Department of Applied Mathematics
and Theoretical Physics
Hanne Kekkonen & Yury Korolev

Inverse Problems

Example sheet **1** Presentation **28 October 2019**, **2-3:30pm**, **MR15**.
Please submit after the lecture on **24 October 2019**.

Exercise 1 (Integral operators - submit)
For $\Omega = [0,1]^2$ and $\mathcal{X} = L^2(\Omega)$, we consider the integral operator $A : \mathcal{X} \to \mathcal{X}$ with

$$(Au)(y) := \int_\Omega k(x,y)u(x)\,dx,$$

for $k \in L^2(\Omega \times \Omega)$. Show that

(a) A is linear with respect to u,

(b) A is a bounded linear operator, i.e. $\|Au\|_\mathcal{X} \le \|A\|_{\mathcal{L}(\mathcal{X},\mathcal{X})}\|u\|_\mathcal{X}$. Give also an estimate for $\|A\|_{\mathcal{L}(\mathcal{X},\mathcal{X})}$,

(c) the adjoint A^* is given via

$$(A^*v)(y) = \int_\Omega k(y,x)v(x)\,dx.$$

Exercise 2 (Inverse problem of differentiation - submit)
We consider the problem of differentiation, formulated as the inverse problem of finding u from $Au = f$ with the integral operator $A : L^2([0,1]) \to L^2([0,1])$ defined as

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

(a) Let f be given by

$$f(x) := \begin{cases} 0 & x < \frac{1}{2}, \\ 1 & x > \frac{1}{2}. \end{cases}$$

Show that $f \in \overline{\mathcal{R}(A)}$.

(b) Let f be given as in a). Show that $f \in \overline{\mathcal{R}(A)} \setminus \mathcal{R}(A)$.
Hint: Consider the Picard criterion.

(c) Prove or falsify: “The Moore-Penrose inverse of A continuous.”

Please turn over!

Exercise 3 (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) $A = (1, 1, \ldots, 1) \in \mathbb{R}^{1\times n}$

(ii) $A = \operatorname{diag}(a_1, \ldots, a_n) \in \mathbb{R}^{n\times n}$ with $a_j \in \mathbb{R}$ for $j \in \{1, \ldots, n\}$

(iii) $A \in \mathbb{R}^{m\times n}$ with $A^T A = I_n$

(b) Let $a, b \in \mathbb{R}$ with $a < b$. Compute the Moore-Penrose inverse of the operator $A : L^2([a, b]) \to \mathbb{R}$ with

$$Au = \int_a^b u(x)\, dx.$$

Exercise 4 (Convolution)

Many forward problems are either modelled as convolutions or they are modelled as the composition of several components one of which is a convolution. Therefore convolutions play an important role in inverse problems. As in Exercise 1, let $\Omega = [0, 1]^2$ be the unit square and let $\mathcal{X} = L^2(\Omega)$. A convolution is the special case of an integral operator $A : \mathcal{X} \to \mathcal{X}$ where the kernel has a simple structure:

$$(Au)(y) := \int_\Omega k(y - x)u(x)\, dx,$$

for $k \in L^2(\Omega)$. It follows easily from Exercise 1 that A is linear and bounded.

(a) Although shown in general in Exercise 1, give an explicit form of the adjoint of the convolution.

(b) Let $f = Au$. It follows from the convolution theorem that a convolution can be inverted by means of the Fourier transform

$$u = (2\pi)^{-\frac{n}{2}} \mathcal{F}^{-1}\left(\frac{\mathcal{F}(f)}{\mathcal{F}(k)}\right), \tag{1}$$

where $\mathcal{F}$ is the Fourier transform and $\mathcal{F}^{-1}$ its inverse. Implement this formula in MATLAB to deblur (deconvolve) the blurry tree image f generated by the script `ex4b_generate_data.m`, which is provided on the course web page. Note that the script also outputs $\mathcal{F}(k)$. Add some noise to the data and show that the inversion formula is ill-conditioned.
Hint: Make use of the MATLAB commands `fft2` and `ifft2`.

(c) Reformulate equation (1) so that the denominator is non-negative and give a stable approximation of this formula. Implement this formula in MATLAB and empirically show that it is stable.
Hint: Make use of the MATLAB command `conj`.

Exercise 5 (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 $j \in \{0, 2, \ldots, 178\}$.

(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 to the sinogram. Create multiple versions with different noise levels.

(c) Compute a singular value decomposition of `R` via the Matlab command `svd` and visualise 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. Regularise the Moore-Penrose inverse using

   (i) Truncated singular value decomposition;

   (ii) Tikhonov regularisation.