TASK 4: Commutation Relations and Euler Decomposition

Aim:
To verify Pauli matrix commutation relations and decompose a gate using Euler angles.
Algorithm:
· Implement commutator and anticommutator functions.
· Verify Pauli commutation and anticommutation rules.
· Decompose Hadamard gate using Euler angles.
· Compare with actual Hadamard matrix.

Program:
```
import numpy as np

print("\n" + "="*50)
print("TASK 4: COMMUTATION RELATIONS AND EULER
ANGLES")
print("="*50)

# Define Pauli matrices
pauli_x = np.array([[0, 1], [1, 0]])
pauli_y = np.array([[0, -1j], [1j, 0]])
pauli_z = np.array([[1, 0], [0, -1]])

def commutator(A, B):
    """Compute commutator [A,B] = AB - BA"""
    return A @ B - B @ A

def anticommutator(A, B):
    """Compute anticommutator {A,B} = AB + BA"""
```

```
    return A @ B + B @ A

# Verify Pauli commutation relations
print("Commutation relations:")
print("[σₓ, σᵧ] =\n", commutator(pauli_x, pauli_y))
print("[σᵧ, σ_z] =\n", commutator(pauli_y, pauli_z))
print("[σ_z, σₓ] =\n", commutator(pauli_z, pauli_x))

print("\nAnticommutation relations:")
print("{σₓ, σᵧ} =\n", anticommutator(pauli_x, pauli_y))
print("{σₓ, σₓ} =\n", anticommutator(pauli_x, pauli_x))

def euler_decomposition(theta, phi, lam):
    """Decompose single-qubit gate using Euler angles"""
    return np.array([
        [np.cos(theta/2), -np.exp(1j*lam) * np.sin(theta/2)],
        [np.exp(1j*phi) * np.sin(theta/2), np.exp(1j*(phi+lam)) * np.cos(theta/2)]
    ])

hadamard = np.array([[1, 1], [1, -1]]) / np.sqrt(2)
euler_h = euler_decomposition(np.pi/2, 0, np.pi)

print(f"\nHadamard gate:\n{hadamard}")
print(f"Euler decomposition:\n{euler_h}")
print(f"Difference: {np.max(np.abs(hadamard - euler_h)):.10f}")
```