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

Reduced Basis Summer School 2014
Münster, Germany, August 18-22, 2014.

Model problem

Given a linear elliptic parametrized PDE of the form

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

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

Forward problem:
Non-linear operator $F : \mathcal{D}(F) \subset Y \to X$ between Hilbert spaces Y, X maps a parameter $\sigma \in \mathcal{D}(F)$ to a solution $u(\sigma) \in X$ of (1)

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

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

Problems with inverse problems

- Naive inversion, i.e. solving $\sigma = F^{-1}(u(\sigma))$ fails for ill-posed inverse problems (F^{-1} is discontinuous!)

  $$\Rightarrow \text{Small errors get amplified!}$$

- In general only noisy data u^δ with $\|u - u^\delta\| < \delta$ (e.g. measurments) given

  $$\Rightarrow F^{-1}(u^\delta) \nrightarrow F^{-1}(u) \text{ for } \delta \to 0$$

- General idea: Replace F^{-1} by continuous approximations R_α
  Aim: $R_{\alpha(\delta,u^\delta)}(u^\delta) \to F^{-1}(u)$ for $\delta \to 0$

Regularization methods for linear problems

Regularization methods

Consider $Ks = y$, $K \in \mathcal{K}(Y, X)$, K^+ Moore-Penrose-Inverse of K.

A family $(R_\alpha)_{\alpha>0}$ of linear operators $R_\alpha : X \to Y$ is called a **regularization** of K^+ for $\alpha \to 0$, if

- $R_\alpha \in \mathcal{L}(X, Y)$, $\forall \alpha > 0$ (stability)
- $R_\alpha y \to K^+ y$, $\forall y \in \mathcal{D}(K^+)$.

Regularization methods

Consider $Ks = y$, $K \in \mathcal{K}(Y, X)$, K^+ Moore-Penrose-Inverse of K.

A family $(R_\alpha)_{\alpha>0}$ of linear operators $R_\alpha : X \to Y$ is called a **regularization** of K^+ for $\alpha \to 0$, if

- $R_\alpha \in \mathcal{L}(X, Y)$, $\forall \alpha > 0$ (stability)
- $R_\alpha y \to K^+ y$, $\forall y \in \mathcal{D}(K^+)$.

A function $\alpha : \mathbb{R}^+ \times X \to \mathbb{R}^+$, $(\delta, y^\delta) \to \alpha(\delta, y^\delta)$ is called a **parameter choice rule**.

Regularization methods

Consider $Ks = y$, $K \in \mathcal{K}(Y, X)$, K^+ Moore-Penrose-Inverse of K.

A family $(R_\alpha)_{\alpha>0}$ of linear operators $R_\alpha : X \to Y$ is called a **regularization** of K^+ for $\alpha \to 0$, if

- $R_\alpha \in \mathcal{L}(X, Y)$, $\forall \alpha > 0$ (stability)
- $R_\alpha y \to K^+ y$, $\forall y \in \mathcal{D}(K^+)$.

A function $\alpha : \mathbb{R}^+ \times X \to \mathbb{R}^+$, $(\delta, y^\delta) \to \alpha(\delta, y^\delta)$ is called a **parameter choice rule**.

A combination of a regularization and a parameter choice rule is called a **regularization method**, if for each $y \in \mathcal{D}(K^+)$

$$R_{\alpha(\delta, y^\delta)} y^\delta \to K^+ y, \text{ for } \|y^\delta - y\| \leq \delta.$$

Tikhonov regularization

- Solve $Ks = y$ as good as possible and also control the norm of that solution
- Define $s_\alpha := R_\alpha y$ as the minimizer of the **tikhonov functional**

$$s_\alpha = \operatorname{argmin}_{s \in \mathcal{D}(K)} \|Ks - y\|_X^2 - \alpha \|s\|_Y^2$$

- Can be formally written as

$$s_\alpha = (K^* K + \alpha \, \mathrm{Id})^{-1} K^* y$$

The discrepancy principle

- Residual $\|Ks_\alpha^\delta - y^\delta\|$ with $s_\alpha^\delta := R_\alpha y^\delta$ and y^δ, $\|y - y^\delta\| \leq \delta$ measures how well solution matches noisy data

  $\rightsquigarrow$ Higher precisision than δ for residual makes no sense

The parameter choice rule
„choose $\alpha(\delta, y^\delta)$, s.t. $\|Ks_\alpha^\delta - y^\delta\| \approx \delta$“ is called **discrepancy principle**.

- Choose strictly decreasing sequence $(\alpha_k)_{k\in\mathbb{N}}$, compute $s_{\alpha_k}^\delta$, check if $\|Ks_{\alpha_k}^\delta - y^\delta\| \leq \tau\delta$, with fixed $\tau > 1$.

Note: Tikhonov regularization with discrepancy principle is a regularisation method!

Back to the nonlinear problem

Recall the Model problem

Given a linear elliptic parametrized PDE of the form

$$\begin{aligned} \mathcal{L}u(\mathbf{x};\sigma) &= f, \text{ in } \Omega \\ u(\mathbf{x};\sigma) &= g, \text{ on } \partial\Omega \end{aligned}$$

Forward problem:
Non-linear operator $F : \mathcal{D}(F) \subset Y \to X$ between Hilbert spaces Y, X maps a parameter $\sigma \in \mathcal{D}(F)$ to a solution $u(\sigma) \in X$ of (1)

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

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

Iterative approach

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

- Consider $\sigma_{n+1}^\delta = \sigma_n^\delta + s_n^\delta$ with starting value $\sigma_0^\delta \in \mathcal{D}(F)$
- Want s_n^δ to approximate $s_n^e := \sigma^+ - \sigma_n^\delta$
- If F is Fréchet-differentiable s_n^e solves the linear system

$$F'(\sigma_n^\delta)s_n^e = u - F(\sigma_n^\delta) - E(\sigma^+, \sigma_n^\delta)$$

Only noisy version of RHS is known!

- Compute s_n^δ as a solution of $F'(\sigma_n^\delta)s = u^\delta - F(\sigma_n^\delta)$

$\rightsquigarrow$ Linear problem!

REGINN(REGularisation based on INexact Newtoniteration)[?]

Algorithm 1 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
$\sigma_{REGINN} := \sigma_n^\delta$

REGINN utilizing RB

Algorithm 2 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

$\sigma_{REGINN,RB} := \sigma_n^\delta$

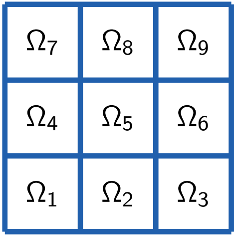
Numerical toy problem

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

$$\operatorname{div}(\sigma \nabla u) = 1 \qquad (2)$$

with $\Omega = [0,1]^2$ and $\sigma \in L_+^\infty(\Omega)$.

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



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

Numerical results - Regularization property of REGINN

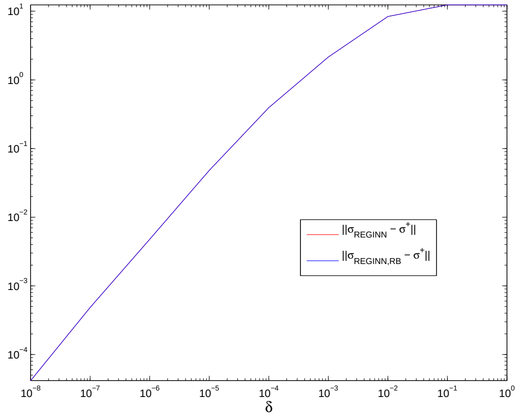


Figure: The regularization property $\|\sigma_{REGINN} - \sigma^{+}\| \to 0$ for $\delta \to 0$ with σ^{+} a random parameter and $\sigma_{start} = (1, 1, 1, 1, 1, 1, 1, 1, 1)$. Also holds for $\sigma_{REGINN,RB}$.

Numerical results - How good is RB?

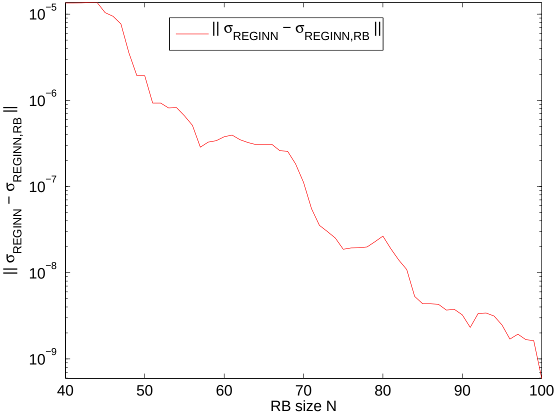


Figure: Comparison of Algorithm 1 and 2 with $\delta = 0.01$, depending on the size N of the used RB. Target parameter was $\sigma^{+} = (2, 2, 2, 2, 9, 2, 2, 2, 2)$, $\sigma_{start} = (1, 1, 1, 1, 1, 1, 1, 1, 1)$.

Numerical results - When does RB pay off?

Average time of REGINN without RB (Alg. 1): ~ 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 2 (using 200 random parameters with starting value $\sigma_{start} = (1, 1, 1, 1, 1, 1, 1, 1, 1)$ and $\delta = 0.01$).

Can we do better?

- Around 80% of the 2 s left are due to the Fréchet-derivative
- Fréchet-derivative in $\sigma \in L^\infty_+(\Omega)$ is given via

$$
\begin{aligned}
F'(\sigma)(\cdot) &: L^\infty_+(\Omega) \subset L^2(\Omega) \to H^1_0(\Omega) \subseteq L^2(\Omega) \\
F'(\sigma)\kappa &= v^\kappa_\sigma,
\end{aligned}
$$

  where v^κ_σ is a solution of

$$
\begin{aligned}
\nabla(\sigma \nabla v) &= -\nabla(\kappa \nabla u(\sigma)), \quad \text{in } \Omega \\
v &= 0, \quad \text{on } \partial\Omega,
\end{aligned}
$$

  with $u(\sigma)$ a solution of (2).
- Due to variable RHS a-priori RB-space cannot be constructed

Limitations of Algorithm 2

A larger dimension of p is desired!

- For large p (> 40) RB-space covering the whole variety in parameter cannot be constructed
- (in our case) computational time of Fréchet-derivative scales with p

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

$n := 0,\ \sigma_0^\delta := \sigma_{start},\ X_{N,1} := \mathrm{span}(\Phi_{N,1}),\ X_{N,2} := \mathrm{span}(\Phi_{N,2})$

while $\|F(\sigma_n^\delta) - u^\delta\| > \tau\delta$ **do**

  $\Phi_{N,1} := \Phi_{N,1} \cup F(\sigma_n^\delta),\ \Phi_{N,2} := \Phi_{N,2} \cup F'(\sigma_n^\delta)$

  $X_{N,1} = \mathrm{span}\{\Phi_{N,1}\},\ X_{N,2} = \mathrm{span}\{\Phi_{N,2}\}$

  $i := 1,\ \sigma_i^\delta := \sigma_n^\delta$

  repeat

    $\sigma_{i+1}^\delta := \sigma_i^\delta + F_N'(\sigma_i^\delta)^*(u^\delta - F_N(\sigma_i^\delta))$

    $i := i + 1$

  until $\|F_N(\sigma_i^\delta) - u^\delta\| > \tau\delta$

  $\sigma_{n+1}^\delta := \sigma_i^\delta$

  $n := n + 1$

end while

$\sigma_{final} := \sigma_n^\delta$

Realization of $F_N'(\sigma_i^\delta)^*(u^\delta - F_N(\sigma_i^\delta))$

- For $w_1,\ \sigma \in L_+^\infty(\Omega),\ w_2 \in H_0^1(\Omega)$ the following holds

$$\langle w_1, F'(\sigma)^* w_2\rangle = \langle F'(\sigma)w_1, w_2\rangle = \int_\Omega F'(\sigma)w_1\, w_2 dx$$
$$= \int_\Omega v_\sigma^{w_1}\, w_2 dx = \cdots = \int_\Omega w_1 \nabla u(\sigma)\cdot\nabla u_{w_2}^\sigma dx$$

where $u(\sigma)$ solves (2) and $u_\sigma^{w_2} \in H_0^1(\Omega)$ solves

$$\operatorname{div}(\sigma\nabla u) = w_2.$$

- Use $u_\sigma^{w_2}$ as snapshots for $X_{N,2}$
- Compute $F_N(\sigma_i^\delta)$ with $X_{N,1}$ and $F_N'(\sigma_i^\delta)^*(u^\delta - F_N(\sigma_i^\delta))$ via above scalarproduct-evaluation using $X_{N,2}$

First Results: comparing Algorithm 1 & 4

p	# DOFs	t Alg. 1 (s)	t Alg. 4 (s)	$\|\sigma_{Alg1} - \sigma_{Alg4}\|$
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 1 & 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!