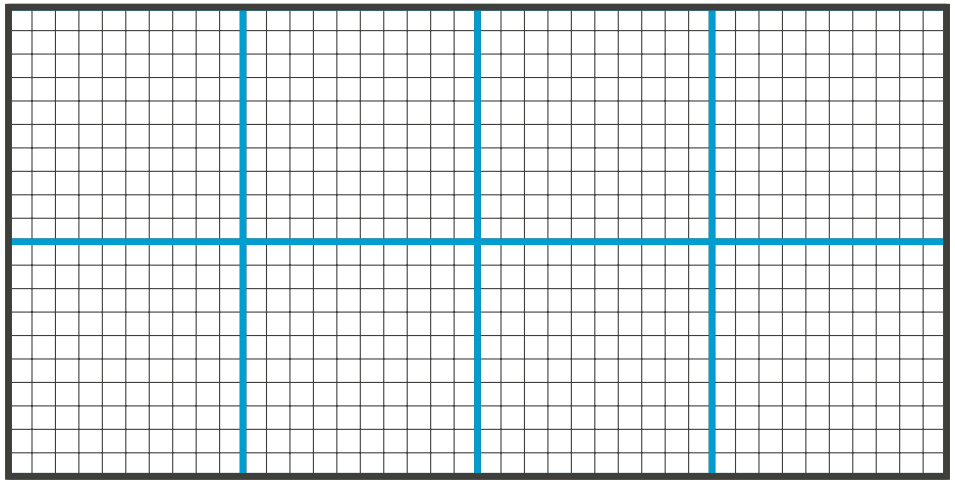


$$b_h = \sum_{T\in\mathcal{T}_H} b_h^T$$

The Localized Reduced Basis Multiscale Method

(block) Discontinuous Galerkin discretization (for elliptic problems)

- introduce a coarse triangulation ($\mathcal{T}_H$) $\left(Q_h = \oplus_{T\in\mathcal{T}_H} Q_h^T\right)$
- use your favorite space and discretization inside each coarse element
- couple with SWIPDG [Ern, Stephansen, Zunino, 2009]

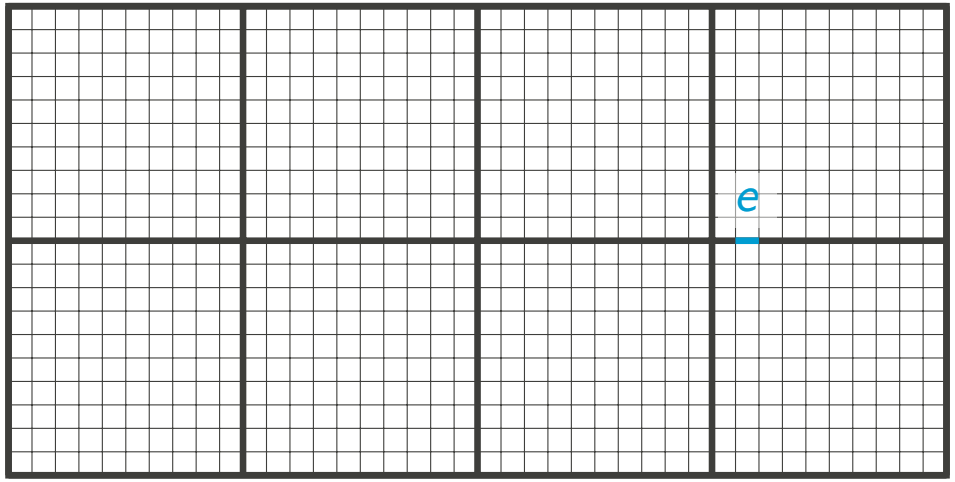


$$b_h = \sum_{T\in\mathcal{T}_H} b_h^T + \sum_{T,S\in\mathcal{T}_H} b_h^{T,S}$$

The Localized Reduced Basis Multiscale Method

(block) Discontinuous Galerkin discretization (for elliptic problems)

- introduce a coarse triangulation ($\mathcal{T}_H$) $\left(Q_h = \oplus_{T\in\mathcal{T}_H} Q_h^T\right)$
- use your favorite space and discretization inside each coarse element
- couple with SWIPDG [Ern, Stephansen, Zunino, 2009]



$$b_h = \sum_{T\in\mathcal{T}_H} b_h^T + \sum_{T,S\in\mathcal{T}_H} b_h^{T,S}$$

$$b_h^{T,S}(p,q)\Big|_e = b_c^e(q,p) + b_c^e(p,q) + b_p^e(q,p)$$

$$b_c^e(p,q) := \int_e -\{\!\{(\lambda\kappa\nabla p)\cdot n_e\}\!\}_e \, [\![q]\!]_e \, \mathrm{d}s$$

$$b_p^e(p,q) := \int_e \sigma_e(\lambda,\kappa)\, [\![p]\!]_e \, [\![q]\!]_e \, \mathrm{d}s$$

The Localized Reduced Basis Multiscale Method

Idea of the LRBMS method

- Find local reduced spaces $Q_{\text{red}}^T \subset Q_h^T$ on each subdomain $T \in \mathcal{T}_H$.
- $Q_{\text{red}} := \oplus_{T \in \mathcal{T}_H} Q_{\text{red}}^T \subset Q_h = \oplus_{T \in \mathcal{T}_H} Q_h^T$.
- Use **localized a posteriori error estimator** to select Q_{red}^T for **online enrichment**.
- Solve **local problems** (on oversampling domain) with old solution at boundary to extend local basis.

The Localized Reduced Basis Multiscale Method

Idea of the LRBMS method

- Find local reduced spaces $Q_{\text{red}}^T \subset Q_h^T$ on each subdomain $T \in \mathcal{T}_H$.
- $Q_{\text{red}} := \oplus_{T \in \mathcal{T}_H} Q_{\text{red}}^T \subset Q_h = \oplus_{T \in \mathcal{T}_H} Q_h^T$.
- Use **localized a posteriori error estimator** to select Q_{red}^T for **online enrichment**.
- Solve **local problems** (on oversampling domain) with old solution at boundary to extend local basis.

Expectations

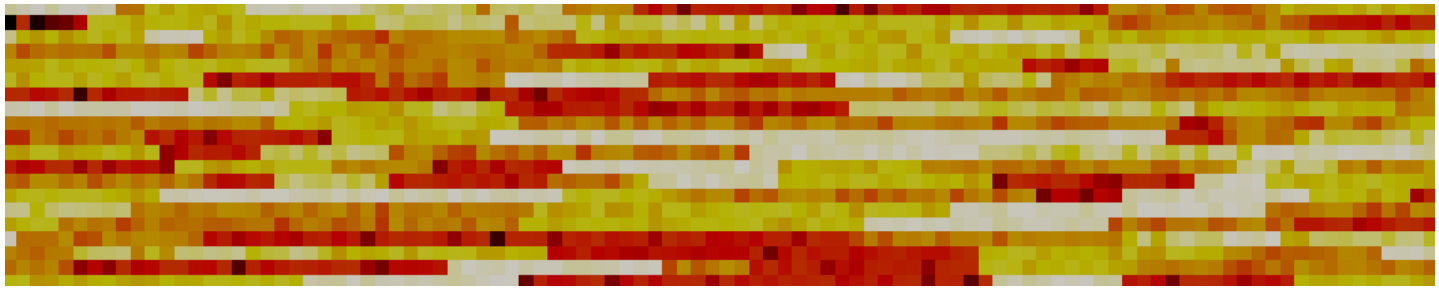
- Increased time-to-solution (online).
- Cheaper basis generation (offline)

Experiment

SPE10: $|\mathcal{T}_h| = 1,014,000$, $|\mathcal{T}_H| = 25 \times 5$ [O., S., 2015]

Solve

$$-\nabla \cdot (\lambda(\mu)\kappa p) = f.$$



$\lambda(1.0)\kappa$

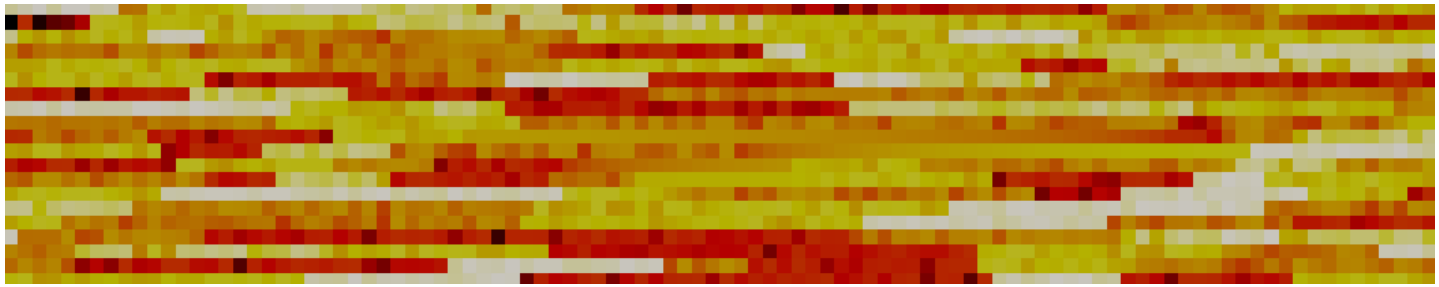
f

Experiment

SPE10: $|\mathcal{T}_h| = 1,014,000$, $|\mathcal{T}_H| = 25 \times 5$ [O., S., 2015]

Solve

$$-\nabla \cdot (\lambda(\mu)\kappa p) = f.$$

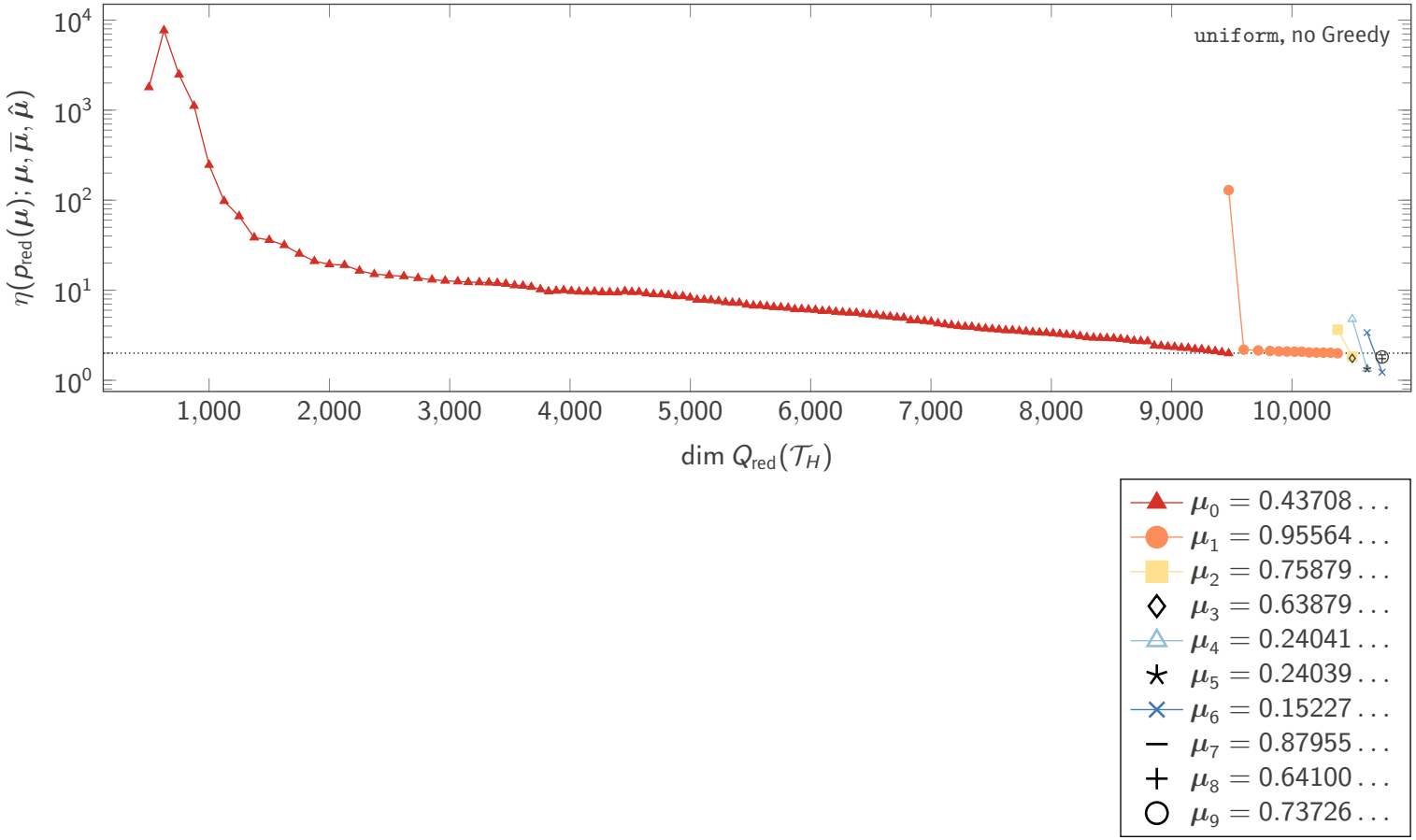


$\lambda(0.1)\kappa$

f

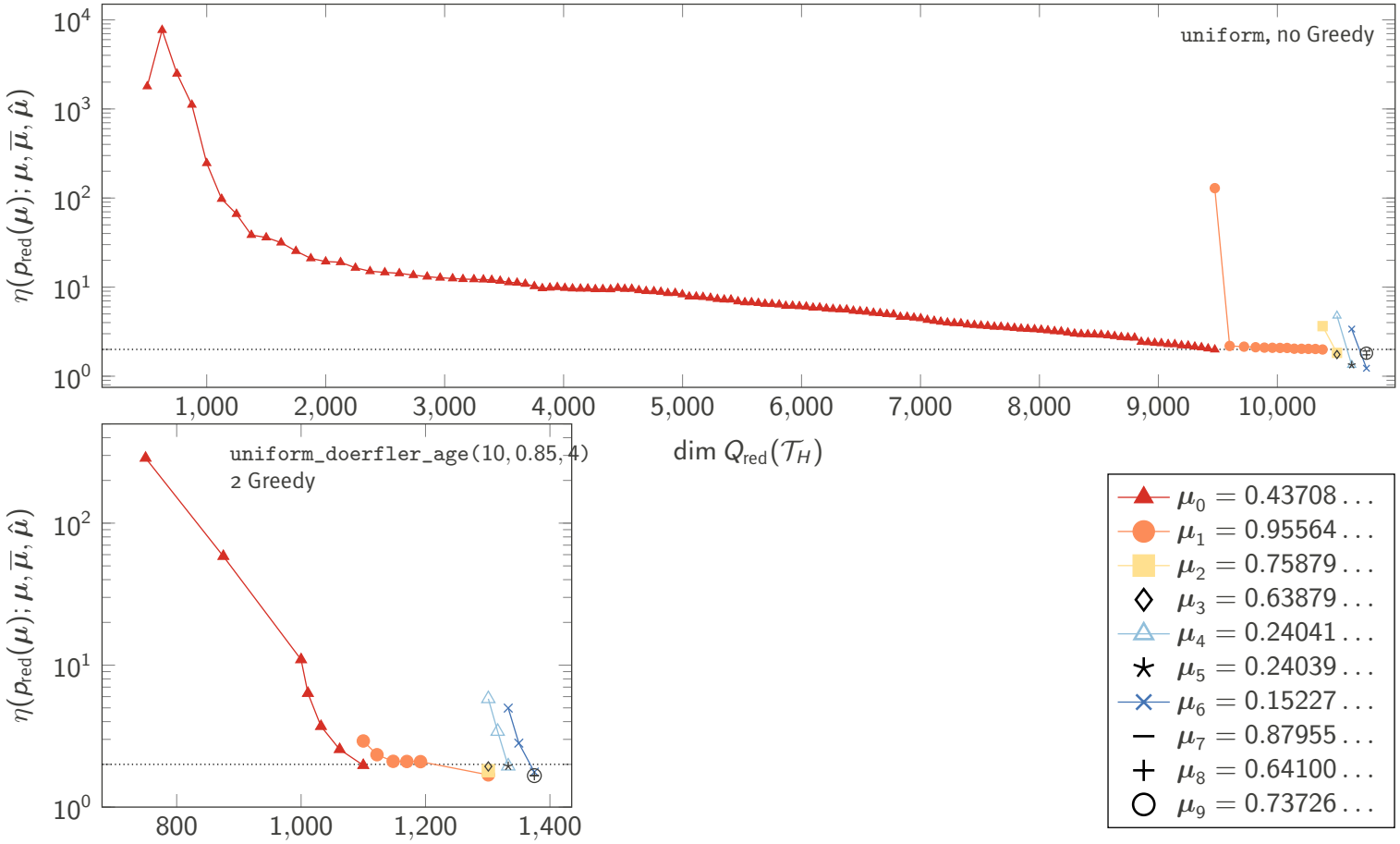
Experiment

SPE10: $|\mathcal{T}_h| = 1,014,000$, $|\mathcal{T}_H| = 25 \times 5$ [O., S., 2015]



Experiment

SPE10: $|\mathcal{T}_h| = 1,014,000$, $|\mathcal{T}_H| = 25 \times 5$ [O., S., 2015]



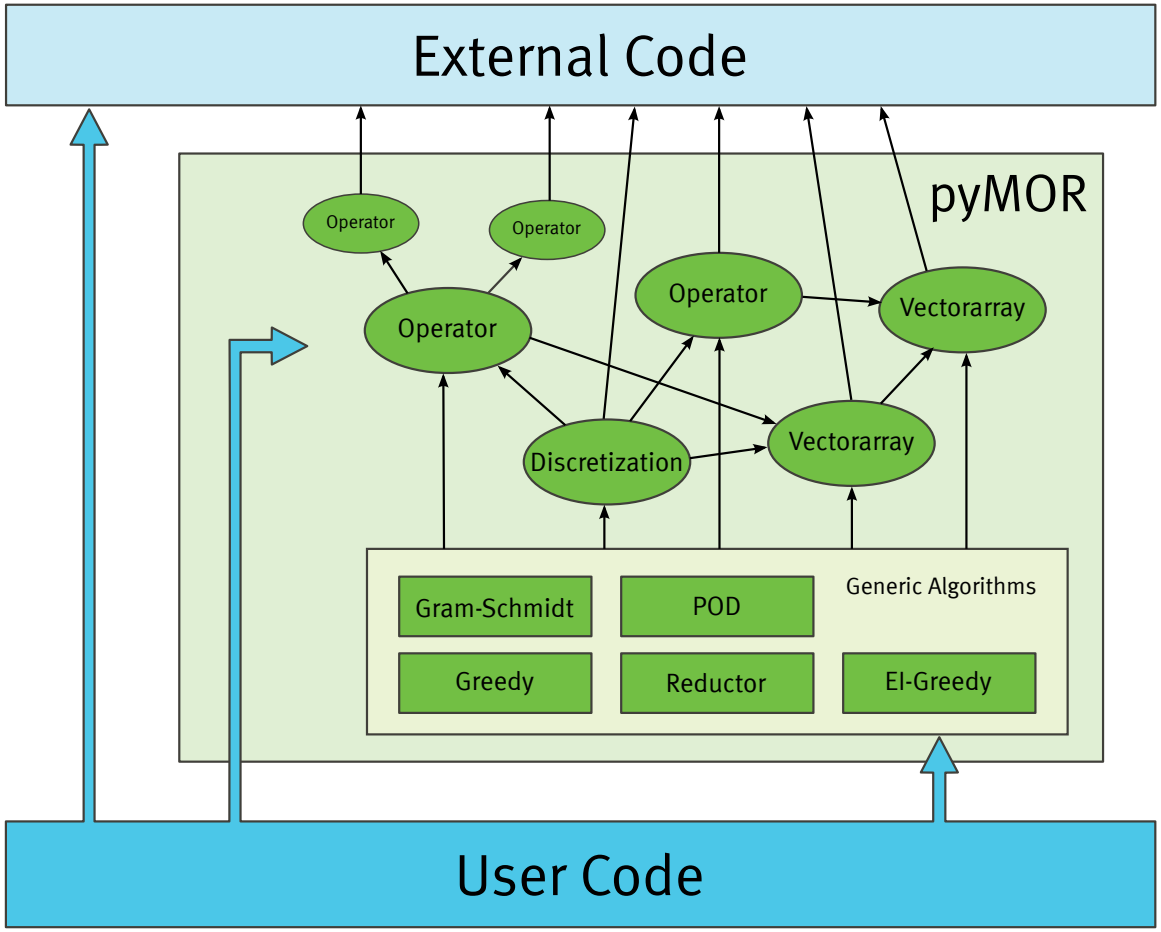
Software

pyMOR

- Software library for writing MOR applications, in particular with the reduced basis method.
- Completely written in Python.
- Started late 2012, 15k lines of code, 2k single commits.
- BSD-licensed, hosted on Github.
- `http://www.pymor.org/`

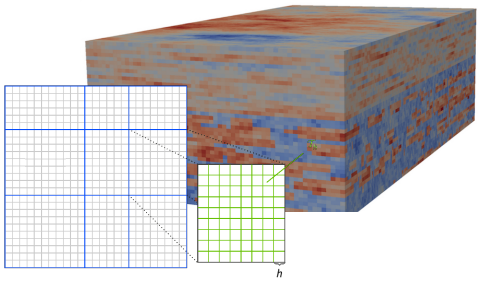
Interfacing external PDE-solvers

with pyMOR

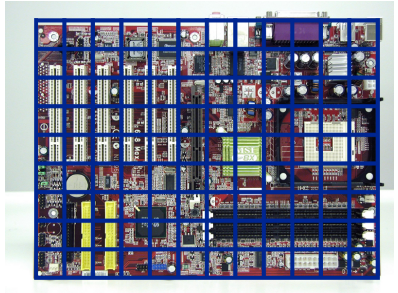


Main Projects

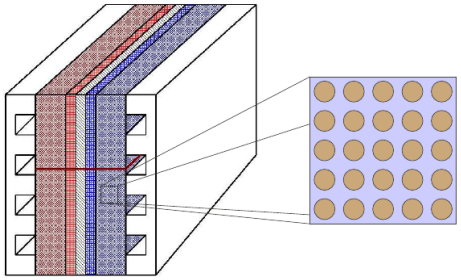
using pyMOR



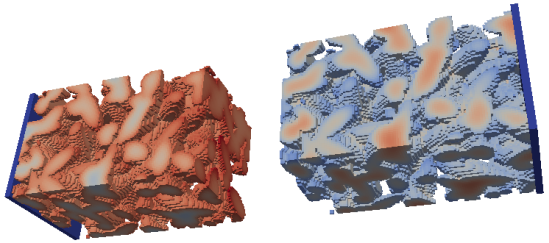
Localized Reduced Basis MultiScale method



Reduction of Maxwell's equations allowing Arbitrary Local Modifications



Reduced basis approximation for multiscale optimization problems



Reduction of microscale Li-ion battery models

Thank you for your attention!

AG Ohlberger
`http://wwwmath.uni-muenster.de/num/ohlberger/`

pyMOR – Model Order Reduction with Python
`http://www.pymor.org/`