

Exercise #4

February 7, 2023

Problem 1.

Implement Algorithm 5.2 (N&W Book, Page Number 112) and use it to solve linear systems in which A is the Hilbert matrix, whose elements are $A_{i,j} = \frac{1}{i+j-1}$. Set the right-hand-side to $b = (1, 1, \dots, 1)^T$ and the initial point to $x_0 = 0$. Try dimensions $n = 5, 8, 12, 20$ and report the number of iterations required to reduce the residual below 10^{-6} .

Solution.

See the file `CG_Hilbert.py` on the wiki-page.

The number of iterations needed are as follows:

n	5	8	12	20
iterations	6	19	38	70

That is, the CG method always requires more iterations than predicted by the theory. The reason for this discrepancy is the fact that the Hilbert matrix is extremely ill-conditioned (and because of that often used for numerical tests). For $n = 5$ for instance, the `linalg` package in `numpy` computes the condition of the Hilbert matrix as approximately 476607; for $n = 20$, it computes it at approximately 10^{18} . (Because the matrix is that ill-conditioned, these values should only be taken as rough approximations to the true condition numbers; the ill-conditioning of the matrix makes it also difficult to compute its condition number.)

Problem 2.

Implement Algorithm 5.4 (N&W Book, Fletcher-Reeves Method) for the *Rosenbrock* function $f: \mathbb{R}^2 \mapsto \mathbb{R}$, $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, where the step length α_k should satisfy the strong Wolfe conditions (5.43) (N&W Book, Page Number 122) with $0 < c_2 < \frac{1}{2}$.

Hint: You can use a bracketing method as discussed in the lecture on January 23 for finding a suitable step length. In this case, you have to recall, though, that the two conditions

$$\begin{aligned} f(x_k + \alpha p_k) &\leq f(x_k) + c_1 \alpha \langle \nabla f(x_k), p_k \rangle, \\ \langle \nabla f(x_k + \alpha p_k), p_k \rangle &\leq -c_2 \langle \nabla f(x_k), p_k \rangle, \end{aligned}$$

indicate that a step length α is not too large (that is, if one of those is violated, then the current step length is too large), while only the weak curvature condition

$$\langle \nabla f(x_k + \alpha p_k), p_k \rangle \geq c_2 \langle \nabla f(x_k), p_k \rangle$$

prevents too small step lengths α .

Alternatively, you can use the algorithm that is described in N&W, Section 3.5.

Solution.

See possible solutions on the wiki page.

Problem 3.

Let $A = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 2 \end{pmatrix}$ and $b = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$. Use the CG-method (Algorithm 5.2, N&W Book) with initialization $x_0 = 0$ for solving the linear system $Ax = b$.

Solution.

By applying Algorithm 5.2 with the initial value $x_0 = (0, 0, 0)$, we obtain the following:

$$\begin{aligned} r_0 &= (-1, 0, -1), p_0 = (1, 0, 1), \alpha_0 = 1, \\ x_1 &= (1, 0, 1), r_1 = (0, -2, 0), \beta_1 = 2, p_1 = (2, 2, 2), \alpha_1 = 1, \\ x_2 &= (3, 2, 3), r_2 = (0, 0, 0). \end{aligned}$$

Since $r_2 = 0$, we stop the process here and conclude that $x = (3, 2, 3)$ solves the linear system $Ax = b$ in two iterative steps.

Problem 4.

Show that when applied to a quadratic function, with exact line searches, both the Polak–Ribière formula given by (5.44) (N&W Book) and the Hestenes–Stiefel formula given by (5.46) (N&W Book) reduce to the Fletcher–Reeves formula (5.41a) (N&W Book).

Hint: Prove $r_{k+1}^T p_k = 0$, $r_k^T p_k = -r_k^T r_k$ and $r_{k+1}^T r_k = 0$ by using $x_{k+1} = x_k + \alpha_k p_k$, $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$ and $\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}$.

Solution.

Base case: for $k = 0$, the result is obvious.

Inductive step: let $\beta_l^{\text{FR}} = \beta_l^{\text{PR}} = \beta_l^{\text{HS}}$, for $l = 0, 1, 2, \dots, k$ is true. Then we have to prove that $\beta_{k+1}^{\text{FR}} = \beta_{k+1}^{\text{PR}} = \beta_{k+1}^{\text{HS}}$. For a quadratic function $f(x) = \frac{1}{2} x^T A x - b^T x$ with symmetric and positive definite matrix A , we have $\nabla f_k = r_k = A x_k - b$. For the nonlinear conjugate gradient methods, we have

$$x_{k+1} = x_k + \alpha_k p_k,$$

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k, \tag{1}$$

where β_k could be one of the following:

$$\beta_{k+1}^{\text{FR}} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}, \beta_{k+1}^{\text{PR}} = \frac{r_{k+1}^T (r_{k+1} - r_k)}{r_k^T r_k}, \beta_{k+1}^{\text{HS}} = \frac{r_{k+1}^T (r_{k+1} - r_k)}{(r_{k+1} - r_k)^T p_k}.$$

For an exact line search, we have

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}.$$

Now, we have to show that $\beta_{k+1}^{\text{FR}} = \beta_{k+1}^{\text{PR}} = \beta_{k+1}^{\text{HS}}$. We start with proving

$$\begin{aligned}
 r_{k+1}^T p_k &= (Ax_{k+1} - b)^T p_k \\
 &= (Ax_k + \alpha_k A p_k - b)^T p_k \\
 &= x_k^T A p_k - \frac{r_k^T p_k}{p_k^T A p_k} p_k^T A p_k - b^T p_k \\
 &= (Ax_k - b)^T p_k - r_k^T p_k \\
 &= r_k^T p_k - r_k^T p_k \\
 &= 0.
 \end{aligned} \tag{2}$$

Further,

$$\begin{aligned}
 r_k^T p_k &= r_k^T (-r_k + \beta_k p_{k-1}) \text{ (by (1))} \\
 &= -r_k^T r_k + \beta_k r_k^T p_{k-1} \\
 &= -r_k^T r_k + 0 \text{ (by (2))} \\
 &= -r_k^T r_k
 \end{aligned} \tag{3}$$

By using (2) and (3), we get

$$\beta_{k+1}^{\text{HS}} = \frac{r_{k+1}^T (r_{k+1} - r_k)}{(r_{k+1} - r_k)^T p_k} = \frac{r_{k+1}^T (r_{k+1} - r_k)}{r_k^T r_k} = \beta_{k+1}^{\text{PR}}. \tag{4}$$

Now, if we show that $r_{k+1}^T r_k = 0$, we get our desired conclusion. If we do not get solution at iterative step k , then by Theorem 5.3 (N&W Book, inequality 5.16), we have $r_{k+1}^T r_k = 0$. On the other hand, if we get solution at iterative step k , then obviously $r_k = 0$, which automatically gives $r_{k+1}^T r_k = 0$.

Thus,

$$\beta_{k+1}^{\text{FR}} = \beta_{k+1}^{\text{PR}}. \tag{5}$$

The expressions (4) and (5) render the conclusion that

$$\beta_{k+1}^{\text{FR}} = \beta_{k+1}^{\text{PR}} = \beta_{k+1}^{\text{HS}}.$$

□

Problem 5.

Prove that Lemma 5.6 (N&W Book, Page Number 125) holds for any choice of β_k satisfying $|\beta_k| \leq \beta_k^{\text{FR}}$.

Solution.

Have already discussed in the class!

Problem 6.

Assume that $m > n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. 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 $s_k \neq 0$:

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

$$\begin{aligned}\beta_{k+1} &\leftarrow \frac{\|s_{k+1}\|^2}{\|s_k\|^2}, \\ p_{k+1} &\leftarrow -s_{k+1} + \beta_{k+1}p_k, \\ k &\leftarrow k + 1.\end{aligned}$$

Assume that the matrix A has full rank. Show that the above mentioned algorithm is actually identical with the CG-algorithm (Algorithm 5.2 (N&W Book)) for the solution of $A^T Ax = A^T b$ (in the sense that the iterates x_k of both methods coincide).

Hint: Prove $r_k^{\text{CG}} = s_k$, $p_k^{\text{CG}} = p_k$, $\alpha_k^{\text{CG}} = \alpha_k$ and $x_k^{\text{CG}} = x_k$, for all $k = 0, 1, 2, \dots$ by the mathematical induction method.

Solution.

We provide an inductive argument, showing that

$$r_k^{\text{CG}} = s_k, p_k^{\text{CG}} = p_k, \alpha_k^{\text{CG}} = \alpha_k \text{ and } x_k^{\text{CG}} = x_k, \forall k = 0, 1, 2, \dots \quad (6)$$

Assuming that x_0 is arbitrary but equal for both methods, with superscript "CG" for the CG-parameters.

Remark: CG-algorithm is well-defined because $A^T A$ is symmetric positive definite (rank $A = n$).

Base case: for $k = 0$, we have

$$r_0^{\text{CG}} = (A^T A)x_0 - A^T b, r_0 = Ax_0 - b, \text{ and } s_0 = A^T r_0 = r_0^{\text{CG}},$$

so that

$$p_0^{\text{CG}} = -r_0^{\text{CG}} = -s_0 = p_0,$$

and

$$\alpha_0^{\text{CG}} = \frac{\|r_0^{\text{CG}}\|^2}{(p_0^{\text{CG}})^T (A^T A)p_0^{\text{CG}}} = \frac{\|r_0^{\text{CG}}\|^2}{\|Ap_0^{\text{CG}}\|^2} = \frac{\|s_0\|^2}{\|Ap_0\|^2} = \alpha_0.$$

Therefore,

$$x_1^{\text{CG}} = x_0 + \alpha_0^{\text{CG}} p_0^{\text{CG}} = x_0 + \alpha_0 p_0 = x_1.$$

Induction step: suppose next that (6) is true for some $k \geq 0$. Then

$$\begin{aligned}r_{k+1}^{\text{CG}} &= r_k^{\text{CG}} + \alpha_k^{\text{CG}} A^T A p_k^{\text{CG}} \\ &= s_k + \alpha_k A^T A p_k \\ &= A^T (r_k + \alpha_k A p_k) \\ &= A^T r_{k+1} \\ &= s_{k+1},\end{aligned}$$

$$p_{k+1}^{\text{CG}} = -r_{k+1}^{\text{CG}} + \frac{\|r_{k+1}^{\text{CG}}\|^2}{\|r_k^{\text{CG}}\|^2} p_k^{\text{CG}} = -s_{k+1} + \frac{\|s_{k+1}\|^2}{\|s_k\|^2} p_k = p_{k+1},$$

and

$$\alpha_{k+1}^{\text{CG}} = \frac{\|r_{k+1}^{\text{CG}}\|^2}{\|Ap_{k+1}^{\text{CG}}\|^2} = \frac{\|s_{k+1}\|^2}{\|Ap_{k+1}\|^2} = \alpha_{k+1},$$

so, most importantly

$$x_{k+1}^{\text{CG}} = x_k^{\text{CG}} + \alpha_k^{\text{CG}} p_k^{\text{CG}} = x_{k+1}.$$

Therefore, (6) holds for $k + 1$ as well. Now, we can say that

$$r_k^{\text{CG}} = s_k, p_k^{\text{CG}} = p_k, \alpha_k^{\text{CG}} = \alpha_k \text{ and } x_k^{\text{CG}} = x_k, \forall k = 0, 1, 2, \dots$$

Consequently, the given iterative algorithm coincide/identical with CG-algorithm for the solution of $A^T Ax = A^T b$. \square