Міністерство освіти і науки

Національний університет „Львівська політехніка”



**Звіт**

з лабораторної роботи №2

з дисципліни: “ Чисельні методи”

Виконав:

Ст. гр. ІР-25

Баланик Б. В.

**Львів**

**2023**

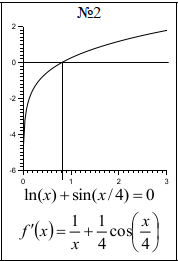
# Мета роботи

Вивчити основні методи уточнення коренів нелінійних рівнянь з одним невідомим.

# Завдання до лабораторної роботи.

**Варіант 2 Група 1**

Метод поділу ділянки навпіл з пошуком ділянки локалізації x є [0,1; 3]



# Короткі теоретичні відомості, що необхідні для виконання лабораторної роботи.

Метод поділу ділянки навпіл — це ітераційний метод розв'язання рівнянь, який полягає у послідовному звуженні діапазону значень, в якому знаходиться корінь, до тих пір, поки його довжина не стане менше заданої точності.

# Блок-схема розробленої програми.

Загальний алгоритм методу поділу проміжку навпіл з пошуком ділянки локалізації:

1. На початку алгоритму задаємо значення величин a, h, ε.

Встановлюємо максимальне значення b\_max = 3.

2. b = a + h

fa = f(a)

fb = f(b)

3. Пошук ділянки локалізації:

ПОКИ (fa \* fb > 0) І (b <= b\_max):

a = b

b = a + h

fa = f(a)

fb = f(b)

4. Якщо b > b\_max:

ВИВІД "Корінь на ділянці [0.1, 3] не знайдений."

ЗАВЕРШЕННЯ РОБОТИ

5. Ітераційний процес:

ПОКИ |b - a| > ε:

x = (b + a) / 2

fx = f(x)

Якщо fx \* fa > 0:

a = x

fa = fx

ІНАКШЕ:

b = x

fb = fx

6. ВИВІД (a + b) / 2

7. Перевірка вірності роботи алгоритму:

ЯКЩО |f(root)| < ε:

вивести "Робота алгоритму вірна"

ІНАКШЕ:

вивести "Робота алгоритму не вірна"

# Список ідентифікаторів констант, змінних, функцій, методів, використаних у програмі, та їх пояснення.

* **a**: Початкова точка діапазону.
* **h**: Крок для пошуку ділянки локалізації.
* **epsilon**: Точність рішення.
* **f(x)**: Функція, корінь якої потрібно знайти.

Остаточна версія програми.

import math

def f(x):

    return math.log(x) + math.sin(x/4)

def bisect\_with\_localization(a, h, b\_max, epsilon):

    b = a + h

    fa = f(a)

    fb = f(b)

    # Пошук ділянки локалізації

    while fa \* fb > 0 and b <= b\_max:

        a = b

        b = a + h

        fa = f(a)

        fb = f(b)

    # Якщо межа b перевищує b\_max, то корінь на даному діапазоні не знайдений

    if b > b\_max:

        return None

    # Ітераційний процес

    while abs(b - a) > epsilon:

        x = (b + a) / 2

        fx = f(x)

        if fx \* fa > 0:

            a = x

            fa = fx

        else:

            b = x

            fb = fx

    return (a + b) / 2

# Задані параметри

a = 0.1

h = 0.1

b\_max = 3

epsilon = 1e-6

root = bisect\_with\_localization(a, h, b\_max, epsilon)

if root:

    print(f"Корінь рівняння на ділянці [{a}, {b\_max}] дорівнює: {root}")

else:

    print(f"Корінь на ділянці [{a}, {b\_max}] не знайдений.")

# Перевірка вірності роботи алгоритму

if abs(f(root)) < epsilon:

    print("Робота алгоритму вірна")

else:

    print("Робота алгоритму не вірна")

Результати виконання програми.

**Корінь рівняння на ділянці [0.1, 3] дорівнює: 0.8165119171142579**

**Робота алгоритму вірна**

Висновки.

Метод поділу ділянки навпіл ефективно дозволяє знаходити корені нелінійних рівнянь із заданою точністю. Для оптимальної роботи методу необхідно правильно обрати початковий діапазон та крок для пошуку ділянки локалізації кореня.