In [7]:
```
import numpy as np
from scipy.linalg import cholesky

def generate_positive_definite_matrix(size):
    random_matrix = np.random.randn(size, size)

    # Compute the Cholesky decomposition
    cholesky_matrix = cholesky(random_matrix @ random_matrix.T, lower=True)

    # product of Cholesky matrix and its transpose
    positive_definite_matrix = cholesky_matrix @ cholesky_matrix.T

    return positive_definite_matrix
```

In [8]:
```
def is_positive_definite(matrix):
    # Check if the matrix is symmetric
    if not np.allclose(matrix, matrix.T):
        return False
    n = len(matrix)
    for i in range(1, n + 1):
        minor = matrix[:i, :i]
        determinant = np.linalg.det(minor)
        if determinant <= 0:
            return False

    return True
```

In [9]:
```
matrix_size = 4

random_positive_definite_matrix = generate_positive_definite_matrix(matrix_size)

# Check if the matrix is positive definite using Sylvester's criterion
is_positive_definite_result = is_positive_definite(random_positive_definite_matrix)

# Print the matrix and the result of the check
print("Random Positive Definite Matrix:")
print(random_positive_definite_matrix)
print("\nIs Positive Definite:", is_positive_definite_result)
```

```
Random Positive Definite Matrix:
[[ 0.38267482  0.85201162  0.45859527 -0.42082227]
 [ 0.85201162  2.27396127  2.03698712 -0.33436697]
 [ 0.45859527  2.03698712  5.14677827  3.42379003]
 [-0.42082227 -0.33436697  3.42379003  4.39139546]]

Is Positive Definite: True
```

In [ ]: