# Numerical Optimization

## Homework 3

Due 11.06.2014

Give your answers with logical and/or mathematical explanations. Hand-in your homework in the beginning of a lecture on due date. Late submissions will not be accepted. Assigned points are shown in square brackets, which will be re-scaled so that the total homeworks points will be 40.

**1.[5]** Consider the (exact) line search problem for a given point $x_k \in \mathbb{R}^n$ and a descent direction $p_k \in \mathbb{R}^n$, searching for $\alpha_k > 0$ that solves

$$\min_{\alpha>0} \quad \phi(\alpha) = f(x_k + \alpha p_k)$$

for a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. Show that for a convex quadratic function $f(x) = \frac{1}{2}x^T Q x + c^T x$ where $Q$ is positive definite, the unique global minimizer $\alpha_k > 0$ of $\phi(\alpha)$ is given by

$$\alpha_k = -\frac{\nabla f(x_k)^T p_k}{p_k^T Q p_k}.$$

Hint: first show that $\phi(\alpha)$ is a strictly convex function.

**2.[5]** For a twice continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, denote its Hessian matrix at a point $x_k \in \mathbb{R}$ by $H_k = \nabla^2 f(x_k)$. Suppose that the Hessian is positive definite and has a uniformly bounded condition number, that is, for there exists a constant $M > 0$ such that

$$\kappa(H_k) = \|H_k\|_2 \|H_k^{-1}\|_2 \le M \quad \forall k.$$

Show in this case the Newton direction $p_k^N = -H_k^{-1}\nabla f(x_k)$ satisfies that

$$\cos\theta_k = \frac{-\nabla f(x_k)^T p_k^N}{\|\nabla f(x_k)\|_2 \|p_k^N\|_2} \in \left[\frac{1}{M}, 1\right].$$

**3.[15]** (Backtracking Linesearch)

A popular line search strategy is called the *backtracking* line search, which typically uses only the Armijo condition to check satisfactory step sizes. A sketch of the backtracking line search algorithm is shown in Algorithm 1. **[3pt]** This algorithm is guaranteed to find an $\alpha_k > 0$ and perhaps not too small one if $\alpha_0$ is sufficiently large. Explain briefly why this is true.

**[12pt]** A typical minimization loop is illustrated in Algorithm 2. Implement the minimization loop with backtracking line search in `R`, using the steepest descent and the Newton directions to find a stationary point of the Rosenbrock function[1]. We've shown that $(1,1)^T$ is a unique minimizer, in homework 2. Try two starting points, $(1.2, 1.2)^T$ and $(-1.2, 1)^T$, for two different choices of search directions, to see if they find the minimizer and how they behave differently.

For implementation, we recommend using the following values:

[1] The `R` code for $f$, $\nabla f$, and $\nabla^2 f$ are available from the lecture website.

- Minimization: $tol = 10^{-16}$ and $max.iter = 100$.
- Line search: $\alpha_0 = 1.0$, $\rho = 0.5$, $c_1 = 10^{-4}$ and $max.ls.iter = 25$.

**Algorithm 1:** Backtracking Linesearch

```
Input: x_k, f, ∇f, α_0 > 0, ρ ∈ (0,1), c_1 ∈ (0,1), and max.ls.iter > 0;
α ← α_0;
ls.iter ← 0;
while ls.iter < max.ls.iter do
    if f(x_k + α p_k) ≤ f(x_k) + c_1 α ∇f(x_k)^T p_k then
        break;
    end
    α ← ρα;
    ls.iter ← ls.iter + 1;
end
Output: α_k = α.
```

**Algorithm 2:** Minimization Loop

```
Input: x_0, f, ∇f, ∇^2 f, tol > 0, and max.iter > 0;
k ← 0;
x_k ← x_0;
while k < max.iter do
    if ||∇f(x_k)||_2 < tol then
        break;
    end
    Choose a descent direction p_k;
    Choose α_k from linesearch;
    x_k ← x_k + α_k p_k;
    k ← k + 1;
end
Output: x* = x_k.
```

Points will be given as follows: running implementation of line search with steepest descent direction (2pt + 3pt if correct) and newton's direction (2pt + 3pt if correct). 2pt will be given to correct observations of the difference between the steepest descent and Newton's methods from the output of your code.

Submit:
(1) print out your `R` code.
(2) for each of two starting points and the two choices of search directions, print and submit the following information in each iteration of Algorithm 2,

- Iteration number $k = 0, 1, \ldots$.
- Chosen step size $\alpha_k$.
- Function value at $x_{k+1}$.
- $\|\nabla f(x_{k+1})\|_2$.

Before beginning the while loop, show the function value at $x_0$ and $\|\nabla f(x_0)\|_2$ and check if $x_0$ satisfies the stopping condition so that the code will immediately return if $x_0 = (1, 1)^T$ is given.