CS5321 Numerical Optimization Homework 4

Due April 26

1. (30%) The conjugate gradient method for solving $Ax = b$ is given in Figure 1, where z_k is the approximate solution. In class, we only showed that $\alpha_k = (\vec{p}_k^T \vec{r}_k)/(\vec{p}_k^T A\vec{p}_k)$ and $\beta_k = -(\vec{p}_k^T A\vec{r}_{k+1})/(\vec{p}_k^T A\vec{p}_k)$. Prove that the above formulas of α_k and β_k are equivalent to the ones in step (3) and step (6). Your may need the relations in step (4) and step (5), and the following facts.

   (a) $\vec{r}_i$ and $\vec{r}_j$ are orthogonal to each other. (If $i \neq j$, $\vec{r}_i^T \vec{r}_j = 0$.)

   (b) $\vec{p}_i$ and $\vec{p}_j$ are A-conjugate to each other. (If $i \neq j$, $\vec{p}_i^T A\vec{p}_j = 0$.)

   (c) $\vec{p}_k$ is a linear combination of $\vec{r}_0, \ldots, \vec{r}_k$, $\vec{p}_k = \sum_{i=1}^{k} \gamma_i \vec{r}_i$. (which can be shown from step (7) by induction.)

---

(1) Given $\vec{z}_0$. Let $\vec{p}_0 = \vec{b} - A\vec{z}_0$, and $\vec{r}_0 = \vec{p}_0$.

(2) For $k = 0, 1, 2, \ldots$ until $\|\vec{r}_k\| \leq \epsilon$

(3) $\quad\quad \alpha_k = (\vec{r}_k^T \vec{r}_k)/(\vec{p}_k^T A\vec{p}_k)$

(4) $\quad\quad \vec{z}_{k+1} = \vec{z}_k + \alpha_k \vec{p}_k$

(5) $\quad\quad \vec{r}_{k+1} = \vec{r}_k - \alpha_k A\vec{p}_k$

(6) $\quad\quad \beta_k = (\vec{r}_{k+1}^T \vec{r}_{k+1})/(\vec{r}_k^T \vec{r}_k)$

(7) $\quad\quad \vec{p}_{k+1} = \vec{r}_{k+1} + \beta_k \vec{p}_k$

Figure 1: The CG algorithm

First, we show that

$$\alpha_k = \frac{\vec{p}_k^T \vec{r}_k}{\vec{p}_k^T A \vec{p}_k} = \frac{\vec{r}_k^T \vec{r}_k}{\vec{p}_k^T A \vec{p}_k}. \tag{1}$$

The only thing to prove is $\vec{p}_k^T \vec{r}_k = \vec{r}_k^T \vec{r}_k$. By using (c),

$$\vec{p}_k^T \vec{r}_k = \sum_{i=1}^{k} \gamma_i \vec{r}_i^T \vec{r}_k.$$

With (a), $\vec{r}_i \vec{r}_k = 0$ except $i = k$. Thus, $\vec{p}_k^T \vec{r}_k = \vec{r}_k^T \vec{r}_k$.

Second, we prove $\beta_k = -\frac{\vec{p}_k^T A \vec{r}_{k+1}}{\vec{p}_k^T A \vec{p}_k}$. We can use the result in the first step to simplify the proof.

$$\beta_k = -\frac{\vec{p}_k^T A \vec{r}_{k+1}}{\vec{p}_k^T A \vec{p}_k} = -\frac{\vec{r}_k^T \vec{r}_k}{\vec{p}_k^T A \vec{p}_k} \frac{\vec{p}_k^T A \vec{r}_{k+1}}{\vec{r}_k^T \vec{r}_k} = -\alpha_k \frac{\vec{p}_k^T A \vec{r}_{k+1}}{\vec{r}_k^T \vec{r}_k}.$$

Thus, we only need to show that $-\alpha_k \vec{p}_k^T A \vec{r}_{k+1} = \vec{r}_{k+1}^T \vec{r}_{k+1}$.

From step (5), $\vec{r}_{k+1} = \vec{r}_k - \alpha_k A \vec{p}_k$.

$$\vec{r}_{k+1}^T \vec{r}_{k+1} = (\vec{r}_k - \alpha_k A \vec{p}_k)^T \vec{r}_{k+1} = \vec{r}_k^T \vec{r}_{k+1} - \alpha_k \vec{p}_k^T A \vec{r}_{k+1}.$$

However, property (a) tells $\vec{r}_k^T \vec{r}_{k+1} = 0$.

Therefore, $-\alpha_k \vec{p}_k^T A \vec{r}_{k+1} = \vec{r}_{k+1}^T \vec{r}_{k+1}$.

2. (70%) Find the minimum of the Rosenbrock function

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2.$$

   (a) Implement the quasi-Newton method (BFGS) with line search, and test it with $(x_0, y_0) = (-1.2, 1.0)$ and initial Hessian $H_0 = I$. Let B_k be the BFGS approximation to the inverse of the Hessian H_k matrix. The formula of updating B_k is

$$B_{k+1} = (I - \rho_k \vec{s}_k \vec{y}_k^T) B_k (I - \rho_k \vec{y}_k \vec{s}_k^T) + \rho_k \vec{s}_k \vec{s}_k^T,$$

   where $\vec{s}_k = x_{k+1} - x_k$, $\vec{y}_k = \nabla f_{k+1} - \nabla f_k$, and $\rho_k = 1/\vec{y}_k^T s_k$.

   (b) Figure 2 shows the truncated Newton's method (TN) with line search. The *inner-loop* of TN, Line (5)-(14), is just like CG. Compare Line (4)-(14) with CG in Figure 1 and point out their differences.

   (c) Implement the truncated Newton's method (TN) with line search, and test it with $(x_0, y_0) = (-1.2, 1.0)$.

(1) Given an initial point $\vec{x}_0$.

(2) For $k = 0, 1, 2, ...$

(3) Compute $H_k, \nabla f_k$ and set $\epsilon_k = \min(0.5, \sqrt{\|\nabla f_k\|}) \times \|\nabla f_k\|$

(4) Given $\vec{z}_0$. Let $\vec{d}_0 = -\nabla f_k - A\vec{z}_0 = -\nabla f_k$, and $\vec{r}_0 = \vec{p}_0$.

(5) For $j = 0, 1, 2, \ldots$

(6) If $(\vec{d}_j^T H_k \vec{d}_j \leq 0)$

(7) $\vec{p}_k = \vec{z}_j$; (if $j = 0$, $\vec{p}_k = -\nabla f_k$.) break;

(8) $\alpha_j = (\vec{r}_j^T \vec{r}_j)/(\vec{d}_j^T H_k \vec{d}_j)$

(9) $\vec{z}_{j+1} = \vec{z}_j + \alpha_j \vec{d}_j$

(10) $\vec{r}_{j+1} = \vec{r}_j - \alpha_k H_k \vec{d}_j$

(11) If $(\|\vec{r}_{j+1}\| \leq \epsilon_k)$

(12) $\vec{p}_k = \vec{z}_{j+1}$; break;

(13) $\beta_j = (\vec{r}_{j+1}^T \vec{r}_{j+1})/(\vec{r}_j^T \vec{r}_j)$

(14) $\vec{d}_{j+1} = \vec{r}_{j+1} + \beta_j \vec{d}_j$

End for

(15) Use line search to find a_k and set $\vec{x}_{k+1} = \vec{x}_k + a_k \vec{p}_k$

End for

Figure 2: The truncated Newton algorithm