Exercise 1 (Right-shift operator)
The right-shift operator $K : \ell^2 \to \ell^2$, $\{u_j\}_{j \in \mathbb{N}} \mapsto \{f_j\}_{j \in \mathbb{N}}$, is given by

$$f_j = (Ku)_j = \begin{cases} 0 & j = 1 \\ u_{j-1} & j \geq 2 \end{cases}.$$  

(a) Compute the range and kernel of $K$, i.e. $\mathcal{R}(K), \mathcal{N}(K)$.

(b) Prove or falsify: “The Moore–Penrose inverse of $K$ continuous.” Argue only with the definition of the operator and your results of (a).

(c) Compute the Moore–Penrose inverse of $K$. It is necessary to also state the domain and the range of $K^\dagger$.

Exercise 2 (Inverse problem of differentiation)
We consider the problem of differentiation, 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

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

(a) Let $f$ be given by $f(x) := \begin{cases} 2x & x < \frac{1}{2} \\ 2x - 1 & x \geq \frac{1}{2} \end{cases}$. Show that $f \in \overline{\mathcal{R}(K)}$.

(b) Let $f$ be given as in Exercise a). Show that $f \in \overline{\mathcal{R}(K)} \setminus \mathcal{R}(K)$

(c) Prove or falsify: “The Moore–Penrose inverse of $K$ continuous.”

Exercise 3 (Differential quotient operator)
As in Exercise (b), we consider the inverse problem of differentiation. As an approximation to $K^\dagger$ we are interested in studying the following differential quotient operator $R_\alpha : L^2([0,1]) \to L^2([0,1])$ with

$$(R_\alpha f)(x) := \frac{1}{\alpha} \begin{cases} f(x + \alpha) - f(x) & x \in [0, \frac{1-\alpha}{2}] \\ f(x + \frac{\alpha}{2}) - f(x - \frac{\alpha}{2}) & x \in [\frac{1-\alpha}{2}, \frac{1+\alpha}{2}] \\ f(x) - f(x - \alpha) & x \in [\frac{1+\alpha}{2}, 1] \end{cases}.$$  

Please turn over!
for \( \alpha \in ]0, 1/2[ \). Further, let \( H^2([0, 1]) \) denote the Hilbert space
\[
H^2([0, 1]) = \{ f \in L^2([0, 1]) \mid f'', f' \in L^2([0, 1]) \}.
\]
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 \( f \in \mathcal{D}(K^\dagger) \).

(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 \( R^\alpha f^\delta \):
\[
\| K^\dagger f - R^\alpha f^\delta \|_{L^2([0, 1])} \leq \frac{\sqrt{6}}{\alpha} \delta + \frac{\sqrt{17}}{4} \alpha c
\]  

(b) Show that \( R^\alpha : L^2([0, 1]) \to L^2([0, 1]) \) is a convergent regularisation method and determine a corresponding a-priori parameter choice rule.

(c) Discretise \( R^\alpha \) by evaluating \( R^\alpha \) at \( 2n \) discrete points \( x_k := (k - 1)\frac{a}{2}, \ k \in \{1, \ldots, 2n\}, \) for \( \alpha = \frac{1}{n-1} \) and \( n \in \mathbb{N} \setminus \{1\} \). This way we obtain a mapping \( \tilde{R}^\alpha : \mathbb{R}^{2n} \to \mathbb{R}^{2n} \). Implement a MATLAB\textsuperscript{©}-function \texttt{diffquot} that takes a vector \( \tilde{R}^\alpha f \) as an input argument and returns the output \( \tilde{R}^\alpha f \).

(d) Test your function for \( \alpha = 2^{-k}, \ k \in \{2, 4, \ldots, 8\} \) and

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

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

and plot the maximum error \( \| \tilde{R}^\alpha f - (f'(x_1), f'(x_2), \ldots, f'(x_{2n}))^T \|_\infty \) dependent on \( \alpha \).

---

Exercise 4 (Deconvolution)

Let \( \Omega := [0, 1]^2 \), \( k \in L^2(\Omega) \) and \( \overline{k} \in L^2(\mathbb{R}^2) \) be the extension of \( k \) with
\[
\overline{k}(z) = \begin{cases} 
 k(z) & z \in \Omega \\
 0 & z \in \mathbb{R}^2 \setminus \Omega 
\end{cases}
\]
and consider the convolution operator \( K : L^2(\Omega) \to L^2(\Omega) \) with
\[
(Ku)(x) := \int_\Omega \overline{k}(x-y)u(y) \, dy.
\]

(a) Compute the singular value decomposition of \( K \).

**Hint:** you can represent a function \( v \in L^2(\Omega) \) as \( v = \sum_{m,n \in \mathbb{Z}} (v, \varphi_{m,n}) \varphi_{m,n} \) with \( \varphi_{m,n}(x_1, x_2) = \exp(-i2\pi(mx_1 + nx_2)) \).
(b) Argue with the singular values whether the inverse problem is ill-posed or not, for the specific choices

(i) \[ k(x_1, x_2) = \frac{1}{h^2} \chi_{[-\frac{1}{2}, \frac{1}{2}]} \left( x_1 - \frac{1}{2} \right) \chi_{[-\frac{1}{2}, \frac{1}{2}]} \left( x_2 - \frac{1}{2} \right) \] for \( 0 < h < 1 \).

(ii) \[ k(x_1, x_2) = \varphi(x_1)\varphi(x_2) \] with \( \varphi(x) := \begin{cases} \exp \left( -\frac{1}{1/4-(x-1/2)^2} \right) & x \in ]0, 1[ \\ 0 & \text{else} \end{cases} \).

Is the ill-posedness mild or severe?

c) Implement the deconvolution as in Exercise 4 of Example Sheet 1. Regularise the problem using

(i) Truncated singular value decomposition;

(ii) Tikhonov regularisation.

How does the latter relate to Exercise 4 c) on Example Sheet 1?

Exercise 5 (The Radon transform)

(a) The MATLAB© command \( f = \text{radon}(u, \phi) \) 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 \texttt{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 \texttt{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.