Bloch Sphere in Quantum Computing

A Bloch Sphere is a unit sphere in 3D. A qubit in any state can be visualized as a point on the surface of this sphere. This blog explains how this is achieved.

We know that any generic qubit is of the following form:

$$|\psi\rangle = (a + ib)|0\rangle + (c + id)|1\rangle$$

Before proceeding further, let's talk about complex numbers a bit. Any complex number $a + ib$ can be written in polar form as follows:

$$re^{i\theta}$$

Where $r\ (magnitude) = \sqrt{a^2 + b^2}$ and $\theta\ (phase) = \tan^{-1}(\frac{b}{a})$

So, we can write the generic qubit in polar form as follows:

$$|\psi\rangle = r_1 e^{i\theta_1}|0\rangle + r_2 e^{i\theta_2}|1\rangle$$

(Eq. 1.0)

Here, the probability equations become:

$$\Pr(|0\rangle) = \left|r_1 e^{i\theta_1}\right|^2 = r1^2$$

$$\Pr(|1\rangle) = \left|r_2 e^{i\theta_2}\right|^2 = r2^2$$

One thing to notice here is that, though the qubit is defined using 4 parameters ($r1, r2,\ \theta1$ and $\theta2$), the actual probabilities are just functions of $r1$ and $r2$ and don't depend on $\theta1$ and $\theta2$. Which means that if we apply any unitary operation or any sequence of unitary operations that just change the phases but no the magnitudes, the probabilities will not change.

Let's consider that following is some unitary matrix:

$$U = \begin{bmatrix} r_a e^{i\theta_a} & r_b e^{i\theta_b} \\ r_c e^{i\theta_c} & r_d e^{i\theta_d} \end{bmatrix}$$

Applying U on $|\psi\rangle$ results in:

$$U(|\psi\rangle) = \begin{bmatrix} r_a e^{i\theta_a} & r_b e^{i\theta_b} \\ r_c e^{i\theta_c} & r_d e^{i\theta_d} \end{bmatrix} * \begin{bmatrix} r_1 e^{i\theta_1} \\ r_2 e^{i\theta_2} \end{bmatrix} = \begin{bmatrix} r_a r_1 e^{i\theta_a + i\theta_1} + r_b r_2 e^{i\theta_b + i\theta_2} \\ r_c r_1 e^{i\theta_c + i\theta_1} + r_d r_2 e^{i\theta_d + i\theta_2} \end{bmatrix}$$

(Eq: 1.1)

Now, let's see how addition works with complex numbers. If we take two complex numbers $x1 + iy1\ and\ x2 + iy2$, the sum is given by $(x1 + x2) + i(y1 + y2)$. But addition rules work a little differently when the complex numbers are in polar form. In polar form, they behave as vectors and we have to use parallelogram law of addition to get the final sum.

Let's say we have two complex numbers $r_x e^{i\theta_x}$ and $r_y e^{i\theta_y}$, we can draw them in the complex plane as follows:
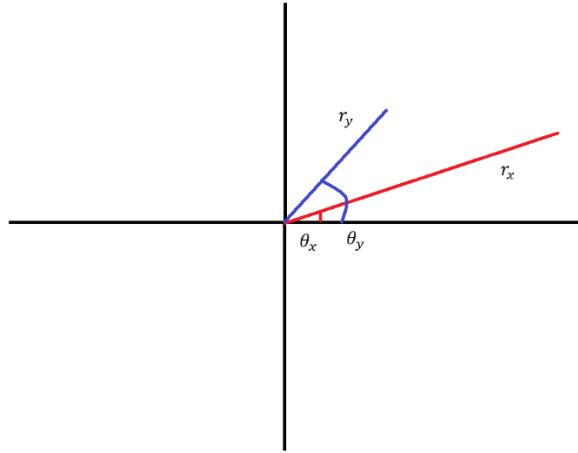

Now, the resultant complex number after adding these two complex numbers is achieved by using the parallelogram law of vectors. We need to attach the tail of the second vector to the head of the first and create a parallelogram. The resultant vector ($r_z e^{i\theta_z}$) will be the diagonal as shown:
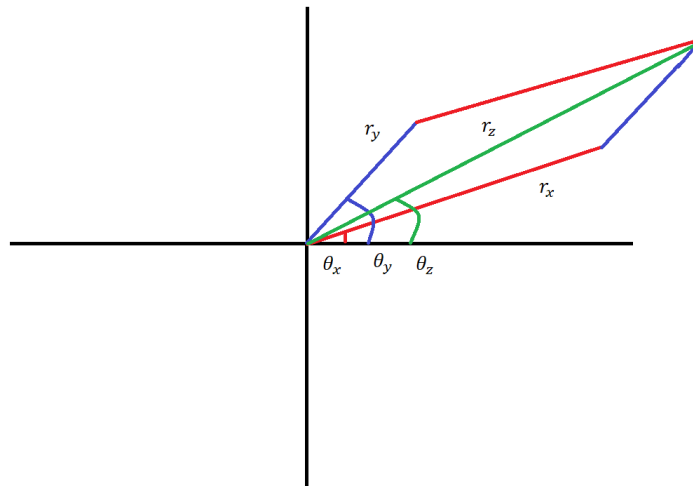
Now, let's say before adding them, we rotate both the complex numbers by some angle θ, then the input complex numbers would look as follows(the magnitudes are kept as same; we added θ to the phases of both the complex numbers) :
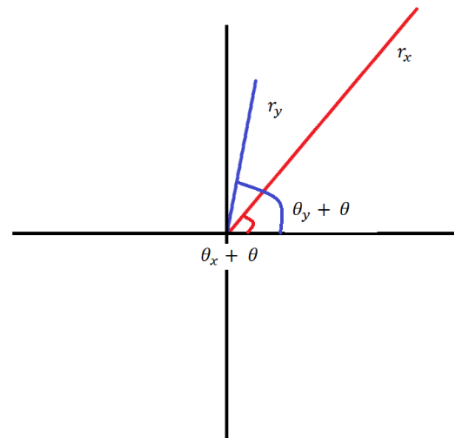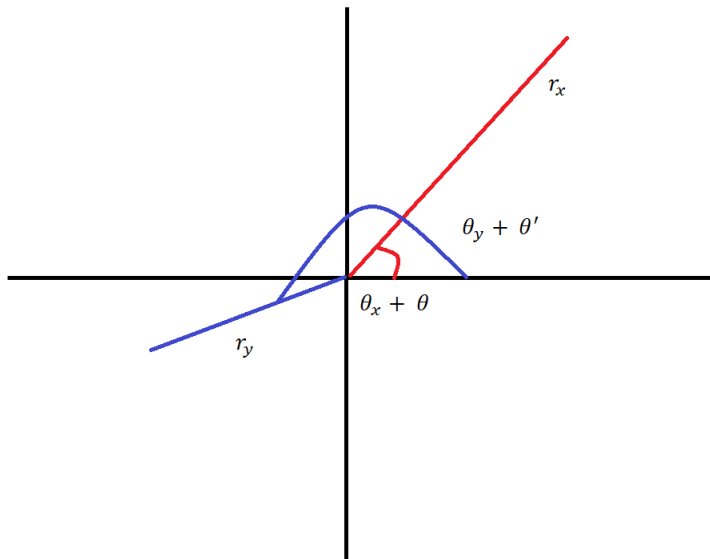

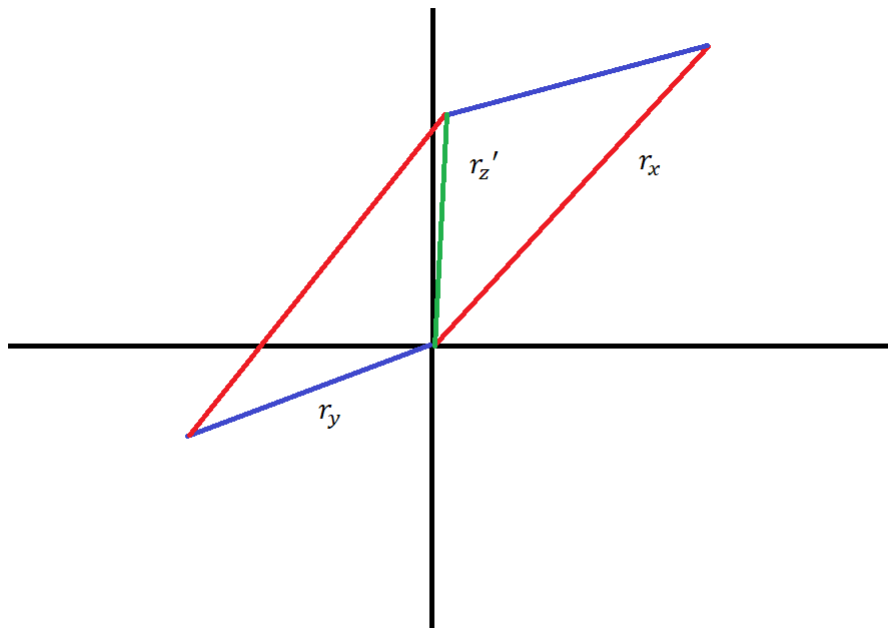Now, if we add them, we get the following result:
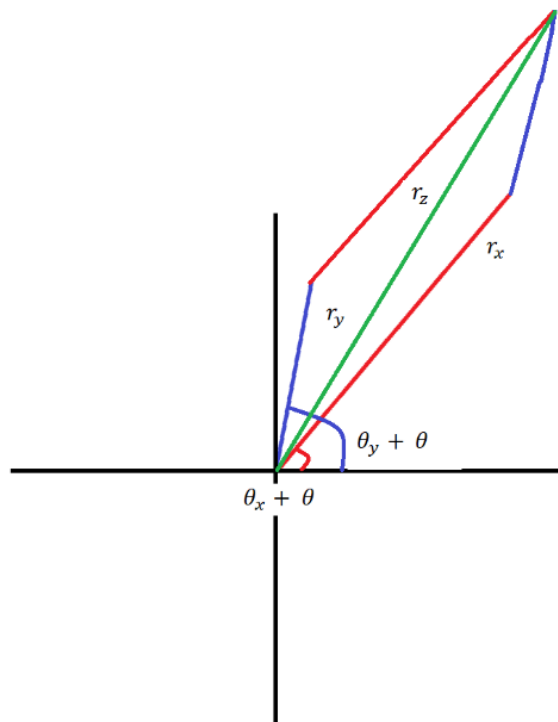
Notice here that even after adding some angle to the input complex numbers, the magnitude of the output complex number didn't change (though the angle changed).

Now, let's see what happens if we add different angles to the input vectors before performing the addition:

After addition:

Here clearly $r_z ! = r_z'$. The moral here is that as long as we add same phase to two complex numbers, the magnitude of the sum of those two complex numbers will not change; however, if we add different phases to the input complex numbers, the magnitude of the sum of them will change.

Now, lets add an angle of θ, to both the states in Eq. 1.0:

$$|\psi'\rangle = r_1 e^{i\theta_1 + i\theta}|0\rangle + r_2 e^{i\theta_2 + i\theta}|1\rangle$$

(Eq. 1.2)

Now, apply U on this state:

$$U(|\psi'\rangle) = \begin{bmatrix} r_a e^{i\theta_a} & r_b e^{i\theta_b} \\ r_c e^{i\theta_c} & r_d e^{i\theta_d} \end{bmatrix} * \begin{bmatrix} r_1 e^{i\theta_1 + i\theta} \\ r_2 e^{i\theta_2 + i\theta} \end{bmatrix} = \begin{bmatrix} r_a r_1 e^{i\theta_a + i\theta_1 + i\theta} + r_b r_2 e^{i\theta_b + i\theta_2 + i\theta} \\ r_c r_1 e^{i\theta_c + i\theta_1 + i\theta} + r_d r_2 e^{i\theta_d + i\theta_2 + i\theta} \end{bmatrix}$$

(Eq. 1.3)

Compare Eq. 1.1 and Eq. 1.3; consider the complex numbers in the first row:

$$r_a r_1 e^{i\theta_a + i\theta_1} + r_b r_2 e^{i\theta_b + i\theta_2}$$

and

$$r_a r_1 e^{i\theta_a + i\theta_1 + i\theta} + r_b r_2 e^{i\theta_b + i\theta_2 + i\theta}$$

The magnitude of the complex number resulting from the first addition will be exactly same as the magnitude of the complex number resulting from the second addition. Same is the case with the complex numbers in the second row. We had proved earlier that the probabilities depend only on the magnitude but not on the phase. So, we can conclude that, as long as we add the same phase to both the probability amplitudes (for $|0\rangle$ and $|1\rangle$), the final probabilities will not change even if we perform any number of unitary operations on this qubit. There difference is not *observable*. So, $|\psi'\rangle$ and $|\psi\rangle$ are effectively the same because the global phase ($e^{i\theta}$) can always be ignored.

Equipped with this information, let's put $\theta = -\theta_1$ in Eq. 1.2

$$|\psi\rangle = r_1 e^{i\theta_1 - i\theta_1}|0\rangle + r_2 e^{i\theta_2 - i\theta_1}|1\rangle$$

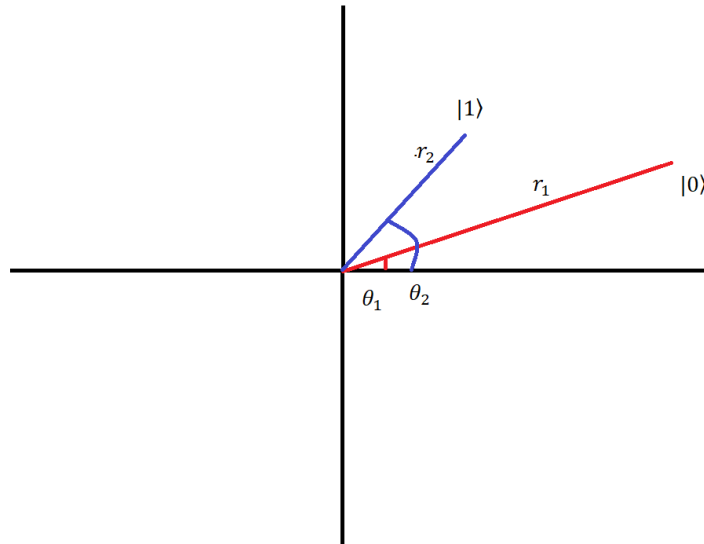$$= r_1|0\rangle + r_2 e^{i\theta_2 - i\theta_1}|1\rangle$$

Let's replace $\theta_2 - \theta_1$ with φ:

$$|\psi\rangle = r_1|0\rangle + r_2 e^{i\varphi}|1\rangle$$

Notice here that we are effectively making the probability amplitude of $|0\rangle$ to always a real number. So, the actual phases of $|0\rangle$ and $|1\rangle$ don't really matter; only thing that matters is the phase difference $\theta_2 - \theta_1 (i.e\ \varphi)$
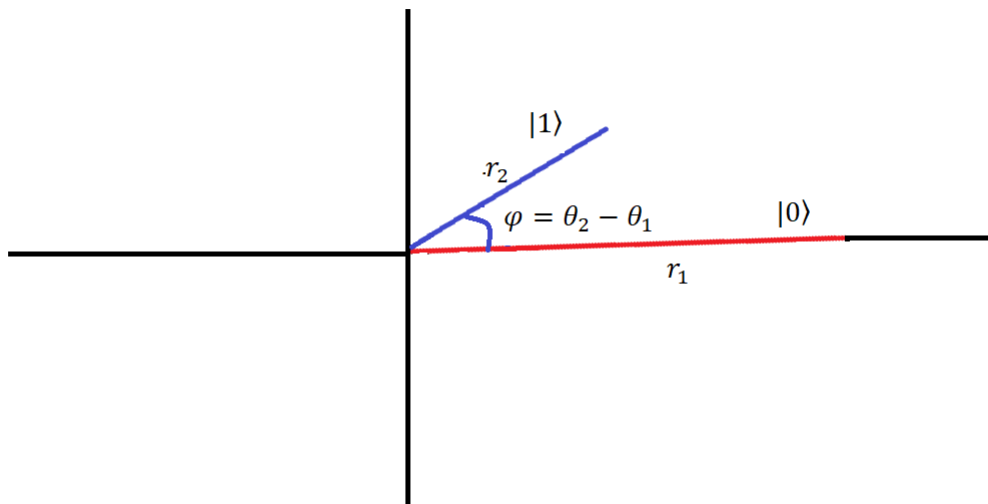
So, given any qubit in the following state:

$$|\psi\rangle = r_1 e^{i\theta_1}|0\rangle + r_2 e^{i\theta_2}|1\rangle$$

We can write it as:

$$|\psi\rangle = r_1|0\rangle + r_2 e^{i\varphi}|1\rangle$$

(Eq. 1.4)

Here, since φ is the phase difference between the two states, the range of it will be 0 - 360.

We also know that $\Pr(|0\rangle) + \Pr(|1\rangle) = 1$

⇨ $r_1{}^2 + r_2{}^2 = 1$ (also both r_1 and r_1 will be non-negative as they are magnitudes of complex numbers)

So, we can say that for every combination of r_1 and r_2 as constrained above, there will be an angle x such that they form the sides of a right angled triangle with hypotenuse of length 1:
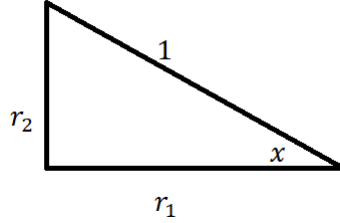
$$r_1 = 1 * \cos x$$

$$r_2 = 1 * \sin x$$

So, Eq. 1.4 can be written as:

$$|\psi\rangle = \cos x |0\rangle + \sin x * e^{i\varphi} |1\rangle$$

Here, the thing to be noticed is that the range of x will be $0 - 90$; because the smallest value will be when $r_2 = 0$ and $r_1 = 1$; and highest will be when $r_2 = 1$ and $r_1 = 0$. For every such x, there will be an angle θ (with the range 0 - 180) such that $\theta = 2x$. So, the above equation becomes:

$$|\psi\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2) * e^{i\varphi} |1\rangle$$

(Eq. 1.5)

Why we had to introduce θ with double the value of x will be clear later, but it is for sure that this is not mathematically incorrect.

What we just did is that given any generic qubit ($r_1 e^{i\theta_1}|0\rangle + r_2 e^{i\theta_2}|1\rangle$), we can write it in the form $|\psi\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2) * e^{i\varphi}|1\rangle$ parameterized by θ (with range $0 - 180$) and φ (with range $0 - 360$) without losing any useful information.

Now, let's proceed with the second half of the problem. Given any unit sphere, what will be the polar co-ordinates of any point on the surface of it?
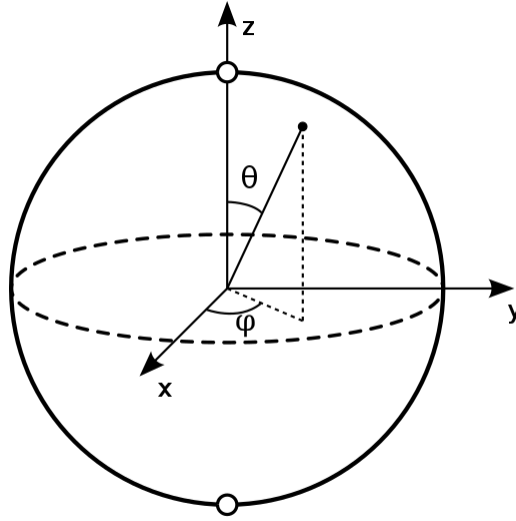
For any given point on the surface of this unit sphere, we can define two angles θ and φ. θ is the angle between the z axis and the line connecting the origin with the point with which we are concerned. Now, drop a perpendicular from this point onto the xy plane and connect that point on xy plane to the origin. The angle made by this line with the x axis is defined as φ. Now, we can check that, if we start from the north pole (where $\theta = 0$), for every miniscule increment of θ, we can sweep an entire layer of this sphere by changing φ from 0 to 360. By the time we reach the south pole (where $\theta = 180$) like this, we would have covered the entire sphere. So, as defined above, any point on the surface of this unit sphere can be parameterized by two angles θ (with range 0 to 180) and φ (with range 0 to 360).

Now, compare this conclusion with that of Eq. 1.5. Thus, we have proved that given any generic qubit state, we can visualize it as a point on a unit sphere.

Let's try to fix few important qubit states on this sphere.

For $|\psi\rangle = |0\rangle$:

To calculate θ and φ for this state, we need to equate this to the Eq. 1.5:

$$|0\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2) * e^{i\varphi}|1\rangle$$

Which means $\cos(\theta/2) = 1$ and $\sin(\theta/2) * e^{i\varphi} = 0$.

If $\cos\left(\frac{\theta}{2}\right) = 1$

$\Rightarrow \left(\frac{\theta}{2}\right) = 0$

$\Rightarrow \theta = 0$

If $\theta = 0$, then $\sin(\theta/2) * e^{i\varphi}$ becomes 0 irrespective of the value of φ.

This means that $|1\rangle$ is represented as the north pole of the Bloch Sphere.

For $|\psi\rangle = |1\rangle$:

$$|1\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2) * e^{i\varphi}|1\rangle$$

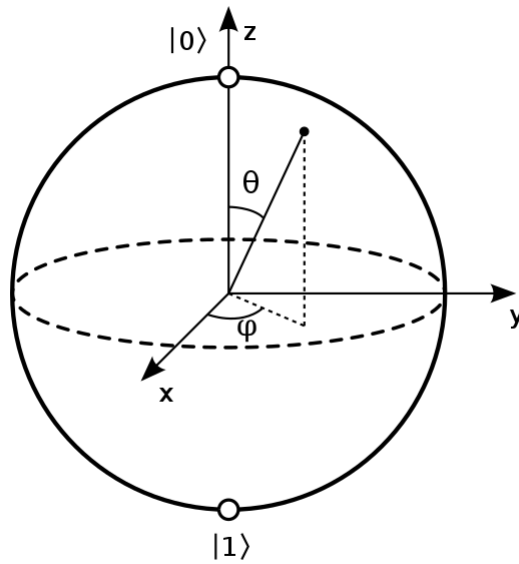Which means $\cos(\theta/2) = 0$ and $\sin(\theta/2) * e^{i\varphi} = 1$.

If $\cos\left(\frac{\theta}{2}\right) = 0$

$\Rightarrow \left(\frac{\theta}{2}\right) = 90°$

$\Rightarrow \theta = 180°$

If $\theta = 180°$, $\sin(180/2) * e^{i\varphi} = 1$ becomes $e^{i\varphi} = 1$, which means $\varphi = 0$.

This represents the point in the south pole of the Bloch Sphere.

Now, it must be clear to the reader why we used $\theta = 2x$ in the earlier derivation; this is to ensure that we get $|0\rangle$ and $|1\rangle$ in the north-pole and south-pole respectively.

For $|\psi\rangle = H(|0\rangle)$:

$$H(|0\rangle) = \left(\frac{|0\rangle}{\sqrt{2}} + \frac{|1\rangle}{\sqrt{2}}\right) = \cos(\theta/2)|0\rangle + \sin(\theta/2) * e^{i\varphi}|1\rangle$$

Which means $\cos(\theta/2) = \frac{1}{\sqrt{2}}$ and $\sin(\theta/2) * e^{i\varphi} = \frac{1}{\sqrt{2}}$

If $\cos\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{2}}$

$\Rightarrow \left(\frac{\theta}{2}\right) = 45°$

$\Rightarrow \theta = 90°$

Now, $\sin(90/2) * e^{i\varphi} = \frac{1}{\sqrt{2}}$

$$\Rightarrow \quad \frac{1}{\sqrt{2}} * e^{i\varphi} = \frac{1}{\sqrt{2}}$$

$$\Rightarrow \quad \varphi = 0$$

This point lies on the x-axis towards the paper.

For $|\psi\rangle = H(|1\rangle)$:

$$H(|1\rangle) = \left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right) = \cos(\theta/2)\,|0\rangle + \sin(\theta/2) * e^{i\varphi}|1\rangle$$

Which means $\cos(\theta/2) = \frac{1}{\sqrt{2}}$ and $\sin(\theta/2) * e^{i\varphi} = \frac{-1}{\sqrt{2}}$

If $\cos\left(\frac{\theta}{2}\right) = \frac{1}{\sqrt{2}}$

$$\Rightarrow \quad \left(\frac{\theta}{2}\right) = 45°$$

$$\Rightarrow \quad \theta = 90°$$

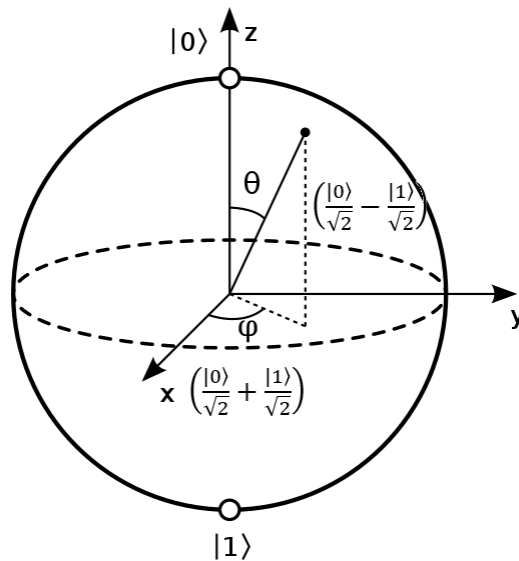Now, $\sin(90/2) * e^{i\varphi} = \frac{-1}{\sqrt{2}}$

$$\Rightarrow \quad \frac{1}{\sqrt{2}} * e^{i\varphi} = \frac{-1}{\sqrt{2}}$$

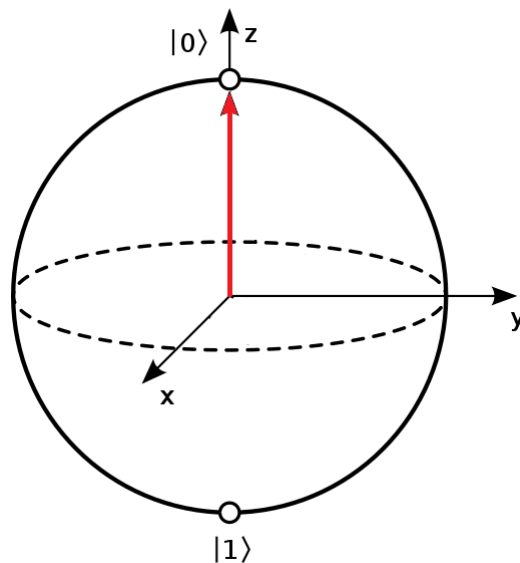$$\Rightarrow \quad e^{i\varphi} = -1$$

$$\Rightarrow \quad \varphi = 180°$$

(because from Euler's formula: $e^{i\varphi} = \cos\varphi + i\sin\varphi$; putting $\varphi = 180$ this becomes $\cos 180 + i\sin 180 = -1 + i0 = -1$)

This point lies on the x-axis away the paper.

The gates, X, Y and Z can be visualized as rotating the Bloch sphere around the corresponding axis by 180 degrees.

Let's say our qubit is in state $|0\rangle$, i.e. it is pointing to the north-pole. Following will be the Bloch Sphere representation of it:

(Fig. 1.0)

Now, if we rotate the Bloch Sphere around the x-axis by 180°, keeping the axes static, the state looks like this:
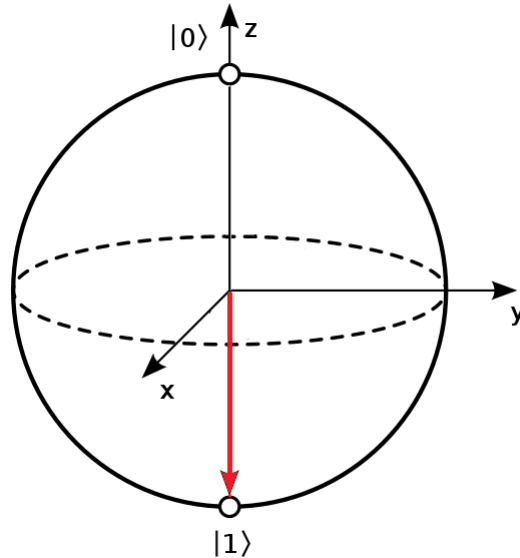
(Fig 1.1)

Which is $|1\rangle$, and is correct because $X(|0\rangle) = |1\rangle$. Now, if we rotate the Bloch Sphere around the x-axis by 180° again, by keeping the axes static, we will get the state in Fig 1.0 again; this is also correct because applying x gate twice is same as applying I.

What happens if we rotate the Bloch Sphere in Fig 1.0 around the y-axis by 180°? It will also result in the state shown in Fig 1.1. But what happens to $|0\rangle$, when we actually multiply with the Y gate. Let's check it:

$$Y(|0\rangle) = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ i \end{bmatrix}$$

We got $\begin{bmatrix} 0 \\ i \end{bmatrix}$, but 180° rotation around Y-axis gave us $|1\rangle$ $i.e$ $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. This apparent discrepancy is because of they way we use Bloch Sphere to represent the qubit. We proved earlier that, all the qubits can be considered to be in the same state as long as they differ only by a global phase. We can write $\begin{bmatrix} 0 \\ i \end{bmatrix}$ as $\begin{bmatrix} 0 \\ e^{i\pi/2} \end{bmatrix}$ (by using the Euler's Formula stated earlier). This can be written as $\begin{bmatrix} 0 * e^{i\pi/2} \\ 1 * e^{i\pi/2} \end{bmatrix}$. Now, we can remove the global phase $e^{i\pi/2}$ which will make the state $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ which represents the south pole.

Now, let's take the state in Fig 1.0 and rotate around the Z-axis by 180°. It must be obvious that, since the state is aligned with the Z-axis, this rotation has no effect. Let's apply the Z-gate on $|0\rangle$:

$$Z(|0\rangle) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We see that the values match.

Now, let's take the state in Fig 1.1 (i.e. $|1\rangle$), and rotate around the Z-axis by 180°. As in the previous case, the state will remain unchanged because it is already aligned with the Z-axis. But, what happens if we apply the Z-gate on $|1\rangle$?

$$Z(|1\rangle) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} * \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

Here we got $\begin{bmatrix} 0 \\ -1 \end{bmatrix}$ instead of $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$. We can prove that in the Bloch Sphere, both these states represent the same point (the south pole) as follows:

$\begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 * e^{i\pi} \\ 1 * e^{i\pi} \end{bmatrix}$ (using the Euler's formula)

$= \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ (removing the global phase $e^{i\pi}$)

Similarly, while representing in a Bloch Sphere, all the following states are represented by the south-pole:

$|1\rangle, -|1\rangle, i|1\rangle, -i|1\rangle, (\frac{\sqrt{3}}{\sqrt{4}} + \frac{\sqrt{1}}{\sqrt{4}})|1\rangle$ etc.

Similarly, infinitely many states can be represented as the north-pole as follows:

$|0\rangle, -|0\rangle, i|0\rangle, -i|0\rangle, (\frac{\sqrt{3}}{\sqrt{4}} + \frac{\sqrt{1}}{\sqrt{4}})|0\rangle$ etc.

Apart from 180° rotation around any axis, we can rotate the Bloch Sphere around these axes by any random angle.

Q# defines the following rotations:

$R_x(\theta) = e^{-i\theta/2\sigma_x}$, where $\sigma_x = Pauli\ X\ matrix$

$R_y(\theta) = e^{-i\theta/2\sigma_y}$, where $\sigma_y = Pauli\ Y\ matrix$

$R_z(\theta) = e^{-i\theta/2\sigma_z}$, where $\sigma_z = Pauli\ Z\ matrix$

Each rotation means, rotating the Bloch Sphere by the specified angle around the specified axis in anti-clock wise direction.

Let's try to find the matrices for these rotations.

We know that:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} \dots$$

Similarly, we know that:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$$

And:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

(For the proof of the above expansions, Bing for Maclaurin expansion for sin and cos.)

Now:

$$e^{-i\theta/2\sigma_x} = 1 + -i\theta/2\sigma_x + \frac{(-i\theta/2\sigma_x)^2}{2!} + \frac{(-i\theta/2\sigma_x)^3}{3!} + \frac{(-i\theta/2\sigma_x)^4}{4!} \dots$$

$$= \cos\left(\frac{\theta}{2}\right) * I - i * \sin\left(\frac{\theta}{2}\right) * \sigma_x$$

(because $\sigma_x * \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$)

Same expansion can be given for the remaining rotations also.

Let's find the matrices for these rotations:

$$R_x(\theta) = e^{-i\theta/2\sigma_x} = \cos\left(\frac{\theta}{2}\right) * I - i * \sin\left(\frac{\theta}{2}\right) * \sigma_x$$

$$= \cos\left(\frac{\theta}{2}\right) * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - i * \sin\left(\frac{\theta}{2}\right) * \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & 0 \\ 0 & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} - \begin{bmatrix} 0 & i * \sin\left(\frac{\theta}{2}\right) \\ i * \sin\left(\frac{\theta}{2}\right) & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i * \sin\left(\frac{\theta}{2}\right) \\ -i * \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

$$R_y(\theta) = e^{-i\theta/2\sigma_y} = \cos\left(\frac{\theta}{2}\right) * I - i * \sin\left(\frac{\theta}{2}\right) * \sigma_y$$

$$= \cos\left(\frac{\theta}{2}\right) * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - i * \sin\left(\frac{\theta}{2}\right) * \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & 0 \\ 0 & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} - \begin{bmatrix} 0 & \sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

$$R_z(\theta) = e^{-i\theta/2\sigma_z} = \cos\left(\frac{\theta}{2}\right) * I - i * \sin\left(\frac{\theta}{2}\right) * \sigma_z$$

$$= \cos\left(\frac{\theta}{2}\right) * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - i * \sin\left(\frac{\theta}{2}\right) * \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & 0 \\ 0 & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} - \begin{bmatrix} i * \sin\left(\frac{\theta}{2}\right) & 0 \\ 0 & -i * \sin\left(\frac{\theta}{2}\right) \end{bmatrix}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) - i * \sin\left(\frac{\theta}{2}\right) & 0 \\ 0 & \cos + i * \sin\left(\frac{\theta}{2}\right) \end{bmatrix}$$

$$= \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \text{ (from Euler's formula)}$$

As you can see, unlike R_x and R_y, R_z doesn't change any probabilities; it changes only the phase difference between the two states by θ; it decreases the phase of $|0\rangle$ by $\frac{\theta}{2}$ and increases the phase of $|1\rangle$ by $\frac{\theta}{2}$. This is not same as adding a global phase, because it is adding different phases to each state.

Let's see how $R_x(\pi)$ (rotation around x-axis by 180° (or $\pi \ radians$)) works:

$$R_x(\pi) = \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & -i*\sin\left(\frac{\pi}{2}\right) \\ -i*\sin\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -i \\ -i & 0 \end{bmatrix}$$

$$= -i* \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
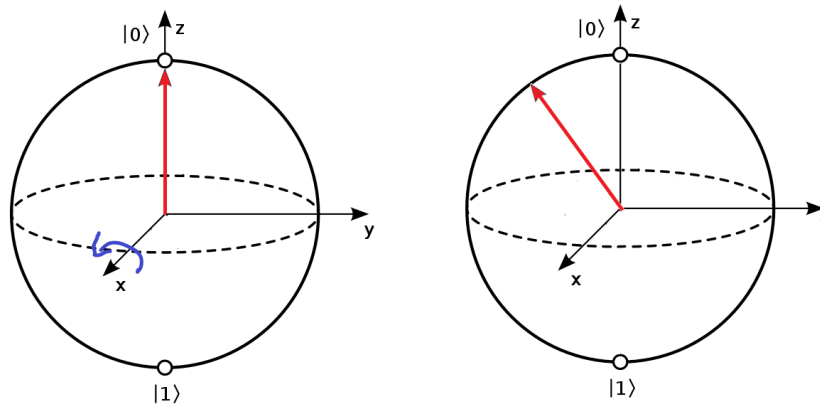
$= -i * X$ (this is similar to applying the X gate, because applying the global phase of $-i$ can be ignored as proved earlier)

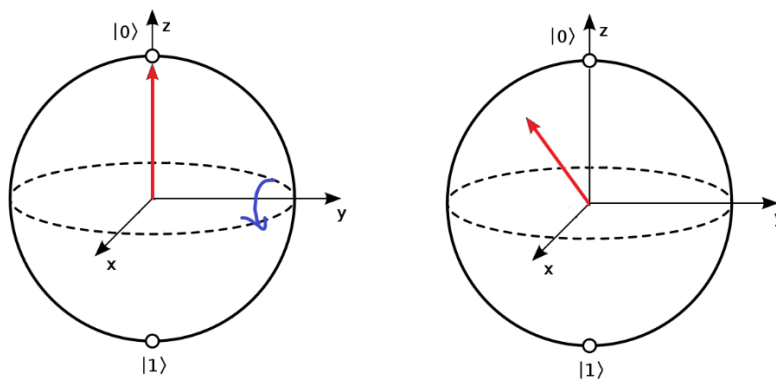Same goes for the other rotations as follows:

$$R_y(\pi) = \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & -\sin\left(\frac{\pi}{2}\right) \\ \sin\left(\frac{\pi}{2}\right) & \cos\left(\frac{\pi}{2}\right) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = -i* \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = -i * Y$$

$$R_z(\pi) = \begin{bmatrix} e^{-i\frac{\pi}{2}} & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix} = \begin{bmatrix} -i & 0 \\ 0 & i \end{bmatrix} = -i* \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = -i * Z$$
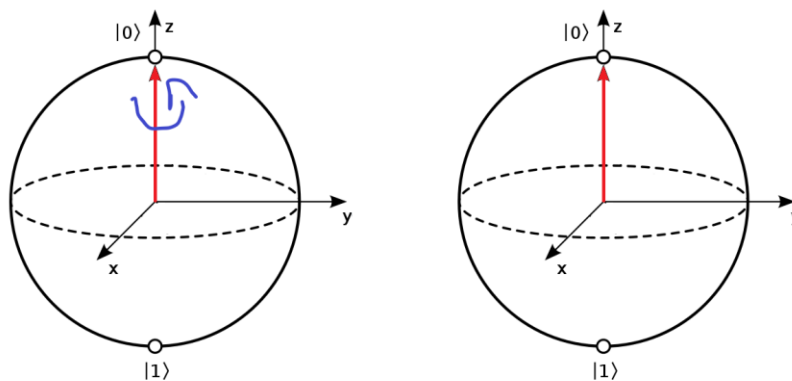
R_x:

R_y:

R_z:

Q# also provides a generic rotation operation R to which you can pass any Pauli axis:

E.g.: R(PauliX, PI(), qubit). This is same as Rx(PI(), qubit).

To this R operation, we can pass PauliI to give a global phase as follows:

$$e^{-i\theta/2\sigma_I} = \cos\left(\frac{\theta}{2}\right) * I - i * \sin\left(\frac{\theta}{2}\right) * \sigma_I$$

$$= \cos\left(\frac{\theta}{2}\right) * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - i * \sin\left(\frac{\theta}{2}\right) * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) - i\sin\left(\frac{\theta}{2}\right) & 0 \\ 0 & \cos\left(\frac{\theta}{2}\right) - i\sin\left(\frac{\theta}{2}\right) \end{bmatrix}$$

$$= \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{-i\frac{\theta}{2}} \end{bmatrix} \text{ (from Euler's formula)}$$

This operation adds a common phase to both the states; so, though technically the qubit state is being changed, the point will not move on the Bloch Sphere as there is no change in the phase difference.

Let's say we start with the qubit at $|0\rangle$ and want to give it some probability like (2/10) for $|0\rangle$ and (8/10) for $|1\rangle$, then we can use $R_y(\theta)$ by passing appropriate value for θ. Here the advantage is that we can do it by keeping the phase as 0. We can use the following process to find θ for these probabilities:

$$\Pr(|0\rangle) = (2/10)$$

$$\Pr(|1\rangle) = (8/10)$$

We can come up with infinite number of probability amplitudes to get these probabilities. But, let's choose a qubit as follows:

$$\frac{\sqrt{2}}{\sqrt{10}}|0\rangle + \frac{\sqrt{8}}{\sqrt{10}}|1\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2) * e^{i\varphi}|1\rangle$$

$$\Rightarrow \cos(\theta/2) = \frac{\sqrt{2}}{\sqrt{10}}$$

$$\Rightarrow \frac{\theta}{2} = \cos^{-1}\left(\frac{\sqrt{2}}{\sqrt{10}}\right)$$

$$\Rightarrow \theta = 2 * \cos^{-1}\left(\frac{\sqrt{2}}{\sqrt{10}}\right)$$

We can pass the θ obtained by the above calculation to the R_y operation to get the desired super position. To get the desired phase you can use the R_z operation.

As you can observe, the R_x and R_y operations change the probabilities, where as R_z changes only the phase of the qubit.

In this blog we are covering qubits only in pure states; a qubit in a mixed state will be represented by a point inside the Bloch Sphere and will be covered in a later blog.

Sample Q# code:

DumpMachine("") will display the state of the current system. For a single qubit system $((a + ib)|0\rangle + (c + id)|1\rangle)$, the output will be as follows:

Ids: [0;]

Wavefunction:

0:   a   b

1:   c   d

Driver.cs:

```csharp
using Microsoft.Quantum.Simulation.Core;
using Microsoft.Quantum.Simulation.Simulators;
using System;

namespace Quantum.DemoRotations2
{
    class Driver
    {
        static void Main(string[] args)
        {
            using (var quantumSimulator = new QuantumSimulator())
            {
                var result = DemoRotations.Run(quantumSimulator).Result;
            }
            Console.WriteLine("Done");
            Console.ReadLine();
        }
    }
}
```

Operation.qs:

```
namespace Quantum.DemoRotations2
{
    open Microsoft.Quantum.Primitive;
    open Microsoft.Quantum.Canon;
    open Microsoft.Quantum.Extensions.Diagnostics;
    open Microsoft.Quantum.Extensions.Math;

    operation DemoRotations () : ()
    {
        body
        {
            using(qubits = Qubit[1])
            {
                DumpMachine("");
            }
        }
    }
}
```

In this code, since we didn't perform any operation, the qubit will be in the default |0⟩ state:

Ids: [0;]

Wavefunction:

0:   1   0

1:   0   0

Further examples:

Code:

```
X(qubits[0]);
DumpMachine("");
```

Output:

Ids: [0;]

Wavefunction:

0:   0   0

1:   1   0

Code:

```
Rx(PI(), qubits[0]);
DumpMachine("");
```

Output:

Ids: [0;]

Wavefunction:

0:   6.12323e-17(almost 0)   0

1:   0   -1

Code:

```
R(PauliX,PI(), qubits[0]);
DumpMachine("");
```

Output (same as above):

Ids: [0;]

Wavefunction:

0:    6.12323e-17   0

1:    0   -1

Code:

```
H(qubits[0]);
DumpMachine("");
```

Output:

Ids: [0;]

Wavefunction:

0:    0.707107   0

1:    0.707107   0

Code:

```
H(qubits[0]);
R(PauliI, PI(), qubits[0]);
```

Output (similar to above, but phase of i added to both the states):

Ids: [0;]

Wavefunction:

0:    4.32978e-17 (almost 0)   -0.707107

1:    4.32978e-17 (almost 0)   -0.707107

Code:

This program creates a qubit with the following probabilities:

$$Pr(|0\rangle) = (2/10)$$

$$Pr(|1\rangle) = (8/10)$$

```
body
{
    mutable countZero = 0;
    mutable countOne = 0;
    mutable iterations = 1000;

    using(qubits = Qubit[1])
    {
        for(i in 1..iterations)
        {
            let probabilityOfZero = 2.0/10.0;
            let amplitudeOfZero = Sqrt(probabilityOfZero);
            let angle = 2.0 * ArcCos(amplitudeOfZero);
            Ry(angle, qubits[0]);

            let result = M(qubits[0]);
            if(result == Zero)
            {
                set countZero = countZero + 1;
            }
            else
            {
                set countOne = countOne + 1;
            }
            ResetAll(qubits);
        }
        Message($"countZero: {countZero}");
        Message($"countOne: {countOne}");

    }
}
```

Output:

countZero: 205

countOne: 795