V. Schulz
M. Klar

SS 2022
Submission: 10.05.21, until 12:15

Numerical Optimization - Sheet 5

If you are a student in mathematics please solve the exercises with no tag and the ones with the tag **Mathematics**. If you are a data science student please solve the problems with no tag and those with the tag **Data Science**. Submissions with tags other than your subject count as bonus points. The tag **Programming** marks programming exercises.

Ex 1 Data Science (4 Points + *2 Bonus Points)

Let the norms $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ be equivalent as defined in the lecture. Show the following properties for a sequence of vectors $\{x^k\}_{k=1}^{\infty}$ converging to $\hat{x}$:

(i) Superlinear convergence in norm $\|\cdot\|_\alpha$ implies superlinear convergence in norm $\|\cdot\|_\beta$.

(ii) Sublinear convergence in norm $\|\cdot\|_\alpha$ implies sublinear convergence in norm $\|\cdot\|_\beta$.

(iii) ***** What (highly restrictive) additional condition do we need to ensure that the same implication holds for linear convergence?

Ex 2 Mathematics (4 Points)

Assume we apply the gradient descent method with exact line search to the problem

$$\min \; f(x) = \frac{1}{2} x^\top Q x$$

where $f : \mathbb{R}^n \to \mathbb{R}$, the scalar product is $(\cdot,\cdot) = (\cdot,\cdot)_2$, and $Q \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix. Please show that

$$f\left(x^{k+1}\right) = \left(1 - \frac{\left(g_k^\top g_k\right)^2}{\left(g_k^\top Q g_k\right)\left(g_k^\top Q^{-1} g_k\right)}\right) f\left(x^k\right)$$

where $g_k = \nabla f(x^k) = Qx^k$.

Ex 3 (*4 Bonus Points)

The introduction of the Wolfe conditions raises the question whether we can guarantee that it is possible to find a point which fulfills them.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and d a descent direction at point x, i.e. $\nabla f(x)^T d < 0$. Let f additionally be lower bounded on the ray $\{x + \alpha d \,|\, \alpha > 0\}$. Show in the following steps that for a fixed $0 < c_1 < c_2 < 1$ there is a step size $\alpha > 0$ such that the Wolfe conditions

$$\phi(\alpha) := f(x + \alpha d) \leq f(x) + c_1 \alpha \nabla f(x)^T d =: \ell(\alpha), \tag{1}$$
$$\phi'(\alpha) = \nabla f(x + \alpha d)^T d \geq c_2 \nabla f(x)^T d, \tag{2}$$

are fulfilled.

(i) Show that there is a smallest step size $\beta > 0$ such that

$$\phi(\beta) = \ell(\beta),$$

and that the inequality

$$\phi(\beta') < \ell(\beta')$$

holds for all $\beta' \in (0, \beta)$.

(ii) Deduce that there is an $\alpha \in (0, \beta)$ which fulfills

$$\phi'(\alpha) = \ell'(\alpha).$$

(iii) Conclude the assertion from (i) and (ii).

Ex 4 Programming (12 Points)

Implement a function `gradientdescent(f, x, tol, maxit, method)` which takes as parameters a callable function `f`, and a numpy array `x`. The float and integer numbers `tol` and `maxit` will terminate the gradient descent algorithm after an error tolerance is reached or the maximum number of allowed iterations is exceeded. Implement the function such that it allows...

(i) to set `method="constant"`, which just evokes a gradient algorithm with constant step size $\alpha = 0.001$, which is not even a descent algorithm in general!

(ii) to set `method="armijo"`, which executes the *backtracking line-search*.

(iii) to set `method="wolfe"`, which executes a *wolfe line-search* by reducing the step size until the wolfe conditions are fulfilled.

(iv) Test your algorithm using the function `test(gradientdescent)` of the provided module `graddesc_test`.

Hint: You might like to incorporate a callback function as it is provided by the `class CallBack` in `graddesc_test`.