Prof. Dr. Karsten Urban
M. Sc. Lewin Ernst

Winter term 2024/2025
Sheet 03, 30 Points

Numerical Optimization
exercise sheet

review on 06.11.2024 during the exercise class

1. *(Convex functions, again)* Let $C \subset \mathbb{R}^n$ be convex. Show:

   (a) *Maximum function:* If $f : C \to \mathbb{R}$ is convex, then $g(x) := \max\{f(x), 0\}$ is also convex on C.

   (b) *Quadratic function:* If $f : C \to \mathbb{R}$ is convex and $f(x) \geq 0$, then $g(x) := f(x)^2$ is also convex on C.

(3 + 5 = 8 Points)

2. *(Derivative-free methods)*

   a) One can prove for the Nelder-Mead algorithm that, if the contraction step of the simplex at iteration k around the best point x_1 (line 17 and line 24 of the algorithm) does not occur, then the average function value
   $$\frac{1}{n+1}\sum_{i=1}^{n+1} f(x_i^{(k)})$$
   at the simplex values $\{x_1^{(k)}, \dots, x_{n+1}^{(k)}\}$ will decrease at each step. *Justify* this statement.

   b) Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex. Show that the contraction step of the simplex around the best point (line 17 and line 24) will not increase the average function value. Show also that unless $f(x_1^{(k)}) = \ldots = f(x_{n+1}^{(k)})$, the average value will in fact decrease.

   c) To which problem would you apply the derivative-free minimization algorithms? Describe one problem.

(4 + 4 + 2 = 10 Points)

3. *(Nelder-Mead and Hooke-Jeeves,* MATLAB*)*
   You will find the MATLAB functions

```
[x,niter] = nelder_mead(f, x0, alpha, beta, gamma, tol, maxIt),
```

```
[xk,niter] = hooke_jeeves(f, x0, h, tol, maxIt, x_sol)
```

and

```
[xk,niter] = gradientDescentForwardDiff(func, x0, alpha, maxIt, tol, x_sol, epsilon)
```

in the material. Note that the latter two have a quite unusual input, namely the solution of the minimization problem.

a) Apply the MATLAB-function `nelder_mead.m` to the function $f : \mathbb{R}^2 \to \mathbb{R}$ with

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

using the initial simplex $x^{(0)} = \{(-1.2, 1),\ (-0.23407, 1.25882),\ (-0.94118, 1.96593)\}$. Create a plot displaying the contour lines of f together with the simplex of each iteration. Discuss your results.

b) Apply the MATLAB-function `nelder_mead.m` to the function $g : \mathbb{R}^2 \to \mathbb{R}$ (McKinnon) with

$$g(x, y) = \begin{cases} 360x^2 + y + y^2, & x < 0 \\ 6x^2 + y + y^2, & x \geq 0 \end{cases}$$

using the initial simplex $x^{(0)} = \{(1, 1),\ (0.8, -0.6),\ (0, 0)\}$. Create a plot displaying the contour lines of g together with the simplex of each iteration. Discuss your results.

c) Apply the MATLAB-functions `hooke_jeeves.m` and `gradientDescentForwardDiff.m` to the functions $f_1, f_2 : \mathbb{R}^2 \to \mathbb{R}$ with

$$f_1(x) = x^T Ax \quad \text{and} \quad f_2(x) = x^T Ax + 10^{-4} \cdot X,$$

where

$$A = \operatorname{diag}(3, 1) \quad \text{and} \quad X \sim \mathcal{U}([0, 1)).$$

The drawing of the random variable $X \sim \mathcal{U}([0, 1))$ at each evaluation of f_2 can be realized by `rand(1)`. The function `gradientDescentForwardDiff.m` implements the gradient descent algorithm with the gradient being approximated by forward differences, i.e.

$$(\nabla f(x))_j \approx \frac{f(x + \epsilon e_j) - f(x)}{\epsilon},$$

where $\epsilon \approx 10^{-8}$. Both functions f_1 and f_2 have a unique minimum point at $x^* = (0, 0)^T$. Start the algorithms with the same error tolerance `tol` and let it run until the tolerance or a maximum number of iteration is reached, i.e. until

$$\|x^* - x^{(k)}\|_2 < \texttt{tol} \text{ or } k \geq \texttt{maxIt}.$$

What do you observe?

(6 + 2 + 4 = 12 Points)