Quantum Gates and Their Matrices

- **Hadamard (H) Gate:** Creates a superposition state from a basis state.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

- **RX (Rotation around X-axis) Gate:** Rotates a qubit state around the X-axis of the Bloch sphere by an angle θ.

$$RX(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

- **RY (Rotation around Y-axis) Gate:** Rotates a qubit state around the Y-axis by an angle ζ.

$$RY(\zeta) = \begin{pmatrix} \cos\left(\frac{\zeta}{2}\right) & -\sin\left(\frac{\zeta}{2}\right) \\ \sin\left(\frac{\zeta}{2}\right) & \cos\left(\frac{\zeta}{2}\right) \end{pmatrix}$$

- **RZ (Rotation around Z-axis) Gate:** Rotates a qubit state around the Z-axis by an angle ϕ.

$$RZ(\phi) = \begin{pmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{i\frac{\phi}{2}} \end{pmatrix}$$

- **CNOT (Controlled-NOT) Gate:** An entangling gate that flips the target qubit if the control qubit is in state $|1\rangle$.

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Quantum Circuit Composition

- **Circuit Composition:** The overall unitary operation represented by this circuit is obtained by multiplying the matrices of the individual gates in the order they are applied. For parameterized gates, you'll have matrices with symbolic entries.

- **Parameterized Circuits:** This circuit uses parameters (θ, ϕ, ζ) to define rotations, making it versatile for various applications, including quantum machine learning and variational algorithms.

Quantum Measurement

- **Projective Measurement:** The `qc.measure_all()` adds measurement to all qubits, projecting their state onto the computational basis ($|0\rangle$ and $|1\rangle$) with probabilities determined by the quantum state immediately before measurement.

Quantum State Evolution

- **Initial State:** Quantum circuits typically start with all qubits in the $|0\rangle$ state.

- **State Evolution:** As the circuit applies gates, the state of the qubits evolves. The final state before measurement can be calculated by applying the circuit's overall unitary matrix to the initial state vector.

Visualization and Simulation

- **Circuit Diagram:** `qc.draw('mpl')` visualizes the quantum circuit, showing the sequence of gates and measurements.

- **Simulation:** To simulate the circuit and observe the probabilities of different outcomes, you'd use a quantum simulator like Qiskit's Aer.