Exercise 1 (4 points). Let \( A \in \mathbb{R}^{m \times n} \), and let \( A = V S U^* \) be the singular value decomposition of \( A \) with sorted singular values \( \sigma_i \), i.e. \( \sigma_i \geq \sigma_{i+1} \), \( S = \text{diag}(\sigma_i) \). Let \( r = \text{rank}(A) \) and define the truncated singular value decomposition as
\[
A_\nu := \sum_{j=1}^{\nu} \sigma_j v_j^* u_j^*
\]
for \( 0 \leq \nu \leq r \). Show that \( A_\nu \) yields the best rank \( \nu \) approximation in the \( \ell^2 \) sense, i.e.
\[
\|A - A_\nu\| = \inf_{B \in \mathbb{R}^{m \times n}, \text{rank}(B) \leq \nu} \|A - B\|
\]
where the norm is the usual norm induced by the \( \ell^2 \) norm on \( \mathbb{R}^m \).
(Hint: First show what \( \|A - A_\nu\| \) is. Then: What is the dimension of \( \text{ker}(B) \)?)

Exercise 2 (4 points). Let \( A \in L(X,Y) \) be compact, \( \text{dim}(\text{range}(A)) = \infty \), and let \( Ax = \sum_{n=1}^{\infty} \sigma_n \langle u_n, x \rangle v_n \) be the singular value decomposition of \( A \). For \( \alpha > 0 \) define \( R_\alpha : Y \to X \) by
\[
R_\alpha y = \sum_{n=1}^{\infty} \frac{1}{\sigma_n + \alpha} \langle y, v_n \rangle u_n.
\]
Show that \( \{R_\alpha\}_{\alpha} \) is a regularization of \( A^\dagger \).

Exercise 3 (4 points). Consider the Landweber-iteration of sheet 2, but now for a general compact \( A \in L(X,Y) \) with infinite dimensional range. We define \( X \ni x_0 = 0 \) and
\[
x^{k+1} = x^k + \tau A^* (y - Ax^k),
\]
for \( 0 < \tau < \frac{2}{\|A\|^2} \). We define the family of operators \( \{R_k\}_{k \in \mathbb{N}}, R_k : Y \to X \) as \( R_k y := x^k \).

- Let \( Ax = \sum_{n=1}^{\infty} \sigma_n \langle u_n, x \rangle v_n \) be the singular value decomposition of \( A \). Determine a function \( g_k(\sigma) \) such that
\[
R_k y = \sum_{n=1}^{\infty} g_k(\sigma_n) \langle v_n, y \rangle u_n.
\]
- Prove that \( g_k(\sigma) \) is bounded and that it converges pointwise to \( \frac{1}{\sigma} \) for \( k \to \infty \).

Please turn!
Exercise 4 (4 points). Implement the following numerical experiment. Load an image $\hat{u}$, create a blur kernel $h$, and compute the Fourier transforms $\mathcal{F}(\hat{u})$ and $\mathcal{F}(h)$ (help fft2).

- Do a pointwise multiplication of the two transformed functions, and invert the Fourier transform to obtain $f = \mathcal{F}^{-1}(\mathcal{F}(\hat{u})\mathcal{F}(h))$.

- Add some small noise to $f$ and visualize the naive inversion

$$u = \mathcal{F}^{-1}\left(\mathcal{F}(f)\right).$$

- Now compute $\frac{\mathcal{F}(f)}{\mathcal{F}(h)}$ and truncate high frequencies. The easiest way to do this in Matlab is to change the representation of the Fourier coefficients using fftshift, and only keep the coefficients within a radius of $r$ around the center. Invert the Fourier transform of the truncated coefficients and visualize the result.

- What happens for different truncation radii $r$?