How Does Shor's Algorithm Work?

Benjamin Tang

June 2026

Outline

The outline of this talk is as follows.

- What Shor's Algorithm Does
- The Qubit
- Quantum Gates
- Quantum Phase Estimation
- Shor's Algorithm

What Does Shor's Algorithm Do?

Shor's algorithm is a quantum algorithm that "quickly" finds the order x modulo N.

This has major implications in cryptography.

To understand how Shor's algorithm works, we need some prerequisites. We will first explain the qubit. Then, we explain the quantum gate that operates on qubits. Next, we will explain quantum phase estimation, which is used by Shor's algorithm. Finally, we will explain how Shor's algorithm works.

Defining the Qubit

Qubits are the quantum analogy of classical bits. Like classical bits, they can be in states $|0\rangle$ or $|1\rangle$. However, they can be in a **superposition** of those states. That is, they can be written in the form

$$\alpha |0\rangle + \beta |1\rangle$$

where α and β are **complex numbers**, representing an **amplitude**.

The qubit is neither $|0\rangle$ nor $|1\rangle$, but a mix of both!

Qubits cont.

If a qubit is in the $|0\rangle$ state, it is equivalent to a classical bit in the 0 state.

Similarly, if a qubit is in the $|1\rangle$ state, it is equivalent to a classical bit in the 1 state.

Intuitively, if the qubit is in a superposition ("mix") of two states, the coefficients for each state cannot be that large (e.g. we cannot have $67\,|0\rangle + 1434i\,|1\rangle$ as the state of our qubit). Instead, they must obey the normalization principle. That is

$$|\alpha|^2 + |\beta|^2 = 1.$$

Visualizing The Qubit

With this normalization principle in mind, we can visualize a qubit as a point on a sphere.

An example for a qubit in the $|0\rangle$ state is shown below.

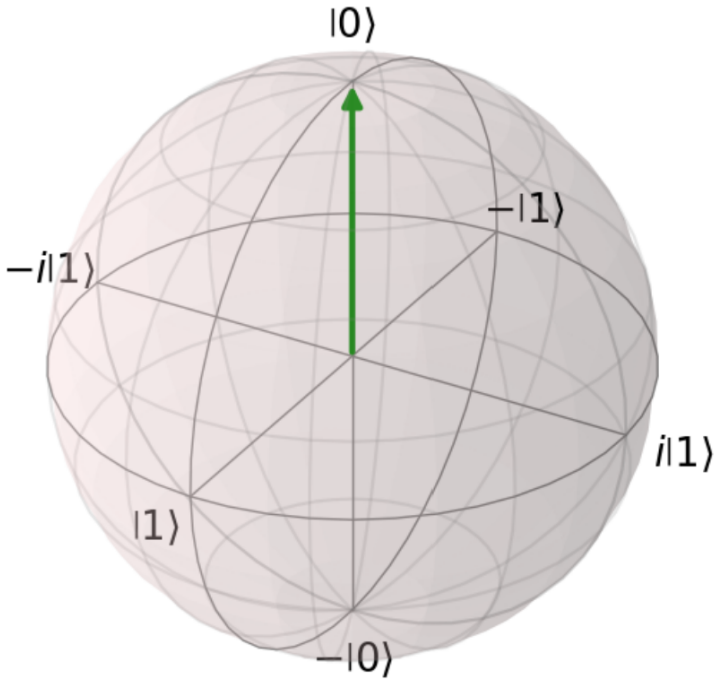

Figure. Qubit Visualization (not the Bloch sphere)

Operating on a Qubit

Shor's algorithm uses quantum gates to achieve its goal. Therefore, we explain quantum gates.

As we know, classical gates operate on classical bits, with some examples of classical gates being the AND gate, the OR gate, the NOT gate, etc.

There are quantum analogs to the classical gates above.

We introduce an important quantum gate in the next slide.

The Hadamard Gate

A **Hadamard gate** is the primary gate used to establish superposition. It is characterized by what it does to the states $|0\rangle$ and $|1\rangle$.

$$|0\rangle \xrightarrow{H} \frac{|0\rangle+|1\rangle}{\sqrt{2}},$$
$$|1\rangle \xrightarrow{H} \frac{|0\rangle-|1\rangle}{\sqrt{2}}.$$

Example

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{H} \alpha\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right) + \beta\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) = \frac{(\alpha+\beta)|0\rangle+(\alpha-\beta)|1\rangle}{\sqrt{2}}.$$

Visualizing The Hadamard

We can visualize the Hadamard gate as the reflection of a point on the sphere about the line containing $(\sin(\frac{\pi}{8}), 0, \cos(\frac{\pi}{8}))$ and the origin.

An animation for the Hadamard gate is shown in the next slide. The blue vector is the line of reflection.

Visualizing The Hadamard

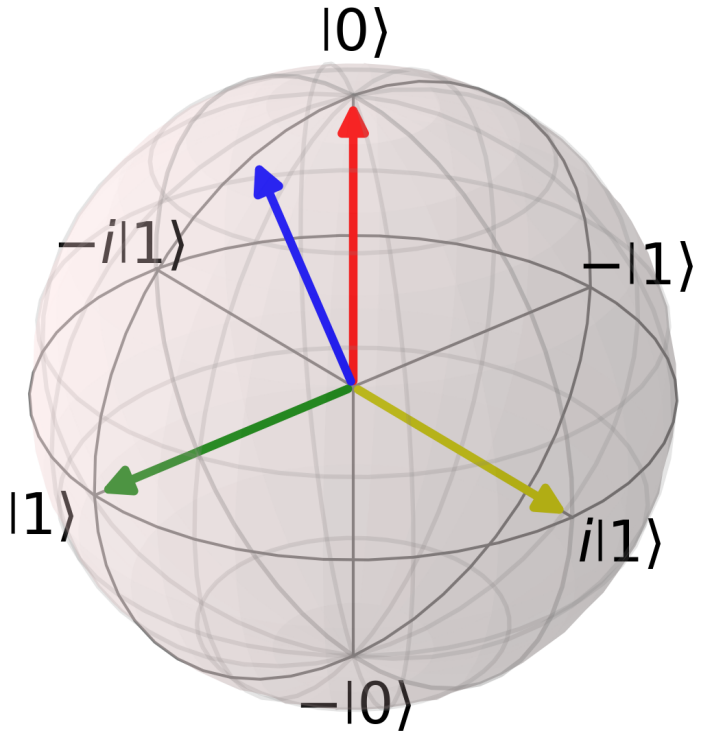

Figure. Hadamard Animation: Initial State

Visualizing The Hadamard

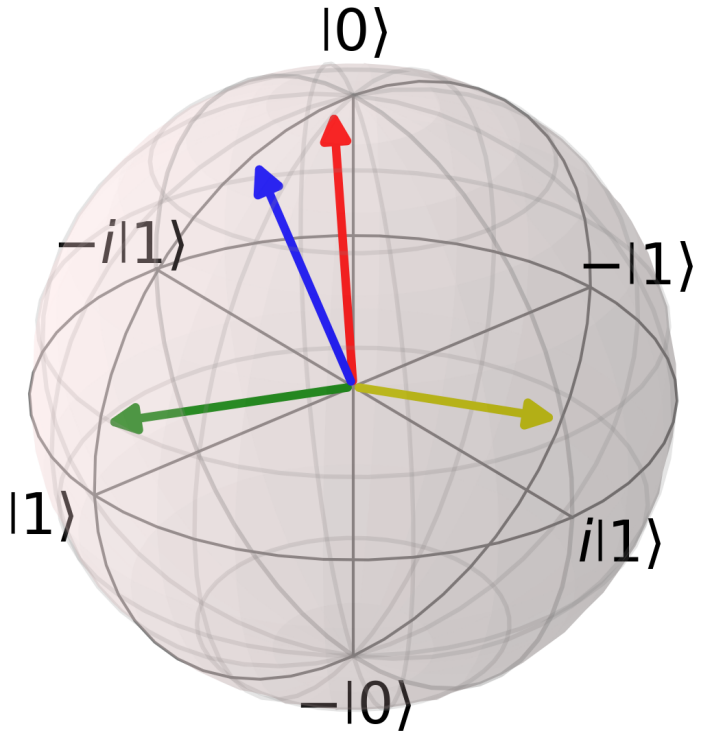

Figure. Hadamard Animation

Visualizing The Hadamard

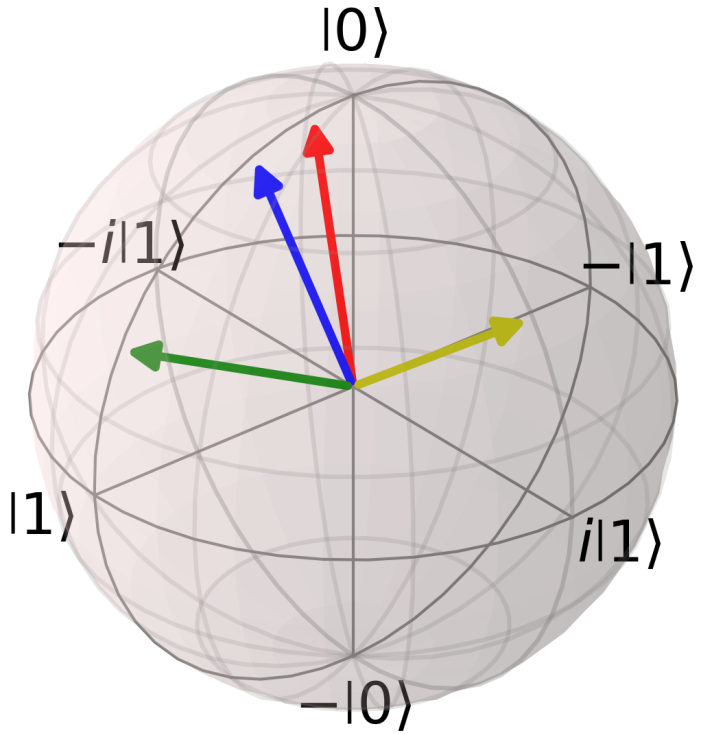

Figure. Hadamard Animation

Visualizing The Hadamard

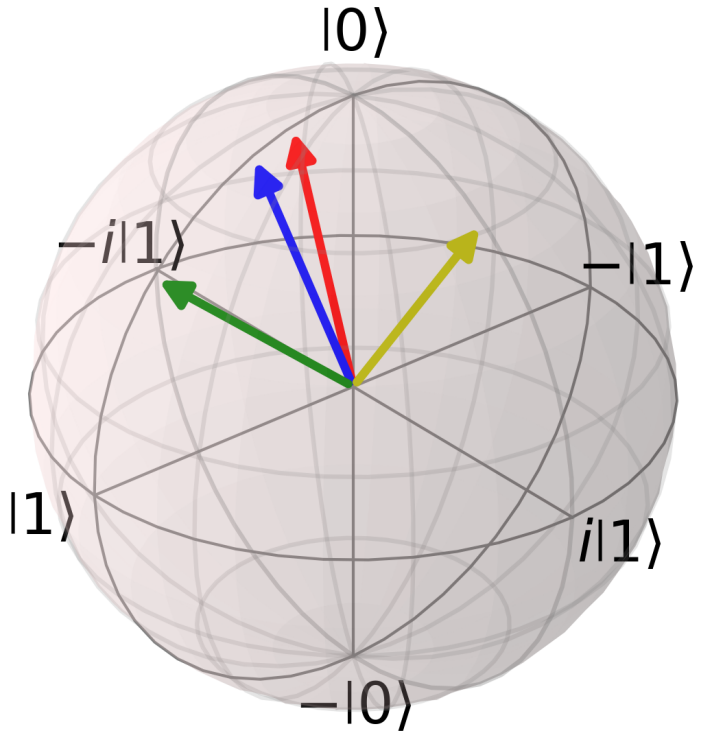

Figure. Hadamard Animation

Visualizing The Hadamard

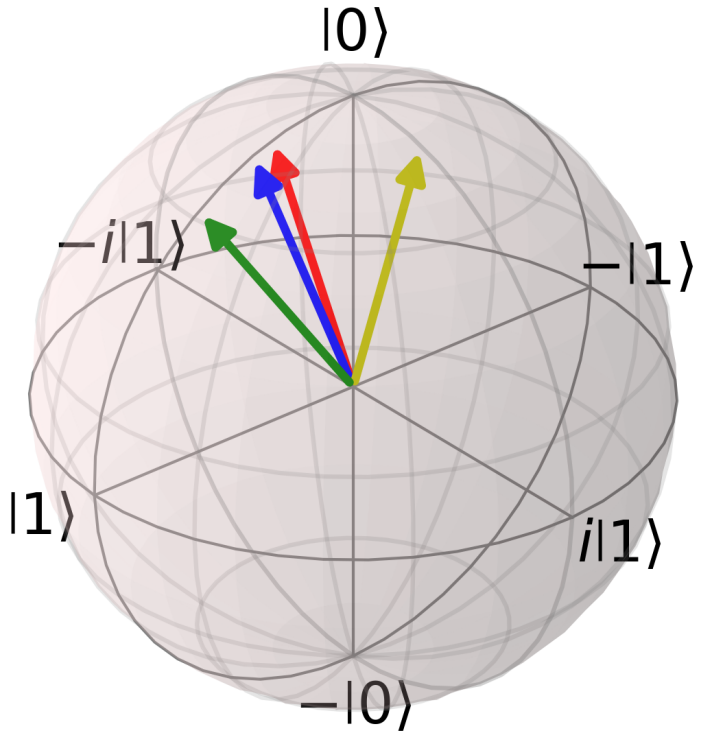

Figure. Hadamard Animation

Visualizing The Hadamard

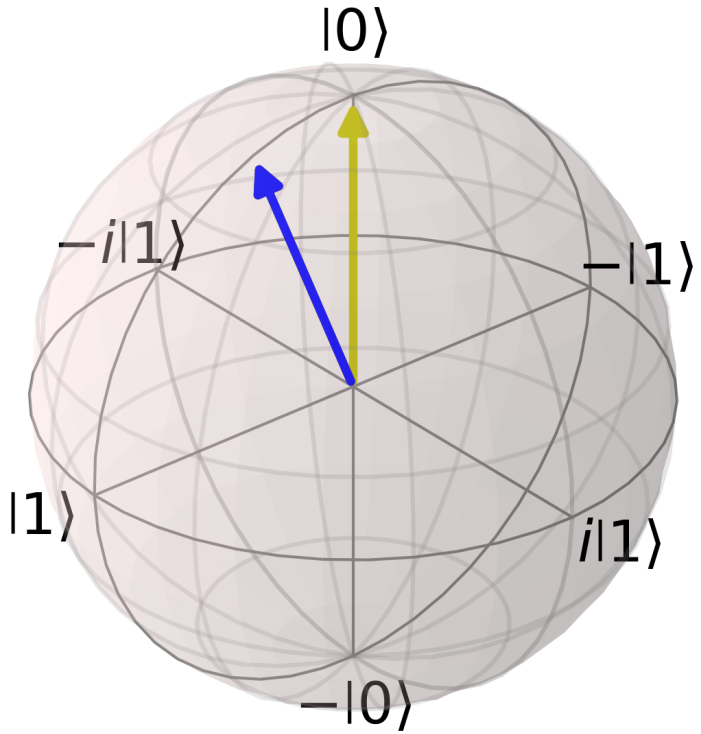

Figure. Hadamard Animation

Visualizing The Hadamard

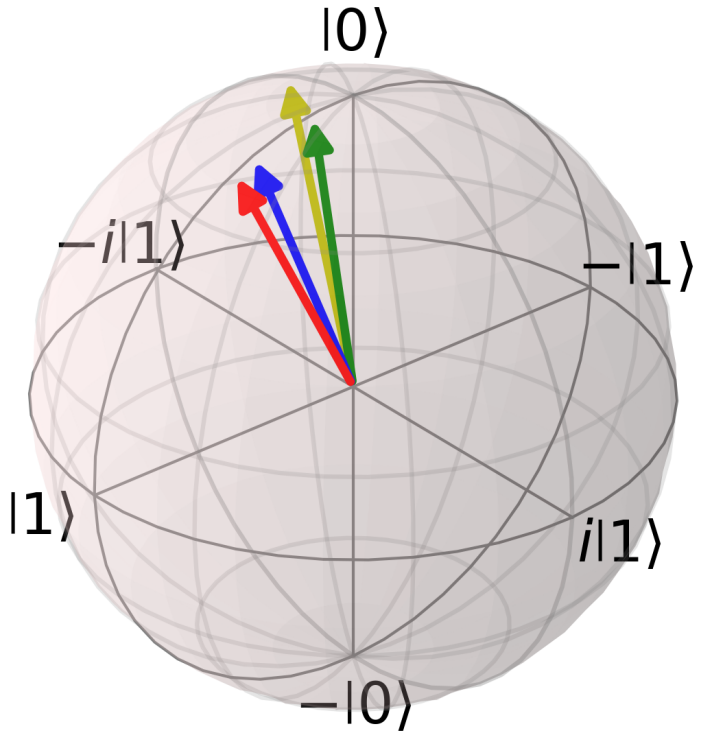

Figure. Hadamard Animation

Visualizing The Hadamard

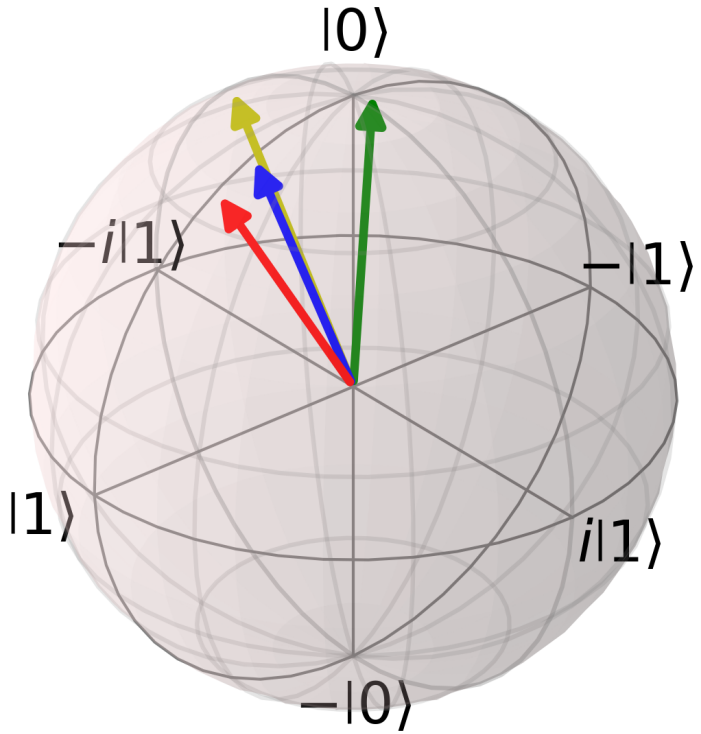

Figure. Hadamard Animation

Visualizing The Hadamard

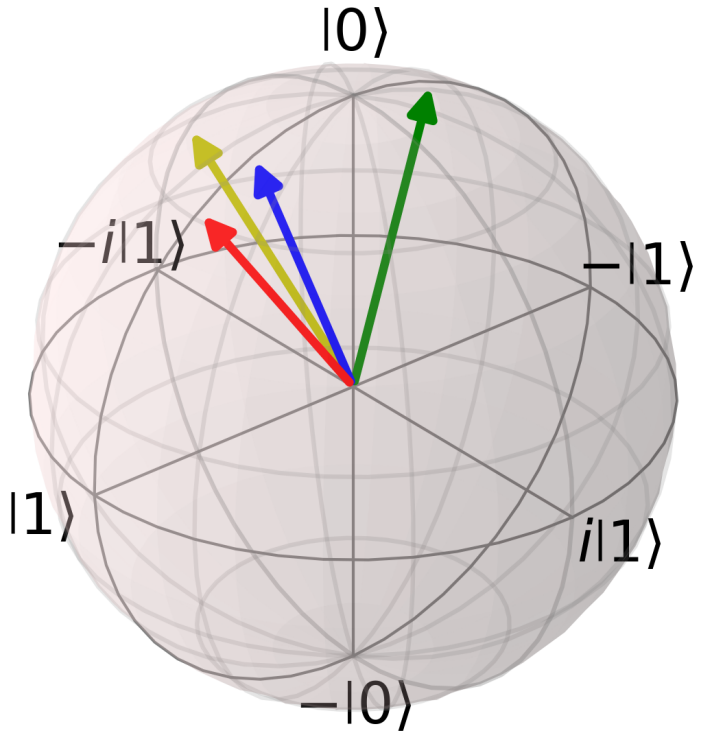

Figure. Hadamard Animation

Visualizing The Hadamard

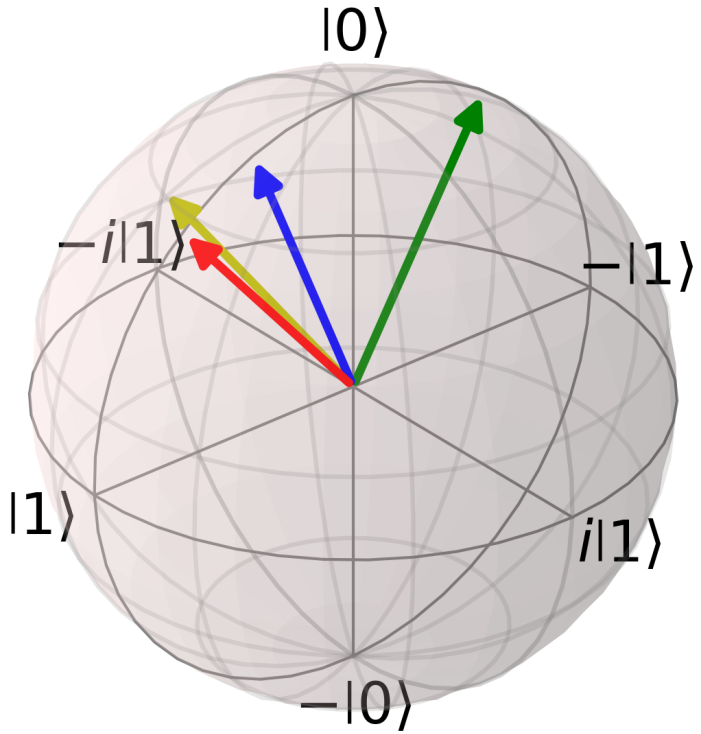

Figure. Hadamard Animation

Visualizing The Hadamard

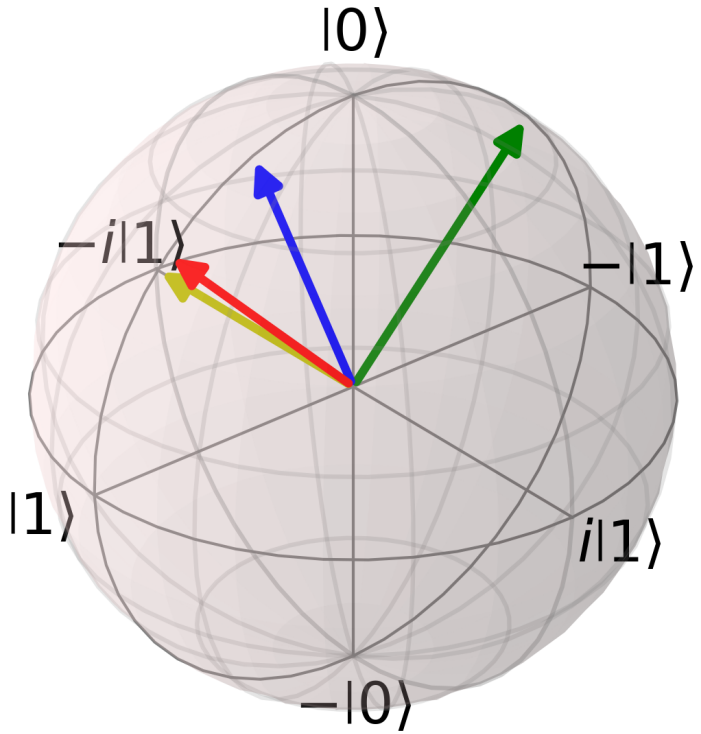

Figure. Hadamard Animation: Final State

Operating On Quantum States

Not every function can operate on a quantum state. This is because of our normalization condition.

We can visualize a quantum gate as an operation that maps points on the unit sphere to points on the unit sphere. Thus, a quantum gate that preserves the normalization condition must be a rigid **rotation** or **reflection** on the sphere.

Why Quantum Computers May Be Faster Than Classical Computers At Some Tasks

There are some reasons why quantum computers may be faster than classical computers.

- Qubits can be in **superposition**, not fixed as either a $|1\rangle$ or a $|0\rangle$.
- **Amplitudes**, not probability, add. Therefore, constructive and destructive interference are possible!

Quantum Phase Estimation (Approximately)

One such task for which a quantum computer is fast is **phase estimation**, which is used by Shor's algorithm. That is, we have a quantum gate that rotates the qubit by an unknown angle θ about the z-axis on the sphere.

Then, the algorithm of **quantum phase estimation** can return the phase θ to a high degree of accuracy.

Quantum Phase Estimation (Approximately): cont

We can visualize the algorithm for **quantum phase estimation**. We plot n qubits on the unit sphere (the more qubits, the more accurate the measurement). The gate is applied a distinct power of 2 times for each qubit.

The animation for the rotation of the n qubits is shown in the next slide.

Overlays

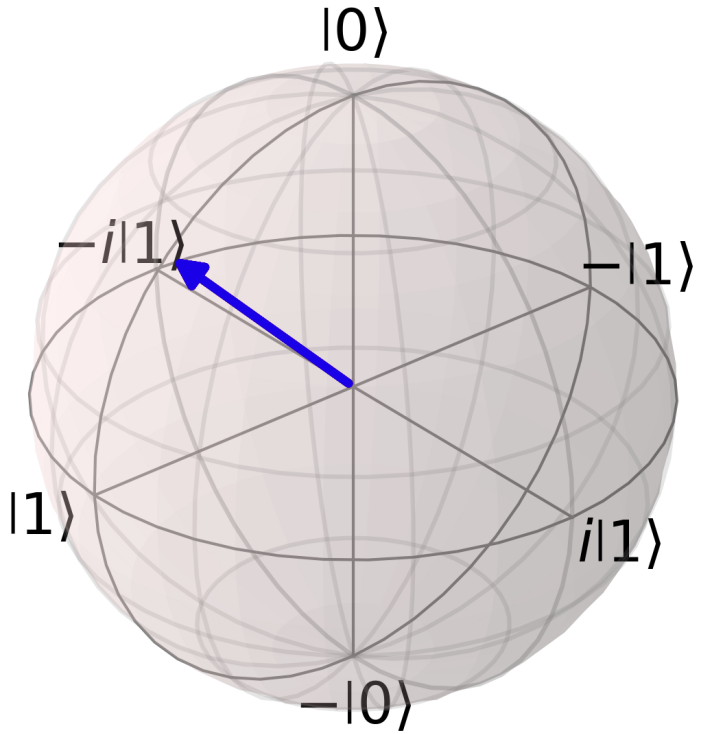

Figure. Quantum Phase Estimation: Initial State

Overlays

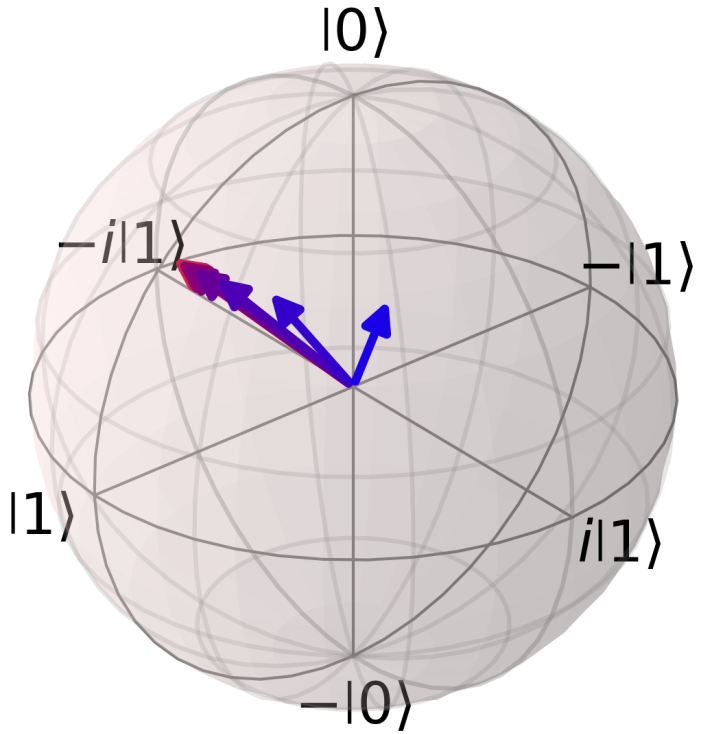

Figure. Quantum Phase Estimation

Overlays

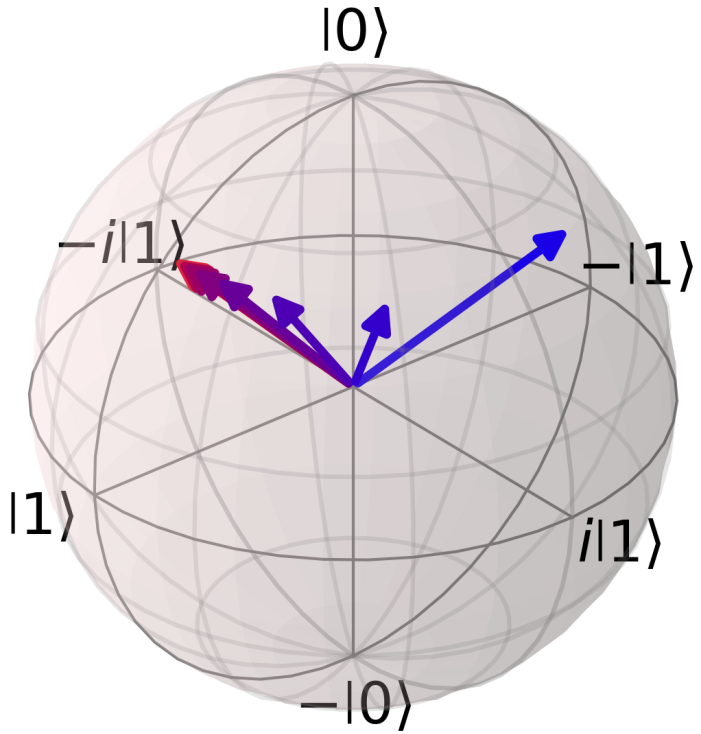

Figure. Quantum Phase Estimation

Overlays

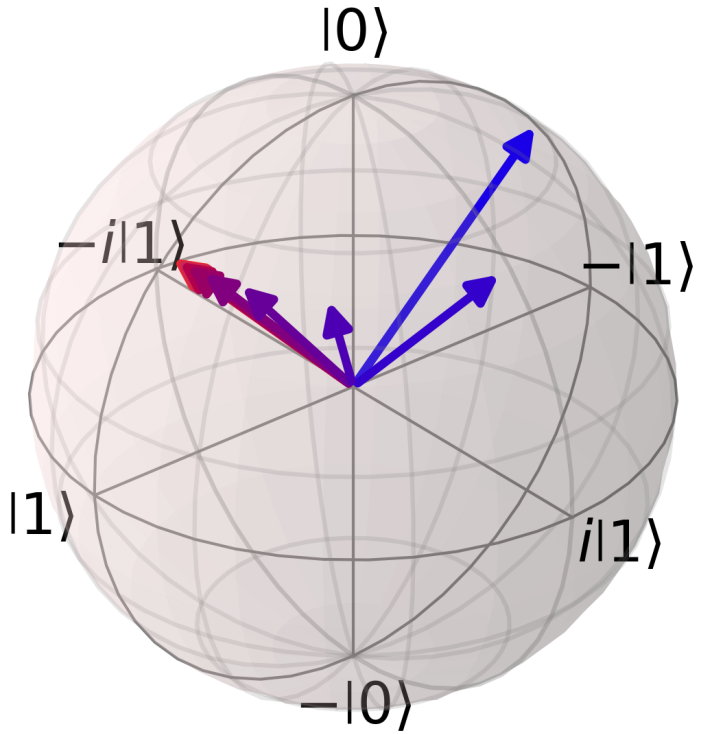

Figure. Quantum Phase Estimation

Overlays

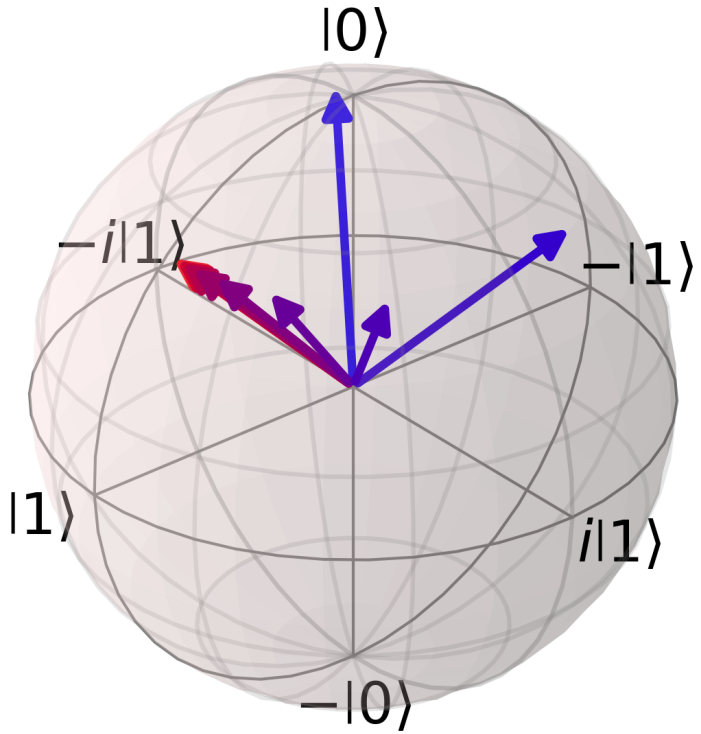

Figure. Quantum Phase Estimation

Overlays

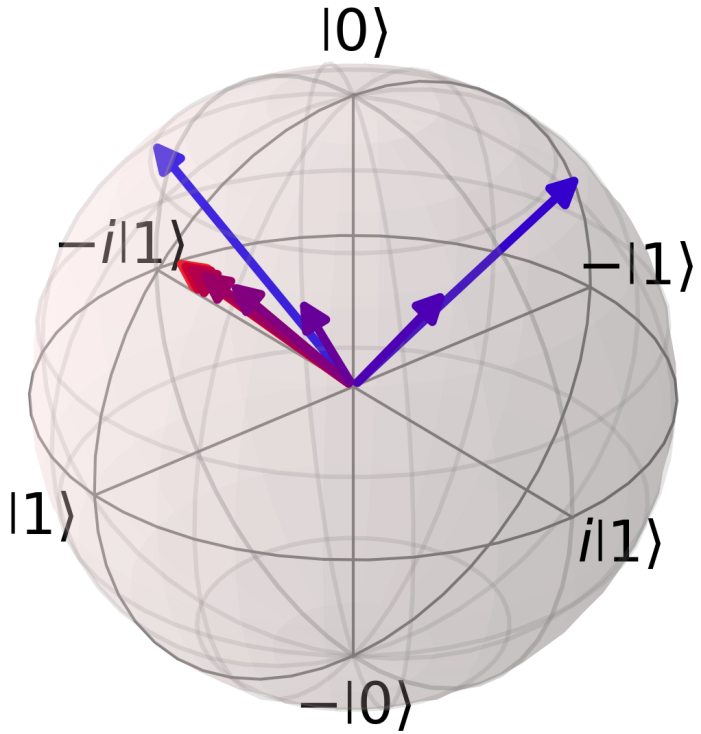

Figure. Quantum Phase Estimation

Overlays

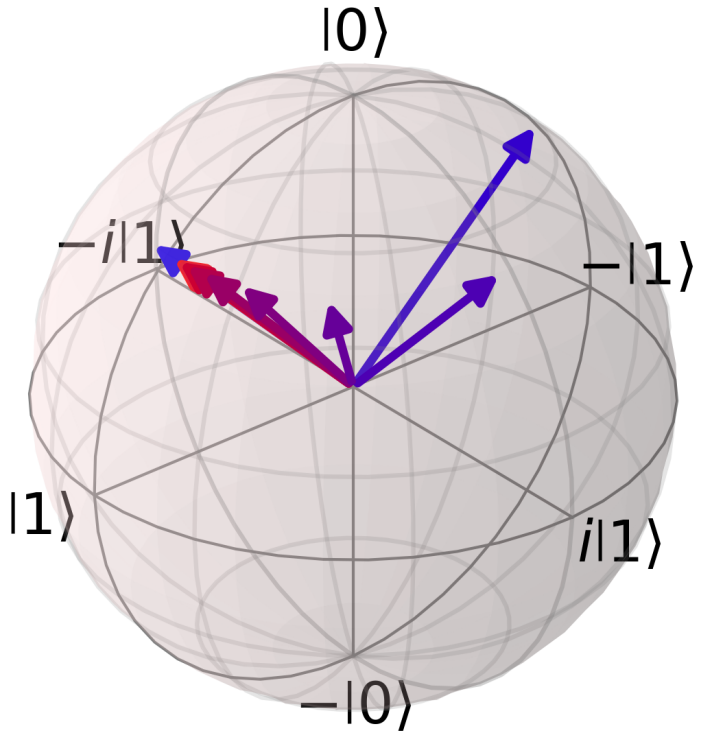

Figure. Quantum Phase Estimation

Overlays

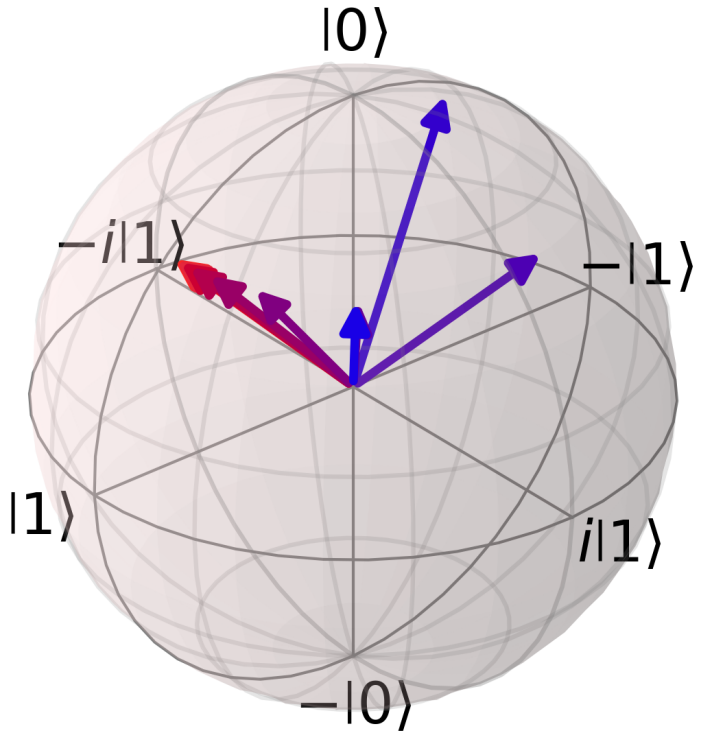

Figure. Quantum Phase Estimation

Overlays

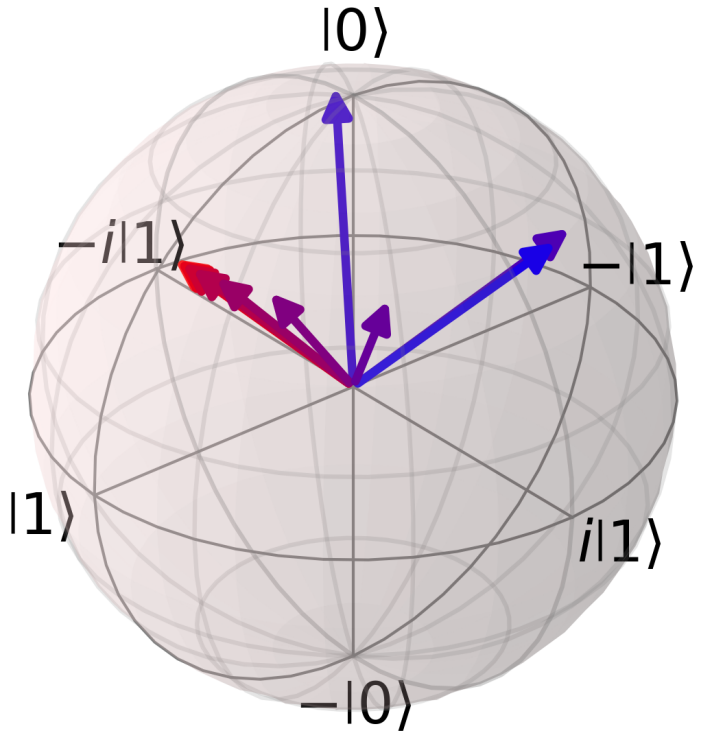

Figure. Quantum Phase Estimation

Overlays

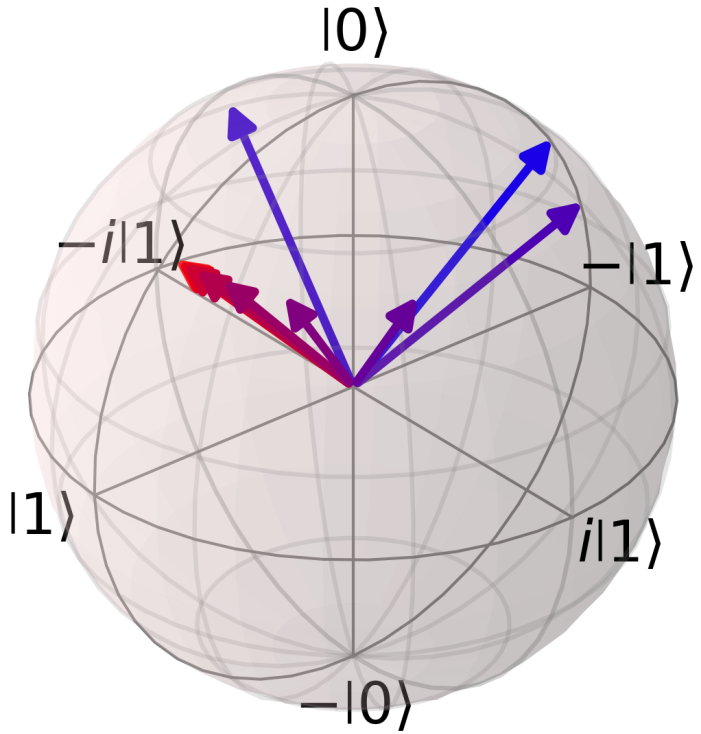

Figure. Quantum Phase Estimation

Overlays

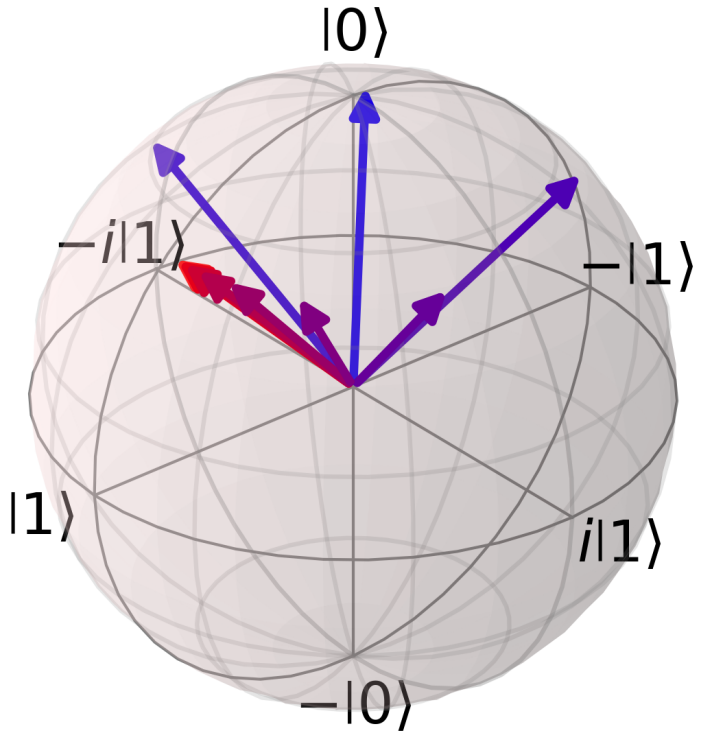

Figure. Quantum Phase Estimation

Overlays

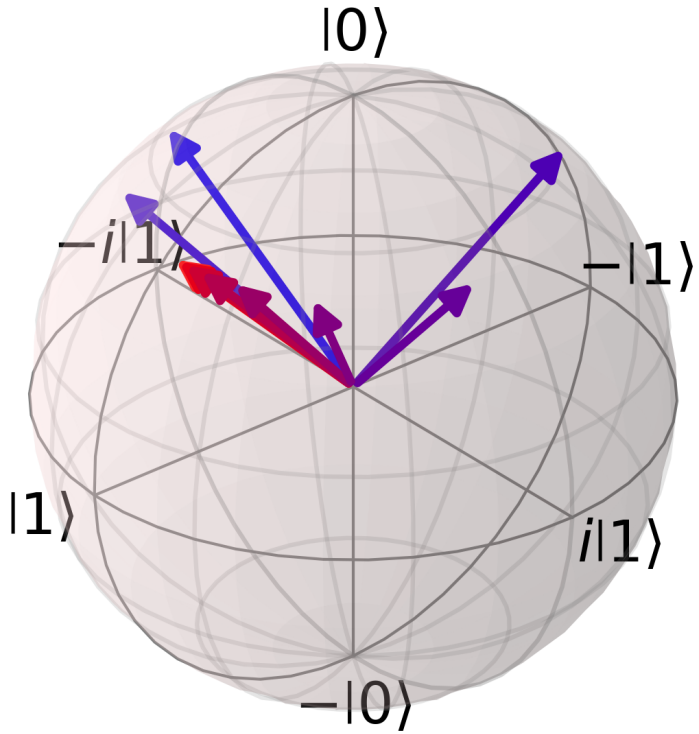

Figure. Quantum Phase Estimation

Overlays

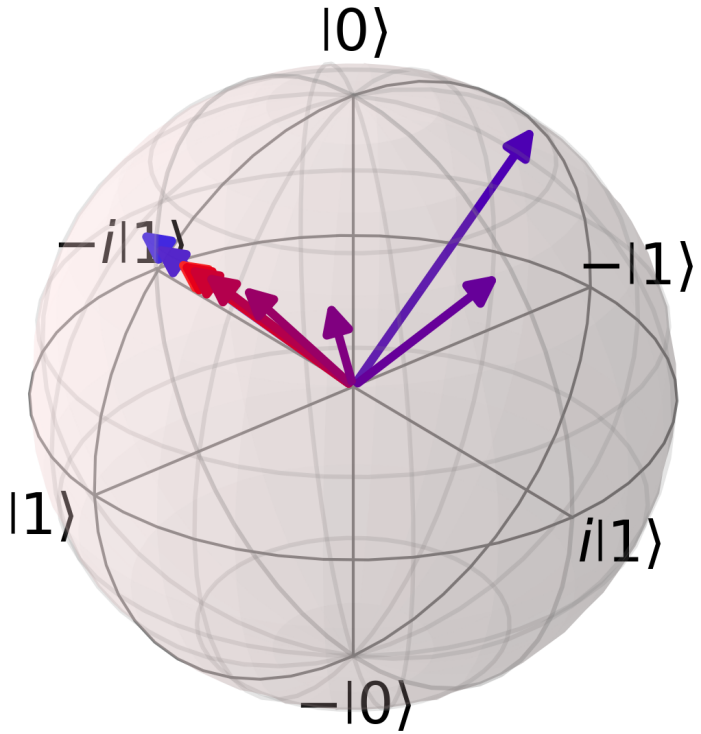

Figure. Quantum Phase Estimation

Overlays

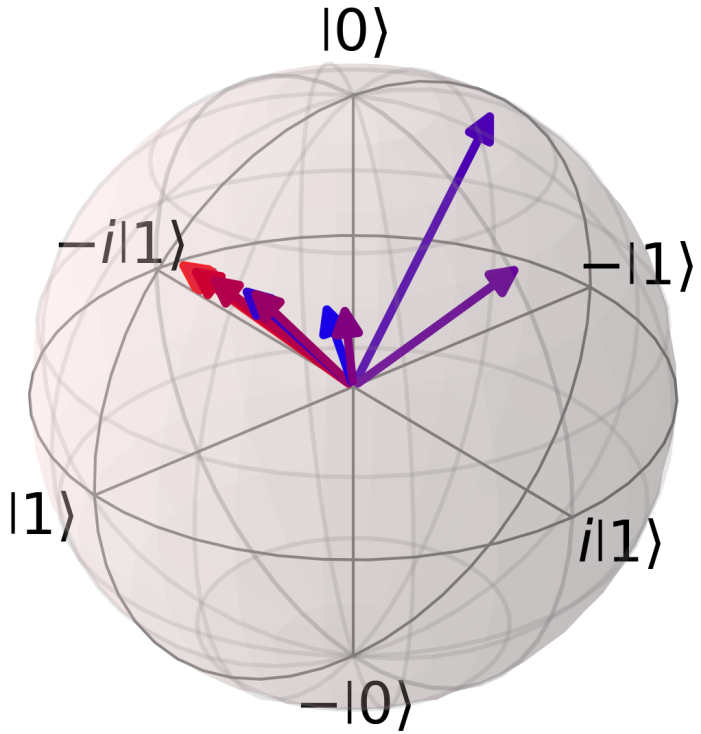

Figure. Quantum Phase Estimation

Overlays

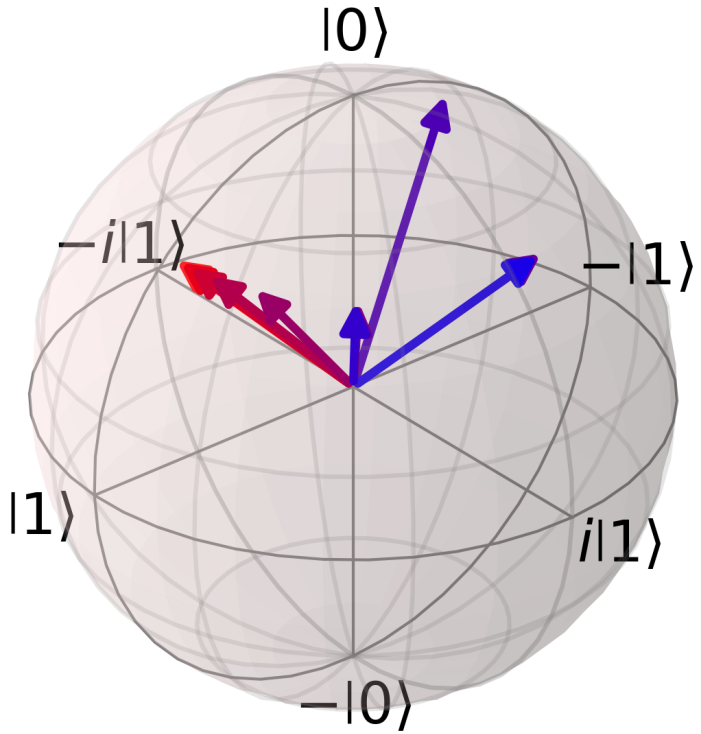

Figure. Quantum Phase Estimation

Overlays

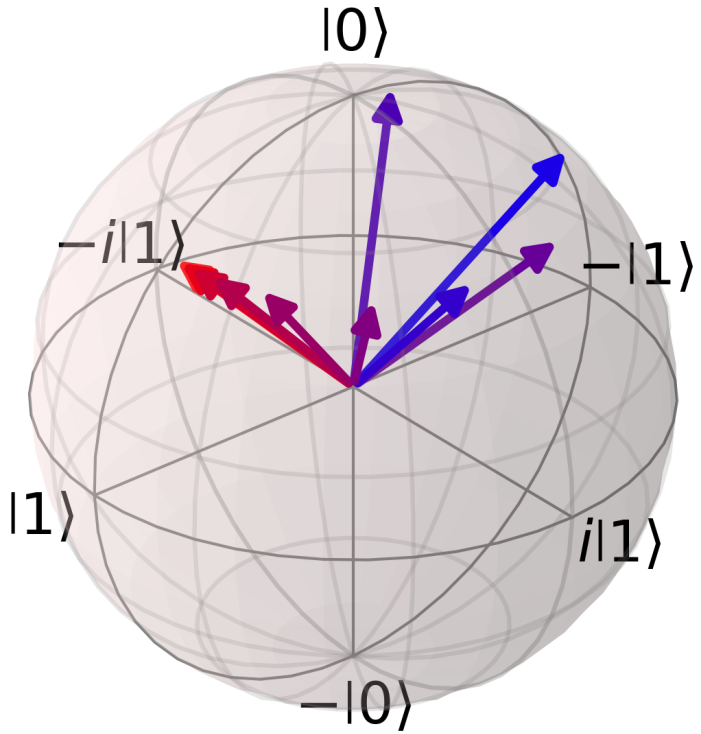

Figure. Quantum Phase Estimation

Overlays

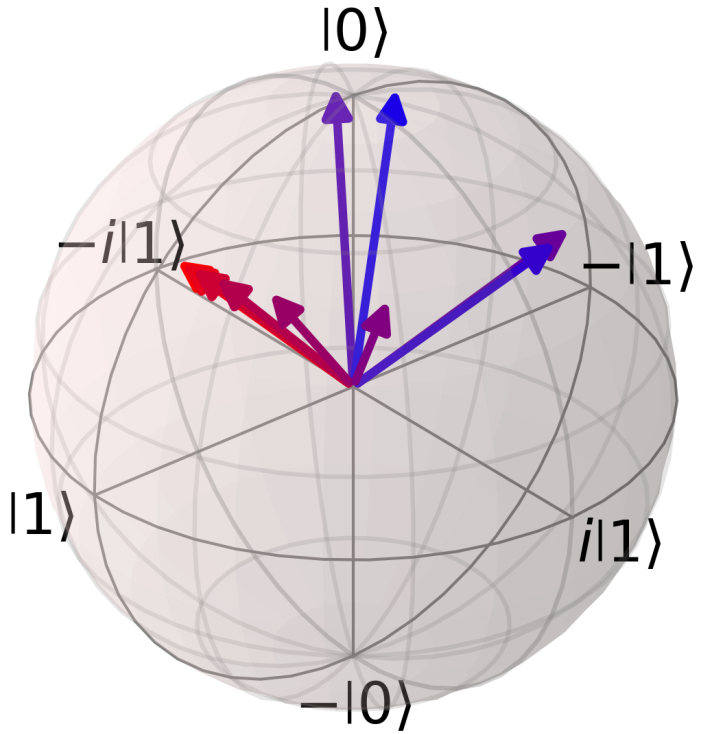

Figure. Quantum Phase Estimation

Overlays

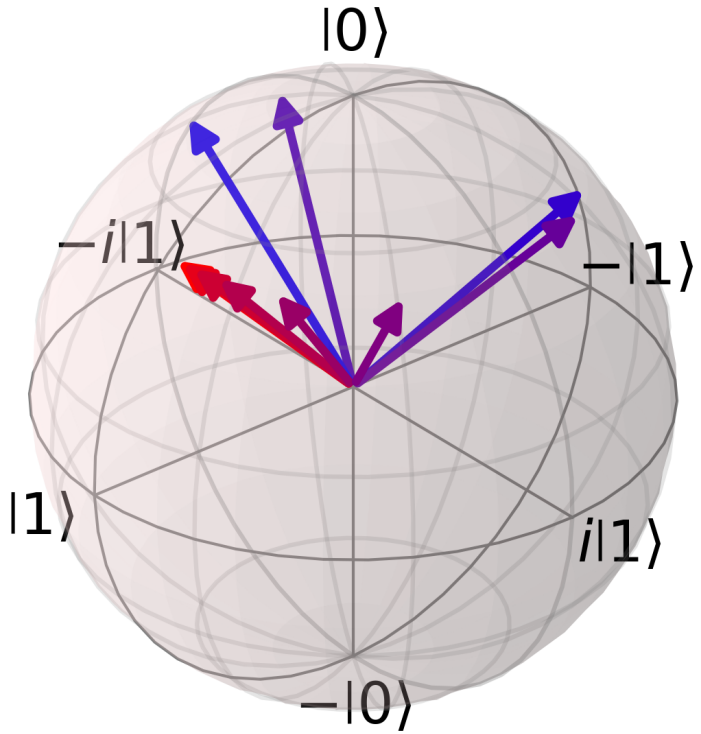

Figure. Quantum Phase Estimation

Overlays

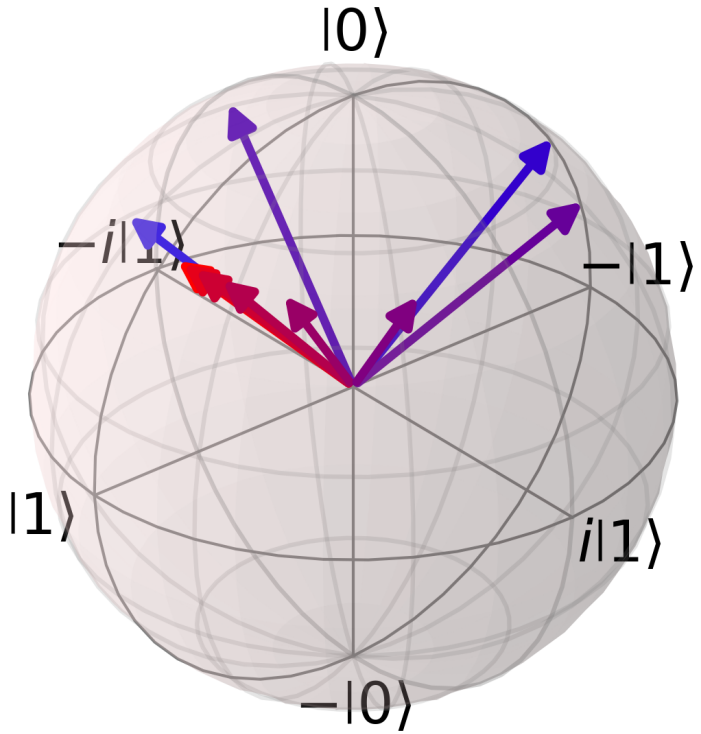

Figure. Quantum Phase Estimation

Overlays

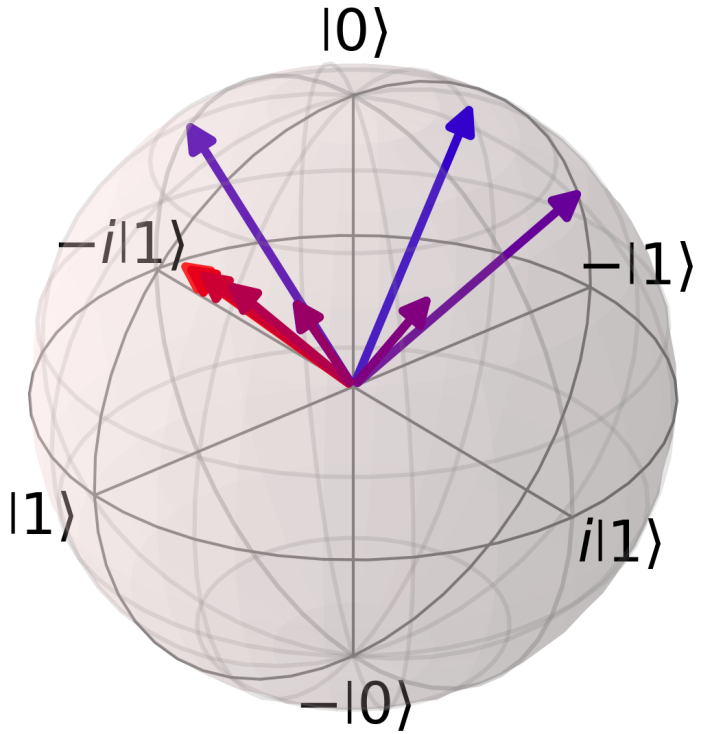

Figure. Quantum Phase Estimation: Final State

Quantum Phase Estimation

After our qubits are in the state above, we proceed with a gate, called the inverse quantum Fourier transform.

For each angle $\phi \in \{, 1 * \frac{2\pi}{2^n}, 2 * \frac{2\pi}{2^n}, 3 * \frac{2\pi}{2^n}, \ldots, (2^n - 1) * \frac{2\pi}{2^n}\}$ the inverse quantum Fourier transform "tests" whether the qubits were rotated by that angle.

Quantum Phase Estimation

If this angle is close enough to the true angle, the probability the inverse quantum Fourier transform returns this angle becomes very large.

Thus, the output of this algorithm will, with high probability, return an angle ϕ close to θ.

What is Shor's Algorithm?

Definition

We define the order of x modulo N as the smallest positive integer a such that $x^a = 1 \pmod{N}$.

Example

The order of 2 modulo 7 is 3 because $2^3 = 8 = 1 \pmod{7}$, and smaller numbers ($a = 1, 2$) do not fit the criteria.

Shor's algorithm is a quantum algorithm invented by Peter Shor in 1994 that finds the order of x modulo N.

This is important because Shor's algorithm can decrypt many commonly used encryption methods such as RSA encryption, elliptic curve cryptography, and Diffie-Hellman key exchange.

Shor's Algorithm

We are now ready to start Shor's algorithm! We wish to find the order of x modulo N, where N is n bits long. Let the order be r.

In Shor's algorithm, we have **two** registers. In our example, let us have the first register contain $2n$ qubits, and the second register contain n qubits.

Shor's Algorithm continued: Modular Multiplication

Let us now define a quantum gate as follows.

$$|i\rangle |j\rangle \xrightarrow{U} |i\rangle |j * x \pmod{N}\rangle$$

for $j < N$.

If $j \geq N$ the quantum gate will act as an identity.

Notice that U^r (applying the quantum gate r times) returns identity.

Thus (we will not prove why), there exists a state of the qubit such that the quantum gate rotates the qubit by an angle $\frac{2\pi j}{r}$ about the z-axis, where j is some integer between 0 and $r - 1$.

Shor's Algorithm continued: Quantum Phase Estimation

We can now use quantum phase estimation to get a number close to $\frac{2\pi j}{r}$.

Getting r still requires some effort, because we do not know j. However, it is possible, using a method called **continued fractions**.

Shor's Algorithm continued: Order retrieved!

Once we use continued fractions to get r, we have found the order!

In conclusion, the algorithm works as follows.

- Use a quantum gate that modular multiplies by x for quantum phase estimation.
- Once we have gotten a phase, use continued fractions to extract r.

Thank You!

This concludes the talk. Any questions?