Department of Applied Mathematics
and Theoretical Physics
Martin Benning

Inverse Problems

Example sheet **2** Presentation **13. November 2015**, 3 p.m.

Exercise 1 (Generalised inverse)

a) Let $m, n \in \mathbb{N}$ with $m \geq n \geq 2$. Compute the Moore-Penrose inverses of the following matrices:

   i. $K = (1, 1, \ldots, 1) \in \mathbb{R}^{1\times n}$
   ii. $K = \mathrm{diag}(a_1, \ldots, a_n) \in \mathbb{R}^{n\times n}$ with $a_j \in \mathbb{R}$ for $j \in \{1, \ldots, n\}$
   iii. $K \in \mathbb{R}^{m\times n}$ with $K^T K = I_n$

b) Let $a, b \in \mathbb{R}$ with $a < b$. Compute the Moore-Penrose inverses of the following operators:

   i. $K : L^2([a, b]) \to \mathbb{R}$ with $Ku = \int_a^b u(x)\, dx$
   ii. $K : L^2([a, b]) \to L^2([a, b])$ with $(Ku)(y) = u(y)g(y)$ for $0 \neq g \in C([a, b])$.

Exercise 2 (One-sided differential quotient operator)
We want to consider the problem of differentiation again, formulated as the inverse problem of finding u from $Ku = f$ with the integral operator $K : L^2([0, 1]) \to L^2([0, 1])$ defined as

$$(Ku)(y) := \int_0^y u(x)\, dx\,.$$

As an approximation to $K^\dagger$ we are interested in the study of the one-sided differential quotient operator $D_h : L^2([0, 1]) \to L^2([0, 1])$ with

$$(D_h f)(x) := \frac{1}{h}\begin{cases}(f(x+h) - f(x)) & x \in \left[0, \frac{1}{2}\right[\\ (f(x) - f(x-h)) & x \in \left[\frac{1}{2}, 1\right]\end{cases},$$

for $h \in]0, 1/2[$. Further, let $H^2([0, 1])$ denote the Hilbert space

$$H^2([0, 1]) = \left\{ f \in L^2([0, 1]) \mid f'', f' \in L^2([0, 1]) \right\}\,.$$

We consider the case of a noisy measurement, i.e. we observe $f^\delta \in L^2([0, 1])$ for which

$$\|f - f^\delta\|_{L^2([0,1])} \leq \delta$$

holds true, for the exact data $f \in \mathcal{D}(K^\dagger)$.

Please turn over!

a) Assume that $f \in H^2([0,1])$ and $\|f''\|_{L^2([0,1])} \leq c$. Verify the following estimate for the overall L^2-error between $u^\dagger$ and $D_h f^\delta$:

$$\|K^\dagger f - D_h f^\delta\|_{L^2([0,1])} \leq \frac{\sqrt{5}\delta}{h} + \frac{ch}{\sqrt{2}} \tag{1}$$

b) Compute $h(\delta)$ such that the right-hand-side of (1) gets minimal.

c) Discretise D_h by evaluating D_h at n discrete points $x_k := (k-1)h$, $k \in \{1, 2, \dots, n\}$, for $h = \frac{1}{n-1}$. This way we obtain a mapping $\tilde{D}_h : \mathbb{R}^n \to \mathbb{R}^n$. Implement a MATLAB-function `diffquot` that takes a vector $\tilde{f} = (f(x_1), f(x_2), \dots, f(x_n))^T \in \mathbb{R}^n$ as an input argument and returns the output $\tilde{D}_h \tilde{f}$.

d) Test your function for $h = 2^{-k}$, $k \in \{2, 3, \dots, 8\}$ and

   i. $f(x) = \cos(\pi x)$ for $x \in [0, 1]$;

   ii. $f(x) = \begin{cases} 0 & x \in \left[0, \frac{1}{3}\right[\\ x - \frac{1}{3} & x \in \left[\frac{1}{3}, \frac{2}{3}\right[\\ \frac{1}{3} & x \in \left[\frac{2}{3}, 1\right] \end{cases}$,

   and plot the maximum error $\|\tilde{D}_h \tilde{f} - (f'(x_1), f'(x_2), \dots, f'(x_n))^T\|_\infty$ dependent on h.

Exercise 3 (Singular value decomposition)
Prove Theorem 2.5 (the existence of a singular value decomposition of compact, linear operators) as stated in the lecture notes.
Hint: Consider the compact operator K^*K at first, and make use of $\overline{\mathcal{R}(K^*K)} = \overline{\mathcal{R}(K^*)}$ later on.

Exercise 4 (The Radon transform)

a) The MATLAB command `f = radon(u, phi);` computes a discretised two-dimensional radon transform of a discrete image `u` for a vector of angles `phi`. Use this command to set up a matrix `R` that maps the column-vector representation of `u` into the column-vector representation of the sinogram `f` for an arbitrary image `u` $\in \mathbb{R}^{64 \times 64}_{\geq 0}$ and angles `phi` with `phi(j) = j` for $j \in \{0, 2, \dots, 178\}$.

b) Create a noisy sinogram by applying `R` to a down-sampled version of the Shepp-Logan phantom (built-in in MATLAB; use the command `phantom`) and subsequently adding non-negative, random numbers (for example normally distributed via `randn`) to the sinogram. Create multiple versions with different noise levels. In order to down-sample you can use the command `imresize`.

c) Compute a singular value decomposition of `R` via the MATLAB-command `svd` and plot selected singular vectors of your choice.

d) Create a 'pseudo'-inverse of `R` by constructing an appropriate matrix with inverted singular values and apply this matrix to the column-vector representations of your noisy sinograms. What do you observe if you truncate some of the small singular values?