Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**4**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | Write a python program for solving the following non-linear equations using Newton-Raphson method correct up to 5 decimal places:   1. cosx = xex (having initial guess x0 = 1) 2. x−2sinx−3 = 0 (having initial guess x0 = 4) |
| 2 | Write a python program for solving the following non-linear equations using fixed point iterative method correct up to 3 decimal places:   1. cos x = 3x – 1 2. 2x3 – 7x2 – 6x + 1 = 0 |

Submitted On:

22 October 2024

\_\_\_\_\_\_\_\_\_\_\_\_

(Date: DD/MM/YY)

**Task 1**

Write a python program for solving the following non-linear equations using Newton-Raphson method correct up to 5 decimal places:

1. cosx = xex (having initial guess x0 = 1)
2. x−2sinx−3 = 0 (having initial guess x0 = 4)

**Solution:**

import numpy as np

def newton\_raphson(func, deriv, x0, tol=1e-5, max\_iter=100):

    x = x0

    for \_ in range(max\_iter):

        fx = func(x)

        dfx = deriv(x)

        if dfx == 0:

            return None

        x\_new = x - fx / dfx

        if abs(x\_new - x) < tol:

            return round(x\_new, 5)

        x = x\_new

    return None

def func\_a(x):

    return np.cos(x) - x \* np.exp(x)

def deriv\_a(x):

    return -np.sin(x) - (np.exp(x) + x \* np.exp(x))

def func\_b(x):

    return x - 2 \* np.sin(x) - 3

def deriv\_b(x):

    return 1 - 2 \* np.cos(x)

x0\_a = 1

x0\_b = 4

solution\_a = newton\_raphson(func\_a, deriv\_a, x0\_a)

solution\_b = newton\_raphson(func\_b, deriv\_b, x0\_b)

print(f"Solution for cos(x) = x \* e^x: {solution\_a}")

print(f"Solution for x - 2sin(x) - 3 = 0: {solution\_b}")

**Output:**

****

**Task 2**

Write a python program for solving the following non-linear equations using fixed point iterative method correct up to 3 decimal places:

1. cos x = 3x – 1
2. 2x3 – 7x2 – 6x + 1 = 0

**Solution:**

import math

def fixed\_point\_iter(g, x0, tol=1e-5, max\_iter=100):

    x = x0

    for i in range(max\_iter):

        x\_new = g(x)

        if abs(x\_new) > 100:

            print("Value too large, stopping iteration.")

            return None

        if abs(x\_new - x) < tol:

            return round(x\_new, 3)

        x = x\_new

    print("Max iterations reached. No solution found.")

    return None

def g\_a(x):

    return (math.cos(x) + 1) / 3

def g\_b(x):

    return (7 \* x\*\*2 + 6 \* x - 1) / 2

x0\_a = 0.5

x0\_b = 3

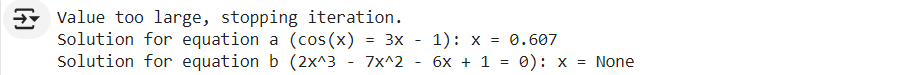
solution\_a = fixed\_point\_iter(g\_a, x0\_a)

solution\_b = fixed\_point\_iter(g\_b, x0\_b)

print(f"Solution for equation a (cos(x) = 3x - 1): x = {solution\_a}")

print(f"Solution for equation b (2x^3 - 7x^2 - 6x + 1 = 0): x = {solution\_b}")

**Output:**

****