Norwegian University of Science
and Technology
Department of Mathematical
Sciences

TMA4180
Optimization I
Spring 2019

Exercise set 4

The second and fourth exercise below are concerned with the usage of the CG-method for the solution of linear least squares problems (which is one approach to the solution of overdetermined linear systems). You can find some complementary information on this topic in Nocedal & Wright, Chapter 10.2. (Later in the course, we will discuss *nonlinear* least squares problems in more detail.)

1 Let

$$A := \begin{pmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 2 \end{pmatrix} \qquad \text{and} \qquad b := \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

Use the CG-method with initialisation $x_0 = 0$ for solving the linear system $Ax = b$.

2 Assume that $A \in \mathbb{R}^{m \times n}$ is a matrix and that $b \in \mathbb{R}^m$.

a) Show that $x^* \in \mathbb{R}^n$ solves the *least squares problem*

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|^2, \tag{1}$$

if and only if x^* satisfies the *normal equations*

$$A^T A x^* = A^T b.$$

b) Show that the optimization problem (1) admits a solution $x^* \in \mathbb{R}^n$.

c) Show that the solution x^* of (1) is unique, if the rank of A equals n.

d) Show that, regardless of the rank of A, the optimization problem

$$\min_{x \in \mathbb{R}^n} \|x\|^2 \qquad \text{s.t. } x \text{ solves (1)} \tag{2}$$

admits a unique solution $x^\dagger \in \mathbb{R}^n$.

3 Assume that $A \in \mathbb{R}^{n \times n}$ is symmetric and positive *semi*-definite and $b \in \operatorname{Range} A$. Show that (in exact arithmetics) the CG algorithm converges for every starting point $x_0 \in \mathbb{R}^n$ in at most $m = \dim(\operatorname{Range} A)$ iterations to a solution of $Ax = b$.

(This shows that at least theoretically the assumption of positive definiteness can be slightly relaxed.)

4 Assume that $m > n$, that $A \in \mathbb{R}^{m\times n}$, and that $b \in \mathbb{R}^m$. Consider the following algorithm:

- Choose $x_0 \in \mathbb{R}^n$ arbitrary, set $r_0 \leftarrow Ax_0 - b$, $s_0 \leftarrow A^T r_0$, $p_0 \leftarrow -s_0$, and $k \leftarrow 0$.
- While $s_k \neq 0$:

$$\begin{aligned}
\alpha_k &\leftarrow \frac{\|s_k\|^2}{\|Ap_k\|^2},\\
x_{k+1} &\leftarrow x_k + \alpha_k p_k,\\
r_{k+1} &\leftarrow r_k + \alpha_k A p_k,\\
s_{k+1} &\leftarrow A^T r_{k+1},\\
\beta_{k+1} &\leftarrow \frac{\|s_{k+1}\|^2}{\|s_k\|^2},\\
p_{k+1} &\leftarrow -s_{k+1} + \beta_{k+1} p_k,\\
k &\leftarrow k + 1.
\end{aligned}$$

Assume that the matrix A has full rank. Show that the algorithm above is actually identical with the CG-algorithm for the solution of $A^TAx = A^Tb$ (in the sense that the iterates x_k of both methods coincide).

5 (Cf. Exercise 5.1 in Nocedal & Wright)

Implement the CG method for the solution of linear systems $Ax = b$ with symmetric and positive definite matrix $A \in \mathbb{R}^{n\times n}$.

Use your method in the case where A is the Hilbert matrix, the elements of which are

$$A_{i,j} = \frac{1}{i+j-1}, \qquad 1 \le i, j \le n.$$

Use the right hand side $b = (1, 1, \ldots, 1)^T$ and the initialisation $x_0 = 0$. Test your code for dimensions $n = 5,\ 8,\ 12,\ 20$. How many iterations are required to reduce the residual below 10^{-6}? Why do your results not contradict the theoretical results concerning the CG method that were discussed in the lecture?

Hint: You might want to have a look at the condition number of the Hilbert matrix.

(Note that in MATLAB the Hilbert matrix can be produced with the command `hilb`, and in Python using `scipy.linalg.hilbert`.)