Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**Звіт**

Про виконання лабораторної роботи №6

**На тему:**

«РОЗВ'ЯЗУВАННЯ ПЕРЕВИЗНАЧЕНИХ СИСТЕМ ЛІНІЙНИХ

АЛГЕБРАЇЧНИХ РІВНЯНЬ»

з дисципліни «чисельні методи»

**Лектор:**

доцент каф. ПЗ

Мельник Н. Б.

**Виконав:**

ст. гр. ПЗ-11

Солтисюк Д.А.

**Прийняла:**

доцент каф. ПЗ

Мельник Н. Б.

« \_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑ = \_\_\_\_\_ \_\_\_\_\_\_\_\_\_ .

Львів – 2022

**Тема.** Розв'язування перевизначених систем лінійних алгебраїчних рівнянь.

**Мета.** Ознайомлення на практиці з методами розв’язування перевизначених систем лінійних алгебраїчних рівнянь.

**Теоретичні відомості**

**Перевизначена система лінійних алгебраїчних рівнянь** – це СЛАР в якій кількість рівнянь є більшою за кількість невідомих. Для того, щоб знайти наближені розв’язки перевизначеної СЛАР використовують метод найменших квадратів(МНК).

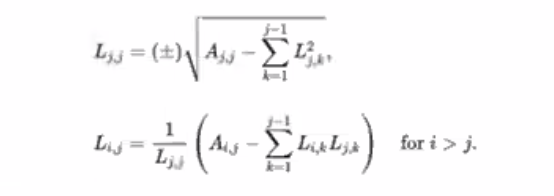
Умови мінімізації функції:

Після того, як продиференціюємо функцію за змінними та прирівняємо вирази, які ми визначили на попередньому кроці до нуля, то отримаємо нормальну систему рівнянь:

в якій кількість рівнянь системи дорівнює кількості невідомих.

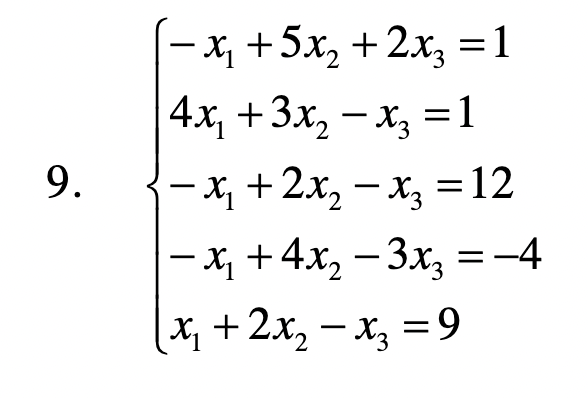
Щоб отримати нову нормальну систему, скористаємося формулою:

Для того, щоб розв’язати нормальну систему рівнянь скористаємось методом квадратного кореня. Для цього розкладемо матрицю за допомогою методу Холецького, який полягає в тому, що задану матрицю розкладаємо на дві матриці . Елементи першої з них визначаємо за формулами:



Після цього метод поділяється на 2 етапи:

1. Прямий(ми знаходимо значення коренів першої системи рівняння, тобто розв’язки рівняння .
2. Зворотній знаходимо корені нормальної матриці розв’язавши рівняння

**Індивідуальне завдання**

Розв’язати перевизначену систему лінійних алгебраїчних рівнянь

методом найменших квадратів. Отриману відповідну нормальну систему

розв’язати методом квадратного кореня.

**Код програми**

**import** math

**from** typing **import** Tuple

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

**from** common.main **import** MatrixOrientedMethod

**from** common.utils **import** back\_sub, forward\_sub, print\_matrix

**class** LeastSquareMethod(MatrixOrientedMethod):

**def** \_\_init\_\_(self, A, B) -> **None**:

super().\_\_init\_\_("Least square", A, B)

**def** square\_decomposition(self,

N: np.ndarray) -> Tuple[np.ndarray, np.ndarray]:

n = N.shape[0]

L = np.zeros(N.shape)

L[0, 0] = math.sqrt(N[0, 0])

**for** i **in** range(0, n):

**for** j **in** range(0, n):

**if** j == 0:

L[i, 0] = N[i, 0] / L[0, 0]

**continue**

**if** i == j:

L[i, i] = math.sqrt(N[i, i] -

sum([L[i, p]\*\*2 **for** p **in** range(0, i)]))

**continue**

**if** i > j:

L[i, j] = (N[i, j] -

sum([L[i, p] \* L[j, p]

**for** p **in** range(0, j)])) / L[j, j]

**return** L, L.transpose()

**def** execute\_method(self):

"""

Transform AX = B -> NX = C, where

[-] N = A transposed \* A

[-] C = A transposed \* B

[-] N = L \* L transposed

[-] L \* Y = C -> forward sub

[-] L transposed \* X = Y -> back sub

"""

AT = self.A.transpose()

N = np.dot(AT, self.A)

C = np.dot(AT, self.B)

N\_det = np.linalg.det(N)

print\_matrix(AT, "A transposed (AT)")

print\_matrix(N, "N = AT \* A")

print\_matrix([C], "C = AT \* B")

**print**(f"\nDeterminant of N = {N\_det}")

**if** N\_det <= 0:

**return** []

L, LT = self.square\_decomposition(N)

print\_matrix(L, "L")

print\_matrix(LT, "L transposed")

y = forward\_sub(L, C)

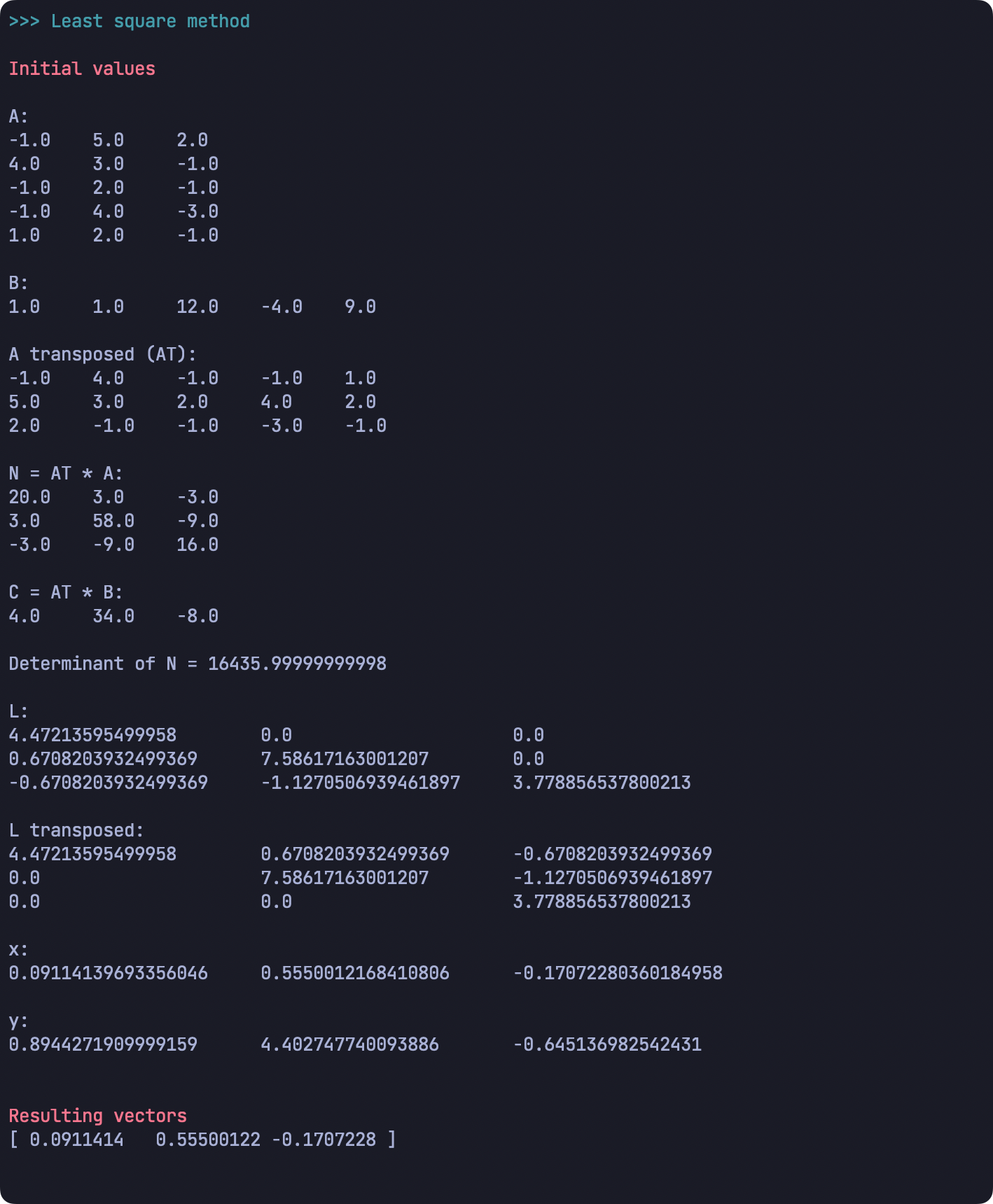
x = back\_sub(LT, y)

print\_matrix([x], "x")

print\_matrix([y], "y")

*# print("\nVerifying results: NX - C = ", np.dot(N, x) - C)*

**return** x

**Протокол роботи**

**Висновки**

Під час виконання лабораторної роботи №6 я ознайомився на практиці з методами розв’язування перевизначених систем лінійних алгебраїчних рівнянь.