

Міністерство освіти і науки, молоді та спорту України  
Національний університет «Львівська політехніка»  
Інститут комп'ютерних наук та інформаційних технологій  
Кафедра програмного забезпечення

## **ЗВІТ**

### **Про виконання лабораторної роботи № 2**

«Розв'язування нелінійних рівнянь методом дихотомії та методом хорд»  
з дисципліни «Чисельні методи»

**Лектор:**

Мельник Н.Б.

**Виконав:**

студент групи ПЗ-11

Солтисюк Д.А.

**Прийняла:**

Мельник Н.Б.

«\_\_\_» \_\_\_\_\_ 2022 р.

$\Sigma$  = \_\_\_\_\_

**Тема роботи:** Розв'язування нелінійних рівнянь методом дотичних та методом послідовних наближень

**Мета роботи:** ознайомлення на практиці з методом дотичних та методом послідовних наближень для розв'язування нелінійних рівнянь.

### Теоретичні відомості

#### Метод Ньютона (метод дотичних)

1. Запишемо рівняння дотичної до кривої  $y=f(x)$  в точці  $(x_i; f(x_i))$   
 $y-f(x_i)=f'(x_i)(x-x_i)$  і визначимо  $x$
2. Ітераційні формули запишемо у вигляді  
 $x_{i+1}=x_i-(f(x_i)/f'(x_i))$ ,  $i=0,1,2,\dots$
3. Для вибору початкового наближення кореня рівняння  $f(x)=0$  необхідно керуватися таким правилом: за початкову точку слід вибрати той кінець відрізка  $[a,b]$ , в якому знак функції  $y=f(x)$  співпадає зі знаком її другої похідної  $f''(x)$
4. Ітераційний процес продовжують до тих пір, поки не виконуватиметься умова  
 $|x_i-x_{i-1}| \leq \epsilon$

#### Метод простої ітерації

1. Розглянемо нелінійне рівняння  $f(x) = 0$ , де  $f(x)$  є неперервною функцією. Потрібно знайти хоча б один дійсний корінь цього рівняння. Рівняння  $f(x) = 0$  запишемо у канонічній формі  $x=f(x)$
2. Довільним способом визначимо наближене значення  $x_0$  кореня рівняння і підставимо його в праву частину співвідношення  $x=f(x)$ . У результаті отримаємо  $x_1=f(x_0)$
3. Повторюючи цей процес, отримаємо ітераційні формули  
 $x_i=f(x_{i-1})$ ,  $i=1,2,3,\dots$
4. Ітераційний процес продовжують до тих пір, поки не виконуватиметься умова  
 $|x_i-x_{i-1}| \leq \epsilon$

### Індивідуальне завдання

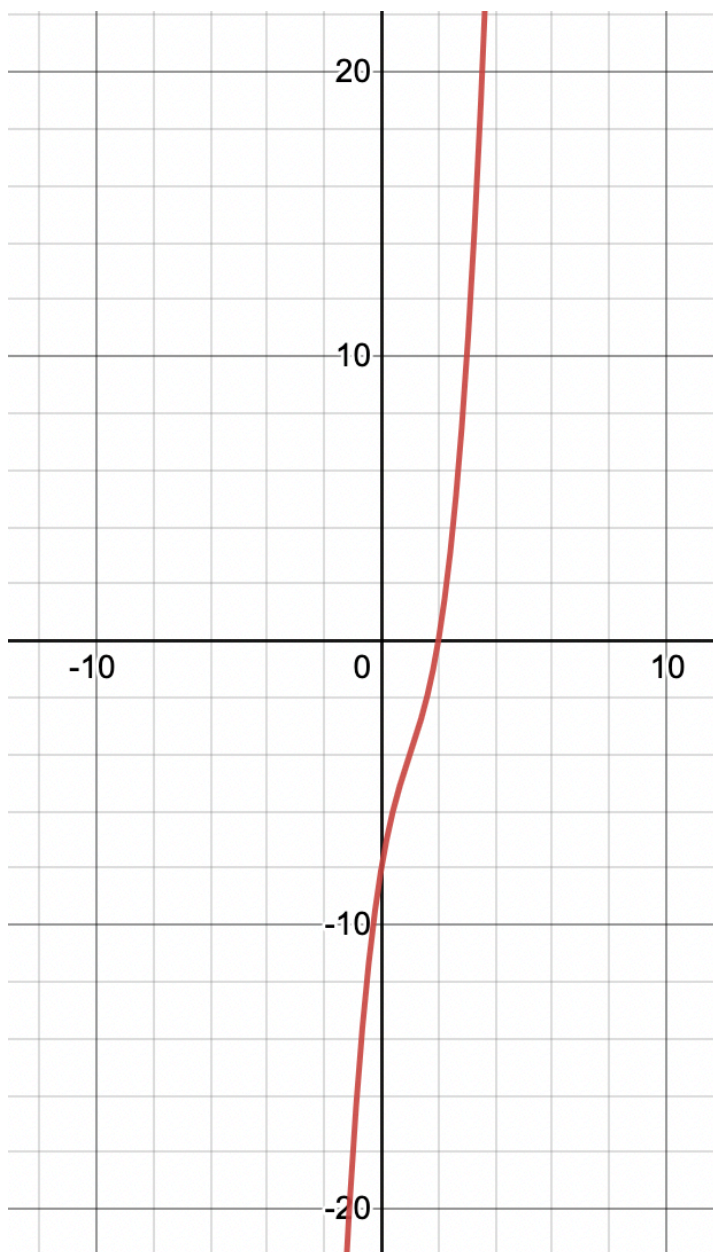
#### Варіант: 24

Відокремити дійсні корені рівняння графічним та аналітичним способами і скласти програму його розв'язування методом дотичних та ітерацій.

Функція:  $x^3 - 3x^2 + 6x - 8 = 0$

## Графічний метод

Графік функції:



Графік перетинає вісь абсцис у точці (2;0)

## Аналітичний метод

- 1) Область допустимих значень:  $D(x) = (-\infty; \infty)$
- 2) Досліджуємо функцію  $y = x^3 - 3x^2 + 6x - 8$  на монотонність та неперервність:  
 $f'(x) = 3x^2 - 6x + 6$   
 $3x^2 - 6x + 6 = 0$   
 $x \notin \mathbb{R}$

Інтервали монотонності функції  $f(x)$  є такими:

$(-\infty; \infty)$

Отже, єдиний дійсний корінь даного рівняння належить відріzkу  $[-10; 10]$ .

### Хід роботи

#### Код програми:

```
#!/usr/bin/env python3
from sympy import Pow, lambdify, symbols

x = symbols("x")

def whatever(iterator, x0, eps):
    current_argument = x0

    i = 1
    while True:
        next_argument = iterator(current_argument)

        if abs(next_argument - current_argument) < eps:
            break

        current_argument = next_argument
        i += 1

    return (current_argument, i)

if __name__ == "__main__":
    a_default = str(0)
    b_default = str(4)
    eps_default = str(0.0001)

    a = float(
        input(f'Please, enter point A value (default: {a_default}): ') or a_default
    )
    b = float(input(f'Now, enter point B value (default: {b_default}): ') or b_default)
    eps = float(
        input(f'It's time for EPS value to roll in (default: {eps_default}): ')
        or eps_default
    )

    y = Pow(x, 3) - 3 * Pow(x, 2) + 6 * x - 8
    y_prime = y.diff(x)
    y_double_prime = y_prime.diff(x)
```

```

z = (Pow(x, 3) - 3 * Pow(x, 2) - 8) / -6

f = lambdify(x, y)
f_prime = lambdify(x, y_prime)
f_double_prime = lambdify(x, y_double_prime)

g = lambdify(x, z)

def newton_iterator(xi):
    return xi - (f(xi) / f_prime(xi))

def simple_iterator(xi):
    return g(xi)

def select_boundpoint():
    if f(a) * f_double_prime(a) > 0:
        return a

    if f(b) * f_double_prime(b) > 0:
        return b

    print("Provided boundaries are not applicable")
    quit()

x0 = select_boundpoint()

print(
    f"""
Results

Newtons method: {whatever(newton_iterator, x0, eps)}
Simple itertaions method: {whatever(simple_iterator, x0, eps)}
"""
)

```

## Результати

```

> ./main.py
Please, enter point A value (default: 0):
Now, enter point B value (default: 4):
Its time for EPS value to roll in (default: 0.0001):

Results

Newtons method: (2.00000000328866285, 6)
Simple itertaions method: (mpf('1.9999013952209506'), 5)

```

### **Висновки**

Виконуючи лабораторну роботу, я навчився використовувати метод дотичних та метод простих ітерацій для визначення коренів нелінійних рівнянь.