Prof. Dr. D. Kressner
A. Cortinovis

Problem 1: Lemma from slide 12

Prove that for a Gaussian matrix Ω and fixed matrices A, B of suitable size it holds that
$$\mathbb{E}\|A\Omega B\|_F^2 = \|A\|_F^2 \cdot \|B\|_F^2.$$

Problem 2: Implementation of low-rank approximation by randomized algorithm

Implement the randomized algorithm from slide 7 in MATLAB.

1. Calculate the overall complexity of the algorithm applied to $m \times n$ matrix A, with respect to $k = r + p$.
2. If A is symmetric, then the columns of Q form a good basis for both the column space and the row space of A so that we have $A \approx QQ^T AQQ^T$. If we know $\|A - QQ^T A\|_2 \leq \varepsilon$, find the bound for $\|A - QQ^T AQQ^T\|_2$.
3. Using Point 2, adjust the randomized algorithm to give a spectral decomposition of a symmetric matrix A.

Problem 3: Lemma from slide 20

For an orthogonal projector P, a nonnegative diagonal matrix $D = \mathrm{diag}(d_1, \dots, d_n)$ and $t \geq 1$ it holds that
$$\|PDP\|_2^t \leq \|PD^t P\|_2. \tag{1}$$
Using this fact, prove that for any matrix A and $q > 0$ it holds that
$$\|PA\|_2 \leq \|P(AA^T)^q A\|_2^{1/(2q+1)}.$$

Hints for proving (1)*:* Consider a unit vector $x = (x_1, \dots, x_n)^T$ such that $x^T PDPx = \|PDP\|_2$, prove that $Px = x$, which implies
$$(x^T PDPx)^t = \left(\sum_{i=1}^n x_i^2 d_i\right)^t;$$
then apply Jensen's inequality to the convex function $y \mapsto |y|^t$.

Problem 4: Norm estimation with Gaussian vectors

Prove that for a matrix A and a Gaussian vector $\omega \sim N(0, I)$ it holds that
$$\mathbb{P}\left(\|A\omega\|_2 \leq \frac{1}{10}\sqrt{\frac{\pi}{2}}\|A\|_2\right) \leq \frac{1}{10}$$
and show that this implies the lemma in the last slide of the lecture.

Hint: As intermediate steps, prove that for a standard normal variable $X \sim N(0, 1)$, for all $\mu > 0$ we have
$$\mathbb{P}\left(\|C\omega\|_2 \leq \mu\|C\|_2\right) \leq \mathbb{P}\left(|X| \leq \mu\right) \leq \sqrt{\frac{2}{\pi}}\mu.$$

Problem 5: Recompression of a structured matrix by randomized algorithm

In MATLAB, for $n = 100$, construct $n \times n$ matrices A and B by evaluating the functions
$$f_A(x_i, x_j) = \frac{1}{x_i + x_j} \text{ and } f_B(x_i, x_j) = \frac{1}{\sqrt{x_i + x_j}},$$
respectively, for $x_i = \frac{i}{n}$, $i = 1, 2, \dots, n$. Let $A = U_A \Sigma_A V_A^T$ and $B = U_B \Sigma_B V_B^T$ be their SVDs with Σ_A square of order $r_A = \mathrm{rank}(A)$, and Σ_B square of order $r_B = \mathrm{rank}(B)$. Recall from the last week's exercise that $C = A + B$ can be represented as
$$C = \begin{bmatrix} U_A & U_B \end{bmatrix} \begin{bmatrix} \Sigma_A & \\ & \Sigma_B \end{bmatrix} \begin{bmatrix} V_A^T \\ V_B^T \end{bmatrix}.$$
Adjust the randomized algorithm to recompress C to rank $r = 17$ by exploiting the given representation.