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).

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m - b_1 = \varepsilon_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m - b_2 = \varepsilon_2, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m - b_n = \varepsilon_n . \end{cases}$$

Умови мінімізації функції:

$$S(x_1,\ x_2,\ \ldots, x_m) = \sum_{j=1}^{m} \varepsilon^2{}_i .$$

Після того, як продиференціюємо функцію за змінними $x_i\left(i = \overline{1,m}\right)$ та прирівняємо вирази, які ми визначили на попередньому кроці до нуля, то отримаємо нормальну систему рівнянь:

$$\sum_{i=1}^{n} a_{ik} \left(\sum_{j=1}^{m} a_{ij}x_j - b_j \right) = 0,\ k = \overline{1,m},$$

в якій кількість рівнянь системи дорівнює кількості невідомих.

Щоб отримати нову нормальну систему, скористаємося формулою:

$$NX = C,$$

$$де\ N = A^T A,\ \ C = A^T B .$$

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 $L\ та\ L^T$. Елементи першої з них визначаємо за формулами:

$$L_{j,j} = (\pm)\sqrt{A_{j,j} - \sum_{k=1}^{j-1} L_{j,k}^2},$$

$$L_{i,j} = \frac{1}{L_{j,j}}\left(A_{i,j} - \sum_{k=1}^{j-1} L_{i,k}L_{j,k}\right) \quad \text{for } i > j.$$

Після цього метод поділяється на 2 етапи:

1. Прямий(ми знаходимо значення коренів першої системи рівняння, тобто розв'язки рівняння $AY = B$).
2. Зворотній знаходимо корені нормальної матриці розв'язавши рівняння $A^T X = Y$.

Індивідуальне завдання

9. $$\begin{cases} -x_1 + 5x_2 + 2x_3 = 1 \\ 4x_1 + 3x_2 - x_3 = 1 \\ -x_1 + 2x_2 - x_3 = 12 \\ -x_1 + 4x_2 - 3x_3 = -4 \\ x_1 + 2x_2 - x_3 = 9 \end{cases}$$

Розв'язати перевизначену систему лінійних алгебраїчних рівнянь методом найменших квадратів. Отриману відповідну нормальну систему розв'язати методом квадратного кореня.

Код програми

```python
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")
```

```python
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

```
>>> Least square method

Initial values

A:
-1.0    5.0     2.0
4.0     3.0     -1.0
-1.0    2.0     -1.0
-1.0    4.0     -3.0
1.0     2.0     -1.0

B:
1.0     1.0     12.0    -4.0    9.0

A transposed (AT):
-1.0    4.0     -1.0    -1.0    1.0
5.0     3.0     2.0     4.0     2.0
2.0     -1.0    -1.0    -3.0    -1.0

N = AT * A:
20.0    3.0     -3.0
3.0     58.0    -9.0
-3.0    -9.0    16.0

C = AT * B:
4.0     34.0    -8.0

Determinant of N = 16435.99999999998

L:
4.47213595499958        0.0                     0.0
0.6708203932499369      7.58617163001207        0.0
-0.6708203932499369     -1.1270506939461897     3.778856537800213

L transposed:
4.47213595499958        0.6708203932499369      -0.6708203932499369
0.0                     7.58617163001207        -1.1270506939461897
0.0                     0.0                     3.778856537800213

x:
0.09114139693356046     0.5550012168410806      -0.17072280360184958

y:
0.8944271909999159      4.402747740093886       -0.645136982542431


Resulting vectors
[ 0.0911414   0.55500122 -0.1707228 ]
```

Висновки

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