Midterm Report:
Quantum Computing, Information and Quantum Technologies

SoS '24

Pranav Malpani
23B1279
Mentor: Anand Narasimhan

June 2024

Abstract

This is a mid-term report for **P02 - Quantum Computing, Information and Quantum Technologies**. It aims to highlight my progress in understanding this topic during the span of the past 4 weeks. My prior introduction to this whole topic was limited to a linear algebra and a quantum mechanics course during my second semester.

The material used to study this topic were mainly *Quantum Computation and Quantum Information* by M. Nielsen and I. Chuang (referred to as *the book* in this report), and *The Qiskit Textbook*, maintained by IBM. The Qiskit textbook in its present form is a github repository that contains all the chapters in a `.ipynb` file format.*Consistent Quantum Theory* by Robert J. Griffiths, which has a great chapter for linear algebra in Dirac notation, was also referred to for the initial portion.

Contents

Chapter 1

Linear Algebra

1.1 Introduction and Recap

Linear algebra is the study of vector spaces and of linear operations on those vector spaces. I had already had some experience with it due to having MA110 in my second semester, but the new *Dirac* notation and terminology needed some getting used to.

Notation	Description
z^*	Complex conjugate of the complex number z.
	$(1+i)^* = 1 - i$
$\|\psi\rangle$	Vector. Also known as a *ket*.
$\langle\psi\|$	Vector dual to $\|\psi\rangle$. Also known as a *bra*.
$\langle\varphi\|\psi\rangle$	Inner product between the vectors $\|\varphi\rangle$ and $\|\psi\rangle$.
$\|\varphi\rangle \otimes \|\psi\rangle$	Tensor product of $\|\varphi\rangle$ and $\|\psi\rangle$.
$\|\varphi\rangle\|\psi\rangle$	Abbreviated notation for tensor product of $\|\varphi\rangle$ and $\|\psi\rangle$.
A^*	Complex conjugate of the A matrix.
A^T	Transpose of the A matrix.
$A^\dagger$	Hermitian conjugate or adjoint of the A matrix, $A^\dagger = (A^T)^*$.
	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^\dagger = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}$.
$\langle\varphi\|A\|\psi\rangle$	Inner product between $\|\varphi\rangle$ and $A\|\psi\rangle$.
	Equivalently, inner product between $A^\dagger\|\varphi\rangle$ and $\|\psi\rangle$.

Figure 1.1: Table of the new notation

The basics all remain conceptually the same, although some additions include the definition of ***Hilbert spaces***, ***inner, outer and tensor products***, ***Normal, Hermitian and Unitary operators*** and ***a new definition of diagonalisation***.

1.2 Dual of a vector

The Dual of a vector written in *ket*-form is a *bra*-vector. Essentially, it is equivalent to a the hermitian conjugate (or simply conjugate) of a vector, that is, for a vector $|v\rangle$, its dual

$$\langle v| = \left(|v\rangle^*\right)^T, or\, |v\rangle^\dagger$$

1.3 Inner Product

The Inner product of two vectors is a function that takes two vectors from a vector space V and outputs a complex number. More specifically,
A function from $V \times V$ to C is an inner product if it satisfies the requirements:

1. It is linear in the second argument,

$$\left(|v\rangle, \sum_i \lambda_i |w_i\rangle\right) = \sum_i \lambda_i \left(|v\rangle, |w_i\rangle\right)$$

2. $(|v\rangle, |w\rangle) = (|w\rangle, |v\rangle)^*$
3. $(|v\rangle, |v\rangle) \geq 0$ with equality iff $|v\rangle = 0$

For computing the inner product $(|v\rangle, |w\rangle)$, we simply multiply the dual of $|v\rangle$ with $|w\rangle$.
We also define the norm of a vector $|v\rangle$ as

$$\|\, |v\rangle \,\| = \sqrt{\langle v|v\rangle},$$

much like the regular definition.

1.4 Hilbert Spaces

All inner product spaces are Hilbert spaces, atleast when they have finite dimensions. An inner product space is any vector space for which an inner product can be defined.

1.5 Outer Product

In the matrix representations, the outer product is a column matrix (ket-vector) multipled by a row matrix (bra-vector), giving a rectangular matrix, or an *operator*. For *eg*, the outer product of $|a\rangle$ and $\langle b|$ is just $|a\rangle \langle b|$.

$$|a\rangle\langle b| \;=\; \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \begin{pmatrix} b_1^* & b_2^* & \cdots & b_n^* \end{pmatrix} \;=\; \begin{pmatrix} a_1 b_1^* & a_1 b_2^* & \cdots & a_1 b_n^* \\ a_2 b_1^* & a_2 b_2^* & \cdots & a_2 b_n^* \\ \vdots & \vdots & \ddots & \vdots \\ a_n b_1^* & a_n b_2^* & \cdots & a_n b_n^* \end{pmatrix}$$

This gives a useful relation known as *the completeness relation*,

$$\left(\sum_i |i\rangle \langle i|\right) |v\rangle = \sum_i |i\rangle \langle i|v\rangle = \sum_i v_i \, |i\rangle = |v\rangle \,,$$

where $\{|i\rangle\}$ is an orthonormal basis for the vector space.

1.6 Normal, Hermitian and Unitary Operators

Normal operators are operators that commute with their conjugates, i.e.

$$A^\dagger A = A A^\dagger$$

Hermitian operators are operators whose conjugate is equal to themselves, i.e.

$$H^\dagger = H.$$

Hermitian operators turn out to be quite important in quantum mechanics, due to the fact that they have purely real eigenvalues, and that their eigenvectors corresponding to different eigenvalues are orthogonal.

Unitary operators are operators whose inverse is equal to their conjugate, i.e.

$$U^\dagger = U^{-1}.$$

Unitary operators turn out to be useful for understanding the time evolution of a state. They preserve the inner product,

$$(|Uv\rangle \,, |Uw\rangle) = \langle v|U^\dagger U|w\rangle = \langle v|w\rangle = (|v\rangle \,, |w\rangle) \,.$$

A useful way of representing a Unitary matrix is its exponential form,

$$U = e^{i\gamma H},$$

where H is a hermitian matrix, and the exponential is understood as being a shorthand to refer to its taylor series expansion. This exponential would give a matrix, whose unitarity is proven by

$$U^\dagger U = (e^{i\gamma H})^\dagger e^{i\gamma H} = e^{-i\gamma H^\dagger} e^{i\gamma H} = e^{-i\gamma H} e^{i\gamma H} = I$$

1.7 A new kind of Diagonalisation

A point of difference in the course MA110 and the book was that of diagonalisability. As I understand it, in the course, the diagonalisability used was the more common one, and in the book, the diagonalisability mentioned is *unitary* diagonalisability.
The difference between the two, seems to be that—

An $n \times n$ matrix A is diagonalisable if it can be represented as a product $P\Lambda P^{-1}$. This turns out to be true iff A has n linearly independent eigenvectors.

An $n \times n$ matrix A is *unitarily* diagonalisable if it can be represented as a product $UDU^\dagger$, where U is a unitary matrix. This turns out to be true iff A has n linearly independent and *orthogonal* eigenvectors.

Diagonalization can be intuitively understood through a change-of-basis transformation. If A has n linearly independent eigenvectors, we can make an invertible $n \times n$ matrix P, using those vectors. Now,

$$AP = (A\left|v_1\right\rangle, \quad \cdots \quad, A\left|v_n\right\rangle) = (\lambda_1\left|v_1\right\rangle, \quad \cdots \quad, \lambda_n\left|v_n\right\rangle),$$

Let the basis constructed by all the eigenvectors (it is a basis since there are n linearly independent eigenvectors) be B. Then, we imagine that the right hand side of the above equation can be written as

$$(\lambda_1\left|v_1\right\rangle, \quad \cdots \quad, \lambda_n\left|v_n\right\rangle) = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}_B = D_B$$

Now, using the change-of-basis matrix (which is basically P itself, since we were originally in the standard basis), we get

$$D_B = PD$$

This is fairly intuitive. It is saying that our linear transformation P changes a matrix (all the column vectors in that matrix, thus the whole matrix itself) from being in the standard basis to the basis B. Now,

$$AP = PD \implies A = PDP^{-1}.$$

All that we required to prove the above was the necessity of n linearly independent eigenvectors. If the matrix P has to be unitary, we have the extra condition of $P^\dagger P = I$.

$$P^\dagger P = \begin{pmatrix} \langle v_1| \\ \langle v_2| \\ \vdots \\ \langle v_n| \end{pmatrix} \begin{pmatrix} |v_1\rangle & |v_2\rangle & \cdots & |v_n\rangle \end{pmatrix} = \begin{pmatrix} \langle v_1|v_1\rangle & \langle v_1|v_2\rangle & \cdots & \langle v_1|v_n\rangle \\ \langle v_2|v_1\rangle & \langle v_2|v_2\rangle & \cdots & \langle v_2|v_n\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle v_n|v_1\rangle & \langle v_n|v_2\rangle & \cdots & \langle v_n|v_n\rangle \end{pmatrix}$$

It is apparent that for $P^\dagger P = I$ to be true, the eigenvectors need to be orthonormal.

From here on out, *diagonalisable* will be used to refer to *unitarily diagonalisable*.

The *diagonal representation* of a matrix A is $A = \sum_i \lambda_i |i\rangle \langle i|$, where vectors $|i\rangle$ form an orthonormal set of eigenvectors for A.

1.8 Spectral Decomposition

The spectral decomposition theorem states that Any normal operator M on a vector space V is diagonal with respect to some orthonormal basis for V. Conversely, any diagonalizable operator is normal.

1.9 Tensor Products

Tensor products are used to describe the *combined state* of a number of qubits. For a tensor product, we multiply two kets together, instead of a ket and a bra as was done previously. It is defined by the Kronecker product, for two vectors:

$$|a\rangle \otimes |b\rangle = \begin{pmatrix} a_1 \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \\ a_2 \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{pmatrix}$$

and two matrices:

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}$$

Tensor products will turn out to be very important when we start with multi-qubit systems.

1.10 Operator functions

Let f be a function defined on the complex numbers, and $A = \sum_a a |a\rangle \langle a|$ be a spectral decomposition for a normal operator A. Then, the function f acting on A, $f(A) = \sum_a f(a) |a\rangle \langle a|$.

Chapter 2

Postulates of Quantum Mechanics

2.1 First Postulate

2.1.1 State space

The first postulate essentially says that to every isolated physical system, we can attach a Hilbert space, known as the *state space*. The system is completely described by its state vector.
An example is a qubit, its state space being the bloch sphere.
A qubit can be mathematically represented as

$$|q\rangle = a\,|0\rangle + b\,|1\rangle \qquad a, b \in \mathbb{C},\ \ |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

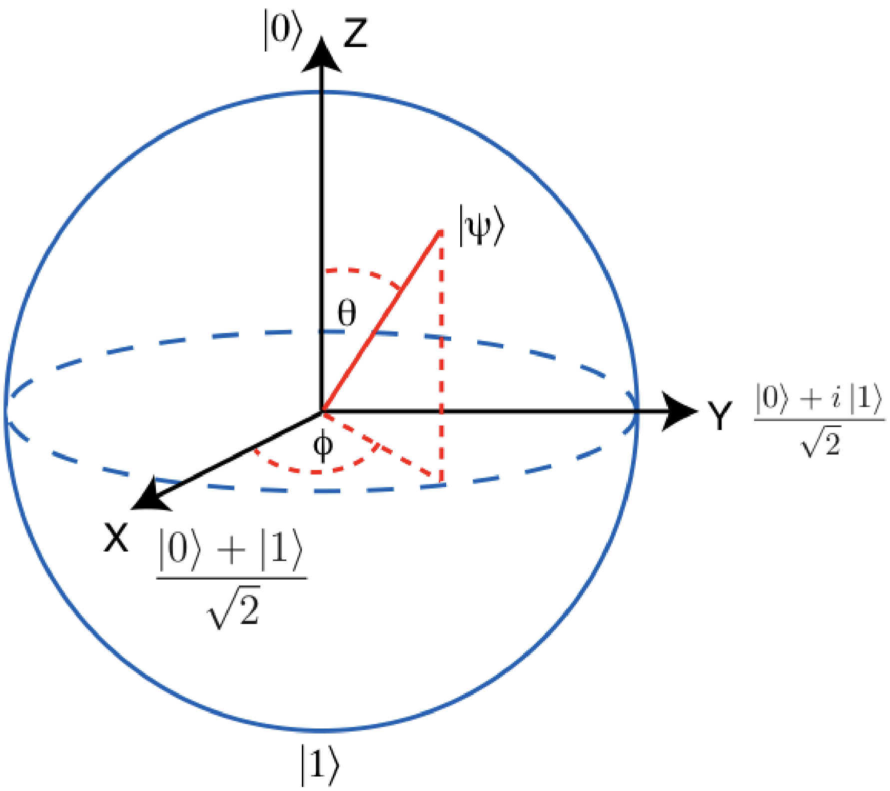
with the added condition of $|a|^2 + |b|^2 = 1$.
Since a and b are complex numbers, they can be written in their polar forms, as $|a|\,e^{i\theta_1}$, and $|b|\,e^{i\theta_2}$. The qubit can then be written as:

$$|q\rangle = e^{i\theta_1}\left(|a|\,|0\rangle + |b|\,e^{i(\theta_2-\theta_1)}\,|1\rangle\right)$$

Here, θ_1 is called the *global phase* and the quantity $(\theta_2 - \theta_1)$ is called *relative phase* and is denoted by ϕ.
I found that the global phase was always ignored from discussion, and the reason seems to be that it is not a measurable quantity, since the measurements of quantum physics rely on probabilities and in finding any probability the $e^{i\theta_1}$ would meet it's conjugate, becoming 1 and therefore going undetected.
The last substitution left is changing the amplitudes $|a|$ and $|b|$ to $\cos\left(\frac{\theta}{2}\right)$ and $\sin\left(\frac{\theta}{2}\right)$, to get:

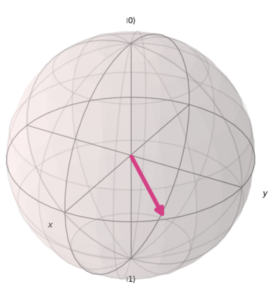
$$|q\rangle = \cos\left(\tfrac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\tfrac{\theta}{2}\right)|1\rangle$$

Since there are two independent angles at play here, we can take this to a sphere.



This is called the *bloch sphere*. Its surface represents the state space of the qubit. A vector from the origin to any point on the surface of the sphere completely describes a unique state of the qubit. It is called a *bloch vector*. For eg, if we plot the bloch vector corresponding to $\theta = \frac{\pi}{2}, \phi = \frac{\pi}{4}$, we get:

```
from qiskit_textbook.widgets import plot_bloch_vector_spherical
coords = [pi/2,pi/4,1] #theta, phi, radius
plot_bloch_vector_spherical(coords)
```



2.2 Second Postulate

2.2.1 Evolution

The evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ at time t_1 is related to the state $|\psi'\rangle$ at time t_2 by a unitary operator which depends only on t_1 and t_2

$$|\psi'\rangle = U\,|\psi\rangle$$

This is verified by the time-dependent Schrödinger equation:

$$i\hbar\frac{d\,|\psi\rangle}{dt} = H\,|\psi\rangle\,,$$

where H is the hamiltonian operator for the specific system. We know that all physical operators are hermitian, since they only have real eigenvalues. We can easily see that a solution to the equation is

$$|\psi(t_2)\rangle = e^{\frac{-iH(t_2-t_1)}{\hbar}}\,|\psi(t_1)\rangle = U(t_2,t_1)\,|\psi(t_1)\rangle\,,$$

since H is hermitian.
In the case of a qubit, a unitary matrix corresponds to a rotation of the $|\psi(t_1)\rangle$ vector, that makes it stay on the surface.

2.3 Third Postulate

2.3.1 Measurement

Quantum measurements are described by a set of measurement operators, $\{M_m\}$. These are operators acting on the state space of the system being measured. m indexes the measurements outcomes that may occur in the experiment. If the state is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle$$

and the state of system after measurement is

$$\frac{M_m\,|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}$$

The measurement operators satisfy the relation

$$\sum_m M_m^\dagger M_m = I.$$

So that the probabilities sum to one.
Let's take an example of a qubit. There are two possible states, so there will be two measurement operators, and they are defined by $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$. Then, for the qubit $|q\rangle = a\,|0\rangle + b\,|1\rangle$, the probability for measuring $|0\rangle$ is $\langle\psi|M_0^\dagger M_0|\psi\rangle$, which is $|a|^2$.

2.3.2 Distinguishing quantum states

Let the states $|\psi_i\rangle$ be orthonormal. A quantum measurement to distinguish these states, can be made by defining a set of measurement operators $M_i = |\psi_i\rangle\langle\psi_i|$, and an M_0 described by the positive square root of $(I - \sum_{i\neq 0}|\psi_i\rangle\langle\psi_i|)$, and if the state $|\psi_i\rangle$ is prepared then $p(i) = \langle\psi_i|M_i|\psi_i\rangle = 1$, so the result i occurs with certainty.
Thus, orthonormal states are easily distinguished. However, it can be proven that if the states $|\psi_i\rangle$ are not orthonormal, they *cannot* be reliably distinguished.

Let us imagine that there are operators M_j, with outcome j each. We can infer the index of the state using some function f such that $f(j) = i$. Suppose $|\psi_1\rangle$ and $|\psi_2\rangle$ are non orthogonal states. This means that $|\psi_2\rangle$ can be decomposed into a component parallel to $|\psi_1\rangle$ and a component orthogonal to it. This means that on applying some measurement operator M_j, there is a non-zero probability of getting an outcome j such that $f(j) = 1$. Thus, we cannot accurately identify which state was prepared.

2.3.3 Projective measurements

A projective measurement is defined by an *observable*, a hermitian operator M,

$$M = \sum_m mP_m,$$

where P_m is a projector onto the eigenspace of M with eigenvalue m, so that the probability of getting m upon measurement is $\langle\psi|P_m|\psi\rangle$. The *observable* comes in handy when calculating the average value of the measurement.

$$\sum_m mp(m) = \sum_m m\langle\psi|P_m|\psi\rangle = \langle\psi|\left(\sum_m mP_m\right)|\psi\rangle = \langle\psi|M|\psi\rangle.$$

Thus, we get the average value of the observable $\langle M\rangle = \langle\psi|M|\psi\rangle$.

2.3.4 POVM measurements

The acronym POVM stands for 'Positive Operator-Valued Measure'. Suppose a measurement is described by operators $\{M_m\}$. Then, the probability of outcome m for a state $|\psi\rangle$ is given by $\langle\psi|M_m^\dagger M_m|\psi\rangle$. We then define $E_m = M_m^\dagger M_m$. The operators $\{E_m\}$ thus are known as a POVM. The probability of outcome m is then:

$$p(m) = \langle\psi|E_m|\psi\rangle$$

2.4 Fourth Postulate

2.4.1 Composite systems

Till now, we have only talked about one qubit at a time. But if we want to use two or more qubits, or in general, we want to make a larger system consisting of many smaller systems, we describe the state space of the larger system to be the tensor product of the state spaces of the smaller systems.

Further, if we have state $|\psi_i\rangle$, i from 1 through n, respectively for the system number i, the combined state is represent by the tensor product $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.

This also helps us to define the rather bizarre idea of *entanglement*. Consider the state

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

The interesting property about it is that it cannot be represented as a tensor product of two qubits $|a\rangle$ and $|b\rangle$, yet, as we will see later, it is possible to construct this state. Due to this property, this state is called an *entangled state*, or a *bell state*.

Chapter 3

Quantum Circuits and Computation

3.1 Single qubit operations

Much like we talk about gates when discussing classical bits, we talk about quantum gates when dealing with qubits. In this case, though, all the gates are represented by 2×2 unitary matrices, that act on the state vector of the qubit. Some of the most important gates are the Pauli gates,

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}; Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

And also some additional gates,

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}; T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}.$$

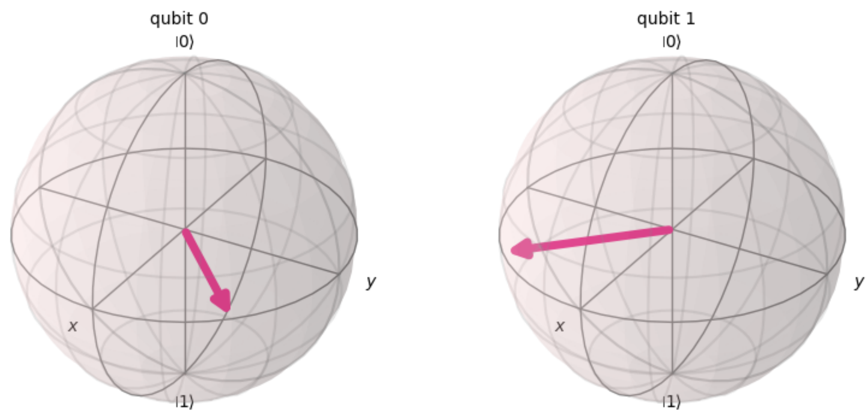
Let's take a look at these gates one-by-one.

3.1.1 X gate

The action of X gate on the computational basis is:

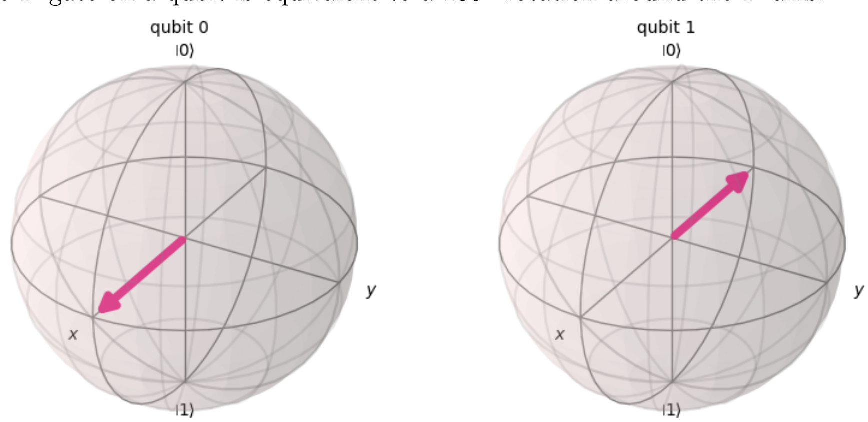
$$X\left|0\right\rangle = \left|1\right\rangle \qquad \text{and} \qquad X\left|1\right\rangle = \left|0\right\rangle$$

Thus, this gate is often called the NOT gate. More generally, for any state $\left|q\right\rangle = a\left|0\right\rangle + b\left|1\right\rangle$, $X\left|q\right\rangle = a\left|1\right\rangle + b\left|0\right\rangle$, switches up the amplitudes. If we try to visualize this on a bloch sphere, we realize that an X gate is equivalent to a 180° rotation around the X axis.



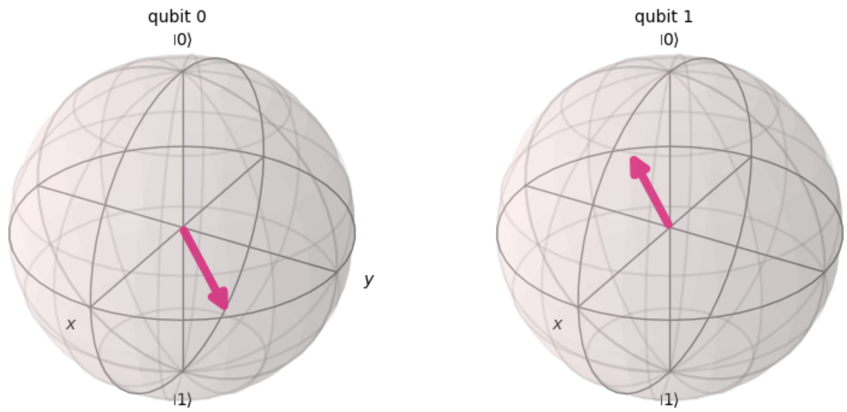
3.1.2 Y gate

The Y gate on a qubit is equivalent to a $180°$ rotation around the Y axis.



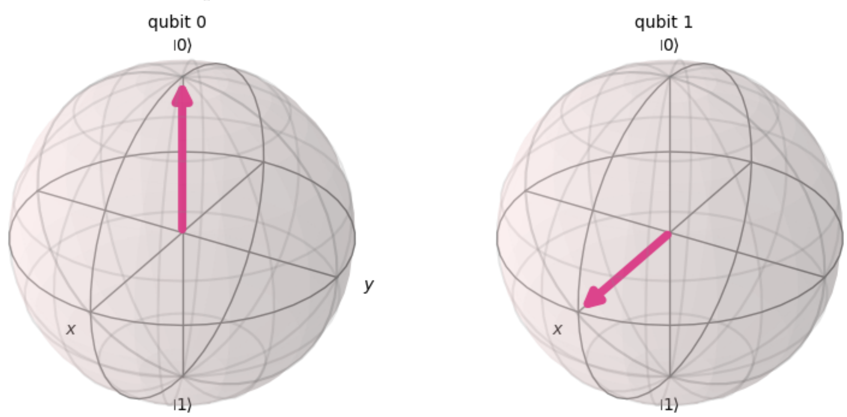
3.1.3 Z gate

The Z gate on a qubit is equivalent to a $180°$ rotation around the Z axis. It is also sometimes called the phase flip gate, since it changes the phase of $|1\rangle$ by π, but leaves $|0\rangle$ unchanged.



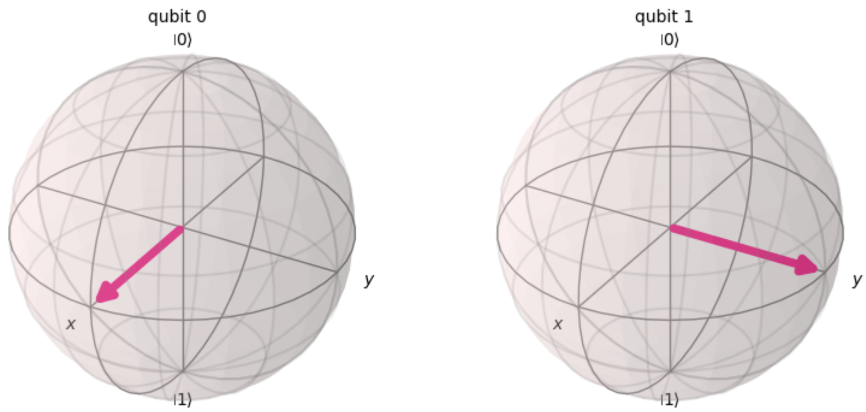
3.1.4 Hadamard H gate

The H gate on a qubit is equivalent to a 180° rotation around the $x = z$ diagonal axis in the XZ plane.



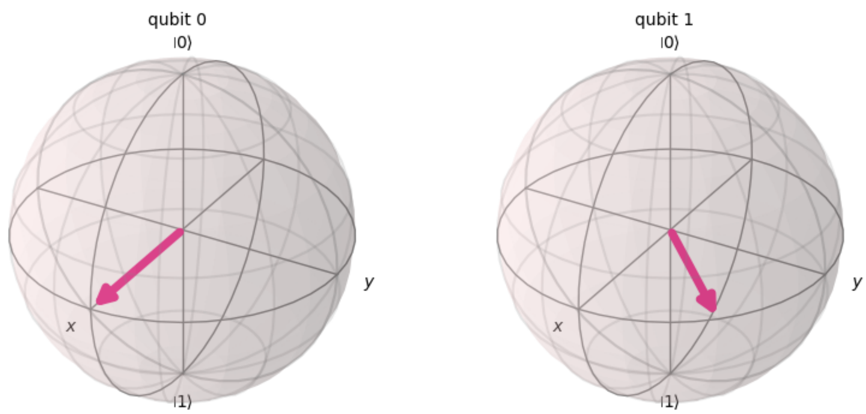
3.1.5 S gate

The S gate on a qubit is equivalent to a 90° anti-clockwise rotation around the Z axis.



3.1.6 T gate

The T gate on a qubit is equivalent to a $45°$ anti-clockwise rotation around the Z axis.



Thus, the T gate is also sometimes called the $\sqrt{S}$ gate.

3.1.7 General Rotation gate

Say that we need to rotate a qubit by an angle θ around an axis $\hat{n}$. Then, the rotation gate can be written by

$$R_{\hat{n}}(\theta) = exp(-i\theta\hat{n} \cdot \vec{\sigma}/2),$$

where $\vec{\sigma}$ is a kind of vector of all the three pauli matrices, (X, Y, Z), and $\hat{n} \cdot \vec{\sigma} = n_x X + n_y Y + n_z Z$. We notice that this will always give a hermitian matrix whose square is I. Let this matrix be named A. Thus, taking the taylor series expansion of the gate, we get

$$R_{\hat{n}}(\theta) = 1 + \frac{-iA\frac{\theta}{2}}{1!} + \frac{(-iA\frac{\theta}{2})^2}{2!} + \cdots = \cos\tfrac{\theta}{2} I - i\sin\tfrac{\theta}{2}\left(n_x X + n_y Y + n_z Z\right).$$

We can also verify this by trying to recreate the X, Y, Z, H, S or T gates using this formula.

3.2 Controlled operations

Controlled operations are basically operations on two or more qubits, in which the state of some qubits decides what changes are made to the state of the other qubits. In the case of two qubits, we call the first qubit the *control qubit*, and the second qubit the *target qubit*.

3.2.1 The CNOT gate

The CNOT (controlled-not), is a gate with two qubit inputs, the control and the target. The action of the CNOT gate is basically that if the control qubit is set to $|1\rangle$, the target qubit is flipped, otherwise the target qubit is left alone. Let us try to construct its matrix representation.

$$C_X\left|00\right\rangle = \left|00\right\rangle, C_X\left|01\right\rangle = \left|01\right\rangle, C_X\left|10\right\rangle = \left|11\right\rangle, C_X\left|11\right\rangle = \left|10\right\rangle.$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \implies \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Similarly, solving the other equations, we get

$$C_X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

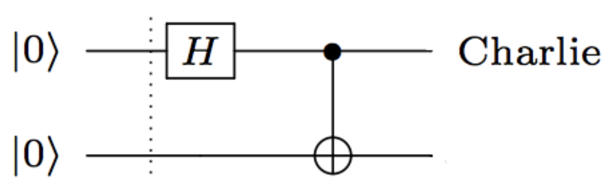
In general, a controlled-Unitary gate is C_U such that:

$$C_U\left|00\right\rangle = \left|00\right\rangle, C_U\left|01\right\rangle = \left|01\right\rangle, C_U\left|10\right\rangle = \left|1\right\rangle\left(U\left|0\right\rangle\right), C_U\left|11\right\rangle = \left|1\right\rangle\left(U\left|1\right\rangle\right).$$

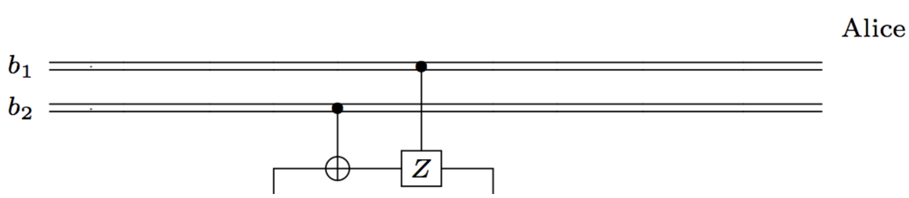
Chapter 4

Superdense Coding

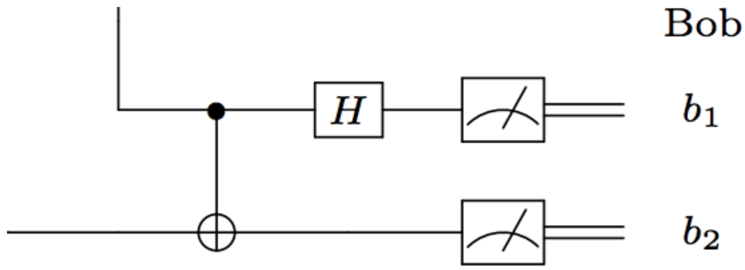
Suppose there are two scientists, Alice and Bob, who have a mutual scientist friend Charlie. Charlie gives them each a qubit, and tells them that they are entangled. He achieves this using a hadamard H gate and a CNOT gate, like following:



This has the effect of converting the statevector into a bell state, i.e. $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Now, he gives the first qubit to Alice, the second to Bob, and tells Alice to pick in her mind two bits of information — 00, 01, 10, or 11 and asks her to set up her apparatus like so:



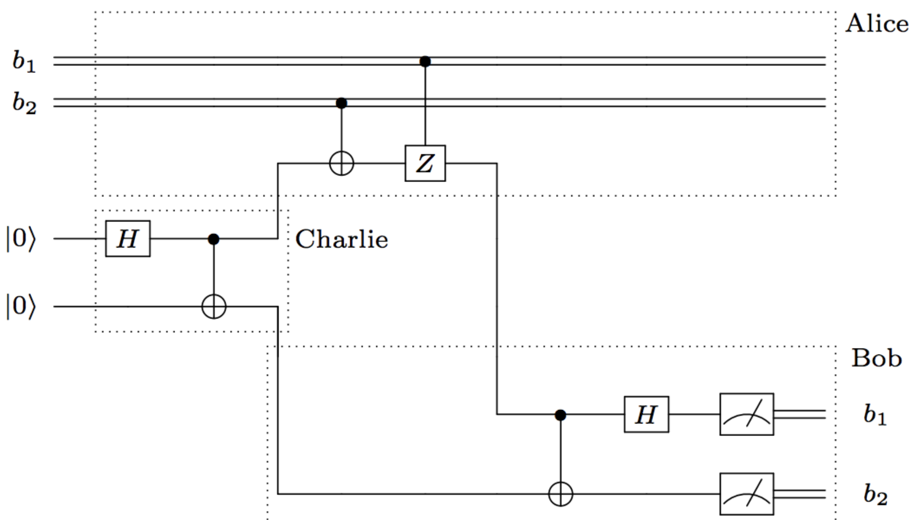
Basically, if b_2 is 1, apply the NOT gate, and if b_1 is 1, apply the Z gate. He now asks Alice to give her qubit to Bob, with the apparatus:



Bob applies his gates and measures both the qubits to get two bits of information, b_1 and b_2, and Alice confirms that it was the same b_1 and b_2 she had chosen! Voilà! They have communicated two classical bits of information by sending just one qubit! They both rejoice but are struck by horror as Charlie vanishes into a puff of smoke and logic.

What is happening here?

Superdense coding is another baffling application of quantum computing. It has rather far-reaching implications. This is what the complete circuit looks like:



Let us take the example of $b_1 = 1$ and $b_2 = 0$ (which only Alice knows), and keep track of the statevector after each gate. The initial statevector is just

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

But, after the hadamard gate on the first qubit, it is now

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$$

Then, comes the CNOT gate.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle \right)$$

Then, since $b_1 b_2$ is 10, only the Z gate is applied to the first qubit, which means

$$\frac{1}{\sqrt{2}} \left((Z|0\rangle)|0\rangle + (Z|1\rangle)|1\rangle \right) = \tfrac{1}{\sqrt{2}} \left(|00\rangle - |11\rangle \right)$$

Now, Bob receives the first qubit, and applies the CNOT again.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \left(|00\rangle - |10\rangle \right)$$

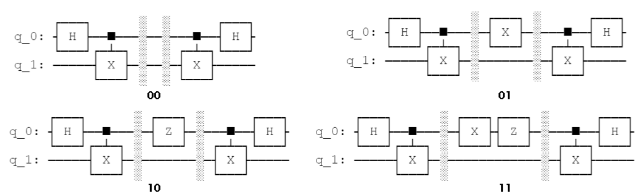
The H gate is now applied to the 1st qubit,

$$\begin{aligned} \frac{1}{\sqrt{2}} \left((H|0\rangle)|0\rangle - (H|1\rangle)|0\rangle \right) &= \frac{1}{\sqrt{2}} \left(\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \\ &= \frac{1}{\sqrt{2}} \left(\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = |1\rangle |0\rangle \end{aligned}$$

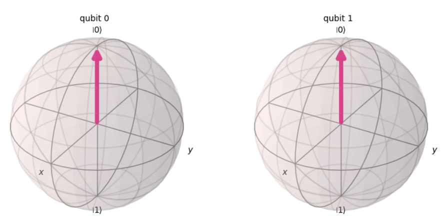
And thus we get back the original 2 bits of information. Here are all the 4 possibilities, tried and tested in qiskit.

4.1 Code

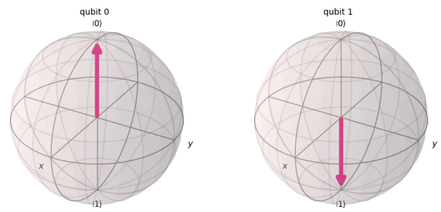
```python
from qiskit import QuantumCircuit, transpile, Aer, execute
from qiskit.visualization import *
backend = Aer.get_backend('statevector_simulator')
qc = QuantumCircuit(2)

#CHARLIE'S ENTANGLEMENT
qc.h(0)
qc.cx(0,1)
qc.barrier()
#ALICE'S ENCODING
b1 = None # 0 or 1
b2 = None # 0 or 1
if(b2):
  qc.x(0)
if(b1):
  qc.z(0)
qc.barrier()
#BOB'S DECODING
qc.cx(0,1)
qc.h(0)
#DRAW CIRCUIT
qc.draw()
```



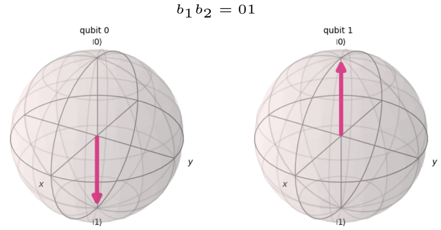

```python
#PLOT BLOCH SPHERE
out = execute(qc,backend).result().get_statevector()
plot_bloch_multivector(out)
```



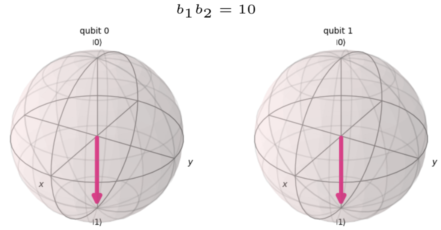
$b_1 b_2 = 00$



$b_1 b_2 = 01$



$b_1 b_2 = 10$



$b_1 b_2 = 11$

```python
#SHOW STATEVECTOR
from qiskit.visualization import array_to_latex
array_to_latex(out, prefix="\\text{Statevector} = ")
```

$$\text{Statevector} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{Statevector} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

00 **01**

$$\text{Statevector} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{Statevector} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$$

10 **11**

Revised PoA

As it stands, I am just a little bit behind according to my first Plan of Action, due in part to a vacation trip, and also being distracted by other chapters in the book that I didn't initially intend on convering in depth, in particular that of Superdense Coding. Now there's just the controlled operations and then the measurement part of quantum circuits left. After that, much of my PoA is still the same, but I shall try to make more room for understanding and playing around with qiskit, because I really enjoyed that part of this project.

- Week 5-6: Quantum Circuits: controlled operations, measurement. Quantum Fourier Transform. Qiskit.
- Week 6-7: Quantum Fourier Transform, Quantum Search Algorithms. Qiskit.
- Week 7-8: Qiskit.
- **Endterm Report Submission**