Norwegian University of Science and Technology
Department of Mathematical Sciences

Exercise set 4

The second and third 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 over-determined linear systems). You can find some complementary information on this topic in Nocedal & Wright, Chapter 10.2. (Later in the course, we will discuss Chapter 10.3 on *nonlinear* least squares problems.)

In Exercise 4, it is easily possible that the results you obtain appear to contradict the theory developed in the course. The reason for this behaviour of CG is explained by the condition number of the Hilbert matrix.

1 Let

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

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*

$$A^{\mathrm{T}} A x^* = A^{\mathrm{T}} b.$$

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, $b \in \operatorname{ran} A$ (equivalently, $b \perp \ker A$, or equivalently there exists a solution to the system $Ax = b$). Show that, in exact arithmetics, the CG algorithm converges in at most $m = \dim \operatorname{ran} A$ iterations to a solution to the system $Ax = b$ from any starting point $x_0 \in \mathbb{R}^n$.

Thus the requirement for A to be positive definite can be somewhat relaxed, and the algorithm still works.

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 $r_0 \leftarrow Ax_0 - b$, $s_0 \leftarrow A^{\mathrm{T}} r_0$, $p_0 \leftarrow -s_0$, and $k \leftarrow 0$.
- While $s_k \neq 0$:

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

Assume that the matrix A has full rank. Show that the algorithm above is actually identical with the CG-algorithm for the solution of $A^{\mathrm{T}}Ax = A^{\mathrm{T}}b$ (in the sense that the iterates x_k of both methods coincide).

5 Exercise 5.1 in Nocedal & Wright.

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

6 Exercise 5.12 in Nocedal & Wright: show that Lemma 5.6 holds for any choice of β_k in the non-linear CG algorithm with $|\beta_k| \leq |\beta_k^{\mathrm{FR}}|$. In particular, this explains the strategy (5.48) in the book (FR–PR CG algorithm).