# Bahria University,

## Karachi Campus



LAB EXPERIMENT NO.

**\_04\_**

LIST OF TASKS

|  |  |
| --- | --- |
| **TASK NO** | **OBJECTIVE** |
| 01 | Write a python program for solving the following non-linear equations using NewtonRaphson method correct up to 5 decimal places:  a. cosx = xex (having initial guess x0 = 1)  b. x−2sinx−3 = 0 (having initial guess x0 = 4) |
| 02 | Write a python program for solving the following non-linear equations using fixed point iterative method correct up to 3 decimal places:  a. cos x = 3x – 1  b. 2x3 – 7x2 – 6x + 1 = 0 |

Submitted On:

Date: 15/10/2024

**Task No 01:**

Write a python program for solving the following non-linear equations using NewtonRaphson 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:**

1. **cosx = xex (having initial guess x0 = 1)**

import math

def f(x):

    return math.cos(x) - x \* math.exp(x)

def f\_prime(x):

    return -math.sin(x) - math.exp(x) \* (x + 1)

def newton\_raphson(x0, tolerance=1e-5, max\_iterations=100):

    x = x0

    for i in range(max\_iterations):

        x\_new = x - f(x) / f\_prime(x)

        if abs(x\_new - x) < tolerance:

            return round(x\_new, 5)

        x = x\_new

    return None

x0 = 1

solution = newton\_raphson(x0)

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

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

import math

def f(x):

    return x - 2 \* math.sin(x) - 3

def f\_prime(x):

    return 1 - 2 \* math.cos(x)

def newton\_raphson(x0, tolerance=1e-5, max\_iterations=100):

    x = x0

    for i in range(max\_iterations):

        x\_new = x - f(x) / f\_prime(x)

        if abs(x\_new - x) < tolerance:

            return round(x\_new, 5)

        x = x\_new

    return None

x0 = 4

solution = newton\_raphson(x0)

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

**Output:**

**A close up of symbols

Description automatically generated**

**Task No 02:**

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:**

**a. \( \cos x = 3x - 1 \)**

import math

def g(x):

    return (math.cos(x) + 1) / 3

def fixed\_point\_iteration(x0, tolerance=1e-3, max\_iterations=100):

    x = x0

    for i in range(max\_iterations):

        x\_new = g(x)

        if abs(x\_new - x) < tolerance:

            return round(x\_new, 3)

        x = x\_new

    return None

x0 = 0.5

solution = fixed\_point\_iteration(x0)

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

1. **\( 2x^3 - 7x^2 - 6x + 1 = 0 \)**

def g(x):

    return ((7 \* x\*\*2 + 6 \* x - 1) / 2)\*\*(1/3)

def fixed\_point\_iteration(x0, tolerance=1e-3, max\_iterations=100):

    x = x0

    for i in range(max\_iterations):

        x\_new = g(x)

        if abs(x\_new - x) < tolerance:

            return round(x\_new, 3)

        x = x\_new

    return None

x0 = 1

solution = fixed\_point\_iteration(x0)

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

**Output:**

****