

## Розв'язання системи лінійних рівнянь методом Гауса

```
In [ ]: import numpy as np

A = np.array([[3.81, 0.25, 1.28, 1.75],
              [2.25, 1.32, 5.58, 0.49],
              [5.31, 7.28, 0.98, 1.04],
              [10.39, 2.45, 3.35, 2.28]])

b = np.array([4.21, 8.97, 2.38, 12.98])
```

Прями хід метода Гауса

```
In [ ]: n = len(b)

# Комбінування матриці A та вектора b в початкову розширену матрицю
augmented_matrix = np.column_stack((A, b))

for i in range(n):
    # Нормалізація поточного рядка
    pivot_row = augmented_matrix[i]
    augmented_matrix[i] = pivot_row / pivot_row[i]

    # Виразимо інші рядки з метою обнулення стовпців під головним елементом
    for j in range(i + 1, n):
        factor = augmented_matrix[j, i]
        augmented_matrix[j] -= factor * augmented_matrix[i]

print(augmented_matrix)
```

```
[[ 1.          0.0656168  0.33595801  0.45931759  1.10498688]
 [ 0.          1.          4.11484989 -0.4635637   5.53052589]
 [-0.         -0.          1.         -0.06186421  1.42611567]
 [-0.         -0.         -0.          1.         -1.0776198 ]]
```

Зворотній хід методу Гауса

```
In [ ]: # Зворотний хід методу Гауса
x = np.zeros(n)
for i in range(n - 1, -1, -1):
    x[i] = augmented_matrix[i, -1] - np.dot(augmented_matrix[i, i+1:n], x[i+1:])

print("Розв'язок системи рівнянь:", x)
```

Розв'язок системи рівнянь: [ 1.18017764 -0.56295045 1.35944957 -1.0776198 ]

Перевірка точності

```
In [ ]: print(np.linalg.norm(np.dot(A,x)-b))
```

1.0433733987432285e-14

## Висновок

Метод Гауса є найефективнішим(асимптотично) та універсальним методом розв'язку, який дає дуууже близький до точного результата(помилка в моєму випадку складає 14 знаків після коми)