Prof. Dr. Manuel Torrilhon
Dr. Robert Speck
Tuan Vo
Edilbert Christhuraj
Jan Habscheid

Applied and Computational Mathematics | RWTH AACHEN UNIVERSITY

Mathematische Grundlagen III (CES) | WS 2022
Programmierübung 4 | 18.01.2023

Hinweise zur Abgabe einer Programmieraufgabe (bitte sorgfältig lesen!):

- Die Programmieraufgaben sind in **Dreier- bis Fünfergruppen** zu bearbeiten.
- Vor Abgabe der Aufgabe müssen alle Programmverifikationen erfolgreich sein.
- Zum Testieren tragen Sie sich bitte ab **heute bis 01.02.2023 14:00 Uhr** in den Terminplaner im Moodle ein.
- Der Testattermin findet am **02.02.2023 (Thu) 14:00–17:00 + 03.02.2022 (Fr) 09:30–12:00** in **Seminarraum 328, Rogowski Gebäude** statt.
- Bringen Sie zu dem Testattermin bitte Ihr lauffähiges `matlab`-Programm mit.
- Sie werden dazu aufgefordert werden Ihren Bildschirm zu teilen.

Optimization using Gradient descent method

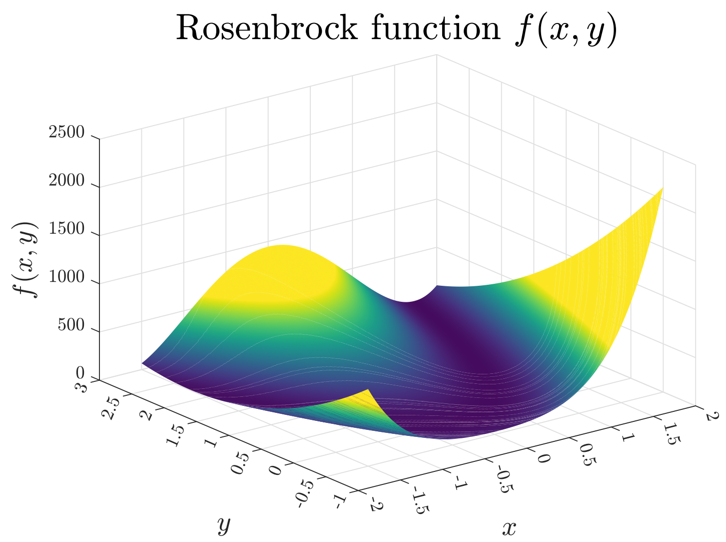

Abbildung 1: *Rosenbrock* function $f(x,y) = a\,(y-x^2)^2 + (1-x)^2$ with $a = 105$.

Gradient descent method, which is also often called the **steepest descent** method, is applied for problems of searching minimum

$$\min_{x\in\mathbb{R}^n} f(x),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a real value function, and f is differentiable. This programming exercise is about a non-restricted optimization problem as there is no constrain needed to fulfil. In this exercise we shall accomplish an implementation for the gradient descent method along with taking into consideration the *Armijo* condition, which is used for controlling the step size. In particular, the *Armijo* condition gives rise to the greatest possible step size s. In this context, the step size s is taken from the list

$$s \in \left\{1, \beta, \beta^2, \beta^3, \dots\right\},$$

in such a way that the following inequality

$$f(x+sr) - f(x) \le s\,\gamma\,\nabla f(x)^{\mathsf{T}}\,r$$

is satisfied. Note in passing that $r = -\nabla f(x)$ is the direction of the gradient descent method, while $\beta, \gamma \in (0,1)$ are the two constants given in advance. In this programming exercise we choose $\beta = 0.5$ and $\gamma = 10^{-4}$.

Program building block

(i) In the main program, e.g. `main.m` in `matlab`, define the function `f` and its gradient `gf`.

(ii) Then, we create a function for the steepest descent method

```
function [x, nsteps] = steepest(f, gf, x0, eps, maxit)
```

where `x0` is the starting point, `eps` is the error tolerance for the steepest descent method, and `maxit` stands for the maximal number allowed for the iteration.

(iii) The `steepest` function should return the minimum `x` and the step number `nsteps`.

(iv) Furthermore, we shall need to create a routine for the *Armijo* condition

```
function [s] = Armijo(f, gf, x)
```

which is used along with the `steepest` function to control the step size `s`. Within the `Armijo` function two constants `beta` and `gamma` can be directly defined.

Functions to be tested
Find the minimum of the following two functions by using Steepest descent method

a) The *Rosenbrock* function f given by

$$f : \begin{cases} \mathbb{R} \times \mathbb{R} \to \mathbb{R}, \\ (x_1, x_2) \mapsto f(x_1, x_2) := a\,(x_2 - x_1^2)^2 + (1 - x_1)^2, \end{cases} \qquad \forall a \in \mathbb{R}_+$$

with the starting point $x_0 = (-1.2,\, 1)^T$.

b) The function g defined as follows

$$g : \begin{cases} \mathbb{R} \times \mathbb{R} \to \mathbb{R}, \\ (x_1, x_2) \mapsto g(x_1, x_2) := x_1^2 + (x_1^2 + 1)(x_2^2 - 1)^2, \end{cases}$$

with the starting point $x_0 = (-10,\, 30)^T$.

To-do list

a) Examine the *Rosenbrock* function:

(i) Let $a = 100$, `maxit = 10000`, and take into consideration different `eps` values

$$\texttt{eps} = \Big\{0.1, 0.01, 0.001, 0.0001, 0.00001\Big\}.$$

(ii) Then, increase the parameter a to $a = 1000$, and another greater, as preferable. What has been observed, and if so, what shall be the reason for this behaviour?

b) Likewise, examine the function g and find its approximated minimum.

c) Extra questions and discussion + Teamwork presentation.

12+6+2 Punkte

Hints: Any other programming language is very much welcome, for example `Julia`, `Mathematica`, `Python`, `C/C++`, `Java`, and so on. The `matlab` code above is one option.