Introduction to Quantum Computing

Kenneth Wang

October 12, 2024

Contents

Why Quantum Mechanics? 1

Born out of the failure of classical physics to explain everyday phenomenon, quantum mechanics, and its extension to quantum field theory, is one of the most powerful and predictive theories in physics to this day. Understanding quantum mechanics has led to technological developments that range from laser barcode scanners to MRI machines. The promises of quantum mechanics in the future include new paradigms of computing and sensing. The goal of this introductory chapter is to gain an appreciation for the key conceptual ideas of quantum mechanics and its applicability in the everyday world.

1.1 Waves as Particles: The photoelectric effect

In a classical wave (think of water or the string of a violin), the energy of the wave is related to its amplitude. A higher wave carries more energy, and if a violin is plucked harder, it'll sound louder. This holds true, even for electromagnetic waves which have energy density given by

$$\mathscr{E}_{\mathrm{EM}} = \frac{1}{2}\epsilon_0|\mathbf{E}|^2 + \frac{1}{2\mu_0}|\mathbf{B}|^2 \tag{1.1}$$

However, experiments performed in the late 19th century began to change this perception. In 1887, Hertz discovered that ultraviolet light, whose distinguishing feature is its high *frequency* not amplitude, could create vigorous sparks between metal electrodes. The ejection of electrons from a material due to light was dubbed the **photoelectric effect**, and was a beautiful confirmation of the relationship between light and electromagnetism, a cornerstone of Maxwell's theory.

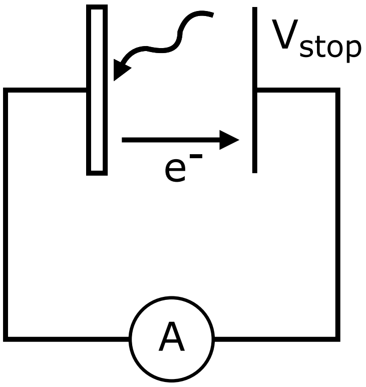

Figure 1.1: A schematic of Lenard's experiment. Light ejects electrons off of a metal towards an electrode. After hitting the electrode, a current is readable on the ammeter. A stopping potential V_{stop} can be applied to measure the energy of the ejected electrons.

In 1902, Lenard set up an ingenious experiment to measure both the **number** and **energy** of the ejected electrons from a metal due to a source of light. The ejected electrons from the metal were guided to a collector and the current was measured with a precision ammeter. The energy of the electrons was measured by applying a potential V_{stop} to the collector to repel the electrons. What Lenard found was that by increasing the brightness (amplitude) of the light source, the current increased, but V_{stop} was unaffected! In fact, the frequency of light impacted V_{stop}, where higher frequencies required larger stopping potentials. This confirmed Hertz's earlier observation of the potency of ultraviolet light.

These observations were finally explained by Einstein in 1905[1]. He postulated that the energy from light must come from *packets* or *quanta*. Each quantum of light has energy that is proportional to frequency

$$E = h\nu \tag{1.2}$$

where h is the Planck constant[2]. Ultraviolet light, which is at a higher frequency, imparted more energy on each ejected electron, while the

[1] This year was known as Einstein's *annus mirabilis* (or miracle year). Einstein published 4 papers: an explanation of the photoelectric effect, Brownian motion, special relativity, and the mass-energy equivalence. He won the Nobel Prize in 1921 for the photoelectric effect. Let us all have our miracle years!

[2] Max Planck introduced this constant to explain the blackbody radiation spectrum and solve the ultraviolet catastrophe in 1900. This experiment is another key result in the beginnings of quantum mechanics.

amplitude increased the *number* of quanta, and hence the number of elected electrons (the current in Lenard's experiments). It is worth noting here that an increased amplitude *does* increase the energy (there is no conflict with classical thinking), but it increases it via the *number* of quanta. Each quanta's energy is related to its frequency.

With the introduction of energy quanta, it became clear that light, which was classically thought of as a wave, *also has particle like behavior*. These particles of light became known as **photons**. Each photon also has a momentum given by

$$p = \frac{E}{c} = \frac{h\nu}{c} = \frac{h}{\lambda} \tag{1.3}$$

where $E = pc$ is the energy of a massless relativistic particle, c is the speed of light and λ is the wavelength.

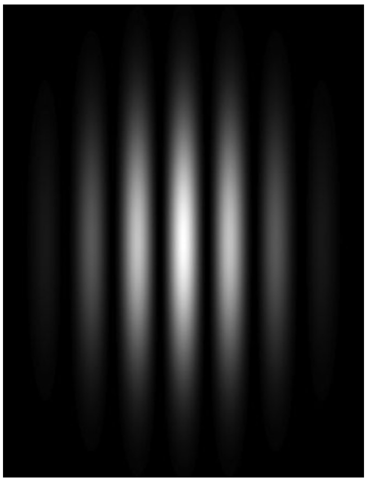

Figure 1.2: A simulated image of the interference pattern when light passes through two slits. Bright stripes correspond to locations of constructive interference, and dark stripes correspond to locations of destructive interference. Without interference, only two spots are expected at the location of the two slits.

1.2 Particles as Waves: Interference of individual electrons

Now, we discuss a situation where particles can exhibit wave-like behavior, which is at the heart of the description of quantum mechanics. Another classical wave phenomenon is interference. Wave amplitudes can add together constructively to create larger waves, but also cancel each other destructively. Applying this principle to waves sent through multiple slits give rise to interference patterns. Thomas Young, in 1803, demonstrated that light was a wave by experimentally verifying these interference patterns.

However, would particles, such as electrons, exhibit such behavior? In 1927, Clinton Davisson and Lester Germer showed that when electrons are diffracted off a crystal of Nickel, an interference pattern due to its crystal structure was visible in the locations of the diffracted electrons. A double-slit experiment was finally performed by Claus Jonsson in 1961 further confirming that electron beams did exhibit wave-like behavior. Perhaps most surprisingly, Pier Giorgio Merli, Giulio Pozzi and GianFranco Missiroli in 1976 performed a double-slit experiment with *single* electrons. Although one might conceive that beams of electrons can contain individual electrons which interfere with others, Merli, Pozzi and Missiroli's experiment showed that single electrons also *interfere*, and they interfere with themselves[3]! In fact, this is in agreement with quantum mechanics where individual particles are treated as having **wavefunctions** and being waves themselves.

Just as light can have a momentum which is related to its wavelength, massive particles have a wavelength that is associated with its momentum. Letting $p = mv$ and using equation 1.3, we can solve for the **de Broglie wavelength**

$$\lambda = \frac{h}{mv} \tag{1.4}$$

[3] Akira Tonomura and colleagues at Hitachi performed a single-electron double-slit experiment in 1989. A video is available on YouTube, where you can see the electrons accumulating one at a time, yet together they eventually display an interference pattern. I encourage you to look it up!

For massive particles, like electrons, to diffract, the size of the slits must be on the same order as the de Broglie wavelength. The idea that particles can behave like waves and waves can behave like particles is known as **wave-particle duality**[4]. This surprising experimentally based discovery required a new understanding of physics and led to the theory of quantum mechanics.

As we begin to think of particles like waves and waves like particles, it is important to stop briefly for a philosophical interlude. It is easy to get caught up in what is truly physical reality or what is actually happening, which is often rooted in one's daily experiences. Unfortunately, daily experiences are heavily influenced by the laws of classical mechanics, which is precisely why theories like electromagnetism and quantum mechanics[5] were difficult to accept and required paradigm shifts. To avoid these hairy issues, I advise you to always remember that quantum mechanics is indeed *backed up by real physical experiments*. Regardless of the exact interpretations (which might be held back by classical conceptions), our goal is to develop a predictive and explanatory framework for physical phenomenon. This goal is not impeded, no matter how one might want to interpret the results, as long as the same results are acquired. With that prelude, we now turn to gain more intuition on how quantum phenomenon *are* actually present in everyday phenomenon, and can start forming the basis for some intuition.

1.3 Line Spectra in Atoms

Another one of the first areas in nature where features of quantum mechanics were elucidated was in the spectra of atoms. The concept of an atom, from the Greek *atomon*, meaning indivisible, was first introduced by Democritus in ancient Greece. He proposed that atoms made up all matter, and this theory gained steam in the 19th century, where a collection of scientists including Antoine Lavoisier verified the composition of water being two parts hydrogen and one part oxygen. Through explorations of cathode rays, J.J. Thompson in 1897 concluded that atoms were divisible after all and included subatomic particles, like negatively charged electrons. The discovery of the electron was awarded the Nobel Prize in 1906. Attention then turned to the structure of the atom. These electrons were initially thought to be evenly distributed across a positively charged background that composed the atom, and was coined the *plum pudding model*. In 1911, Ernest Rutherford bombarded gold foil with alpha particles, and found that most particles passed straight through. This experiment confirmed that the atom was mostly empty space, and he posited that a positively charged nucleus was concentrated in the center of the atom. Around the same time and earlier, scientists discovered that light is composed of different colors, and that gases of certain chemical elements emitted light of particular colors or frequencies. Using the insight from Einstein, these frequencies correspond to a discrete set of energies. In order for atoms to emit only a discrete spectrum, Niels Bohr postulated that electrons orbited the nucleus at fixed distances, which corresponded to a discrete set of possible energies.

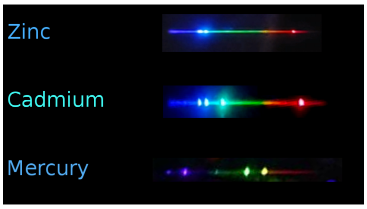

Figure 1.3: Line spectra for various different elements. There are distinct wavelengths where there is a peak of emission.

[4] As we will see in this book, quantum mechanics places an emphasis on treating particles like waves. In this sense, particles have a very special place, as wavefunctions are written for particles or collections of particles. However, in many high energy processes, particles can be created or destroyed. The key insight in **quantum field theory** is that the fundamental objects are actually quantum fields that pervade all space. Particles are merely excitations in the quantum field. This is a topic for another course.

[5] For more information, I recommend Peter Dear's book, The Inteligibility of Nature.

Attempting to match experimentally measured spectra, Bohr proposed that these orbits were quantized such that the angular momentum of the electron were integer multiples of $\hbar = h/(2\pi)$

$$mvr = n\hbar \tag{1.5}$$

where $n = 1, 2, 3, ...$ Here, Bohr indeed finds that the same Planck constant that had been the key to understanding photons and blackbody radiation should be used here! The circular orbit requires that the centripetal force equal the electrostatic force between the electron of charge $-e$ and nucleus (for Hydrogen) of charge e:

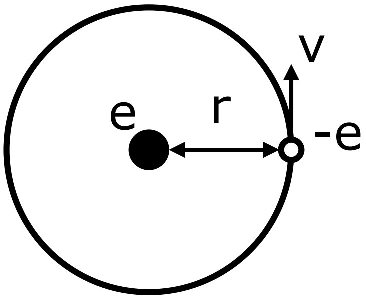

Figure 1.4: A simplified model of the Hydrogen atom where an electron circularly orbits a proton at radius r, and velocity v.

$$\frac{mv^2}{r} = \frac{e^2}{4\pi\epsilon_0 r^2} \tag{1.6}$$

Combining equations 1.5 and 1.6, we can solve for the unknowns, v and r of the orbits:

$$v = \frac{e^2}{4\pi\epsilon_0 n\hbar}, r = \frac{n^2\hbar^2(4\pi\epsilon_0)}{me^2}. \tag{1.7}$$

The energy of the orbit is given by the kinetic plus the potential energy:

$$E = \frac{mv^2}{2} - \frac{e^2}{4\pi\epsilon_0 r} \tag{1.8}$$

$$= -\frac{me^4}{2(4\pi\epsilon_0)^2 n^2\hbar^2} = -\frac{\text{Ry}}{n^2} \tag{1.9}$$

where the Rydberg constant is defined from this formula. Remarkably, this simple model, along with this phenomenological quantization that depends on h, does in fact agree with experiment and agrees with a more sophisticated model that we will develop later in this book. This quantization of energy, first seen in atoms is key to understanding many physical phenomena as well as designing technologies that are used everyday.

1.4 Everyday Quantum Mechanics

The idea that energy levels in atoms and molecules are discretized is leveraged in many modern technologies. Discretization leads to atoms and molecules having particular absoprtion and emission transition frequencies. In contrast to incandescent light bulbs, which produce a continuous (white) light spectrum from heat[6], neon lights use neon gas to give off a distinctive orange glow. Other so called "neon lights" use hydrogen for purple-red light or mercury for blue light. In all these cases, the gas is ionized via *electricity*, and then they relax to lower energy states emitting particular wavelengths. A more controlled emitter and the basis for many modern lasers are based on semiconductors. Even in the solid state, quantum mechanics predicts a "band-gap", or a gap in the energy spectrum, preferentially allowing for emission at a particular frequency.

[6] A hot light bulb can be approximated as a blackbody

LED (light-emitting diode) lights use a combination of laser diodes at different frequencies to produce white (or apparently white) light of different "temperatures". The reason LED lights are far more efficient than incandescent ones is precisely the ability to concentrate the energy into particular frequencies combined with the human eye's lack of need for a continuous spectrum. We see white just fine, even if it's actually only a few wavelengths put together[7].

Concentration of energy at a particular frequency is a simplified definition for a laser[8]. Lasers have seen use as barcode scanners where the reflection is measured to determine the pattern of black and white that form a barcode. The same principle is also used in security systems where a blocked laser beam indicates an obstruction, which could signal a thief in your home. When you make a phone call with your cellphone, a *proximity sensor* uses a infrared laser to sense that you are close to your phone and therefore locks your screen to prevent "ear touches". This concentration of energy in a laser can also be used to heat up objects. Have you ever tried to kill an ant with a magnifying glass in the sun? In this case, you are concentrating the energy from the sun. High power lasers are used in laser cutting, which can even cut through steel.

High powers are required when an object does not absorb very much in the frequency range of your source. However, as we have discussed atoms and molecules have very particular frequencies. Water is known to have transitions in the microwave region, which is how the modern microwave works. It targets the frequency of a transition in water around 2.45 GHz, causing the water molecules to heat up and warm up your food. It is dangerous to microwave metal, precisely since it does not have a transition and this frequency and will reflect the powerful waves that your microwave emits, causing other areas of the microwave to be hit by the radiation.

The selectivity of radiation also allows for non-invasive imaging that is essential in modern medicine. Our skin reflects visible light[9], but it allows radio frequencies through just fine. This transparency to radio frequencies allows this radiation to probe structures *inside* our bodies. It turns out the nuclei of hydrogen (protons) have transitions in these radio frequencies. Since hydrogen is prevalent in the body in many compounds, radio frequencies can be used to measure water and fat content in the body[10]. This imaging technique is known as magnetic resonance imaging (MRI) and is a key diagnostic tool for soft-tissue, which does not have the type of crystalline solid structure to diffract x-rays. MRI imaging has been used in joints to identify tears in ligaments, but also in the brain to identify the onset of Alzheimer's disease.

These particular examples just scratch the surface of how quantum phenomenon *are* present in our everyday lives. As we progress in this book, other relevant examples will be brought up when necessary. Since quantum mechanics is a very non-intuitive subject, I believe that real world examples are especially important to grasp the concepts. Plus, they are just really cool.

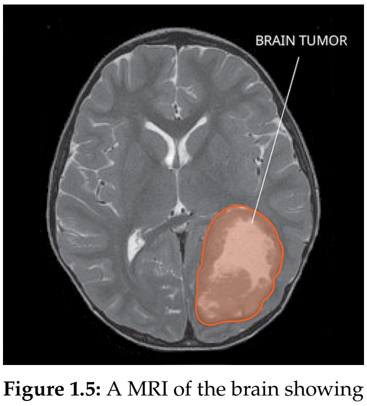

Figure 1.5: A MRI of the brain showing a brain tumor.

[7] In many cases, it's only 3! The so-called RGB color encoding comes from this idea.

[8] A laser is an acronym for **l**ight **a**mplification by **s**timulated **e**mission of **r**adiation. The stimulated here, in some sense, refers to a transition frequency being favored. Lasers are said to emit *coherent* light, compared to a light bulb's *incoherent* light. Higher quality lasers often have a smaller *linewidth*, which describes exactly how narrow their spectrum is. Typical semiconductor lasers for research use have linewidths of 1 MHz or lower. Laser pointers can have linewidths of a few nanometers. Visible light has frequencies of 100s of THz (or wavelengths of 100s of nanometers), so it is quite remarkable how concentrated coherent light from a laser is! It is worth noting that another requirement typical for a laser is a good spatial *mode*. In other words, lasers should propagate in a straight line, and not diverge too much. LEDs for lighting do not have this requirement, since it would be hard to light up a room with individual laser beams.

[9] Though not the same can be said about our friend *C. elegans*.

[10] If you've ever taken a MRI, you may have realized that you need to be placed into a big magnet. This giant magnet is required to create the discrete spectrum needed for spectral selectivity. It is worth noting that the chemical environment of the protons (what molecule the protons are a part of) also impacts these very precise frequencies. Chemists use this fact all the time in a technique called nuclear magnetic resonance (NMR) to fingerprint their compounds.

1.5 Spin and Measurement

We have discussed many of the salient features of quantum mechanics that are leveraged in modern technologies, namely the concept of quantization of energy. However, experiments uncovered another key area where quantization appears that will need to be captured in any quantum theory, and also served as a verification of the powerful framework of quantum mechanics that began in the early 20th century. This area is the quantization of *intrinsic angular momentum*, typically called **spin**.

Otto Stern and Walther Gerlach in 1922 performed their famous **Stern-Gerlach experiment**, where a beam of hot silver atoms was sent through an inhomogeneous magnetic field. If these silver atoms have a magnetic moment, $\boldsymbol{\mu}$, then their energy in this magnetic field is given by

$$U = -\boldsymbol{\mu} \cdot \mathbf{B} \tag{1.10}$$

and subsequent force is given by:

$$\mathbf{F} = \boldsymbol{\mu} \cdot \nabla \mathbf{B} \tag{1.11}$$

The inhomogeneous magnetic field in the Stern-Gerlach experiment created a gradient in the z-direction, which then deflects the atoms in the z direction. After this magnetic field region, the silver atoms would be detected on a screen. One would expect a distribution of silver atoms at different z positions on the screen arising from different z components of the angular momentum. Surprisingly[11], the experiment showed two distinct spots of atoms. This observation suggests:

- Atoms can only have two discretized values of the magnetic moment in the z direction.
- Even though the silver atoms were initially in a thermal distribution of magnetic moments that can point in any direction, the apparatus itself has *forced* the atoms to have a value for the magnetic moment in the z direction.

The first consequence is simply a verification of the quantization of angular momentum. However, the second consequence has a deeper mysterious meaning in quantum mechanics, which has been the subject of much debate and many interpretations of quantum mechanics[12]. The idea that the atoms have been *forced* to have a value in the z direction when it could have conceivably not is known as **measurement**. The apparatus could be said to be a *detector* for the magnetic moment in the z direction. Upon detection, the particle will only take on its allowed values[13]. Furthermore, if a second Stern-Gerlach apparatus is placed after just one of these beams, the particles will deflect in the *same* direction as in the first apparatus. This phenomenon has come to be known as **collapse of the wavefunction**, which captures the idea that after measurement, a quantum system will then be in the state that the measurement suggests. Despite initially being in a thermal distribution, the silver atoms have taken a stand, so to speak, and are now firmly in this state, even upon subsequent measurement. Measurement has been one of the most puzzling features of quantum mechanics and will need to be dealt with carefully. However, it is once again apt to mention that

[11] Surprising to us, but Stern and Gerlach had expected this result! The experiment was designed specifically to test the quantization of angular momentum (and thus magnetic moment) that was used by Bohr to explain the line spectra of atoms.

[12] For a discussion of this, check out Steven Weinberg's *Lectures on Quantum Mechanics*

[13] Precisely, the eigenvalues of the observable, but that will be discussed later!

we are developing mathematical frameworks of conceptual models to understand and predict experimental results! If it is non-intuitive, stick to the guns of the framework and don't get too caught up on what it actually means!

1.6 Summary

Through many beautiful experiments starting from the 19th century and continuing to the modern day, quantum mechanics has been verified time and time again. Grossly non-intuitive, quantum mechanics posits that light, classically thought of as waves, requires a particle like description and likewise, electrons, classically thought of as particles, requires a wave like description. To understand the line spectra of atoms, a heuristic quantization condition was introduced by Bohr. Although we did not elaborate here, quantization can be seen as a consequence of the wave-like behavior of particles analogous to how a string can only support a discrete set of frequencies, corresponding to the supported standing waves. Discretization of frequencies enables technologies like efficient lighting, lasers and medical imaging. Equating all of quantum mechanics to a discretization of energy is a gross oversimplification, but it is worth noting all the consequences that such a discretization of energy results in. One of these consequences is the ability to well-define an isolated pair of states to act as a 0 or 1 for the purposes of computing. This is the goal of these notes. With this prelude, I hope I have made you excited about the journey we are about to embark on. To really appreciate, understand and make use of quantum mechanics for quantum computing, we will need to engage in mathematical framework that will require concepts from linear algebra to calculus. However, I hope you always keep this introductory chapter in mind, and not get lost in the throws of mathematical calculation.

What is computing? 2

Since the dawn of time, humans have needed to perform calculations. Calculations lie at the heart of any sort of market and thus is responsible for trading, which then allows specialization. After all, I no longer need to be a farmer if I can trade some meat I hunt for your corn. While it might be easy to give you 100 grams of beef for 500 grams of corn[1], what if I had 1500 grams of beef? Or 1688 grams of beef? The first systems of calculation involve tracking physical objects, such as an abacus, where each stone represents a quantity of 1, 10 or 100 depending on which row it is in. Our first attempt at making these calculations probably involved a piece of paper and some set of instructions for performing operations such as addition, subtraction, multiplication and division. These days, you'd likely open the calculator app on your smartphone and press some buttons which magically[2] delivers the right answer to your question. You may even use an artificial intelligence tool[3] like chatGPT to learn about quantum computing, which is a far more abstract question than adding two numbers. I would classify all of these as computations and the job of a *computer* to calculate. So far, (nearly) all calculations have been performed using systems that more or less make use of *classical* physics, but the goal of these notes is to ask whether features of quantum physics can change the *types* of problems that can be calculated? Or whether it can perform these calculations more *efficiently*? In fact, these questions are all active areas of research and the full answers are *unknown*. We will attempt to make some headway on this problem in these notes.

Let's now ask what are some of the fundamental components of computation. Well, if we think carefully, each of the examples above required some sort of **representation** of our computational object (which we can refer to as data), and then some sort of **instructions** for how to manipulate the representation and read out our answer. In the case of the abacus, the representation is via physical stones, while in modern day, the representation is with arabic numerals and symbols that can be printed with ink on a page. While we keep the notion of instructions abstract for our purposes [4], we simply take note that in these very different representations, the instructions may be *different*. Another key point about instructions is that they themselves follow logical flows. Precisely, instructions can involve the english words, and mathematical concepts of "if", "and", or "or". Fundamentally, computations simply involve the manipulation of input data that then returns an answer, which we hope solves our initial problem.

While the first computers and algorithms involve purely mechanical objects [5] and movements, these quickly become cumbersome. Implementing logical operations can be quite difficult. With the advent of electrical signals and circuits, computing power has exploded. It turns out it is much easier (not necessarily simpler!) to manipulate electrical signals, and an electrical representation of our data than mechanical ones. It is also far faster, as electrical signals can change more quickly than mechanical ones. We now turn to understanding the simple building blocks of a computer.

[1] Don't ask me if these are reasonable numbers. Also, as a sidenote, a system of weights is also crucial for trade!

[2] n.b. It is not magic

[3] n.b. This is not magic either.

[4] Fear not, this can also be made very precise with something known as a *Turing machine*

[5] Some of these are amazing! Read up on the slide-rule calculator, if you don't know it already.

2.1 Data Representation as a Bit

Electrical signals have a *voltage* corresponding to them. You can think of the plus and minus terminals of a battery as having low and high voltage respectively [6]. If your electrical circuit cares about the precise value of the voltage, it is known as an **analog** circuit. For instance, we can represent the number 3.47 as 3.47 V on an electrical signal. However, analog circuits are highly susceptible to *noise*, which is any source of unwanted changes in the voltage. These can come from thermal fluctuations for instance. An important innovation in computing was the realization that a more noise-resistant representation of information can be used if we limit it to discrete options. For example, we can simply state that *any* voltage above 1.8 V is state 1, and *any* voltage below 1.8 V is state 0. As long as we stay safely above or below this threshold (higher or lower than any amount of noise), our data becomes robust. This object that can be either 0 or 1 is known as a **bit**, and is the fundamental data object for classical computing. Circuits using bits as signals are known as **digital** circuits.

Wait one second though! Analog voltages can represent an infinite set of values, but it seems that a bit can only represent a finite set of values (actually only 2), and thus can only carry a finite amount of information. This is indeed true, so in order to represent a greater set of values, we need to combine bits together. If we had two bits, we can now represent four values 00, 01, 10, 11. The growth of the amount of information that can be represented by n bits grows very quickly as 2^n. With 10 bits, we can represent $2^{10} = 1024$ states, and by the time we get to 64 bits, we arrive at an astounding $2^{64} \approx 1.84 \times 10^{19}$ states. This is good from an information representation standpoint, but it is precisely this same reason (as we will see) which makes understanding quantum systems difficult and is perhaps where some of the mystifying power of quantum computing comes from.

The most rudimentary data to represent are simply integers. We can map a bitstring to an integer in the following way:

$$10110 \longrightarrow 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 22 \tag{2.1}$$

You may have encountered this previously as representing a number in base 2. Base 2 arithmetic works similarly to base 10 arithmetic, and many algorithms can be adapted. For instance when adding two base 2 numbers, you will need to “carry" every time you get to 2 instead of 10 in base 10. However, mappings to states can be very general. In addition to representing numbers, bits can represent, well basically anything! For instance, we can decide (somewhat arbitrarily) that 00 ought to represent the letter “A", 01 to represent “B", 10 to represent “C", and 11 to represent “D". We can also use it to represent symbols in your language. There are standards known as ASCII and Unicode which standardize the mapping between bits to characters in any language.

In addition to representing letters, they can also represent groupings of numbers of letters. For instance, the pixels on your computer monitor is likely described by a triplet of numbers, which give the red, green and blue value of that pixel. Suffice it to say, *everything* in your computer is represented by bits, and bits are indeed capable of representing anything

[6] Technically, voltage is always a difference between two terminals. We can say that the negative terminal is at low voltage (actually zero) *relative* to the negative terminal, and the positive terminal is at some positive voltage relative to the negative terminal. Typically, this voltage is around a few volts depending on the battery type.

you can think of. These bits themselves are represented by electrical states.

2.2 Manipulation of Bits

Now that we have bits to represent our objects, we now need to know how to manipulate them. These bits are manipulated by logic gates, which themselves are made up of small electrical components known as **transistors**. A scientific description of how transistors work is beyond the scope of these notes, but the first transistor was made in 1947 by John Bardeen, Walter Brattain and William Shockley at Bell Labs which got them the Nobel Prize in 1956. The transistor was really the birth of digital computing with the computers as we know them today. Anyway, back to our logical gates. Our simplest gate for a single bit is the NOT gate, which simply inverts or negates a signal. In other words, if it's high, it becomes low, and if it's low, it becomes high. A gate is described by its truth table, which tells us what the gate does to every input. The truth table for the NOT gate is quite simple:

x	$\neg x$
0	1
1	0

The NOT operation can also be represented graphically as a circuit symbol and also by the symbol ¬ (e.g. $\neg x$). This is about all you can do, except just passing a signal through unchanged, which does not warrant a table of symbol.

With two bits, we now have more interesting operations such as AND ($\wedge$) and OR ($\vee$) with the following truth tables.

x	y	$x \wedge y$	$x \vee y$
0	0	0	0
1	0	0	1
0	1	0	0
1	1	1	1

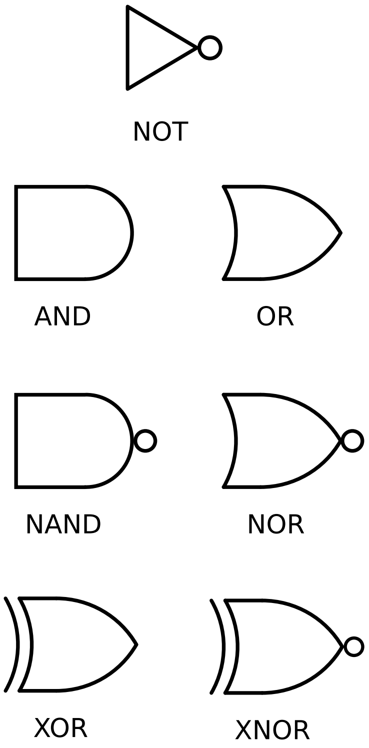

Figure 2.1: Circuit symbols for the various gates discussed in the main text. Note that NOT tends to be represented by a circle.

They have the intuitive reasoning that AND only returns true if both inputs are true, while OR only returns true if either one of the inputs is true. Note that we are implicitly using 1 to represent the concept of "true" and 0 to represent "false". There are the "not" versions of the AND and OR, namely NAND ($\neg(x \wedge y)$)and NOR ($\neg(x \vee y)$). Another important gate is the exclusive or, otherwise known as XOR, symbolized with $\oplus$. XOR is 1 as long as either 1 of the inputs is 1, but *not* both. The truth table is as follows

x	y	$x \oplus y$
0	0	0
1	0	1
0	1	1
1	1	0

We note that the XOR operation is equivalent to addition modulo 2, which simply means that after adding two numbers, we take the remainder of division by 2. Let's put together these gates to do something useful.

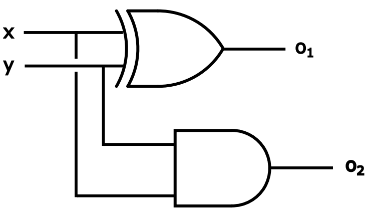

Figure 2.2: Circuit for example 2.2.1

Example 2.2.1 Adder. Consider the following circuit in figure 2.2.

Let's work out the truth table. If x and y are both 0, then the XOR will return 0, and the AND will also return 0. This results in an output o_2o_1 of 00. Now, if either x or y is 1, then the XOR will return 1, and the AND will return 0. Thus, the output is 01. Lastly, if both x or y is 1, then the XOR will return 0, while the AND will return 1, resulting in the output 10. Thus, this circuit is an adder and works out $x + y = o_2o_1$ in binary, where we note that o_2o_1 is *not* o_2 times o_1, but is just a representation of bit concatenation.

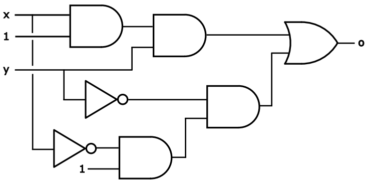

Figure 2.3: Circuit for example 2.2.2

Example 2.2.2 Classical Controlled NOT. Consider the circuit in figure 2.3.

For this one, we should just work out the full truth table first, before we try to interpret.

x	y	o
0	0	0
1	0	1
0	1	1
1	1	0

We can think of this as a NOT operation on y, controlled by x. In other words if x is 0, then y just passes through to the output. However, if x is 1, then NOT is performed on y. From this perspective, we see that the two left most AND gates are simply checking if x is equal to 0 or if $\neg x$ is equal to 0, which selects the branch. The $\neg x$ branch is then ANDed with $\neg y$, while the x branch is ANDed with y. Lastly, the two branches are combined with an OR, since we are guaranteed that one of the two branches is false, and thus its AND output is 0. Thus, this final OR gate will simply return the final output.

Here, two remarks are in order.

1. If we wanted to, we could express the circuit using algebraic notation. However, it appears quite cumbersome.

$$o = ((x \land 1) \land y) \lor ((\neg x \land 1) \land \neg y) \tag{2.2}$$

   The advantage of this notation is that it may help us simplify circuits or express them using different logical gates. For instance, one can make use of De Morgan's laws ($\neg(x \land y) = \neg x \lor \neg y$ and $\neg(x \lor y) = \neg x \land \neg y$) to do some of these conversions.

2. A nice procedural way to describe this circuit was described in the above text and in the following "pseudocode", which for now, we can just think of as some instructions.

```
if x is 1:
    return not y
else:
        return y

```

We will see that procedural instructions are usually a more natural way to compute and easier to come up with. It is worth noting that as far as a circuit like figure 2.3 is concerned, the operation is *not* sequential as the instructions would suggest, and in fact the wires change their state nearly instantaneously (up to some gate delays, which are typically at the nanosecond scale). For instance, the circuit clearly calculates *both* y and $\neg y$ and has both at some intermediate point in the circuit. The pseudocode suggests that one only calculates $\neg y$ if $x = 1$. This slight, but subtle distinction should be appreciated. Procedural sets of instructions are most alike those running currently on your computer and is closely connected to the idea of a *program*. Microprocessors are great at these sorts of things, and have fixed instruction sets. Circuits are created in labs from gates in chip form. They are also the goal of the FPGA (field programmable gate arrays), which have dynamically programmable circuits. It is worth noting that if you need a task done quickly, FPGAs will typically be faster than microprocessors. After all, the circuit is *already* synthesized on the hardware, and will “calculate" as fast as nature allows[a]! It takes time to interpret instructions and run them.

[a] Typically, we throttle the calculation speed with a clock for the purposes of synchronization.

It turns out, somewhat remarkably, that *any* function of n input bits to m output bits has a circuit representation! Thus, any computation can be represented by a circuit! Furthermore, all circuits can be composed with only NAND gates, making the NAND gate a so-called **universal gate**. Now that we have both a data representation and the ability to manipulate this data at will, we essentially have ourselves a computer! This was all made possible with the advent of electronic signals and the transistor.

Exercise 2.2.1 Draw a circuit for the NOT, AND, OR and XOR operations using only NAND gates. You are allowed to use extra fixed input bits, sometimes called ancilla bits.

2.3 Programs and Algorithms

As we saw above, an alternative to the circuit model of computation, there exists a procedural/programmatic model of computation. In this case, we represent our computation as a **program**, which is simply a list of instructions that should be done. These instructions can manipulate memory, which typically consists of variables whose bit values can change. Another key instruction is the jump instruction which allows one to move from one instruction of a program to another. Jumps allow for conditional operations where entire blocks of instructions can be ignored based on some conditions. It also allows for the essential concept of loops, where the same instructions can be repeated many times until an exit condition is satisfied. Programs are more intuitive than circuits, since they most align with how we make decisions ourselves. For instance, the concept of “if I see a fire, I better run away" is much better implemented in a program than trying to think of it in terms of NANDs, which we saw was universal. It turns out that any circuit can be modeled as a program and vice versa, so these two models of computation are perfectly equivalent[7].

An algorithm is a program that sets out to achieve a particular goal. For instance, to make pancakes, you may follow a recipe. We can call that recipe an algorithm to follow to make great pancakes each and every time. Algorithms may also be probabilistic, meaning that they don’t always achieve their goal, but have a high chance of doing so. This may or may not be very useful depending on how high these probabilities are or whether error can be tolerated![8] Other than completion, what other metrics should we look for in evaluating the performance of an algorithm?

In general, algorithms have two types of cost, space and time. In our cooking analogy, space corresponds to the amount of counter space you may have, while time is how long it takes to cook your dish. Some recipes/styles of cooking require lots of counter space to store all kinds of ingredients before they are then quickly sent to the wok to cook. Others may require many steps and time. In a computer, space is how much memory is required to run your algorithm, while time is exactly as it sounds, the amount of time required to run your algorithm to completion. Typically (though not always), there is a tradeoff between space and time. Algorithms may be sped up by larger use of memory to store useful quantities.

The *exact* space and time requirements of algorithm can depend on the exact computer or implementation of the algorithm. For instance, the absolute time required to perform operations on modern smartphones are much faster than the first computers of the 70s. Compiled programming languages like C++ tend to be faster than interpreted ones like Python, despite using the exact same set of instructions. The hardware (circuit) implementations of the same algorithm will be even faster! Thus, the exact space and time an algorithm uses or takes is not a property of the algorithm itself. Furthermore, these quantities may also depend on the exact inputs of the algorithm. For example, sorting a deck of cards may depend one exactly how scrambled the deck is to begin with.

To counter these issues, **complexity theory** was formulated. In particular, we will only concern ourselves with an *average* or *worst* case scenario when it comes to the input to the algorithm. The worst case provides an upper bound for the time the algorithm will take, and is thus important to know. We don’t want to rely on luck for our algorithm to work! A property of an algorithm that is independent of the actual computer/implementation is how the algorithm scales as the size of the problem grows. This is especially useful, because algorithms that scale better will remain relevant even as the problem size grows. Larger scales are a very natural concept, as growth always causes problems to get larger! This is all a bit abstract, and it is best to tackle a few examples of search algorithms, and learn about their scalings.

[7] The concept of a program can be made precise via a Turing machine.

[8] An algorithm that determines how a self-driving car should behave on the road when a pedestrian is detected should not be too probabilistic at all!

Example 2.3.1 Linear search. Consider a list of N names in alphabetical order (i.e. the list is sorted). We are looking for one particular name in the list. Now, consider the following algorithm:

1. Start with the first element of the list and check if it is the name we are looking for. If it is, then we are done. If not, proceed to the next name. Then, repeat.

Let's now think about the best and worst case runtime for this algorithm. The best case is that we are very lucky and find the name immediately! The best case has no scaling with N. In other words, no matter what N is, the best case only requires 1 iteration of the algorithm. The worst case is that we need to perform all N iterations before we find the name. The average case is that the name is somewhere in the middle and thus will take around $N/2$ iterations. Although there is a difference between N and $N/2$, these cases both scale linearly! This type of difference can also arise from the implementation or the computer. In notation, we use something called **big-O-notation**, and say both the average and worst case for the linear search is $\mathcal{O}(N)$. The best case is $\mathcal{O}(1)$, which means that it doesn't scale with N. In big-O-notation, we ignore all constant factors and focus on the scaling.

Example 2.3.2 Binary search. Consider a list of N names in alphabetical order (i.e. the list is sorted). We are looking for one particular name in the list. Now, consider the following algorithm:

1. Start in the middle of the list, and ask if the name I'm looking for is later or earlier in the alphabet.
2. If it is later in the alphabet, reduce the list to the $N/2$ later elements, and return to step 1. If it is earlier in the alphabet, reduce the list to the first $N/2$ elements, and return to step 1.

The best case for this algorithm is once again we are lucky and the name happens to be in the exact middle of the list! The worst case is that we need to perform the reduction over and over again, until we are reduced down to a single element. How many iterations is this? In other words, how many times do we need to divide N by 2, before it is equal to 1? This is given approximately by $\log_2(N)$. This is nifty, and in fact may be your preferred way of looking through the phonebook (if you still do that). Thus, the worst case is $\mathcal{O}(\log_2(N))$. Most of the most efficient algorithms use this sort of divide and conquer approach.

While polynomial scaling is manageable, there is a type of scaling that really becomes untenable. In fact, we've already seen this scaling before! Imagine an algorithm to list all the possible length N bitstrings. Although this algorithm may be easy to write out, the number of bitstrings possible is 2^N, so this algorithm (in all cases!) will take $\mathcal{O}(2^N)$ time. This is known as exponential scaling, and it grows faster than any polynomial scaling at large enough N. In Figure 2.4, we plot these various polynomials as a function of N. 2^N outpaces any polynomial eventually while $log_2(N)$ beats any polynomial.

Reducing exponential scaling to polynomial scaling is in fact how quantum computers may be useful. It has been shown by Peter Shor that

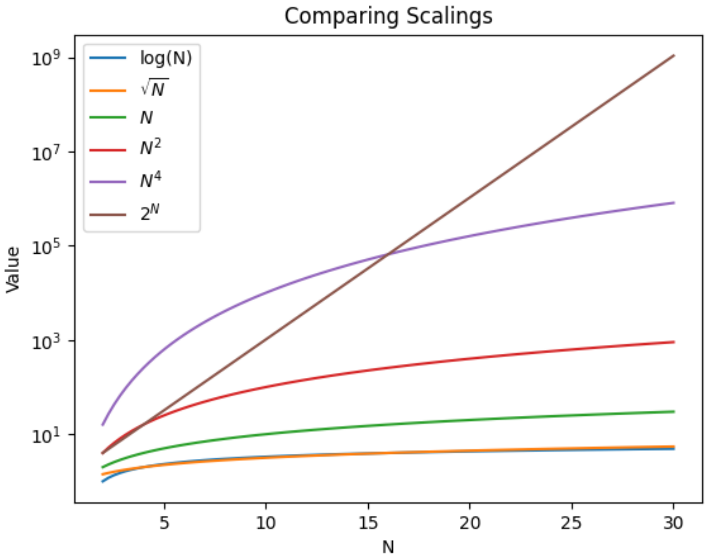

Figure 2.4: The growth of various polynomial scalings compared to the log and exponential scaling. Note the log scale on the y axis. All the polynomial curves "flatten" out on this axis, but the exponential scaling grows quickly. On the flip side, the log scaling flattens out more quickly than any polynomial.

factoring a large number with a quantum computer can be performed in polynomial time, while no *known* classical algorithm is any faster than exponential time. In fact, it is because factoring a large number can only be performed in exponential time classically that it is the basis for most encryption schemes to-date. Note that it is unknown *whether* a classical polynomial time algorithm exists for factoring, but one is known for quantum computers. Thus, quantum computers pose a serious challenge for classical encryption schemes. Luckily for us, there are quantum-proof encryption schemes available as long as one can use quantum resources. One of our goals in these notes is not only to understand how classical encryption is broken, but also how it can be mended with **quantum key distribution**.

Although breaking cryptography is not a fantastic application when it comes to financial security, possible applications are vast. The **HHL (Harrow-Hassidim-Lloyd) algorithm** speeds up the process of solving a system of linear equations by reducing a cubic scaling ($\mathcal{O}(N^3)$) to a log scaling ($\mathcal{O}(\log(N))$). Linear equations lie at the heart of many linear algebra calculations (which is at the heart of big data and machine learning) and are also prevalent in differential equation solving. There are also applications of quantum algorithms to quantum chemistry, which can have an impact on faster drug discovery.

2.4 Summary

Computing has gone a long way from the days of the abacus and mechanical computers. These computers are slow and inflexible. Central to all computing is some form of data representation and then manipulation. With the advent of electronics, the bit emerged as a robust form of data representation and could be easily implemented in electrical signals. Since the representation is electronic, manipulation can also be performed electronically. With the transistor, universal logical gates, which can perform any computation, can be implemented. Not all computations are equal though, and some are more efficient than others. We can quantify efficiency by asking how the operation scales with the size of the problem. While polynomial scalings are okay, exponential scalings become very cumbersome. It is in reducing exponential scalings to polynomial scalings where quantum computers show exceptional promise. These speedups can revolutionize fields from cryptography to machine learning.

With the context set, we are ready to begin understanding where these speedups in quantum computing come from. We will see that they arise fundamentally from a very different data representation and manipulation scheme. Understanding exactly how these features relate to computing is an area of active research, but plentiful examples and progress has already been made which will be the focus of our notes.

The Qubit 3

3.1 Qubit Representations

3.1.1 Algebraic Representation

In the last chapter, we saw that data representation forms the basis of any computing system. It defines how we store data and how we may manipulate it. To utilize the features of quantum mechanics, we use something called a **quantum bit** or a **qubit**. It is useful to retain the two-state nature of a bit[1], so a **qubit** is comprised of two components or **computational basis states** which we denote as $|0\rangle$ and $|1\rangle$. These are called **kets**, and suffice it to say that they are used to simply distinguish these states from the numbers 0 and 1. The sense that these states form a **basis** is that these are not the *only* states that are valid for the qubit. Here comes the strangeness and equally fascinating part of quantum mechanics. A generic qubit can actually be in both $|0\rangle$ and $|1\rangle$ at the same time! We call this a **superposition**. In fact, a general qubit state can be written as

$$|\psi\rangle = c_0\,|0\rangle + c_1\,|1\rangle \tag{3.1}$$

where c_0 and c_1 are **complex** numbers! The states $|0\rangle$ ($|1\rangle$) are just special cases of the general qubit when $c_0 = 1(0)$ and $c_1 = 0(1)$. How do we interpret c_0 and c_1? What do they mean?[2]

c_0 and c_1, to some extent, represent how much of the qubit is in the $|0\rangle$or $|1\rangle$ state. This leads us to an important **postulate** of quantum mechanics. Upon **measurement**[3] of the qubit in the computational basis, we simply arrive at the information that the qubit is in $|0\rangle$ or the $|1\rangle$ state. We do not find out what c_0 or c_1 are in a single measurement. In fact, it is not at all obvious how a complex number is measured in the first place. How a state is measured is slightly less confusing. We can determine properties of these states and test them as a way to distinguish them. Furthermore, after **measurement**, the qubit is said to have **collapsed** into that state. After we determine (with measurement) that the qubit is in state $|0\rangle$, it does not magically return to $c_0\,|0\rangle + c_1\,|1\rangle$. It stays in $|0\rangle$ for future manipulation. The collapse of the qubit means that measuring the system has fundamentally *disturbed* the system. We will see later that this feature is exactly what quantum communication schemes rely on to avoid any eavesdroppers. Where c_0 and c_1 come in is that $|c_0|^2$ and $|c_1|^2$ represent the *probabilities* of measuring $|0\rangle$ and $|1\rangle$ to be the state that the qubit is in. Note that these need to squared because probabilities have to be real and non-negative! Let's do a example.

Example 3.1.1 Consider a qubit in the state

[1] Recall that a huge innovation in computing was to use a digital representation of two discrete states instead of an analog representation

[2] Here, it is important to emphasize that the extent to which any of our following interpretation is *real* is not really the right question. Remember that quantum mechanics is a *mathematical* and predictive framework which has never been contradicted by experiment. How one chooses to think about it is up to them. As long as the predictions are correct, I have no complaints. I simply offer up how I find intuition in these strange concepts.

[3] What constitutes a measurement is an active area of research and debate. Its mathematical definition is more or less agreed upon though.

$$|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{i}{2}|1\rangle \tag{3.2}$$

a) What is the probability of measuring the qubit in the state $|0\rangle$? What state is the qubit in after measurement?
b) What is the probability of measuring the qubit in the state $|1\rangle$? What state is the qubit in after measurement?

For this qubit state, we identify $c_0 = \sqrt{3}/2$ and $c_1 = i/2$. For part a), to calculate the probability of measuring the qubit in the state $|0\rangle$, we need to calculate $|c_0|^2$. In this case, c_0 is real, so we simply need to square c_0 to obtain 3/4 as the probability. For the probability of measuring state $|1\rangle$, we need to calculate $|c_1|^2$. Recall that $|z|^2 = z^*z = zz^*$ for any complex number z. The $*$ operation is complex conjugation, which converts $z = a + bi$ to $z^* = a - bi$, where every i gets replaced by $-i$. Thus the calculation for $|c_1|^2$ proceeds as

$$|c_1|^2 = \frac{i}{2} \times \frac{-i}{2} = \frac{1}{4} \tag{3.3}$$

Thus, the probability of measuring $|1\rangle$ is 1/4, which is unsurprising, since the probability of measuring $|0\rangle$ is 3/4. Upon measurement of either of these, the qubit will collapse into the $|0\rangle$ and $|1\rangle$ state.

You might have noticed that since probabilities need to add up to 1, c_0 and c_1 are not completely unconstrained! In fact

$$|c_0|^2 + |c_1|^2 = 1 \tag{3.4}$$

and this is the **normalization** constraint.

Although the probabilities are what we tend to engage with when it comes to these quantum states, their complex nature is also important. Namely, you might have realized that there can be multiple states with the same probability distribution! Consider the following states, all with equal probabilities in the $|0\rangle$ and $|1\rangle$ state.

$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{3.5}$$

$$|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \tag{3.6}$$

$$|+\rangle_y = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle \tag{3.7}$$

$$|-\rangle_y = \frac{1}{\sqrt{2}}|0\rangle - \frac{i}{\sqrt{2}}|1\rangle \tag{3.8}$$

We will see that there is a method to the madness with the naming convention in due time. We would be unable to distinguish between these states with measurements in the computational basis. However, we will see that we can distinguish between these states by measuring in different **bases**, so these are perfectly valid distinguishable states! Hold tight on what this even means or how to do this!

It is useful now to introduce another very useful mathematical tool, which will often simplify notation and provide further intuition. Paired with each quantum state is its **dual**[4], which is a linear function that we denote as $\langle\psi|$, and is known as a **bra**[5]. Before we discuss how to calculate the dual of a generic quantum state, to gain some intuition, let's consider the duals of the basis states $\langle 0|$ and $\langle 1|$. What do they do? Why are they functions? Let's define $\langle 0|$ as the object with the following properties:

$$\langle 0|0\rangle = 1 \tag{3.9}$$

$$\langle 0|1\rangle = 0 \tag{3.10}$$

$$\langle 0|\left(a|\psi\rangle + b|\phi\rangle\right) = a\langle 0|\psi\rangle + b\langle 0|\phi\rangle \tag{3.11}$$

The first two equations express the notion that these bra objects are functions! We can think of them as a hungry pacman ready to eat states and return a number! In slightly less flashy terms, we can say that $\langle 0|$ is a function that acts on a ket, and returns a number. The number that it returns depends on whether it acts on state $|0\rangle$ or $|1\rangle$. The dual of $|0\rangle$, which is $\langle 0|$, will return 1 when acting on state $|0\rangle$. The last equation indicates that this bra object is linear. It chomps through addition, subtraction and scalar multiplication! The dual of $|1\rangle$ acts similarly as the dual of $|0\rangle$ in that $\langle 1|0\rangle = 0$ and $\langle 1|1\rangle = 1$. Let's see how this notation can be used to express the probabilities we were discussing above.

Example 3.1.2 Let's express $|c_0|^2$ and $|c_1|^2$, the probabilities of measuring $|0\rangle$ or $|1\rangle$ for an arbitrary qubit $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$. Consider

$$\langle 0|\psi\rangle = c_0\langle 0|0\rangle + c_1\langle 0|1\rangle = c_0 \tag{3.12}$$

We can think of $\langle 0|$ as sort of an extractor for the coefficient c_0. Thus, the probability of measuring $|0\rangle$ can be written as $|\langle 0|\psi\rangle|^2$. One can show that the expression for $|c_1|^2$ is $|\langle 1|\psi\rangle|^2$.

The bra can be used to define an **inner product**, which you may have encountered as a dot product. We can think of $\langle\psi|$ as an extractor for how much a particular state is of the character $|\psi\rangle$[6]. If it is exactly zero, the states are said to be **orthogonal** or have no overlap. The states $|0\rangle$ and $|1\rangle$ are in fact **orthonormal**, meaning that they are not only orthogonal, but also normalized. This is required for two states to form a **basis**[7]. This is summarized in the equation

$$\langle i|j\rangle = \delta_{ij} \tag{3.13}$$

where δ_{ij} is the Kronecker delta with the property that

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{3.14}$$

To enforce normalization for an arbitrary quantum state $\langle\psi|\psi\rangle = 1$, we define the dual of an arbitrary state to be

[4] You may have encountered this in a higher level linear algebra class. It is the same notion of dual.

[5] So that $\langle\psi|\psi\rangle$ can be called a braket.

[6] We will see in the vector representation, this represents the angle between two vectors

[7] We can think of basis here as being capable or representing an arbitrary qubit

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle \longrightarrow \langle\psi| = c_0^* \langle 0| + c_1^* \langle 1| \tag{3.15}$$

where $*$ is once again the complex conjugation operation. Let's now confirm $\langle\psi|\psi\rangle = 1$. We obtain:

$$\begin{aligned}\langle\psi|\psi\rangle &= (c_0^* \langle 0| + c_1^* \langle 1|)(c_0 |0\rangle + c_1 |1\rangle) \\ &= |c_0|^2 \langle 0|0\rangle + c_0^* c_1 \langle 0|1\rangle + c_1^* c_0 \langle 1|0\rangle + |c_1|^2 \langle 1|1\rangle \\ &= |c_0|^2 + |c_1|^2 = 1 \end{aligned} \tag{3.16}$$

where in the last line, we enforced the normalization constraint. Let's practice with some of these manipulations.

Example 3.1.3 Alternate bases. Show that the $|+\rangle$ and $|-\rangle$ states are orthonormal. Recall that these states written in the computational basis are given in equations

$$\begin{aligned}|+\rangle &= \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \\ |-\rangle &= \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle\end{aligned}$$

For normalization, let's now evaluate $\langle +|+\rangle$ and confirm it is 1. From our general calculation above, it should come as no surprise.

$$\begin{aligned}\langle +|+\rangle &= \left(\frac{1}{\sqrt{2}} \langle 0| + \frac{1}{\sqrt{2}} \langle 1|\right)\left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle\right) \\ &= \frac{1}{2} \langle 0|0\rangle + \frac{1}{2} \langle 1|1\rangle = 1\end{aligned} \tag{3.17}$$

It is left to the reader to confirm $\langle -|-\rangle = 1$. Now, let's calculate $\langle +|-\rangle$.

$$\begin{aligned}\langle +|-\rangle &= \left(\frac{1}{\sqrt{2}} \langle 0| + \frac{1}{\sqrt{2}} \langle 1|\right)\left(\frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle\right) \\ &= \frac{1}{2} \langle 0|0\rangle - \frac{1}{2} \langle 1|1\rangle = 0\end{aligned} \tag{3.18}$$

These two states are orthogonal! Note that $\langle -|+\rangle = 0$, and in fact $\langle +|-\rangle$ is related to $\langle -|+\rangle$ as we will see in exercise 3.1.2.

Exercise 3.1.1 Another basis. Show that the $|+\rangle_y$ and $|-\rangle_y$ states are orthonormal. They are defined in equations 3.7 and 3.8.

Exercise 3.1.2 Property of the inner product. Show that for two generic qubit states $|\psi\rangle$ and $|\phi\rangle$, $\langle\psi|\phi\rangle = \langle\phi|\psi\rangle^*$.

We will have more to see about different bases later in the chapter, but for now, let's just acknowledge their existence. It is worth mentioning now that in addition to normalization, there is one additional feature of the state that prevents them from being completely arbitrary. States are equivalent up to an arbitrary global phase[8]. Namely the states $|\psi\rangle$ and $e^{i\varphi}|\psi\rangle$ are considered equivalent. We can already see that this phase factor φ does not impact any probabilities

$$|\langle 0|e^{i\varphi}|\psi\rangle|^2 = |e^{i\varphi}c_0|^2 = e^{-i\varphi}c_0^* e^{i\varphi}c_0 = |c_0|^2 \tag{3.19}$$

While the **relative phase** between the $|0\rangle$ and $|1\rangle$ coefficients in a state does have measurable consequences, a global phase does not! Note that an overall negative sign (-1) in a state is a special case of the arbitrary global phase when $\varphi = \pi$. We will have more to say about these phases soon, and also a more intuitive graphical representation of a state. Suffice it to say for now that relative phases can be measured and global phases cannot. Thus, to detect physical effects, relative phase shifts (or amplitude change) between two coefficients must occur. A global phase shift to both coefficients is undetectable.

3.1.2 Vector Representation

Another useful way to represent a quantum state is using a **vector**. This representation makes concepts of orthonormality slightly more intuitive. It is also a powerful computational tool, and is how the simplest quantum calculations on a computer represent the state. The mapping is straight forward

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \rightarrow \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \tag{3.20}$$

where we represent states as a **column** vector, using the two complex coefficients. Note that these vectors do not live on a simple xy plane, since c_0 and c_1 are complex! They are actually in some 4 dimensional space with the real and imaginary axes of both c_0 and c_1. The computational basis states can be expressed as:

$$|0\rangle \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{3.21}$$

$$|1\rangle \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{3.22}$$

The notion of orthogonality is slightly more obvious here. The dot product between the $|0\rangle$ and $|1\rangle$ state is clearly 0. They are also obviously orthogonal if we think of them as vectors in the xy plane (which we can do here only because both coefficients are real!), since $|0\rangle$ is along the x axis and $|1\rangle$ is along the y axis. The normalization constraint requires these states to be unit vectors.

The dual also has a nice vector representation. A linear function acting on column vectors and returning a number can be represented by a row vector. Thus, we obtain the mapping:

[8] Phase is used here in the complex number sense when represented in polar notation. Recall that $e^{i\varphi} = \cos\varphi + i\sin\varphi$

$$\langle\psi| = c_0^* \langle 0| + c_1^* \langle 1| \rightarrow \begin{pmatrix} c_0^* & c_1^* \end{pmatrix} \tag{3.23}$$

Let's do an example with vector notation.

Example 3.1.4 Express $|+\rangle_y$ and $|-\rangle_y$ in vector notation. Show that these two states are orthonormal using a vector calculation. Using the definitions of these two states from equation 3.7 and 3.8, we obtain the vector representations:

$$|+\rangle_y \rightarrow \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix}, |-\rangle_y \rightarrow \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -i \end{pmatrix} \tag{3.24}$$

Their corresponding duals are:

$$\langle +|_y \rightarrow \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -i \end{pmatrix}, \langle -|_y \rightarrow \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \end{pmatrix} \tag{3.25}$$

Do not forget the complex conjugation in the dual! Let's now calculate $\langle +|_y|+\rangle_y$[9]. We obtain

$$\langle +|_y|+\rangle_y = \frac{1}{2}\begin{pmatrix} 1 & -i \end{pmatrix}\begin{pmatrix} 1 \\ i \end{pmatrix} = \frac{1}{2}(1+1) = 1 \tag{3.26}$$

where we have applied the standard matrix multiplication rules in linear algebra. We leave the calculation of $\langle -|_y|-\rangle_y$ to the reader. Now, we calculate $\langle +|_y|-\rangle_y$ and obtain

$$\langle +|_y|-\rangle_y = \frac{1}{2}\begin{pmatrix} 1 & -i \end{pmatrix}\begin{pmatrix} 1 \\ -i \end{pmatrix} = \frac{1}{2}(1-1) = 0 \tag{3.27}$$

Thus, we find that these two states are orthonormal!

It is good to have the vector representation in your toolbox for computations. We note that the language of linear algebra is very natural for quantum computing, because the fundamental time-evolution equation of quantum mechanics, known as the **Schrödinger equation** is linear. We will have more on that later! Now, we move to a final graphical representation which can be quite useful for thinking about a single qubit, but is nearly impossible to generalize to larger numbers of qubits or systems with more than two states, such as **qutrits** (3 states) or **qudits** (arbitrary n-state system).

3.1.3 Bloch Sphere

Recall that a generic qubit can be written in the form

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle . \tag{3.28}$$

which, since c_0 and c_1 are complex, appears to require 4 independent variables to describe fully. These 4 are the real and imaginary parts of

[9] This notation is a bit cumbersome. I know.

c_0 and c_1. However, these 4 variables can actually be reduced to 2. We eliminate one variable from the normalization requirement

$$|c_0|^2 + |c_1|^2 = 1 \tag{3.29}$$

One way to parameterize our state and ensure normalization is to write our state as

$$|\psi\rangle = \cos(\theta/2)e^{i\varphi_0}|0\rangle + \sin(\theta/2)e^{i\varphi_1}|1\rangle . \tag{3.30}$$

Another is eliminated from the arbitrariness of the global phase, namely that $e^{i\varphi}|\psi\rangle$ is the same as $|\psi\rangle$. Thus, we can always presume c_0 as real and choose $\varphi_0 = 0$, and keep just a relative phase on state $|1\rangle$. Thus, we can parameterize our qubit in the following way

$$|\psi\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\varphi}|1\rangle . \tag{3.31}$$

Exercise 3.1.3 Verify equation 3.31 is normalized.

The θ and φ variables which now parameterize our qubit can be visualized in a nice way. They are simply the spherical coordinates that parameterize the unit sphere (which is a 2 dimensional surface). This unit sphere which hosts all our states is called the **Bloch sphere** (Figure ??) and is critical for visualizing single qubit states and how they might transform in gate operations.

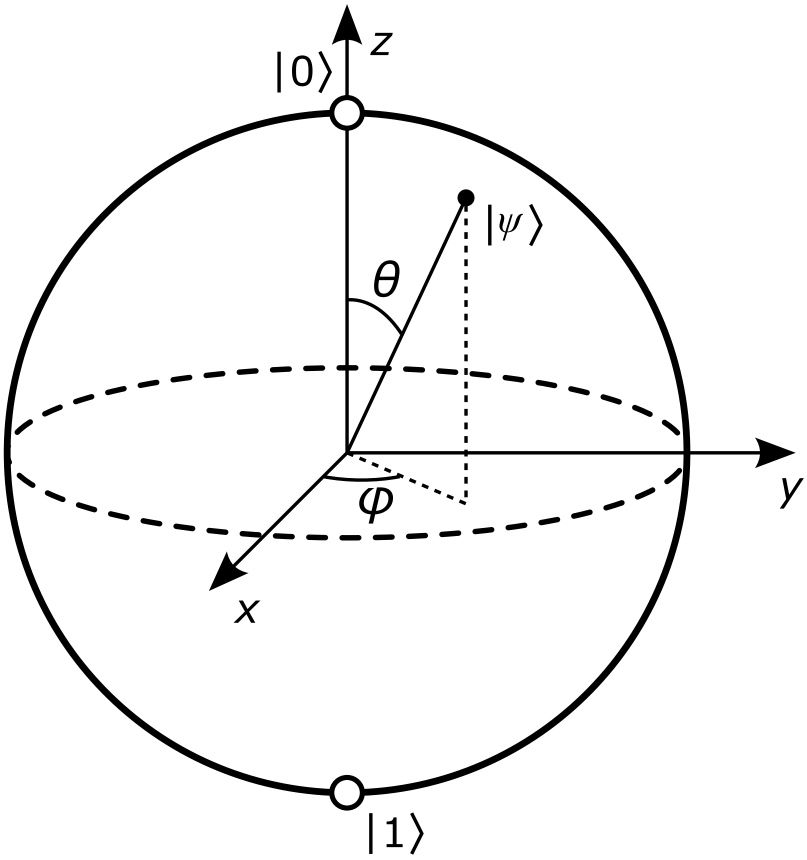

Figure 3.1: The Bloch sphere, which represents a single qubit. The North pole represents the $|0\rangle$ state and the South pole represents the $|1\rangle$ state. Along the equator are all the different states with equal probability in both $|0\rangle$ and $|1\rangle$. Latitude describes how much the state consists of $|0\rangle$ or $|1\rangle$, while the longitude describes how the relative phase between the $|0\rangle$ and $|1\rangle$ state.

States are represented by vectors or points (if you prefer) on the surface of the sphere. There is a one-to-one mapping of a particular state to a point or vector on the sphere which is described by a polar angle $0 \leq \theta \leq \pi$ and an azimuthal angle $0 \leq \varphi < 2\pi$. Let's make some observations about the Bloch sphere. You should feel comfortable showing all of these statements mathematically.

1. The north pole ($\theta = 0$) and south pole ($\theta = \pi$) represent the $|0\rangle$ and $|1\rangle$ state.
2. The northern hemisphere ($\theta < \pi/2$) and southern hemisphere ($\theta > \pi/2$) represent states with higher probability of being in $|0\rangle$ and $|1\rangle$ respectively.
3. The equator ($\theta = \pi/2$) represents all states with an equal probability of being in $|0\rangle$ and $|1\rangle$. These states differ in the relative phase between the amplitudes (or coefficients) of each state.
4. States along the x axis ($\varphi = 0, \pi$) have only real coefficients.
5. States along the y axis ($\varphi = \pi/2, 3\pi/2$) have a $|1\rangle$ amplitude which is purely imaginary.

Example 3.1.5 Show statement 2 above.

We start with our general equation for a qubit on the Bloch sphere in terms of the polar angle θ and azimuthal angle φ.

$$|\psi\rangle = \cos(\theta/2)\,|0\rangle + \sin(\theta/2)e^{i\varphi}\,|1\rangle \tag{3.32}$$

In the northern hemisphere, $\theta < \pi/2$, so $\theta/2 < \pi/4$. For these angles, $\cos(\theta/2) > \sin(\theta/2)$, so the probabilities of finding the qubit in state $|0\rangle$ (which is $\cos^2(\theta/2)$) is higher than finding the qubit in state $|1\rangle$ (which is $\sin^2(\theta/2)$). A similar line of reasoning follows for the southern hemisphere.

Exercise 3.1.4 Write down the states pointing along the $\pm x$ direction and the $\pm y$ direction on the Bloch sphere. Are they related to the equal superposition states we introduced in equations 3.5-3.8?

Exercise 3.1.5 The Bloch sphere contains 8 octants intersecting at the origin. Write down a valid state in each octant.

One oddity to keep in mind is what orthogonality means on the Bloch sphere. Orthogonality in typical Euclidean space is thought of as vectors that make right angles with each other. However, on the Bloch sphere, orthogonality manifests as oppositely pointed vectors. For instance, $|0\rangle$ and $|1\rangle$ point along the $+z$ and $-z$ axes, which are diametrically opposite. Try exercise 3.1.6 to show this fact in general.

Exercise 3.1.6 Show that diametrically opposite vectors on the Bloch sphere are orthogonal.

We will see that the Bloch sphere is convenient to think about operations that can happen to qubits. For instance, we can immediately see that any operation that changes one state to another will transform one vector to another on the Bloch sphere, lending it a geometric interpretation. Now, we are ready to start tackling quantum operations on a qubit.

3.2 Single Qubit Gates

Now that we have established our data representation, to compute we need to know how to manipulate this data. We will see, in due time, important constraints on what these operations can be, but for now, we will simply work some examples to get us started. The equivalent of the NOT gate in classical computing translates to quantum computing (which we will call[10] X for reasons we will see later) and can be described as the following operation:

$$X|0\rangle = |1\rangle \tag{3.33}$$
$$X|1\rangle = |0\rangle \tag{3.34}$$

This operation, also called a **X-gate**, essentially performs a NOT operation which flips 0 to 1 and 1 to 0. What does this gate do to an arbitrary qubit which can be in a superposition? Quantum mechanics happens[11] to be a linear theory, so knowing what X does to $|0\rangle$ and $|1\rangle$ actually allows us to know what happens to any state. In particular $X(a|\psi\rangle + b|\phi\rangle) = aX|\psi\rangle + bX|\phi\rangle$. This is the definition of linearity of the **operator**[12] X.

It is because the gate acts linearly that vectors and matrices are a very natural representation for quantum computing. In particular, we have represented states as vectors, but gates will be represented as matrices. The matrix for the X gate is given by

$$X \rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{3.35}$$

The simple way to determine the matrix corresponding to a gate is to realize that for a generic matrix

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{3.36}$$

the first column can be determined by acting the matrix on $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$. Namely, we notice that

$$M\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ c \end{pmatrix}. \tag{3.37}$$

To obtain the second column, we act the matrix on $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and note that

$$M\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} b \\ d \end{pmatrix}. \tag{3.38}$$

[10] Note that we are using a capital letter, which is a common convention for operations on quantum states!

[11] There is no proof for this! It is simply a fact of nature.

[12] We're throwing around lots of interchangeable terms here. Operator and gate are equivalent in meaning here

Let's get some practice with this gate in the next very important example, where we'll see the significance of the $|+\rangle$ and $|-\rangle$ states.

Example 3.2.1 Determine how the X gate acts on the states

$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{3.39}$$

$$|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \tag{3.40}$$

both algebraically and with matrices.

First, we do an algebraic manipulation. Utilizing linearity, we obtain:

$$X|+\rangle = \frac{1}{\sqrt{2}}X|0\rangle + \frac{1}{\sqrt{2}}X|1\rangle = \frac{1}{\sqrt{2}}|1\rangle + \frac{1}{\sqrt{2}}|0\rangle = |+\rangle \tag{3.41}$$

$$X|-\rangle = \frac{1}{\sqrt{2}}X|0\rangle - \frac{1}{\sqrt{2}}X|1\rangle = \frac{1}{\sqrt{2}}|1\rangle - \frac{1}{\sqrt{2}}|0\rangle = -|-\rangle \tag{3.42}$$

Let's now verify this with matrices. The calculations proceed as

$$X|+\rangle \rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rightarrow |+\rangle \tag{3.43}$$

$$X|-\rangle \rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \rightarrow -|-\rangle \tag{3.44}$$

In other words, the X gate does **not** change the composition of the $|+\rangle$ or $|-\rangle$ states, but only adds a phase of π to $|-\rangle$. This phase of π results in a factor of -1 and is why we have been calling the $|-\rangle$ state in such a way! The states $|+\rangle$ and $|-\rangle$ are said to be eigenvectors[13] of the gate X with eigenvalues 1 and -1. It turns out that every gate on a single qubit will have a pair of eigenstates or eigenvectors that are left unchanged (except by a -1 sign) that are *also* orthogonal. This is a strong and provable statement, but we will not do so here! Thus, every gate via its eigenvectors also defines a **basis** that we can use to describe any quantum state. We will have more to say about this later.

The X gate has sort of a classical interpretation. It is the quantum version of a NOT gate which flips a bit. However, a since quantum bits carry phase, we can also have a gate Z that does the following

$$Z|0\rangle = |0\rangle \tag{3.45}$$

$$Z|1\rangle = -|1\rangle \tag{3.46}$$

which has **no** classical analog! It imposes a π phase shift on $|1\rangle$, and is why it is also called the **phase** gate. However, I thought that global phases do not matter? Thus, how is something like equation 3.46 useful?

[13] Formally, an eigenvector of a matrix M is a vector $\mathbf{v}$ such that $M\mathbf{v} = \lambda\mathbf{v}$ for a scalar λ.

Well let's see what happens when we act Z on $|+\rangle$ and $|-\rangle$.

$$Z|+\rangle = \frac{1}{\sqrt{2}} Z|0\rangle + \frac{1}{\sqrt{2}} Z|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = |-\rangle \tag{3.47}$$

$$Z|-\rangle = \frac{1}{\sqrt{2}} Z|0\rangle - \frac{1}{\sqrt{2}} Z|1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = |+\rangle \tag{3.48}$$

The Z gate actually flips $|+\rangle$ to $|-\rangle$ and $|-\rangle$ to $|+\rangle$! Whereas the X gate flips between $|0\rangle$ and $|1\rangle$, our computational basis choice, the Z gate flips between the alternate X basis states $|+\rangle$ and $|-\rangle$. This is a good thing to remember about the Z gate! It is a NOT gate, but in a different basis, which is a purely quantum mechanical possibility. The matrix which represents the Z gate is given by

$$Z \to \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{3.49}$$

Exercise 3.2.1 The Y gate. Consider the gate obtained by subsequently applying a Z gate followed by the X gate and then multiplying by i, algebraically represented by $Y = iXZ$.

a. Determine the action of Y on $|0\rangle$ and $|1\rangle$, and use it to write the matrix representation of Y.
b. Determine how Y acts on $|+\rangle_y$ and $|-\rangle_y$ and justify their naming scheme.