Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра «Комп’ютеризовані системи автоматики»



ЗВІТ

про виконання лабораторної роботи №9

«ФУНКЦІЇ. ІТЕРАЦІЙНІ ПРОЦЕСИ (RUST)»

із навчальної дисципліни: «Алгоритми та структури даних»

Варіант №11

Виконав:

студент групи ІР-24

Кіндрат Віктор Романович

Прийняв:

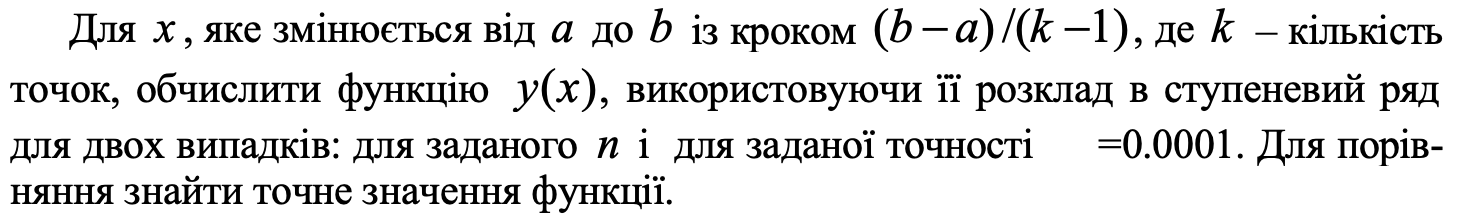
к.т.н., доцент

Сиротюк С. В.

Львів 2024

**Мета:** Формування уміння визначати й використовувати функції для організації ітераційних процесів

**Завдання:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Функція** | **Діапазон зміни ар- гументу** | **n** | **Сума** |
| 11 | 2  *y*  (1  2*x*2 )*ex* | 0  *x*  1 | 5 | *S*  1  3*x* 2  ....  2*n*  1 *x* 2*n* , *n*  0  *n*! |

**Програмний код мовою Rust:**

use std::f64::consts::E;

// Function to calculate the k-th term of the series

fn term(x: f64, k: u32) -> f64 {

(2 \* k + 1) as f64 \* x.powi(2 \* k as i32) / (factorial(k) as f64)

}

// Function to calculate factorial (helper for term calculation)

fn factorial(n: u32) -> u64 {

(1..=n as u64).product()

}

// Function to calculate the series sum with a fixed number of terms n

fn sum1(x: f64, n: u32) -> f64 {

let mut z = 0.0;

for k in 0..=n {

z += term(x, k);

}

z

}

// Function to calculate the series sum with a specified precision eps

fn sum2(x: f64, eps: f64) -> f64 {

let mut z = 0.0;

let mut ak: f64 = 1.0; // Ensure `ak` is of type f64

let mut k = 0;

while ak.abs() > eps {

ak = term(x, k);

z += ak;

k += 1;

}

z

}

// Function to calculate the exact value of y(x)

fn y(x: f64) -> f64 {

(1.0 + 2.0 \* x \* x) \* E.powf(x)

}

fn main() {

let x1 = 0.0;

let x2 = 1.0; // Range for x

let k = 10; // Number of points

// Take input for n

println!("Enter the number of terms n:");

let mut input\_n = String::new();

std::io::stdin().read\_line(&mut input\_n).unwrap();

let n: u32 = input\_n.trim().parse::<u32>().expect("Failed to parse input for n");

// Take input for delta

println!("Enter the accuracy delta:");

let mut input\_delta = String::new();

std::io::stdin().read\_line(&mut input\_delta).unwrap();

let delta: f64 = input\_delta.trim().parse::<f64>().expect("Failed to parse input for delta");

// Print table header

println!("{:>10} {:>15} {:>15} {:>15}", "x", "Sn (Fixed n)", "Se (Accuracy)", "y(x)");

println!("{}", "-".repeat(55));

// Loop to calculate and print values for each x

for j in 1..=k {

let x = x1 + (x2 - x1) \* (j as f64 - 1.0) / 9.0;

let sn = sum1(x, n); // Sum with fixed n terms

let se = sum2(x, delta); // Sum with specified accuracy

let exact = y(x); // Exact value of the function

// Print values in table format

println!("{:>10.5} {:>15.5} {:>15.5} {:>15.5}", x, sn, se, exact);

}

}

**Результат виконання програми:**

~/Workspace/NULP/algos\_part\_3/labs/lab9/main git:[main]

./main

Enter the number of terms n:

5

Enter the accuracy delta:

0.00005

x Sn (Fixed n) Se (Accuracy) y(x)

-------------------------------------------------------

0.00000 1.00000 1.00000 1.00000

0.11111 1.03742 1.03742 1.14511

0.22222 1.15439 1.15439 1.37219

0.33333 1.36586 1.36586 1.70575

0.44444 1.69973 1.69973 2.17577

0.55556 2.20204 2.20205 2.81878

0.66667 2.94581 2.94595 3.67905

0.77778 4.04561 4.04659 4.81008

0.88889 5.68097 5.68601 6.27626

1.00000 8.13333 8.15484 8.15485

**Висновок:**

Під час виконання лабораторної роботи сформував уміння визначати й використовувати функції для організації ітераційних процесів, засвоїв на практиці навички роботи із методами введення/виведення, роботи з циклами for та while мовою програмування Rust, ознайомився із методами бібліотек.

Після виконання програмного коду виводиться очікуваний результат. У таблиці отримуємо порівняння методів обчислення (з апроксимацією та при фіксованому n)