Solution 10 – MATH-250 Advanced Numerical Analysis I

The exercise sheet is divided into two sections: quiz and exercises. The quiz will be discussed in the beginning of the lecture on Thursday, May 15. The exercises marked with ($\star$) are graded homework. The exercises marked with **(Python)** are implementation based and can be solved in the Jupyter notebooks which are available on Moodle/Noto. **The deadline for submitting your solutions to the homework is Friday, May 16 at 10h15.**

Quiz

(a) Let $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and $\mathbf{b} \in \mathbb{R}^m$. If A has rank smaller than n then $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2$ has infinitely many solutions.

☐ True ■ False

(b) Let $R \in \mathbb{R}^{n \times n}$ be upper triangular then

$$\|R^{-1}\|_2 \leq n \cdot \max\{|r_{11}|^{-1}, \dots, |r_{nn}|^{-1}\}.$$

In particular, this means that $\|R^{-1}\|_2$ can only be large when R has small diagonal entries.

☐ True ■ False

(c) Consider the problem $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_1$ with $A \in \mathbb{R}^{m \times n}$, $m \geq n$, and $\mathbf{b} \in \mathbb{R}^m$. Then there is always a minimizer $\mathbf{x}^\star$ such that the residual $A\mathbf{x}^\star - \mathbf{b}$ has at least one zero entry.

■ True ☐ False

Solution.

(a) We assume that there exists a minimizer $\mathbf{x} \in \mathbb{R}^m$. This is true if the rank of A is larger than zero because then range$(A) \neq \emptyset$ and we can simply choose $\mathbf{x}$ such that $A\mathbf{x}$ is the projection of $\mathbf{b}$ onto range(A). Then, for any $\mathbf{y} \in$ kern(A) it holds that $A(\mathbf{x} + \mathbf{y}) = A\mathbf{x} + A\mathbf{y} = A\mathbf{x}$ which is hence also a minimizer. The kernel of A has a dimension of at least one because A is not full rank, wherefore there exist infinitely many solutions.

(b) We define R such that $r_{ii} = 1$ for all $i = 1, 2, \dots, n$ and $r_{ij} = -1$ if $j > i$. We now apply Gaussian eliminiation to see that the inverse S of R has the entries $s_{ii} = 1, i = 1, 2, \dots, n$ and $s_{ij} = 2^{j-i-1}, j > i$. We now compute $\|S\|_1$. We know that the largest sum of entries of S is contained in the first row, namely $1 + \sum_{i=0}^{n-2} 2^i$. Applying the usual sum of powers of two formula we get that the sum of the elements of the first row of S is equal to 2^{n-1}. Lastly, $\|A\|_2 \leq \sqrt{n}\|A\|_2$ means that the inequality from the claim cannot be true for $n \geq 4$.

(c) We rewrite the problem $\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_1$ into the combined linear program

$$\min_{\mathbf{v}=[\mathbf{x}^\top \mathbf{z}^\top]^\top \in \mathbb{R}^{2n}} \sum_{i=1}^{n} z_i \quad \text{s.t.} \quad -\begin{bmatrix} 0 & I \end{bmatrix} \mathbf{v} \leq 0, \quad \begin{bmatrix} A & -I \\ -A & -I \end{bmatrix} \mathbf{v} \leq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix},$$

where we understand the vector inequalities to hold for each component. This is means we need to minimize a concave function on a convex constraint set, implying that the minimizer lies on the boundary of the constraint set. In turn, this requires that at least one of the inequality constraints is fulfilles with equality because the optimization takes place on a convex simplex. If this is the case for one of the inequalities in $\begin{bmatrix} A & -I \\ -A & -I \end{bmatrix} \mathbf{v} \leq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix}$, then it holds that the residual $A\mathbf{x} - \mathbf{b}$ in that coordinate is zero and hence $z_i = (A\mathbf{x} - \mathbf{b})_i = 0$. On the other hand, if the equality is found in the constraint $-\begin{bmatrix} 0 & I \end{bmatrix} \mathbf{v} \leq 0$, then this means that one of the z_i is zero and hence $(A\mathbf{x} - \mathbf{b})_i = 0$. This concludes the proof.

Exercises

Problem 1. (Python)

(a) Write a function `gradient` in Python that solves $A\mathbf{x} = \mathbf{b}$ using the Gradient method. Your function should take as inputs

- The symmetric positive definite matrix $A \in \mathbb{R}^{n \times n}$
- The right hand side $\mathbf{b} \in \mathbb{R}^n$
- The tolerance *rtol*

The function should output

- A vector $\hat{\mathbf{x}} \in \mathbb{R}^n$ such that $\frac{\|A\hat{\mathbf{x}} - \mathbf{b}\|_2}{\|\mathbf{b}\|_2} < rtol$.
- The number of iterations required to achieve a relative error smaller than *tol*.
- A vector consisting of the norms of the residuals at each iteration $\|\mathbf{r}^{(k)}\|_2$.

(b) Apply your function to the two linear systems $A\mathbf{x} = \mathbf{b}$ where

(1)

$$A_1 = \begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}, \quad \mathbf{b}_1 = \begin{pmatrix} 2 \\ -8 \end{pmatrix}$$

(2) $A_2 \in \mathbb{R}^{1024 \times 1024}$ with right hand side $\mathbf{b}_2 \in \mathbb{R}^{1024}$ generated by the following code

```
import numpy as np
import scipy as sp

n = 32
a = sp.sparse.diags([-1, 2, -1], [-1, 0, 1], shape=(n, n))
I = sp.sparse.eye(n)
A_2 = sp.sparse.kron(I, a) + sp.sparse.kron(a, I)

```

```
def f(x, y):
    return -(12 * x ** 2 - 6 * x) * y * (y - 1) - 2 * x ** 3 * (x - 1)

t = np.tile(np.arange(1, n + 1) / (n + 1), (n, 1))
x = t.T.flatten()
y = t.flatten()
b_2 = f(x, y) / ((n + 1) ** 2)
```

For $\mathsf{tol} = 10^{-8}$ plot the norm of the residuals $\|\mathbf{r}^{(k)}\|_2$ versus k. Compare your method with the Conjugate Gradient method. You may use the built-in conjugate gradient method `scipy.sparse.linalg.cg`/`sp.sparse.linalg.cg` in Python.[1]

Solution. Available in the Jupyter notebook `serie10-sol.ipynb` on Moodle.

Problem 2.

Prove Lemma 6.2 in the lecture notes. That is, show that if $A \in \mathbb{R}^{m\times n}$ has rank n then $A^\top A$ is symmetric positive definite.

Solution. Clearly $A^\top A$ is symmetric since $A^\top A = (A^\top A)^\top$. It is positive definite since for all $\mathbf{x} \neq 0$ we have $\mathbf{x}^\top A^\top A\mathbf{x} = \|A\mathbf{x}\|_2^2 > 0$.

Problem 3. Assume that you are given data $t_1, \ldots, t_m \in \mathbb{R}$ and $y_1, \ldots, y_m \in \mathbb{R}$. Suppose that x_1 and x_2 are chosen such that

$$\sum_{i=1}^{m} (x_1 + x_2 t_i - y_i)^2$$

is minimized. Further, define $\hat{y}_i = x_1 + x_2 t_i$ and $r_i = y_i - \hat{y}_i$ for $i = 1, \cdots, m$. Show that

(a) $\sum_{i=1}^{m} r_i = 0$

(b) Let $\bar{t} = \frac{1}{m}\sum_{i=1}^{m} t_i$ and $\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$. Then $\bar{y} = x_1 + x_2\bar{t}$.

(c) $\sum_{i=1}^{m} t_i r_i = 0$

(d) $\sum_{i=1}^{m} \hat{y}_i r_i = 0$

Solution.

(a) Define $f(x_1, x_2) := \sum_{i=1}^{m} (x_1 + x_2 t_i - y_i)^2$. Differentiating with respect to x_1 gives

$$\frac{\partial f}{\partial x_1} = 2\sum_{i=1}^{m}(x_1 + x_2 t_i - y_i) = 2\sum_{i=1}^{m} r_i$$

Since x_1, x_2 are optimal we must have $\frac{\partial f}{\partial x_1} = 0$ and hence $\sum_{i=1}^{m} r_i = 0$.

[1] Note that the function `scipy.sparse.linalg.cg` in Python does not return the norm of the residual for each iteration. An ugly solution is to call the function multiple times in a loop, increasing the maximum number of iterations until convergence, and computing by hand the residual norm each time.

(b) This follows immediately from (a). We have

$$\sum_{i=1}^{m} r_i = 0 \Leftrightarrow \sum_{i=1}^{m} y_i = mx_1 + x_2 \sum_{i=1}^{m} t_i$$

Dividing by m yields the required result.

(c) With $f(x_1, x_2)$ defined as in (a). Since x_1, x_2 are optimal we must have

$$\frac{\partial f}{\partial x_2} = 2 \sum_{i=1}^{m} t_i r_i = 0$$

which gives the required result.

(d) Inserting the definition of $\hat{y}_i$ gives

$$\sum_{i=1}^{m} \hat{y}_i r_i = x_1 \sum_{i=1}^{n} r_i + x_2 \sum_{i=1}^{m} t_i r_i = 0 + 0 = 0$$