

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи

з дисципліни «АЛГОРИТМИ І СИСТЕМИ КОМП'ЮТЕРНОЇ
МАТЕМАТИКИ 1.МАТЕМАТИЧНІ АЛГОРИТМИ»

на тему

“Розв’язання систем лінійних рівнянь”

Виконала:

студентка групи КМ-01

Резниченко Є. С.

Перевірила:

Асистент кафедри ПМА

Ковальчук-Химюк Л. О.

Зміст

Вступ.....	3
Основна частина	4
Варіант 15.....	4
Опис роботи програми.....	4
Висновки	5
Відповіді на контрольні питання	6
Перелік посилань.....	7
Додаток А.....	8
Додаток Б.....	9
Вміст файлу main.py.....	9
Вміст файлу main.m	10

Вступ

Мета роботи – вивчити правила використання програмних засобів для факторизації матриць і розв’язання систем лінійних алгебраїчних рівнянь, розв’язувати задану систему рівнянь, провести порівняльний аналіз вивчених чисельних методів розв’язання СЛР.

Основна частина

Варіант 15

матриця					вектор
10.9	1.2	2.1	1.0		-0.7
1.2	11.2	1.5	1.0		5.3
2.1	1.5	9.8	1.0		10.3
0.9	2.5	12.1	1.0		24.6

Опис роботи програми

У обидвох мовах було застосовано LU алгоритм [\[1\]](#)

Python:

Використано бібліотечну функцію `numpy.linalg.solve()` [\[3\]](#)

Octave:

Використано вбудовану функцію розв'язку СЛАР

Основний алгоритм LU-розкладу є модифікованим методом Гауса і вимагає $2n^3 / 3$ арифметичних операцій. Ось як це працює:

1. Позначимо як l_{ij} , u_{ij} , a_{ij} елементи матриць L , U та A відповідно.
2. З означення LU-розбиття $l_{ij} = 0$ ($j > i$), $u_{ij} = 0$ ($j < i$), $u_{ii} = 1$.
3. Звідси легко отримати вирази для елементів матриць L та U .

Цей метод має перевагу над методом Гауса, оскільки дозволяє більш просто отримати розв'язок для різних векторів b системи лінійних алгебраїчних рівнянь $A * x = b$.

Висновки

	Python	Octave
Час виконання (с)	0.007	0.0086839
Похибка	1.459e-28	2.019e-28
Результат	[4.38112979 5.22361945 6.23205926 -67.80998246]	[4.38112978733865; 5.22361944925462; 6.23205925893555; -67.8099824648614]

У ході розв'язку СЛАР отримали результати наведені в таблиці. Похибка і час виконання малі для обох мов програмування.

Додаток А містить скріншоти роботи програми.

Відповіді на контрольні питання

1. Система лінійних алгебраїчних рівнянь має однозначний розв'язок, коли число рівнянь в системі дорівнює числу невідомих. Це також можливо, якщо детермінант матриці системи не є нулем.
2. Факторизація матриць допомагає розв'язати систему лінійних рівнянь, перетворюючи початкову матрицю на трикутну форму за допомогою лінійних перетворень.
3. Методи розв'язання систем лінійних алгебраїчних рівнянь поділяються на прямі (або точні) та ітераційні (або наближені). Прямий метод, наприклад метод Гаусса, дає точний розв'язок за визначену кількість кроків. Ітераційний метод, наприклад метод Зейделя, навпаки, дає можливість отримати розв'язок з заданою точністю.
4. Компактна схема методу Гаусса - це спосіб запису обчислень, який повністю виконує традиційний метод Гаусса, але в більш економному форматі.
5. Метод Зейделя - це варіант методу простих ітерацій. В процесі обчислення i -ої координати нової точки x використовуються вже обчислені $(i-1)$ координати нової точки та $(n-i+1)$ координати попередньої точки.
6. Збіжність ітераційного процесу означає, що процес ітерацій продовжується до досягнення необхідної точності розв'язку.
7. Самовиправляючими називають методи, які використовуються для зменшення впливу помилок, які можуть виникати під час обчислень, та для покращення точності результатів. Основні методи автоматичного виправлення помилок включають: • Метод найменших квадратів (МНК): Цей метод дозволяє знаходити найкращий наближений розв'язок СЛАР, навіть якщо вихідні дані містять помилки або шум. • Метод QR-розкладу: Цей метод дозволяє розкласти матрицю системи на добуток ортогональної матриці і верхньотрикутної матриці. • Методи регуляризації: Ці методи використовуються для стабілізації обчислень і запобігання перенаванчання в разі надмірного обміну даними. • Методи ітераційного покращення: Ці методи включають ітераційні алгоритми, такі як метод Гаусса-Зейделя, які дозволяють поетапно покращувати розв'язок СЛАР. Вони можуть бути використані для автоматичного виправлення результатів, якщо вони не задовольняють точності. Ці методи допомагають покращити стійкість і точність обчислень в системах лінійних алгебраїчних рівнянь в разі наявності помилок або неточностей в даних або обчисленнях.

Перелік посилань

1. LU alhorithm - https://johnfoster.pge.utexas.edu/numerical-methods-book/LinearAlgebra_LU.html
2. Octave guide - <https://oct2py.readthedocs.io/en/latest/examples.html>
3. Numpy.linalg.solve - <https://numpy.org/doc/stable/reference/generated/numpy.linalg.solve.html>

Додаток А

```
Час виконання: 0.002 сек  
Похибка: 1.459e-28  
Результат:  
[ 4.38112979  5.22361945  6.23205926 -67.80998246]
```

Рис. 1 – Виконання коду на Python

```
Час виконання: 0.0086839 сек  
Похибка: 2.019e-28  
Результат: [4.38112978733865;5.22361944925462;6.23205925893555;-67.8099824648614]  
>> |
```

Рис. 2 – Виконання коду на Octave

Додаток Б

Вміст файлу main.py :

```
import time
import os

import numpy as np

def read_matrix(filename):
    """Функція для зчитування матриці з файлу
    """
    try:
        with open(filename, 'r') as file:
            lines = file.readlines()
            matrix = []
            for line in lines:
                row = [float(x) for x in line.strip().split()]
                matrix.append(row)
            return np.array(matrix)
    except FileNotFoundError:
        print('ERROR', f'Помилка: Файл `{filename}` не знайдено.')
    except ValueError:
        print('ERROR', f'Помилка: Матриця в файлі `{filename}` містить неправильні
значення.')
    except Exception as e:
        print('ERROR', f'Помилка: Виникла невідома помилка при зчитуванні матриці:
{str(e)}')
```

```
def read_vector(filename):
    """Функція для зчитування вектора з файлу
    """
    try:
        with open(filename, 'r') as file:
            lines = file.readlines()
            vector = [float(line.strip()) for line in lines]
            return np.array(vector)
    except FileNotFoundError:
        print('ERROR', f'Помилка: Файл `{filename}` не знайдено.')
    except ValueError:
        print('ERROR', f'Помилка: Вектор в файлі `{filename}` містить неправильні
значення.')
    except Exception as e:
        print('ERROR', f'Помилка: Виникла невідома помилка при зчитуванні вектора:
{str(e)}')
```

```

if __name__ == '__main__':
    os.system('cls') # Очищення консолі
    start_python_time = time.time()

    # Зчитування матриці A та вектора b з файлів
    matrix_A = read_matrix('D:\\KPI\\ASKM\\Labs\\Reznichenko\\Lab_1\\matrix.txt')
    vector_b = read_vector('D:\\KPI\\ASKM\\Labs\\Reznichenko\\Lab_1\\vector.txt')

    if len(vector_b) != len(matrix_A):
        print('ERROR', f'Помилка: Вектор b не відповідає розмірності матриці A.')

    if len(vector_b) >= 10 or len(matrix_A) >= 10:
        print(f'Введіть матрицю меншої розмірності')

    # Розв'язання СЛАР  $Ax = b$ 
    try:
        solution = np.linalg.solve(matrix_A, vector_b)

        # Обчислення похибки
        accuracy = np.mean((vector_b - np.dot(matrix_A, solution)) ** 2)
        python_run_time = time.time() - start_python_time
        print()
        print(f'Час виконання: {round(python_run_time, 4)} сек')
        print(f'Похибка: "{:.3e}".format(accuracy)')
        print(f'Результат: \n{solution}')
    except:
        pass

```

Вміст файлу main.m :

% Зчитування матриці A та вектора b з файлів

try

tic; % Початок вимірювання часу

matrix_A = dlmread('D:\\KPI\\ASKM\\Labs\\Reznichenko\\Lab_1\\matrix.txt');

vector_b = dlmread('D:\\KPI\\ASKM\\Labs\\Reznichenko\\Lab_1\\vector.txt');

catch

disp('ERROR: Помилка при зчитуванні матриці або вектора');

end_try_catch

if length(vector_b) != size(matrix_A, 1)

```

disp('ERROR: Помилка: Вектор b не відповідає розмірності матриці A.');
```

end

```

if length(vector_b) >= 10 || size(matrix_A, 1) >= 10
    disp('Введіть матрицю меншої розмірності');
```

end

```

% Розв'язання СЛАР  $Ax = b$ 
try
    solution = matrix_A \ vector_b;

    % Обчислення похибки
    accuracy = mean((vector_b - matrix_A * solution) .^ 2);
    execution_time = toc; % Кінець вимірювання часу
    disp(['Час виконання: ', num2str(execution_time), ' сек']);
    disp(['Похибка: ', num2str(accuracy, '%.3e')]);
    disp(['Результат: ', mat2str(solution)]);
catch
    disp('ERROR: Помилка при розв'язанні СЛАР');
end_try_catch
```