Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота № 2

з дисципліни «Спеціальні розділи математики-2.  
Чисельні методи»

на тему

«**Розв’язання систем лінійних алгебраїчних рівнянь (СЛАР) прямими методами. Звичайний метод Гауса та метод квадратних коренів**»

Виконала:

студент гр. ІС-34

Колосов Ігор

Викладач:

доц. Рибачук Л.В.

Київ – 2024

**Зміст**

[**Зміст** 2](#_Toc158533784)

[**1 Постановка задачі** 3](#_Toc158533785)

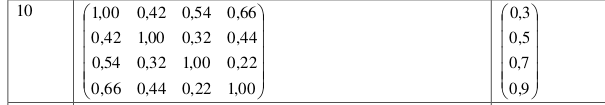
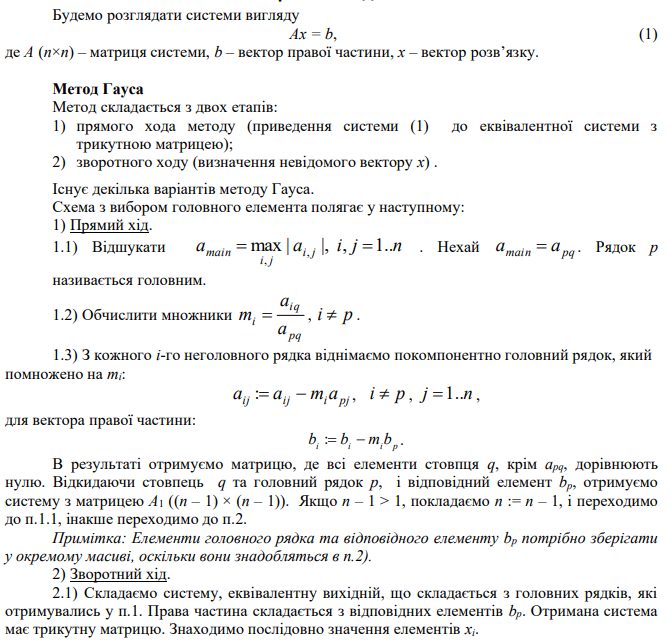
[**2 Розв’язок** 4](#_Toc158533786)

[**3 Контрольні питання** 5](#_Toc158533787)

[4 Висновок 6](#_Toc158533788)

**1 Постановка задачі**

Розв’язати систему рівнянь методом Гауса з кількістю значущих цифр m = 6.  
*Варіант №10*



Скріншот 1 Варіант завдання

Вивести всі проміжні результати та розв’язок системи. Навести результат перевірки: вектор нев’язки r = b – Ax, де x – отриманий розв’язок. Розв’язати задану систему рівнянь за допомогою програмного забезпечення Mathcad. Навести результат перевірки: вектор нев’язки r = b – Axm, де xm – отриманий у Mathcad розв’язок. Порівняти корені рівнянь, отримані у Mathcad, із власними результатами за допомогою методу середньоквадратичної похибки.

**2 Розв’язок**

Система вирішено обчислити за допомогою матричного способу.

Рис 1.1 Вхідні матриці, розв’язок системи, матриця х(округлена до 0,000001)

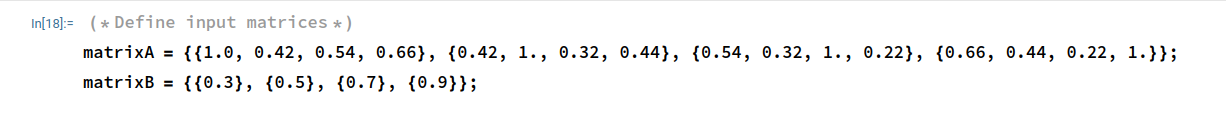


Рис 1.2 Рівняння вектора нев’язки та його значення.

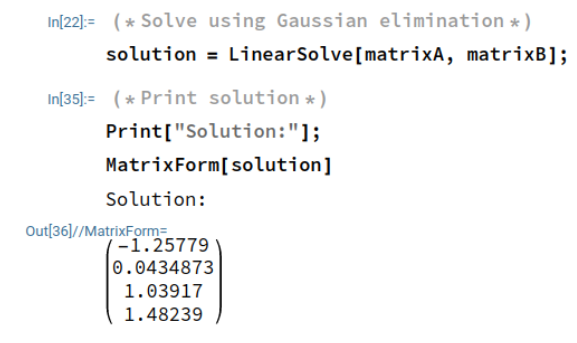
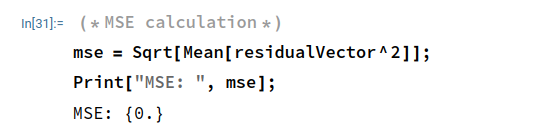


Рис 1.3 середньоквадратична похибка розв’язку програми та Mathcad



Отже, за результатами обчислень похибка відсутня. Метод реалізований добре.

**3 Лістинг програми**

**import** warnings

warnings.filterwarnings("ignore")

**import** numpy **as** np

**def** square\_root\_method(*A*, *b*, *precision***=**6):

n **=** len(A)

T **=** np.zeros((n, n))

print("Початкова матриця A:")

print(np.round(A, precision))

print("\nПочатковий вектор b:")

print(np.round(b, precision))

**for** i **in** range(n):

**for** j **in** range(i, n):

**if** i **==** j:

sum\_k **=** sum(T[k, i]**\*\***2 **for** k **in** range(i))

T[i, i] **=** np.sqrt(A[i, i] **-** sum\_k)

**else**:

sum\_k **=** sum(T[k, i] **\*** T[k, j] **for** k **in** range(i))

T[i, j] **=** (A[i, j] **-** sum\_k) **/** T[i, i]

T[i, j] **=** round(T[i, j], precision)

print(**f**"Матриця під час факторизації ({i}й рядок)\n {T}")

print("\nМатриця T після факторизації:")

print(np.round(T, precision))

y **=** np.zeros(n)

**for** i **in** range(n):

sum\_k **=** sum(T[k, i] **\*** y[k] **for** k **in** range(i))

y[i] **=** (b[i] **-** sum\_k) **/** T[i, i]

y[i] **=** round(y[i], precision)

print("\nПроміжний вектор y:")

print(np.round(y, precision))

x **=** np.zeros(n)

**for** i **in** range(n**-**1, **-**1, **-**1):

sum\_k **=** sum(T[i, k] **\*** x[k] **for** k **in** range(i**+**1, n))

x[i] **=** (y[i] **-** sum\_k) **/** T[i, i]

x[i] **=** round(x[i], precision)

**return** x

*# Використання методу*

A **=** np.array([

[1.0, 0.42, 0.54, 0.66],

[0.42, 1.0, 0.32, 0.44],

[0.54, 0.32, 1.0, 0.22],

[0.66, 0.44, 0.22, 1.0]

])

b **=** np.array([0.3, 0.5, 0.7, 0.9])

x **=** square\_root\_method(A, b)

print("\nРозв'язок x:")

print(np.round(x, 6))

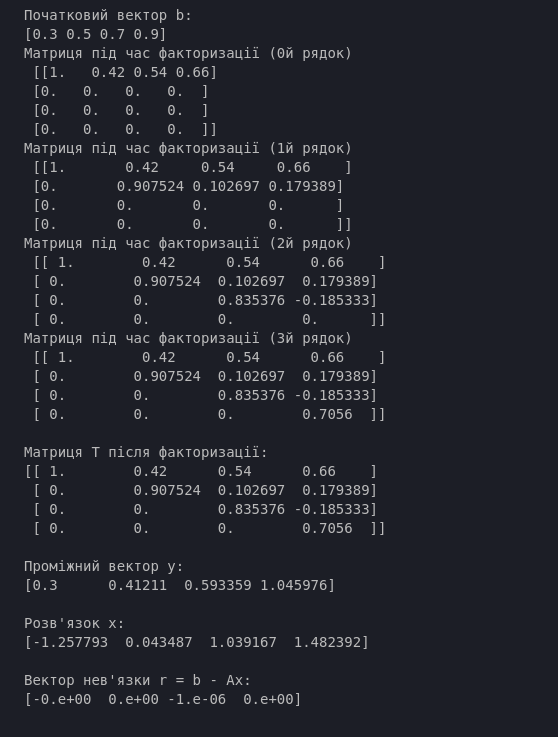
*# Перевірка*

r **=** b **-** np.dot(A, x)

print("\nВектор нев'язки r = b - Ax:")

print(np.round(r, 6))

# **Результат роботи програми:**



# **4 Висновок**

Під час виконання лабораторної роботи я дізнався як можна розв’язувати алгебраїчні рівняння квадратичним методом, реалізованого програмою власноруч. За результати лабораторної роботи можна зробити висновок, шо реалізований код працює коректно.