Universität Konstanz
Fachbereich Mathematik und Statistik
Prof. Dr. Stefan Volkwein
Sabrina Rogg

Sommersemester 2015

Optimierung

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

Program 1 (6 Points)

Submission by E-Mail: 2015/05/18, 10:00 h

Note:

- 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. The input arguments are as follows:

- `fhandle`: function handle
- `x`: current point
- `d`: descent direction
- `t0`: initial stepsize
- `alpha` and `beta`: parameters known from the lecture
- `amax`: maximum number of iterations

Implement the general descent method (Algorithmus 3.5) 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 input arguments

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

The program should return a matrix `X = [x0, x1, x2, ...]` containing the whole iterations. Note that the program should be able to deal with an arbitrary dimension $n \in \mathbb{N}$ ($f : \mathbb{R}^n \to \mathbb{R}$).

Test your program by using the following functions and parameters:

1. The test function

$$f = -e^{-((x_1-\pi)^2+(x_2-\pi)^2)} + c\sin(x_1)\cos(x_2 + \frac{\pi}{2}), \quad x = (x_1, x_2)^\top \in \mathbb{R}^2,$$

   with $c = 0.1$, `epsilon=1.0e-2`, `t0=1`, `alpha=1.0e-2`, `beta=0.5`, `amax=30` and `nmax=100`. Consider two different initial points `x0`: `x0=[5.5;5.5]` and `x0=[5;1.3]`. What happens if you set $c = 0$? 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=30`. Call the function for `nmax=100` as well as for `nmax=4000`. Comment on the results in the written report.

Herefore, write function files `testfunction.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.