

Assignment 2

Arifur Rahman
400300356

```
def forward_substitution(L, b):
    n = len(b)
    x = [[0.0] * n for i in range(n)]
    for i in range(n):
        if L[i][i] == 0.0:
            return None
        sum = 0
        for j in range(i):
            sum += L[i][j] * x[j]
        x[i] = (b[i] - sum) / L[i][i]
    return x

def backward_substitution(U, b):
    n = len(b)
    x = [[0.0] * n for i in range(n)]
    for i in range(n - 1, -1, -1):
        if U[i][i] == 0.0:
            return None
        sum = 0.0
        for j in range(i + 1, n):
            sum += U[i][j] * x[j]
        x[i] = (b[i] - sum) / U[i][i]
    return x

def upper_triangular_matrix_U(k, n, L, U):
    # Get the upper triangular matrix U
    for j in range(k, n):
        sum = 0.0
        for p in range(k):
            sum += L[k][p] * U[p][j]
        U[k][j] = A[k][j] - sum
    print(f"upper triangular matrix step {k+1}: {U}")

def lower_triangular_matrix_L(k, n, L, U, A):
    # Get the lower triangular matrix L
    for i in range(k + 1, n):
        sum = 0.0
        for p in range(k):
            sum += L[i][p] * U[p][k]
        L[i][k] = (A[i][k] - sum) / U[k][k]
    print(f"Lower triangular matrix step {k+1}: {L}\n")
```

```

def LU_decomposition(A):
    n = len(A)
    L = [[0.0] * n for i in range(n)]
    U = [[0.0] * n for i in range(n)]

    for k in range(n):
        if A[k][k] == 0:
            return None

        L[k][k] = 1
        upper_triangular_matrix_U(k, n, L, U)
        lower_triangular_matrix_L(k, n, L, U, A)

    return L, U

if __name__ == '__main__':
    # gauss_elimination
    print("===== gauss_elimination =====")

    A = [[1, 2, 1, -1], [3, 2, 4, 4], [4, 4, 3, 4], [2, 0, 1, 5]]
    b = [5, 16, 22, 15]
    res = gauss_elimination(A, b)
    print(f"A = {A}")
    print(f"b = {b}")
    print(f"Guass Elimination Solution = {res}")

```

OUTPUT

upper triangular matrix step 1:

```

[[1.0, 2.0, 1.0, -1.0],
 [0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0]]

```

upper triangular matrix step 2:

```

[[1.0, 2.0, 1.0, -1.0],
 [0.0, -4.0, 1.0, 7.0],
 [0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0]]

```

upper triangular matrix step 3:

```

[[1.0, 2.0, 1.0, -1.0],
 [0.0, -4.0, 1.0, 7.0],
 [0.0, 0.0, -2.0, 1.0],
 [0.0, 0.0, 0.0, 0.0]]

```

upper triangular matrix step 4:

```
[[1.0, 2.0, 1.0, -1.0],  
[0.0, -4.0, 1.0, 7.0],  
[0.0, 0.0, -2.0, 1.0],  
[0.0, 0.0, 0.0, -1.0]]
```

Lower triangular matrix step 1:

```
[[1, 0.0, 0.0, 0.0],  
[3.0, 0.0, 0.0, 0.0],  
[4.0, 0.0, 0.0, 0.0],  
[2.0, 0.0, 0.0, 0.0]]
```

Lower triangular matrix step 2:

```
[[1, 0.0, 0.0, 0.0],  
[3.0, 1, 0.0, 0.0],  
[4.0, 1.0, 0.0, 0.0],  
[2.0, 1.0, 0.0, 0.0]]
```

Lower triangular matrix step 3:

```
[[1, 0.0, 0.0, 0.0],  
[3.0, 1, 0.0, 0.0],  
[4.0, 1.0, 1, 0.0],  
[2.0, 1.0, 1.0, 0.0]]
```

Lower triangular matrix step 4:

```
[[1, 0.0, 0.0, 0.0],  
[3.0, 1, 0.0, 0.0],  
[4.0, 1.0, 1, 0.0],  
[2.0, 1.0, 1.0, 1]]
```

A = [[1, 2, 1, -1], [3, 2, 4, 4], [4, 4, 3, 4], [2, 0, 1, 5]]

b = [5, 16, 22, 15]

Gauss Elimination Solution = [16.0, -6.0, -2.0, -3.0]T