

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

Звіт  
із лабораторної роботи №4  
із дисципліни «Розподілені і хмарні обчислення»

Виконав:  
студент групи КМ-01  
Скорденко Д. О.

Керівник:  
доцент кафедри ПМА  
Ліскін В. О.

## ЗМІСТ

Вступ .....	3
1 Основна частина .....	4
2 Опис програми [Тестовий приклад] .....	5
Висновки .....	7
Додаток Код лістинги .....	8

## ВСТУП

**Мета:** розпаралелити метод Гаусса для вирішення СЛАР.

Дослідний приклад:

$$A = \begin{bmatrix} 8 & 7 & 3 \\ -7 & -4 & -4 \\ -6 & -5 & -4 \end{bmatrix} \quad b = \begin{bmatrix} 18 \\ -11 \\ -15 \end{bmatrix}$$

Рішення:

$$X = \begin{bmatrix} 5 \\ -1 \\ -5 \end{bmatrix}$$

## 1 ОСНОВНА ЧАСТИНА

**Опис програми:** Для реалізації паралелізму буде використовуватись 'Rayon'. Для порівняння швидкості обчислень буде використовуватись 'Criterion'. Для матриць / векторів буде використовуватись 'ndarray'.

Порівняння буде проведено на різних  $k$ -стях відрізків

$n \in [10, 100, 1000]$ , та при різних  $k$ -сті потоків  $nworkers \in [1, 2, 4, 8]$



Таблиця 2.1 – Порівняння швидкодії

n	nworkers	time		
10	1	10.621 $\mu$ s	10.930 $\mu$ s	11.370 $\mu$ s
10	2	17.354 $\mu$ s	17.776 $\mu$ s	18.270 $\mu$ s
10	4	32.124 $\mu$ s	32.884 $\mu$ s	33.736 $\mu$ s
10	8	54.917 $\mu$ s	55.216 $\mu$ s	55.558 $\mu$ s
100	1	1.5261 ms	1.5530 ms	1.5841 ms
100	2	1.6093 ms	1.6221 ms	1.6357 ms
100	4	1.9064 ms	1.9908 ms	2.0801 ms
100	8	2.6843 ms	2.7242 ms	2.7629 ms
1000	1	1.7285 s	1.7489 s	1.7719 s
1000	2	1.2366 s	1.2536 s	1.2761 s
1000	4	1.0471 s	1.0660 s	1.0836 s
1000	8	987.86 ms	994.92 ms	1.0031 s

## ВИСНОВКИ

На малих об'ємах обчислень збільшення к-сті потоків призводить до погіршення продуктивності.

На більших об'ємах збільшення потоків призводить до збільшення продуктивності, однак після певної к-сті потоків ефект покращення продуктивності стає незначним.

## Додаток

## Код лістинги

\*Примітка: У код лістингах при копіюванні втрачається форматування (не копіюються пробіли). Файли прикріплено до цього pdf (вкладка "прикріплені файли").

## Listing 1: lineareq.rs

```
use rayon::prelude::*;
use ndarray::prelude::*;

pub fn gauss (mut m: Array<f64, Dim<[usize; 2]>>) → Array<f64, Dim<[usize; 2]>>) {
    let (nrows, ncols) = m.dim();

    // Прямий хід
    (0..nrows).into_iter().for_each(|i| {
        (0..ncols).rev().into_iter().for_each(|j| {
            m[[i,j]] /= m[[i,i]];
        });

        let _m = m.clone();
        m
            .slice_mut(s![i+1..nrows, ..])
            .axis_iter_mut(Axis(0))
            .into_par_iter()
```



```

        .for_each(|mut row| {
            let scale = row[i];

            (i..ncols).into_iter().for_each(|k| {
                row[k] -= scale * _m[[i,k]];
            });
        });
    });

    // Зворотній xid
    let mut xx = Array::<f64, _>::zeros(nrows);
    xx[nrows - 1] = m[[nrows - 1, nrows]];

    (0..nrows-1).rev().into_iter().for_each(|i| {
        xx[i] = m[[i, nrows]];

        (i+1..nrows).into_iter().for_each(|j| {
            xx[i] -= m[[i,j]] * xx[j];
        });
    });

    xx
}

```

Listing 2: lib.rs

```
pub mod lineareq;
```

## Listing 3: main.rs

```
use lab_4::lineareq::gauss;
use ndarray::prelude::*;

fn main() {
    let m = array![
        [8., 7., 3., 18.],
        [-7., -4., -4., -11.],
        [-6., 5.0, -4.0, -15.0],
    ];

    println!("Input: {}", m);

    let m = gauss(m);
    println!("Result: {}", m);
}
```