Introduction to Quantum Computing

1. **All About Qubits**
   Qbit is a basic build block of a quantum computing. Unlike classical bits qbit is either 0 or 1. A voltage passing through a electronic circuit can be treated as a qubit. Any voltage greater than threshold sets the circuit to be in state 1. And if the voltage is below the threshold the circuit can be set to be in state 0. Mathematically the qubit is represented in vector notation. State 0 is [1,0] and state 1 is [0,1]. These states are always normalized. To perform meaningful computation, we need to know the information about the state of the qubit at the end of the algorithm. This is called measurement; measurements are probabilistic in quantum computing.
   Prob(measuring state 0) is |a| * |a|
   Prob(measuring state 1) is |b| * |b|
   We can perform operations on qubits in the form of qubits. Every operations takes the qubit to different state.

2. **Quantum Circuits**
   Quantum circuits are pictorial representation of operations that are performed on qubits. A circuit is sequence operations performed on qubit. In pennylane a circuit starts with set of wires represents qubits. A group of registers are called Quantum register. Each operation on the qubit is called a gate. There are many different types of gates which have different effects on qubits. Every gate will take qubit to a different state, few gates operate on the single qubit and few of them operate on combination of qubits.

   Quantum circuit in pennylane is defined by a quantum function. These are regular python functions that do special operations by applying gates and return one or more measurements.

   Pennylane requires a device to run a circuit on and a QNode that binds the quantum function with the device.

3. **Unitary Matrices**
   A single quit operation takes a qubit from present state to other valid state, this is done by using matric-vector multiplication by 2 * 2 matrix. Thus, a valid state is always normalized, and the structure have to preserve all its quantum properties the operation has to be unitary. And gates of this type are called unitary matrices.
   Unitary matric property is UU’dag = U’dag U = I

Single Qubit Gates

1. **X and H gates**
   X and H are very important gates using which we can explore important aspects of Quantum Computing. X is bit flip gate like NOT gate, X gate bits Quantum state. X applied to state 0 moves state to state 1 and vice versa. Hadamard gate will put the Quantum state into qual superposition of 1 and 0.

X|1> -> |0>
X|0> -> |1>
H|0> -> |+>
H|1> -> |->

2. Qubit rotations and Global Phases

Let's consider an arbitrary quantum state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

and separate out the real and complex components of the amplitudes by writing them in polar form, i.e., $\alpha = ae^{i\theta}, \beta = be^{i\varphi}$. We can factor out the complex part:

$$|\psi\rangle = ae^{i\theta}|0\rangle + be^{i\varphi}|1\rangle = e^{i\theta}(a|0\rangle + be^{i(\varphi-\theta)}|1\rangle). \tag{2}$$

Notice how the term $e^{i\theta}$ out front doesn't affect the measurement outcome probabilities *at all*! Without loss of generality, we can totally ignore this **global phase**, and describe exactly the same quantum state:

$$|\psi\rangle = a|0\rangle + be^{i(\varphi-\theta)}|1\rangle = a|0\rangle + be^{i\phi}|1\rangle. \tag{3}$$

This remaining complex value, $e^{i\phi}$ is known as a **relative phase**. If you look at the measurement outcome probabilities of $|0\rangle$ and $|1\rangle$, you might notice that this phase doesn't affect them either; we will learn later, though, that it *can* affect the measurement outcomes if the measurements are performed in a different way.

3. Rx and Ry Rotations

Now that we have some intuition about what these rotations are, we can begin to express RX and RY in terms of matrices and describe how they act on the basis states.

The matrix representation of an X rotation is

$$RX(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \tag{3}$$

The matrix representation of a Y rotation looks very similar to that of RX, however there is no complex component:

$$RY(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \tag{4}$$

4. Universal Gate Sets

Recall that the most general expression of a single-qubit unitary matrix looks like this:

$$U(\phi, \theta, \omega) = \begin{pmatrix} e^{-i(\phi+\omega)/2}\cos(\theta/2) & -e^{i(\phi-\omega)/2}\sin(\theta/2) \\ e^{-i(\phi-\omega)/2}\sin(\theta/2) & e^{i(\phi+\omega)/2}\cos(\theta/2) \end{pmatrix}. \tag{1}$$

It's possible to find a set of angles ϕ, θ, ω for all the gates we encountered in previous nodes, such that $U(\phi, \theta, \omega)$ is equivalent to the gate up to a global phase. However, U can actually be expressed in a simpler way.

5. **Preparations and Measurements**
   Having studied unitary operations, we've now learned about how quantum computers are manipulated. But if we want to tackle a real problem, we need a way of getting information out of the system after we've done something to it! We need methods to measure our qubits and interpret those results so that we can relate them back to the problem at hand.

   In earlier nodes, you may recall that we alluded to the idea of measurements when discussing superposition, and the probability of different measurement outcomes after applying certain quantum operations. In the next two sections, we will formalize and expand on those ideas. First, we will explore projective measurements.

Circuits with Multiple Qubits

In PennyLane, qubits are indexed numerically from left to right. Therefore, a state such as indicates that the first and third qubit (or, wires `0` and `2`) are in state and the second, fourth, and fifth qubit are in state When drawing quantum circuits, our convention is that the leftmost (first) qubit is at the *top* of the circuit, such that qubits starting in state 0.

When we put more than one qubit together, we must learn how these vector spaces are composed. Hilbert spaces are combined using an operation called the tensor product. This operation is best understood through examples. Suppose we have a pair of two-dimensional vectors (e.g., two single-qubit states).

The most straightforward type of multi-qubit operation is separable. This is what you've been doing so far: every qubit got acted on by single-qubit gates, and there were no interactions between them.

We now come to a very important property of quantum systems: entanglement. Along with superposition, entanglement is one of the two hallmark features of quantum computing that underscore its advantage. Entanglement is used as a resource in many quantum algorithms, including quantum teleportation.

The following two-part exercise will show you what it means for a multi-qubit state to be entangled. Earlier, we learned about universal gate sets for single-qubit operations. What about for multi-qubit operations? It turns out that we just need one more gate: the The sets and are both universal gate sets for multi-qubit computation. Isn't that amazing?

While we could do everything with just the it is just one of many possible controlled operations. Any operation, on any number of qubits, can be implemented as a controlled operation, controlled on the state of one or more other qubits.