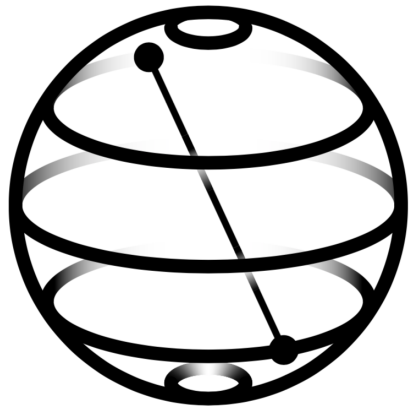
Get Started with Quantum Computing and Qiskit

Matthew Treinish
Software Engineer - IBM Research
mtreinish@kortar.org
`mtreinish on Freenode`
https://github.com/mtreinish/open-source-quantum-computing/tree/HVOpen

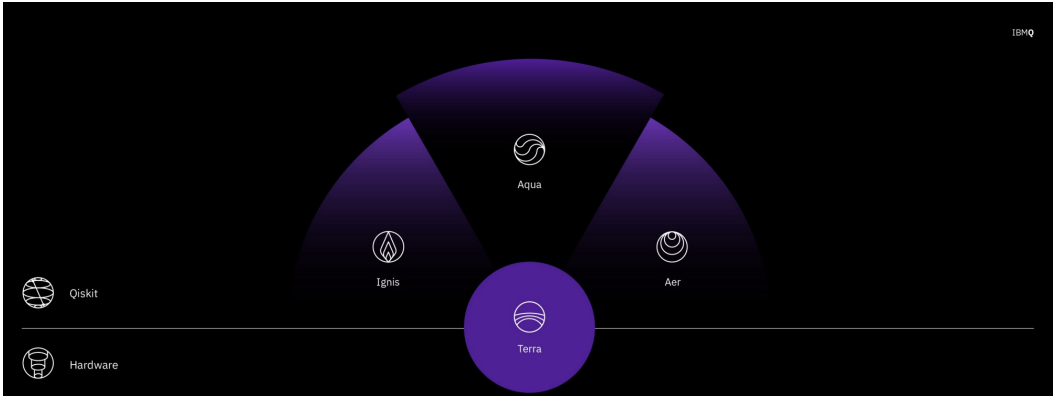
March 15, 2019

What is Qiskit?

- SDK for working with Noisy Intermediate-Scale Quantum (NISQ) computers
- Apache 2.0 License
- Designed to be backend agnostic
- Includes out-of-the-box local simulators and support for running on IBMQ



Qiskit Elements



Terra

Terra is a collection of core, foundational tools for communicating with quantum devices and simulators. Users can write quantum circuits, and address real hardware constraints with Terra. Its modular design simplifies adding extensions for quantum circuit optimizations and backends.

Ignis

Controlling fire was a turning point in human evolution. Learning how to fix or control quantum errors will be a turning point in the evolution of quantum computing. Users can access better characterization of errors, improve gates, and compute in the presence of noise with Ignis. It is designed for researching and improving errors or noise in near-term quantum systems.

Aqua

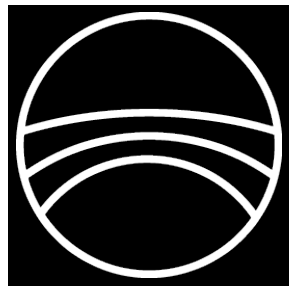
Aqua is a modular and extensible library for experimenting with quantum algorithms on near-term devices. Users can build domain-specific applications, such as chemistry, AI and optimization with Aqua. It bridges quantum and classical computers by enabling classical programming to run on quantum devices.

Aer

Aer permeates all other Qiskit elements. Users can accelerate their quantum simulator and emulator research with Aer, which helps to better understand the limits of classical processors by demonstrating their ability to mimic quantum computation. Users can also verify current and near-term quantum computer functionality with Aer.

Qiskit Terra

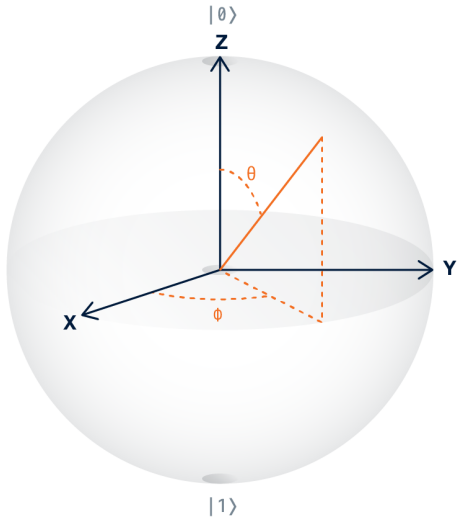
- Is the base layer for applications, provides interface to hardware and simulators
- Provides an SDK for working with quantum circuits
- Compiles circuits to run on different backends
- Written in Python



The Qubit

- The bloch sphere provides a representation of qubit state
- State can be at any point along surface of sphere
- Measuring a qubit occurs along the Z axis. (also called basis states)
- Measuring a qubit is irreversible and will either be 0 or 1

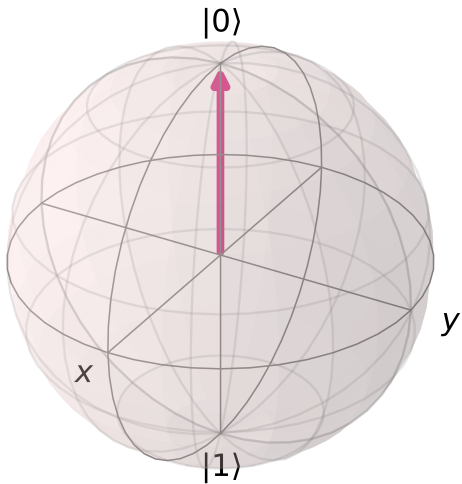
Bloch Sphere:



The Qubit

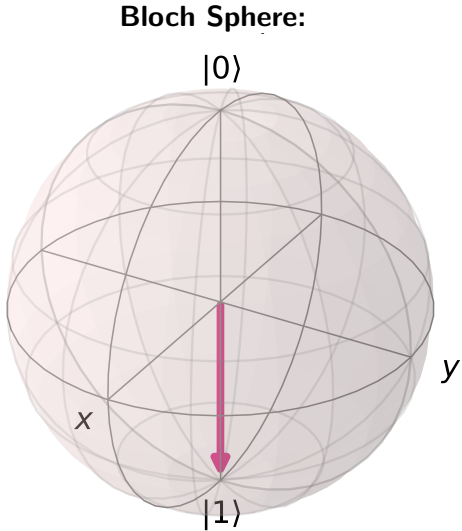
- The bloch sphere provides a representation of qubit state
- State can be at any point along surface of sphere
- Measuring a qubit occurs along the Z axis. (also called basis states)
- Measuring a qubit is irreversible and will either be 0 or 1

Bloch Sphere:



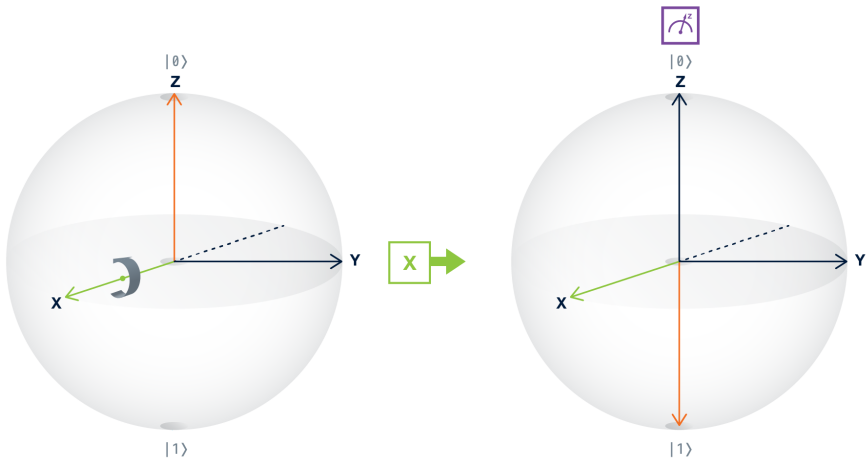
The Qubit

- The bloch sphere provides a representation of qubit state
- State can be at any point along surface of sphere
- Measuring a qubit occurs along the Z axis. (also called basis states)
- Measuring a qubit is irreversible and will either be 0 or 1



Quantum Gates

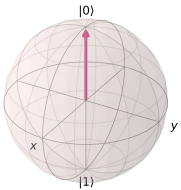
- Quantum Logic Gates are used perform operations on qubits
- Gates are reversible
- Gates can be represented as unitary matrices



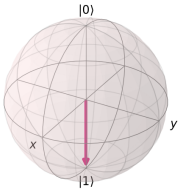
Superposition

- Identically prepared qubits can still behave randomly
- The randomness is inherent in nature

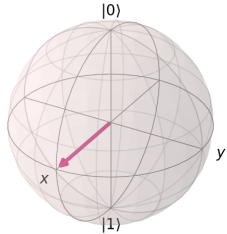
$|0\rangle$



$|1\rangle$

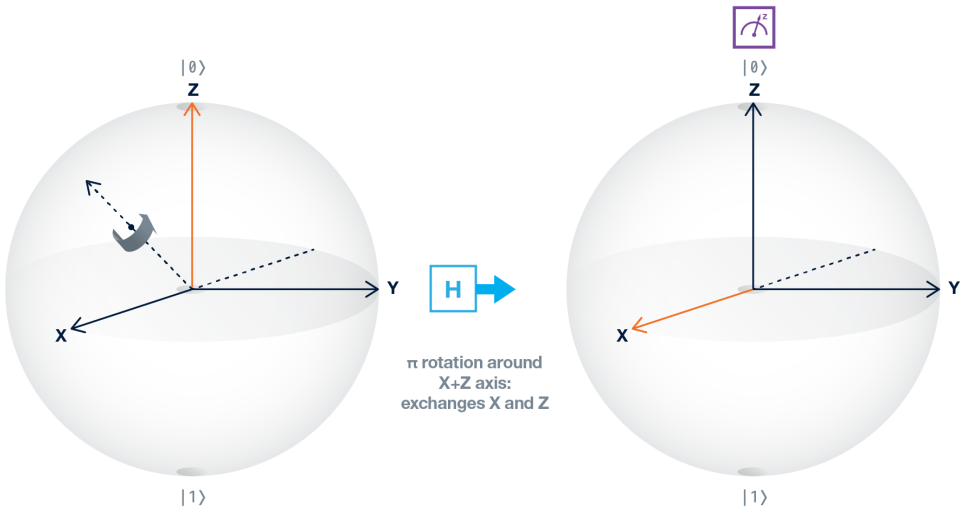


$|0\rangle + |1\rangle$



$\sim$50/50 chance of being $|0\rangle$ or $|1\rangle$

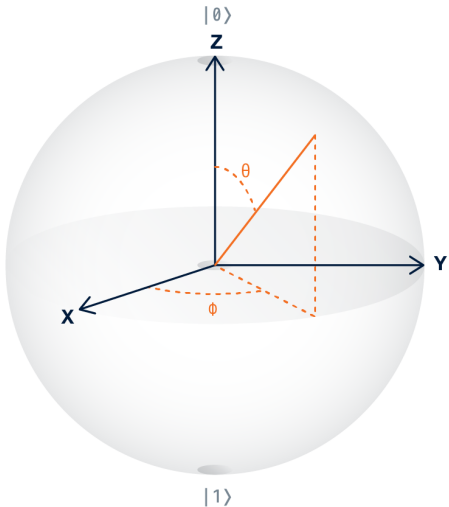
Hadamard Gate



Qubit Phase

- While qubits are read along the basis vectors you can still use the other dimensions
- The phase can be leveraged to encode more information in the qubit

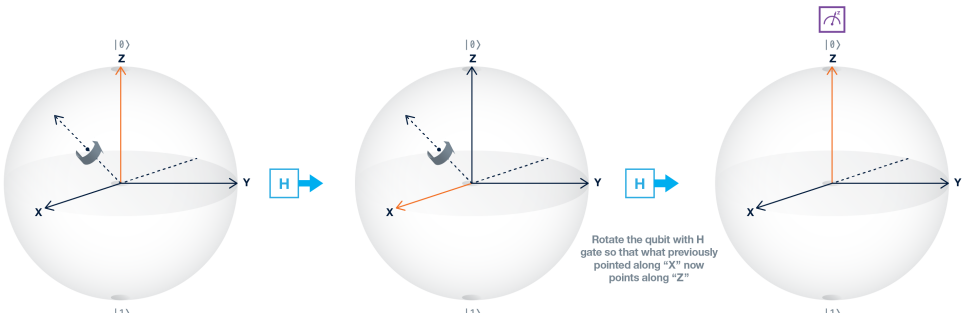
Phase is ϕ:



Hadamard and Phase

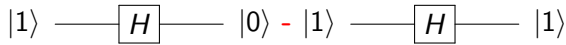
Hadamard gates are self inverses:

$$|0\rangle \;\text{—}\boxed{H}\text{—}\; |0\rangle + |1\rangle \;\text{—}\boxed{H}\text{—}\; |0\rangle$$

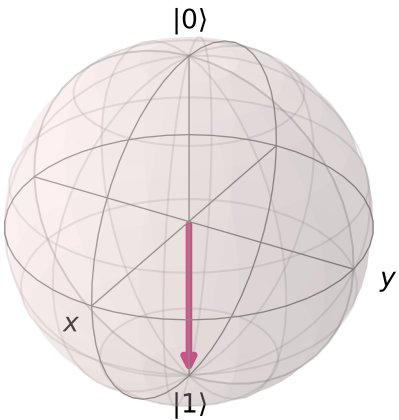


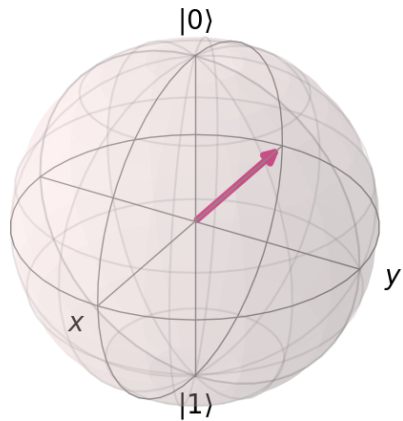
Hadamard and Phase

Hadamard gates are self inverses:



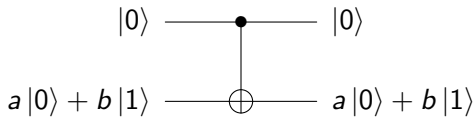
Phase

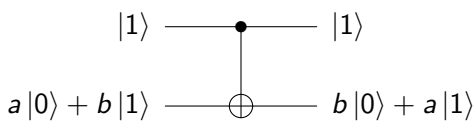




Controlled Not Gate

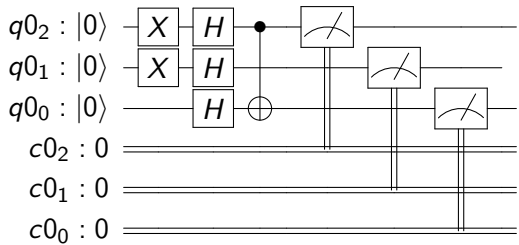
CNOT flips the *target* bit if the *control* bit is 1





Quantum Circuits

Putting it together you build a circuit like:



- Each row represents a bit, either quantum or classical
- The operations are performed each qubit left to right
- Shows dependencies of operations