In [ ]:
```
Computational Mathematic Lab 5 Methods for solving SLAE (Gaussian elimination and Gauss-Seidel)
Nurshanov Dias
IT3-2208
```

In [2]:
```
# 1) Implementation of each method.
import numpy as np

def gauss_seidel_modified(A, b, x0, tolerance=1e-4, max_iterations=100):
    n = len(b)
    x_current = np.array(x0)

    for iteration in range(max_iterations):
        x_new = np.zeros_like(x_current)

        for i in range(n):
            sum_previous = np.dot(A[i, :i], x_new[:i])
            sum_current = np.dot(A[i, i + 1:], x_current[i + 1:])
            x_new[i] = (b[i] - sum_previous - sum_current) / A[i, i]

        difference = np.linalg.norm(x_new - x_current, ord=np.inf)
        if difference < tolerance:
            return np.round(x_new, 4), iteration + 1

        x_current = x_new

    return np.round(x_current, 4), max_iterations

def gaussian_elimination_modified(A, b):
    n = len(b)
    A = np.array(A, dtype=float)
    b = np.array(b, dtype=float)

    for i in range(n):
        for j in range(i + 1, n):
            multiplier = A[j, i] / A[i, i]
            A[j, i:] -= multiplier * A[i, i:]
            b[j] -= multiplier * b[i]
    x = np.zeros(n)
    for i in range(n - 1, -1, -1):
        x[i] = (b[i] - np.dot(A[i, i + 1:], x[i + 1:])) / A[i, i]

    return np.round(x, 4)
```

In [5]:
```
#2) Output of the code.
A = np.array([
    [3, 1, 2],
    [2, 4, 1],
    [1, 2, 5]
], dtype=float)

b = np.array([10, 11, 13], dtype=float)
x0 = np.zeros_like(b)

# Gauss-Seidel solution
solution_gs, iterations_gs = gauss_seidel_modified(A, b, x0)
print("Gauss-Seidel solution:", solution_gs)
print("Iterations:", iterations_gs)

# Gaussian Elimination solution
solution_ge = gaussian_elimination_modified(A, b)
print("Gaussian Elimination solution:", solution_ge)
```

```
Gauss-Seidel solution: [1.7333 1.4667 1.6667]
Iterations: 8
Gaussian Elimination solution: [1.7333 1.4667 1.6667]
```

In [ ]:
```
#3 solution by hand
Gauss-Seidel Method - Solution by Hand (2 iterations)

Initial guess:
x1 = 0, x2 = 0, x3 = 0

First Iteration:
From equation 1: 3x1 + x2 + 2x3 = 10
x1 = (10 - 0 - 2 * 0) / 3 = 3.333

From equation 2: 2x1 + 4x2 + x3 = 11
x2 = (11 - 2 * 3.333 - 0) / 4 = 1.083

From equation 3: x1 + 2x2 + 5x3 = 13
x3 = (13 - 3.333 - 2 * 1.083) / 5 = 1.500
```

```
After first iteration:
x1 = 3.333, x2 = 1.083, x3 = 1.500

Second Iteration:
From equation 1: 3x1 + x2 + 2x3 = 10
x1 = (10 - 1.083 - 2 * 1.500) / 3 = 1.972

From equation 2: 2x1 + 4x2 + x3 = 11
x2 = (11 - 2 * 1.972 - 1.500) / 4 = 1.389

From equation 3: x1 + 2x2 + 5x3 = 13
x3 = (13 - 1.972 - 2 * 1.389) / 5 = 1.650

After second iteration:
x1 = 1.972, x2 = 1.389, x3 = 1.650
```