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.

∑ = \_\_\_\_\_ \_\_\_\_\_\_\_\_\_ .

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 | < e .

Зображення, що містить текст, наручний годинник, вимірювальний прилад

Автоматично згенерований описКількість ітерацій, необхідних для досягнення заданої точності обчислень, визначається за формулою , 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.

Зображення, що містить текст, наручний годинник, годинник

Автоматично згенерований описЗапишемо рівняння хорди, яка проходить через точки (a; f (a)) і (b; f (b)) у вигляді:

Зображення, що містить текст

Автоматично згенерований описДля довільного (i +1)-го наближення точного значення кореня \* x для заданого рівняння використовують формулу

Дугу кривої стягують хордою доти, поки виконується умова | xi+1 – xi | < ε.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

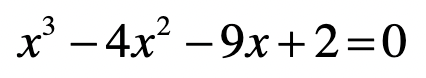
**Варіант 24**

**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**

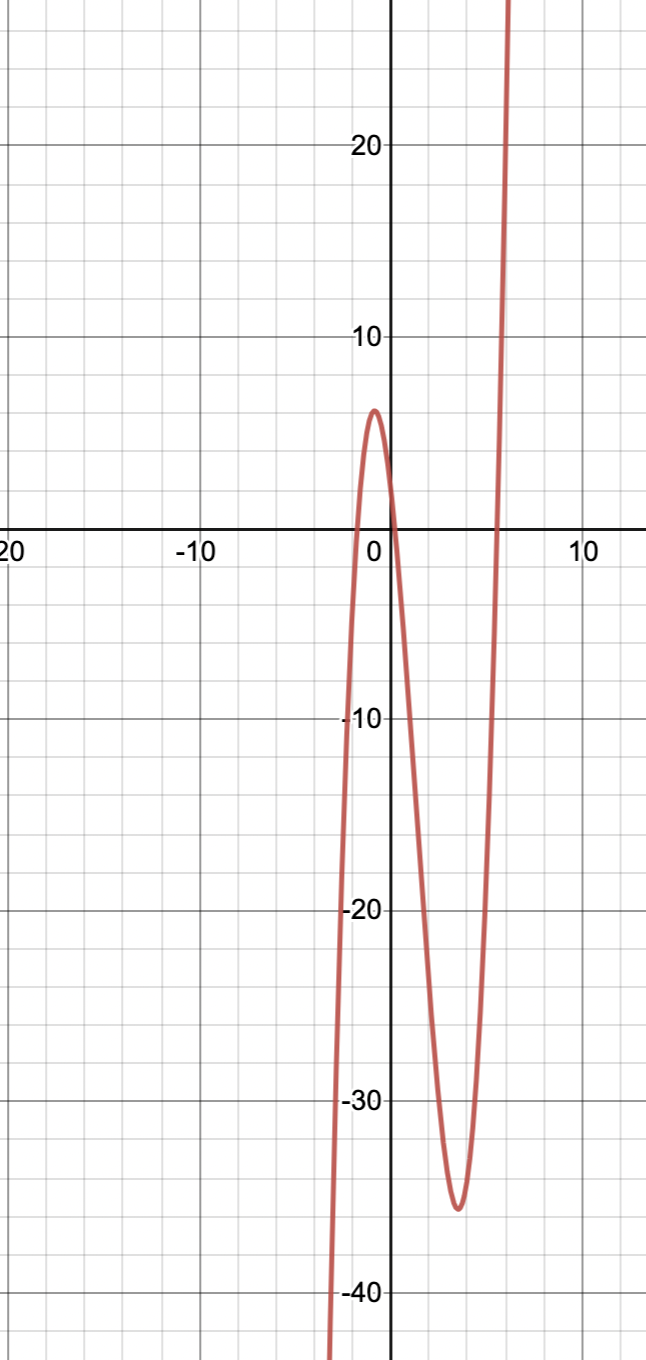
- задана згідно індивідуального варіанту неперервна монотонна функція. Єдиний корінь знаходиться на проміжку [4, 5].

Рис. 1. Графік даної функції

**Код функцій програми:**

use text\_io::read;

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 main() {

fn f(x: f64) -> f64 {

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

}

println!("\nEnter a value: ");

let a: f64 = read!();

println!("\nEnter b value: ");

let b: f64 = read!();

println!("Enter eps value: ");

let eps: f64 = read!();

if (f(a) \* f(b)) > 0.0 {

eprintln!("\nNo 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!(

"\n\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()

);

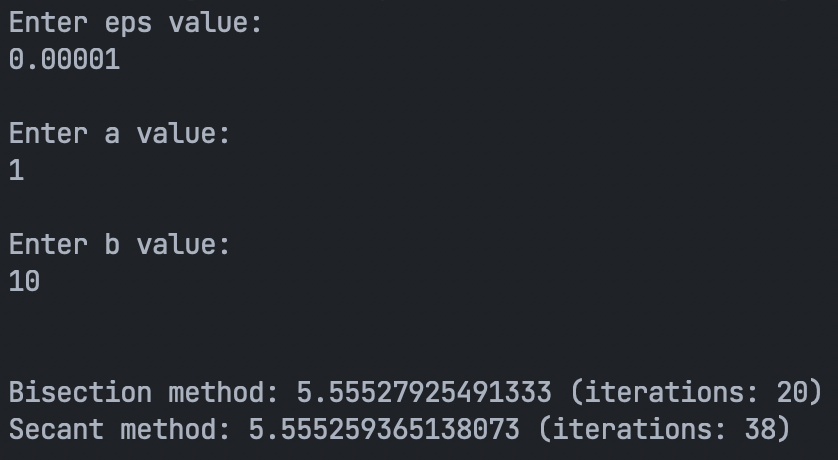
}

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.