

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
Факультет прикладної математики  
Кафедра прикладної математики

Звіт  
із лабораторної роботи №3  
із дисципліни «Розподілені і хмарні обчислення»

Виконав:

студент групи КМ-01

*Романецький М.С.*

Керівник:

*Доцент кафедри ПМА*

*Ліскін В. О.*

Київ — 2023

Мета роботи: Проінтегрувати методом редукції. Проінтегрувати методом Сімпсона за різної кількості потоків та інтервалів розбиття.

$$\int_1^2 \frac{e^x - 1}{e^x + 1} dx$$

Опис програми: Бібліотека ‘Rayon’ буде використовуватись для паралелізму. Для порівняння часу роботи буде використовувати ‘Criterion’

Інтервали розбиття	Потоки	Середній час виконання
10	1	3.8007 мкс
10	2	6.4917 мкс
10	4	2.7270 мкс
10	8	4.2251 мкс
$10^3$	1	257.71 мкс
$10^3$	2	154.58 мкс
$10^3$	4	38.784 мкс
$10^3$	8	34.539 мкс
$10^6$	1	285.08 мс
$10^6$	2	146.75 мс
$10^6$	4	46.162 мс
$10^6$	8	18.695 мс

```
Task 1:
f = (e^x-1) / (e^x+1), a=1, b=2, steps=100
Result: 0.6273326470577413
```

Завдання 1

```
Запускаємо cargo test
  Finished test [unoptimized + debuginfo] target(s) in 0.71s
  Running unittests src\lib.rs (target\debug\deps\lab_3-3d619a89cf0197c1.exe)

running 1 test
test integration::tests::test_example_sample ... ok

test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

  Running unittests src\main.rs (target\debug\deps\lab_3-5998dad825358018.exe)

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

  Doc-tests lab_3

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

cargo test виконано успішно
```

Тестування



Інтерфейс

Код програми:

```
// cd /d/KPI/Distributed_computing/Labs/Lab_3

use std::{sync::Arc, f64::consts::E};
use lab_3::integration::integral_reduction;
use eframe::{epi, egui::{self, CtxRef}};
use std::process::Command;
use egui::vec2;

struct MyApp {
    // дані та стан програми
}

impl Default for MyApp {
    fn default() -> Self {
        Self {
            // Ініціалізація стану
        }
    }
}

impl epi::App for MyApp {
    fn name(&self) -> &str {
        "Romanetskiy | Lab 3"
    }

    fn update(&mut self, ctx: &CtxRef, _frame: &mut epi::Frame) {
        egui::CentralPanel::default().show(ctx, |ui| {
            let button_size = vec2(ui.available_width(), 95.0);
```

```

        if ui.add_sized(button_size, egui::Button::new("Run Task
1")).clicked() {
            task_1()
        }
        if ui.add_sized(button_size, egui::Button::new("Run Task
2")).clicked() {
            task_2()
        }
        if ui.add_sized(button_size, egui::Button::new("Run Test")).clicked()
{
            testing()
        }
    });
}
}

fn task_1() {
    println!("\nTask 1:");
    let f = |x: f64| (E.powf(x) - 1.0) / (E.powf(x) + 1.0);
    let a = 1.0;
    let b = 2.0;
    let steps = 100;
    println!("f = (e^x-1) / (e^x+1), a={}, b={}, steps={}", a, b, steps);

    let af = Arc::new(f);

    let res = integral_reduction(&af, a, b, steps);
    println!("Result: {}", res);
}

fn task_2() {
    println!("\nTask 2:");
    println!("Запускаємо cargo bench");
    match Command::new("cargo").args(&["bench"]).status() {
        Ok(status) => if status.success() {
            println!("cargo bench виконано успішно");
        } else {
            eprintln!("cargo bench завершилося з помилкою");
        },
        Err(e) => eprintln!("Помилка при запуску cargo bench: {}", e),
    }
}

fn testing() {
    println!("\nЗапускаємо cargo test");
    match Command::new("cargo").args(&["test"]).status() {
        Ok(status) => if status.success() {
            println!("cargo test виконано успішно");
        } else {

```

```
        eprintln!("cargo test завершилось с ошибкой");
    },
    Err(e) => eprintln!("Ошибка при запуске cargo test: {}", e),
}

fn main() {
    let app = MyApp::default();
    let mut native_options = eframe::NativeOptions::default();
    native_options.initial_window_size = Some(egui::vec2(400.0, 300.0));
    eframe::run_native(Box::new(app), native_options);
}
```