Universität Konstanz
Fachbereich Mathematik und Statistik
Prof. Dr. Stefan Volkwein
Roberta Mancini, Carmen Gräßle, Laura Lippmann, Kevin Sieg

Sommersemester 2014

Optimierung

http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/

Program 1 (6 Points)

Submission by E-Mail: 2014/05/21, 10:00 h

Note:

- Work in **groups of 2 to 3 members**! The members have to be participants of the **same tutorial group**!
- Do not forget to write **name** and **email adress** of the authors in each file and document your code well!
- Only **running programs** will be considered!
- Stick to the **given function and parameter definitions** as described below! You should not modify them in name or concerning the input and output arguments.

Implement the Armijo stepsize algorithm from the lecture using MATLAB. Write herefore a function

```
function [t] = armijo(fhandle, x, d, t0, alpha, beta, amax)
```

in a file `armijo.m`. The function returns the stepsize `t` that complies with the Armijo condition. As input arguments the function accepts a function handle `fhandle`, current point `x`, descent direction `d`, initial stepsize `t0`, parameters `alpha` and `beta` as known from the lecture and `amax` the maximum number of iterations.

Implement the general descent method (Algorithmus 3.4) with direction $d^k := -\frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}$ using the Armijo stepsize strategy. Write a file `gradmethod.m` for the function

`function [X] = gradmethod(fhandle, x0, epsilon, nmax, t0, alpha, beta, amax).`

with initial point `x0`, tolerance `epsilon` for the termination condition $\|\nabla f(x^k)\| < \epsilon$, `nmax` the maximum number of iterations, and parameters `t0`, `alpha`, `beta` and `amax` for the Armijo rule.

The program should return a matrix `X = [x0; x1; x2; ...]` containing the whole iterations.

Test your program by using the following functions and parameters:

1. The function $f(x) = \cos(x)/x$ with $x \in [2\pi, 6\pi]$, `epsilon=1.0e-3`, `nmax=100`, `t0=1`, `alpha=1.0e-2`, `beta=0.5` and `amax=100`. Consider two different initial points `x0`: `x0=12` and `x0=14`. Explain the results in the written report.

2. The Rosenbrock function $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, $x = (x_1, x_2)^\top \in \mathbb{R}^2$, with `x0 =[1;-0,5]`, `epsilon=1.0e-2`, `t0=1`, `alpha=1.0e-2`, `beta=0.5` and `amax=100`. Call the function for `nmax=1000` as well as for `nmax=4000`: Comment on the results in the written report.

Herefore, write function files `cosinus.m` and `rosenbrock.m` which accept an input argument `x` and return the function and gradient values at `x`.

Finally, write a file `main.m` where you set the parameters and input functions and call the descent algorithm.

Send your MATLAB source code files to the corresponding tutor.