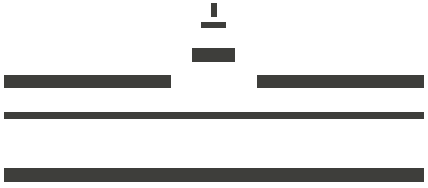


Westfälische
Wilhelms-Universität
Münster

Reduced Basis Approximation of Lithium-Ion Battery Models

living.knowledge
WWU Münster

Stephan Rave

RWTH Aachen
December 8, 2016

Outline

- The Reduced Basis Method in a Nutshell.
- Reduced basis Approximation of Microscale Li-ion Battery Models.
- Software Design.
- Distributed Hierarchical POD Computation.

The Reduced Basis Method in a Nutshell

Parametric Model Order Reduction

Consider parametric problems

$$\Phi : \mathcal{P} \to V, \qquad s : V \to \mathbb{R}^S$$

where

- $\mathcal{P} \subset \mathbb{R}^P$ *compact* set (parameter domain).
- V Hilbert space (solution state space, dim $V \gg 0$, possibly dim $V = \infty$).
- Φ maps parameters to solutions (*hard* to compute).
- s maps state vectors to quantities of interest.

Objective

Compute

$$s \circ \Phi : \mathbb{R}^P \to V \to \mathbb{R}^S$$

for *many* $\mu \in \mathcal{P}$ or *quickly* for unknown single $\mu \in \mathcal{P}$.

The Reduced Basis Method in a Nutshell

Objective

Compute

$$s \circ \Phi : \mathbb{R}^P \to V \to \mathbb{R}^S.$$

- When Φ, s are sufficiently smooth, a quickly computable low-dimensional approximation of $s \circ \Phi$ should exist.
- **Idea 1:** State space reduction:
  - Define approximation $\Phi_N : \mathcal{P} \to V_N$ via Galerkin projection, $\dim V_N =: N \ll \dim V$.
  - Approximate $s \circ \Phi \approx s \circ \Phi_N$.
- **Idea 2:** $V_N \subseteq \operatorname{span}\{\Phi(\mu_1), \dots, \Phi(\mu_k)\}$.
- **Idea 3:** Construct V_N iteratively via greedy search of $\mathcal{P}$ using quickly computable surrogate $\eta_N(\Phi_N(\mu), \mu) \geq \|\Phi(\mu) - \Phi_N(\mu)\|$.

The Easiest Case

Full order problem

$\Phi(\mu) = u_\mu \in V$ is the solution of variational problem

$$a_\mu(u_\mu, v) = f(v) \qquad \forall v \in V,$$

where $a_\mu : V \times V \to \mathbb{R}$ is continuous, coercive bilinear form, $f \in V'$.

Reduced order problem

For given $V_N \subset V$, let $\Phi_N(\mu) := u_{\mu,N} \in V_N$ be the Galerkin projection of u_μ onto V_N, i.e.

$$a_\mu(u_{\mu,N}, v) = f(v) \qquad \forall v \in V_N.$$

- Since a_μ is coercive, $u_{\mu,N}$ is well-defined.

Error Estimates

Theorem (Céa)

Let c_μ denote the coercivity constant of a_μ. Then

$$\|u_\mu - u_{\mu,N}\| \leq \frac{\|a_\mu\|}{c_\mu} \inf_{v \in V_N} \|u_\mu - v\|.$$

Proposition

The quantity $\Delta_\mu(u_{\mu,N}) := c_\mu^{-1} \cdot \|f(\cdot) - a_\mu(u_{\mu,N}, \cdot)\|_{-1}$ is a reliable and efficient a posteriori estimate for the model reduction error:

$$\|u_\mu - u_{\mu,N}\| \leq \Delta_\mu(u_{\mu,N}) \leq \frac{\|a_\mu\|}{c_\mu^{-1}} \cdot \|u_\mu - u_{\mu,N}\|.$$

Offline-/Online-Decomposition

Affinely decomposed bilinear form

$$a_\mu = \sum_{q=1}^{Q} \theta_q(\mu) \cdot a_q \qquad \forall \mu \in \mathcal{P},$$

Proposition

Let $\varphi_1, \cdots, \varphi_N$ be a basis of V_N. If $\left[a_q(\varphi_l, \varphi_k)\right]_{k,l}$ are precomputed, the reduced problem can be solved with effort $\mathcal{O}(QN^2 + N^3)$. Δ_μ can be evaluated with effort $\mathcal{O}(Q^2N^2)$.

Reduced Basis Approximation of Microscale Li-Ion Battery Models

The MULTIBAT Project

ulm university universität ulm

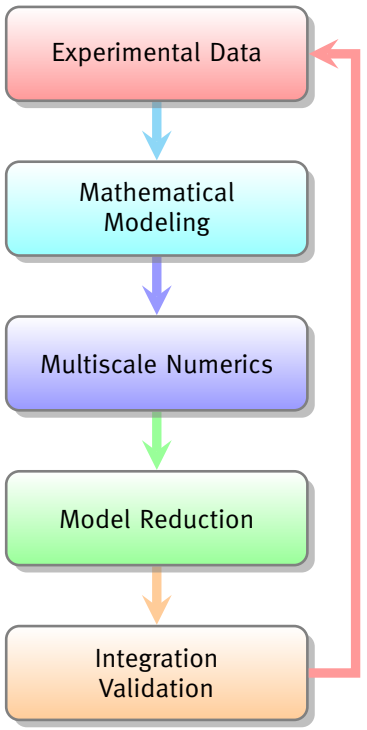
DLR Institute of Technical Thermodynamics

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 Setting

- 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.**

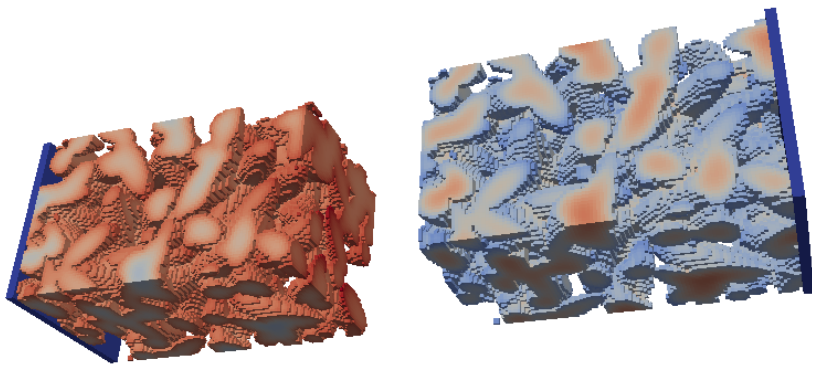
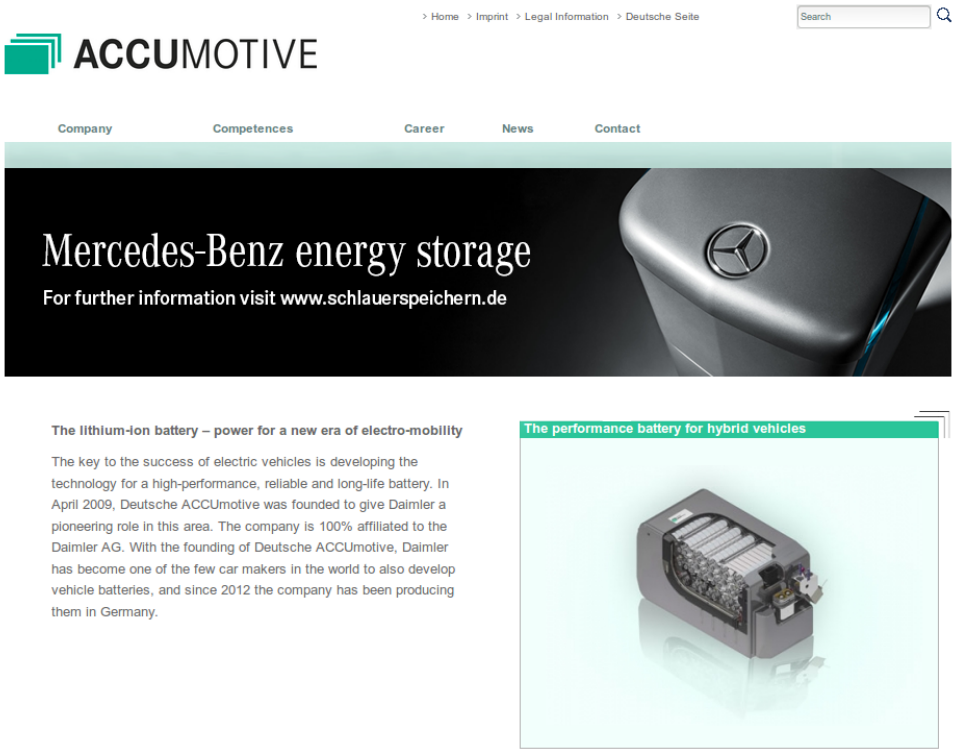


Figure: Simulation of microscale battery model on $246\mu m \times 60\mu m \times 60\mu m$ domain with random electrode geometry.

Our Industry Partner

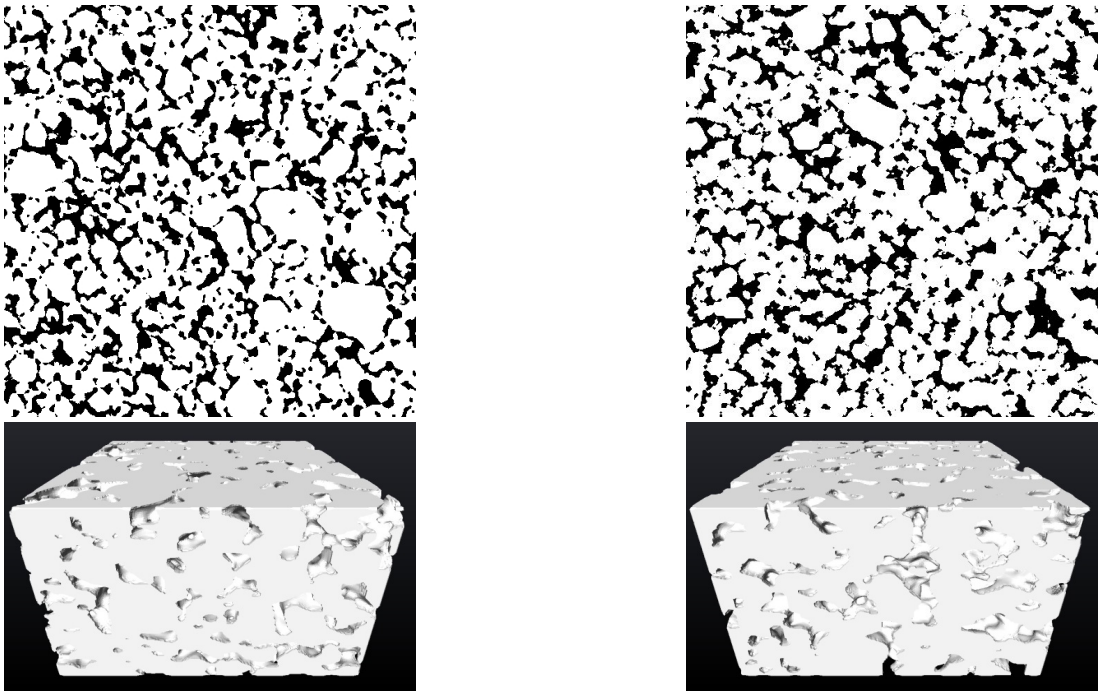


Provides:

- synchrotron imaging data of battery electrodes
- industrial insights

Imaging and Stochastic Structure Modeling

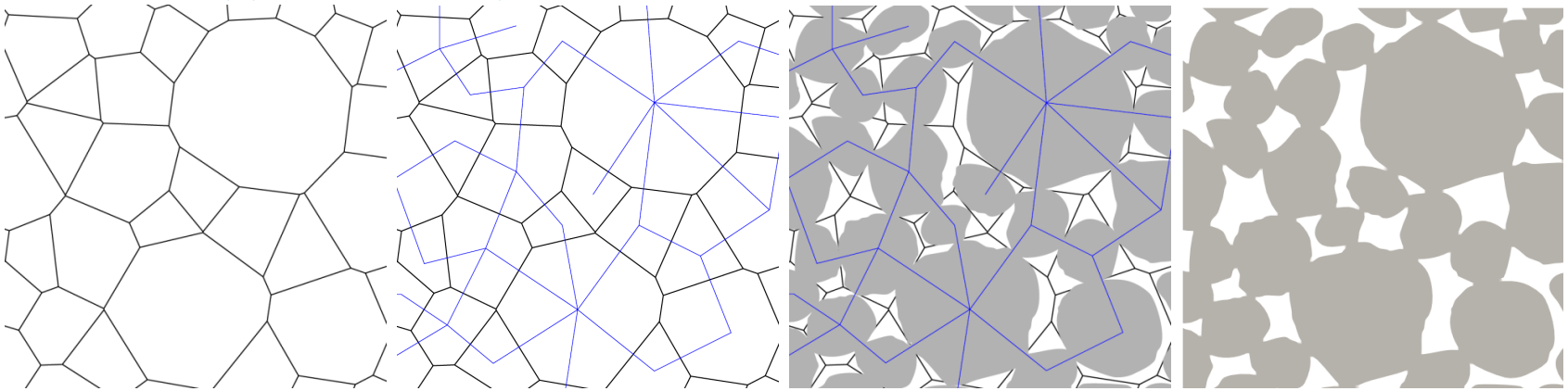
Feinauer, Schmidt, Westhoff (Ulm, Accumotive)



- Visual comparison of 2D and 3D cut-outs of experimental data (left) and simulated (right) shows good agreement.

Imaging and Stochastic Structure Modeling

Modeling Approach: Complete Simulation Model



- Create realization φ of the random Laguerre tesselation.
- Construct the connectivity graph.
- For each Laguerre cell $C \in \varphi$:
  - Define constraints $A \cdot c = b$ for particle placed in centroid x of C.
  - Sample coefficients c that fulfill $A \cdot c = b$ form $\mathcal{N}(\mu, \Sigma)$.
  - Reconstruct particle from coefficients c.
- Smooth structure with morphological closing.

Microscale Modeling

Hein, Latz (DLR at Helmholtz Institute Ulm)

Variables:

c : Li^+ concentration ϕ : electrical potential

Electrolyte:

$$\frac{\partial c}{\partial t} - \nabla \cdot (D_e \nabla c) = 0$$

$$-\nabla \cdot (\kappa \frac{1 - t_+}{F} RT \frac{1}{c} \nabla c - \kappa \nabla \phi) = 0$$

Electrodes:

$$\frac{\partial c}{\partial t} - \nabla \cdot (D_s \nabla c) = 0$$

$$-\nabla \cdot (\sigma \nabla \phi) = 0$$

Coupling: Normal fluxes at interfaces given by Butler-Volmer kinetics

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

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

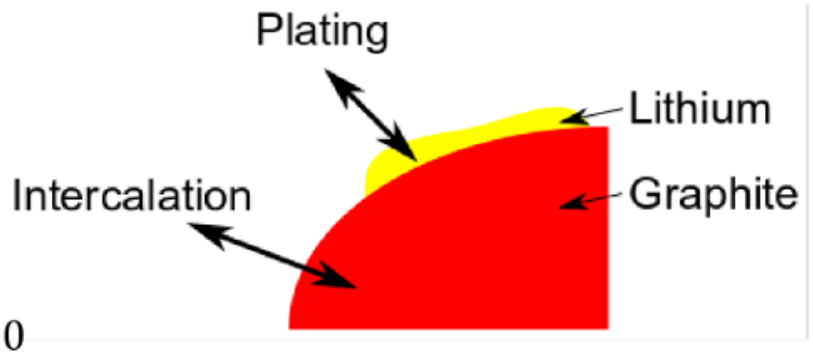
Microscale Modeling – Lithium Plating

Two possible reaction at negative electrode (Graphite):

- Intercalation $\mathrm{Li}^{+}_{\mathrm{Electrolyte}} + \mathrm{e}^{-}_{\mathrm{Solid}} \rightleftharpoons \mathrm{LiC}_{6,\mathrm{Solid}}$
- Lithium plating $\mathrm{Li}^{+}_{\mathrm{Electrolyte}} + \mathrm{e}^{-}_{\mathrm{Solid}} \rightleftharpoons \mathrm{Li}^{\Theta}_{\mathrm{Solid}}$

Overpotential with lithium reference:

- $\eta_{\mathrm{i}} = \Phi_{\mathrm{Solid}} - \varphi^{\mathrm{Li}^{+}}_{\mathrm{Electrolyte}} - U_0(c_{\mathrm{Solid}})$
- $\eta_{\mathrm{p}} = \Phi_{\mathrm{Solid}} - \varphi^{\mathrm{Li}^{+}}_{\mathrm{Electrolyte}}$



Lithium plating if $\eta_{\mathrm{p}} \leq 0$ $\quad \eta_{\mathrm{i}} + U_0(c_{\mathrm{So}}) \leq 0$

Active material and Electrolyte

$$i_{\mathrm{Inter}} = i_{\mathrm{I},0}\left(\exp\left[\frac{F}{2RT}\eta_i\right] - \exp\left[-\frac{F}{2RT}\eta_i\right]\right)$$

$$i_{\mathrm{I},0} = i_{\mathrm{I},00} \cdot \sqrt{c_{\mathrm{E}} \cdot c_{\mathrm{S}} \cdot \left(c_{\mathrm{S}}^{\max} - c_{\mathrm{S}}\right)}$$

Plated Lithium and Electrolyte

$$i_{\mathrm{Li}} = i_{\mathrm{Li},0}\left(\exp\left[\frac{F}{2RT}\eta_{\mathrm{Li}}\right] - \exp\left[-\frac{F}{2RT}\eta_{\mathrm{Li}}\right]\right)$$

$$i_{\mathrm{Li},0} = i_{\mathrm{Li},00} \cdot \sqrt{c_E}$$

Discretization

Iliev, Schmidt, Zausch (Fraunhofer ITWM)

Full Order Model

Cell centered finite volume discretization on voxel grid + implicit Euler leads to nonlinear equation systems of the form:

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

- 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 (e.g. temperature T, charge rate).

Model Order Reduction

Ohlberger, R

Reduced Order Model

Find $[\tilde{c}_\mu^{(t)}, \tilde{\phi}_\mu^{(t)}] \in \tilde{V}_c \oplus \tilde{V}_\phi = \tilde{V}$ solving projected equation

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

- **Basis generation:** POD of a priori selected solution trajectories, separately for c and ϕ (different scales).
- **POD (=Proper Orthogonal Decomposition):** truncated singular value decomposition of snapshot matrix (a.k.a. principal component analysis)
- **Future:**
  - efficient a posteriori error bound $\rightarrow$ POD-Greedy
  - localized MOR ($\rightarrow$ LRBMS, ArbiLoMod)

Empirical Operator 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

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

Empirical Operator Interpolation (2)

Empirical Operator Interpolation (EIM, $\approx$ DEIM)

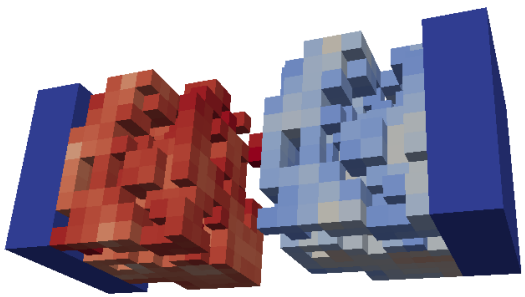
$$P_{\tilde{V}} \circ A_\mu \approx P_{\tilde{V}} \circ (I_M \circ \tilde{A}_{M,\mu} \circ R_{M'}) =: P_{\tilde{V}} \circ \mathcal{I}_M[A_\mu]$$

Basis Generation:

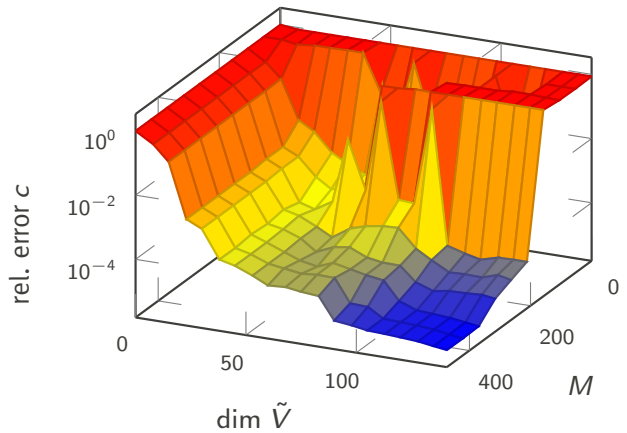
- Compute operator evaluations on solution snapshots (including Newton stages).
- Iteratively extend interpolation basis with worst-approximated evaluation. Choose new interplation DOF where new vector is maximal (EI-Greedy).
- Interpolate Butler-Volmer part of A_μ and $1/c \cdot \nabla c$ separately (ϕ-part of A_μ vanishes for solutions).
- Future: Build RB and interplation basis simultaneously using error estimator to select snapshots (POD-EI-Greedy).

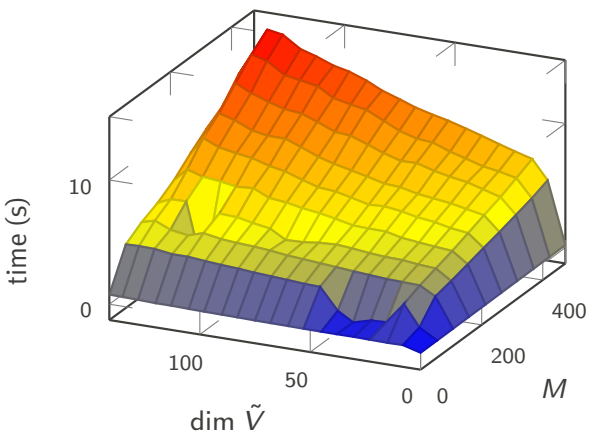
Experiments

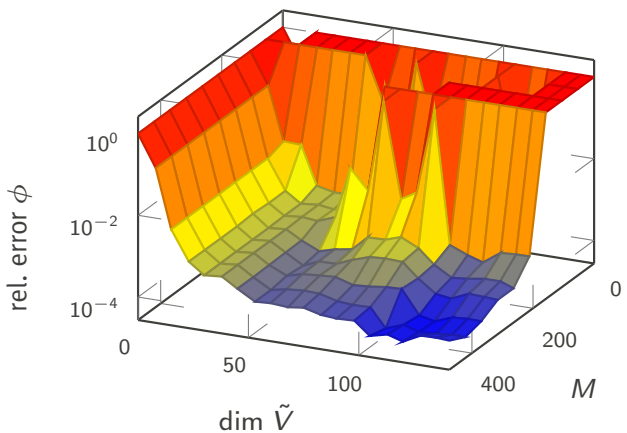
Small Geometry without Plating



- 4.600 DOFs, 20 snaphots
- $T = 298K$, $I \in [0.1C, 1C]$

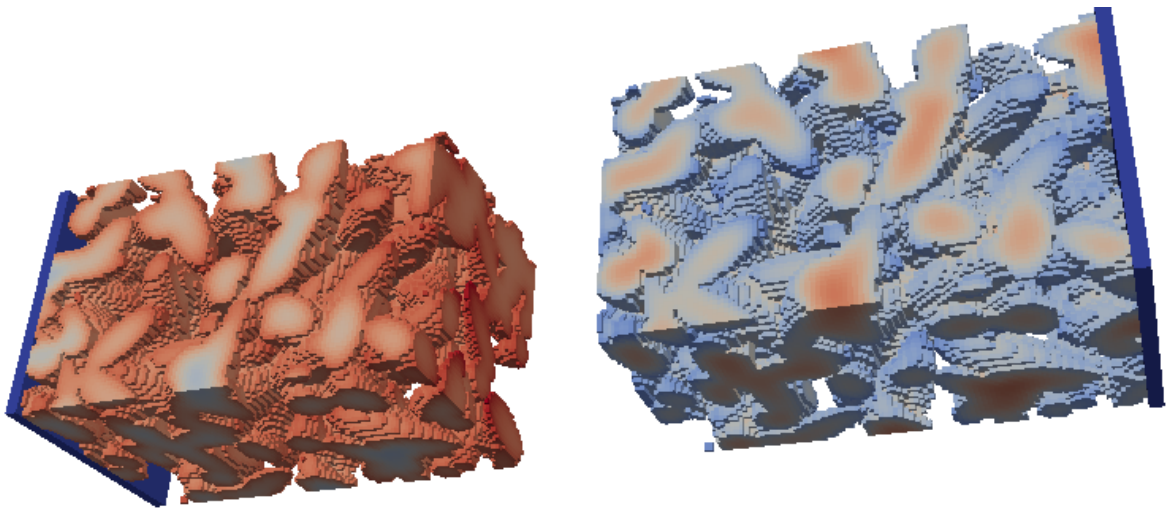






Experiments

Large Geometry without Plating

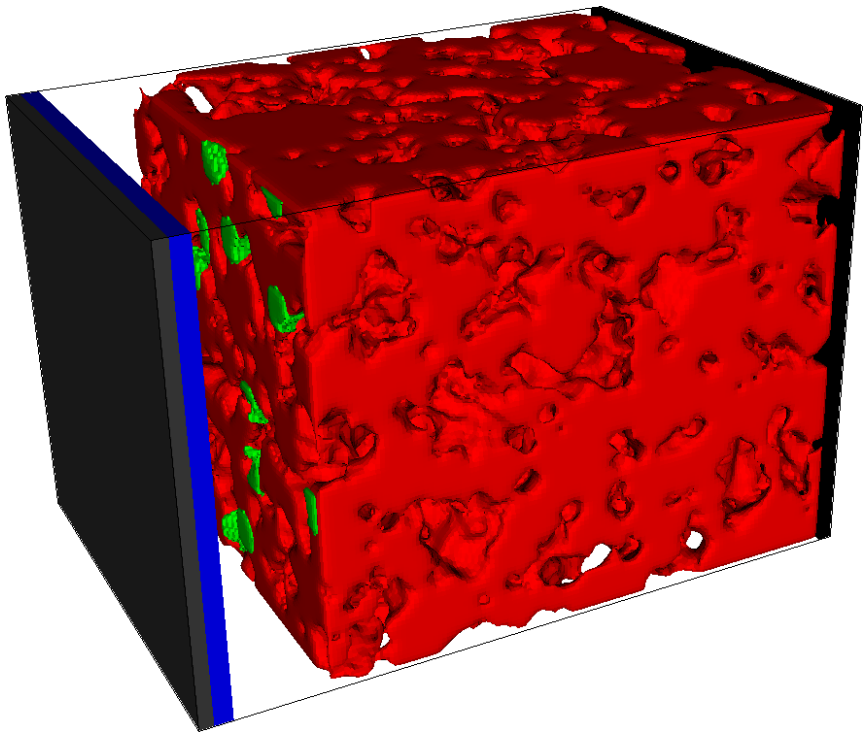


- 1.749.600 DOFs, solution time: 6.5h.
- 2 solution snapshots.

dim $\tilde{V}$	11	21	30	40
rel. error c	$9.26 \cdot 10^{-3}$	$3.96 \cdot 10^{-3}$	$3.05 \cdot 10^{-3}$	$2.93 \cdot 10^{-3}$
rel. error ϕ	$2.07 \cdot 10^{-3}$	$1.50 \cdot 10^{-3}$	$1.46 \cdot 10^{-3}$	$1.26 \cdot 10^{-3}$
time (s)	82	81	79	81
speedup	279	285	290	283

Experiments

Large Geometry with Plating

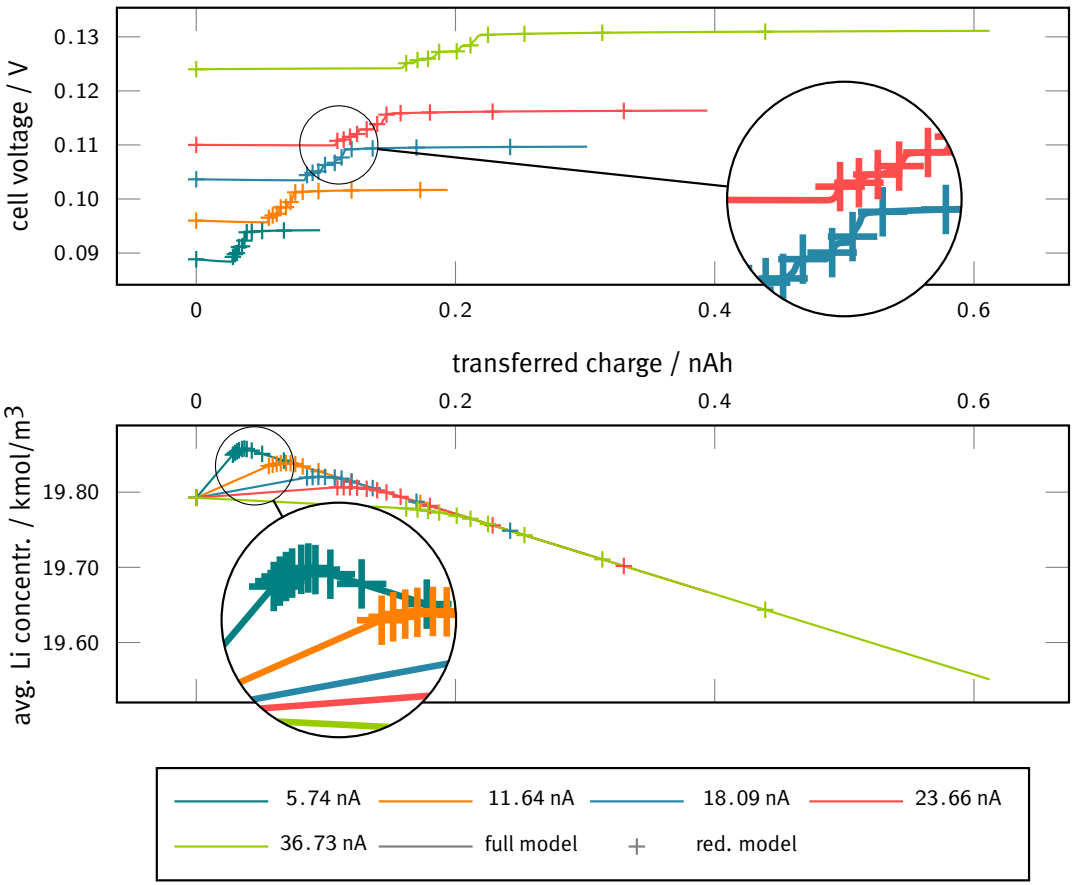


- 3D microstructure based on the stochastic generation algorithm developed in MULTIBAT.
- Half-cell with lithium metal as cathode.
- 2.920.000 DOFs.
- Plated lithium shown as green spots on the separator-graphite-interface.
- Lithium stripping during delithiation of electrode.
- Adaptive time stepping due to faster time scale during stripping.

Experiments

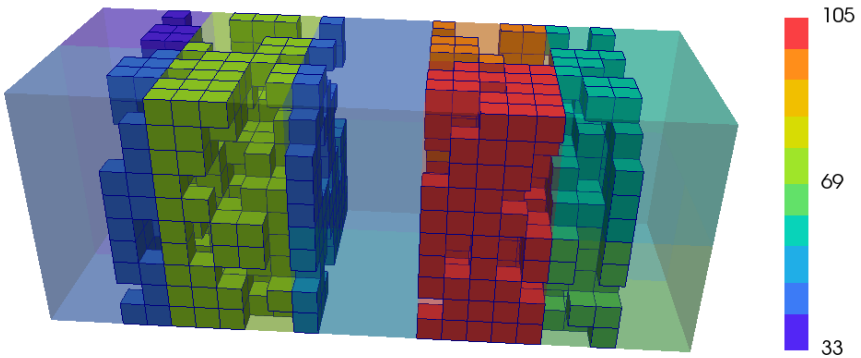
Large Geometry with Plating

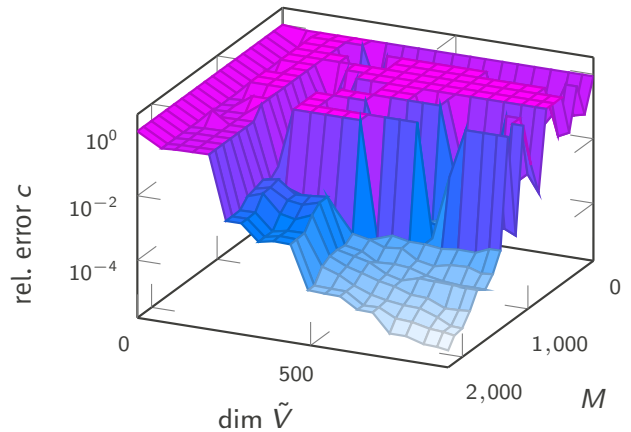
- Snapshots: 3
- dim $\tilde{V} = 98 + 47$
- $M = 710 + 774$
- Rel. err.: $< 1.5 \cdot 10^{-3}$
- Full model: $\approx$ 13h
- Reduction: $\approx$ 9h
- Red. model: $\approx$ 5m
- Speedup: 154

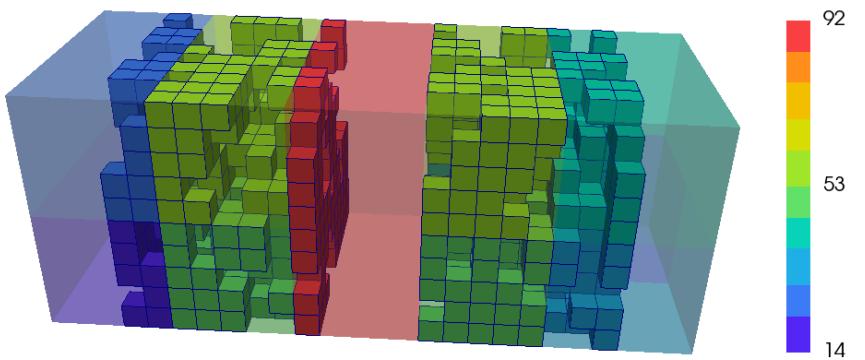


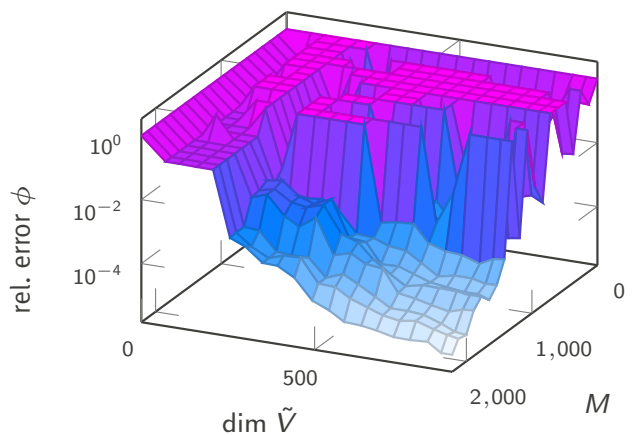
Localized Reduced Basis Approximation

- First experiments on small geometry (no enrichment, no parallelization).



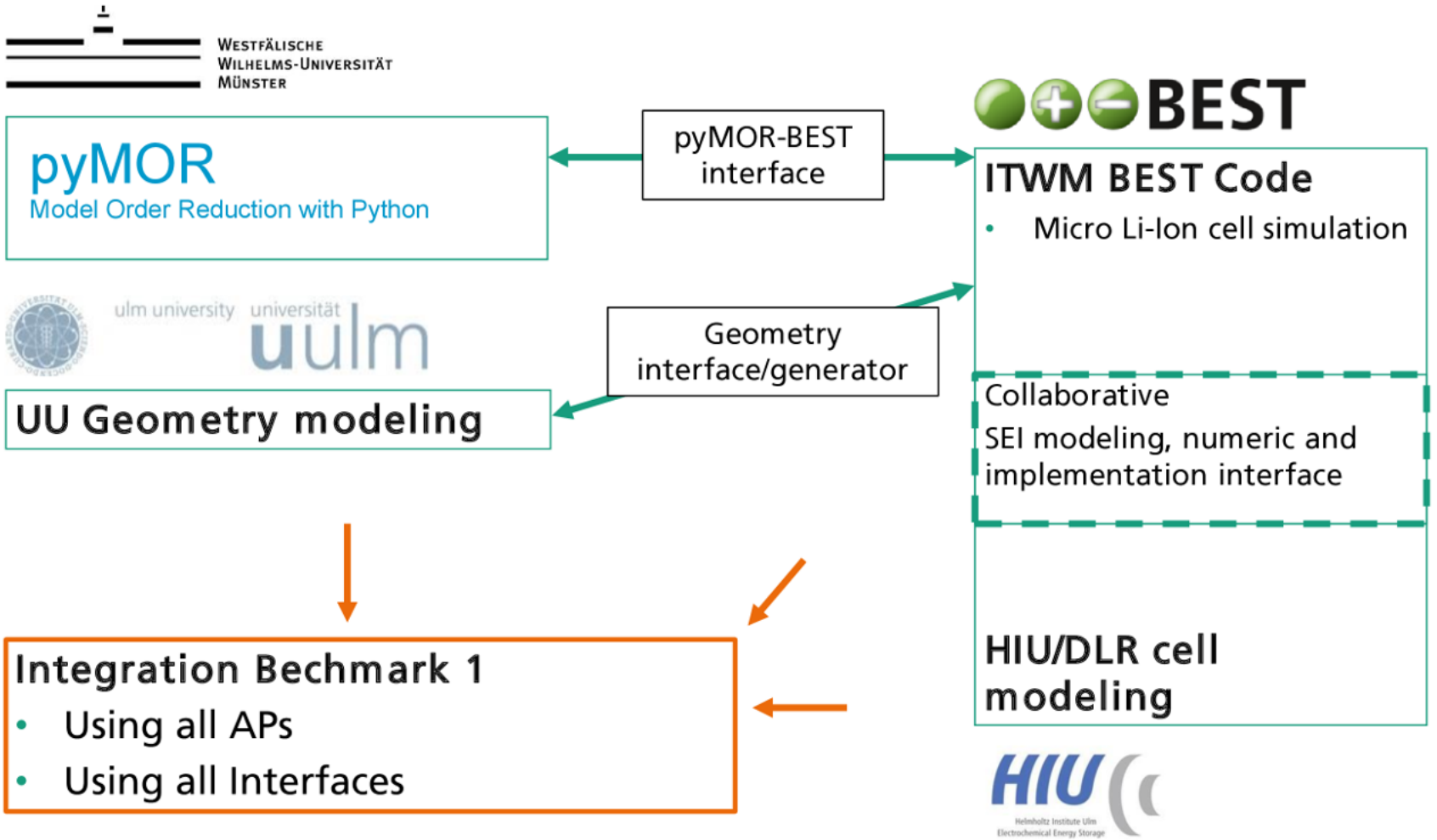






Software Design

Software Interfaces in MULTIBAT



Software Interfaces in MULTIBAT

Interfaces allow us to:

- easily exchange Dune solver with BEST.
- independently develop MOR algorithms.
- easily apply MOR algorithms to updated models in BEST.
- reuse MOR algorithms for other problems.

- Using all APs
- Using all Interfaces

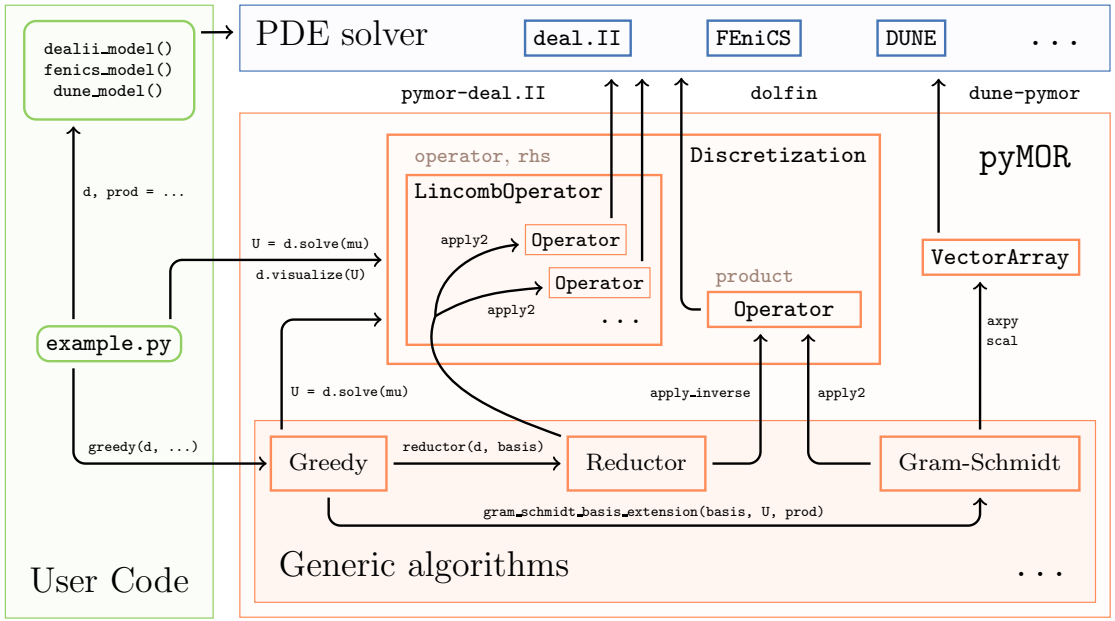
pyMOR – Model Order Reduction with Python

Goal

One tool for algorithm development *and* large-scale applications.

- Started late 2012, 20k lines of Python code, 3k single commits.
- BSD-licensed, fork us on Github!
- Quick prototyping with Python 2/3.
- Comes with small `NumPy`/`SciPy`-based discretization toolkit for getting started quickly.
- Seamless integration with high-performance PDE solvers.

Interfacing External Solvers



- `VectorArray`, `Operator`, `Discretization` classes represent objects in solver's memory.
- No communication of high-dimensional data.
- Possible implementations:
  - Build solver as Python extension module.
  - Communicate with solver via network protocol.

FEniCS Support Included

- Directly interfaces FEniCS LA backend, no copies needed.
- Use same MOR code with both backends!
- Only 150 SLOC for bindings.
- Thermal block demo:
  30 SLOC FEniCS +
  15 SLOC wrapping for pyMOR.
- Easily increase FEM order, etc.

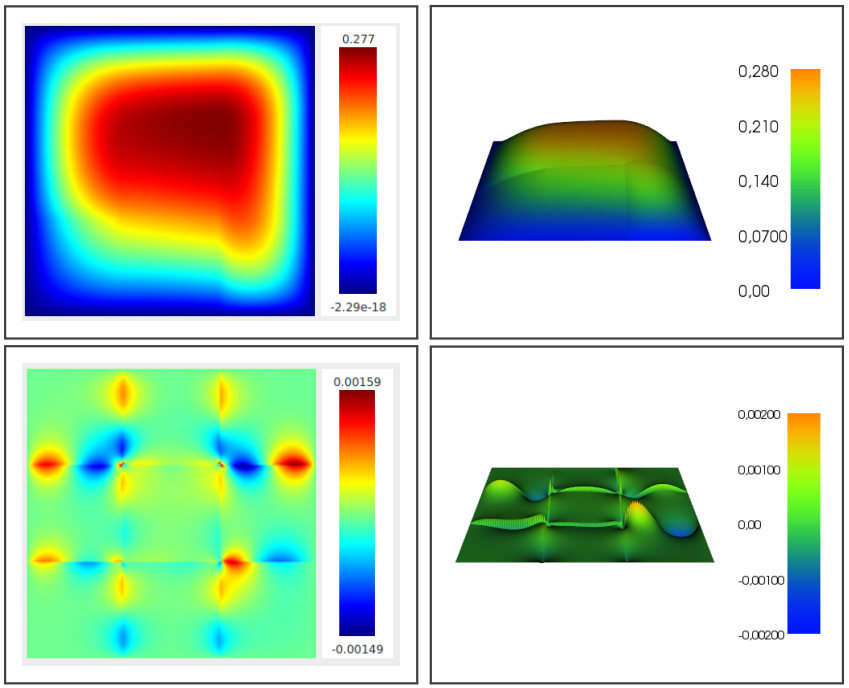


Figure: 3x3 thermal block problem
top: red. solution, bottom: red. error
left: pyMOR solver, right: FEniCS solver

deal.II Support

- `pymor-deall.II` support module (prototype)
- `https://github.com/pymor/pymor-deal.II`
- Python bindings for
  - `dealii::Vector`,
  - `dealii::SparseMatrix`.
- pyMOR wrapper classes.
- MOR demo for linear elasticity example from tutorial.

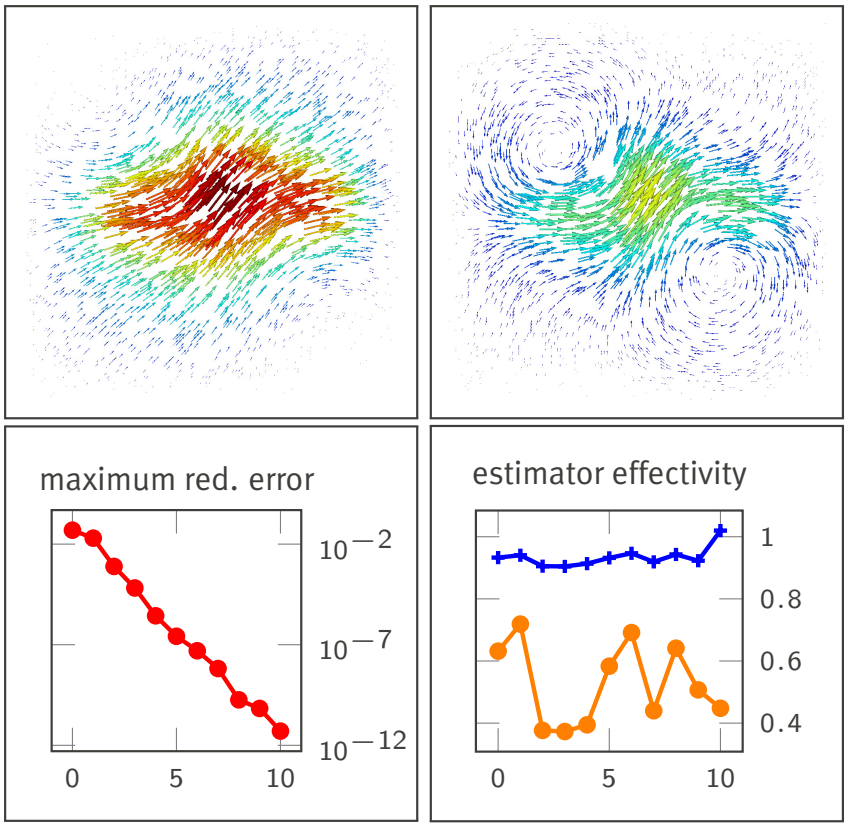


Figure: top: Solutions for $(\mu, \lambda) = (1, 1)$ and $(\mu, \lambda) = (1, 10)$, bottom: red. errs. and max./min. estimator effectivities vs. dim V_N.

Upcoming: NGSolve Support Included

- Based on `NGS-Py` Python bindings for NGSolve.
- Check out `ngsolve` branch of pyMOR repo.
- pyMOR wrappers for vector and matrix classes.
- 3d thermal block demo included.
- Joint work with Christoph Lehrenfeld.

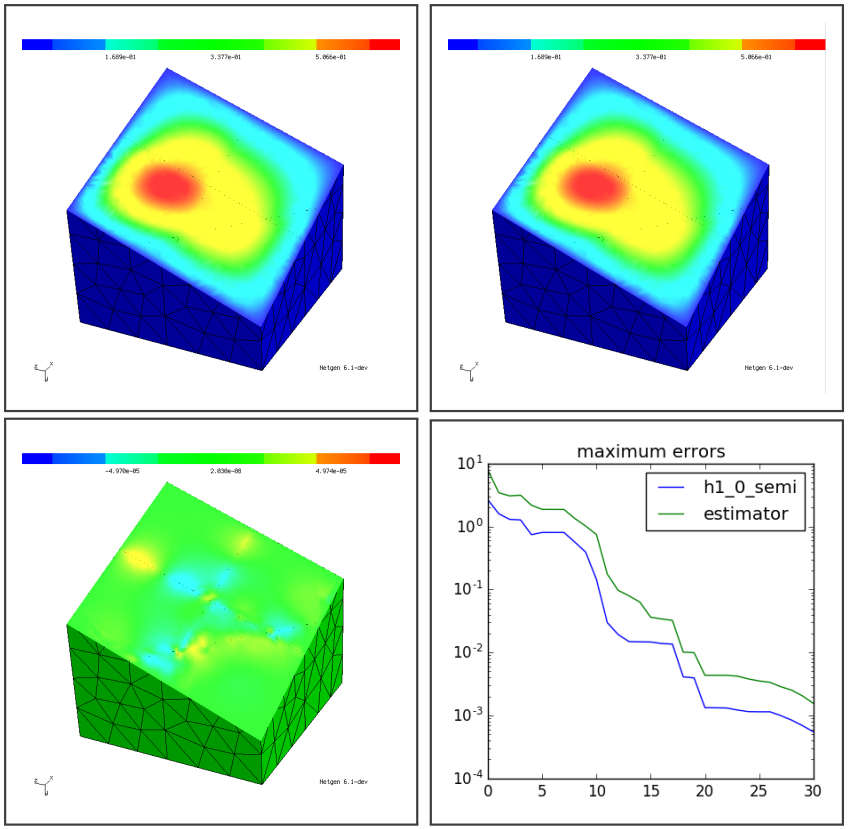


Figure: 3d thermal block problem top: full/red. sol., bottom: err. for worst approx. μ and max. red. error vs. dim V_N.

Tools for interfacing MPI parallel solvers

- Automatically make sequential bindings MPI aware.
- Reduce HPC-Cluster models without thinking about MPI at all.
- Interactively debug MPI parallel solvers.

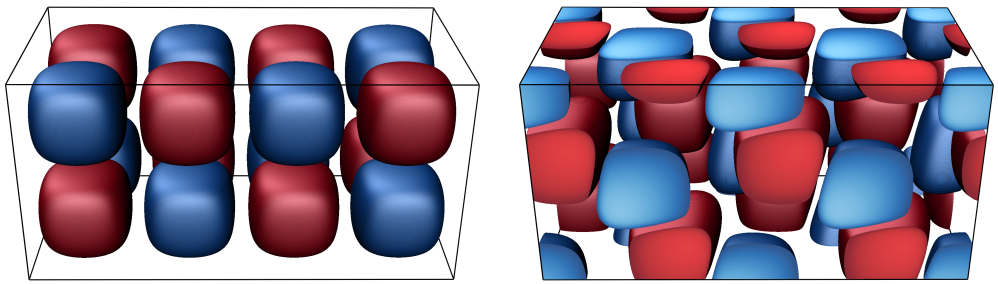


Figure: FV solution of 3D Burgers-type equation ($27.6 \cdot 10^6$ DOFs, 600 timesteps) using Dune.

Table: Time (s) needed for solution using `DUNE` / `DUNE` with `pyMOR` timestepping.

`MPI` ranks	1	2	3	6	12	24	48	96	192
`DUNE`	17076	8519	5727	2969	1525	775	395	202	107
`pyMOR`	17742	8904	6014	3139	1606	816	418	213	120
overhead	3.9%	4.5%	5.0%	5.7%	5.3%	5.3%	6.0%	5.4%	11.8%

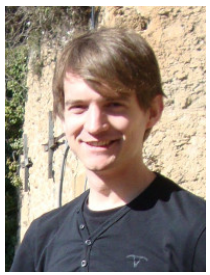
pyMOR Developers and Contributors


Rene Milk

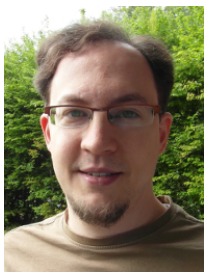

Petar Mlinarić


Stephan Rave


Felix Schindler


Andreas Buhr

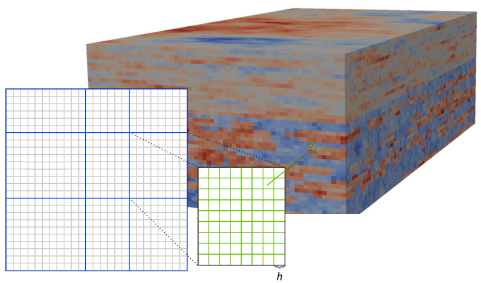

Michael Laier


Michael Schaefer

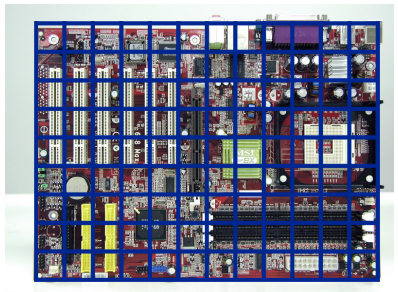

Mario Ohlberger

Some Projects

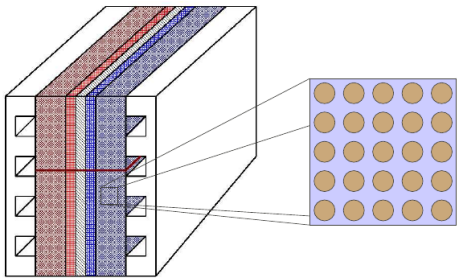
using pyMOR



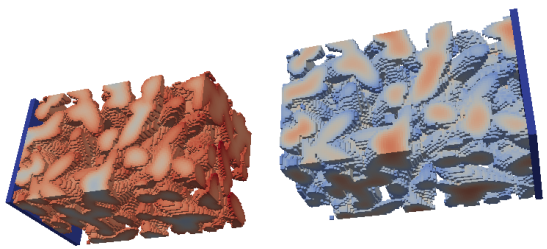
Localized Reduced Basis MultiScale method



Reduction of Maxwell's equations allowing
Arbitrary Local Modifications



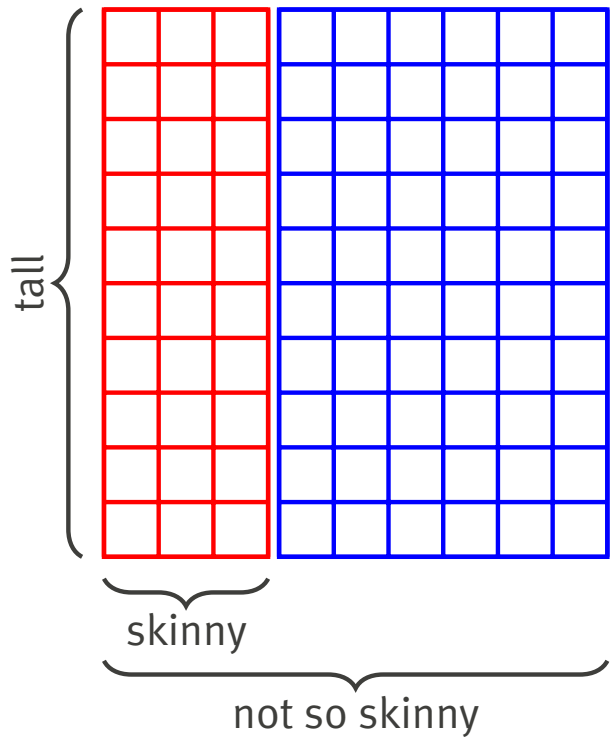
Reduced basis approximation for multiscale
optimization problems



Reduction of microscale Li-ion battery models

Distributed Hierarchical POD Computation

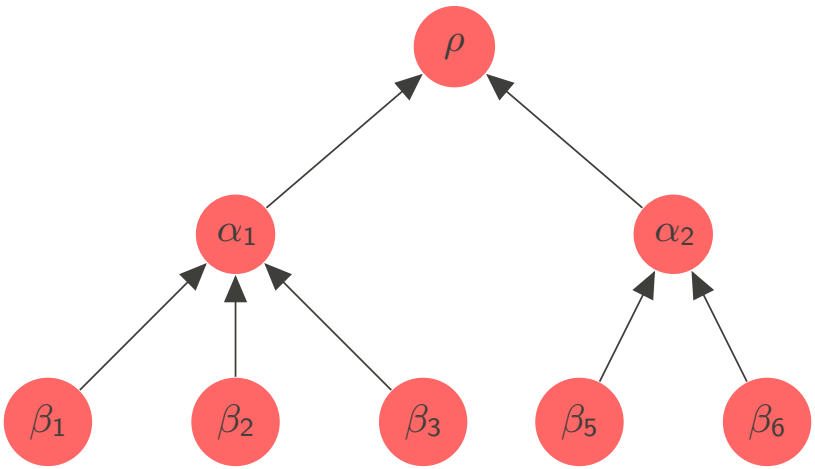
Are your tall and skinny matrices not so skinny anymore?



- Computational effort for POD grows quickly with increasing number of snapshots.
- Hard to parallelize.
- Really slow if data does not fit into RAM.
- **Idea:** PODs of PODs!

HAPOD – Hierarchical Approximate POD

Himpe, Leibner, R



- Input: Assign snapshot vectors to leaf nodes β_i as input.
- At each node:
  1. Perform POD of input vectors with given local error tolerance.
  2. Scale POD modes by singular values.
  3. Send scaled modes to parent node as input.
- Output: POD modes at root node ρ.

HAPOD – Hierarchical Approximate POD

Theorem (Error and mode bounds)

Choose local POD error tolerances $\varepsilon_{\mathcal{T}}$ for l^2-mean approximation error as:

$$\varepsilon_{\mathcal{T}}(\rho) := \frac{\sqrt{|\mathcal{S}|}}{\sqrt{\mathcal{I}_\rho}} \cdot \omega \cdot \varepsilon^*, \qquad \varepsilon_{\mathcal{T}}(\alpha) := \frac{\sqrt{|\mathcal{S}_\alpha|}}{\sqrt{\mathcal{I}_\alpha \cdot (L-1)}} \cdot \sqrt{1-\omega^2} \cdot \varepsilon^*.$$

Then:

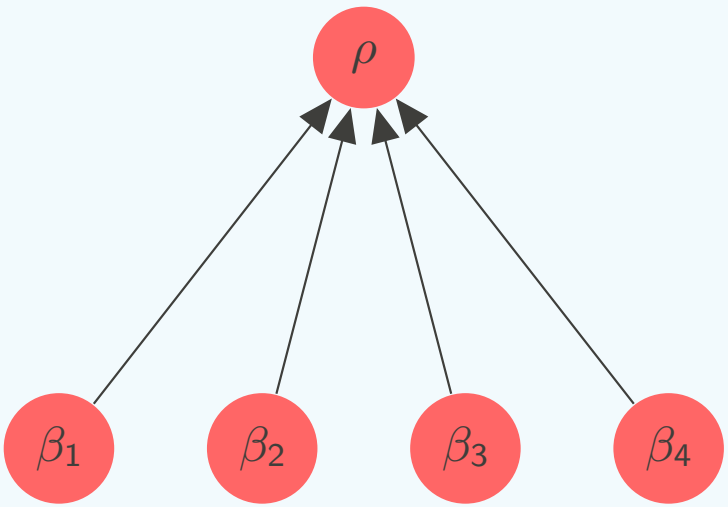
$$\frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \|s - P(s)\|^2 \leq (\varepsilon^*)^2 \quad \textit{and} \quad |\operatorname{HAPOD}[\mathcal{S}, \varepsilon_{\mathcal{T}}]| \leq |\operatorname{POD}(\mathcal{S}, \omega \cdot \varepsilon^*)|.$$

Moreover:

$$\begin{aligned} |\operatorname{HAPOD}[\mathcal{S}, \varepsilon_{\mathcal{T}}](\alpha)| &\leq |\operatorname{POD}(\mathcal{S}_\alpha, (L-1)^{-1/2} \cdot \sqrt{1-\omega^2} \cdot \epsilon^*)| \\ &\leq \min_{N \in \mathbb{N}} (d_N(\mathcal{S}) \leq (L-1)^{-1/2} \cdot \sqrt{1-\omega^2} \cdot \epsilon^*). \end{aligned}$$

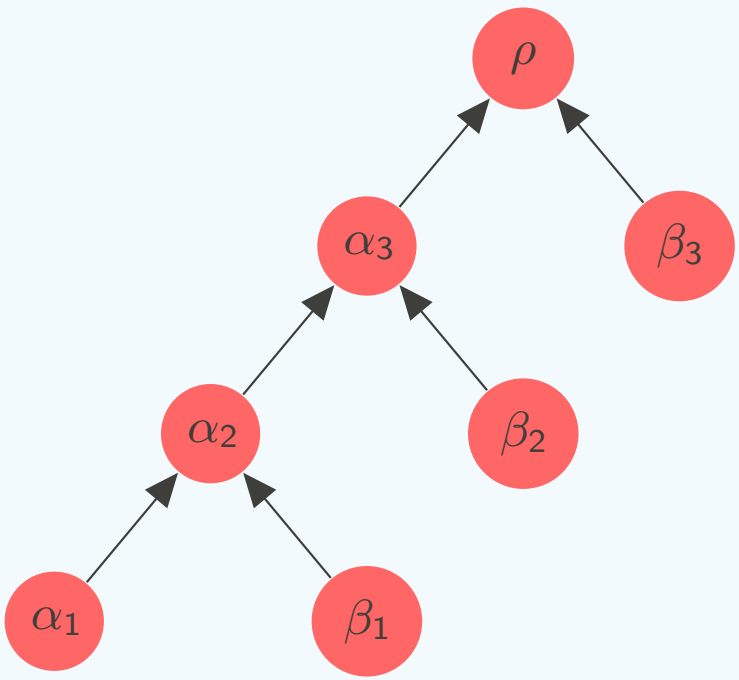
HAPOD – Special Cases

Distributed HAPOD



- Local and parallel snapshot compression.

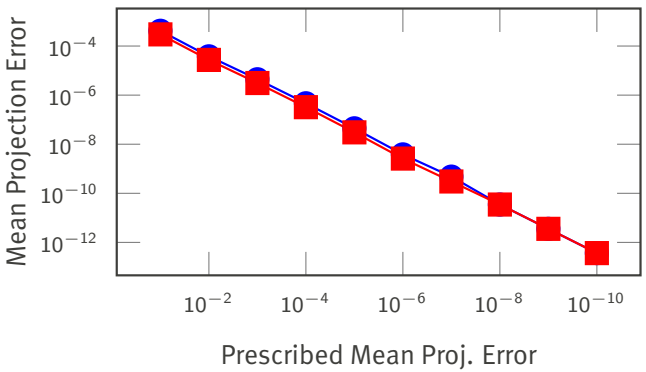
Live HAPOD

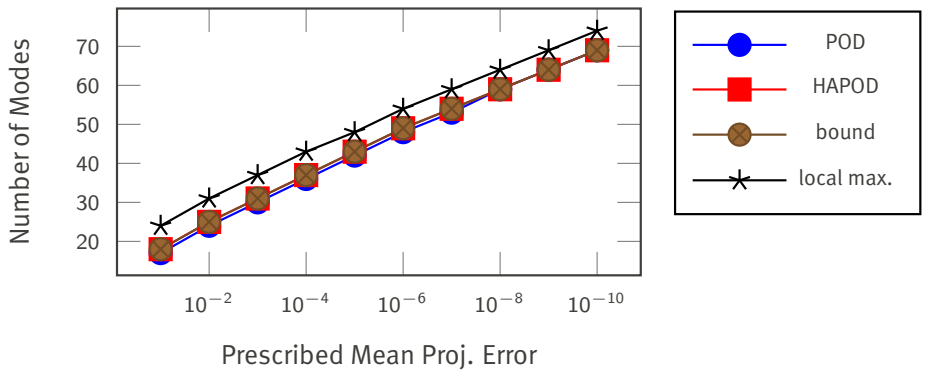


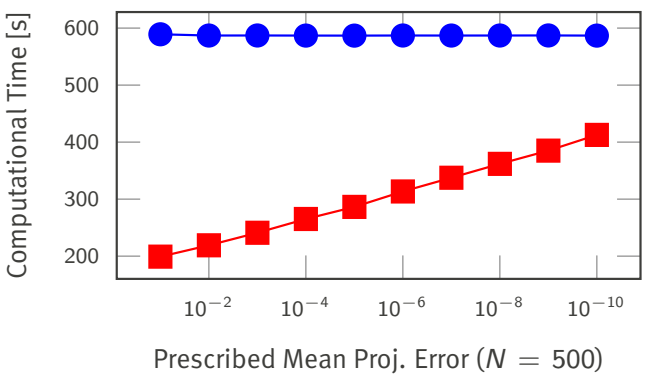
- On-the-fly compression of large trajectories.

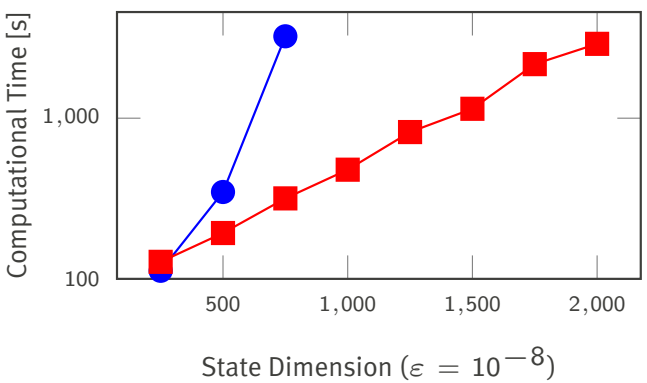
HAPOD – Live HAPOD Example

- State space trajectory of “synthetic” MORWiki benchmark model excited with random input.
- All computations on Raspberry Pi 1B single board computer.



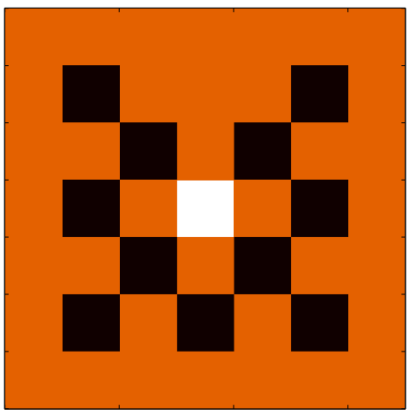


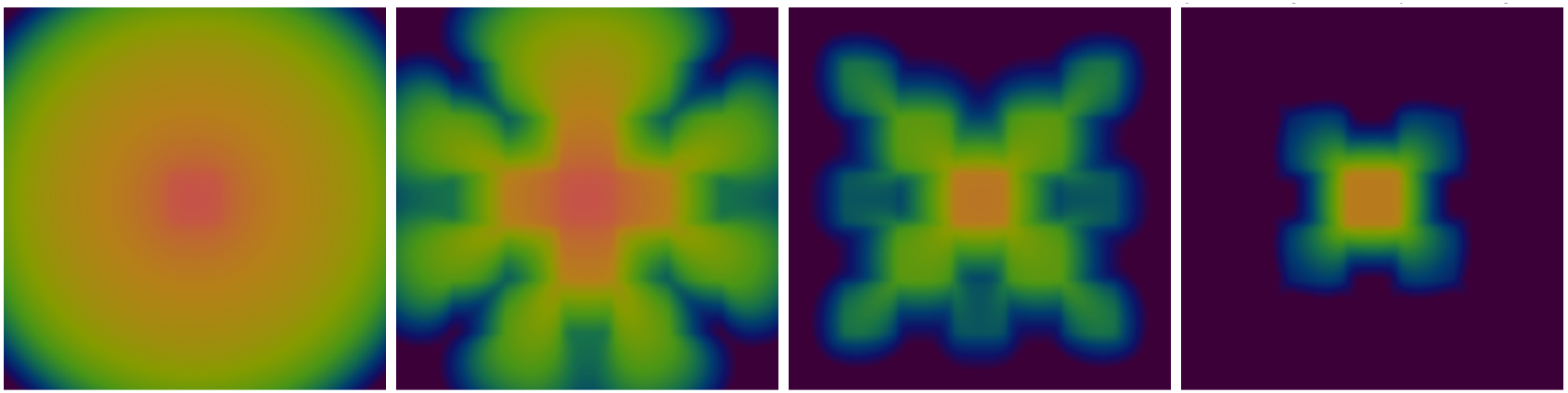




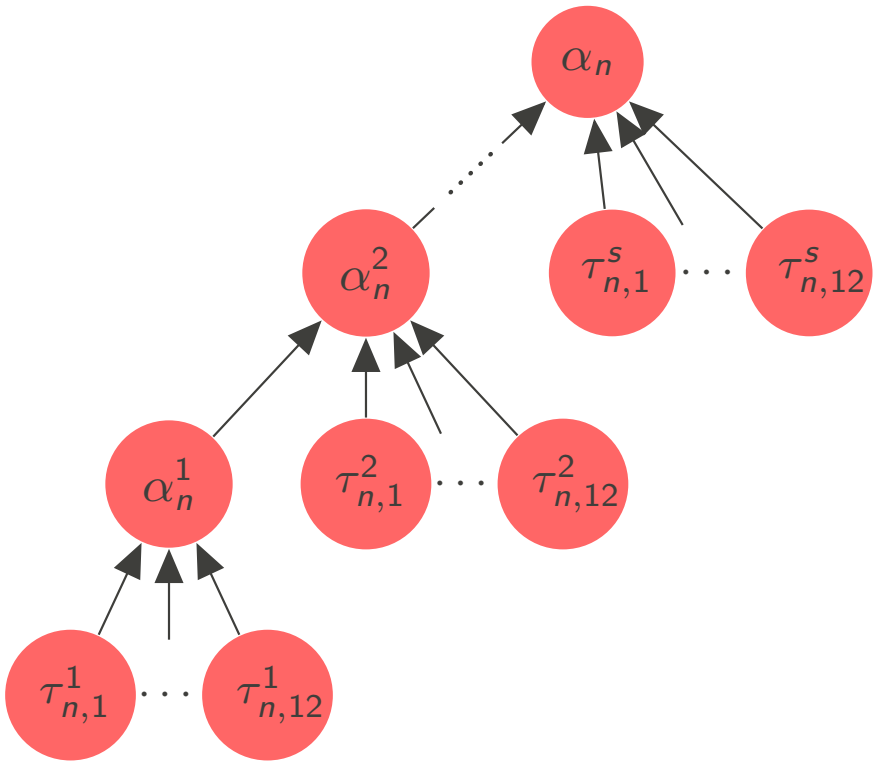
HAPOD – HPC Example

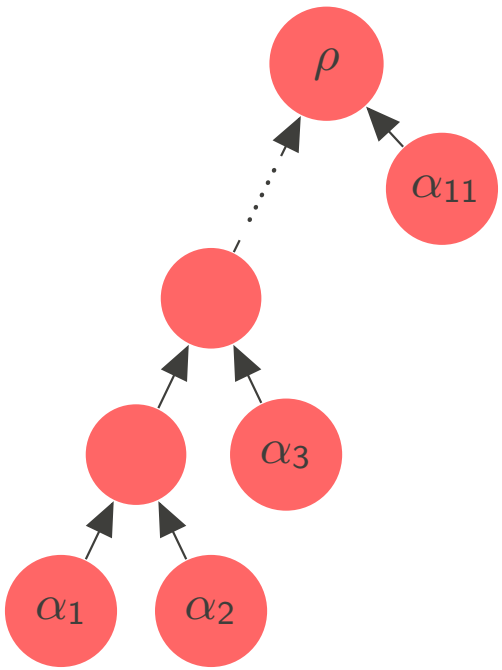
- 2D neutron transport equation.
- Moment closure/FV approximation.
- Varying absorbtion and scattering coefficients.
- Distributed snapshot and HAPOD computation on PALMA cluster (125 cores).





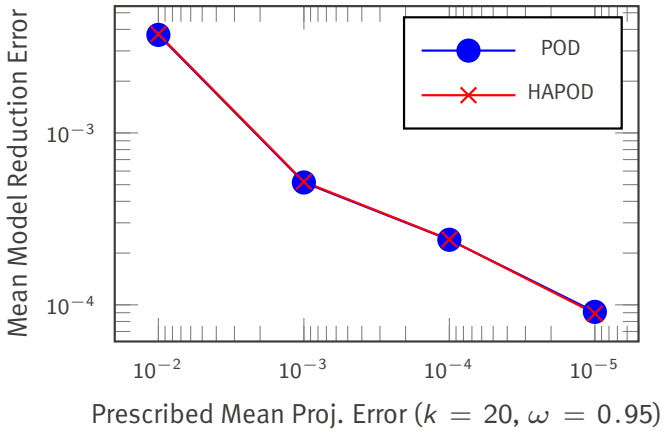
HAPOD – HPC Example

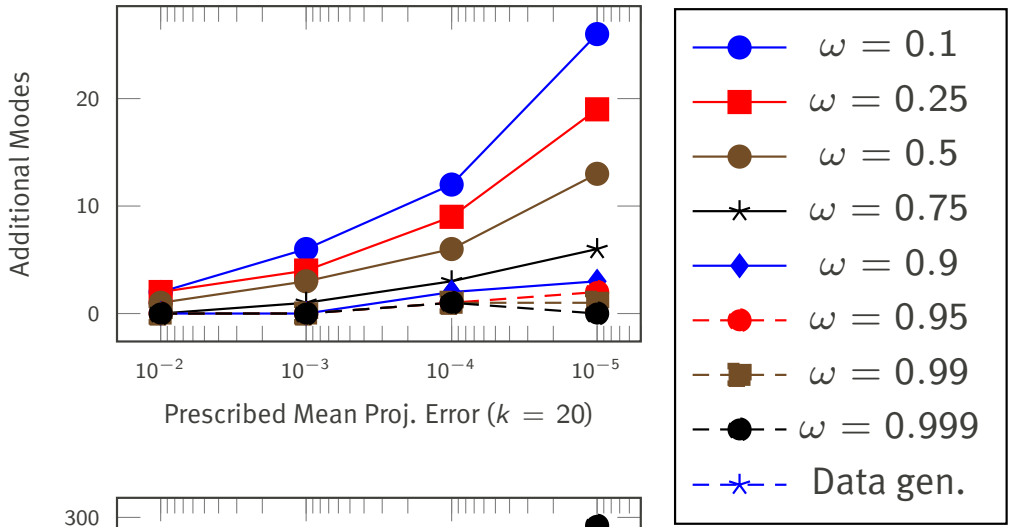


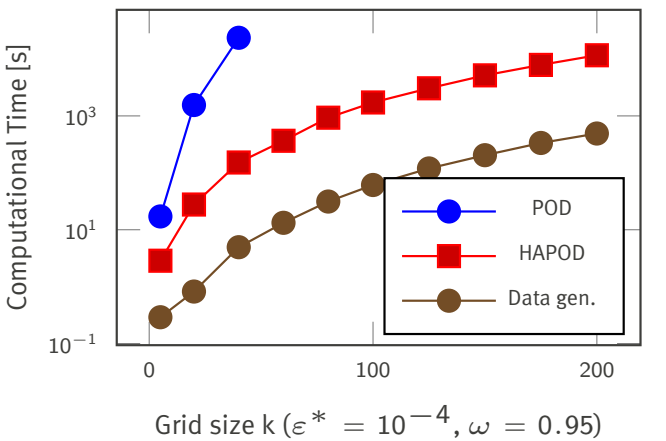


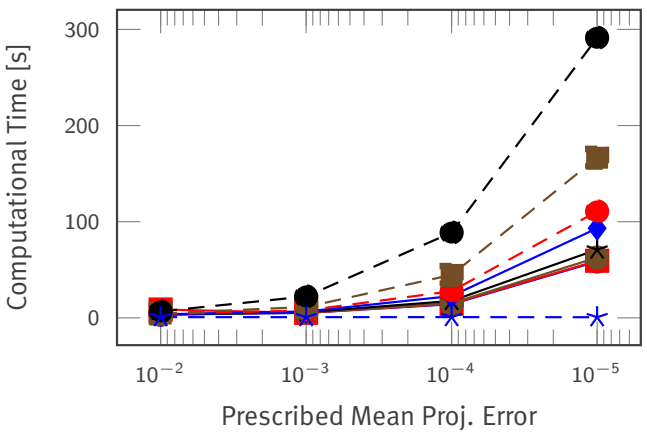
- HAPOD on compute node n. Time steps are split into s slices. Each processor cores computes one slice at a time, performs POD and sends resulting modes to main MPI rank on the node.
- Live HAPOD is performed on MPI rank o with modes collected on each node.

HAPOD – HPC Example

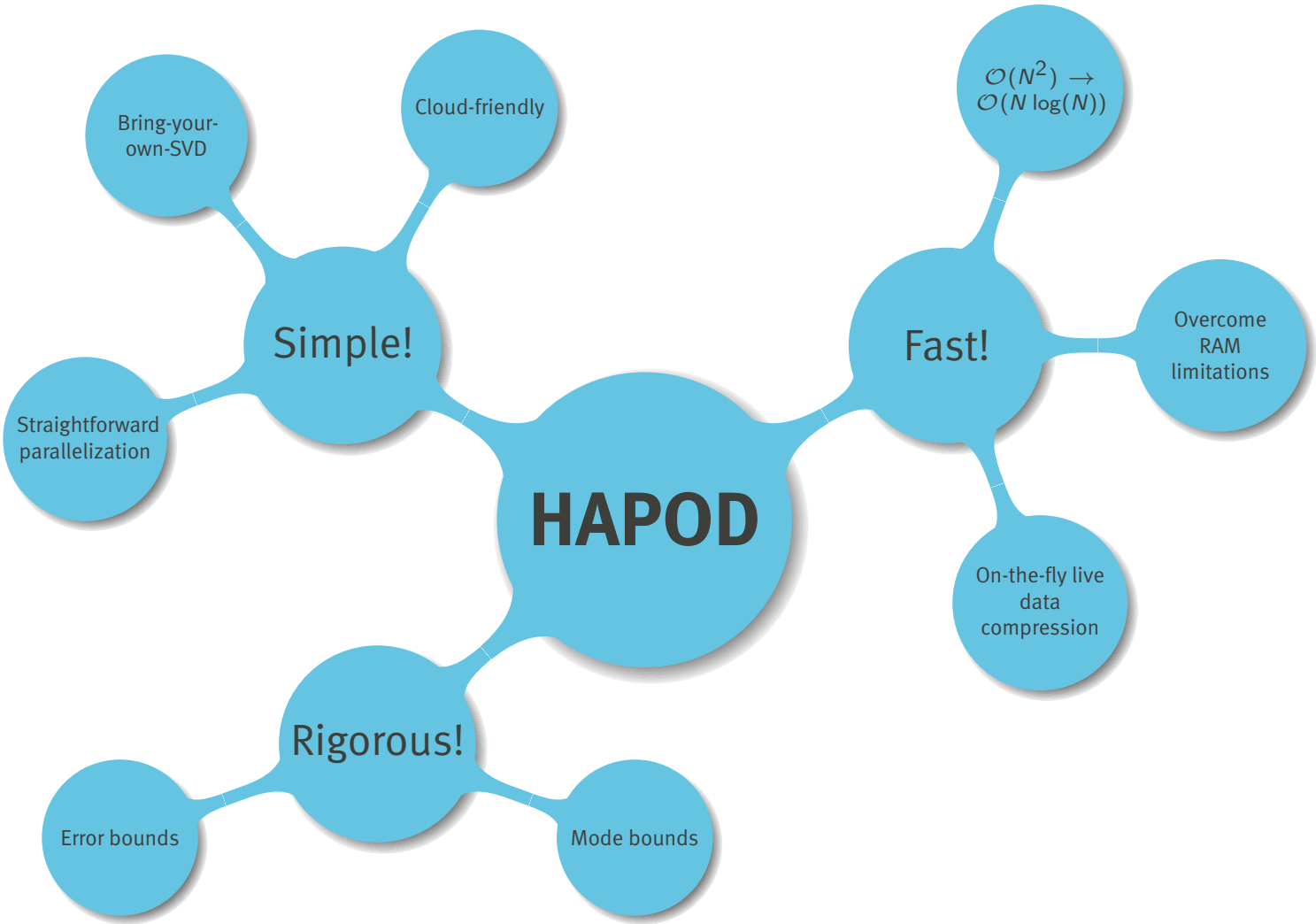








- $\approx 39.000 \cdot k^3$ doubles per trajectory ($\approx$ 2.5 terabyte for $k = 200$).


HAPOD
Simple!
Bring-your-own-SVD
Cloud-friendly
Straightforward parallelization
Fast!
$\mathcal{O}(N^2) \rightarrow \mathcal{O}(N \log(N))$
Overcome RAM limitations
On-the-fly live data compression
Rigorous!
Error bounds
Mode bounds

Thank you for your attention!

My homepage
`http://stephanrave.de/`

MULTIBAT: Unified Workflow for fast electrochemical 3D simulations of lithium-ion cells combining virtual stochastic microstructures, electrochemical degradation models and model order reduction
(under preparation)
`http://j.mp/multibat`

pyMOR – Generic Algorithms and Interfaces for Model Order Reduction
SIAM J. Sci. Comput., 38(5), pp. S194–S216
`http://www.pymor.org/`

Hierarchical Approximate Proper Orthogonal Decomposition
arXiv:1607.05210