

Sprawozdanie - Metody numeryczne i optymalizacja

Jakub Andryszczak 259519,
Jakub Żak 244255,
Maciej Cierpisz 249163

Spis treści

1	Zadanie nr. 1	3
2	Zadanie nr. 2	3
3	Zadanie nr. 3	3
4	Zadanie nr. 4	3
5	Zadanie nr. 5	4

1 Zadanie nr. 1

2 Zadanie nr. 2

3 Zadanie nr. 3

Rozwiąż poniższy układ równań nieliniowych:

$$\begin{cases} 2x_1 = x_2 = \exp(-x_1) \\ -x_1 + 2x_2 = \exp(-x_2) \end{cases} \quad (1)$$

dla punktu startowego $x_0 = [-55]^t$.

Poniżej kod realizujący zadanie:

```
import numpy as np
from scipy.optimize import fsolve

# Define the system of nonlinear equations
def equations(vars):
    x1, x2 = vars
    eq1 = 2 * x1 - x2 - np.exp(-x1)
    eq2 = -x1 + 2 * x2 - np.exp(-x2)
    return [eq1, eq2]

# Initial guess
x0 = [-5, -5]

# Solve the system of equations
solution = fsolve(equations, x0)

# Print the solution
print("Solution:")
print(f"x1 = {solution[0]}")
print(f"x2 = {solution[1]}")
```

Wyniki:

$$\begin{cases} x_1 = 0.5671432904097838 \\ x_2 = 0.567143290409784 \end{cases} \quad (2)$$

4 Zadanie nr. 4

Znajdź rozwiązanie minimalizujące funkcję celu $\sum_{k=1}^{10} (2 + 2k - \exp(kx_1) - \exp(kx_2))^2$ dla punktu początkowego $x_0 = [0.3, 0.4]^T$.

Poniżej kod realizujący zadanie:

```
import numpy as np
from scipy.optimize import minimize

# Define the objective function
def objective(x):
    x1, x2 = x
    total = 0
    for k in range(1, 11):
        total += (2 + 2*k - np.exp(k*x1) - np.exp(k*x2))**2
    return total

# Initial guess
x0 = [0.3, 0.4]

# Minimize the objective function
result = minimize(objective, x0)

# Print the solution
print("Solution:")
print(f"x1 = {result.x[0]}")
print(f"x2 = {result.x[1]}")
```

Wyniki:

$$\begin{cases} x_1 = 0.25782520984040996 \\ x_2 = 0.2578252098334402 \end{cases} \quad (3)$$

5 Zadanie nr. 5