Quantum Computing Setup & Visualization – Complete Notes

1. Installing Python 3.11.9

Download Python 3.11.9 from the official Python website. During installation, check 'Add Python to PATH', choose Custom Installation, enable pip and environment variables, and complete setup. Verify installation using 'python --version' and 'pip --version'.

2. Installing Packages in VS Code

Create a project folder and open it in VS Code. Create a virtual environment using 'python -m venv .venv'. Activate the environment and select it as the Python interpreter. Upgrade pip and install required packages: numpy, scipy, pandas, tqdm, plotly, jupyter, qiskit, pennylane, cirq, and qutip.

3. What is a Qubit?

A qubit is the fundamental unit of quantum information. Unlike classical bits, a qubit can exist in a superposition of |0■ and |1■. Its state is represented as |ψ■ = α|0■ + β|1■, where probabilities are given by |α|² and |β|².

Qubits exhibit key quantum properties such as superposition, phase, and entanglement. Measurement collapses the qubit into a definite classical outcome.

4. Using Qubits in Practice

Qubits are manipulated using quantum gates such as X, H, Z, and CNOT. Frameworks like Qiskit, PennyLane, Cirq, and QuTiP allow simulation of quantum circuits, statevectors, density matrices, and measurements.

5. Matplotlib vs Plotly for Graphs

Matplotlib is best for static scientific plots, publications, and reports. Plotly is designed for interactive visualization with zoom, pan, hover, and 3D support.

Use Matplotlib for final figures and Plotly for exploration, presentations, and interactive notebooks. Both are widely used in scientific and quantum computing workflows.

6. Recommended Learning Workflow

Start with single-qubit gates and Bloch sphere visualization. Move to multi-qubit entanglement, measurement statistics, noise models, and finally quantum algorithms. Use Plotly for exploration and Matplotlib for final results.

This document consolidates Python setup, quantum basics, simulators, and visualization tools.