DDA5002_HW5

by Xiaocao_225040374

Problem 4 code backup

In [2]:

```
import numpy as np
def equations(vars):
    x1, x2, lambda_ = vars
    eq1 = 6*x1 + 2*x2 + lambda_ * (x1 * (1 + x1**2 + x2**2)**(-0.5) + 1)
    eq2 = 2*x1 + 8*x2 + lambda_ * x2 * (1 + x1**2 + x2**2)**(-0.5)
    eq3 = lambda_ * (np.sqrt(1 + x1**2 + x2**2) + x1 - 2)
    return [eq1, eq2, eq3]
from scipy.optimize import fsolve
initial_guess = [0.0, 0.0, 0.0]
solution = fsolve(equations, initial_guess)
x1_sol, x2_sol, lambda_sol = solution
print(f"Solution: x1 = {x1_sol}, x2 = {x2_sol}, lambda = {lambda_sol}")
```

```
Solution: x1 = 0.0, x2 = 0.0, lambda = 0.0
```

Problem 5

In [3]:

```
import math
l = 10**-4 # bracketing interval length
a0, b0 = 0, 1 # initial interval

# define the objective function
def f(x):
    return math.exp(x) - 2*x
```

Bisection Method

In [8]:

```
def bisection_method(a, b, l=1e-4):
    iterations = 0
    while (b - a) > l:
        c = (a + b) / 2
        # derivative: f'(x) = e^x - 2
        if math.exp(c) - 2 > 0:
            b = c
        else:
            a = c
        iterations += 1
    return (a + b) / 2, iterations

x_bisect, it_bisect = bisection_method(a0, b0)

print("Optimal x =", round(x_bisect, 4))
print("Iterations =", it_bisect)
```

```
Optimal x = 0.6931
Iterations = 14
```

Golden Section Method

In [7]:

```
def golden_section_method(a, b, l=1e-4):
    phi = (math.sqrt(5) - 1) / 2  # the golden ratio
    iterations = 0

    # initial interior points
    x1 = b - phi*(b - a)
    x2 = a + phi*(b - a)

    while (b - a) > l:
        if f(x1) < f(x2):  # keep left side
            b = x2
            x2 = x1
            x1 = b - phi*(b - a)
        else:              # keep right side
            a = x1
            x1 = x2
            x2 = a + phi*(b - a)

        iterations += 1

    return (a + b) / 2, iterations

x_golden, it_golden = golden_section_method(a0, b0)
print("Optimal x =", round(x_golden, 4))
print("Iterations =", it_golden)
```

```
Optimal x = 0.6931
Iterations = 20
```

Output reports

According to the outputs of two methods, both methods converge to the optimal solution x* = 0.6931 while meeting the desired accuracy. The number of iterations required for bisection method is 14, while golden section method takes 20 iterations.