Norwegian University of Science and Technology
Department of Mathematical Sciences

# TMA4180 Optimisation I Spring 2018

## Solutions to exercise set 5

$\boxed{1}$ This exercise illustrates numerically the relationship between BFGS and CG with exact linesearch applied to convex quadratic problems. As in Exercise 1 from the previous week, 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 BFGS method with $x_0 = 0$ and $H_0 = I$ for solving minimizing the function $f(x) = 0.5x^\top Ax - b^\top x$. Compare the results with those produced by CG (see Theorem 6.4 in N&W).

**Solution:** We perform the computation:

$$\begin{aligned}
&x_0 = (0,0,0), && && \\
&g_0 = \nabla f(x_0) = (-1,0,-1), && p_0 = -H_0 g_0 = (1,0,1), && \alpha_0 = -\frac{g_0^\top p_0}{p_0^\top A p_0} = 1, \\
&x_1 = x_0 + \alpha_0 p_0 = (1,0,1), && g_1 = \nabla f(x_1) = (0,-2,0), && \\
&s_0 = x_1 - x_0 = (1,0,1), && y_0 = g_1 - g_0 = (1,-2,1), && \rho_0 = 1/(y_0^\top s_0) = 0.5, \\
&H_1 = (I - \rho_0 s_0 y_0^\top) H_0 (I - \rho_0 y_0 s_0^\top) + \rho_0 s_0 s_0^\top && = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} && \\
& && p_1 = -H_1 g_1 = (2,2,2), && \alpha_1 = -\frac{g_1^\top p_1}{p_1^\top A p_1} = 1, \\
&x_2 = x_1 + \alpha_1 p_1 = (3,2,3), && g_2 = \nabla f(x_2) = (0,0,0). &&
\end{aligned}$$

Note that the method has converged at this point — and performed exactly the same iterations as CG applied to this problem. Since we did not have to perform 3 iterations for this 3-dimensional problem, the matrix $H$ at the last iteration does not equal to

$$A^{-1} = \frac{1}{3} \begin{bmatrix} 5 & 3 & 4 \\ 3 & 3 & 3 \\ 4 & 3 & 5 \end{bmatrix}$$

$\boxed{2}$ We consider limited memory BFGS (L-BFGS) method, where we store only one pair of vectors $s_k, y_k$ at each iteration, and simply set the initial approximation for the inverse Hessian to be $H_k^0 = I$.

Show that this method with exact linesearch is equivalent with Hestenes–Stiefel nonlinear CG method (cf. N&W, p. 123) with exact linesearch.

**Solution:** The two linesearch methods in question (with the same/exact linesearch) will be equivalent if we can show that the search directions at each iteration are equivalent.

Since we use exact linesearch we have the relation

$$g_{k+1}^\top p_k = \nabla f(x_{k+1})^\top p_k = \nabla f(x_k + \alpha_k p_k)^\top p_k = \frac{d}{d\alpha} f(x_k + \alpha p_k)|_{\alpha=\alpha_k} = 0,$$

owing to the first-order necessary optimality conditions. As a consequence, at each iteration we have the equality $s_k^\top g_{k+1} = (x_{k+1} - x_k)^\top g_{k+1} = \alpha_k p_k^\top g_{k+1} = 0$. Therefore we can write

$$\begin{aligned} p_{k+1} &= -H_{k+1} g_{k+1} = -[(I - \rho_k s_k y_k^\top) H_k^0 (I - \rho_k y_k s_k^\top) + \rho_k s_k s_k^\top] g_{k+1} \\ &= -(I - \rho_k s_k y_k^\top) g_{k+1} = -g_{k+1} + \frac{(g_{k+1} - g_k)^\top g_{k+1}}{(g_{k+1} - g_k)^\top \alpha_k p_k} \alpha_k p_k = -g_{k+1} + \beta_{k+1}^{\mathrm{HS}} p_k, \end{aligned}$$

exactly as claimed.

$\boxed{3}$ Some analysis of Nedler–Mead method is deferred to the exercises in Nocedal & Wright.

**a)** Exercise 9.11: show that the average function value at the Nedler–Mead simplex points will decrease over one step if any of the points $\tilde{x}(-1)$, $\tilde{x}(-2)$, $\tilde{x}(-1/2)$, $\tilde{x}(1/2)$ are adopted as a replacement for $x_{n+1}$.

**Solution:** Note that $\tilde{x}(-1)$ is accepted when either $f_{-1} < f(x_n) \le f(x_{n+1})$ or $f_{-1} < f(x_1) \le f(x_{n+1})$.
$\tilde{x}(-2)$ is accepted when $f_{-2} < f_{-1} < f(x_1) \le f(x_{n+1})$.
$\tilde{x}(-1/2)$ is accepted when $f_{-1/2} \le f_{-1} < f(x_{n+1})$.
Finally, $\tilde{x}(1/2)$ is accepted when $f_{1/2} < f(x_{n+1})$.
Therefore, in all cases the function value at the new point is strictly smaller than $f(x_{n+1})$, and the average value over the simplex vertices (all other points being the same) will improve.

**b)** Exercise 9.12: show that if $f$ is a convex function, the shrinkage step in the Nedler–Mead simplex method will not increase the average function value over the simplex. Show that unless $f(x_1) = f(x_2) = \cdots = f(x_{n+1})$, the average value will in fact decrease.

**Solution:** Note that in the shrinkage step we replace $x_i$ with $\hat{x}_i = (x_1 + x_i)/2$, $i = 2, \dots, n+1$. As a result of convexity, $f(\hat{x}_i) \le [f(x_1) + f(x_i)]/2 \le f(x_i)$, $i = 2, \dots, n+1$, where the latter inequality is owing to the fact that $f(x_1)$ is the smallest function value over the simplex vertices.
Therefore the average over the simplex vertices cannot increase. Furthermore, unless all function values over the simplex points are the same, at least one of these inequalities is going to be strict.

Note that since there is no "sufficient decrease" guarantee in these exercises, the method may stagnate! Indeed, the following example is due to McKinnon (https://doi.org/10.1137/S1052623496303482):

**c)** Consider

$$f(x,y) = \begin{cases} \theta\phi|x|^\tau + y + y^2, & \text{if } x < 0, \\ \theta x^\tau + y + y^2, & \text{if } x \geq 0, \end{cases}$$

where $\theta$, $\phi$, $\tau$ are positive constants. For $\tau > 2$ the function $f$ is twice continuously differentiable, strictly convex, and the point $0, 0$ is not a point of its minimum (e.g., the direction $(0, -1)$ is a descent direction at this point). In fact, the minimum of this separable function is atained at $(0, -1/2)$.

Let $\lambda_1 = (1+\sqrt{33})/8$, $\lambda_2 = (1-\sqrt{33})/8$, and consider the simplex with vertices $(0,0)$, $(\lambda_1^n, \lambda_2^n)$, and $(\lambda_1^{n+1}, \lambda_2^{n+1})$ for some integer $n > 0$. It turns out that for carefully select parameters $\theta$, $\phi$ and $\tau$, Nedler–Mead algorithm for this function performs the "inside contraction" step, resulting in a sequence of simplices with the same structure as the starting simplex, $n \to \infty$. The resulting sequence of simplices "converges" to the non-stationary point $(0,0)$. Therefore, even for strictly convex functions in 2D the method may fail.

Implement Nedler–Mead algorithm, test it on this function to confirm the described behaviour. You can for example use $\tau = 3$, $\theta = 6$, and $\phi = 400$.

**Solution:** See a possible implementation on the wiki.

$\boxed{4}$ Implement the linesearch Newton-CG algorithm (algorithm 7.1 in the book) and test it on the Rosenbrock function for larger values of $n$:

$$f(x) = \sum_{i=1}^{n-1} [\alpha(x_{i+1} - x_i^2)^2 + (1 - x_i)^2],$$

where $\alpha > 0$ is a parameter, e.g. $\alpha = 100.0$. The global minimum is attained at $x^* = (1, 1, \ldots, 1)$.