Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут комп’ютерних наук та інформаційних технологій
Кафедра програмного забезпечення

Звіт

Про виконання лабораторної роботи №1

На тему:

«Розв’язування нелінійних рівнянь
методом бісекцій та методом хорд»

з дисципліни «Чисельні методи»

Лектор:
доцент каф. ПЗ
Мельник Н. Б.
Виконав:
ст. гр. ПЗ-11
Солтисюк Д.А.
Прийняла:
доцент каф. ПЗ
Мельник Н. Б.

« __ » ________ 2022 р.

Σ = _____ _________ .

Львів – 2022

Тема: Розв’язування нелінійних рівнянь методом бісекцій та методом хорд
Мета: Навчитись розв’язувати нелінійні рівняння методами бісекцій та хорд

ТЕОРЕТИЧНІ ВІДОМОСТІ

Локалізація коренів – це визначення інтервалів функції,
що містять єдиний корінь

Метод бісекцій (метод дихотомії) – метод поділу відрізка навпіл.
Розглянемо рівняння f (x) = 0, де f (x) – неперервна монотонна нелінійна функція. Необхідно знайти розв’язок цього рівняння на проміжку [a,b] з точністю ε. Знайдемо середину відрізка [a,b] за формулою x = (a + b) / 2.
Якщо f(x) = 0, то шуканий корінь – середина відрізка [a,b]. Якщо f (a)*f (x) < 0 , то змінюємо праву межу b = x , у протилежному випадку змінюємо ліву межу a = x . Процес поділу відрізка [a,b] навпіл продовжуємо доти, поки не виконається умова | b – a | < ε .
Кількість ітерацій, необхідних для досягнення заданої точності обчислень, визначається за формулою $n = \left[\log_2 \frac{b-a}{\varepsilon} - 1 \right]$, де [ε]– ціла частина числа ε.

Метод хорд (метод пропорційних частин).
Розглянемо рівняння f (x) = 0, де f (x) – неперервна монотонна нелінійна функція. Необхідно знайти розв’язок цього рівняння на проміжку [a,b] з точністю ε. Суть методу хорд полягає в тому, що на відрізку [a,b] малої довжини дугу функції f (x) замінюють хордою ab , яка її стягує. За наближене значення кореня приймають абсцису точки перетину хорди з віссю Ox.
Запишемо рівняння хорди, яка проходить через точки (a; f (a)) і (b; f (b)) у вигляді: $\frac{y - f(x)}{f(b) - f(a)} = \frac{x-a}{b-a}$.
Для довільного (i +1)-го наближення точного значення кореня ∗ x для заданого рівняння використовують формулу $x_{i+1} = x_i - \frac{f(x_i)(b - x_i)}{f(b) - f(x_i)}, \quad i = 0,1,2,\ldots$
Дугу кривої стягують хордою доти, поки виконується умова $| x_{i+1} – x_i | < ε$.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Варіант 24

Завдання 1
Написати програму розв’язку нелінійного рівняння відповідно до варіанту методом бісекцій
Завдання 2
Написати програму розв’язку нелінійного рівняння відповідно до варіанту методом хорд

ХІД РОБОТИ

$x^3 - 4x^2 - 9x + 2 = 0$ - задана згідно індивідуального варіанту неперервна монотонна функція. Єдиний корінь знаходиться на проміжку [4, 5].

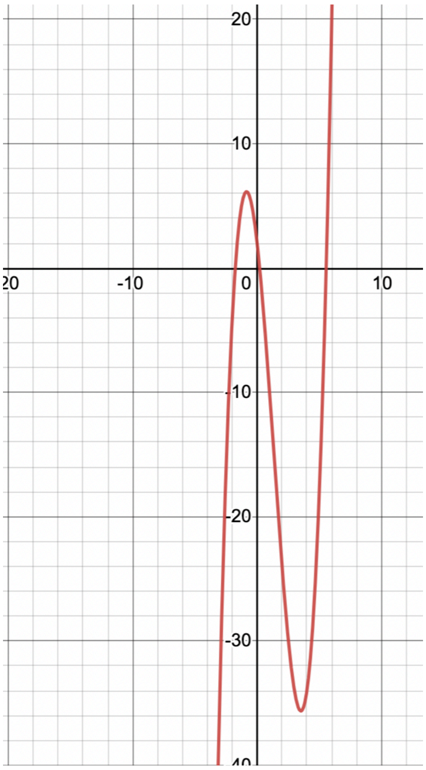

Рис. 1. Графік даної функції

Код функцій програми:

```
fn bisection<F: Fn(f64) -> f64>(f: F, interval: (f64, f64), eps: f64) -> (f64, i32) {
    fn calculate_argument(a: f64, b: f64) -> f64 {
        (a + b) / 2.0
    }

    let (mut a, mut b) = interval;

    let mut x = calculate_argument(a, b);

    let mut n = 1;
    loop {
        {
            if (f(a) * f(x)).is_sign_negative() {
                b = x;
            } else {
                a = x
            };

            x = calculate_argument(a, b);
        }

        if f(x) == 0.0 {
            break;
        }

        if (b - a).abs() < eps {
```

```
            break;
        }

        n += 1;
    }

    (x, n)
}

fn secant<F: Fn(f64) -> f64>(f: &F, interval: (f64, f64), eps: f64) -> (f64, i32) {
    fn calculate_argument<F: Fn(f64) -> f64>(f: F, a: f64, b: f64) -> f64 {
        a - (f(a) * (b - a)) / (f(b) - f(a))
    }

    let (mut a, b) = interval;

    let mut current = calculate_argument(f, a, b);
    let mut previous = a;

    let mut n = 1;
    loop {
        if (current - previous).abs() < eps {
            break;
        }

        {
            a = current;
            previous = current;
            current = calculate_argument(f, a, b);
        }

        n += 1;
    }

    (current, n)
}

fn get_eps_input() -> f64 {
    println!("Enter eps value (should be above zero): ");

    let mut buf = String::new();

    std::io::stdin()
        .read_line(&mut buf)
        .ok()
        .expect("Failed reading eps value");

    let eps = buf.trim().parse::<f64>().unwrap();

    if eps <= 0.0 {
        eprintln!("\nInvalid eps value");

        std::process::exit(1);
    }

    eps
}

fn main() {
    let a = (4.0 - 43.0_f64.sqrt()) / 3.0;
    let b = (4.0 + 43.0_f64.sqrt()) / 3.0;

    fn f(x: f64) -> f64 {
```

```
        x.powi(3) - 4.0 * x.powi(2) - 9.0 * x + 2.0
    }

    let eps = get_eps_input();

    if (f(a) * f(b)) > 0.0 {
        eprintln!("No valid x in this bounds");

        std::process::exit(1);
    }

    let bisection_result = bisection(f, (a, b), eps);
    let secant_result = secant(&f, (a, b), eps);

    println!(
        "\nBisection method: {} (iterations: {})",
        bisection_result.0.to_string(),
        bisection_result.1.to_string()
    );
    println!(
        "Secant method: {} (iterations: {})",
        secant_result.0.to_string(),
        secant_result.1.to_string()
    );
}
```

```
Enter eps value (should be above zero):
0.0000000001

Bisection method: 0.2045733761781442 (iterations: 36)
Secant method: 0.20457337614850996 (iterations: 9)
```

Рис. 2. Результати виконання програми

ВИСНОВКИ

Виконуючи лабораторну роботу №1, я навчився програмувати наближені розв’язки нелінійних рівнянь методами бісекції та хорд з вказаною точністю.