**Date:**

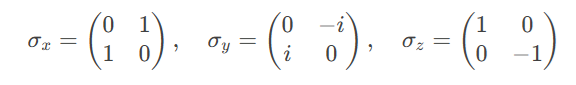
**TASK 2: Pauli Matrices and Eigenvalues/Eigenvectors**

**Aim:**

To analyze Pauli matrices through application on qubit states and eigenvalue decomposition.

1. **Mathematical Model**

The **Pauli matrices** are a set of three 2×2 complex Hermitian and unitary matrices that are widely used in **quantum mechanics**, particularly in spin systems (spin-1/2 particles), quantum computing, and quantum information theory. They are denoted as *σx*​, *σy*​, and *σz*​, and are defined as follows:

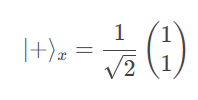


**1.1. Eigenvalues and Eigenvectors of the Pauli Matrices**

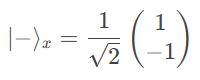
Each Pauli matrix has eigenvalues *λ* = ±1 and corresponding eigenvectors:

1. ***σx*​ (Pauli-X Matrix)**

* **Eigenvalues**: *λ* = +1, −1
* **Eigenvectors**:
* For *λ* = +1

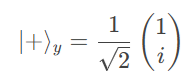


* For λ = −1

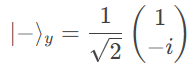


1. ***σy*​ (Pauli-Y Matrix)**

* **Eigenvalues**: *λ* = +1, −1
* **Eigenvectors**:
  + For *λ* = +1

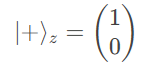


* + For *λ* = −1

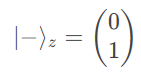


1. ***σz*​(Pauli-Z Matrix)**

* **Eigenvalues**: *λ* = +1, −1
* **Eigenvectors**:
  + For *λ* = +1



* + For *λ* = −1



1. **Observations**

* All three Pauli matrices have eigenvalues ±1.
* Their eigenvectors are orthonormal, i.e.,  ⟨+∣−⟩ = 0 and ⟨±∣±⟩ = 1.
* The eigenvectors of *σx*​ and *σy*​ are superpositions of the eigenvectors of *σz*​, reflecting the non-commutativity of the Pauli matrices ([*σi*​,*σj*​]=2*iϵijk*​*σk*​).

1. **Physical Interpretation**

In quantum mechanics, the Pauli matrices represent spin measurements along the *x*, *y*, and *z* axes for a spin-1/2 particle (like an electron). The eigenvalues ±1 correspond to the possible outcomes of a spin measurement (spin-up or spin-down), and the eigenvectors represent the spin states along the respective axes.

1. **Algorithm**

* Define Pauli-X, Y, and Z matrices.
* Apply these matrices to |0⟩ and |1⟩ states.
* Use linear algebra to compute eigenvalues and eigenvectors.
* Print matrix properties.

1. **Program**

print("\n" + "="\*50)

print("TASK 2: PAULI MATRICES AND EIGEN-ANALYSIS")

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]])

print("Pauli-X matrix:")

print(pauli\_x)

print("\nPauli-Y matrix:")

print(pauli\_y)

print("\nPauli-Z matrix:")

print(pauli\_z)

# Apply to qubit states

qubit\_0 = np.array([1, 0]) # |0⟩

qubit\_1 = np.array([0, 1]) # |1⟩

print("\nApplying Pauli-X to |0⟩:", pauli\_x @ qubit\_0)

print("Applying Pauli-X to |1⟩:", pauli\_x @ qubit\_1)

# Compute eigenvalues and eigenvectors

def analyze\_operator(matrix, name):

eigenvals, eigenvecs = eig(matrix)

print(f"\n{name} Eigenvalues:", eigenvals)

print(f"{name} Eigenvectors:")

for i, vec in enumerate(eigenvecs.T):

print(f" λ={eigenvals[i]:.1f}: {vec}")

analyze\_operator(pauli\_x, "Pauli-X")

analyze\_operator(pauli\_y, "Pauli-Y")

analyze\_operator(pauli\_z, "Pauli-Z")

**Output:**

