Optimierung I

Problem Sheet 9

Due date: 17. June 2014

Problem 9.1: [Q-Convergence order]
Determine the Q-convergence order of the sequence $(x_k)_k$ with

- $x_k = \frac{1}{k}$
- $x_k = 1 + (\frac{1}{2})^{2^k}$
- $x_k = (\frac{1}{k})^k$

Problem 9.2: [Gradient-like methods (I)]
Take $f \in C^1(\mathbb{R}^n, \mathbb{R})$. For the general descent method, show that, if the descent directions are gradient-like (Gradientenähnlich) and the stepsizes are chosen according to the Armijo rule, i.e.

$$\alpha_k = \max \left\{ \gamma^i \,|\, i \in \mathbb{N} \text{ such that } f(x_k + \gamma^i d_k) \leq f(x_k) + \sigma \gamma^i \nabla f(x^k)^T d^k \right\}, \tag{1}$$

then every accumulation point of the sequence $(x_k)_k$ generated by the method is a stationary point of f.

Problem 9.3: [Gradient-like methods (II)]
Take $f \in C^1(\mathbb{R}^n, \mathbb{R})$ and $(H_k)_k$ in $\mathbb{R}^{n \times n}$ to be a sequence of symmetric, positive definite matrices satisfying, for $\mu_1, \mu_2 > 0$,

$$\mu_1 \|x\|^2 \leq x^T H_k x \leq \mu_2 \|x\|^2$$

for all $x \in \mathbb{R}^n$. For the general descent method, define the descent directions from

$$H_k d^k = -\nabla f(x^k).$$

Show that, if the stepsizes are chosen again according to the Armijo rule, then every accumulation point of the sequence $(x_k)_k$ generated by the method is a stationary point of f.

Problem 9.4: [Convergence of gradient like methods]
Take $f \in C^1(\mathbb{R}^n, \mathbb{R})$ and suppose that f is coercive, i.e. for any sequence $(z_k)_k$ in $\mathbb{R}^n$ we have

$$\|z_k\| \to \infty \Rightarrow f(z_k) \to \infty \text{ as } k \to \infty.$$

Take x^* to be a global minimizer of f. Suppose again that (x_k) is a sequence generated by a gradient like method for f with Armijo stepsize choice. Show the following

i) (x_k) has a subsequence converging to a stationary point of f.

ii) If f is convex, $f(x_k) \to f(x^*)$ as $k \to \infty$.

iii) If f is strictly convex, $x_k \to x^*$ as $k \to \infty$.

Problem 9.5: [Programming Exercise: Newtons Method with CG]
Write a Matlab/Octave implementation of Newtons method. Implement the conjugate gradient method to solve the inversion of the Hessian. As a results, you should provide the following function (with hollerm replaced accordingly):

$$[x, e] = hollerm(x0, en, ecg, Nn, Ncg)$$

where

- x0 is the initial point for the Newton iteration
- The Newton method stops if $\|\nabla f(x_k)\| < en$ or $k >= Nn$
- The inner CG method stops if $\|\nabla f(x_k)\| < ecg$ or $k >= Ncg$
- x is the solution of your method
- e is an error indicator, $e = 0$ if the method was successful, $e = 1$ if the CG method reached the maximal iteration number, $e = 2$ if the Newton method reached the maximal iteration number.

Use 0 as initialization for the CG method in the first Newton iteration and the solution of the CG method of the previous iteration as initialization for the next CG method. You can assume that in the current folder the functions

g = grad(x)   h = hessian(x)

are available, providing a Gradient and Hessian evaluation at the point x.
Test your program with different examples, e.g.

- $f(x) = x^T Ax + b^T x$, for example with $A = [14, 23, 22; 23, 86, 14; 22, 14, 51]$; and $b = [2; -3; 1]$;.
- $f(x) = (x^3)/3 - 2x$. (Interpret the result)

Test different initializations, error bounds and iteration numbers. Follow the rules of the previous programming exercise, in particular send your main file to your group leader at least one hour before class, using *Optimierung 1, Abgabe 9.5* as subject.