University of Mannheim
Scientific Computing, B6 26, C312, D-68131 Mannheim
Dr. Andreas Sommer (ansommer@mail.uni-mannheim.de)

Nonlinear Optimization (FSS 2023)

Exercise Sheet #2

Due on Wednesday, 01.03.2023 (before 18:00).

---

1. Effect of scaling on the method of steepest descent [4 points]

Let $A \in S^n$ be positive definite, $f : \mathbb{R}^n \to \mathbb{R}$ continuously differentiable. Using the change of variables $x := Ay$, define $h(y) := f(Ay)$.

(a) Compute the gradient of h.

(b) Suppose you perform the method of steepest descent on both $f(x)$ and $h(y)$. How are the respective iterates related? What can be said about the numerical minimization of h versus f?

2. Steepest descent for induced norms [4 points]

Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, $H \in S^n$ be positive definite and $\|\cdot\|_H$ be the norm defined by

$$\|x\|_H := \sqrt{x^\top H x}\ .$$

Show that the direction of steepest descent of f in $x \in \mathbb{R}^n$ w.r.t. the norm $\|\cdot\|_H$, i.e. the solution of the following minimization problem

$$\min_{d \in \mathbb{R}^n} \nabla f(x)^\top d \qquad \text{s.t. } \|d\|_H = 1$$

is given by

$$d = -\frac{H^{-1}\nabla f(x)}{\|H^{-1}\nabla f(x)\|_H}\ .$$

3. Programming Exercise: Steepest Descent [10 points]

Implement the steepest descent method with a backtracking line search. Proceed as follows:

(a) Write a function which computes a step length by the Armijo rule:

```
function sigma = armijo(f,gradf,x,d,gamma,beta)
```

Here, `f` and `gradf` are functions which return for given x the functional value $f(x)$ and the gradient $\nabla f(x)$. Furthermore, `x` denotes the current iterate, d the descent direction and `gamma` and `beta` are the constants introduced in the lecture.

The return value `sigma` is a step length which satisfies the Armijo condition. (Hint: familiarize yourself with function handles (operator `@`), which can be used to pass a function to another one.)

(b) Implement the steepest descent method

```
function X = steepestdesc(f,gradf,x0,e,maxit) ,
```

which calls the function `armijo` to determine the step size in each iteration. The functions `f` and `gradf` are defined as described above, `x0` denotes the starting point.

The iteration should be terminated if $\|\nabla f(x^k)\| \le \epsilon$ for a given tolerance `e` $= \epsilon > 0$ or a maximum number of iterations `maxit` has been reached. The return value contains all iterates x^k as a matrix.

(c) Test your implementation on the Rosenbrock function

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Use (1) $x^{(0)} = (1, -0, 5)^\top$ and (2) $x^{(0)} = (-1.2, 1)^\top$ as initial guesses and $\epsilon = 10^{-3}$, $maxit = 10000$, $\gamma = 10^{-3}$ and $\beta = 1/2$.

Plot the trajectories of the iterates for each starting value.

(d) Consider the function $f(x) = \frac{2}{3}x^3 + \frac{1}{2}x^2$. Calculate all (local) minima and maxima. Test your implementation on that function with three different initial guesses (1) $x^{(0)} = 1$, (2) $x^{(0)} = 0.5$, and (3) $x^{(0)} = 0.1$, using the same parameters as in part (c). For each initial guess, list the first five ($k = 1...5$) iterates $x^{(k)}$, search directions $d^{(k)} = -\nabla f(x^{(k)})$, step lengths σ_k and function values $f(x^{(k)})$. Describe your observations.