TU Chemnitz
C. Helmberg

Chemnitz, November 1, 2010
due on November 18, 2010

Nonlinear Optimization
Exercises 3

1. (3) Complete the proof of the Kantorovich inequality: Given $0 < \lambda_1 \leq \cdots \leq \lambda_n$, then

$$\max_{\xi_i \geq 0, \sum \xi_i = 1} \left(\sum_{i=1}^n \xi_i \lambda_i\right)\left(\sum_{i=1}^n \xi_i \frac{1}{\lambda_i}\right) = \frac{1}{4}\frac{(\lambda_1+\lambda_n)^2}{\lambda_1\lambda_n}.$$

2. (3) Given the optimization problem $\min f(x,y)$ with $f(x,y) = \frac{cx^2+y^2}{2}$, $\quad (x,y) \in \mathbb{R}^2$, $\quad 0 < c < 1$ and starting point $x_0 = (1,c)$, prove that steepest descent with exact line search generates the sequence

$$x_k = (q^k, (-1)^k c q^k), \qquad k = 1,2,3,\ldots$$

where $q = \frac{1-c}{1+c}$. How many iterations are needed in order to get within 10^{-6} of the optimum for $c = 10^{-3}$?

3. (4) Complete the proof of Theorem II.6 by showing
   (a) $\|x_k + p_k - x^*\| = O(\|x_k - x^*\|^2) + o(\|p_k\|) \quad \Rightarrow \quad \|p_k\| = O(\|x_k - x^*\|)$
   (b) $\|x_k + p_k - x^*\| = o(\|x_k - x^*\|) \;\Rightarrow\; \|p_k\| = O(\|x_k - x^*\|)$ and $\|p_k - p_k^N\| = o(\|p_k\|)$

4. (2) Let $X \in \mathbb{R}^{n\times n}$ denote a symmetric matrix variable and define the inner matrix product of matrices $A, B \in \mathbb{R}^{m\times n}$ by $\langle A, B\rangle = \sum_{i,j} a_{ij} b_{ij}$. Prove:
   (a) Any $n \times n$ matrix C can be decomposed orthogonally into a symmetric and a skew symmetric part by $C = \frac{1}{2}(C + C^T) + \frac{1}{2}(C - C^T)$. In particular, $\langle C, X\rangle = \frac{1}{2}\left\langle C + C^T, X\right\rangle$ and $\frac{1}{2}\left\langle C - C^T, X\right\rangle = 0$.
   (b) For $f(X) = \frac{1}{2}\langle QXQ, X\rangle + \langle H, X\rangle$ with $Q \succ 0$, $H \in \mathbb{R}^{n\times n}$ the slightly incorrect but standard expression for stationarity $\nabla f(X) = QXQ + \frac{1}{2}(H + H^T) = 0$ does indeed produce the correct result.

5. (8) Implement a BFGS-Quasi-Newton-method in Matlab with calling sequence

```
[xsol]=bfgsnewton(@fun,xstart)
```

using your line search routine of the last exercises. As before, `fun` represents an arbitrary sufficiently smooth function $f(x)$ of the form

```
[fval,fgrad]=function_name(x)
```

with `fval`$= f(x)$ and `fgrad`$= \nabla f(x)$. Test it on the Rosenbrock function for different starting points.