Лабораторна робота 2
Тема: Методи розв'язування лінійних алгебраїчних рівнянь

Метод Гауса (Прямий хід)

У загальному випадку метод Гауса складається з двох етапів:

1 Прямий хід, у результаті якого система, що розв'язується, перетворюється в еквівалентну систему з верхньою трикутною матрицею коефіцієнтів такого виду:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ 0 \cdot x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ 0 \cdot x_1 + 0 \cdot x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

Метод Гауса (Зворотний хід)

2 Зворотний хід, що дозволяє визначити вектор розв'язку, починаючи з останнього рівняння системи шляхом підстановки координат вектора невідомих, отриманих на попередньому кроці.

Система лінійних рівнянь може:

1) мати єдиний розв'язок;

2) мати нескінченно багато розв'язків;

3) не мати розв'язків (бути несумісною).

Код програми Python (Прямий хід)

```python
# Матриця A та вектор b
A = [[3,  0, 0,  7,  1],
     [9,  7, 9, -1,  4],
     [0,  2, 2,  0,  0],
     [5, -2, 5,  0,  6],
     [1,  9, 7,  8, -3]]
b = [7, -9, 6, 15, -7]

# Довжина списку
d = len(b)

# Прямий хід
for m in range(0,d-1):
    for j in range(m+1,d):
        k = A[j][m] / A[m][m]
        for i in range(m,d):
            A[j][i] = A[j][i] - k*A[m][i]
        b[j] = b[j] - k*b[m]
```

Код програми Python (Зворотний хід)

```python
# Скалярний добуток
def scalar_product(A, b, n):
    s = 0
    for i in range(n+1, d):
        s = s + A[n][i]*b[i]
    return s

# Зворотний хід
for i in range(d-1, -1, -1):
    b[i] = (b[i] - scalar_product(A, b, i)) / A[i][i]

# Відображення результатів
print('Матриця A після перетворень прямого ходу')
for i in range(0,d):
    print(A[i])
print()
print("Вектор розв'язку x")
for i in range(0,d):
    print(b[i])
```

Тестування програми (Матричний калькулятор)

$$\begin{cases} 3\cdot x_1 + 7\cdot x_4 + x_5 = 7 \\ 7\cdot x_2 + 9\cdot x_3 - 22\cdot x_4 + x_5 = -30 \\ -0{,}\overline{5714286}\cdot x_3 + 6{,}\overline{2857143}\cdot x_4 - 0{,}\overline{2857143}\cdot x_5 = 14{,}\overline{5714286} \\ 65{,}\overline{3}333333\cdot x_4 + 0{,}8\overline{3}33333\cdot x_5 = 187{,}8\overline{3}33333 \\ -2{,}125\cdot x_5 = -40{,}375 \end{cases}$$

$$X = \begin{pmatrix} -10{,}\overline{1428571} \\ 9{,}0408163 \\ -6{,}0408163 \\ 2{,}6326531 \\ 19 \end{pmatrix}$$

Результати роботи програми

```
Матриця A після перетворень прямого ходу
[3, 0, 0, 7, 1]
[0.0, 7.0, 9.0, -22.0, 1.0]
[0.0, 0.0, -0.5714285714285712, 6.285714285714286, -0.2857142857142857]
[0.0, 0.0, 0.0, 65.33333333333337, 0.8333333333333317]
[0.0, 0.0, 8.881784197001252e-16, 0.0, -2.1249999999999982]

Вектор розв'язку x
-10.142857142857148
9.040816326530626
-6.040816326530626
2.6326530612244894
19.000000000000014
```

Модуль NumPy

```python
import numpy as np
from numpy.linalg import solve

A = np.array([[3, 0, 0, 7, 1],
              [9, 7, 9,-1, 4],
              [0, 2, 2, 0, 0],
              [5,-2, 5, 0, 6],
              [1, 9, 7, 8,-3]])
b = np.array([7, -9, 6, 15, -7])

x = solve(A,b)

for i in range(len(x)):
    print(x[i])
```

```
-10.142857142857139
9.040816326530603
-6.040816326530604
2.6326530612244885
18.999999999999986
```

Модуль SymPy

```python
import sympy

A = sympy.Matrix([[3, 0, 0, 7, 1],
                  [9, 7, 9,-1, 4],
                  [0, 2, 2, 0, 0],
                  [5,-2, 5, 0, 6],
                  [1, 9, 7, 8,-3]])
b = sympy.Matrix([7, -9, 6, 15, -7])

x = A.solve(b)

print(x)
```

```
Matrix([[-71/7], [443/49], [-296/49], [129/49], [19]])
```

Модуль SymPy

```
import sympy

a11 = sympy.symbols('a11')
a12 = sympy.symbols('a12')
a21 = sympy.symbols('a21')
a22 = sympy.symbols('a22')

b1 = sympy.symbols('b1')
b2 = sympy.symbols('b2')

A = sympy.Matrix([[a11, a12],
                  [a21, a22]])
b = sympy.Matrix([b1, b2])

x = A.solve(b)

print(x)
```

```
Matrix([[(-a12*b2 + a22*b1)/(a11*a22 - a12*a21)],
[(a11*b2 - a21*b1)/(a11*a22 - a12*a21)]])
```

Завдання:

1) Ознайомитись з документацією модулів NumPy, SymPy. Заповнити таблицю:

Характеристики	NumPy	SymPy
Основне призначення		
Типи даних		
Переваги		
Недоліки		
Галузі використання		

2) Реалізувати наведені вище приклади.
3) Розв’язати рівняння відповідно до варіанту, використовуючи модулі NumPy, SymPy.

1. $A=\begin{pmatrix}4 & 8 & 4\\ 1 & 1 & -2\\ 2 & -3 & -1\end{pmatrix}$, $B=\begin{pmatrix}24\\ 1\\ -5\end{pmatrix}$,

2. $A=\begin{pmatrix}1 & 2 & 5\\ -2 & -2 & -4\\ 3 & -1 & -3\end{pmatrix}$, $B=\begin{pmatrix}3\\ 0\\ -9\end{pmatrix}$,

3. $A=\begin{pmatrix}1 & 1 & -2\\ 1 & 2 & 3\\ 4 & 5 & -2\end{pmatrix}$, $B=\begin{pmatrix}-4\\ 5\\ -5\end{pmatrix}$,

4. $A=\begin{pmatrix}1 & 1 & 3\\ 2 & -3 & -1\\ 3 & 2 & 2\end{pmatrix}$, $B=\begin{pmatrix}-5\\ -1\\ -2\end{pmatrix}$,

5. $A=\begin{pmatrix}2 & 1 & -1\\ 3 & -2 & 1\\ 1 & 3 & -2\end{pmatrix}$, $B=\begin{pmatrix}0\\ 6\\ -4\end{pmatrix}$,

6. $A=\begin{pmatrix}1 & 4 & -3\\ 1 & -3 & 2\\ 2 & -5 & -1\end{pmatrix}$, $B=\begin{pmatrix}6\\ -3\\ -9\end{pmatrix}$,

7. $A=\begin{pmatrix}1 & -2 & 1\\ 2 & 3 & -1\\ 1 & -1 & 2\end{pmatrix}$, $B=\begin{pmatrix}-1\\ -5\\ 0\end{pmatrix}$,

8. $A=\begin{pmatrix}1 & 2 & 1\\ 2 & -3 & -2\\ 2 & 1 & 1\end{pmatrix}$, $B=\begin{pmatrix}1\\ 1\\ 3\end{pmatrix}$,

9. $A=\begin{pmatrix}3 & 1 & 2\\ 2 & 2 & -3\\ 2 & -1 & 2\end{pmatrix}$, $B=\begin{pmatrix}-3\\ 8\\ -5\end{pmatrix}$,

10. $A=\begin{pmatrix}1 & 2 & -1\\ 2 & -1 & 3\\ 3 & -3 & -2\end{pmatrix}$, $B=\begin{pmatrix}-2\\ 6\\ 4\end{pmatrix}$,

11. $A=\begin{pmatrix}1 & 2 & 1\\ 2 & -3 & -1\\ 1 & 1 & 2\end{pmatrix}$, $B=\begin{pmatrix}6\\ -5\\ 5\end{pmatrix}$,

12. $A=\begin{pmatrix}2 & 1 & -1\\ 1 & -2 & -3\\ 3 & 3 & 2\end{pmatrix}$, $B=\begin{pmatrix}-5\\ -5\\ -4\end{pmatrix}$,

13. $A=\begin{pmatrix}3 & -3 & -2\\ 2 & -1 & 3\\ 1 & 2 & -1\end{pmatrix}$, $B=\begin{pmatrix}2\\ 9\\ -3\end{pmatrix}$,

14. $A=\begin{pmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 2 & -3 & -2\end{pmatrix}$, $B=\begin{pmatrix}-1\\ 0\\ 1\end{pmatrix}$,

15. $A=\begin{pmatrix}2 & 2 & -3\\ 2 & 1 & 2\\ 1 & -2 & 1\end{pmatrix}$, $B=\begin{pmatrix}-3\\ 3\\ 4\end{pmatrix}$,

4) Розв'язати задачу оптимізація виробництва та розподілу ресурсів. Три підприємства A, B і C виробляють три види продукції: продукція X, продукція Y і продукція Z. Кожне підприємство використовує три основні ресурси: робочу силу, матеріали та енергію. Мета полягає у визначенні обсягу виробництва кожного виду продукції на кожному підприємстві, щоб задовольнити потреби ринку та ефективно використати наявні ресурси.

Вихідні дані:

Кількість робочих годин, необхідних для виробництва одиниці продукції:

Продукція X: підприємство A — 2 години, підприємство B — 3 години, підприємство C — 1 година.

Продукція Y: підприємство A — 4 години, підприємство B — 1 година, підприємство C — 2 години.

Продукція Z: підприємство A — 3 години, підприємство B — 2 години, підприємство C — 5 годин.

Кількість матеріалів (в одиницях), необхідних для виробництва одиниці продукції:

Продукція X: підприємство A — 5 одиниць, підприємство B — 4 одиниці, підприємство C — 3 одиниці.

Продукція Y: підприємство A — 2 одиниці, підприємство B — 5 одиниць, підприємство C — 6 одиниць.

Продукція Z: підприємство A — 3 одиниці, підприємство B — 4 одиниці, підприємство C — 2 одиниці.

Енергетичні витрати (в кВт·год) для виробництва одиниці продукції:

Продукція X: підприємство A — 6 кВт·год, підприємство B — 7 кВт·год, підприємство C — 5 кВт·год.

Продукція Y: підприємство A — 3 кВт·год, підприємство B — 4 кВт·год, підприємство C — 5 кВт·год.

Продукція Z: підприємство A — 2 кВт·год, підприємство B — 3 кВт·год, підприємство C — 4 кВт·год.

Загальні обмеження ресурсів:

Максимальна кількість робочих годин: 240 годин.

Максимальна кількість матеріалів: 500 одиниць.

Максимальна кількість енергії: 300 кВт·год.

Необхідно скласти систему алгебраїчних рівнянь для оптимізації обсягів виробництва продукції X, Y та Z на підприємствах A, B та C, щоб використати наявні ресурси та задовольнити ринковий попит.