Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**3**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | Write a python program for approximating the roots of the following functions using the bisection method:   1. x3−9x+1 starting with the interval [2, 4] 2. 3x=√(1 + sin x)​ starting with the interval [0, 1] |
| 2 | Write a python program for approximating the roots using false-position method.   1. f(x) = (x-4)2 (x+2) starting with interval [-2.5, -1.0] 2. f(x) = ex (3.2sin(x) - 0.5cos(x)) starting with the interval [3,4] |

Submitted On:

8 October 2024

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

(Date: DD/MM/YY)

**Task 1**

**Write a python program for approximating the roots of the following functions using the bisection method:**

1. **x3−9x+1 starting with the interval [2, 4]**
2. **3x=√(1 + sin x)​ starting with the interval [0, 1]**

**Solution:**

import math

def bisection\_method(func, a, b, tol=1e-6, max\_iter=100):

    for i in range(max\_iter):

        c = (a + b) / 2  # Midpoint

        if abs(func(c)) < tol or (b - a) / 2 < tol:

            return c  # Root found

        elif func(c) \* func(a) < 0:

            b = c  # Narrowing down the interval

        else:

            a = c

    return (a + b) / 2  # Return midpoint as an approximation of the root

# Function a: x^3 - 9x + 1

def func\_a(x):

    return x\*\*3 - 9\*x + 1

# Function b: 3x = sqrt(1 + sin(x)) => f(x) = 3x - sqrt(1 + sin(x))

def func\_b(x):

    return 3\*x - math.sqrt(1 + math.sin(x))

# Approximating the root for function a

a\_root = bisection\_method(func\_a, 2, 4)

print(f"Root of x^3 - 9x + 1 in the interval [2, 4]: {round(a\_root,3)}")

# Approximating the root for function b

b\_root = bisection\_method(func\_b, 0, 1)

print(f"Root of 3x = sqrt(1 + sin(x)) in the interval [0, 1]: {round(b\_root,3)}")

**Output:**

****

**Task 2**

**Write a python program for approximating the roots using false-position method.**

1. **f(x) = (x-4)2 (x+2) starting with interval [-2.5, -1.0]**
2. **f(x) = ex (3.2sin(x) - 0.5cos(x)) starting with the interval [3,4]**

**Solution:**

import math

def false\_position\_method(func, a, b, tol=1e-6, max\_iter=100):

    if func(a) \* func(b) >= 0:

        raise ValueError("Function values at the endpoints must have opposite signs.")

    for i in range(max\_iter):

        # Calculate the root using the false position formula

        c = (a \* func(b) - b \* func(a)) / (func(b) - func(a))

        if abs(func(c)) < tol or abs(b - a) < tol:

            return c  # Root found

        elif func(c) \* func(a) < 0:

            b = c  # Narrow down the interval

        else:

            a = c

    return (a + b) / 2  # Return midpoint as an approximation of the root

# Function a: f(x) = (x - 4)^2 \* (x + 2)

def func\_a(x):

    return (x - 4)\*\*2 \* (x + 2)

# Function b: f(x) = e^x \* (3.2\*sin(x) - 0.5\*cos(x))

def func\_b(x):

    return math.exp(x) \* (3.2 \* math.sin(x) - 0.5 \* math.cos(x))

# Approximating the root for function a

a\_root = false\_position\_method(func\_a, -2.5, -1.0)

print(f"Root of f(x) = (x - 4)^2 \* (x + 2) in the interval [-2.5, -1.0]: {a\_root}")

# Approximating the root for function b

b\_root = false\_position\_method(func\_b, 3, 4)

print(f"Root of f(x) = e^x \* (3.2\*sin(x) - 0.5\*cos(x)) in the interval [3, 4]: {b\_root}")

**Output:**

****