Week 3 - Mathematical structure of quantum theory

This notebook provides an introduction to quantum computing and quantum mechanics using Qiskit, an open-source quantum computing framework developed by IBM. The notebook covers basic quantum mechanics concepts such as superposition, entanglement, the uncertainty principle, and wave-particle duality, as well as quantum computing concepts such as qubits, quantum gates, quantum circuits, and quantum algorithms. The notebook also provides hands-on exercises using Qiskit to simulate and visualize quantum circuits and algorithms, including creating a quantum teleportation circuit and implementing the Grover's algorithm for searching an unstructured database. Overall, this notebook provides a solid introduction to the fundamentals of quantum computing and quantum mechanics using Qiskit, and is suitable for beginners with no prior knowledge in these areas.

Qiskit overview

What is Qiskit?

Qiskit is an open-source quantum computing software development kit (SDK) for building and running quantum programs. It provides tools for simulating quantum circuits, optimizing them for specific quantum hardware, and running them on real quantum devices through IBM Quantum Experience. Qiskit consists of four main components: Terra, Aer, Ignis, and Aqua.

- Qiskit Terra provides the foundational tools for working with quantum circuits, including the ability to create quantum circuits, compile them to run on specific backends, and simulate their execution using a statevector or a density matrix.
- Qiskit Aer provides high-performance simulators that can simulate the execution of quantum circuits on classical hardware. It includes a statevector simulator, a density matrix simulator, and a QASM simulator that can simulate the effects of noise on a quantum system.
- Qiskit Ignis provides tools for mitigating the effects of noise on quantum circuits, including tools for calibrating and characterizing quantum hardware, and for implementing quantum error correction and fault-tolerant protocols.
- Qiskit Aqua provides higher-level tools for solving problems in optimization, machine learning, and quantum chemistry using quantum computing.

Qiskit was first released in March 2017 by IBM, and has since become one of the most widely used quantum computing SDKs. It has been used in a variety of applications, including quantum chemistry, optimization, cryptography, and more.

Current capabilities of Qiskit

Qiskit provides a wide range of capabilities for working with quantum circuits, including:

- Support for quantum circuit design, simulation, and optimization using a variety of optimization algorithms and techniques.
- Integration with IBM Quantum Experience, which allows users to run their circuits on real quantum hardware.
- Tools for calibrating and characterizing quantum hardware, and for implementing quantum error correction and fault-tolerant protocols.
- Support for working with multipartite quantum registers and entangled states.
- High-performance simulators for simulating the effects of noise on quantum circuits.

Installing Qiskit

To use Qiskit, we first need to install it on our computer. Qiskit can be installed using pip, a package manager for Python. We recommend using a virtual environment to manage your Qiskit installation.

Here are the steps to install Qiskit on Windows:

1. Install Anaconda for Python 3.7 or higher from the Anaconda website: https://www.anaconda.com/products/individual
2. Open Anaconda Prompt from the Windows Start menu.
3. Create a new virtual environment: conda create --name myenv
4. Activate the new environment: conda activate myenv
5. Install Qiskit: pip install qiskit
6. Verify the installation: python -c "import qiskit; print(qiskit.**version**)"

To install Qiskit on macOS or Linux, the steps are similar, but you can use the terminal instead of Anaconda Prompt.

In [ ]:

```
import numpy as np
import scipy as sp
import math as mt
import matplotlib.pyplot as plt

from qiskit import *
from qiskit.visualization import *

%matplotlib inline
%config InlineBackend.figure_format = 'svg'
```

EXERCISES:

Today's exercises are about using Qiskit to simulate some very simple quantum phenomena.

Exercise 1: *Superposition principle*

Consider a spin-1/2 particle that can be in a superposition of spin-up and spin-down states. Create a quantum circuit using Qiskit that prepares the particle in the state $\frac{1}{\sqrt{2}}(1 \; i)$.

X gate: $|0\rangle \leftrightarrow |1\rangle$

H gate: $|0\rangle \leftrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle \leftrightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Z gate: $|0\rangle \leftrightarrow |0\rangle$ and $|1\rangle \leftrightarrow -|1\rangle$

For $|0\rangle$:

$|0\rangle \xrightarrow{\hat{Y}} i|1>$

For $|1\rangle$:

$|1\rangle \xrightarrow{\hat{Y}} -i|0>$

$$U(\theta,\phi,\lambda)=\begin{bmatrix}\cos(\frac{\theta}{2}) & -e^{i\lambda}\sin(\frac{\theta}{2})\\ e^{i\phi}\sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)}\cos(\frac{\theta}{2})\end{bmatrix}$$

$$|\psi\rangle=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$$

$$A|\psi\rangle=\frac{1}{\sqrt{2}}(|0\rangle+i|1\rangle)\Leftrightarrow A=\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$$

In [ ]:

```
from qiskit import QuantumCircuit, assemble, Aer
from qiskit.visualization import plot_histogram, plot_bloch_vector
from qiskit.quantum_info import Statevector
import math

state0 = Statevector.from_int(0, 2)
state1 = Statevector.from_int(1, 2)
state2=(state0+state1)/(math.sqrt(2))
qc=QuantumCircuit(1)
qc.u(0,0,np.pi/2,0)
state0hu=state2.evolve(qc)
state0hu.draw('latex')
```

Out[ ]:

$$\frac{\sqrt{2}}{2}|0\rangle+\frac{\sqrt{2}i}{2}|1\rangle$$

Exercise 2: *Uncertainty principle*

Suppose we have a qubit in a superposition of states $|0\rangle$ and $|1\rangle$, given by the state $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$. We want to measure the position and momentum of the qubit. However, due to the uncertainty principle, we cannot measure both properties with arbitrary precision.

Calculate the position and momentum operators for the qubit. Calculate the expectation values of the position and momentum operators for the given state. Calculate the variances of the position and momentum operators for the given state. Show that the product of the variances is greater than or equal to $\frac{1}{4}$, which demonstrates the uncertainty principle.

The position operator for a qubit is given by

$$X=|0\rangle\langle 0|-|1\rangle\langle 1|$$

and the momentum operator is given by

$$P=\frac{1}{2\pi}\int dxe^{-ipx}|x\rangle\langle x|.$$

Operators

Position operator

$$X = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Momentum operator

$$P = \begin{pmatrix} e^{-ip} & 0 \\ 0 & e^{ip} \end{pmatrix}$$

Expectation values

Position operator

$$\langle\psi|X|\psi\rangle = \frac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 0$$

Momentum operator

$$\langle\psi|P|\psi\rangle = \frac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} e^{-ip} & 0 \\ 0 & e^{ip} \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = (e^{-ip} + e^{ip})/2 = \cos(p)$$

Variances

Position operator

$$\langle\psi|X^2|\psi\rangle = \frac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = 1$$

$$\Delta X = \sqrt{\langle X^2\rangle - \langle X\rangle^2} = 1$$

Momentum operator

$$\langle\psi|P^2|\psi\rangle = \frac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix}\begin{pmatrix} e^{-2ip} & 0 \\ 0 & e^{2ip} \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \cos(2p)$$

$$\Delta P = \sqrt{\langle P^2\rangle - \langle P\rangle^2} = \sqrt{\cos(2p) - \cos^2(p)} = \sqrt{(\cos(2p) - 1)/2}$$

Heisenberg Uncertainty

$$\Delta X \Delta P = \sqrt{(\cos(2p) - 1)/2}$$

Exercise 3: *Wave-particle duality*

Unfortunately, it is not possible to simulate the double-slit experiment using Qiskit or any other quantum computing framework as it requires a physical setup and experimental apparatus. The double-slit experiment is a classic example of wave-particle duality and demonstrates the interference pattern produced by a wave-like behavior of particles.

However, we can still simulate a simplified version of the double-slit experiment using Qiskit to demonstrate the concept of interference.

Consider a quantum circuit with two qubits in an equal superposition state, and apply a Hadamard gate to each qubit. Then, apply a CNOT gate with the first qubit as the control and the second qubit as the target. Finally, measure both qubits. What is the expected outcome of the measurement, and what does it demonstrate?

$$\psi_1 = |0\rangle \xrightarrow{\hat{H}} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$\psi_2 = |0\rangle \xrightarrow{\hat{H}} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

They result in

$$\Psi = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

With the first qubit as the control and the second as the target we get

$$CNOT\Psi = \frac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |01\rangle)$$

I expect the same number of measurements for each outcome

```
In [ ]: qc=QuantumCircuit(2)
        qc.h(0)
        qc.h(1)
        qc.cnot(0,1)
        qc.measure_all()
        sim=Aer.get_backend("qasm_simulator")
        job=execute(qc,sim,shots=10000)
        result=job.result()
        counts=result.get_counts()
        print(counts)
```

```
{'11': 2465, '00': 2521, '10': 2491, '01': 2523}
```

This uses the same concept of the DSE, because here we have the same superposition state interacting with "itself", interaction between $\hat{H}|\psi_1\rangle$ and $\hat{H}|\psi_2\rangle$ and changing the final outcomes

Today's afterthoughts:
Qiskit is based on "digital" quantum states, a natural extension of the classical bit, but you have probably heard about "continuous variable states". What are your opinions about using these systems to implement a quantum computer?

To implement an extension of a countinous variable state would require an even greater complexity to the system, but even a countinous variable would be a discrete one in the real world.