

Programmieraufgabe I (5+5+5 = 15 Punkte)

Abgabe in der Woche 15.-18. April

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
```

Teilaufgabe a)

Machen Sie sich mit der Funktionalität der Python-Bibliothek **matplotlib.pyplot** vertraut. Erzeugen Sie anschauliche Höhenlinien- bzw. Surface-Plots der folgenden Funktionen:

- $f(x_1, x_2) := 100(x_2 - x_1^2)^2 + (1 - x_1)^2$
- $g(x_1, x_2) := (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$
- $h(x_1, x_2) := (x_1 - 2)^4 + (x_1 - 2x_2)^2$
- $k(x_1, x_2) := \frac{1}{2}(x_1^2 + x_2^2) - \cos(x_1^2) - \cos(x_2^2) + 2$

In []:

Teilaufgabe b)

Implementieren Sie das Mutations-Selektions-Verfahren. Input soll eine Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$, ein Startwert $x_0 \in \mathbb{R}^n$, eine Schrittweite $\sigma > 0$ sowie die Anzahl der durchzuführenden Iterationen N sein. Ausgegeben werden soll die letzte Iterierte sowie eine Liste mit allen anderen Iterierten.

```
In [2]: def mutationselektion(f, x0, sigma, N):
return x, xarray
```

Testen Sie Ihre Implementierung für $\sigma = 0.05$ und $N = 1000$ Iterationen an den Funktionen f , g und h jeweils für den Startwert $x_0 = (1, 0)^T$. Plotten Sie jeweils den Verlauf der Iterierten in einen Höhenlinienplot der Funktion. Achten Sie dabei auf eine sinnvolle Skalierung der Achsen.

Plotten Sie auch jeweils die Funktionswerte im Laufe der Iteration. Skalieren Sie die Achsen derart, dass Sie erkennen können, wo auch im späteren Verlauf der Iteration noch Änderungen auftreten.

In []:

Wiederholen Sie Ihren Test für die Funktion g , Schrittweite $\sigma = 0.05$ und $N = 1000$, allerdings nun mit verschiedenen Startwerten $x_0 \in \{(0, 0)^T, (0, -1)^T, (0, 0)^T\}$. Lassen Sie erneut die Iterierten in einen Höhenlinienplot einzeichnen.

In []:

Führen Sie schließlich 20 Anwendungen des Mutations-Selektionsverfahrens für die Funktion g mit Startwert $x_0 = (0, 0)^T$, Schrittweite $\sigma = 0.05$ und $N = 1000$ durch und speichern Sie jeweils nur die letzte Iterierte in einer Liste. Was fällt auf?

In []:

Teilaufgabe c)

Implementieren Sie **Simulated Annealing**. Input soll eine Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$, die Temperaturfunktion $t : \mathbb{N} \rightarrow (0, \infty)$, die Anzahl der Iterationen N sowie ein Startwert $x_0 \in \mathbb{R}^n$ und die Schrittweite $\sigma > 0$ sein.

Ausgegeben werden soll die letzte Iterierte sowie eine Liste mit allen Iterierten sowie den aktuell besten Iterierten.

```
In [3]: def simulatedannealing(f, t, N, x0, sigma):
        return xbest, xarray, xbestarray
```

Testen Sie Ihre Implementierung an der Funktion k mit Startwert $x_0 = (2, 1)^T$ und $N = 10000$. Variieren Sie die Schrittweite σ und probieren Sie verschiedene Temperaturfunktionen $t_1(n) = 0.95^n$, $t_2(n) = 1/n$, $t_3(n) = 1/\sqrt{n}$, $t_4(n) = 1/\log(n)$ aus.

Plotten Sie die Folge der Iterierten jeweils wieder in einen Höhenlinienplot von k und plotten Sie auch den Verlauf der Funktionswerte der jeweils aktuell besten Iterierten im Laufe der Iteration.

Welchen Unterschied zum Mutations-Selektions-Verfahren können Sie beobachten?

In []:

In []: