Reduced Basis Methods for inverse problems

Dominik Garmatter

dominik.garmatter@mathematik.uni-stuttgart.de

Chair of Optimization and Inverse Problems, University of Stuttgart, Germany

Joint work with Bastian Harrach and Bernard Haasdonk

Distinguished Lectures on Inverse Problems
Helsinki, Finland, August 4-8, 2014.

Reduced Basis Methods

Motivation I

- Linear elliptic parametrized PDE of the form

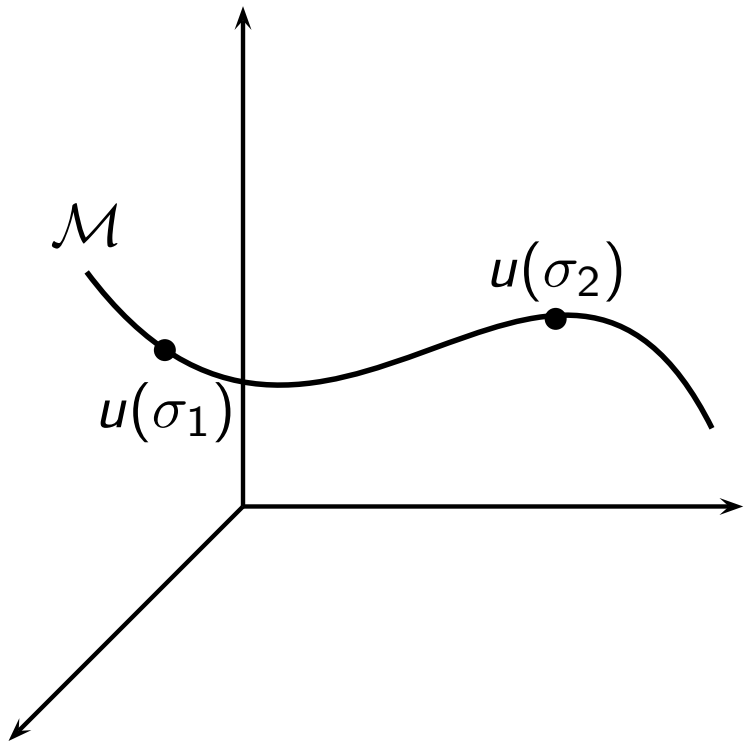
$$\mathcal{L}u(\mathbf{x};\sigma) = f, \text{ in } \Omega \tag{1a}$$
$$u(\mathbf{x};\sigma) = g, \text{ on } \partial\Omega \tag{1b}$$

- Solution of (1) for many different parameters in a small amount of time (e.g. design optimization, optimal control, online-simulation, inverse problems)
- Detailed solution (e.g. FEM, FV, FD) is rather expensive

$\rightsquigarrow$ **model order reduction**

Motivation II

- Approximating the solution manifold $\mathcal{M} := \{u(\sigma) | \sigma \in \mathcal{P}\}$, $\mathcal{P} \subset \mathbb{R}^p$ a bounded parameter domain, with a low dimensional subspace $X_N \subset X$
- Construction of X_N with so called „snapshots“, i.e. $X_N \subseteq \text{span}\{u(\sigma_1), \ldots, u(\sigma_N)\}$ with *meaningful* parameters $\sigma_1, \ldots, \sigma_N \in \mathcal{P}$



The detailed and reduced problem

Detailed problem:
Let X be a Hilbert space, $\mathcal{P} \subset \mathbb{R}^p$ a bounded parameter domain, $b : X \times X \times \mathcal{P} \to \mathbb{R}$ a continuous, coercive, symmetric bilinearform, $f : X \times \mathcal{P} \to \mathbb{R}$ a continuous linearform. For a given parameter $\sigma \in \mathcal{P}$, we seek the **detailed solution** $u(\sigma) \in X$ of

$$b(u(\sigma), v; \sigma) = f(v; \sigma), \quad \forall v \in X. \tag{P}$$

The detailed and reduced problem

Detailed problem:
Let X be a Hilbert space, $\mathcal{P} \subset \mathbb{R}^p$ a bounded parameter domain, $b : X \times X \times \mathcal{P} \to \mathbb{R}$ a continuous, coercive, symmetric bilinearform, $f : X \times \mathcal{P} \to \mathbb{R}$ a continuous linearform. For a given parameter $\sigma \in \mathcal{P}$, we seek the **detailed solution** $u(\sigma) \in X$ of

$$b(u(\sigma), v; \sigma) = f(v; \sigma), \quad \forall v \in X. \tag{P}$$

Reduced problem:
Let a Problem (P) be given and let $X_N \subset X$ be a reduced basis space (RB-space). For a given parameter $\sigma \in \mathcal{P}$, we seek the **reduced solution** $u_N(\sigma) \in X_N$ of

$$b(u_N(\sigma), v; \sigma) = f(v; \sigma), \quad \forall v \in X_N. \tag{PN}$$

The reduced basis space X_N

Error estimator

$$\|u - u_N\|_X \leq \Delta_N(\sigma) := \frac{\|v_r\|_X}{\alpha(\sigma)}, \text{ with}$$

$$\langle v_r, v \rangle_X := r(v;\sigma) := f(v;\sigma) - b(u_N(\sigma), v;\sigma), \forall v \in X$$

Algorithm 1 Greedy-Procedure

1: $X_N := \{0\},\ \Phi_N := \emptyset,\ M_{train},\ \varepsilon_{tol},\ \Delta_N(\sigma)$
2: **repeat**
3: $\quad \sigma^\star := \arg\max_{\sigma \in M_{train}} \Delta_N(\sigma)$
4: $\quad \phi := u(\sigma^\star),\ \Phi_N := \Phi_N \cup \phi,\ X_N := X_N + \text{span}(\phi)$
5: $\quad \varepsilon := \max_{\sigma \in M_{train}} \Delta_N(\sigma)$
6: **until** $\varepsilon \leq \varepsilon_{tol}$
7: **return** $\Phi_N,\ X_N$

Offline/Online Decomposition

Assumption: b and f are affine in parameter, i.e.

$$b(u, v; \sigma) = \sum_{q=1}^{Q_b} \Theta_b^q(\sigma) b^q(u, v),$$
$$f(v; \sigma) = \sum_{q=1}^{Q_f} \Theta_f^q(\sigma) f^q(v), \forall u, v \in X.$$

Offline-phase (once)

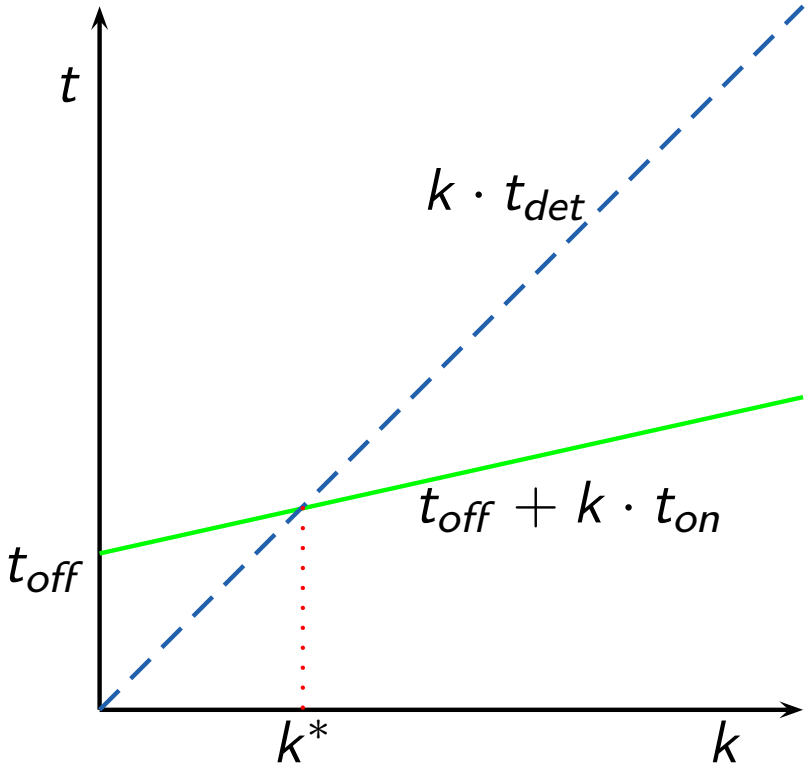
- Compute RB $\Phi_N = \{\phi_1, \dots, \phi_N\}$ and RB-space X_N
- Compute **parameter-independant** components $\underline{B}_N^q := (b^q(\phi_i, \phi_j))_{i,j=1}^N \in \mathbb{R}^{N \times N}$ and $\underline{f}_N^q := (f^q(\phi_i))_{i=1}^N \in \mathbb{R}^N$

Online-phase (for each new σ)

- Evaluate **parameter-dependant** coefficients $\Theta_b^q(\sigma)$, $\Theta_f^q(\sigma)$
- Assemble $\underline{B}_N(\sigma)$, $\underline{f}_N(\sigma)$ and solve $\underline{B}_N \underline{u}_N = \underline{f}_N$
- Reconstruct solution of (PN) via $u_N = \sum_i \underline{u}_{N,i} \phi_i$

Conclusion RBM

- Can accelerate forward solver of a PDE in a multi-query context due to efficient offline/online decomposition
- Greedy-scheme to construct RB-space X_N
- Approximates high-dimensional solution u (FEM, FD, FV)



RBM for inverse problems

Model problem

Forward problem:
Non-linear operator $F : \mathcal{D}(F) \subset Y \to X$ between Hilbert spaces maps a parameter $\sigma \in \mathcal{D}(F)$ to a solution u of a given PDE:

$$F(\sigma) = u.$$

Inverse problem:
For a given PDE-solution $u \in X$ find the corresponding parameter $\sigma \in \mathcal{D}(F)$ with $F(\sigma) = u$.

Numerical approach

Given u and exact solution σ^+ (s.t. $F(\sigma^+) = u$) and noisy data u^δ with $\|F(\sigma^+) - u^\delta\| \leq \delta$

- Consider $\sigma^\delta_{n+1} = \sigma^\delta_n + s^\delta_n$ with starting value $\sigma^\delta_0 \in \mathcal{D}(F)$
- Want s^δ_n to approximate $s^e_n := \sigma^+ - \sigma^\delta_n$
- If F is Fréchet-differentiable s^e_n solves the linear system

$$F'(\sigma^\delta_n)s^e_n = u - F(\sigma^\delta_n) - E(\sigma^+, \sigma^\delta_n)$$

- Compute s^δ_n as a solution of $F'(\sigma^\delta_n)s = u^\delta - F(\sigma^\delta_n)$

REGINN(REGularisation based on INexact Newtoniteration) [1]

Algorithm 2 REGINN($\sigma_{start}, \tau, \{\Theta_n\}$)

$n := 0,\ \sigma_0^\delta := \sigma_{start}$
while $\|F(\sigma_n^\delta) - u^\delta\| > \tau\delta$ **do**
  $i := 1$
  repeat
    $i := 2i$
    $s_{n,i} := \left(F'(\sigma_n^\delta)^* F'(\sigma_n^\delta) + \frac{1}{i}\,\mathrm{Id}\right)^{-1} F'(\sigma_n^\delta)^*(u^\delta - F(\sigma_n^\delta))$
  until $\|F'(\sigma_n^\delta)s_{n,i} + F(\sigma_n^\delta) - u^\delta\| < \Theta_n \|F(\sigma_n^\delta) - u^\delta\|$
  $\sigma_{n+1}^\delta := \sigma_n^\delta + s_{n,i}$
  $n := n + 1$
end while
$\sigma_{REGINN} := \sigma_n^\delta$

[1] see A.Rieder: Keine Probleme mit Inversen Problemen, Seite 251.

REGINN utilizing RB

Algorithm 3 REGINN_utilizing_RB($\sigma_{start}, \tau, \{\Theta_n\}, X_N$)

$n := 0,\ \sigma_0^\delta := \sigma_{start}$
while $\|F_N(\sigma_n^\delta) - u^\delta\| > \tau\delta$ **do**
  $i := 1$
  repeat
    $i := 2i$
    $s_{n,i} := \left(F'(\sigma_n^\delta)^* F'(\sigma_n^\delta) + \frac{1}{i}\,\mathrm{Id}\right)^{-1} F'(\sigma_n^\delta)^* (u^\delta - F_N(\sigma_n^\delta))$
  until $\|F'(\sigma_n^\delta) s_{n,i} + F_N(\sigma_n^\delta) - u^\delta\| < \Theta_n \|F_N(\sigma_n^\delta) - u^\delta\|$
  $\sigma_{n+1}^\delta := \sigma_n^\delta + s_{n,i}$
  $n := n + 1$
end while
$\sigma_{REGINN,RB} := \sigma_n^\delta$

Numerical toy problem

Find a solution $u \in H_0^1(\Omega) \subseteq L^2(\Omega)$ of

$$\operatorname{div}(\sigma \nabla u) = 1, \qquad \text{in } \Omega = [0,1]^2,$$

with $\sigma \in L_+^\infty(\Omega)$.

- Reduced Basis requires $\mathcal{P} \subset \mathbb{R}^p$
- Restrict to $\sigma = \sum_{i=1}^{9} \sigma_i \chi_{\Omega_i}$ with $\sigma_i \in [0.1, 10]$

Ω_7	Ω_8	Ω_9
Ω_4	Ω_5	Ω_6
Ω_1	Ω_2	Ω_3

Numerical results - How good is RB?

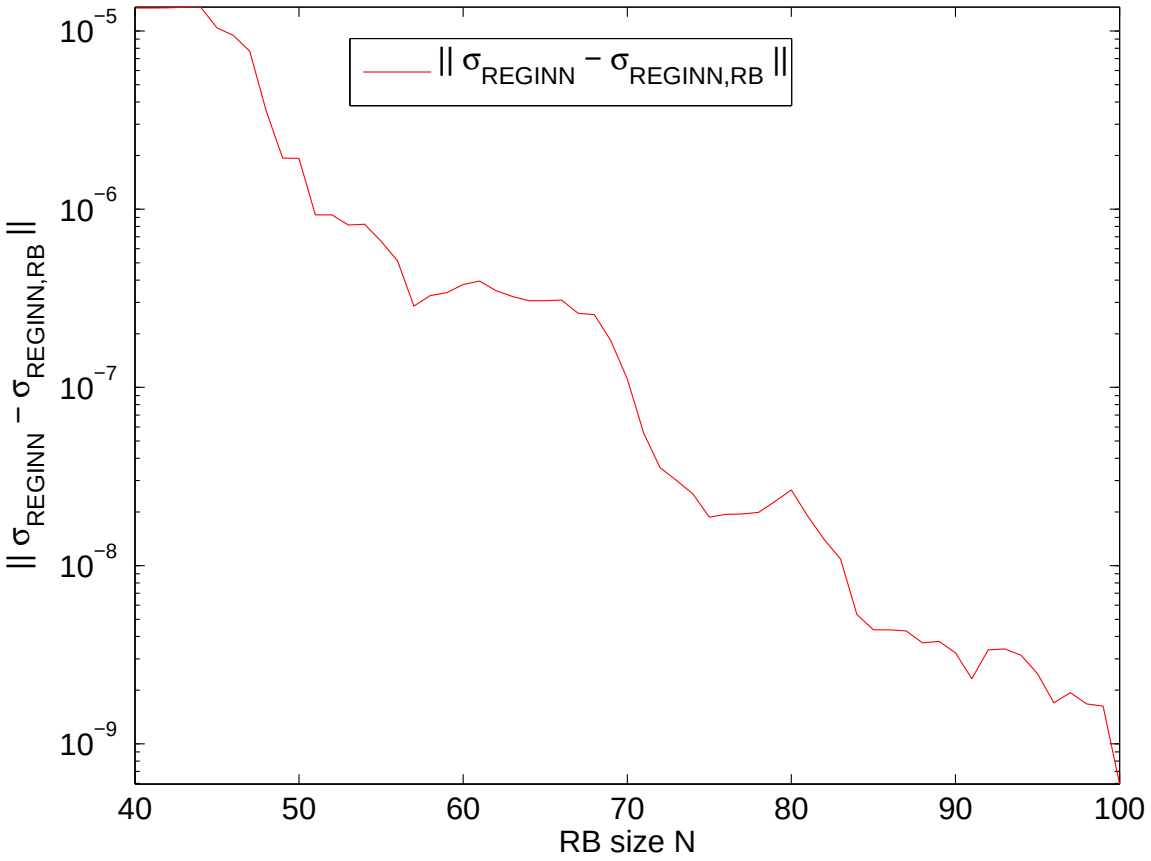


Figure : Comparison of Algorithm 2 and 3 with $\delta = 0.01$, depending on the size N of the used RB. Target parameter was $\sigma^+ = (2, 2, 2, 2, 9, 2, 2, 2, 2)$, $\sigma_{start} = (1, 1, 1, 1, 1, 1, 1, 1, 1)$.

Numerical results - When does RB pay off?

Average time of REGINN without RB (Alg. 2): ~ 4.6861 s

size RB	REGINN_RB (s)	Offline-time RB (s)	querys required
40	1.9967	8457	3144
50	2.0070	11012	4110
60	2.0091	13846	5172
70	2.0229	17331	6507
80	2.0522	21568	8188
90	2.0292	26601	10012
100	2.0293	33103	12460

Table : Average time of Algorithm 3 (using 200 random parameters with starting value $\sigma_{start} = (1, 1, 1, 1, 1, 1, 1, 1, 1)$ and $\delta = 0.01$).

Limitations of Algorithm 3

Increasing dimension of parameter space p very problematic

- For large p (> 40) RB-space covering the whole variety in parameter cannot be constructed
- (in our case) computation of Fréchet-derivative becomes more and more expensive

Current approach not feasible for large p

New approach (inspired by V. Druskin and M. Zaslavsky, 2007)

- Don't construct an a-priori RB-space approximating the whole solution manifold
- Construct a small, problem-oriented RB-space $X_{N,1}$ while solving the inverse problem
- Use a second RB-space $X_{N,2}$ containing informtion about the derivative
- Utilize the property of the regularization algorithm to determine a new meaningful parameter to enrich the RB-space

New approach - Pseudocode

Algorithm 4 new_reduced_Landweber($\sigma_{start}, \tau, \Phi_{N,1}, \Phi_{N,2}$)

1: $n := 0,\ \sigma_0^\delta := \sigma_{start},\ X_{N,1} := \text{span}(\Phi_{N,1}),\ X_{N,2} := \text{span}(\Phi_{N,2})$
2: **while** $\|F(\sigma_n^\delta) - u^\delta\| > \tau\delta$ **do**
3: $\quad \Phi_{N,1} := \Phi_{N,1} \cup F(\sigma_n^\delta),\ \Phi_{N,2} := \Phi_{N,2} \cup F'(\sigma_n^\delta)$
4: $\quad X_{N,1} = \text{span}\{\Phi_{N,1}\},\ X_{N,2} = \text{span}\{\Phi_{N,2}\}$
5: $\quad i := 1,\ \sigma_i^\delta := \sigma_n^\delta$
6: $\quad$ **repeat**
7: $\quad\quad \sigma_{i+1}^\delta := \sigma_i^\delta + F_N'(\sigma_i^\delta)^*(u^\delta - F_N(\sigma_i^\delta))$
8: $\quad\quad i := i + 1$
9: $\quad$ **until** $\|F_N(\sigma_i^\delta) - u^\delta\| > \tau\delta$
10: $\quad \sigma_{n+1}^\delta := \sigma_i^\delta$
11: $\quad n := n + 1$
12: **end while**
13: $\sigma_{final} := \sigma_n^\delta$

First Results: comparing Algorithm 2 & 4

p	# DOFs	t Alg. 2 (s)	t Alg. 4 (s)	$\|\sigma_{Alg2} - \sigma_{Alg4}\|$
9	22801	4.69	7.91	0.56
25	63001	30.12	33.66	1.51
49	123201	131.62	114.29	2.13
100	251001	546.48	394.79	3.14

Table : Average time and normdifference of Algorithm 2 & 4 (using 20 random parameters with starting value $\sigma_{start} = (1, 1, 1, 1, 1, 1, 1, 1, 1)$ and $\delta = 0.01$) for different settings.

Outlook

- Investigate Algorithm 4 and provide further numerical results
- Provide theoretical background
- Extend to more realistic examples
  - problems with an actual application (e.g. EIT)
  - parameter functions

Outlook

- Investigate Algorithm 4 and provide further numerical results
- Provide theoretical background
- Extend to more realistic examples
  - problems with an actual application (e.g. EIT)
  - parameter functions

Thank you for your attention!