Westfälische Wilhelms-Universität Münster

Reduced Basis Methods

From Theory to Implementation

Stephan Rave

January 14, 2015

Outline

- The Reduced Basis Method
- Reduction of Li-Ion Battery Models
- Advection Dominated Problems and the Method of Freezing
- Model Order Reduction with pyMOR

The Reduced Basis Method

Parametrized Model Order Reduction

Want to evaluate some solution map

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

from some compact set $\mathcal{P}$ into normed space V (and quantities of interest derived from $\Phi(\mu)$).

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

Parametrized Model Order Reduction

Want to evaluate some solution map

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

from some compact set $\mathcal{P}$ into normed space V (and quantities of interest derived from $\Phi(\mu)$).

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.)

Parametrized Model Order Reduction

Want to evaluate some solution map

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

from some compact set $\mathcal{P}$ into normed space V (and quantities of interest derived from $\Phi(\mu)$).

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!

Parametrized Model Order Reduction

Want to evaluate some solution map

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

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

Parametrized Model Order Reduction

Want to evaluate some solution map

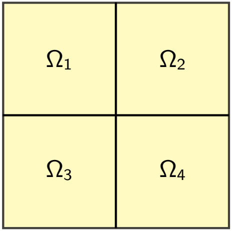
$$\Phi : \mathcal{P} \longrightarrow V$$

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

Build reduced model by finding:

1. low dimensional subspace $V_N \subset V$ for approximating $\Phi(\mathcal{P})$.
   ($100 \approx N = \dim V_N \ll \dim V$)
2. quickly computable approximation $\Phi_N : \mathcal{P} \longrightarrow V_N$ s.t. $||\Phi(\mu) - \Phi_N(\mu)|| < \varepsilon$
3. quickly computable upper bound $\Delta_N(\Phi_N(\mu)) \geq ||\Phi(\mu) - \Phi_N(\mu)||$.

Model Problem



$$\Omega = \bigcup_{i=1}^{4} \Omega_i, \quad \mathcal{P} = [\alpha, 1]^4, \ \alpha > 0$$

$$a_\mu(x) = \sum_{i=1}^{4} \mu_i \cdot \chi_{\Omega_i}(x), \qquad x \in \Omega, \mu \in \mathcal{P}$$

$$f \in L^2(\Omega)$$

Thermal-Block Problem

For $\mu \in \mathcal{P}$, find $\Phi(\mu) := u_\mu \in H_0^1(\Omega) =: V$ s.t.

$$-\nabla \cdot (a_\mu \nabla u_\mu) = f$$

Existence of good V_N

- V_N only needs to approximate solution manifold $\Phi(\mathcal{P}) \subset V$ for a specific problem.
  (Compare finite element spaces.)

Existence of good V_N

- V_N only needs to approximate solution manifold $\Phi(\mathcal{P}) \subset V$ for a specific problem.
  (Compare finite element spaces.)

- Assume $\mathcal{P} \subset \mathbb{R}^P$ is compact, Φ smooth. Then $\Phi(\mathcal{P})$ is compact and manifold(-ish).

Existence of good V_N

- V_N only needs to approximate solution manifold $\Phi(\mathcal{P}) \subset V$ for a specific problem.
  (Compare finite element spaces.)

- Assume $\mathcal{P} \subset \mathbb{R}^P$ is compact, Φ smooth. Then $\Phi(\mathcal{P})$ is compact and manifold(-ish).

- Since $\Phi(\mathcal{P})$ smooth and $\dim \Phi(\mathcal{P}) \ll \dim V$, can hope to approximate with low-dim. linear spaces.

Approximation Theory

Definition

Let $\mathcal{M} \subset V$ be subset of normed space. The *Kolmogorov n-width* $d_n(\mathcal{M})$ is given as

$$d_n(\mathcal{M}) = \inf_{\substack{V_n \subseteq V \\ \text{lin subsp.} \\ \dim V_n \leq n}} \sup_{m \in \mathcal{M}} \inf_{v \in V_n} ||m - v||.$$

Approximation Theory

Definition

Let $\mathcal{M} \subset V$ be subset of normed space. The *Kolmogorov n-width* $d_n(\mathcal{M})$ is given as

$$d_n(\mathcal{M}) = \inf_{\substack{V_n \subseteq V \\ \text{lin subsp.} \\ \dim V_n \leq n}} \sup_{m \in \mathcal{M}} \inf_{v \in V_n} ||m - v||.$$

- Cannot beat n-width.
- For elliptic problems with fixed operator and arbitrary RHS in some unit ball: Polynomial decay of $d_n(\mathcal{M})$.
- Hope for exponential decay of $d_n(\Phi(\mathcal{P}))$.

Approximation Theory

Proposition (Cohen, DeVore, 2014)

Let $F : V \times X \longrightarrow W$ holomorphic map between Banach spaces and $\mathcal{P} \subseteq X$.
If for all $\mu \in \mathcal{P}$

- $\Phi(\mu) := u_\mu$ is the unique solution of $F(u_\mu, \mu) = 0$
- $\partial_u F(u_\mu, \mu) : V \longrightarrow W$ is invertible,

then there is holomorphic extension $\Phi : O \longrightarrow W$ with $\mathcal{P} \subseteq O \subseteq X$ open.

Approximation Theory

Proposition (Cohen, DeVore, 2014)

Let $F : V \times X \longrightarrow W$ holomorphic map between Banach spaces and $\mathcal{P} \subseteq X$. If for all $\mu \in \mathcal{P}$

- $\Phi(\mu) := u_\mu$ is the unique solution of $F(u_\mu, \mu) = 0$
- $\partial_u F(u_\mu, \mu) : V \longrightarrow W$ is invertible,

then there is holomorphic extension $\Phi : O \longrightarrow W$ with $\mathcal{P} \subseteq O \subseteq X$ open.

Proof

Implicit function theorem (for complex Banach spaces).

Approximation Theory

Corollary

There are $M, a, \alpha > 0$ s.t.

$$d_n(\Phi(\mathcal{P})) < Me^{-an^\alpha}$$

Approximation Theory

Corollary

There are $M, a, \alpha > 0$ s.t.

$$d_n(\Phi(\mathcal{P})) < Me^{-an^\alpha}$$

Proof

Truncated power series expansion of Φ.

Approximation Theory

Corollary

There are $M, a, \alpha > 0$ s.t.

$$d_n(\Phi(\mathcal{P})) < Me^{-an^\alpha}$$

Proof

Truncated power series expansion of Φ.

- Note that $\alpha \sim 1/\dim \mathcal{P}$!
- There are results for $\mathcal{P} \subseteq B_1(L^p(\Omega))$ yielding polynomial decay of d_n.

Approximation Theory (Model Problem)

Thermal-Block Problem

For $\mu \in \mathcal{P}$, find $\Phi(\mu) := u_\mu \in H_0^1(\Omega) =: V$ s.t.

$$-\nabla \cdot (\sum_{i=1}^{4} \mu_i \chi_{\Omega_i} \nabla u_\mu) = f$$

Approximation Theory (Model Problem)

Thermal-Block Problem

For $\mu \in \mathcal{P}$, find $\Phi(\mu) := u_\mu \in H_0^1(\Omega) =: V$ s.t.

$$-\nabla \cdot \left(\sum_{i=1}^{4} \mu_i \chi_{\Omega_i} \nabla u_\mu\right) = f$$

Let

$$A_i := -\nabla \cdot (\chi_{\Omega_i} \nabla) : H_0^1(\Omega) \longrightarrow H^{-1}(\Omega)$$

then

$$F(u, \mu) := f - \sum_{i=1}^{4} \mu_i A_i(u)$$

fulfills assumptions of theorem.

How to find good V_N?

Definition (Weak Greedy Algorithm)

For given $\mathcal{M} \subseteq V$ let $0 < \gamma \leq 1$ and $s_1, s_2, \ldots \in \mathcal{M}$ be such that

$$\inf_{v \in V_{n-1}} ||s_n - v|| \geq \gamma \cdot \sup_{m \in \mathcal{M}} \inf_{v \in V_{n-1}} ||m - v|| \qquad V_n := \operatorname{span}\{s_1, \ldots s_n\}$$

then $(s_n)_n$ is called weak greedy sequence for $\mathcal{M}$ with parameter γ.

How to find good V_N?

Definition (Weak Greedy Algorithm)

For given $\mathcal{M} \subseteq V$ let $0 < \gamma \leq 1$ and $s_1, s_2, \ldots \in \mathcal{M}$ be such that

$$\inf_{v \in V_{n-1}} ||s_n - v|| \geq \gamma \cdot \sup_{m \in \mathcal{M}} \inf_{v \in V_{n-1}} ||m - v|| \qquad V_n := \mathrm{span}\{s_1, \ldots s_n\}$$

then $(s_n)_n$ is called weak greedy sequence for $\mathcal{M}$ with parameter γ.

- Greedy algorithm for $\gamma = 1$.
- How to find such s_n? See below ...

Stephan Rave (stephan.rave@wwu.de)

How to find good V_N?

Theorem (Binev, Cohen, Dahmen, DeVore, Petrova, Wojtaszczyk, 2011)

Let V be Hilbert space an $\mathcal{M} \subseteq V$ be given such that for $M, a, \alpha > 0$

$$d_n(\mathcal{M}) \leq Me^{-an^\alpha}.$$

If $V_n := \operatorname{span}\{s_1, \dots s_n\}$ for a weak greedy sequence $(s_n)_n$ for $\mathcal{M}$ with parameter γ, then there are $C, c > 0$ only depending on a, α, γ, s.t. with $\beta := \alpha/(\alpha + 1)$

$$\sup_{m\in\mathcal{M}} \inf_{v\in V_n} ||m - v|| \leq CMe^{-cn^\beta}.$$

How to find good V_N?

Theorem (Binev, Cohen, Dahmen, DeVore, Petrova, Wojtaszczyk, 2011)

Let V be Hilbert space an $\mathcal{M} \subseteq V$ be given such that for $M, a, \alpha > 0$

$$d_n(\mathcal{M}) \leq Me^{-an^\alpha}.$$

If $V_n := \operatorname{span}\{s_1, \dots s_n\}$ for a weak greedy sequence $(s_n)_n$ for $\mathcal{M}$ with parameter γ, then there are $C, c > 0$ only depending on a, α, γ, s.t. with $\beta := \alpha/(\alpha + 1)$

$$\sup_{m \in \mathcal{M}} \inf_{v \in V_n} ||m - v|| \leq CMe^{-cn^\beta}.$$

Corollary

For affinely decomposed problems, weak greedy algorithm is a *constructive* method for finding approximation spaces V_N with exponentially fast decreasing best-approximation error.

Parametrized Model Order Reduction

Want to compute the solutions $\Phi(\mu) := u_\mu$ of the equation

$$F(u_\mu, \mu) = 0$$

with V Hilbert space, X, W Banach spaces, $\mathcal{P} \subseteq X$ and $F : V \times X \longrightarrow W$ holomorphic.

Build reduced model by finding:

1. low dimensional subspace $V_N \subset V$ for approximating $\Phi(\mu)$.
   use weak greedy algorithm
2. quickly computable approximation $\Phi_N : \mathcal{P} \longrightarrow V_N$ s.t. $||\Phi(\mu) - \Phi_N(\mu)|| < \varepsilon$
3. quickly computable upper bound $\Delta_N(\Phi_N(\mu)) \geq ||\Phi(\mu) - \Phi_N(\mu)||$.

Definition of Φ_N

Assume that $\Phi(\mu) := u_\mu \in V$ is given as solution of a weak problem

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

with coercive, continuous bilinear forms B_μ and $f \in V'$.

Definition of Φ_N

Assume that $\Phi(\mu) := u_\mu \in V$ is given as solution of a weak problem

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

with coercive, continuous bilinear forms B_μ and $f \in V'$.

Reduced problem

Define the reduced approximation $\Phi_N(\mu) := u_{\mu,N} \in V_N$ to be the Galerkin projection of u_μ onto V_N, i.e. the solution of

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

Since B_μ is coercive, $u_{\mu,N}$ is well-defined!

Stephan Rave (stephan.rave@wwu.de)

Definition of Φ_N

Lemma (Céa)

Let C_μ denote the coercivity constant of B_μ. Then

$$||u_\mu - u_{\mu,N}|| \leq \frac{||B_\mu||}{C_\mu} \inf_{v \in V_N} ||u_\mu - v||$$

Definition of Φ_N

Lemma (Céa)

Let C_μ denote the coercivity constant of B_μ. Then

$$||u_\mu - u_{\mu,N}|| \leq \frac{||B_\mu||}{C_\mu} \inf_{v \in V_N} ||u_\mu - v||$$

If $B_\mu = \sum_{i=1}^{I} \mu_i B_i$, let $F_i : V \longrightarrow V'$, $F_i(u) := B_i(u, \cdot)$ and $F(u, \mu) := f - \sum_{i=1}^{I} \mu_i F_i(u)$, then u_μ is the solution of $F(u, \mu) = 0$.

Corollary

For affinely decomposed coercive problems and reduced spaces V_N resulting from weak greedy sequence, there are constants A, a, α, s.t.

$$||u_\mu - u_{\mu,N}|| \leq Ae^{-aN^\alpha} \qquad \forall \mu \in \mathcal{P}.$$

Definition of Φ_N (Model Problem)

Thermal-Block Problem

For $\mu \in \mathcal{P}$, find $\Phi(\mu) := u_\mu \in H_0^1(\Omega) =: V$ s.t.

$$-\nabla \cdot \left(\sum_{i=1}^{4} \mu_i \chi_{\Omega_i} \nabla u_\mu\right) = f$$

Let

$$B_i(u, v) := \int_{\Omega_i} \nabla u(x) \nabla v(x) dx, \qquad B_\mu := \sum_{i=1}^{4} \mu_i B_i$$

then u_μ satisfies

$$B_\mu(u_\mu, v) = \int_\Omega f(x) v(x) dx \qquad \forall v \in H_0^1(\Omega).$$

B_μ is coercive for each $\mu \in \mathcal{P} = [\alpha, 1]^4$.

Parametrized Model Order Reduction

Want to compute the solutions $\Phi(\mu) := u_\mu$ of the equation

$$B(u_\mu, v) = \sum_{q=1}^{Q} \mu_q B_q(u_\mu, v) = f(v) \qquad \forall v \in V$$

with V Hilbert space, B_q continuous bilinear forms, $f \in V'$, B_μ coercive for $\mathcal{P} \subseteq \mathbb{R}^P$.

Build reduced model by finding:

1. low dimensional subspace $V_N \subset V$ for approximating $\Phi(\mu)$.
   use weak greedy algorithm
2. quickly computable approximation $\Phi_N : \mathcal{P} \longrightarrow V_N$.
   Galerkin projection onto V_N
3. quickly computable upper bound $\Delta_N(\mu) \geq ||\Phi(\mu) - \Phi_N(\mu)||$.

A-Posteriori Error Estimator

Define residual $\mathcal{R}_\mu(u) \in V'$ as

$$\mathcal{R}_\mu(u)(v) := f(v) - B_\mu(u, v).$$

Then

$$\begin{aligned}||u_\mu - u_{\mu,N}||^2 &\leq C_\mu^{-1} B_\mu(u_\mu - u_{\mu,N}, u_\mu - u_{\mu,N}) \\ &= C_\mu^{-1} \mathcal{R}_\mu(u_{\mu,N})(u_\mu - u_{\mu,N}) \leq C_\mu^{-1} ||\mathcal{R}_\mu(u_{\mu,N})|| \, ||u_\mu - u_{\mu,N}||\end{aligned}$$

Proposition

$$||u_\mu - u_{\mu,N}|| \leq \Delta_\mu(u_{\mu,N}) := C_\mu^{-1} ||\mathcal{R}(u_{\mu,N})|| \leq ||B_\mu|| C_\mu^{-1} ||u_\mu - u_{\mu,N}||$$

How to find good V_N (cont.)

Greedy algorithm with error estimator

Choose snapshots $s_n := u_{\mu_n}$ where μ_n is given by

$$\mu_n := \underset{\mu \in \mathcal{P}}{\arg\max}\, \Delta_{n-1}(u_{\mu,n-1})$$

How to find good V_N (cont.)

Greedy algorithm with error estimator

Choose snapshots $s_n := u_{\mu_n}$ where μ_n is given by

$$\mu_n := \operatorname*{arg\,max}_{\mu \in \mathcal{P}} \Delta_{n-1}(u_{\mu,n-1})$$

Then

$$\begin{aligned}\inf_{v \in V_{n-1}} ||s_n - v|| &\gtrsim ||u_{\mu_n} - u_{\mu_n,n-1}|| \\ &\gtrsim \Delta_{n-1}(u_{\mu_n,n-1}) \\ &\geq \Delta_{n-1}(u_{\mu,n-1}) \gtrsim ||u_\mu - u_{\mu,n-1}|| \geq \inf_{v \in V_{n-1}} ||u_\mu - v||\end{aligned}$$

Proposition

The greedy algorithm with error estimator generates a weak greedy sequence.

Parametrized Model Order Reduction

Want to compute the solutions $\Phi(\mu) := u_\mu$ of the equation

$$B(u_\mu, v) = \sum_{q=1}^{Q} \mu_q B_q(u_\mu, v) = f(v) \qquad \forall v \in V$$

with V Hilbert space, B_q continuous bilinear forms, $f \in V'$, B_μ coercive for $\mathcal{P} \subseteq \mathbb{R}^P$.

Build reduced model by finding:

1. low dimensional subspace $V_N \subset V$ for approximating $\Phi(\mu)$.
   use greedy algorithm with error estimator
2. quickly computable approximation $\Phi_N : \mathcal{P} \longrightarrow V_N$.
   Galerkin projection onto V_N
3. quickly computable upper bound $\Delta_N(\mu)$.
   residual-based error estimator

living.knowledge WWU Münster

Offline-Online Decomposition

Affinely Decomposed Problem

For $\mu \in \mathcal{P}$, find $\Phi(\mu) := u_\mu \in V$ s.t. $\sum_{q=1}^{Q} \mu_q B_q(u_\mu, v) = f(v) \quad \forall v \in V$.

Let $\varphi_1, \cdots, \varphi_N$ be a basis of V_N (the reduced basis!), then $u_{\mu,N}$ is given as

$$u_{\mu,N} = \sum_{l=1}^{N} \varphi_l \cdot \underline{u}_{\mu,N,l}$$

where

$$\sum_{q=1}^{Q} \mu_q \cdot \left[B_q(\varphi_l, \varphi_k)\right]_{k,l} \cdot \underline{u}_{\mu,N,l} = \left[f(\varphi_k)\right]_k \qquad (1)$$

Offline-Online Decomposition

Affinely Decomposed Problem

For $\mu \in \mathcal{P}$, find $\Phi(\mu) := u_\mu \in V$ s.t. $\sum_{q=1}^{Q} \mu_q B_q(u_\mu, v) = f(v) \quad \forall v \in V$.

Let $\varphi_1, \cdots, \varphi_N$ be a basis of V_N (the reduced basis!), then $u_{\mu,N}$ is given as

$$u_{\mu,N} = \sum_{l=1}^{N} \varphi_l \cdot \underline{u}_{\mu,N,l}$$

where

$$\sum_{q=1}^{Q} \mu_q \cdot \left[B_q(\varphi_l, \varphi_k)\right]_{k,l} \cdot \underline{u}_{\mu,N,l} = \left[f(\varphi_k)\right]_k \tag{1}$$

Warning

Snapshot basis $m_1, \ldots, m_N$ of V_N leads to (really!) badly conditioned reduced system matrices! Orthogonalize!

Offline-Online Decomposition

Affinely Decomposed Problem

For $\mu \in \mathcal{P}$, find $\Phi(\mu) := u_\mu \in V$ s.t. $\sum_{q=1}^{Q} \mu_q B_q(u_\mu, v) = f(v) \quad \forall v \in V$.

Let $\varphi_1, \cdots, \varphi_N$ be a basis of V_N (the reduced basis!), then $u_{\mu,N}$ is given as

$$u_{\mu,N} = \sum_{l=1}^{N} \varphi_l \cdot \underline{u}_{\mu,N,l}$$

where

$$\sum_{q=1}^{Q} \mu_q \cdot \left[B_q(\varphi_l, \varphi_k)\right]_{k,l} \cdot \underline{u}_{\mu,N,l} = \left[f(\varphi_k)\right]_k \tag{1}$$

Proposition

If $\left[B_q(\varphi_l, \varphi_k)\right]_{k,l}$ are pre-computed, (1) can be solved with effort $\mathcal{O}(QN^2 + N^3)$.

Offline-Online Decomposition (Error estimator)

Let $R : V' \longrightarrow V$ be the Riesz isomorphism. Then

$$\begin{aligned}||\mathcal{R}_\mu(u_{\mu,N})|| &= ||R(\mathcal{R}_\mu(u_\mu, N))|| \\ &= ||R(f) + \sum_{q=1}^{Q}\sum_{n=1}^{N} \underline{u}_{\mu,N,n} R(B_q(\varphi_n, \cdot))||\end{aligned}$$

Offline-Online Decomposition (Error estimator)

Let $R : V' \longrightarrow V$ be the Riesz isomorphism. Then

$$\begin{aligned} ||\mathcal{R}_\mu(u_{\mu,N})|| &= ||R(\mathcal{R}_\mu(u_\mu, N))|| \\ &= ||R(f) + \sum_{q=1}^{Q}\sum_{n=1}^{N} \underline{u}_{\mu,N,n} R(B_q(\varphi_n, \cdot))|| \end{aligned}$$

Pre-compute all $(1+QN)^2$ cross-terms in scalar-product evaluation. Online effort: $\mathcal{O}((1+QN)^2) = \mathcal{O}(Q^2N^2)$. However, only bad numerical stability (half machine precision).

Better approach:

Stable Estimator Decomposition (Buhr, R, 2014)

Project $R(\mathcal{R}_\mu)$ onto V_N and $\mathrm{span}\{R(f), R(B_q(\varphi_n, \cdot))\}$ using orthonormal bases.

The Reduced Basis Method

Want to compute the solutions $\Phi(\mu) := u_\mu$ of the equation

$$B(u_\mu, v) = \sum_{q=1}^{Q} \mu_q B_q(u_\mu, v) = f(v) \qquad \forall v \in V$$

with V Hilbert space, B_q continuous bilinear forms, $f \in V'$, B_μ coercive for $\mathcal{P} \subseteq \mathbb{R}^P$.

The Reduced Basis Method

Want to compute the solutions $\Phi(\mu) := u_\mu$ of the equation

$$B(u_\mu, v) = \sum_{q=1}^{Q} \mu_q B_q(u_\mu, v) = f(v) \qquad \forall v \in V$$

with V Hilbert space, B_q continuous bilinear forms, $f \in V'$, B_μ coercive for $\mathcal{P} \subseteq \mathbb{R}^P$.

- Replace u_μ by good-enough (but expensive to compute) discrete approximation $u_{\mu,h} \in V_h$ satisfying

$$B_h(u_{\mu,h}, v_h) = \sum_{q=1}^{Q} \mu_q B_{q,h}(u_{\mu,h}, v_h) = f_h(v_h) \qquad \forall v_h \in V_h$$

  with $B_{q,h}$ discrete bilinear forms, $f_h \in V_h'$, $B_{\mu,h}$ coercive for $\mu \in \mathcal{P}$ (FEM, DG, etc.)

The Reduced Basis Method

Offline Phase

Compute reduced basis using greedy search with error estimator on finite training set $\mathcal{S} \subseteq \mathcal{P}$, i.e. $s_n := u_{\mu_n,h}$, $V_n := \operatorname{span}\{s_1, \ldots s_n\}$, $n = 1, \ldots, N$ where

$$\mu_n := \operatorname*{arg\,max}_{\mu \in \mathcal{S}} \Delta_{n-1}(u_{\mu,n-1})$$

The Reduced Basis Method

Offline Phase

Compute reduced basis using greedy search with error estimator on finite training set $\mathcal{S} \subseteq \mathcal{P}$, i.e. $s_n := u_{\mu_n,h}$, $V_n := \mathrm{span}\{s_1, \ldots s_n\}$, $n = 1, \ldots, N$ where

$$\mu_n := \arg\max_{\mu \in \mathcal{S}} \Delta_{n-1}(u_{\mu,n-1})$$

Online Phase

Compute $u_{\mu,N}$, $\Delta_N(u_{\mu,N})$ for arbitrary new $\mu \in \mathcal{P}$.

Stephan Rave (stephan.rave@wwu.de)

Reduction of Li-Ion Battery Models

The MULTIBAT Project

ulm university universität uulm

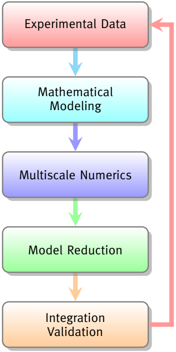
Institute of Technical Thermodynamics DLR

MULTIBAT

SPONSORED BY THE Federal Ministry of Education and Research

Fraunhofer ITWM

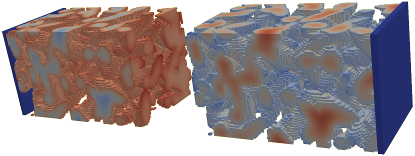
Westfälische Wilhelms-Universität Münster



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

Problem

- Li-plating is initiated at micrometre scale at interface between active electrode particles and electrolyte.
- Need microscale models which resolve active particle geometry.
- Result: huge non-linear discrete models.
  - Cannot be solved at cell scale on current hardware.
  - **Parameter studies extremely expensive, even on small domains.**



Microscale Battery Models

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

$$\begin{aligned} \frac{\partial c}{\partial t} - \nabla \cdot (\alpha(c,\phi)\nabla c + \beta(c,\phi)\nabla\phi) &= 0 \qquad c : \text{Li}^+ \text{ concentration} \\ -\nabla \cdot (\gamma(c,\phi)\nabla c + \delta(c,\phi)\nabla\phi) &= 0 \qquad \phi : \text{potential} \end{aligned}$$

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

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

$$\begin{aligned} j_{se} &= 2k\sqrt{c_e c_s(c_{max} - c_s)}\,\sinh\left(\frac{\phi_s - \phi_e - U_0(\frac{c_s}{c_{max}})}{2RT}\cdot F\right) \\ N_{se} &= \frac{1}{F}\cdot j_{se} \end{aligned}$$

Microscale Model

- Finite volume discretization with implicit Euler leads to

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

- Model has been implemented at Fraunhofer ITWM in BEST.
- $\mu \in \mathcal{P}$ indicates dependence on model parameters we want to vary (e.g. temperature T, charge rate).

Reduced Basis Approximation

- **Online phase:** Determine reduced solution by solving projected equation

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

Reduced Basis Approximation

- **Online phase:** Determine reduced solution by solving projected equation

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

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

function $\textsc{Greedy}(\mathcal{S}_{train} \subset \mathcal{P}, \varepsilon, \tilde{V}_c^0, \tilde{V}_\phi^0)$
    $\tilde{V}_c, \tilde{V}_\phi \leftarrow \tilde{V}_c^0, \tilde{V}_\phi^0$
    while $\max_{\mu \in \mathcal{S}_{train}} \textsc{Err-Est}(\textsc{RB-Solve}(\mu), \mu) > \varepsilon$ **do**
        $\mu^* \leftarrow \operatorname{arg-max}_{\mu \in \mathcal{S}_{train}} \textsc{Err-Est}(\textsc{RB-Solve}(\mu), \mu)$
        $\tilde{V}_c, \tilde{V}_\phi \leftarrow \textsc{Basis-Ext}(\tilde{V}_c, \tilde{V}_\phi, \textsc{Solve}(\mu^*))$
    end while
    return $\tilde{V}_c, \tilde{V}_\phi$
end function

Empirical Interpolation

- Evaluation of

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

still costly.

Empirical Interpolation

- Evaluation of

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

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

Empirical Interpolation

- Evaluation of

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

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

$$P_{\tilde{V}} \circ A_\mu \approx P_{\tilde{V}} \circ (I_M \circ \tilde{A}_\mu \circ R_{M'})$$

  where

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

Empirical Interpolation

- Evaluation of

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

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

$$P_{\tilde{V}} \circ A_\mu \approx P_{\tilde{V}} \circ (I_M \circ \tilde{A}_\mu \circ R_{M'})$$

  where

  $\tilde{A}_\mu$: A_μ restricted to M interpolation DOFs
  I_M: Interpolation operator
  $R_{M'}$: Restriction to M' DOFs needed for evaluation
- Use greedy algorithms to determine DOFs and interpolation basis.

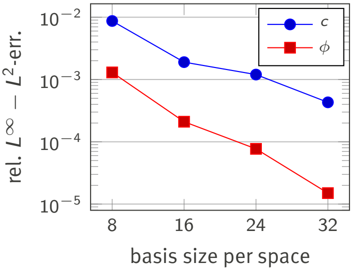
Implementation

- Experimental implementation of battery model with Dune software framework.
- Model reduction with pyMOR.
- Integration of pyMOR with BEST.

Implementation

- Experimental implementation of battery model with Dune software framework.
- Model reduction with pyMOR.
- Integration of pyMOR with BEST.

- Small 3D test case ($3.2 \cdot 10^4$ DOFs)
- $T \in [250,\ 350]\ K$
  $I_{charge} \in [10^{-4},\ 10^{-3}]\ A/cm^2$
- without operator interpolation
  ERR-EST = true error



Stephan Rave (stephan.rave@wwu.de)

Advection Dominated Problems and the Method of Freezing

The Problem

Model reduction for parameter dependent, convection dominated, nonlinear Cauchy problem

$$\partial_t u_\mu(t) + \mathcal{L}_\mu(u_\mu(t)) = 0, \quad u_\mu(0) = u_0$$

where

- $\mu \in \mathcal{P}$ (parameter space)
- $u_\mu(t) \in V$, $t \in [0, T]$ for appropriate function space V
- $\mathcal{L}_\mu$ partial differential operator

using reduced basis approach, i.e. ...

The Problem

$$\partial_t u_\mu(t) + \mathcal{L}_\mu(u_\mu(t)) = 0, \quad u_\mu(0) = u_0$$

Assume we have

- H-dimensional linear discrete space V_h ($0 \ll H$)
- Operator $\mathcal{L}_{\mu,h}$ on V_h approximating $\mathcal{L}_\mu$
- N-dimensional linear RB-space $V_N \subset V_h$ ($N \ll H$)

and solve

$$\partial_t u_{N,\mu}(t) + P_N(\mathcal{L}_{\mu,h}(u_{N,\mu}(t))) = 0, \quad u_{N,\mu}(0) = P_N(u_{h,0})$$

with appropriate projection $P_N : V_h \longrightarrow V_N$.

The Problem

$$\partial_t u_{N,\mu}(t) + P_N(\mathcal{L}_{\mu,h}(u_{N,\mu}(t))) = 0, \quad u_{N,\mu}(0) = P_N(u_{h,0})$$

- Empirical operator interpolation to handle $\mathcal{L}_\mu$

Stephan Rave (stephan.rave@wwu.de)

The Problem

$$\partial_t u_{N,\mu}(t) + P_N(\mathcal{L}_{\mu,h}(u_{N,\mu}(t))) = 0, \quad u_{N,\mu}(0) = P_N(u_{h,0})$$

- Empirical operator interpolation to handle $\mathcal{L}_\mu$
- Greedy search to construct V_N and interpolation basis for $\mathcal{L}_\mu$, e.g. [Drohmann, Haasdonk, Ohlberger, 2012]

The Problem

$$\partial_t u_{N,\mu}(t) + P_N(\mathcal{L}_{\mu,h}(u_{N,\mu}(t))) = 0, \quad u_{N,\mu}(0) = P_N(u_{h,0})$$

- Empirical operator interpolation to handle $\mathcal{L}_\mu$
- Greedy search to construct V_N and interpolation basis for $\mathcal{L}_\mu$, e.g. [Drohmann, Haasdonk, Ohlberger, 2012]
- Exponential convergence rates are preserved by greedy search [Haasdonk, 2011]

The Problem

$$\partial_t u_{N,\mu}(t) + P_N(\mathcal{L}_{\mu,h}(u_{N,\mu}(t))) = 0, \quad u_{N,\mu}(0) = P_N(u_{h,0})$$

- Empirical operator interpolation to handle $\mathcal{L}_\mu$
- Greedy search to construct V_N and interpolation basis for $\mathcal{L}_\mu$, e.g. [Drohmann, Haasdonk, Ohlberger, 2012]
- Exponential convergence rates are preserved by greedy search [Haasdonk, 2011]
- However . . .

The Problem

$$\partial_t u_{N,\mu}(t) + P_N(\mathcal{L}_{\mu,h}(u_{N,\mu}(t))) = 0, \quad u_{N,\mu}(0) = P_N(u_{h,0})$$

- V_N has to approximate $u_{\mu,h}(t)$ for every t.

The Problem

$$\partial_t u_{N,\mu}(t) + P_N(\mathcal{L}_{\mu,h}(u_{N,\mu}(t))) = 0, \quad u_{N,\mu}(0) = P_N(u_{h,0})$$

- V_N has to approximate $u_{\mu,h}(t)$ for every t.
- If $u_{\mu,h}$ has low regularity and moves in space, this is really bad.

The Problem

$$\partial_t u_{N,\mu}(t) + P_N(\mathcal{L}_{\mu,h}(u_{N,\mu}(t))) = 0, \quad u_{N,\mu}(0) = P_N(u_{h,0})$$

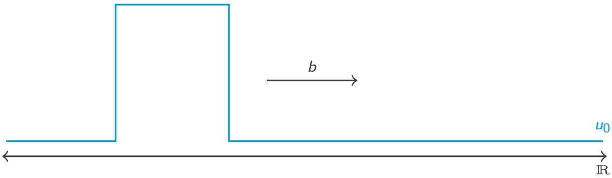
- V_N has to approximate $u_{\mu,h}(t)$ for every t.
- If $u_{\mu,h}$ has low regularity and moves in space, this is really bad.
- Even for a single parameter!

The Problem

$$\partial_t u_{N,\mu}(t) + P_N(\mathcal{L}_{\mu,h}(u_{N,\mu}(t))) = 0, \quad u_{N,\mu}(0) = P_N(u_{h,0})$$

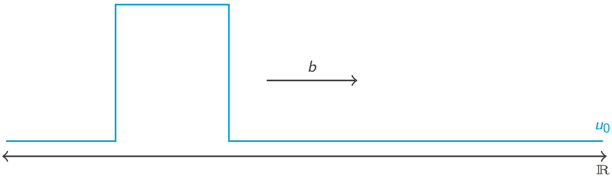
- V_N has to approximate $u_{\mu,h}(t)$ for every t.
- If $u_{\mu,h}$ has low regularity and moves in space, this is really bad.
- Even for a single parameter!
- Worse: Velocity can depend on parameter!

The Problem



$$\partial_t u(t,x) + b \cdot \partial_x u(t,x) = 0$$
$$u(0,x) = u_0(x)$$
$$x \in \mathbb{R}, t \in [0, T]$$

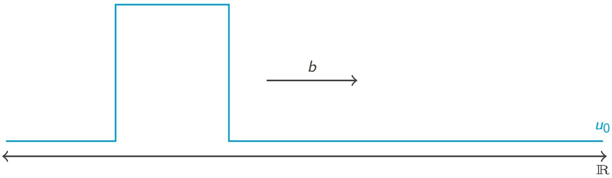
The Problem



$$\partial_t u(t,x) + b \cdot \partial_x u(t,x) = 0$$
$$u(0,x) = u_0(x)$$
$$x \in \mathbb{R}, t \in [0, T]$$

- Need $\mathcal{O}(\varepsilon^{-2})$ basis functions for L^2-approximation error $< \varepsilon$

The Problem



$$\partial_t u(t,x) + b \cdot \partial_x u(t,x) = 0$$
$$u(0,x) = u_0(x)$$
$$x \in \mathbb{R}, t \in [0,T]$$

- Need $\mathcal{O}(\varepsilon^{-2})$ basis functions for L^2-approximation error $< \varepsilon$
- **However**, we can describe solution easily by:

$$u(t,x) = u_0(x - bt)$$

Nonlinear Approximation

- Rewrite $u(t, x)$ as

$$u(t, x) = u_0(x - bt) = \Phi_{bt}(u_0)(x)$$

with $\Phi_g(v)(x) := v(x - g)$.

Nonlinear Approximation

- Rewrite $u(t,x)$ as
$$u(t,x) = u_0(x - bt) = \Phi_{bt}(u_0)(x)$$
with $\Phi_g(v)(x) := v(x-g)$.

- $g.v := \Phi_g(v)$ defines action of additive group $\mathbb{R}$:
$$(g+h).v = \Phi_{g+h}(v) = \Phi_g(\Phi_h(v)) = g.(h.v)$$

Nonlinear Approximation

- Rewrite $u(t,x)$ as

$$u(t,x) = u_0(x - bt) = \Phi_{bt}(u_0)(x)$$

with $\Phi_g(v)(x) := v(x - g)$.

- $g.v := \Phi_g(v)$ defines action of additive group $\mathbb{R}$:

$$(g+h).v = \Phi_{g+h}(v) = \Phi_g(\Phi_h(v)) = g.(h.v)$$

- **General idea:** Write $u(t,x)$ as

$$u(t,x) = g(t).v(t,x)$$

for group G acting on function space V.

Nonlinear Approximation

- **General idea:** Write $u(t, x)$ as

$$u(t,x) = \underbrace{g(t)}_{\substack{\text{dynamics of } u \\ \text{large variation in time}}}.\underbrace{v(t,x)}_{\substack{\text{shape of } u \\ \text{small variation in time}}}$$

Nonlinear Approximation

- **General idea:** Write $u(t,x)$ as

$$u(t,x) = g(t).v(t,x)$$

dynamics of u
large variation in time

shape of u
small variation in time

- If this can be done, then $v(t,x)$ will be easier to approximate by a low-dimensional linear space than $u(t,x)$.

Stephan Rave (stephan.rave@wwu.de)

Lie Groups

- Problem: How to calculate/make sense of

$$\frac{d}{dt}\, g(t).v(t,x)\ ?$$

Have to derive $g(t) \in G$ and the action of G on V.

Lie Groups

- Problem: How to calculate/make sense of

$$\frac{d}{dt}\, g(t).v(t,x)\ ?$$

Have to derive $g(t) \in G$ and the action of G on V.

Definition

A Lie group is a group G which is at the same time a smooth manifold such that group multiplication and inversion are smooth maps.

The Method of Freezing

$$\partial_t u(t) + \mathcal{L}(u(t)) = 0, \quad u(0) = u_0$$

- Substitue the *ansatz* $u(t) = g(t).v(t)$:

$$\partial_t g(t).v(t) + g(t).\partial_t v(t) + \mathcal{L}(g(t).v(t)) = 0$$

(G Lie group, action smooth)

The Method of Freezing

$$\partial_t u(t) + \mathcal{L}(u(t)) = 0, \quad u(0) = u_0$$

- Substitue the *ansatz* $u(t) = g(t).v(t)$:

$$\partial_t g(t).v(t) + g(t).\partial_t v(t) + \mathcal{L}(g(t).v(t)) = 0$$

(G Lie group, action smooth)

- Multiply by $g(t)^{-1}$:

$$\partial_t v(t) + g(t)^{-1}.\,\mathcal{L}(g(t).v(t)) + \mathfrak{g}(t).v(t) = 0$$
$$\mathfrak{g}(t) = g(t)^{-1}\partial_t g(t).$$

The Method of Freezing

$$\partial_t v(t) + g(t)^{-1}.\mathcal{L}(g(t).v(t)) + \mathfrak{g}(t).v(t) = 0$$
$$\mathfrak{g}(t) = g(t)^{-1}\partial_t g(t)$$

The Method of Freezing

$$\partial_t v(t) + g(t)^{-1}.\mathcal{L}(g(t).v(t)) + \mathfrak{g}(t).v(t) = 0$$
$$\mathfrak{g}(t) = g(t)^{-1}\partial_t g(t)$$

- Have $\dim(G)$ additional degrees of freedom!

The Method of Freezing

$$\partial_t v(t) + g(t)^{-1}.\mathcal{L}(g(t).v(t)) + \mathfrak{g}(t).v(t) = 0$$
$$\mathfrak{g}(t) = g(t)^{-1}\partial_t g(t)$$

- Have $\dim(G)$ additional degrees of freedom!
- Add additional algebraic constraint (phase condition)

$$\Phi(v(t), \mathfrak{g}(t)) = 0.$$

The Method of Freezing

$$\partial_t v(t) + g(t)^{-1}.\mathcal{L}(g(t).v(t)) + \mathfrak{g}(t).v(t) = 0$$
$$\mathfrak{g}(t) = g(t)^{-1}\partial_t g(t)$$

- Have $\dim(G)$ additional degrees of freedom!
- Add additional algebraic constraint (phase condition)

$$\Phi(v(t), \mathfrak{g}(t)) = 0.$$

- Further assume invariance of $\mathcal{L}$ under action of G:

$$h^{-1}.\mathcal{L}(h.w) = \mathcal{L}(w) \quad \text{for all } h \in G, w \in V.$$

The Method of Freezing

Definition

The method of freezing for $\partial_t u(t) + \mathcal{L}(u(t)) = 0$ consists in solving

$$\begin{aligned} \partial_t v(t) + \mathcal{L}(v(t)) + \mathfrak{g}(t).v(t) &= 0 \\ \Phi(v(t), \mathfrak{g}(t)) &= 0 \end{aligned} \qquad \text{frozen PDAE}$$

$$\mathfrak{g}(t) = g(t)^{-1}\partial_t g(t) \qquad \text{reconstruction equation}$$

with initial conditions $v(0) = u(0),\ g(0) = e$.

living.knowledge WWU Münster

The Method of Freezing

Definition

The method of freezing for $\partial_t u(t) + \mathcal{L}(u(t)) = 0$ consists in solving

$$\begin{aligned} \partial_t v(t) + \mathcal{L}(v(t)) + \mathfrak{g}(t).v(t) &= 0 \\ \Phi(v(t), \mathfrak{g}(t)) &= 0 \end{aligned} \qquad \text{frozen PDAE}$$

$$\mathfrak{g}(t) = g(t)^{-1}\partial_t g(t) \qquad \text{reconstruction equation}$$

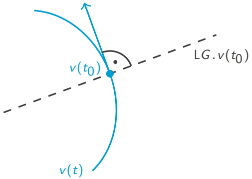
with initial conditions $v(0) = u(0),\ g(0) = e$.

- Introduced for stability analysis of relative equilibria [Beyn, Thümmler, 2004] and [Rowley et. al., 2003]

Phase Conditions

- Possible choice:

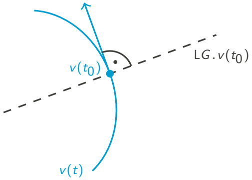
$$\Phi(v, \mathfrak{g}) = 0 \;:\Longleftrightarrow\; \partial_t v(t) \perp \mathrm{L}G.v(t)$$
$$\Longleftrightarrow\; (\mathcal{L}(v) + \mathfrak{g}.v,\; \mathfrak{h}.v) = 0 \quad \forall \mathfrak{h} \in \mathrm{L}G$$



Phase Conditions

- Possible choice:

$$\begin{aligned}\Phi(v, \mathfrak{g}) = 0 \;:&\Longleftrightarrow \partial_t v(t) \perp LG.v(t) \\ &\Longleftrightarrow (\mathcal{L}(v) + \mathfrak{g}.v,\ \mathfrak{h}.v) = 0 \quad \forall \mathfrak{h} \in LG\end{aligned}$$



- Other choices: minimize $||\partial_t v||$ or $||v - v_0||$ for some template function v_0

Example: 2D-Shifts

- $G = \mathbb{R}^2$, $\mathrm{L}G = \mathbb{R}^2$,

$$
\begin{aligned}
g.u(x) &:= u(x-g), \quad x \in \mathbb{R}^2 \\
\mathfrak{g}.u &= -\mathfrak{g} \cdot \nabla u
\end{aligned}
$$

Example: 2D-Shifts

- $G = \mathbb{R}^2$, $\mathrm{L}G = \mathbb{R}^2$,

$$\begin{aligned} g.u(x) &:= u(x-g), \quad x \in \mathbb{R}^2 \\ \mathfrak{g}.u &= -\mathfrak{g} \cdot \nabla u \end{aligned}$$

- Phase Condition:

$$\Phi(v, \mathfrak{g}) = 0 \iff (\mathcal{L}(v) + \mathfrak{g}.v,\ \mathfrak{h}.v) = 0 \quad \forall \mathfrak{h} \in \mathrm{L}G$$

Example: 2D-Shifts

- $G = \mathbb{R}^2$, $\mathrm{L}G = \mathbb{R}^2$,

$$\begin{aligned} g.u(x) &:= u(x-g), \quad x \in \mathbb{R}^2 \\ \mathfrak{g}.u &= -\mathfrak{g}\cdot\nabla u \end{aligned}$$

- Phase Condition:

$$\begin{aligned} \Phi(v,\mathfrak{g}) = 0 &\Longleftrightarrow (\mathcal{L}(v) + \mathfrak{g}.v,\, \mathfrak{h}.v) = 0 \quad \forall \mathfrak{h} \in \mathrm{L}G \\ &\Longleftrightarrow \left[(\partial_{x_i} v,\, \partial_{x_j} v)\right]_{i,j} \cdot \left[\mathfrak{g}_j\right]_j = \left[(\mathcal{L}(v),\, v_{x_r})\right]_i \\ &\qquad\qquad\qquad\qquad 1 \le i,j \le 2 \end{aligned}$$

Example: 2D-Shifts

The Method of Freezing for 2D-Shifts

Solve

$$\partial_t v(t) + \mathcal{L}(v(t)) - \mathfrak{g}(t) \cdot \nabla v(t) = 0$$
$$\left[(\partial_{x_i} v,\, \partial_{x_j} v)\right]_{i,j} \cdot \left[\mathfrak{g}_j\right]_j = \left[(\mathcal{L}(v),\, \partial_{x_i} v)\right]_i$$

and

$$\partial_t g(t) = \mathfrak{g}(t)$$

with initial conditions $v(0) = u(0)$, $g(0) = (0,0)^T$.

Example

Consider on $\Omega = [0,2] \times [0,1]$ the two-dimensional Burgers-type problem

$$\partial_t u = -\nabla \cdot (bu^\mu)$$
$$u(0, x_1, x_2) = 1/2(1 + \sin(2\pi x_1)\sin(2\pi x_2))$$

for $t \in [0, 0.3]$, $b = (1,1)^T$ with periodic boundary conditions and $\mu \in \mathcal{P} = [1,2]$.

Example

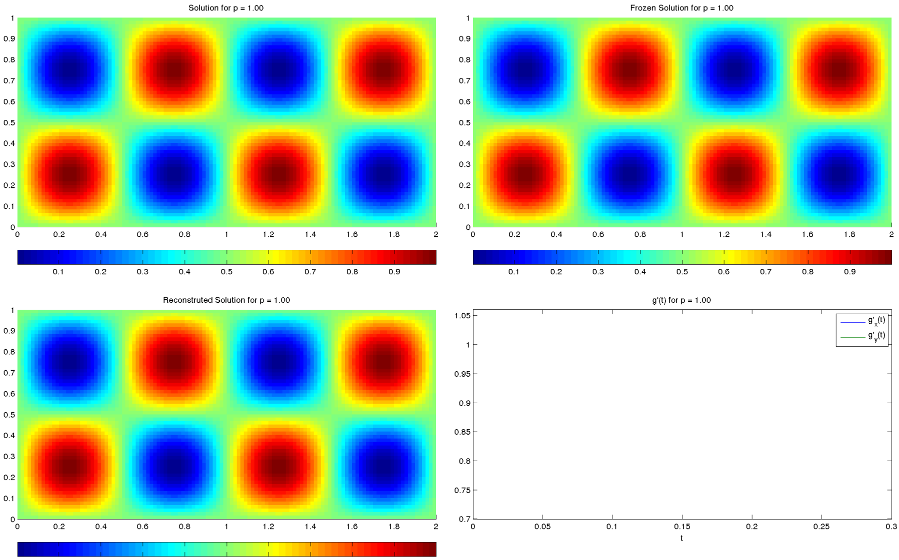
Consider on $\Omega = [0,2] \times [0,1]$ the two-dimensional Burgers-type problem

$$\partial_t u = -\nabla \cdot (bu^\mu)$$
$$u(0, x_1, x_2) = 1/2(1 + \sin(2\pi x_1)\sin(2\pi x_2))$$

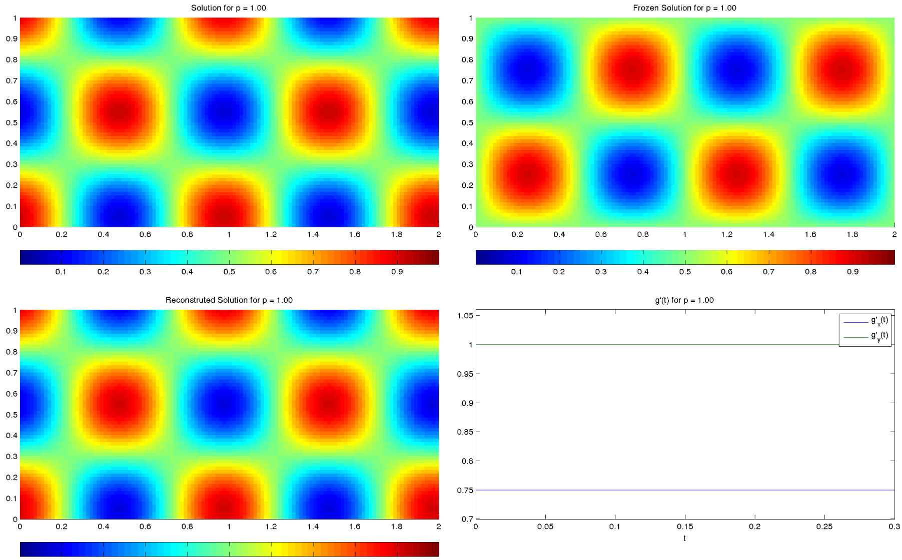
for $t \in [0, 0.3]$, $b = (1,1)^T$ with periodic boundary conditions and $\mu \in \mathcal{P} = [1,2]$.

- Finite volume discretization on 120 x 60 grid, explict Euler time-stepping
- Same problem as in [Drohmann, Haasdonk, Ohlberger, 2012]

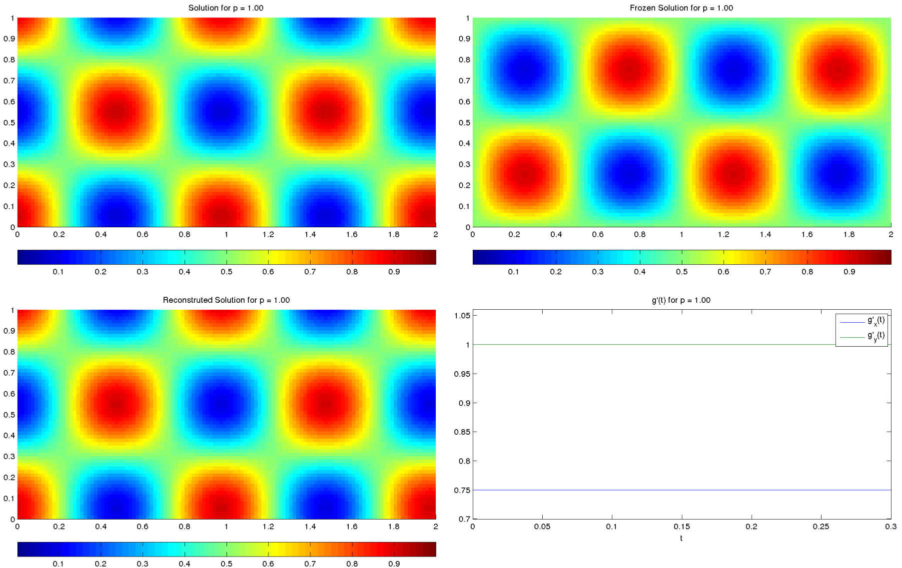
Frozen vs. Non-frozen Solution (μ=1, b=(0.75,1))



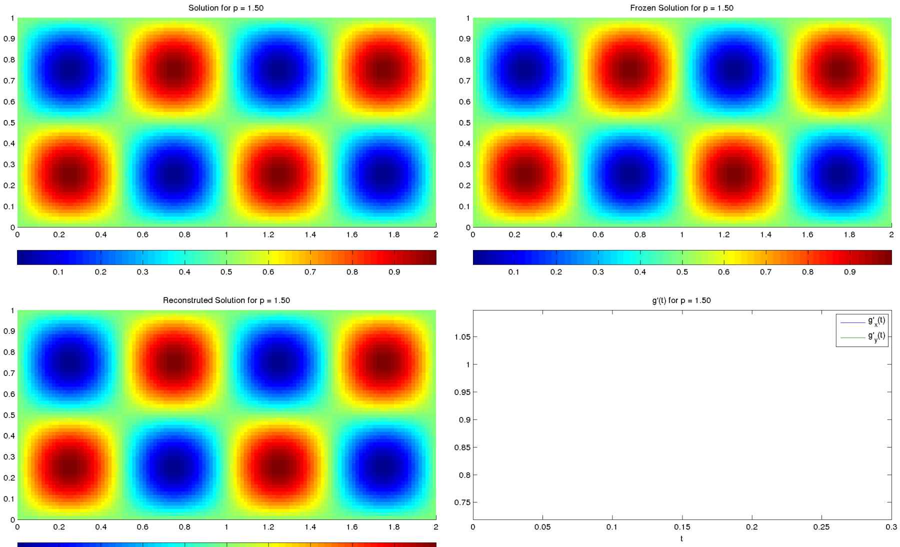
Frozen vs. Non-frozen Solution (μ=1, b=(0.75,1))



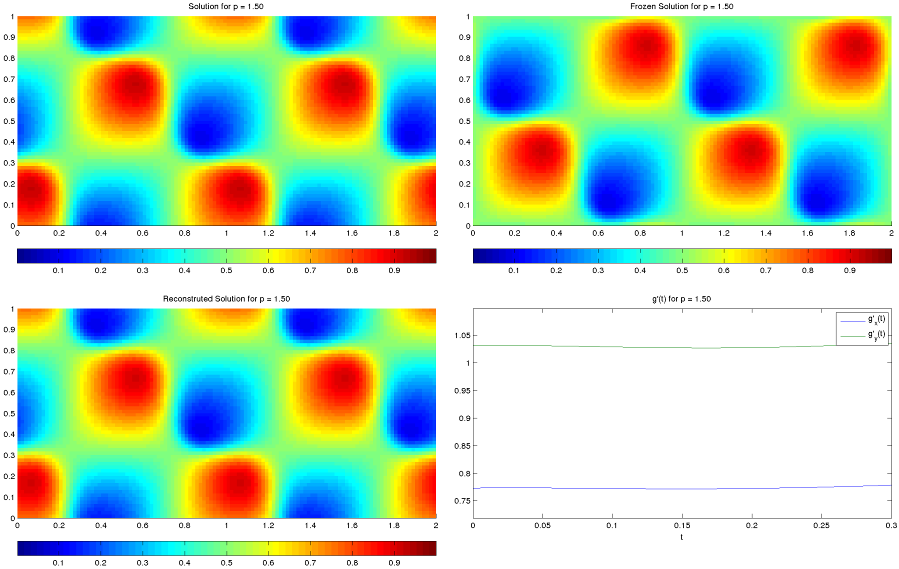
Frozen vs. Non-frozen Solution (μ=1, b=(0.75,1))



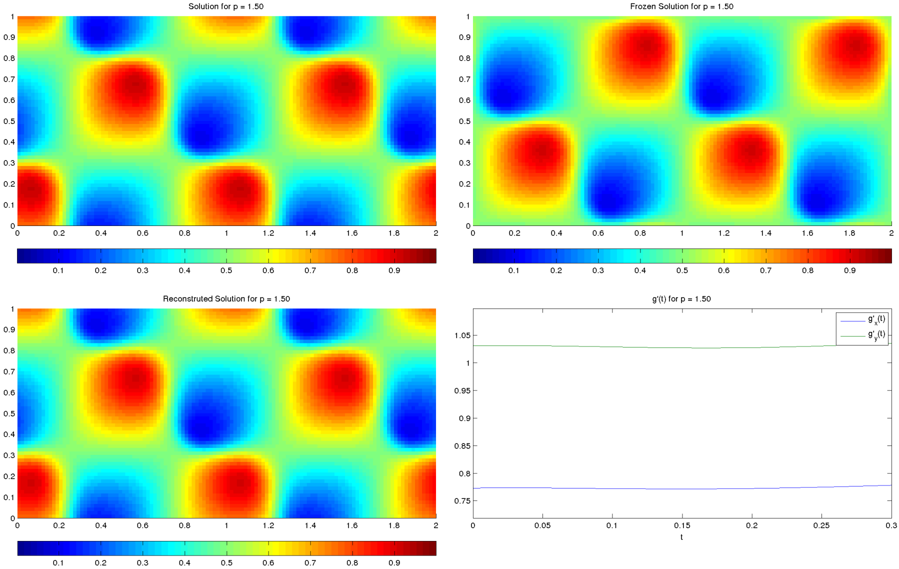
Frozen vs. Non-frozen Solution (μ=1.5, b=(0.75,1))



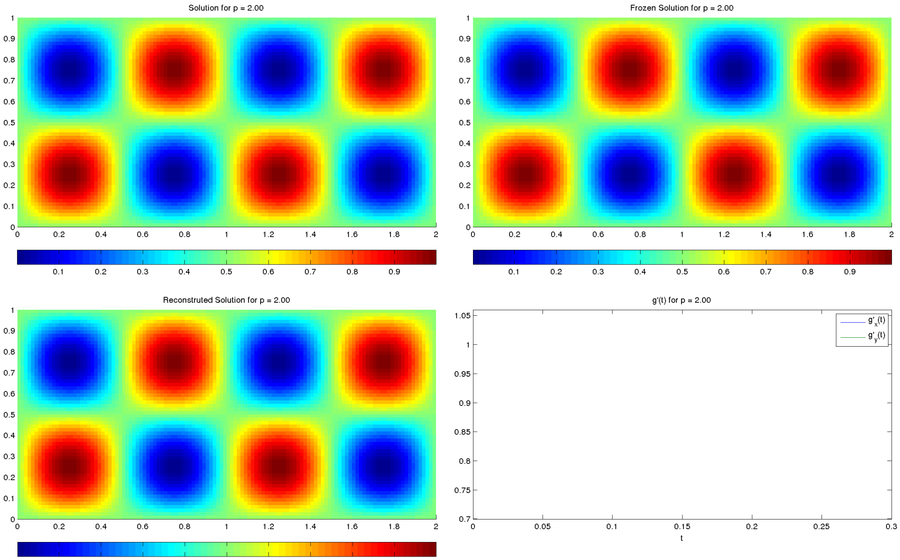
Frozen vs. Non-frozen Solution (μ=1.5, b=(0.75,1))



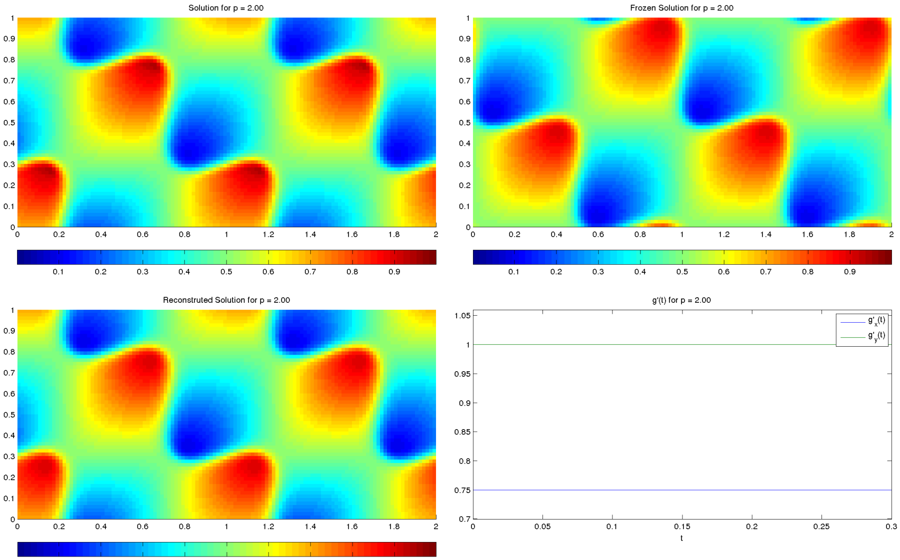
Frozen vs. Non-frozen Solution (μ=1.5, b=(0.75,1))



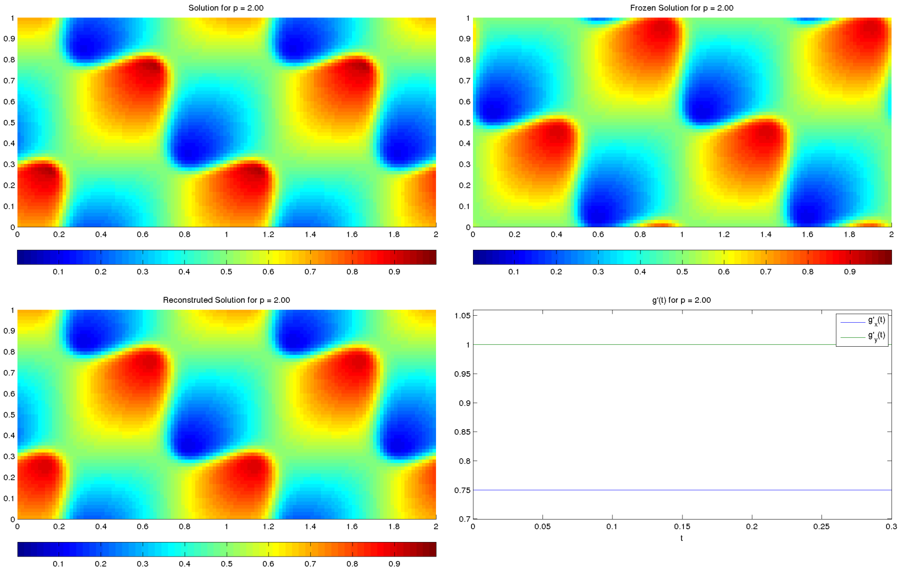
Frozen vs. Non-frozen Solution (μ=2, b=(0.75,1))



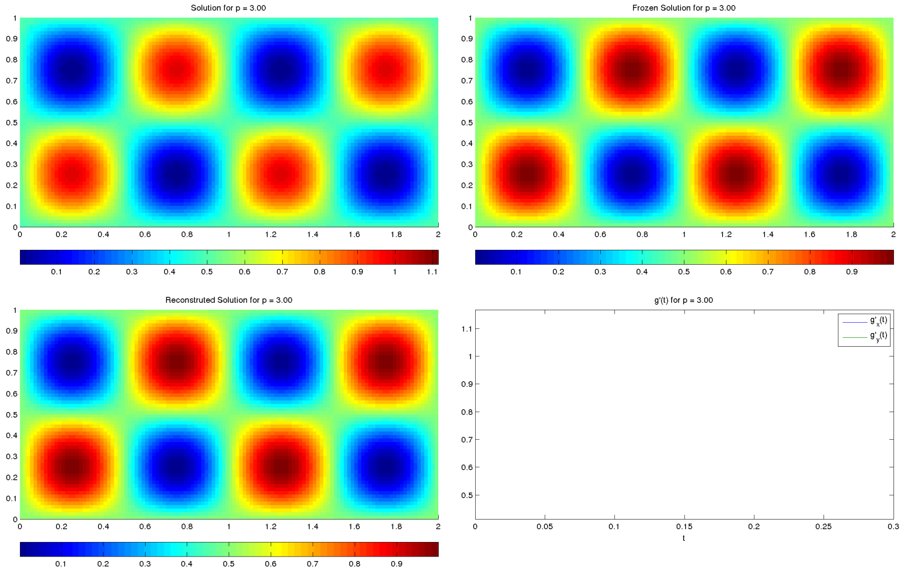
Frozen vs. Non-frozen Solution (μ=2, b=(0.75,1))



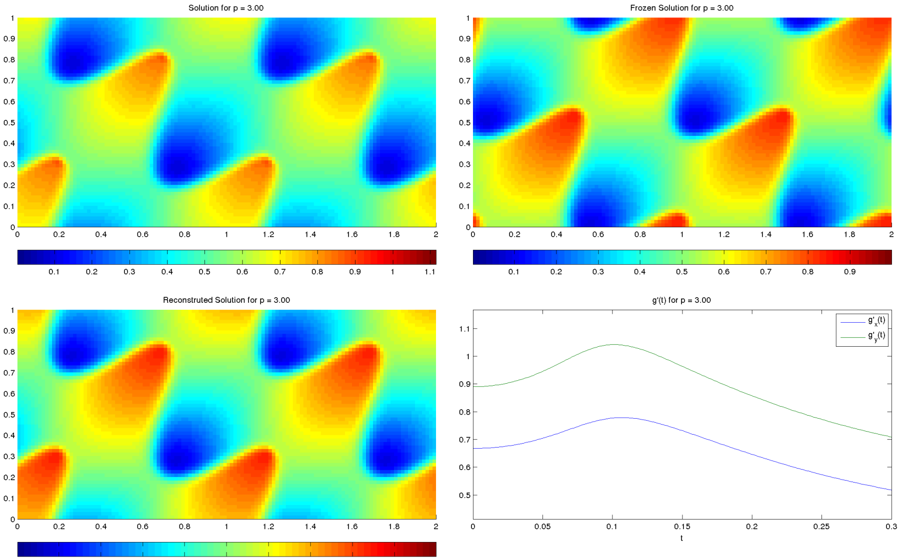
Frozen vs. Non-frozen Solution (μ=2, b=(0.75,1))



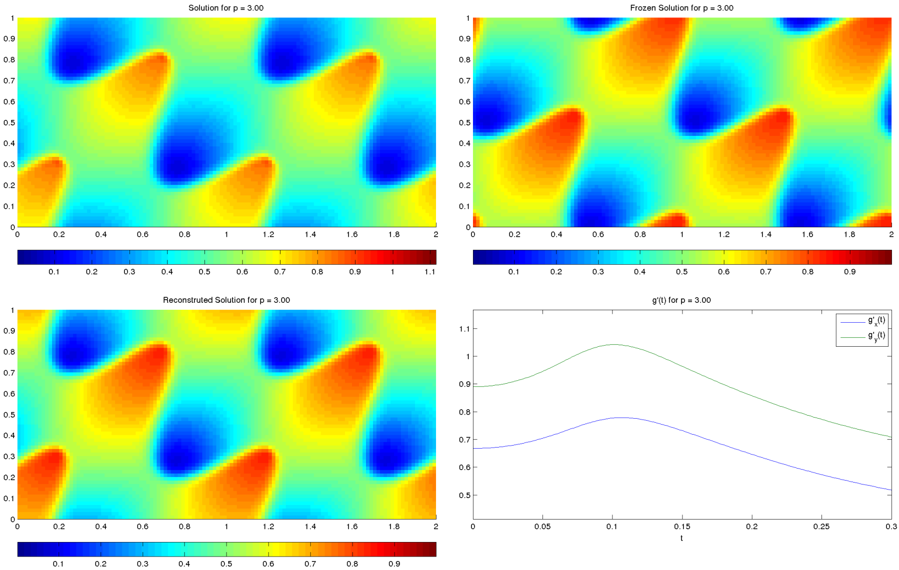
Frozen vs. Non-frozen Solution (μ=3, b=(0.75,1))



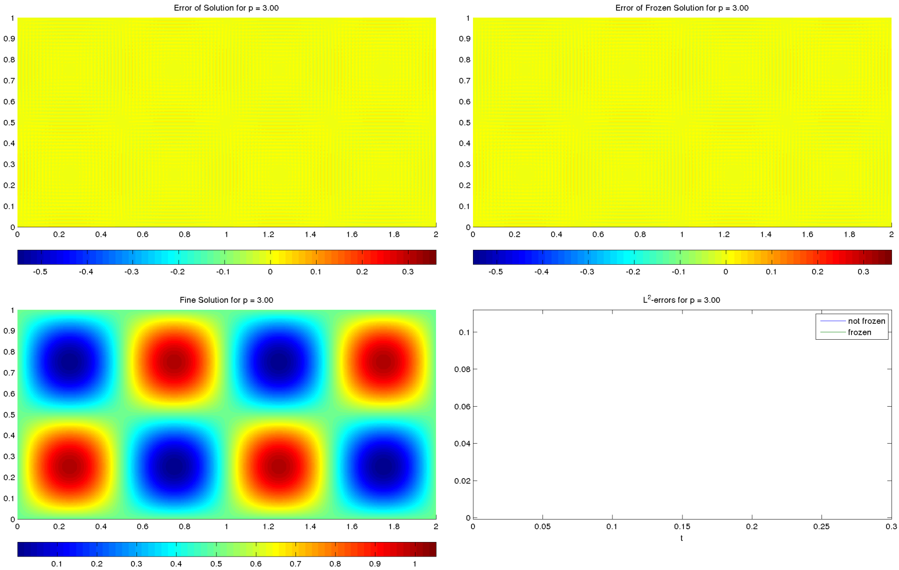
Frozen vs. Non-frozen Solution (μ=3, b=(0.75,1))



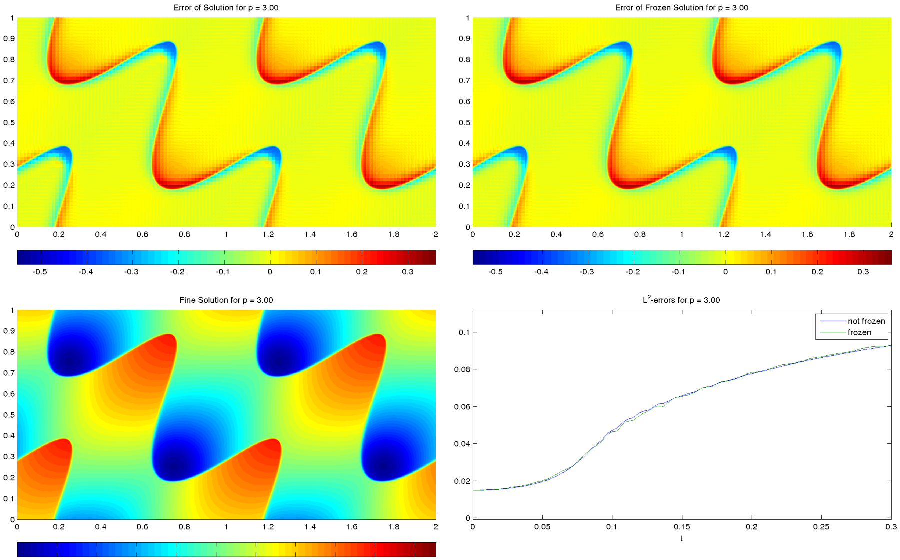
Frozen vs. Non-frozen Solution (μ=3, b=(0.75,1))



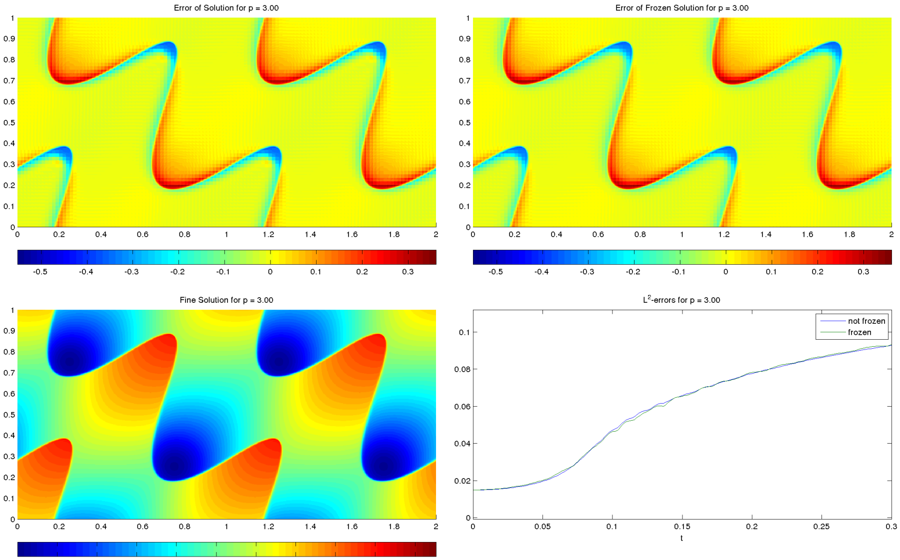
Errors of Frozen and Non-frozen Solution (μ=3)



Errors of Frozen and Non-frozen Solution (μ=3)



Errors of Frozen and Non-frozen Solution (μ=3)



RB-Approximation

RB-Approximation

FrozenRB-Scheme [Ohlberger, R., 2013]

1. Replace original parameterized PDE with frozen parameterized PDAE and reconstruction ODE
2. Discretize
3. Use EI and greedy basis generation

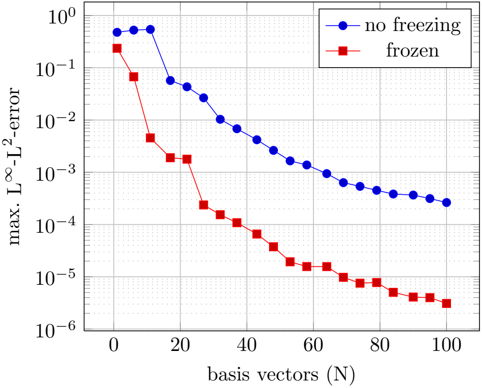
RB-Approximation

FrozenRB-Scheme [Ohlberger, R., 2013]

1. Replace original parameterized PDE with frozen parameterized PDAE and reconstruction ODE
2. Discretize
3. Use EI and greedy basis generation

- Offline/online decomposition possible
- No additional evaluations of nonlinearity (small overhead)

RB-Approximation Error for Burgers Problem



Model Order Reduction with pyMOR

pyMOR

- Software library for writing MOR applications, in particular with the reduced basis method.
- Joint with Felix Schindler and Rene Milk.
- Completely written in Python.
- Started 2012, 14k lines of code.
- BSD-Licensed, hosted on Github.
- `http://pymor.org/`

Goals of Development

- Tool for education.
- Research platform for rapid development of new model reduction methods.
- Ready to use in real world applications.
- In particular, high interoperability with foreign code.

Main Design Principles

- Provide abstract interfaces between MOR code and high-dimensional solver.
  - Deep access to solver code allowing various algorithms to use the same interface.
  - No MOR-specific code inside solver.
  - Think of solver as a library.
  - Agnostic about the specific implementation of communication between pyMOR and solver.

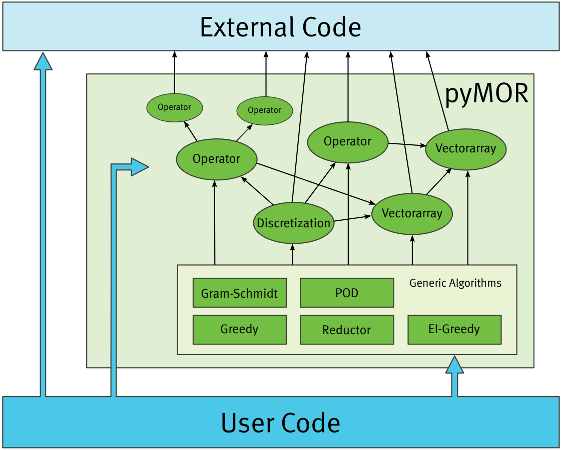
Main Design Principles

- Provide abstract interfaces between MOR code and high-dimensional solver.
  - Deep access to solver code allowing various algorithms to use the same interface.
  - No MOR-specific code inside solver.
  - Think of solver as a library.
  - Agnostic about the specific implementation of communication between pyMOR and solver.
- Implement broad library of MOR algorithms in terms of these interfaces.
  - Make it easy to tests the mathematics and care of about performance later.
  - Provide building blocks, not complete solutions.

Main Design Principles

- Provide infrastructure for running MOR algorithms:
  - Handling of parameters and parameter spaces.
  - Caching (memory, disk) of high-dimensional solutions.
  - Handling of application-wide defaults.
  - Visualization.
- Implement basic high-dimensional discretizations to get started quickly.

Architecture of pyMOR



Algorithms

- Gram-Schmidt, POD.
- Greedy basis generation.
- Automatic reduction of arbitrarily nested affine combinations of operators.
- Interpolation of arbitrary (nonlinear) operators, EI-Greedy, DEIM.
- Iterative linear solvers, Newton algorithm.
- Time-stepping algorithms.

Why Python?

- Not as fast as C, but fast enough.
- Agile software development.
  - No static typing.
  - Interactive interpreter/debugger sessions.
  - Expressive syntax.
- Great for scientific computing
  - `NumPy` matrix class with `MATLAB`™-like performance. (Similar interface, 1 day to learn major differences.)
  - Great support for nd-arrays.
  - Huge ecosystem of scientific computing libraries: SciPy, Matplotlib, Pandas, Pillow, IPython, SymPy, scikit-learn, scikit-bio, PyAMG, FENICS
  - Annual international conferences: SciPy (US), EuroSciPy, ...

Why Python?

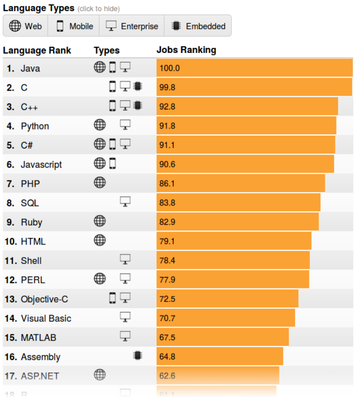
- Plays nicely with other programming languages.
- 51909 packages on PyPI (Python Package Index).
  - GUI-Toolkits.
  - Databases.
  - Networking.
  - File formats.
- Open source.
  - It's for free! (Also for your cluster.)
  - No proprietary algorithms.
  - Easy to extend/modify core features.
  - Easy to contribute.
  - Future independent of well-being of single company.
  - Sometimes a bit chaotic.

Why Python?

- It's a great language!
  - Beautiful (=readable) code.
  - Object orientation.
  - Namespaces.
  - Does not blow up your terminal if you forget a `;`
  - `a += 2`
  - `def func(x, y, option1=value1, option2=value2):`

Python in Education

- Clean language design.
- Easy to learn.
- Showcases many important concepts in programming.
- First language in many CS courses.
- Widely adopted in industry.



source: IEEE.Spectrum 2014 programming language ranking

Live Demo!

Thank you for your attention!

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

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

Model Reduction for Parameterized Systems
`http://morepas.org`