An Undergraduate Course on Quantum Computing

(Second Edition)

Peter Young

October 4, 2021

Contents

Preface

This material has been given as a one-quarter course for undergraduates in the physical sciences at the University of California Santa Cruz. When supplemented by homework problems it could provide the basis for an undergraduate course at other institutions as well.

In order that the course be accessible to majors other than physics, the rules of quantum mechanics were taught from scratch in the first part. While some of my physics colleagues were surprised that this could be done, it is perfectly feasible because much of what is included in a traditional physics course on quantum mechanics concerns continuous degrees of freedom so one has to cover complicated topics such as partial differential equations, boundary conditions, angular momentum, and a plethora of special functions. All this can be omitted in a quantum computing course which is focused on 2-state systems. A solid background in linear algebra *is* required. A brief review of this is given at the start, but the treatment is fast and it is assumed that the students will have seen the material before.

The aim of the course is to get students to the level where they can understand the two most important topics covered: Shor's algorithm in Chapter 18 and quantum error correction in Chapter 19. Unlike quantum algorithms proposed previously, Shor's algorithm for factoring integers gives a spectacular speedup on a problem of practical importance (encryption of data sent down a public channel). Considerable experimental challenges remain to implement Shor's algorithm for a large number of qubits but quantum error correction will be essential in order to achieve this, because qubits are highly susceptible to noise. Incorporating quantum error correction still leaves huge experimental challenges before achieving the goal of factoring integers larger than what is possible classically, but without quantum error correction it would clearly be impossible.

The goal, then, is to present a course at the undergraduate level, but which still goes into enough depth to give a good understanding of Shor's algorithm and the basics of quantum error correction. No details will be given on the many experimental approaches to building a quantum computer, which is a huge topic that would merit a separate course in its own right.

There are, of course, excellent more advanced texts, such as the monumental classic by Nielsen and Chuang [NC00] and the books by Mermin [Mer07] and Rieffel and Pollack [RP14]. The book closest in level and spirit to the present text is the one by Vathsam [Vat16], which I found very useful when preparing this material. My hope is that this these lectures will take students to a level where they can follow the rapidly-moving advanced literature in the field.

Peter Young
University of California, Santa Cruz
October 4, 2021

Chapter 1

The Strange World of Quantum Mechanics

1.1 Introduction

The quantum world is strange, and different from the classical world that we see around us. Our intuition obtained from everyday experience is for objects that we can see. It does not apply to the quantum world where we are dealing with very small objects, objects that (in most cases) are too small to see. I give two quotations, from eminent physicists, which illustrate the strangeness of the quantum world:

> "Anyone who is not shocked by quantum mechanics hasn't understood it."
> (Attributed to Niels Bohr).

> "I think I can say that nobody understands quantum mechanics."
> (Richard Feynman).

The big question which we will address in this course is whether we can use the difference between the quantum and classical worlds to find more efficient algorithms to solve certain problems by treating the data in a quantum computer in which it is processed according to quantum rules rather than classical rules. We shall see that for some problems the answer is "yes". I should mention now that there is a *practical* question of whether we can actually build a useful quantum computer. The difficulties of building such a device have not yet been overcome, though much progress has been made. In this course, which focuses on theory, we will not describe the many experimental approaches that are being implemented to try to achieve this goal. However, we will discuss in Chapter 19 how one can reduce errors caused by an imperfect device, a topic called "Quantum Error Correction".

A quantum computer, then, is one in which data is processed by quantum, rather than classical rules. What do we mean by this? In a classical computer the data is stored in bits, which take two values 0 and 1. A quantum computer also uses 2-state systems called qubits. We indicate these two states by $|0\rangle$ and $|1\rangle$, a notation introduced by the physicist Paul Dirac. The difference from classical bits is that the general state of a qubit, which we will write as $|\psi\rangle$, is a *superposition* of states $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1.1}$$

where α and β are numbers (complex in general). For reasons that will be explained later, we need the condition $|\alpha|^2 + |\beta|^2 = 1$. One sometimes says loosely that a qubit in the state described by Eq. (1.1) is simultaneously in states $|0\rangle$ *and* $|1\rangle$. This is to be contrasted with a classical bit which takes value 0 *or* 1.

Our main goal in this course will be to see if one can gain computationally from superposition states.

In the next two sections of this chapter I describe experiments which illustrate the strangeness of the quantum world. More information on this topic can be found in Refs. [NC00, Mer07, Vat16] and in Ch. 1, Vol. 3 of the Feynman Lectures on Physics [FLS64].

1.2 The Two-Slit Experiment

You are probably familiar with experiments involving light going through slits which demonstrate that light, being a wave, shows interference.

First consider just one slit. If the slit width d is very large compared with the wavelength of light λ (the geometrical optics limit) then, to a good approximation, the light continues in a straight line. However, if the slit width is comparable to, or less than, λ, the light spreads out after passing through the slit, which is called diffraction. Figure 1.1 sketches the intensity of light observed on a screen behind the slit.

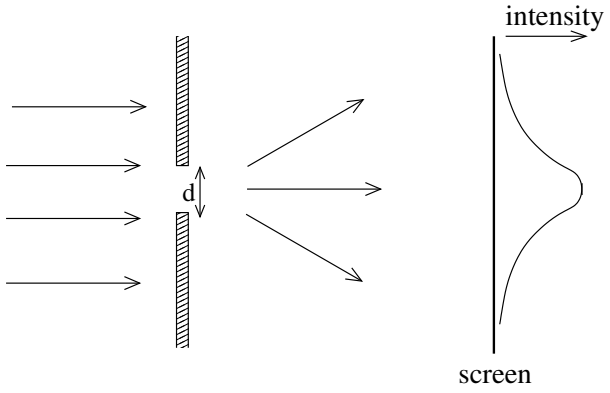


Figure 1.1: A beam of light spreads out (diffracts) when passing though a slit of width d which is comparable to, or smaller than, the wavelength of the light λ. The figure shows a sketch of the intensity of the beam on a screen after it has passed through the slit.

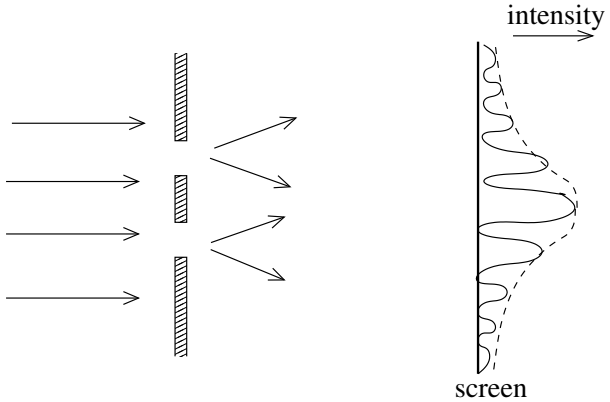


Figure 1.2: A two slit experiment. Interference fringes, oscillations of strong and weak intensity, are seen due destructive and constructive interference. The overall envelope of the intensity has a similar form to that from a single slit shown in Fig. 1.1.

If the light beam passes two slits, as shown in Fig. 1.2 one observes interference fringes, oscillations of strong and weak intensity, due to interference between between the beams going through the two slits. If the difference in path length $|r_1 - r_2|$ (see Fig. 1.3) satisfies $|r_1 - r_2| = n\lambda$ (for integer n) one has constructive interference and a maximum intensity, whereas if $|r_1 - r_2| = (n + \frac{1}{2})\lambda$ one has a minimum intensity. Hence, as one moves along the screen one alternately gets regions of low intensity and high intensity. These are called interference fringes.

This is the classical picture. That is, we shine a beam at the two slits, some of it goes through one slit, some goes through the other slit, and when these two beams recombine they interfere.

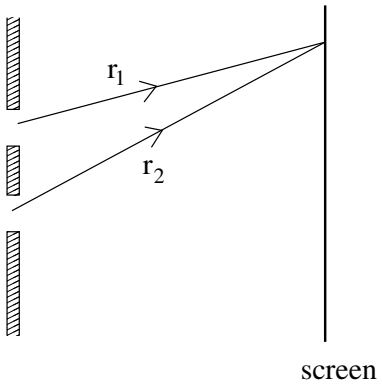


Figure 1.3: The difference in the length of the paths taken by the beams going through the two slits is $|r_1 - r_2|$. This varies as a function of the location on the screen, so the interference changes from constructive, where $|r_1 - r_2| = n\lambda$, to destructive, where $|r_1 - r_2| = (n + \frac{1}{2})\lambda$ with n an integer.

Now we reduce the intensity of the light. At some point we notice that light is not a continuous wave but consists of discrete bunches of energy called *photons*. To detect individual photons, we place an array of photon counters on the screen and count the number of discrete clicks in each counter, see Fig. 1.4. We record the number of clicks for counters placed at different points on the screen.

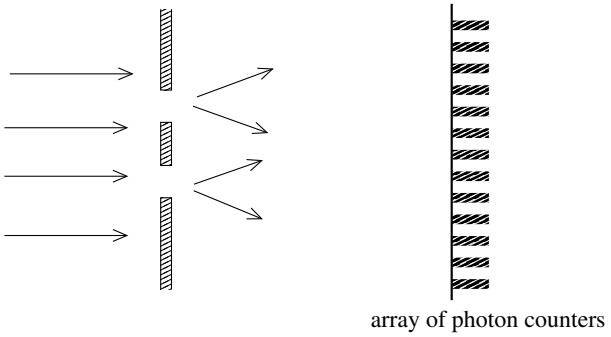


Figure 1.4: The two slit experiment where individual photons are detected by an array of counters which count the number of photons at their location.

Suppose we reduce the intensity so much that the time between emitting photons is greater than the time it takes a photon to pass through the experimental setup, i.e. the photons go through *one at a time*. Do we see an interference pattern? Using our classical intuition we would say "no" because surely each photon "must" either go through the upper slit or the lower one and can therefore not interfere with itself. In other words we would expect the intensity of clicks in the counters to vary smoothly along the screen, as in the classical single slit experiment shown in Fig. 1.1.

Amazingly, this is not so and *we do see an interference pattern*. In other words the number of clicks in the counters varies rapidly and in a oscillatory manner as we move along the screen, just as in the classical two-slit experiment shown in Fig. 1.2. It looks as though a single photon *does* go through both slits. You may already be feeling (correctly) that this looks suspiciously like a superposition state such as the one we we wrote down in Eq. (1.1), where now $|0\rangle$ refers to photon through the upper slit and $|1\rangle$ to photon through the lower slit.

You might ask "why don't we just *look* and see which slit the photon went through". Well, photons being electrically neutral are hard to observe unless we absorb them (which we want to do only when they reach the screen). The rate of scattering of one photon by another is immeasurably small. So, with photons we can't observe which slit they went though. However, we can do the same experiment with electrons rather than photons. Like photons, electrons have both particle and wave-like properties, but, being charged, they readily scatter light so we can see observe them by shining light on them. The discussion which follows is based on Ch. 1, Vol. 3 of Feynman [FLS64].

In this new version of the experiment we send electrons through the slits one at a time. To see which slit they went through we shine light of wavelength λ at the slits and observe a flash of light every time an electron goes through.

Suppose that we choose a light source that has a wavelength λ which is bigger than the slit spacing d. We do see a flash every time an electron passes through, and observe that there is *still* an interference pattern but, the flash of light is of size λ which is greater than the separation of the slits, so we can't tell which slit the electron went through. Clearly we need to use a light source with wavelength less than d. When we do this, indeed we see a flash at either the upper slit or the lower slit every time an electrons passes, so we've achieved our goal of observing which slit each electron goes through. But alas, when we look at the counts registered on the detectors we see that the interference fringes have been washed out, and we have just a smooth variation in the number of clicks along the screen. Observations such as these show that it is not possible to determine which slit each electron goes through *and* observe interference fringes.

This observation guides us to a second piece of intuition regarding quantum mechanics (the first, mentioned above, is that a quantum system can be in a superposition state), namely that a measurement can unavoidably change a quantum state, and in particular can destroy a superposition.

Classically, measurements are passive, and can be done in a delicate way so they simply reveal a reality which is already present whether we observe it or not. Quantum mechanically, measurements play a much more active role and can change the state of the system. In particular, we shall see that if we observe a system in a particular state, we can't necessarily say that it was in that state before the measurement.

1.3 Stern-Gerlach Experiment

We will now discuss a second experiment which gives additional insight into superposition states.

Consider the hydrogen atom, which consists of one proton (the nucleus), which has a positive electric charge, and one electron which has a negative charge. In its ground state the electron has a symmetric distribution of velocities and so there is no net circulating electric current around the proton. Hence the orbital motion of the electron does not give rise to a magnetic moment which could interact with an external magnetic field. However, the electron has an internal state, called *spin*, which does give rise to a magnetic moment[1] $\vec{\mu}$, proportional to the spin angular momentum.

There is a force on a magnetic moment in a field if the field is non-uniform. To see this, recall that the energy of a magnetic moment in a magnetic field $\vec{B}$ is $-\vec{\mu}\cdot\vec{B}$ and therefore the force, which is

[1]The proton also has a spin and hence a magnetic moment but, because of its much larger mass, its magnetic moment is much smaller than that of the electron and so does not play a role in our discussion.

minus the spatial gradient of the energy, is given by

$$\vec{F} = \vec{\nabla}\left(\vec{\mu}\cdot\vec{B}\right) \tag{1.2}$$

so

$$F_z = \vec{\mu}\cdot\frac{d\vec{B}}{dz}, \tag{1.3}$$

where we have assumed, without loss of generality, that the field changes as function of z. Hence a beam of hydrogen atoms in a non-uniform field varying in the z-direction will be deflected in the z-direction. For simplicity we assume that the field itself is also (predominantly) along the z-direction, see Fig. 1.5, so

$$F_z = \mu_z\frac{dB_z}{dz}, \tag{1.4}$$

and hence the deflection will be proportional to μ_z.

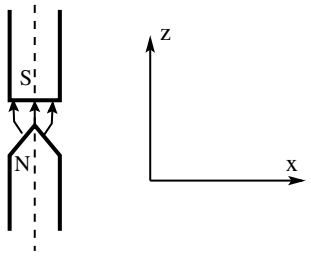


Figure 1.5: A cross section of the magnet in the Stern-Gerlach experiment. The beam goes between the poles of the magnet, into the plane i.e. in the y-direction, and intersects the symmetry axis (which is in the z-direction and shown by the dashed line).

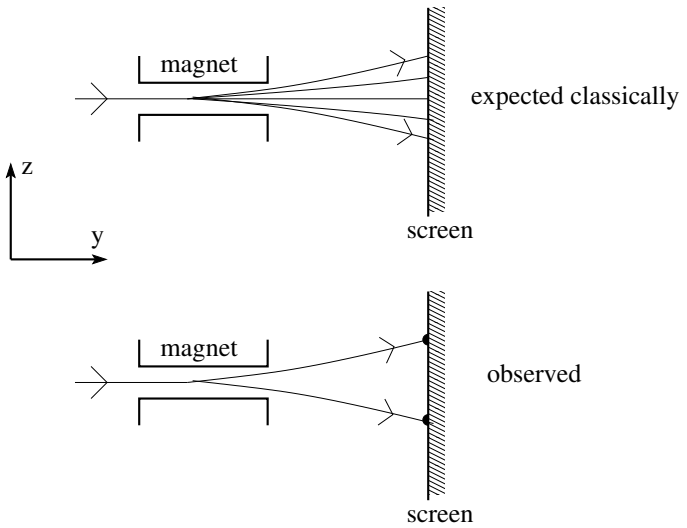


Figure 1.6: The Stern-Gerlach apparatus.

We send in a beam of unpolarized hydrogen atoms into a non-uniform field. This is the famous Stern-Gerlach (SG) experiment. Since the direction of $\vec{\mu}$ is random, classically μ_z takes a range of values, so we would expect a continuous range of deflections. However, it is found that only two beams

emerge, which are deflected in opposite directions, see Fig. 1.6. Since $\vec{\mu}$ is proportional to the spin it seems that the spin component along z has only two components, corresponding to states which we might label as[2] $|\uparrow_z\rangle$ and $|\downarrow_z\rangle$, or alternatively as $|0\rangle$ and $|1\rangle$ respectively.

Now suppose that we orientate the magnet so the field and its gradient are in the x-direction. Again we will see two beams emerging, indicating that μ_x has only two possible values $|\uparrow_x\rangle$ and $|\downarrow_x\rangle$.

How are $|\uparrow_x\rangle$ and $|\downarrow_x\rangle$ related to $|\uparrow_z\rangle$ and $|\downarrow_z\rangle$? We can get an idea of this if we run our beam first through a SG setup with the field in the z-direction and then pass one of the resulting beams through an SG setup in the x-direction as shown in Fig. 1.7. The final result is found to be two beams of equal intensity.

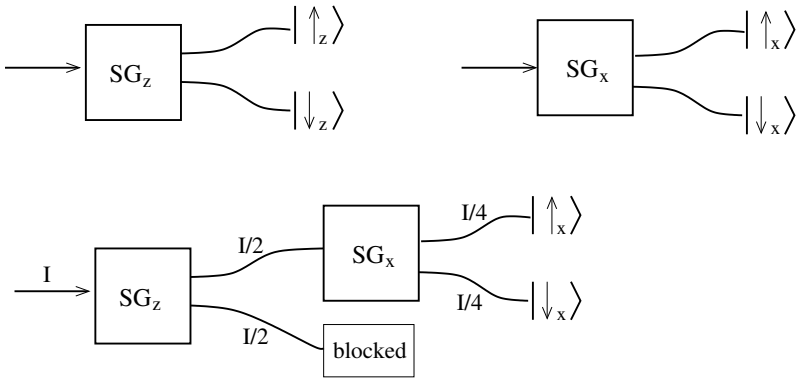


Figure 1.7: The upper figure shows schematically separate Stern-Gerlach experiments with the field in the z-direction (SG_z) and in the x direction (SG_x). The lower figure shows a double Stern-Gerlach experiment in which the beam is passed first through an SG apparatus with a field in the z-direction and then one of the beams is passed through an SG apparatus with the field in the x-direction.

It looks as though $|\uparrow_z\rangle$ can be thought of as $|\uparrow_x\rangle$ with probability 1/2 and $|\downarrow_x\rangle$ with probability 1/2. We will see in a future lecture that $|\uparrow_z\rangle$ is actually a superposition of $|\uparrow_x\rangle$ and $|\downarrow_x\rangle$ as follows:

$$|\uparrow_z\rangle = \frac{1}{\sqrt{2}}\left(|\uparrow_x\rangle + |\downarrow_x\rangle\right), \tag{1.5}$$

where we say that there is an *amplitude*[3] $1/\sqrt{2}$ for $|\uparrow_z\rangle$ to be $|\uparrow_x\rangle$ and amplitude $1/\sqrt{2}$ for it to be $|\downarrow_x\rangle$. As we shall also see later, the probability that a measurement gives a certain result is the square of the modulus of corresponding amplitude[4] so the probability of measuring $|\uparrow_x\rangle$ after the SG_x apparatus is 1/2 (as observed) and the same for $|\downarrow_x\rangle$.

It is also true that

$$|\uparrow_x\rangle = \frac{1}{\sqrt{2}}\left(|\uparrow_z\rangle + |\downarrow_z\rangle\right), \tag{1.6}$$

so if we run one of the beams from the SG_x apparatus in Fig. 1.7 through another SG_z apparatus we will get beams with equal intensity for $|\uparrow_z\rangle$ and $|\downarrow_z\rangle$, see Fig. 1.8. Note a surprising aspect of this result. After the first SG_z apparatus, there is zero probability for getting $|\downarrow_z\rangle$ (because we blocked it off), but after the SG_x apparatus there is a 50% probability for finding $|\downarrow_z\rangle$. In other words, a non-zero probability for getting $|\downarrow_z\rangle$ has been *generated* by the measurement. This is a clear example of a measurement (in this case that done by the SG_x apparatus) affecting the state of the system.

[2]The electron is the simplest two-state system so, in principle, it could be used as a qubit, but because it is charged it interacts too strongly with its environment to be useful.

[3]Sometimes called a probability amplitude

[4]The fact that probabilities add to 1, is why $|\alpha|^2 + |\beta|^2 = 1$ in Eq. (1.1).

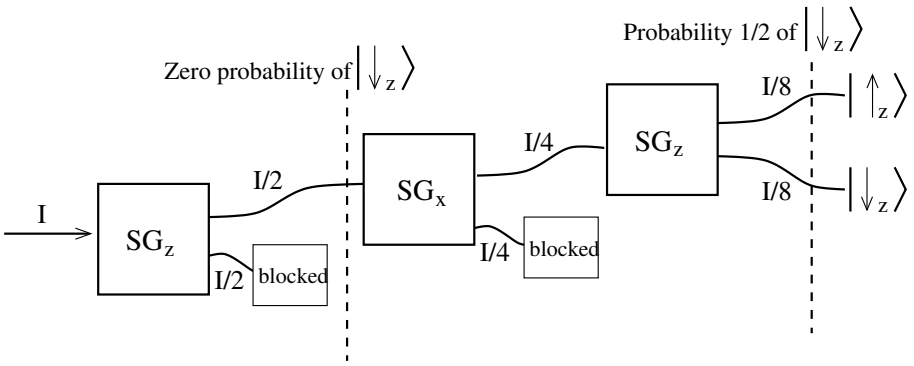


Figure 1.8: We now add another SG_z apparatus after the SG_x apparatus in Fig. 1.7. The result is equal intensity in the beams for $|\uparrow_z\rangle$ and $|\downarrow_z\rangle$. After each SG apparatus the upper line is for the "up" spin and the lower line for the "down" spin.

1.4 Photons

In the previous section we noted that the spin of the electron is a two-state quantum system. Here we discuss another two-state quantum system, the photon, the quantum of light.

Light is a transverse electric field, in which the electric field $\vec{E}$ and magnetic field $\vec{B}$ are perpendicular both to each other and to the direction of propagation specified by the wavevector $\vec{k}$. For example, if $\vec{E}$ is in the x direction, $\vec{B}$ in the y direction, and $\vec{k}$ in the z direction we have[5]

$$\begin{aligned}\vec{E} &= E_0\,\hat{x}\,e^{i(kz-\omega t)},\\ \vec{B} &= B_0\,\hat{y}\,e^{i(kz-\omega t)}.\end{aligned} \tag{1.7}$$

The direction of $\vec{E}$ is called the polarization direction. There are two distinct polarizations which we can call "horizontal" (along $\hat{x}$)

$$|\leftrightarrow\rangle, \qquad \text{equivalent to} \quad |\uparrow_z\rangle \equiv |0\rangle, \tag{1.8}$$

and "vertical", (along $\hat{y}$)

$$|\updownarrow\rangle, \qquad \text{equivalent to} \quad |\downarrow_z\rangle \equiv |1\rangle. \tag{1.9}$$

What are the analogs of $|\uparrow_x\rangle$ and $|\downarrow_x\rangle$? The answer is diagonal polarizations:

$$\begin{aligned}|\nearrow\rangle &\equiv \frac{1}{\sqrt{2}}\left(|\updownarrow\rangle + |\leftrightarrow\rangle\right), \qquad \text{equivalent to} \quad |\uparrow_x\rangle \equiv \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right),\\ |\nwarrow\rangle &\equiv \frac{1}{\sqrt{2}}\left(|\updownarrow\rangle - |\leftrightarrow\rangle\right), \qquad \text{equivalent to} \quad |\downarrow_x\rangle \equiv \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right).\end{aligned} \tag{1.10}$$

More details on the correspondence between photon polarization and qubit states will be given in Sec. 4.1.

Photons do not interact with each other to a measurable extent, and can not readily be stored, so they are unsuitable for most types of quantum computer, but have the advantage that they can be transmitted over great distances down optical fibers, preserving their polarization. These properties will be useful for some quantum protocols to be discussed in Chapter 21.

[5] We understand that the physical fields are the real parts of these expressions.

Chapter 2

Review of Linear Algebra

The theory of quantum mechanics is based on linear algebra which is a pre-requisite for the course and is standard material available in many books. In this chapter we summarize those topics in linear algebra which will be needed for this class. The treatment is quick and is intended as a review for students, assuming that they have seen the material before.

2.1 Vectors

An abstract vector $\vec{v}$ can be represented in terms of its N components $v_i, (i = 1, \cdots, N)$

$$\vec{v} = \sum_{i=1}^{N} v_i \hat{e}_i, \tag{2.1}$$

with respect to a set of basis vectors $\hat{e}_i$, which form an orthonormal set, i.e.

$$\vec{e}_i \cdot \vec{e}_j = \delta_{ij}, \tag{2.2}$$

where the left hand side is a scalar product

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{N} a_i \, b_i, \tag{2.3}$$

and δ_{ij} is the Kronecker delta function,

$$\delta_{ij} = \begin{cases} 1 & (i = j), \\ 0 & (i \neq j), \end{cases} . \tag{2.4}$$

We say that a vector $\vec{v}$ is normalized if $\vec{v} \cdot \vec{v} = 1$, and that two vectors $\vec{a}$ and $\vec{b}$ are orthogonal if $\vec{a} \cdot \vec{b} = 0$. A set of vectors is said to be orthonormal if each is normalized and every pair is orthogonal. The number of independent basis states required to represent any vector is called the size of the "vector space". It is denoted here by N.

The vector $\vec{v}$ can be represented in terms of its components v_i as a column vector

$$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}, \tag{2.5}$$

and its transpose as a row vector

$$\vec{v}^T = \begin{pmatrix} v_1 & v_2 & \cdots & v_N \end{pmatrix}. \tag{2.6}$$

The length of a vector is given by

$$|v| = \left(\sum_{i=1}^{N} v_i^2 \right)^{1/2} = (\vec{v} \cdot \vec{v})^{1/2}. \tag{2.7}$$

One can represent a vector with respect to different orthonormal bases rotated with respect to each other. If a vector has components v_i' with respect to the new basis, there is a linear relation between the old and new components,

$$\vec{v'} = M\vec{v}, \tag{2.8}$$

or, in terms of components

$$v_i' = \sum_{j=1}^{N} M_{ij} v_j, \tag{2.9}$$

where M is an $N \times N$ matrix with elements M_{ij}. In order that M describes a rotation (which preserves lengths of vectors and angles between them), it is necessary that M be an orthogonal matrix, i.e.

$$M^{-1} = M^T, \tag{2.10}$$

where M^T is the transpose matrix, and M^{-1} is the matrix inverse which means that $M^{-1}M = MM^{-1} = \mathbb{1}$ where $\mathbb{1}$ is the identity matrix. An example of a rotation matrix for two-component vectors is

$$M = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \tag{2.11}$$

where θ is the rotation angle.

The scalar product of two vectors is independent of basis, so

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{N} a_i b_i = \sum_{i=1}^{N} a_i' b_i'. \tag{2.12}$$

This is why $\vec{a} \cdot \vec{b}$ is called a *scalar* product.

2.2 Complex Vectors

In quantum mechanics, we need complex vectors, i.e. vectors with complex coefficients. The main new feature compared with real vectors is a slight difference in the definition of the scalar product, namely one takes the complex conjugate of the left hand vector, i.e.

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{N} a_i^\star b_i. \tag{2.13}$$

In terms of rules for matrix multiplication one can view the scalar product as the matrix product of the complex conjugate of the transpose vector (row vector) for a with the vector (column vector) for b, i.e.

$$\vec{a} \cdot \vec{b} \equiv \left(a^T\right)^\star b = \begin{pmatrix} a_1^\star & a_2^\star & \cdots & a_N^\star \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}, \tag{2.14}$$

in which $\left(a^T\right)^\star$ is an $1 \times N$ dimensional matrix, b is an $N \times 1$ dimensional matrix, and $\left(a^T\right)^\star b$ denotes matrix multiplication with the result being a single number (scalar).

The length of a complex vector, called the norm $|a|$ from now on, is still the square root of the scalar product of the vector with itself, i.e.

$$|a| = (\vec{a} \cdot \vec{a})^{1/2} = \left(\sum_{i=1}^{N} |a_i|^2 \right)^{1/2}. \tag{2.15}$$

2.3 Matrices

If A and B are matrices then the matrix product $C = AB$ is given in terms of its elements by

$$C_{ij} = \sum_{k=1}^{M} A_{ik}\, B_{kj}. \tag{2.16}$$

We assume here that A is of dimension $N \times M$ (N rows and M columns), in which case B must have M rows. If B has P columns then C is of dimension $N \times P$. It will sometimes be useful to think of a column vector as an $N \times 1$ dimensional matrix (N rows and 1 column), and a row vector as a $1 \times N$ dimensional matrix. Apart from vectors, the matrices in this course will be square (number of rows equals number of columns).

Matrix multiplication has the property that the order of multiplication matters in general. We define the commutator of two matrices by

$$[A, B] \equiv AB - BA. \tag{2.17}$$

If $[A, B] = 0$ we say that A and B commute. However, in general matrices do not commute, i.e. their commutator is non-zero. Lack of commutation of matrices will have important consequences in quantum mechanics.

Some important, special types of matrices are:

- Symmetric: $M^T = M$ (M^T is the transpose, so $\left(M^T\right)_{ij} = M_{ji}$).
- Orthogonal: $M^T = M^{-1}$.

In quantum mechanics we will deal with complex matrices, as well as complex vectors. In the case of complex matrices, one is usually interested in Hermitian matrices rather than symmetric ones, and unitary matrices rather than orthogonal ones, where these are defined by:

- Hermitian: $M^\dagger = M$ ($M^\dagger$ is the adjoint, the complex conjugate of the transpose so $M^\dagger = \left(M^T\right)^\star$).
- Unitary: $M^\dagger = M^{-1}$.
  Unitary matrices have the useful property that the rows form orthonormal vectors, as do the columns. To determine if a matrix is unitary it may be easier to do this check rather than compute the inverse.

Hermitian and unitary matrices play important roles in quantum mechanics.

2.4 Matrix Diagonalization

Let A be $N \times N$ matrix and $\vec{x}$ an N-component vector. Then if $A\vec{x}$ is proportional to $\vec{x}$ itself, i.e. if

$$\begin{aligned} A\vec{x} &= \lambda\vec{x} \quad \text{or, in terms of elements,} \\ \sum_{j=1}^{N} A_{ij}x_j &= \lambda x_i, \end{aligned} \tag{2.18}$$

then we say that λ is an eigenvalue and $\vec{x}$ the corresponding eigenvector of A. There are N eigenvalues which may not all be distinct. If two or more eigenvalues are equal we say that they are degenerate. We can always multiply an eigenvector by a constant and it remains an eigenvector. In quantum mechanics we will need to choose this multiplicative constant so the vector is "normalized", i.e. has unit length.

The eigenvalues are obtained from solving

$$\det(A - \lambda\mathbb{1}) = 0, \tag{2.19}$$

where det is short for determinant. Expanding out the determinant gives an N-th order polynomial equation for λ. One can then get the eigenvectors by solving the linear equations in Eq. (2.18) for each value of λ.

Once one has the eigenvectors, a matrix A can be "diagonalized" as follows[1]:

$$D = S^{-1}AS, \tag{2.20}$$

where D is a diagonal matrix with the eigenvalues of A on the diagonal,

$$D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix}, \tag{2.21}$$

and the matrix S, which effects the diagonalization, is constructed out of the eigenvectors of A as follows:

$$S = \left(\vec{e}^{(1)}, \quad \vec{e}^{(2)}, \quad \cdots, \vec{e}^{(N)}\right), \tag{2.22}$$

where $\vec{e}^{(i)}$ is the i-th eigenvector of A written as a column vector.

The eigenvalues and eigenvectors of Hermitian matrices have special properties:

- The eigenvalues are all real.
- Eigenvectors corresponding to unequal (non-degenerate) eigenvalues are orthogonal. For eigenvectors corresponding to degenerate eigenvalues, one can form linear combinations which are orthogonal.

If we consider two $N \times N$ matrices A and B, one can show that they have the same eigenvectors if and only if the matrices commute, i.e. if $[A, B] \equiv AB - BA = 0$. This result will have important consequences in quantum mechanics.

[1] There are some matrices with degenerate eigenvalues which have less than N independent eigenvectors. These can not be diagonalized. However, this situation does not occur for Hermitian or unitary matrices, the two categories that are of principle interest in quantum mechanics, and so we will ignore non-diagonalizable matrices in this course.

2.5 Some Important 2×2 matrices

In quantum computing one deals with 2×2 matrices because qubits have two states. Important examples of 2×2 Hermitian matrices are the Pauli (spin) matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.23}$$

(called σ_x, σ_y and σ_z in the physics literature).

Any 2×2 matrix can be expressed as a linear combination of the three Pauli matrices plus the identity. To see this note that X, Y, Z and $\mathbb{1}$ are linearly independent (we can't write any one as a linear combination of the others). Also a general 2×2 matrix

$$A = \begin{pmatrix} t & u \\ v & w \end{pmatrix} \tag{2.24}$$

has 4 complex elements, and so a total of 8 real parameters. If we write

$$A = a_0 \mathbb{1} + a_x X + a_y Y + a_z Z \tag{2.25}$$

then there are also 4 complex coefficients (8 real parameters). Hence there are just the right number of coefficients to specify any 2×2 matrix, so Eq. (2.25) is a general expression for a 2×2 matrix.

Let's determine the eigenvalues and eigenvectors of X. The eigenvalues λ are obtained from

$$\begin{vmatrix} 0-\lambda & 1 \\ 1 & 0-\lambda \end{vmatrix} = 0, \tag{2.26}$$

which gives $\lambda^2 - 1 = 0$ or $\lambda = \pm 1$. These are real, which they must be since X is Hermitian.

Let us now get the eigenvectors. We denote the corresponding normalized eigenvectors by $\vec{e}_{+1}$ and $\vec{e}_{-1}$ and indicate the coefficients by a and b.

- $\lambda = +1$.

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}, \tag{2.27}$$

  which gives the equations $b = a$ and $a = b$, which are the same. To normalize the eigenvector, we take $a = b = 1/\sqrt{2}$, so

$$\vec{e}_{+1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{2.28}$$

- $\lambda = -1$.

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = - \begin{pmatrix} a \\ b \end{pmatrix} \tag{2.29}$$

  which gives the two equations $b = -a$ and $a = -b$ (which are equivalent). The normalized eigenvector is therefore

$$\vec{e}_{-1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}. \tag{2.30}$$

  The eigenvectors $\vec{e}_{+1}$ and $\vec{e}_{-1}$ are orthogonal, as we know they must be since X is Hermitian.

It is instructive for the student to show that the eigenvalues of Y and Z are also ± 1 and to determine their eigenvectors. The student should also be able to show that X, Y and Z are not only Hermitian but also unitary.

Pauli matrices have the property that the commutator of the two of them is proportional to the third one, e.g.

$$[X, Y] = 2iZ, \tag{2.31}$$

and similarly $[Y, Z] = 2iX$ and $[Z, X] = 2iY$. Furthermore, if we define the *anti-commutator* of two matrices by

$$\{A, B\} \equiv AB + BA, \tag{2.32}$$

then, interestingly, different Pauli matrices *anti-commute*, e.g.

$$\{X, Y\} = 0, \tag{2.33}$$

and similarly $\{Y, Z\} = \{Z, X\} = 0$.

Another 2×2 matrix which is very important in quantum computing is the Hadamard, defined by

$$H = \frac{1}{\sqrt{2}}(X + Z) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.34}$$

The Hadamard also has eigenvalues ± 1.

2.6 Properties of Matrices

Two properties of square matrices will be important: the trace, which is the sum of the diagonal elements, and the determinant. It is left as an exercise for the student to show (i) that the trace is the sum of the eigenvalues, and (ii) that the trace of a product of matrices is invariant under a cyclic permutation of the matrices so, for example, $\mathrm{Tr}\, AB = \mathrm{Tr}\, BA$ even if A and B don't commute.

We will now show (iii) that the determinant is the product of the eigenvalues. If we multiply Eq. (2.20) on the left by S and on the right by S^{-1} we get

$$A = SDS^{-1}. \tag{2.35}$$

An important result of linear algebra, which is not as well known in the physics community as it should be, is that determinant of a product of matrices is equal to the product of the determinants, i.e.

$$\det(AB) = \det A \, \det B. \tag{2.36}$$

Taking the determinant of both sides of Eq. (2.35) gives

$$\begin{aligned} \det A &= \det S \det D \det S^{-1} \\ &= \det D \det S \det S^{-1} \\ &= \det D \det \left(SS^{-1}\right) \\ &= \det D = \prod_{m=1}^{N} \lambda_m, \end{aligned} \tag{2.37}$$

which is the desired result.

Chapter 3

Introduction to Quantum Mechanics

In this chapter we give an introduction to quantum mechanics. A good textbook on the subject, at an undergraduate level, is Griffiths [Gri05].

3.1 Quantum States as Complex Vectors

In Chapter 2 we reviewed linear algebra, including vectors, generalized to the case where the coefficients of the vectors are complex.

We now describe the basic postulates of quantum mechanics. We will see that the framework is precisely that of complex vectors. The notation, however, is quite different and so, for the next few equations, we will show both a statement concerning quantum mechanics in quantum mechanics notation, *and* the corresponding statement for complex vectors in the standard notation of linear algebra.

While the discussion which follows may seem very abstract don't forget that quantum mechanics is arguably the most successful theory in all of physics, with countless precise comparisons between theory and experiment, some to the most exquisite accuracy[1].

Now we get started with quantum mechanics:

Ansatz 1: The state of a quantum system is a complex vector (which we shall often call a "state vector" or just a vector).

In quantum computing one uses the notation of Dirac, in which a quantum state is written as $|\psi\rangle$.

$$\text{QM state}: \quad |\psi\rangle, \quad \Longleftrightarrow \quad \text{complex vector}: \quad \vec{v}. \tag{3.1}$$

In equations with the double arrow $\Longleftrightarrow$ in the middle, the part to the left of the arrow is in the notation of quantum mechanics, and the part to the right is the corresponding statement in standard linear algebra notation. The state $|\psi\rangle$ can be expressed as a linear combination of basis states $|n\rangle$,

$$|\psi\rangle = \sum_{n=1}^{N} c_n |n\rangle, \quad \Longleftrightarrow \quad \vec{v} = \sum_{n=1}^{N} v_n \hat{e}_n. \tag{3.2}$$

We can write the state as a column vector

$$|\psi\rangle = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} \quad \Longleftrightarrow \quad \vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}. \tag{3.3}$$

[1] For example, experimental and theoretical values for the magnetic moment of the electron agree to better than a part in a *trillion*, see Eq. (16.13d) of `https://www.mdpi.com/2218-2004/7/2/45/pdf`.

We also introduce the dual state vector, denoted by $\langle\psi|$. This corresponds to the complex conjugate of the transpose vector introduced in Eq. (2.14) in the context of the scalar product of a complex vector. In other words, if $|\phi\rangle$ is represented as a column vector by

$$|\phi\rangle = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{pmatrix}, \tag{3.4}$$

then the corresponding dual vector is

$$\langle\phi| = \begin{pmatrix} d_1^\star & d_2^\star \cdots & d_N^\star \end{pmatrix} \quad \Longleftrightarrow \quad \left(v^T\right)^\star = \begin{pmatrix} v_1^\star & v_2^\star \cdots & v_N^\star \end{pmatrix}, \tag{3.5}$$

i.e. a row vector in which the coefficients are the complex conjugate of the coefficients in the original column vector.

The scalar product of two vectors is called the "inner product" in a general context and this nomenclature will be used here from now on. In quantum mechanics, the inner product of a vector $|\psi\rangle$ with vector $|\phi\rangle$ is written as $\langle\phi|\psi\rangle$.

$$\langle\phi|\psi\rangle = \sum_{n=1}^{N} d_n^\star c_n \quad \Longleftrightarrow \quad \vec{a}\cdot\vec{b} = \sum_{n=1}^{N} a_n^\star b_n. \tag{3.6}$$

From this definition it follows that

$$\langle\phi|\psi\rangle = \langle\psi|\phi\rangle^\star. \tag{3.7}$$

The length of a vector in quantum mechanics is called the "norm" and written $\|\psi\|$. As with ordinary vectors, the norm of a state vector in quantum mechanics is the square root of the inner product with itself, i.e.

$$\|\psi\| = \langle\psi|\psi\rangle^{1/2} = \left(\sum_{n=1}^{N} |c_n|^2\right)^{1/2} \quad \Longleftrightarrow \quad |v| = (\vec{v}\cdot\vec{v})^{1/2} = \left(\sum_{i=1}^{n} |v_i|^2\right)^{1/2}. \tag{3.8}$$

As we shall see later, in quantum mechanics state vectors must have unit norm. Such vectors are said to be normalized.

Orthogonality. Two state vectors are said to be orthogonal if their inner product is zero:

$$\langle\phi|\psi\rangle = \langle\psi|\phi\rangle = 0, \quad \Longleftrightarrow \quad \vec{a}\cdot\vec{b} = \vec{b}\cdot\vec{a} = 0. \tag{3.9}$$

We choose basis states $|n\rangle$ which are orthonormal, i.e. normalized and orthogonal,

$$\langle n|m\rangle = \delta_{nm}, \quad \Longleftrightarrow \quad \vec{e}_n\cdot\vec{e}_m = \delta_{nm}. \tag{3.10}$$

So far, in this chapter we have emphasized the correspondence between quantum mechanical states and complex vectors. Now that we are familiar with this correspondence, from now on we will describe the formulation of quantum mechanics using only quantum mechanics notation.

It will be useful to rewrite Eq. (3.2) for a linear superposition in a different way. Starting with Eq. (3.2),

$$|\psi\rangle = \sum_{n=1}^{N} c_n |n\rangle, \tag{3.11}$$

we take the inner product of both sides with the dual of one of the basis states, $\langle m|$ say. Using the orthonormality property in Eq. (3.10) gives us[2]

$$c_n = \langle n|\psi\rangle, \tag{3.12}$$

so we can rewrite Eq. (3.11) as

$$|\psi\rangle = \sum_{n=1}^{N} |n\rangle\langle n|\psi\rangle. \tag{3.13}$$

We call $\langle n|\psi\rangle$ the *probability amplitude* for the state $|\psi\rangle$ to be in basis state $|n\rangle$. Equation (3.13) shows us that

$$\sum_{n=1}^{N} |n\rangle\langle n| = \mathbb{1}, \tag{3.14}$$

the identity matrix. Equation (3.14) is sometimes called a completeness relation. A single term in this sum, $|n\rangle\langle n|$ is an $N \times N$ matrix with all elements 0 except that the n-th diagonal element is 1.

To make our discussion more concrete consider the following example of a 2-state system, i.e. a single qubit,

$$|\psi'\rangle = \begin{pmatrix} 1 \\ 2i \end{pmatrix}. \tag{3.15}$$

This is not normalized because the norm is

$$\|\psi'\| = \sqrt{1^2 + |2i|^2} = \sqrt{1+4} = \sqrt{5}. \tag{3.16}$$

To get a valid quantum state it must be properly normalized so we divide by the norm. Hence

$$|\psi\rangle = \frac{1}{\sqrt{5}}|\psi'\rangle = \frac{1}{\sqrt{5}}\begin{pmatrix} 1 \\ 2i \end{pmatrix} \tag{3.17}$$

is a valid quantum state. To get the dual state vector we take the complex conjugate of the transpose, so

$$\langle\psi| = \frac{1}{\sqrt{5}}\begin{pmatrix} 1 & -2i \end{pmatrix}. \tag{3.18}$$

Suppose we also have a second state,

$$|\phi\rangle = \frac{1}{\sqrt{5}}\begin{pmatrix} 2 \\ -i \end{pmatrix}, \tag{3.19}$$

which we see is normalized because $\sqrt{2^2 + |-i|^2} = \sqrt{5}$. What then is the inner product $\langle\psi|\phi\rangle$? We have

$$\langle\psi|\phi\rangle = \frac{1}{5}\begin{pmatrix} 1 & -2i \end{pmatrix}\begin{pmatrix} 2 \\ -i \end{pmatrix} = \frac{1}{5}(1\cdot 2 + (-2i)\cdot(-i)) = 0, \tag{3.20}$$

so $|\psi\rangle$ and $|\phi\rangle$ are actually orthogonal. In this example we had to be careful with the factors of i because a complex conjugate is taken when we form the dual vector (which we need to get the inner product with another vector).

To make sure we haven't forgotten it, let's reiterate (with a bit more math jargon) the first Ansatz of quantum mechanics which we stated at the beginning of this section:
Ansatz 1: The state of a quantum system is a vector in a complex vector space (technically a Hilbert space though we won't need that level of mathematical sophistication here).

[2]For ordinary vectors the corresponding expression would be $v_n = \vec{e}_n \cdot \vec{v}$.

3.2 Phases

At this point it is convenient to discus an important topic, namely *phases*. Suppose we have a 2-state system with complex amplitudes, which we write in polar form as

$$|\psi\rangle = r_0 e^{i\theta_0}|0\rangle + r_1 e^{i\theta_1}|1\rangle, \tag{3.21}$$

where $r_0^2 + r_1^2 = 1$ for normalization. Let's take out the factor of $e^{i\theta_0}$, so

$$|\psi\rangle = e^{i\theta_0}\left(r_0|0\rangle + r_1 e^{i(\theta_1-\theta_0)}|1\rangle \right). \tag{3.22}$$

We call θ_0 the *global* phase which turns out to have no physical significance, while $\theta_1 - \theta_0$ is the *relative* phase (of basis states $|1\rangle$ and $|0\rangle$) which is important because it gives rise to interference. It is crucial to understand the difference between global phase and relative phase. **States which differ only in the overall phase are physically identical**. As we will see in Sec. 3.6 the reason for this is that no measurement can distinguish states which only differ by a global phase. By contrast, **states which differ in a relative phase are physically distinct** because measurements can distinguish between them.

For example,

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}, \qquad |\psi_2\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\end{pmatrix}, \tag{3.23}$$

describe the same state because one is just the negative of the other. By contrast,

$$|\psi_1'\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}, \qquad |\psi_2'\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}, \tag{3.24}$$

describe different states because the relative phase of $|1\rangle$ and $|0\rangle$ is different in the two cases (0 for $|\psi_1'\rangle$ and π for $|\psi_2'\rangle$).

3.3 Observables

How is all this abstract stuff about complex vectors related to the real world, i.e. to quantities that we can measure.

The answer is that an observable quantity will be an *operator*, $\hat{O}$ say, acting on these vectors. The "hat" symbol " $\hat{}$ " indicates an operator, though for simplicity of notation we will usually omit the hat when context makes clear that we are dealing with an operator. In terms of components, operators are represented by matrices.

An operator acting on a state vector gives another state vector, so

$$\hat{O}|\psi\rangle = |\phi\rangle, \tag{3.25}$$

A crucial point is that operators in quantum mechanics are *linear*, so

$$\hat{O}\left(a|\psi\rangle + b|\phi\rangle \right) = a\,\hat{O}|\psi\rangle + b\,\hat{O}|\phi\rangle. \tag{3.26}$$

This brings us to the second Ansatz of quantum mechanics:

Ansatz 2: Observables are represented by linear Hermitian operators. The result of a measurement is one of the eigenvalues of the corresponding operator $\hat{O}$. After the measurement, the system is in the eigenstate corresponding to the measured eigenvalue.

Why is it assumed that quantity which can be measured is represented by a *Hermitian* operator? The answer is that the eigenvalues of a Hermitian operator (matrix) are guaranteed to be real, and we know that the results of a measurement must be real.

We now discuss how to represent operators as a matrix using the Dirac notation. We take orthonormal basis vectors $|n\rangle$ which have the property $\langle m|n\rangle = \delta_{mn}$. In terms of components, $|n\rangle$ will be a column vector with the n-th entry equal to 1 and all the others zero. In other words

$$|n\rangle = \qquad (\text{row } n) \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix}. \tag{3.27}$$

Consider the action of an operator A on one of the basis vectors $|n\rangle$. It will give a linear combination of the basis vectors.

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} & \cdots & A_{1,N-1} & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2n} & \cdots & A_{2,N-1} & A_{2N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} & \cdots & A_{n,N-1} & A_{nN} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ A_{N-1,1} & A_{N-1,2} & \cdots & A_{N-1,n} & \cdots & A_{N-1,N-1} & A_{N-1,N} \\ A_{N1} & A_{N2} & \cdots & A_{Nn} & \cdots & A_{N,N-1} & A_{NN} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ \vdots \\ c_{N-1} \\ c_N \end{pmatrix}. \tag{3.28}$$

We see that c_k is equal to the element of A on the k-th row and n-th column, i.e. A_{kn}. We can therefore write Eq. (3.28) as

$$A|n\rangle = \sum_k A_{kn}|k\rangle. \tag{3.29}$$

Acting on the left with the dual vector $\langle m|$ and using the orthonormality of the basis vectors, we get

$$A_{mn} = \langle m|A|n\rangle, \tag{3.30}$$

which is the connection between the usual suffix notation for an element of a matrix, A_{mn}, and the Dirac notation for the same thing, $\langle m|A|n\rangle$. They both refer to the m-th row and nth column of the matrix A.

Recall that the definition of the adjoint of a matrix is $A^\dagger = \left(A^T\right)^\star$. Hence, in Dirac notation,

$$\langle m|A^\dagger|n\rangle = \langle n|A|m\rangle^\star. \tag{3.31}$$

If A is Hermitian then it is equal to its adjoint so

$$\langle m|A|n\rangle = \langle n|A|m\rangle^\star \qquad (\text{for } A \text{ Hermitian}). \tag{3.32}$$

To gain still more familiarity with the Dirac notation consider $\langle\phi|A|\psi\rangle$. If we write this out in components in some basis, then $|\psi\rangle$ is a column vector, A is a matrix and $\langle\phi|$ is a row vector, i.e. we have

$$\langle\phi|A|\psi\rangle = \begin{pmatrix} \phi_1^\star & \phi_2^\star & \cdots & \phi_N^\star \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_N \end{pmatrix}, \tag{3.33}$$

in an obvious notation. The multiplication can be done either by acting with A on $|\psi\rangle$ to get $|A\psi\rangle$ and then taking the inner product with $\langle\phi|$, or by acting with A to the left on $\langle\phi|$ and then taking the inner product with $|\psi\rangle$. But what does acting with A to the left on $\langle\phi|$ mean? Let's suppose that

$$\langle\phi|A = \langle\mu|. \tag{3.34}$$

Then we have

$$\begin{pmatrix}\phi_1^\star & \phi_2^\star & \cdots & \phi_N^\star\end{pmatrix}\begin{pmatrix}A_{11} & A_{12} & \cdots & A_{1N}\\ A_{21} & A_{22} & \cdots & A_{2N}\\ \vdots & \vdots & \ddots & \vdots\\ A_{N1} & A_{N2} & \cdots & A_{NN}\end{pmatrix} = \begin{pmatrix}\mu_1^\star & \mu_2^\star & \cdots & \mu_N^\star\end{pmatrix}. \tag{3.35}$$

Evaluating components gives

$$\mu_m^\star = \sum_k \phi_k^\star A_{km}. \tag{3.36}$$

This can be rearranged as

$$\begin{aligned}\mu_m &= \sum_k \phi_k A_{km}^\star\\ &= \sum_k \left(A^T\right)_{mk}^\star \phi_k = \sum_k A_{mk}^\dagger \phi_k,\end{aligned} \tag{3.37}$$

or, for the vector as a whole

$$|\mu\rangle = A^\dagger|\phi\rangle. \tag{3.38}$$

which is equivalent to Eq. (3.34). Hence the action of A acting to the left on $\langle\phi|$ can be written as

$$\langle\phi|A = \langle\, A^\dagger\phi\,|. \tag{3.39}$$

Summarizing, we see that in $\langle\phi|A|\psi\rangle$, the operator A can be considered to act either to the left or the right as follows:

$$\langle\phi|A|\psi\rangle = \langle\, A^\dagger\phi\,|\psi\rangle = \langle\phi|\,A\psi\,\rangle. \tag{3.40}$$

In quantum mechanics A will commonly be a Hermitian operator (since observables are represented by Hermitian operators) for which $A^\dagger = A$, so A acts equally to the right and to the left as follows:

$$\langle\phi|A|\psi\rangle = \langle\, A\phi\,|\psi\rangle = \langle\phi|\,A\psi\,\rangle \qquad \text{(for } A \text{ Hermitian)}. \tag{3.41}$$

When dealing with standard vectors, we know that we can work with different sets of bases rotated with respect to each other. In quantum mechanics, too, it will be convenient to represent state vectors in terms of different bases, transformed with respect to each other.

A common basis for a single qubit comprises the states $|0\rangle$ and $|1\rangle$, and in this basis the Pauli operator Z is diagonal, see Eq. (2.23). This is the most common basis used in quantum computing, and is called the computational basis. Since Z is diagonal in this basis the eigenvectors of Z are the basis vectors. For this reason the computational basis is sometimes called the Z-basis.

We will also need to consider other bases, one of the most common being the X-basis, i.e. the basis in which X (see Eq. (2.23)) is diagonal. You should be able to show that the eigenvalues of X are $+1$ and -1, with corresponding eigenvectors, called $|+\rangle$ and $|-\rangle$ (sometimes $|0_x\rangle$ and $|1_x\rangle$), given by

$$\begin{aligned}|0_x\rangle \equiv |+\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\\ |1_x\rangle \equiv |-\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right).\end{aligned} \tag{3.42}$$

From these results it follows that, in the X basis, the Pauli X-matrix is written as

$$X = \begin{matrix} \\ \langle + | \\ \langle - | \end{matrix} \begin{matrix} \begin{matrix} |+\rangle & |-\rangle \end{matrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{matrix}, \tag{3.43}$$

which looks just like the Pauli-Z matrix in the Z (computational) basis.

There is a linear relation between the new basis vectors and the old ones. Denoting the old basis vectors by Latin letters, e.g. $|n\rangle$, and the new basis vectors by Greek letters, e.g. $|\alpha\rangle$, we write

$$|\alpha\rangle = \sum_n U_{\alpha n} |n\rangle. \tag{3.44}$$

The new basis vectors must be orthonormal, like the old set, and this constrains the matrix of coefficients U in a way that we will now determine. Writing the equivalent of Eq. (3.44) in terms of row vectors and taking the complex conjugate, we get the following transformation for the dual basis state vectors

$$\langle \beta | = \sum_k U^\star_{\beta k} \langle k |. \tag{3.45}$$

Taking the inner product of Eqs. (3.44) and (3.45) gives

$$\begin{aligned} \langle \beta | \alpha \rangle &= \sum_{n,k} U^\star_{\beta k} U_{\alpha n} \langle k | n \rangle \\ &= \sum_n U^\star_{\beta n} U_{\alpha n} \\ &= \sum_n U_{\alpha n} \left(U^T\right)^\star_{n\beta} = \sum_n U_{\alpha n} U^\dagger_{n\beta} = \left(U U^\dagger\right)_{\alpha\beta}, \end{aligned} \tag{3.46}$$

where we used that $\langle k | n \rangle = \delta_{kn}$ to get the second line. However, $\langle \beta | \alpha \rangle = \delta_{\alpha\beta}$ and so we must have $U U^\dagger = \mathbb{1}$, the identity matrix. Thus the matrix of coefficients which transforms from one basis to another as in Eq. (3.44) must be unitary.

As an example, according to Eq. (3.42) the matrix which transforms from the Z-basis to the X basis for one qubit is

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3.47}$$

We can verify that this matrix is unitary by evaluating its inverse and checking that $U^{-1} = U^\dagger$, or, more simply, by recalling that the rows of a unitary matrix are orthonormal vectors, and the same for the columns. By inspection, this is the case here.

3.4 Outer Product Notation

For orthonormal basis vectors, we have $\langle i | j \rangle = \delta_{ij}$. As a further exercise in familiarization with the Dirac notation, consider what we mean if we write the vector and the dual vector the other way round i.e. $|i\rangle\langle j|$, which is called an "outer product". It is actually a matrix. By sandwiching it on the left and right by basis states we see that it is a matrix, whose entries are all zero except for the element in

the i-th row and j-th column which is 1. In other words

$$\text{(col. } j\text{)} \tag{3.48}$$

$$|i\rangle\langle j| = (\text{row } i) \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \end{pmatrix}. \tag{3.49}$$

If $j = i$, then we have a 1 in the i-th diagonal element and 0 everywhere else. This is a projection operator on to state i, so we denote it by P_i, i.e.

$$P_i = |i\rangle\langle i|. \tag{3.50}$$

One can see it is a projection operator because, if it acts on an arbitrary state $|\psi\rangle$, we have

$$P_i|\psi\rangle = |i\rangle\,\langle i|\psi\rangle, \tag{3.51}$$

which is the amplitude $\langle i|\psi\rangle$ for $|\psi\rangle$ to be along $|i\rangle$, times the state $|i\rangle$.

Clearly $\sum_i P_i$ has 1 on all the diagonal elements and is zero otherwise, so it is the identity matrix, i.e.

$$\sum_i P_i \equiv \sum_i |i\rangle\langle i| = \mathbb{1}, \tag{3.52}$$

which is also known as a completeness relation, see Eq. (3.14).

In our discussion of the density matrix in Chapter 5 we will come across outer products of the form $|\psi\rangle\langle\psi|$ where $|\psi\rangle$ is some arbitrary state which can be written as a linear combination of basis states,

$$|\psi\rangle = \sum_n c_n|n\rangle, \tag{3.53}$$

or equivalently, in Dirac notation

$$|\psi\rangle = |n\rangle\,\langle n|\psi\rangle, \tag{3.54}$$

since $c_n = \langle n|\psi\rangle$, see Eq. (3.12). Hence the matrix element of $|\psi\rangle\langle\psi|$ on the n-th row and m-th column, is

$$\langle n|\psi\rangle\langle\psi|m\rangle \quad (= c_n c_m^*). \tag{3.55}$$

3.5 Functions of operators

We will need to evaluate functions of operators. For example what is e^A? In this case there is a convergent series expansion which can be used to evaluate the function;

$$e^A = 1 + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots. \tag{3.56}$$

In some cases the infinite series can be evaluated in closed form. Consider for example e^{cX} where c is a constant and X is given in Eq. (2.23). We have $X^2 = \mathbb{1}$ and so $X^3 = X^5 \cdots = X^{2n+1} \cdots = X$, while

$X^2 = X^4 \cdots = X^{2n} \cdots = \mathbb{1}$. Hence

$$\begin{aligned} e^{cX} &= \mathbb{1}\left(1 + \frac{c^2}{2!} + \frac{c^4}{4!} + \cdots\right) + X\left(c + \frac{c^3}{3!} + \frac{c^5}{5!} + \cdots\right), \\ &= \mathbb{1}\,\cosh c + X\,\sinh c = \begin{pmatrix} \cosh c & \sinh c \\ \sinh c & \cosh c \end{pmatrix}. \end{aligned} \tag{3.57}$$

More generally, we can evaluate a function of an operator by diagonalizing it. Consider first a diagonal matrix,

$$D = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix}. \tag{3.58}$$

When multiplying D by itself n times, say, all that happens is each diagonal element is multiplied by itself n times. Hence if $f(D)$ is some function of D which can be represented by a series expansion, we have

$$f(D) = \begin{pmatrix} f(\lambda_1) & 0 & \cdots & 0 \\ 0 & f(\lambda_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f(\lambda_N) \end{pmatrix}. \tag{3.59}$$

If the function $f(x)$ for scalar argument x does not have a series expansion, we take Eq. (3.59) as the *definition* of the matrix function $f(D)$ for a diagonal matrix D.

In general, a matrix A is not already in diagonal form. However, we can diagonalize it by a similarity transform, see Eq. (2.35), which we repeat here:

$$A = SDS^{-1}, \tag{3.60}$$

where D is a diagonal matrix with the eigenvalues of A on the diagonal. Hence it follows that

$$\begin{aligned} A^2 &= SDS^{-1}SDS^{-1} = SD^2S^{-1}, \\ A^3 &= SDS^{-1}SDS^{-1}SDS^{-1} = SD^3S^{-1}, \quad \text{and so} \\ A^n &= SD^nS^{-1}, \quad \text{and hence} \\ f(A) &= Sf(D)S^{-1} \\ &= S\begin{pmatrix} f(\lambda_1) & 0 & \cdots & 0 \\ 0 & f(\lambda_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f(\lambda_N) \end{pmatrix} S^{-1}, \end{aligned} \tag{3.61}$$

which is the desired expression showing how to construct a function of a matrix from its eigenvalues and eigenvectors.

3.6 Measurements

Now we have to discuss in detail the vexed topic of measurement in quantum mechanics. The reason for using the term "vexed" will become clear later, especially in Chapter 7 when we discuss a famous thought experiment of Einstein, Podolsky and Rosen (EPR).

In a measurement, our delicate quantum system is brought into contact with a macroscopic experimental apparatus. Measurement is an irreversible process and as such has a special status in quantum mechanics.

Assume that the Hermitian operator A corresponding to the measured quantity of interest has eigenvalues λ_n and normalized eigenvectors $|n\rangle$. Because A is Hermitian the eigenvalues are real. In addition, for a Hermitian matrix of size N there are N orthogonal eigenvectors which can therefore be used as a basis. Hence, we can write the state of the system before measurement, $|\psi\rangle$, as a linear superposition of the eigenvectors of A,

$$\begin{aligned} |\psi\rangle &= \sum_{n=1}^{N} a_n |n\rangle. \\ &= \sum_{n=1}^{N} |n\rangle \langle n|\psi\rangle, \end{aligned} \tag{3.62}$$

where the last line is from Eq. (3.13).

According to ansatz 2 in Sec. 3.3, a measurement will give one of the eigenvalues, λ_n, but which one? To answer this question, we need to add one more ingredient to our Ansatz 2, one which was first proposed by Born in a footnote in a 1926 paper, and which is therefore called the "Born rule". This states that the probability, $P(n)$, to get eigenvalue λ_n (and after the measurement to leave the system in eigenstate $|n\rangle$), is the square of the modulus of the amplitude a_n, i.e.

$$P(n) = |a_n|^2 \equiv |\langle n|\psi\rangle|^2 \equiv \langle \psi|n\rangle \langle n|\psi\rangle, \tag{3.63}$$

where we used Eq. (3.12) and that $\langle \psi|n\rangle = \langle n|\psi\rangle^\star$. Since probabilities must add up to 1, it follows that state vectors in quantum mechanics must be normalized to unity, i.e.

$$1 = \sum_n P(n) = \sum_n |a_n|^2 = \sum_n |\langle n|\psi\rangle|^2 = \sum_n \langle \psi|n\rangle \langle n|\psi\rangle = \langle \psi|\psi\rangle. \tag{3.64}$$

Note that the probability of a getting a particular measured value only depends on the square of the modulus of the amplitude of the corresponding eigenstate. This means that the global phase of a state has no physical significance since no measurement can distinguish two states which differ only by a global phase.

However if two states differ in a relative phase there are measurements which can distinguish between them. For example, $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ are eigenstates of X with eigenvalues $+1$ and -1 respectively, so a measurement of X will give different results ($+1$ for $|+\rangle$ and -1 for $|-\rangle$, with probability 1 in both cases).

We therefore have to complete our Ansatz 2 to include the probabilities of different results:

Ansatz 2′: "Observables are represented by a linear Hermitian operator. Measurement of an observable corresponding to a (linear) Hermitian operator $\hat{O}$ gives one of the eigenvalues of $\hat{O}$. The probability of getting an eigenvalue is the square of the modulus of the amplitude for the state of the system to be in the corresponding eigenstate of $\hat{O}$. After the measurement, the system is in this eigenstate."

The fact that probabilities enter into the results of measurements has led to a lot of "vexed" discussion. Your first reaction might be "What's the fuss? After all, don't probabilities enter in classical physics too? If one tosses a coin isn't the result is randomly heads or tails with equal probability?" Well, is it *really* random? If one could measure with sufficient precision the initial momentum and angular momentum of the coin, and integrate the equations of motion for its trajectory, including the effects of air resistance, to sufficient accuracy then one would be able to compute, *with certainty*,

on which side it would land. The difficulty is that the coin toss has great sensitivity to the initial conditions, which means that if one changes the initial velocity by an immeasurably small amount the result changes. In other words, *for all practical purposes* (FAPP) a coin toss *is* random. Nonetheless, from a fundamental point of view it is not, since it is uniquely determined by the initial conditions. However, the situation in quantum mechanics is different since, as far as we know, probabilities *enter in a fundamental way.*

The most famous critic of probabilities being part of a fundamental theory of physics was Einstein, who had many discussions on the topic with Niels Bohr. As we shall see in our study of the EPR thought experiment in Chapter 7, despite Einstein's claim that "God doesn't play dice with the universe", quantum mechanics has been repeatedly vindicated.

We have said that after a measurement the system is left in eigenstate $|n\rangle$. Measurement therefore "projects" the initial state $|\psi\rangle$ on to $|n\rangle$. This is accomplished by the projection operator

$$\hat{P}_n = |n\rangle\langle n| \tag{3.65}$$

so

$$\hat{P}_n|\psi\rangle = |n\rangle\langle n|\psi\rangle, \tag{3.66}$$

(no sum on n). The sum of the projection operators must add to the identity, i.e.

$$\sum_n \hat{P}_n \equiv \sum_n |n\rangle\langle n| = \mathbb{1}. \tag{3.67}$$

The fact that $\sum_n |n\rangle\langle n|$ can be replaced by the identity is called a "completeness" relation.

Note that the state in Eq. (3.66) is not normalized. If we continue to follow the system after the measurement then we need to multiply the state by $1/|\langle n|\psi\rangle|$, so it is again correctly normalized and the sum of probabilities of results of a future measurement will add to unity. We note that something similar is also done in classical statistics. If we have a sequence of measurements, and we know the result of the first one, then we can determine the "conditional probability" of subsequent measurements, given the result of the first measurement, and these conditional probabilities add to unity. In effect, this is what is done by multiplying a state by a constant to get its norm back to 1 after a measurement. The resulting state will give the conditional probabilities for a subsequent measurement given the result of the first measurement.

Next a comment on notation. The convention is that the state $|0\rangle$ is an eigenstate of Z with eigenvalue $+1$ and $|1\rangle$ is eigenstate of Z with eigenvalue -1. One might have guessed that state $|1\rangle$ would be the state with eigenvalue $+1$ but this is not the convention that has been adopted.

Let's give a simple example of a measurement. Consider one qubit in state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and measure Z. The eigenstates of Z are $|0\rangle$ and $|1\rangle$ with eigenvalues $+1$ and -1 respectively. Hence the results of a measurement of Z are

$$\begin{aligned} +1, \qquad & \text{prob. } \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}, \\ -1, \qquad & \text{prob. } \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}. \end{aligned} \tag{3.68}$$

Now suppose that we measure X. The eigenstates of X are shown in Eqs. (2.28) and (2.30) to be $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. Hence $|\psi\rangle$ is the eigenstate with eigenvalue $+1$, so the result of the measurement of X is $+1$ with 100% probability. Similarly a measurement of X on state $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ would give -1 with probability 1.

We see that if the initial state is an eigenstate of the operator being measured, then the result will, with certainty, be the corresponding eigenvalue, and the state will remain unchanged after the

measurement. However, if this is not the case, i.e. if the initial state is in a superposition of eigenstates of the measurement operator, then (i) the result of the measurement will take one of several values with appropriate probabilities, and (ii) the measurement changes the state, leaving it in the eigenstate corresponding to the eigenvalue which is measured.

3.7 Statistics of Measurements

If we prepare many identical copies of the system and measure each of them what can we say about the statistics of the measured values λ_n, the eigenvalues of A. First of all, what would be the mean of the measurements $\langle A\rangle$? We have

$$\begin{aligned}\langle A\rangle &= \sum_n P(n)\lambda_n \\ &= \sum_n |\langle n|\psi\rangle|^2\lambda_n = \sum_n \langle\psi|n\rangle\lambda_n\langle n|\psi\rangle \\ &= \sum_n \langle\psi|A|n\rangle\langle n|\psi\rangle \\ &= \langle\psi|A|\psi\rangle\,. \end{aligned} \tag{3.69}$$

where use used Eq. (3.63) to get the second line, we used that $A|n\rangle = \lambda_n|n\rangle$ to get the third line, and Eq. (3.67) to get the last line. The final result, $\langle\psi|A|\psi\rangle$ is called the "expectation value" of A in state $|\psi\rangle$.

In addition to the average result we are also often interested in the *scatter* about the average. This is characterized by the standard deviation defined by

$$\Delta A = \left(\left\langle\, (A - \langle A\rangle\,)^2 \,\right\rangle\right)^{1/2}, \tag{3.70}$$

which is the root mean square deviation about the mean. It can be expressed in a slightly simpler form since

$$\begin{aligned}\left\langle\, (A - \langle A\rangle)^2 \,\right\rangle &= \langle A^2 - 2A\langle A\rangle + \langle A\rangle^2\rangle \\ &= \langle A^2\rangle - 2\langle A\rangle^2 + \langle A\rangle^2 \\ &= \langle A^2\rangle - \langle A\rangle^2, \end{aligned} \tag{3.71}$$

so

$$\Delta A = \left(\,\langle A^2\rangle - \langle A\rangle^2\,\right)^{1/2} \tag{3.72}$$

We will call ΔA the uncertainty in A.

Let's illustrate this with the example we considered just above, namely $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. If we measure Z we have

$$\langle Z\rangle = \langle\psi|Z|\psi\rangle = \frac{1}{2}\begin{pmatrix}1 & 1\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}1\\ 1\end{pmatrix} = 0. \tag{3.73}$$

This agrees with our previous discussion where we found $+1$ and -1 with equal probability. We also have $Z^2 = \mathbb{1}$ and, since the average of 1 is always 1,

$$\langle Z^2\rangle = 1, \tag{3.74}$$

so

$$\Delta Z = \left(\,\langle Z^2\rangle - \langle Z\rangle^2\,\right)^{1/2} = 1. \tag{3.75}$$

For a measurement of X we already showed that $|\psi\rangle$ is an eigenstate with eigenvalue 1 and so the measured value is always 1. If we use Eqs. (3.69) and (3.72), we obtain $\langle X\rangle = 1, \langle X^2\rangle = 1$, and so $\Delta X = 0$ as expected.

3.8 Composite Systems

So far, we have described states of just a single qubit. How should we describe states of the many qubits which we will need for a quantum computer? Suppose, as an example, we have two qubits A and B. We can label the states of qubit A by $|0_A\rangle$ and $|1_A\rangle$, and similarly the states of qubit B by $|0_B\rangle$ and $|1_B\rangle$. A state of both qubits is written as a "tensor product", also known as a "direct product", e.g. $|0_A\rangle \otimes |1_B\rangle$, which in this example indicates that qubit A is in state $|0\rangle$ and qubit B is in state $|1\rangle$.

This notation is heavy so we will usually write the same state more compactly as $|0_A\rangle|1_B\rangle$, or even more concisely as $|01\rangle$ provided a specification of the order of the qubits has been given. In this notation, the four possible states of two qubits are

$$|00\rangle, \quad |01\rangle, \quad |10\rangle, \quad |11\rangle. \tag{3.76}$$

Note that the label of each state is a number in binary notation from 0 to 3. This provides an even more compact notation, which is particularly convenient when the number of qubits is large, namely $|x\rangle_2$, where $x = 0, 1, 2$ or 3. It is necessary to indicate the number of qubits by a subscript on the bracket to avoid ambiguity. For example just writing a state as $|2\rangle$ we wouldn't know if it is state $|10\rangle$ for 2 qubits, or $|010\rangle$ for 3 qubits and so on. An exception to this will be states $|0\rangle$ and $|1\rangle$ (without subscript) which always refer to the 1-qubit basis states. The four states in Eq. (3.76) can therefore also be written as

$$|0\rangle_2, \quad |1\rangle_2, \quad |2\rangle_2, \quad |3\rangle_2. \tag{3.77}$$

Similarly for three qubits, we can specify the 8 possible states by $|x\rangle_3$ where $x = 0, 1, \cdots, 7$, and for n qubits the 2^n states are indicated by $|x\rangle_n$, where $x = 0, 1, \cdots, 2^n - 1$. We see that to use this convenient binary notation we need to label the states starting from 0 rather than 1. The last state then has label $2^n - 1$.

We need to be familiar with these ways of labeling multi-qubit states.

Next we discuss matrix representations of operators on multiple qubits, and we take as an example, the case of two qubits. An operator acting on the space of two qubits is a 4×4 matrix. We will write the four basis states as $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$. Consider an operator where X acts on the first (left hand) qubit and the identity $\mathbb{1}$ acts on the second (right hand) qubit. The 2-qubit operator is a tensor product of the 1-qubit operators, i.e. $X \otimes \mathbb{1}$. Its action on the four basis states is as follows:

$$\begin{aligned} X \otimes \mathbb{1}|00\rangle &= |10\rangle, \\ X \otimes \mathbb{1}|01\rangle &= |11\rangle, \\ X \otimes \mathbb{1}|10\rangle &= |00\rangle, \\ X \otimes \mathbb{1}|11\rangle &= |01\rangle, \end{aligned} \tag{3.78}$$

so its matrix representation is

$$X \otimes \mathbb{1} = \begin{array}{c} \\ \langle 00| \\ \langle 01| \\ \langle 10| \\ \langle 11| \end{array} \begin{array}{c} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array} = \begin{pmatrix} 0 & \mathbb{1} \\ \mathbb{1} & 0 \end{pmatrix}, \tag{3.79}$$

where in the last expression each entry is a 2×2 block. Note how this block structure reflects the operators in the tensor product on the left of the expression. The 2×2 block structure is that of X (the left hand operator) while each block is made up of the identity $\mathbb{1}$ (the right hand operator).

As a second example consider $X \otimes Z$. We have

$$\begin{aligned} X \otimes Z|00\rangle &= |10\rangle, \\ X \otimes Z|01\rangle &= -|11\rangle, \\ X \otimes Z|10\rangle &= |00\rangle, \\ X \otimes Z|11\rangle &= -|01\rangle, \end{aligned} \tag{3.80}$$

so its matrix representation is

$$X \otimes Z = \begin{array}{c} \\ \langle 00| \\ \langle 01| \\ \langle 10| \\ \langle 11| \end{array} \begin{array}{c} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \end{array} = \begin{pmatrix} 0 & Z \\ Z & 0 \end{pmatrix}. \tag{3.81}$$

Again notice how the block structure in the last expression reflects the operators in the tensor product.

3.9 Generalized Born Rule

In Sec. 3.6 we gave the standard physics text book discussion of measurement in quantum mechanics. For quantum computing we need to extend this to deal with situations involving multiple qubits where we measure only *some* of the qubits and we need to know the state of the remaining qubits after the measurement. As a simple example, suppose we have 2 qubits A and B, in a state

$$|\psi\rangle = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle, \tag{3.82}$$

where the left qubit is A and the right qubit is B. Because the state has to be normalized we need $|a_0|^2 + |a_1|^2 + |a_2|^2 + |a_3|^2 = 1$. Suppose we measure Z for qubit-A, the left qubit. We want to know what are the possible measurement results, what are the probabilities of the different results, and, for each case, in what state is qubit B after the measurement.

We will rewrite Eq. (3.82), grouping together all the terms where qubit A is $|0\rangle$ (more generally an eigenstate of the operator acting on A), and all the terms where qubit A is $|1\rangle$ (the other eigenstate). The terms involving qubit A in state $|0\rangle$ are $a_0|00\rangle + a_1|01\rangle$. We write this as

$$a_0|00\rangle + a_1|01\rangle = \alpha_0|0_A\rangle \left(\frac{a_0}{\alpha_0}|0_B\rangle + \frac{a_1}{\alpha_0}|1_B\rangle \right) = \alpha_0|0_A\rangle|\phi_{0,B}\rangle, \tag{3.83}$$

where

$$|\alpha_0|^2 = |a_0|^2 + |a_1|^2 \tag{3.84}$$

and

$$|\phi_{0,B}\rangle = \frac{1}{\alpha_0}\left(a_0|0_B\rangle + a_1|1_B\rangle\right), \tag{3.85}$$

is a *normalized* state for qubit B. Similarly

$$a_2|10\rangle + a_3|11\rangle = \alpha_1|1_A\rangle \left(\frac{a_2}{\alpha_1}|0_B\rangle + \frac{a_3}{\alpha_1}|1_B\rangle \right) = \alpha_1|1_A\rangle|\phi_{1,B}\rangle, \tag{3.86}$$

where

$$|\alpha_1|^2 = |a_2|^2 + |a_3|^2 \tag{3.87}$$

and

$$|\phi_{1,B}\rangle = \frac{1}{\alpha_1}\left(a_2|0_B\rangle + a_3|1_B\rangle\right), \tag{3.88}$$

is normalized.

Combining we get

$$|\psi\rangle = \alpha_0|0_A\rangle|\phi_{0,B}\rangle + \alpha_1|1_A\rangle|\phi_{1,B}\rangle, \tag{3.89}$$

where we emphasize that all the states in this expression are normalized.

The inner product of $|\phi_0\rangle$ and $|\phi_1\rangle$ is

$$\langle\phi_{0,B}|\phi_{1,B}\rangle = \frac{a_0^\star a_2 + a_1^\star a_3}{\sqrt{|a_0|^2+|a_1|^2}\,\sqrt{|a_2|^2+|a_3|^2}}, \tag{3.90}$$

and there is no reason for this to be zero in general. Hence $|\phi_{0,B}\rangle$ and $|\phi_{1,B}\rangle$ are not necessarily orthogonal.

The natural extension of the Born rule, called the "generalized Born" rule, is that, when the two qubits are in the state given in Eq. (3.89), the possible results of the measurement of Z on qubit A, are

$$\begin{aligned} &\text{result } +1, \quad \text{probability } |\alpha_0|^2, \quad \text{final state } \ |0_A\rangle|\phi_{0,B}\rangle, \\ &\text{result } -1, \quad \text{probability } |\alpha_1|^2, \quad \text{final state } \ |1_A\rangle|\phi_{1,B}\rangle. \end{aligned} \tag{3.91}$$

It is straightforward to generalize this result to an arbitrary situation in which there are $n+m$ qubits, n of which are measured and we want to know the possible final states of the remaining m qubits after the measurement, and to arbitrary measurement operators.

3.10 The Uncertainty Principle

Now we come to a key concept in quantum mechanics, the *uncertainty principle*. We shall see that some variables are incompatible with each other, which means that one can not have definite values for both of them in *any* state. The important quantity to see if two operators, A and B say, are compatible is their commutator

$$[A,B] \equiv AB - BA. \tag{3.92}$$

If $[A,B] \neq 0$ then it is shown in linear algebra texts that A and B have different eigenvectors. We have already noted that we only get a definite value for some operator if the state is in an eigenstate of that operator. Hence, if $[A,B] \neq 0$, so A and B have different eigenvectors, there is no state which will give a definite value for both of them.

As an example of a commutator consider X and Z. We have

$$[Z,X] = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} - \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} = \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix} - \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix} = 2\begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix} = 2iY, \tag{3.93}$$

where Y is defined in Eq. (2.23). Since the commutator is non-zero it is impossible to find a state which is simultaneous eigenstate of both X and Z and so either ΔX or ΔZ, or both, must be non-zero.

An important inequality involving the uncertainties ΔA and ΔB of two operators in a state $|\psi\rangle$ is

$$(\Delta A\,\Delta B)_\psi \geq \frac{1}{2}\left|\langle[A,B]\rangle_\psi\right|, \tag{3.94}$$

which is known as the Heisenberg uncertainty principle. We shall not prove this result. The most famous case of the uncertainty principle is for $A = x$, the position of a particle, and $B = p$, its momentum, for which the commutator is a constant[3] $i\hbar$ so

$$\Delta x \Delta p \geq \frac{\hbar}{2}. \tag{3.95}$$

However, this particular version of the uncertainty principle does not play a role in quantum computing which is concerned with (discrete) 2-state systems, rather than (continuous) trajectories of particles.

3.11 Time Evolution of Quantum States

So far, we have described fixed quantum states. Now we need to discuss how they evolve with time. If the state at an initial time is $|\psi\rangle$ and the state at a later time is $|\psi'\rangle$, then, according to quantum mechanics, that there is a linear relation between the two, so

$$|\psi'\rangle = U|\psi\rangle, \tag{3.96}$$

for some linear operator U. The normalization condition must be preserved so $\langle\psi'|\psi'\rangle = \langle\psi|\psi\rangle = 1$. This provides a constraint on the form of U as we will now show. The equation corresponding to Eq. (3.96) for the dual vector $\langle\psi'|$ is

$$\langle\psi'| = \langle\psi|U^\dagger. \tag{3.97}$$

To see this compare Eqs. (3.38) and (3.34) and note that $\left(A^\dagger\right)^\dagger = A$. Combining Eqs. (3.96) and (3.97) we find

$$\langle\psi'|\psi'\rangle = \langle\psi|U^\dagger U|\psi\rangle. \tag{3.98}$$

Since we must have $\langle\psi'|\psi'\rangle = \langle\psi|\psi\rangle$ $(= 1)$ for any initial state $|\psi\rangle$ it follows that

$$U^\dagger U = \mathbb{1}, \tag{3.99}$$

so U has to be unitary.

In quantum computing we change the state of the qubits by a sequence of *discrete* unitary transformations. Note that for a unitary operator $U^{-1} = U^\dagger$, and $U^\dagger$ is well defined, so the inverse transformation, which acts on the final state and converts it to the initial state, exists. Thus quantum transformations are *reversible*. The exception is measurement, in which the quantum system is coupled to a macroscopic, external apparatus which leads to an irreversible change. As we shall see, standard classical gates which manipulate the bits in a classical computer are irreversible. The necessity of doing reversible operations in a quantum computer will be a major difference compared with a classical computer.

In a quantum computer, as noted above, we act on the qubits with a series of discrete unitary operations. These come from the effect of some operation acting for a finite amount of time.

Microscopically, quantum states evolve *continuously* with time, and we will finish this chapter with a brief discussion of continuous time evolution in quantum mechanics (even though it will not be needed in the rest of the course). Time evolution is determined by the Hamiltonian (energy), $\mathcal{H}$ a Hermitian operator, according to Ansatz 3:

Ansatz 3: The time dependence of a state is given by Schrödinger's equation

$$i\hbar \frac{\partial}{\partial t}|\psi(t)\rangle = \mathcal{H}\,|\psi(t)\rangle. \tag{3.100}$$

[3] $\hbar$ is Planck's constant divided by 2π. It is of paramount importance in physics but does not play a role in the theory of quantum computation.

Assuming that $\mathcal{H}$ does not change with time, we can integrate Eq. (3.100) to get

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle, \tag{3.101}$$

where

$$U(t) = e^{-i\mathcal{H}t/\hbar}. \tag{3.102}$$

Since $\mathcal{H}$ is Hermitian we can show that U is unitary by the following argument. To get the adjoint of U we take its complex conjugate and replace any operators in the expression for U by their adjoint. Since $\mathcal{H}$ is self-adjoint (Hermitian) we have

$$U^\dagger(t) = e^{i\mathcal{H}t/\hbar}, \tag{3.103}$$

from which one sees that

$$U^\dagger(t)U(t) = e^{i\mathcal{H}t/\hbar}e^{-i\mathcal{H}t/\hbar} = e^{i(\mathcal{H}t-\mathcal{H}t)/\hbar} = \mathbb{1}, \tag{3.104}$$

so U is unitary as required. Note that if we have operators in exponentials which don't commute, we can't manipulate them as we do with ordinary numbers. For example $e^A e^B$ does *not* equal e^{A+B} unless $[A, B] = 0$. However, here both A and B are proportional to $\mathcal{H}$ which commutes with itself, so combining the exponentials as done in Eq. (3.104) *is* valid.

Chapter 4

General state of a qubit, no-cloning theorem, entanglement and Bell states

4.1 General qubit states

As already discussed in Sec. 2.5, the following 2×2 matrices, called Pauli matrices, acting on the states of a single qubit will be important in the rest of the course:

$$X \equiv \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{4.1a}$$

$$Y \equiv \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{4.1b}$$

$$Z \equiv \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{4.1c}$$

In the physics literature the notation used is σ_x etc., but in this course we shall use the quantum computing notation: X, Y, and Z. As shown in Sec. 2.4, an arbitrary 2×2 matrix can be written as a linear combination of the three Pauli matrices plus the 2×2 identity matrix. These matrices are Hermitian, and have eigenvalues ± 1, see Sec. 2.4.

If the qubit is the spin of an electron, then the eigenstate with $Z = 1$ has spin along the $+z$ direction, and analogously the eigenstate with $Y = 1$ has spin along the $+y$ direction, and the eigenstate with $X = 1$ has spin along the $+x$ direction. Also, the eigenstate with $Z = -1$ has the spin pointing in the $-z$ direction, and analogously for $X = -1$ and $Y = -1$.

How can we specify a *general* state of a qubit? First of all, how many parameters do we need to specify a general state? A qubit vector has two complex components making a total of four. However, one of these can be eliminated because the state must be normalized, and another can be eliminated because an overall phase is unimportant. This leaves two parameters necessary to describe a general qubit state.

We shall see that we can conveniently take these two parameters to be the two angles which describe a direction in space in spherical polar coordinates. To see this we compute the eigenstates for the spin of an electron aligned along a general direction with polar angle θ and azimuthal angle ϕ, which describe a unit vector $\hat{n}$ where

$$\hat{n} = (\sin\theta\, \cos\phi,\ \sin\theta\, \sin\phi,\ \cos\theta), \tag{4.2}$$

so $n_x = \sin\theta\, \cos\phi$ etc. In other words we compute the eigenvalues and eigenvectors of $\vec{\sigma} \cdot \hat{n}$. We have

$$\vec{\sigma} \cdot \hat{n} = \begin{pmatrix} n_z & n_x - i n_y \\ n_x + i n_y & -n_z \end{pmatrix} \tag{4.3}$$

so the eigenvalues are given by

$$\begin{vmatrix} n_z - \lambda & n_x - in_y \\ n_x + in_y & -n_z - \lambda \end{vmatrix} = 0. \tag{4.4}$$

Expanding the determinant, and using that $n_x^2 + n_y^2 + n_z^2 = 1$, we find the eigenvalues to be

$$\lambda = \pm 1. \tag{4.5}$$

Thus, the eigenvalues are not only ± 1 when measured along the Cartesian directions, but take the same values along *any* direction.

Next we look at the eigenvectors. First the eigenvector for eigenvalue $+1$ is

$$|0_{\hat{n}}\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \tag{4.6}$$

where

$$\begin{pmatrix} \cos\theta & \sin\theta\, e^{-i\phi} \\ \sin\theta\, e^{i\phi} & -\cos\theta \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}, \tag{4.7}$$

where we used Eqs. (4.2) and (4.3). Writing out the two equations we get

$$\sin\theta\, e^{-i\phi}\, b = a(1 - \cos\theta), \tag{4.8a}$$

$$\sin\theta\, e^{i\phi}\, a = b(1 + \cos\theta). \tag{4.8b}$$

Both these equations are satisfied by

$$b\, \cos\tfrac{\theta}{2} = a\, e^{i\phi} \sin\tfrac{\theta}{2}, \tag{4.9}$$

(recall the expressions for sines and cosines of double angles). We require the state to be normalized, i.e. $|a|^2 + |b|^2 = 1$, so we get

$$|0_{\hat{n}}\rangle = \begin{pmatrix} \cos\frac{\theta}{2} \\ e^{i\phi}\, \sin\frac{\theta}{2} \end{pmatrix}, \tag{4.10}$$

or equivalently, in Dirac notation,

$$|0_{\hat{n}}\rangle = \cos\tfrac{\theta}{2}\, |0\rangle + e^{i\phi}\, \sin\tfrac{\theta}{2}\, |1\rangle\,. \tag{4.11a}$$

A similar calculation gives the eigenstate corresponding to eigenvalue -1 to be

$$|1_{\hat{n}}\rangle = -\sin\tfrac{\theta}{2}\, |0\rangle + e^{i\phi}\, \cos\tfrac{\theta}{2}\, |1\rangle\,. \tag{4.11b}$$

It is straightforward to see that the states in Eqs. (4.11) are normalized, i.e.

$$\langle 0_{\hat{n}} | 0_{\hat{n}} \rangle = 1, \qquad \langle 1_{\hat{n}} | 1_{\hat{n}} \rangle = 1, \tag{4.12}$$

and are mutually orthogonal

$$\langle 0_{\hat{n}} | 1_{\hat{n}} \rangle = 0. \tag{4.13}$$

Note that we can always multiply eigenstates by an arbitrary phase factor so you might see expressions for these eigenstates which *look* different from Eqs. (4.11a) and (4.11b), but which are actually equivalent.

If we consider a point on a unit sphere (often called the Bloch sphere) with polar angles θ and ϕ, then the eigenstate of spin in that direction with eigenvalue $+1$ is given by Eq. (4.11a), see Fig. 4.1. Similarly, the eigenstate with eigenvalue -1 is given by Eq. (4.11b), which corresponds to the antipodal point where $\theta \to \pi - \theta, \phi \to \phi + \pi$.

It is useful to consider four special cases of Eqs. (4.11):

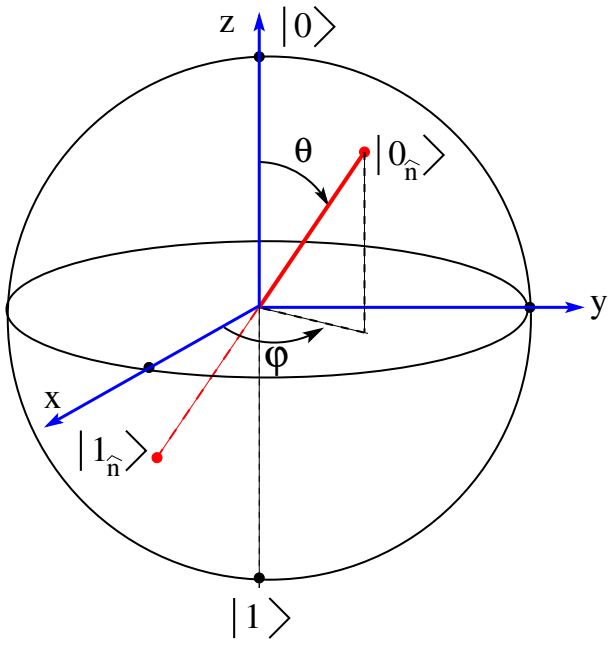


Figure 4.1: The Bloch sphere.

(i) $(\theta = \phi = 0)$, the z direction. Clearly $|0_{\hat{z}}\rangle = |0\rangle$ and $|1_{\hat{z}}\rangle = |1\rangle$ as required.

(ii) $(\theta = \pi/2, \phi = 0)$, the x direction:

$$|0_{\hat{x}}\rangle = \frac{1}{\sqrt{2}}\left(\,|0\rangle + |1\rangle\,\right) = |+\rangle, \tag{4.14}$$

$$|1_{\hat{x}}\rangle = \frac{1}{\sqrt{2}}\left(\,-|0\rangle + |1\rangle\,\right) = -|-\rangle. \tag{4.15}$$

These are the eigenstates of X as expected. ($|1_{\hat{x}}\rangle$ has the opposite sign to the conventionally defined state $|-\rangle$, but the overall sign of a state is of no importance.)

(iii) (θ arbitrary $\phi = 0$), a direction $\hat{n}$, in the x-z plane at an angle θ to the z axis:

$$|0_{\hat{n}}\rangle = \cos\tfrac{\theta}{2}\,|0\rangle + \sin\tfrac{\theta}{2}\,|1\rangle \qquad (\phi = 0), \tag{4.16}$$

$$|1_{\hat{n}}\rangle = -\sin\tfrac{\theta}{2}\,|0\rangle + \cos\tfrac{\theta}{2}\,|1\rangle. \tag{4.17}$$

(iv) $(\theta = \pi/2, \phi = \pi/2)$, the y direction:

$$|0_{\hat{y}}\rangle = \frac{1}{\sqrt{2}}\left(\,|0\rangle + i\,|1\rangle\,\right), \tag{4.18}$$

$$|1_{\hat{y}}\rangle = \frac{1}{\sqrt{2}}\left(\,-|0\rangle + i\,|1\rangle\,\right). \tag{4.19}$$

These are the eigenstates of Y as expected.

Even if the qubit is not an electron spin, Eq. (4.11a) provides a convenient description of an arbitrary qubit state. As shown above, it is an $+1$ eigenstate of $\vec{\sigma}\cdot\hat{n}$, where n is in direction (θ, ϕ) with θ and ϕ the polar and azimuthal angles of a point on a unit sphere, see Fig. 4.1.

We mentioned in Sec. 1.4 that for certain quantum protocols photons make good qubits, with the state of the qubit being characterized by its polarization (the direction and phase of the electric field). Using Eqs. (1.7)–(1.10) and (4.11a), we find that the electric field of a photon propagating in the $\hat{z}$ direction, corresponding to a qubit $|0_{\hat{n}}\rangle$ specified by angles θ and ϕ, is given by

$$\vec{E} = \Re\left[E_0 \cos(\theta/2) e^{-i(kz-\omega t)}\,\hat{x} + E_0 \sin(\theta/2) e^{i\phi}\, e^{-i(kz-\omega t)}\,\hat{y}\right], \tag{4.20}$$

where $\Re$ means real part, so

$$\begin{aligned} E_x &= E_0 \cos(\theta/2)\, \cos(\omega t - kz), \\ E_y &= E_0 \sin(\theta/2)\, \cos(\omega t - kz - \phi). \end{aligned} \tag{4.21}$$

Hence one can create an arbitrary qubit state by an appropriate choice of photon polarization. The polarization states for a photon for each of the four special cases given above are:

(i) ($\theta = \phi = 0$), i.e. $|0_{\hat{z}}\rangle \equiv |0\rangle$. Linearly polarized along $\hat{x}$. (The photon corresponding to $|1\rangle$ is polarized along $\hat{y}$.)

(ii) ($\theta = \pi/2, \phi = 0$), i.e. $|0_{\hat{x}}\rangle$. Linearly polarized along a diagonal direction. (The photon corresponding to $|1_{\hat{x}}\rangle$ is polarized along the other diagonal direction.)

(iii) (θ arbitrary, $\phi = 0$). Linearly polarized with the polarization direction at an angle $\theta/2$ to the x-axis.

(iv) ($\theta = \pi/2, \phi = \pi/2$), i.e. $|0_{\hat{y}}\rangle$. Circularly polarized[1] with the $\vec{E}$ vector rotating in a particular sense as a function of time. (The photon corresponding to $|1_{\hat{y}}\rangle$ is is circularly polarized with the $\vec{E}$ vector rotating in the opposite sense.)

4.2 No-cloning theorem

A classical bit, 0 or 1, can be copied, i.e. cloned. You just observe it and create another one. With qubits, however, it turns out to be not possible to clone an arbitrary, unknown state. This is called the "no-cloning theorem". It imposes an important limitation on our ability to manipulate quantum states. We now give the simple derivation of this important result.

Consider the general qubit state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \text{where } |\alpha|^2 + |\beta|^2 = 1. \tag{4.22}$$

We can't determine the state by measuring it because a measurement gives $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$, i.e. it destroys the superposition.

Can we clone the state *without* measuring it? If so, there must be a unitary operator U which acts on $|\psi\rangle$ and an ancilla qubit, which is initialized to $|0\rangle$ say, and clones $|\psi\rangle$ as follows:

$$U|\psi\rangle\,|0\rangle = |\psi\rangle\,|\psi\rangle. \tag{4.23}$$

We shall see that no such operator can exist, because operators in quantum mechanics are *linear*. Suppose that

$$\begin{aligned} U|\psi\rangle\,|0\rangle &= |\psi\rangle\,|\psi\rangle, \\ U|\phi\rangle\,|0\rangle &= |\phi\rangle\,|\phi\rangle. \end{aligned} \tag{4.24}$$

[1] Recall that $\cos(x - \pi/2) = \sin(x)$

Then, by linearity,

$$U\,(\,\alpha|\psi\rangle + \beta|\phi\rangle\,)\,|0\rangle = \alpha|\psi\rangle\,|\psi\rangle + \beta|\phi\rangle\,|\phi\rangle\,. \tag{4.25}$$

However, this is not a clone of $\alpha|\psi\rangle + \beta|\phi\rangle$ which would be

$$(\,\alpha|\psi\rangle + \beta|\phi\rangle\,)\,(\,\alpha|\psi\rangle + \beta|\phi\rangle\,) = \alpha^2|\psi\rangle\,|\psi\rangle + \alpha\beta|\psi\rangle\,|\phi\rangle + \alpha\beta|\phi\rangle\,|\psi\rangle + \beta^2|\phi\rangle\,|\phi\rangle. \tag{4.26}$$

There is an inconsistency so a unitary operator U for cloning does not exist.

The no-cloning theorem will be an important limitation when designing quantum algorithms.

4.3 Entanglement and Bell states

A striking aspect of quantum states of more than one qubit, which seems mysterious and plays a crucial role in quantum algorithms, is called "*entanglement*". Here we will illustrate this concept for the simplest case of two qubits.

Let's suppose that the first qubit is in state $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and the second qubit is in state $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$. The state of the two-qubit system is the tensor product

$$|\psi_1\rangle \otimes |\psi_2\rangle = \begin{pmatrix}\alpha_1\\ \beta 1\end{pmatrix} \otimes \begin{pmatrix}\alpha_2\\ \beta 2\end{pmatrix} = \begin{pmatrix}\alpha_1\alpha_2\\ \alpha_1\beta_2\\ \beta_1\alpha_2\\ \beta_1\beta_2\end{pmatrix}. \tag{4.27}$$

This is called a *product state*.

However, a general qubit state is *not* a product state. It can be written as

$$|\phi\rangle_2 = c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle, \tag{4.28}$$

or equivalently as

$$|\phi\rangle_2 = c_0|0\rangle_2 + c_1|1\rangle_2 + c_2|2\rangle + c_3|3\rangle = \sum_{x=0}^{3} c_x|x\rangle_2, \tag{4.29}$$

where the notation $|x\rangle_2$ indicates that we have a state of two qubits and the states of the individual qubits are represented by the bits of the integer x.

The product state has

$$c_0 = \alpha_1\alpha_2, \qquad c_1 = \alpha_1\beta_2, \qquad c_2 = \beta_1\alpha_2, \qquad c_3 = \beta_1\beta_2, \tag{4.30}$$

and so satisfies

$$c_0c_3 = c_1c_2. \tag{4.31}$$

This is the condition for a 2-qubit state to be a product state. States which do not have this property are said to be *entangled*.

The most-studied entangled states are so-called Bell states which involve two qubits. They are named in honor of the physicist John Bell whose inequalities (to be discussed later) demonstrated that the description of nature provided by quantum mechanics is fundamentally different from the classical description. The Bell states are defined by

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\,(\,|00\rangle + |11\rangle\,)\,, \tag{4.32a}$$

$$|\beta_{01}\rangle = \frac{1}{\sqrt{2}}\,(\,|01\rangle + |10\rangle\,)\,, \tag{4.32b}$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}\,(\,|00\rangle - |11\rangle\,)\,, \tag{4.32c}$$

$$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}\,(\,|01\rangle - |10\rangle\,)\,. \tag{4.32d}$$

These four equations can be combined as follows:

$$|\beta_{xy}\rangle = \frac{1}{\sqrt{2}} \left(|0y\rangle + (-1)^x |1\overline{y}\rangle \right) , \tag{4.33}$$

where $\overline{y}$ is the complement of y, i.e. $\overline{y} = 1 - y$. The Bell states are clearly entangled.

There are correlations between the qubits in the Bell states (quite generally between the qubits in entangled states). For example, if we consider $|\beta_{00}\rangle$ and do a measurement on qubit 1, then a measurement of qubit 2 (if performed) would find the same result with 100% probability. We will discuss quantum correlations in entangled states in some detail in Chapter 7 when we investigate the claim of Einstein-Podolsky-Rosen (EPR) that quantum mechanics is incomplete.

For the case of two qubits, Eq. (4.31) is a convenient way to test if a state is a product state or entangled. In a more general case where we have, say, $n = n_A + n_B$ qubits, we may want to know whether a partition of the system into the two subsystems A, with n_A qubits, and B, with n_B qubits, gives a product state, i.e. if

$$|\psi\rangle_n = |\psi_A\rangle_{n_A} \otimes |\psi_B\rangle_{n_B}, \tag{4.34}$$

or whether the state is entangled with respect to this partition. In the case with more than $n = 2$ qubits, there is no simple expression analogous to Eq. (4.31) for the 2^n coefficients $c_x, (x = 0, 1, \cdots, 2^n - 1)$, which indicates a product state. Instead, a systematic way to investigate whether such a state is entangled or a product state is to use the density matrix, discussed in Chapter 5.

Appendix

4.A Angular Momentum Eigenstates

Physics students learn about quantum states which are eigenstates of angular momentum. This appendix relates Bell states to spin angular momentum eigenstates of two electrons. It is intended for physics students and is not essential reading for students of other disciplines.

The spin of an electron $\vec{s}$ is given by

$$\vec{s} = \frac{\hbar}{2}\vec{\sigma}, \tag{4.35}$$

where $\hbar$ is Planck's constant divided by 2π and the Pauli operators $\vec{\sigma}$ are defined in Eqs. (4.1).

In general, spin angular momentum states, $|S, m\rangle$, are specified by two quantum numbers S and m. The total spin quantum number S is defined by

$$s_x^2 + s_y^2 + s_z^2 = \hbar^2 S(S+1), \tag{4.36}$$

so $|S, m\rangle$ is an eigenvalue of $(\vec{s})^2$ with eigenvalue $\hbar^2 S(S+1)$. The quantum number m is defined such that $|S, m\rangle$ is an also eigenstate of s_z with eigenvalue $\hbar m$, where m ranges from $-S$ to S in integer steps (so there are $2S + 1$ values of m for a given S). Thus the spin of an electron has $S = 1/2$, and its two basis states are $|S = 1/2,\, m = 1/2\rangle$ and $|S = 1/2,\, m = -1/2\rangle$, which are often written as $|\uparrow\rangle$ and $|\downarrow\rangle$ respectively. The latter notation indicates that one thinks of these two states as spin "up" and spin "down". By convention, the correspondence between the basis states of the electron spin in physics, $|\uparrow\rangle$ and $|\downarrow\rangle$, and the computational basis states in quantum computer science, $|0\rangle$ and $|1\rangle$, is taken to be

$$|\uparrow\rangle \equiv |0\rangle, \quad |\downarrow\rangle \equiv |1\rangle. \tag{4.37}$$

If we have two particles with total spin quantum numbers S_1 and S_2 then, as shown in textbooks on quantum mechanics [Gri05], the "vector rule" for addition of angular momentum states that the total spin quantum number of the combined system, $S_{\rm tot}$, takes integer values between $S_1 + S_2$ and

$|S_1 - S_2|$. Thus, two electrons can have combined total spin quantum number $S_{\rm tot} = 1$ (for which there are 3 values of $m_{\rm tot}$, namely $1, 0$ and -1, and $S_{\rm tot} = 0$ (for which there is only one value of $m_{\rm tot}$, namely 0). These are called "triplet" and "singlet" states respectively. Note that the total number of states works out right since there are $2^2 = 4$ states out of which 3 have $S_{\rm tot} = 1$ and 1 has $S_{\rm tot} = 0$, (i.e. $2 \times 2 = 3 + 1$).

It is also shown in the quantum mechanics textbooks that the states of two spin-1/2 particles with specified values of $S_{\rm tot}$ and $m_{\rm tot}$ are given by

$$|S_{\rm tot} = 1, m_{\rm tot} = 1\rangle \quad = |\uparrow\uparrow\rangle \quad \equiv |00\rangle, \tag{4.38a}$$

$$|S_{\rm tot} = 1, m_{\rm tot} = 0\rangle \quad = \frac{1}{\sqrt{2}}\left(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle\right) \quad \equiv \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right), \tag{4.38b}$$

$$|S_{\rm tot} = 1, m_{\rm tot} = -1\rangle \quad = |\downarrow\downarrow\rangle \quad \equiv |11\rangle, \tag{4.38c}$$

$$|S_{\rm tot} = 0, m_{\rm tot} = 0\rangle \quad = \frac{1}{\sqrt{2}}\left(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle\right) \quad \equiv \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right). \tag{4.38d}$$

Eqs. (4.38a)–(4.38c) are the triplet states while Eq. (4.38d) is the singlet state.

Comparing with Eqs. (4.32) we see that

$$|S_{\rm tot} = 1, m_{\rm tot} = 1\rangle \quad = \frac{1}{\sqrt{2}}\left(|\beta_{00}\rangle + |\beta_{10}\rangle\right), \tag{4.39a}$$

$$|S_{\rm tot} = 1, m_{\rm tot} = 0\rangle \quad = |\beta_{01}\rangle, \tag{4.39b}$$

$$|S_{\rm tot} = 1, m_{\rm tot} = -1\rangle = \frac{1}{\sqrt{2}}\left(|\beta_{00}\rangle - |\beta_{10}\rangle\right), \tag{4.39c}$$

$$|S_{\rm tot} = 0, m_{\rm tot} = 0\rangle \quad = |\beta_{11}\rangle, \tag{4.39d}$$

Equations (4.39) connect Bell states and angular momentum states, while Eqs. (4.38) connect computational basis states and angular momentum states.

In this chapter we have encountered three sets of states which can describe 2 qubits:

- the computational basis states $|xy\rangle$,
- the Bell states $|\beta_{xy}\rangle$, and
- the angular momentum states $|S_{\rm tot},\, m_{\rm tot}\rangle$.

Each of these forms a basis set. In quantum computing we generally use computational basis states but sometimes the Bell basis will be useful. However, there does not seem to be a use for angular momentum basis states in quantum computing.

Chapter 5

The Density Matrix

5.1 Introduction

We will be interested in situations where a system is in contact with another, possibly much larger, system. Let's call the system of interest *sub*system A, and denote the other system by *sub*system B. We use the word "*sub*system" for A and B, since we now consider them as the two parts of the combined AB *system*. We want to describe the properties of subsystem A without explicitly including the degrees of freedom of subsystem B. This is accomplished by the "density matrix". Two situations where the density matrix is useful are:

- To determine whether a state is a product state or entangled with respect to a partition of the system into two subsystems.
- Quantum computers, where A is some qubits of the computer and B is the environment which inevitably couples to the computational qubits. The environment is very complicated with a huge (essentially infinite) number of degrees of freedom, so we cannot include it explicitly and we need a description involving just the degrees of freedom of A, in which the effects of the environment have been averaged over in some sense. This description is provided by the density matrix. In practice, approximations will have to be made to determine it. We will discuss the effects of the environment on a quantum computer in the Chapter 19.

For further reading on the density matrix see Refs. [NC00, Vat16, RP14].

5.2 Definition of the Density Matrix

To become familiar with the notation in a gentle way we first consider the density matrix of a system in a well-defined quantum state. This is not terribly useful in itself, and will just be a rewriting of results we have already obtained, but doing this will help us understand the much more useful case of the density matrix of a subsystem A, say, coupled to another system B, such that the total system $A \otimes B$ is in a well defined quantum state, but this is entangled with respect to the A-B subdivision so neither A nor B are in a well defined state.

5.2.1 Density matrix of a system in a well defined state

Consider, then, a quantum system in a well defined quantum state, $|\psi\rangle$. We *define* its density matrix ρ by the outer product

$$\rho = |\psi\rangle\langle\psi|. \tag{5.1}$$

We will understand the reason for this definition as we go along.[1] The matrix elements of ρ are

$$\langle n|\rho|m\rangle = \langle n|\psi\rangle\langle\psi|m\rangle. \tag{5.2}$$

Note that its diagonal elements are $|\langle n|\psi\rangle|^2$ which are the probabilities, P_n, of a measurement finding the system in state $|n\rangle$. Since probabilities add up to one, the trace (sum of diagonal elements) must satisfy

$$\mathrm{Tr}\,\rho = 1. \tag{5.3}$$

This will turn out to be a general property of a density matrix.

We shall now show that expectation values of operators in state $|\psi\rangle$ can be expressed in terms of ρ. We showed in Eq. (3.69) that the expectation value of an operator $\hat{O}$ is given by

$$\langle\hat{O}\rangle = \langle\psi|\hat{O}|\psi\rangle. \tag{5.4}$$

This can be re-expressed in terms of the density matrix ρ since

$$\begin{aligned}\langle\psi|\hat{O}|\psi\rangle &= \sum_m \langle\psi|\hat{O}|m\rangle\langle m|\psi\rangle \\ &= \sum_m \langle m|\psi\rangle\langle\psi|\hat{O}|m\rangle \\ &= \sum_m \langle m|\rho\,\hat{O}|m\rangle, \\ &= \mathrm{Tr}\,(\rho\,\hat{O}),\end{aligned} \tag{5.5}$$

where we used Eq. (5.1) and the completeness relation $\sum_n |n\rangle\langle n| = \mathbb{1}$, the identity. Hence expectation values can be obtained directly from the density matrix.

5.2.2 Density matrix of a subsystem when the combined system is in a well defined state

Here, and in the rest of this chapter, we will consider a system composed of two subsystems A and B, such that the combined system is in a single quantum state. In general, subsystems A and B will be entangled, so neither subsystem is in a well defined state. We will only be interested in one of the subsystems, A say, and would like a description in terms of just the states of A. This is where the density matrix becomes very useful.

As in the previous subsection, the density matrix of the whole system is given by

$$\rho^{AB} = |\psi_{AB}\rangle\langle\psi_{AB}|. \tag{5.6}$$

This is a matrix involving the states of both A and B. We shall now show that information about averages of the A degrees of freedom can be obtained without explicitly considering the B degrees of freedom from the density matrix ρ^A where[2]

$$\rho^A = \mathrm{Tr}_B\,\rho^{AB} = \sum_n \langle n_B|\psi_{AB}\rangle\langle\psi_{AB}|n_B\rangle, \tag{5.7}$$

which is a matrix in the space of the states of A only. We say that we have "traced out" the B states.

[1] In standard linear algebra notation, $\rho_{nm} = c_n c_m^\star$ and $\rho_{mn} = c_m c_n^\star = (\rho_{nm})^\star$, so ρ is Hermitian.

[2] As stated above, in this chapter we assume that the combined AB system is in single quantum state. If, instead, the combined system is itself described by a non-trivial density matrix ρ^{AB}, then the reduced density matrix for subsystem A is still given by $\rho_A = \mathrm{Tr}_B\,\rho^{AB}$ but ρ^{AB} is no longer given by Eq. (5.6).

The state $|\psi_{AB}\rangle$ can be expressed in terms of basis states. In the Dirac notation this has the rather cumbersome form

$$|\psi_{AB}\rangle = \sum_{i=1}^{N_A}\sum_{n=1}^{N_B} |i_A\rangle|n_B\rangle\langle i_A|\langle n_B|\psi_{AB}\rangle. \tag{5.8}$$

Here $N_A(=2^{n_A})$ is the number of states of subsystem A, $N_B(=2^{n_B})$ is the number of states of subsystem B, the $|i_A\rangle$ are a basis for A, and the $|n_B\rangle$ are a basis for B. At this point I prefer to use standard matrix notation with indices, rather than the Dirac notation, writing Eq. (5.8) as

$$|\psi_{AB}\rangle = \sum_{i=1}^{N_A}\sum_{n=1}^{N_B} c_{in}\,|i_A\rangle\,|n_B\rangle. \tag{5.9}$$

From Eqs. (5.7) and (5.9), the matrix elements of ρ^A are given in terms of the amplitudes c_{in} by[3]

$$\langle i|\rho^A|i'\rangle = \sum_n c_{in}\,c^*_{i'n}, \tag{5.10}$$

omitting for conciseness the label A on $|i_A\rangle$ and $|i'_A\rangle$ when there is no ambiguity. Because $|\psi_{AB}\rangle$ is normalized we have

$$\mathrm{Tr}\,\rho^A = \sum_{i,n} |c_{in}|^2 = 1. \tag{5.11}$$

As discussed earlier, this condition is a general property of density matrices.

We want to compute the expectation value of some operator $\hat{O}_A$ acting only on the A degrees of freedom, i.e.

$$\begin{aligned}
\langle\hat{O}_A\rangle &= \langle\psi_{AB}|\hat{O}_A|\psi_{AB}\rangle \\
&= \sum_{i,i'}\sum_{n,n'}\langle n'_B\,i'_A|\hat{O}_A|i_A\,n_B\rangle\,c^*_{i'n'}\,c_{in} \\
&= \sum_{i,i'}\sum_{n,n'}\langle n'_B|n_B\rangle\langle i'_A|\hat{O}_A|i_A\rangle\;c^*_{i'n'}\,c_{in} \\
&= \sum_{i,i'}\sum_{n} c_{in}\,c^*_{i'n}\;\langle i'_A|\hat{O}_A|i_A\rangle,
\end{aligned} \tag{5.12}$$

where in the third line we used that $\hat{O}_A$ does not depend on the B degrees of freedom, and in the fourth line we used that $\langle n'|n\rangle = \delta_{nn'}$.

Hence, from Eq. (5.10),

$$\begin{aligned}
\langle\hat{O}_A\rangle &= \sum_{i,i'}\langle i|\rho^A|i'\rangle\langle i'|\hat{O}_A|i\rangle \\
&= \sum_{i}\langle i|\rho^A\hat{O}_A|i\rangle \\
&= \mathrm{Tr}_A\left(\rho^A\hat{O}_A\right),
\end{aligned} \tag{5.13}$$

which has the same form as Eq. (5.5). Thus we can compute averages of quantities involving subsystem A from a knowledge of the density matrix ρ^A, without needing to explicitly consider subsystem B. All necessary information about B is contained in the density matrix ρ^A. Note that ρ^A is the same no matter what quantity of system A is to be calculated, and so it only has to be calculated *once*.

[3] See footnote 1 on page 42.

One can equivalently trace out the degrees of freedom in A to get the density matrix for subsystem B, i.e. $\rho^B = \mathrm{Tr}_A\, \rho^{AB}$, so

$$\langle n|\rho^B|n'\rangle = \sum_i c_{in}\, c^*_{in'}. \tag{5.14}$$

As we shall see, it is useful to diagonalize the density matrix, obtaining its eigenvalues λ_α and eigenvectors $|\phi_\alpha\rangle$. Since the sum of the eigenvalues is equal to the trace we have, according to Eq. (5.11),

$$\sum_\alpha \lambda_\alpha = 1, \tag{5.15}$$

which suggests that the eigenvalues can be interpreted as probabilities (since probabilities also sum to 1). We shall now see that this interpretation is correct.

Let's consider Eq. (5.13) in the basis where ρ^A is diagonal. We have

$$\begin{aligned} \langle \hat{O}_A \rangle &= \mathrm{Tr}\, \left(\rho^A \hat{O}_A \right) \\ &= \sum_\alpha \lambda_\alpha \langle \phi_\alpha | \hat{O}_A | \phi_\alpha \rangle. \end{aligned} \tag{5.16}$$

Thus we get the expectation value of $\hat{O}_A$ in state $|\psi_{AB}\rangle$ by (i) computing the expectation of $\hat{O}_A$ in state $|\phi_\alpha\rangle$ (an eigenvector of ρ^A), (ii) multiplying by λ_α (the corresponding eigenvalue of ρ^A), and (iii) summing over α. This clearly shows that λ_α should be thought of as the probability that subsystem A is in state $|\alpha\rangle$. To emphasize this, from now on we will denote the eigenvalues of the density matrix by p_α.

To summarize, to determine the properties of a subsystem from the density matrix when the state of the whole system is in a single quantum state:

1. We compute the elements of the density matrix according to Eq. (5.10).
2. The density matrix is Hermitian and so has real eigenvalues, p_i. The sum of the eigenalues is equal to one, and the eigenvalues are interpreted as probabilities.
3. If the eigenstate corresponding to eigenvalue p_i is denoted by $|u_i\rangle$ then the significance of the density matrix is that the subsystem can be thought of as being in state $|u_i\rangle$ with probablity p_i.
4. Expectation values of operators acting on the subsystem can be obtained from Eq. (5.13).

5.3 Determining if a state is entangled

One use of the density matrix is that it gives a systematic prescription for determining whether a state is a product state or entangled with respect to a partition into subsystems A and B. If it is not a product state we say that it is a mixed state and is "entangled" with respect to this partition. We shall use the terms "mixed state" and "entangled state" interchangeably.

To see how the density matrix can determine if a state is a product state or is entangled with respect to partition into A-B subsystems, let's assume initially that $|\psi_{AB}\rangle$ is a product state, i.e.

$$|\psi_{AB}\rangle = |\phi\rangle_A\, |\mu\rangle_B. \tag{5.17}$$

In this case subsystem A is definitely in state $|\phi\rangle$, so the eigenvalues of ρ^A must be $p_1 = 1$ and $p_\alpha = 0$ for $\alpha \neq 1$. Also the eigenvector for the non-zero eigenvalue must be given by $|\phi_1\rangle = |\phi\rangle$.

Hence, if the state of the combined AB system is a product state then one of the eigenvalues of the density matrix of A (or of B) will be 1 and the others zero. Conversely if more than one of the

eigenvalues of the density matrix are positive (since they are probabilities they can only be positive or zero) the state is mixed, i.e. entangled.

It is actually not necessary to diagonalize the density matrix to determine if the state is a product state or entangled. Instead it is sufficient to take its square. To see this note that

$$\text{Tr } \left(\rho^A\right)^2 = \sum_\alpha p_\alpha^2, \tag{5.18}$$

where we used that the trace is the sum of the eigenvalues, see Sec. 2.6, and that the eigenvalues of the square of a matrix are the square of the eigenvalues of that matrix. Since the p_α must lie between 0 and 1 and $\sum_\alpha p_\alpha = 1$, one can show that $\sum_\alpha p_\alpha^2 \leq 1$, with the equality only holding if one of the p_α is 1 and the others zero. As an example, consider the case of two states, for which the eigenvalues are p and $1-p$ with $0 \leq p \leq 1$. Now

$$\text{Tr } \left(\rho^A\right)^2 = \sum_{\alpha=1}^{2} p_\alpha^2 = p^2 + (1-p)^2 = 1 - 2p + 2p^2 = 1 - 2p(1-p). \tag{5.19}$$

For $0 \leq p \leq 1$, we see that $0 \leq 2p(1-p) \leq 1/2$ and is only zero for $p = 0$ and 1. Consequently, $\text{Tr } \left(\rho^A\right)^2 < 1$ unless $p = 0$ or 1.

Hence we have the following general criterion:

$$\text{if Tr } \left(\rho^A\right)^2 \begin{cases} = 1, & \text{then we have a product state,} \\ < 1, & \text{then we have a mixed (entangled) state,} \end{cases} \tag{5.20}$$

We emphasize again that $\text{Tr}\, \rho^A = 1$ always.

Sometimes one defines the Von Neumann entanglement entropy by

$$S(\rho^A) = -\text{Tr}\, \rho^A \log \rho^A \ (= -\sum_\alpha p_\alpha \log p_\alpha). \tag{5.21}$$

It is easy to see that $S(\rho^A) = 0$ if the state is a product state since $\lim_{x\to 0}(x \ln x) = 0$. In the opposite limit, of a maximally entangled state where $p_\alpha = 1/N_A$ for all α, one has $S(\rho^A) = \log N_A$. For the case where subsystem A is a single qubit, this gives $S(\rho^A) = \log 2$.

5.4 Some Simple Examples

In this section we consider some simple examples where subsystems A and B each have just a single qubit.

5.4.1 Example 1:

We take

$$|\psi_{AB}\rangle = \frac{1}{2}\left(|0_A 0_B\rangle + |0_A 1_B\rangle - |1_A 0_B\rangle - |1_A 1_B\rangle\right) \tag{5.22}$$

Note: We can see "by inspection" that this is a product state

$$|\psi_{AB}\rangle = \frac{1}{\sqrt{2}}\left(|0_A\rangle - |1_A\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0_B\rangle + |1_B\rangle\right). \tag{5.23}$$

We shall now show how this result is obtained from the density matrices ρ^A and ρ^B.

We have

$$c_{00} = c_{01} = \tfrac{1}{2}, \qquad c_{10} = c_{11} = -\tfrac{1}{2}, \tag{5.24}$$

so, from Eq. (5.10),

$$\begin{aligned}\rho^A_{00} &= c_{00}c_{00} + c_{01}c_{01} = \tfrac{1}{2}\\ \rho^A_{01} &= c_{00}c_{10} + c_{01}c_{11} = -\tfrac{1}{2}\\ \rho^A_{10} &= c_{10}c_{00} + c_{11}c_{01} = -\tfrac{1}{2}\\ \rho^A_{11} &= c_{10}c_{10} + c_{11}c_{11} = \tfrac{1}{2},\end{aligned} \tag{5.25}$$

and hence

$$\rho^A = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}. \tag{5.26}$$

The eigenvalues are given by

$$\begin{vmatrix} \frac{1}{2}-\lambda & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2}-\lambda \end{vmatrix} = 0 \tag{5.27}$$

so

$$\left(\lambda - \tfrac{1}{2}\right)^2 - \left(-\tfrac{1}{2}\right)^2 = 0 \tag{5.28}$$

which gives $\lambda = 1$ and 0. Since only one eigenvalue is non-zero this is a product state, as we saw above.

One easily finds that

$$\left(\rho^A\right)^2 = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \tag{5.29}$$

and so $\mathrm{Tr}\left(\rho^A\right)^2 = 1$ as required for a product state.

The eigenvector with eigenvalue $\lambda = 1$ is given by

$$\frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}. \tag{5.30}$$

Both the resulting equations give $b = -a$ so the normalized eigenvector is

$$|\phi_{1,A}\rangle = \frac{1}{\sqrt{2}}|\left(|0_A\rangle - |1_A\rangle\right). \tag{5.31}$$

Hence, with probability 1, subsystem A is in state $|\phi_1\rangle$, in agreement with Eq. (5.23).

One can repeat the same calculation for ρ^B. The results are

$$\begin{aligned}\rho^B_{00} &= c_{00}c_{00} + c_{10}c_{10} = \tfrac{1}{2}\\ \rho^B_{01} &= c_{00}c_{01} + c_{10}c_{11} = \tfrac{1}{2}\\ \rho^B_{10} &= c_{01}c_{00} + c_{11}c_{10} = \tfrac{1}{2}\\ \rho^B_{11} &= c_{01}c_{01} + c_{11}c_{11} = \tfrac{1}{2},\end{aligned} \tag{5.32}$$

so

$$\rho^B = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \tag{5.33}$$

The eigenvalues are given by

$$\begin{vmatrix} \frac{1}{2}-\lambda & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2}-\lambda \end{vmatrix} = 0 \tag{5.34}$$

so

$$\left(\lambda - \tfrac{1}{2}\right)^2 - \left(\tfrac{1}{2}\right)^2 = 0 \tag{5.35}$$

which gives $\lambda = 1$ and 0, the same as for ρ^A. It is true in general that the non-zero eigenvalues of ρ^A and ρ^B must be equal, provided that the combined AB system is in a single quantum state. This is discussed further in the more advanced material in Appendix 5.A

The eigenvector with eigenvalue $\lambda = 1$ is given by

$$\frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & 1\end{pmatrix}\begin{pmatrix}a\\ b\end{pmatrix} = \begin{pmatrix}a\\ b\end{pmatrix}. \tag{5.36}$$

Both the resulting equations give $b = a$ so the normalized eigenvector is

$$|\sigma_{1,B}\rangle = \frac{1}{\sqrt{2}}|\left(|0_B\rangle + |1_B\rangle\right). \tag{5.37}$$

Hence, with probability 1, subsystem B is in state $|\sigma_1\rangle$, again in agreement with Eq. (5.23).

5.4.2 Example 2:

In this example we take one of the Bell states,

$$|\psi_{AB}\rangle = \frac{1}{\sqrt{2}}\left(|0_A 0_B\rangle + |1_A 1_B\rangle\right), \tag{5.38}$$

which is clearly entangled. Here we have

$$c_{00} = c_{11} = \tfrac{1}{\sqrt{2}}, \qquad c_{10} = c_{01} = 0. \tag{5.39}$$

Hence

$$\begin{aligned}\rho^A_{00} &= c_{00}c_{00} + c_{01}c_{01} = \tfrac{1}{2}\\ \rho^A_{01} &= c_{00}c_{10} + c_{01}c_{11} = 0\\ \rho^A_{10} &= c_{10}c_{00} + c_{11}c_{01} = 0\\ \rho^A_{11} &= c_{10}c_{10} + c_{11}c_{11} = \tfrac{1}{2},\end{aligned} \tag{5.40}$$

so

$$\rho^A = \frac{1}{2}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}. \tag{5.41}$$

This is already in diagonal form so we read off that the two eigenvalues are both equal to $1/2$. Since more than one eigenvalue is positive, the state is mixed. A density matrix like this, with all eigenvalues equal, is *maximally entangled*. It is easy to see that the same eigenvalues are obtained from ρ^B.

Trivially

$$\mathrm{Tr}\left(\rho^A\right)^2 = \frac{1}{4}\mathrm{Tr}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} = \frac{1}{2}\ (< 1), \tag{5.42}$$

which indicates, again, that Eq. (5.38) is a mixed state.

5.4.3 Example 3:

This example is slightly more complicated but it is useful to go through it in detail. We take

$$|\psi_{AB}\rangle = \frac{1}{\sqrt{8}}\left(|0_A 0_B\rangle + \sqrt{3}|0_A 1_B\rangle - \sqrt{3}|1_A 0_B\rangle - |1_A 1_B\rangle\right), \tag{5.43}$$

so

$$c_{00} = \tfrac{1}{\sqrt{8}}, \qquad c_{01} = \sqrt{\tfrac{3}{8}}, \qquad c_{10} = -\sqrt{\tfrac{3}{8}}, \qquad c_{11} = -\tfrac{1}{\sqrt{8}}. \tag{5.44}$$

It follows that

$$\begin{aligned}\rho^A_{00} &= c_{00}c_{00} + c_{01}c_{01} = \tfrac{1}{2}\\ \rho^A_{01} &= c_{00}c_{10} + c_{01}c_{11} = -\tfrac{\sqrt{3}}{4}\\ \rho^A_{10} &= c_{10}c_{00} + c_{11}c_{01} = -\tfrac{\sqrt{3}}{4}\\ \rho^A_{11} &= c_{10}c_{10} + c_{11}c_{11} = \tfrac{1}{2},\end{aligned} \tag{5.45}$$

so

$$\rho^A = \frac{1}{4}\begin{pmatrix} 2 & -\sqrt{3} \\ -\sqrt{3} & 2 \end{pmatrix}. \tag{5.46}$$

The eigenvalues are found to be

$$p_1 = \frac{1}{4}\left(2+\sqrt{3}\right), \qquad p_2 = \frac{1}{4}\left(2-\sqrt{3}\right), \tag{5.47}$$

with corresponding eigenvectors

$$\begin{aligned}|\phi_{1,A}\rangle &= \frac{1}{\sqrt{2}}\left(|0_A\rangle - |1_A\rangle\right)\\ |\phi_{2,A}\rangle &= \frac{1}{\sqrt{2}}\left(|0_A\rangle + |1_A\rangle\right).\end{aligned} \tag{5.48}$$

Thus subsystem A can be regarded as being in state $|\phi_1\rangle$ with probability p_1 and in state $|\phi_2\rangle$ with probability p_2.

It is straightforward to show that

$$\left(\rho^A\right)^2 = \frac{1}{16}\begin{pmatrix} 7 & -4\sqrt{3} \\ -4\sqrt{3} & 7 \end{pmatrix} \tag{5.49}$$

and so

$$\mathrm{Tr}\left(\rho^A\right)^2 = \frac{7}{8}\ (<1), \tag{5.50}$$

in agreement with Eq. (5.43) being a mixed state.

Repeating the same arguments for ρ^B gives

$$\begin{aligned}\rho^B_{00} &= c_{00}c_{00} + c_{10}c_{10} = \tfrac{1}{2}\\ \rho^B_{01} &= c_{00}c_{01} + c_{10}c_{11} = \tfrac{\sqrt{3}}{4}\\ \rho^B_{10} &= c_{01}c_{00} + c_{11}c_{10} = \tfrac{\sqrt{3}}{4}\\ \rho^B_{11} &= c_{01}c_{01} + c_{11}c_{11} = \tfrac{1}{2},\end{aligned} \tag{5.51}$$

so

$$\rho^B = \frac{1}{4}\begin{pmatrix} 2 & \sqrt{3} \\ \sqrt{3} & 2 \end{pmatrix}. \tag{5.52}$$

The eigenvalues are found to be again given by Eq. (5.47) and the corresponding eigenvectors are

$$\begin{aligned}|\sigma_{1,B}\rangle &= \frac{1}{\sqrt{2}}\left(|0_B\rangle + |1_B\rangle\right)\\ |\sigma_{2,B}\rangle &= \frac{1}{\sqrt{2}}\left(-|0_B\rangle + |1_B\rangle\right).\end{aligned} \tag{5.53}$$

Subsystem B can therefore be regarded as being in state $|\sigma_1\rangle$ with probability p_1 and in state $|\sigma_2\rangle$ with probability p_2.

It is interesting to note that if we define

$$c_1 = \frac{1}{2}\sqrt{2+\sqrt{3}}, \qquad c_2 = \frac{1}{2}\sqrt{2-\sqrt{3}}, \tag{5.54}$$

so

$$p_1 = c_1^2, \qquad p_2 = c_2^2, \tag{5.55}$$

then a bit of algebra[4] shows that

$$|\psi_{AB}\rangle = c_1|\phi_{1,A}\rangle \otimes |\sigma_{1,B}\rangle + c_2|\phi_{2,A}\rangle \otimes |\sigma_{2,B}\rangle. \tag{5.56}$$

This is an example of Schmidt decomposition which is described in the more advanced material in Appendix 5.A. The coefficients c_1 and c_2 are known as Schmidt coefficients.

According to Eq. (5.56) we can decompose $|\psi_{AB}\rangle$ in the following way: with probability $p_1 = c_1^2$ subsystem A is in state $|\psi_{1,A}\rangle$ and subsystem B is in state $|\sigma_{1,B}\rangle$, and with probability $p_2 = c_2^2$ $(= 1-p_1)$ subsystem A is in state $|\psi_{2,A}\rangle$ and subsystem B is in state $|\sigma_{2,B}\rangle$. In this way one can see why the non-zero eigenvalues of the two subsystem density matrices ρ^A and ρ^B must be equal, namely for both matrices the eigenvalues are given by c_1^2 and c_2^2.

5.5 Systems not in a single quantum state

An additional application for the density matrix is for systems which are not described by a single quantum state. An example would be to characterize the behavior of a stream of particles (electrons, say) which are polarized in different directions. We need to average over the different spin orientations using standard classical statistics.

Suppose for example that a fraction p of the electrons are polarized in the $+z$ direction, i.e are in state $|0\rangle$, and a fraction $1-p$ are in the -z direction. The density matrix for particles in state $|0\rangle$ is

$$|0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \tag{5.57}$$

and that for state $|1\rangle$ is

$$|1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{5.58}$$

The density matrix of the stream of electrons is therefore

$$\rho = p|0\rangle\langle 0| + (1-p)|1\rangle\langle 1| = \begin{pmatrix} p & 0 \\ 0 & 1-p \end{pmatrix}. \tag{5.59}$$

For a less trivial example, consider the case that a fraction p of the electrons are in state $|0\rangle$ (polarized in the $+z$ direction) while fraction $1-p$ are polarized in state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (polarized in the $+x$ direction). The density matrix can then be conveniently written as

$$\rho = p|0\rangle\langle 0| + (1-p)|+\rangle\langle +|. \tag{5.60}$$

Note that states $|0\rangle$ and $|+\rangle$ are not orthogonal. If we rewrite Eq. (5.60) in terms of orthogonal states, (for example computational basis states) it becomes more complicated. To do this we note that

$$|+\rangle\langle +| = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \tag{5.61}$$

[4]Note that $\sqrt{2+\sqrt{3}} - \sqrt{2-\sqrt{3}} = \sqrt{2}$ and $\sqrt{2+\sqrt{3}} + \sqrt{2-\sqrt{3}} = \sqrt{6}$, which are proved by squaring both sides.

so the density matrix for the beam of electrons can be written in the computational basis as

$$\rho = \begin{pmatrix} (1+p)/2 & (1-p)/2 \\ (1-p)/2 & (1-p)/2 \end{pmatrix}. \tag{5.62}$$

The eigenvalues of ρ are

$$\lambda_{\pm} = \frac{1}{2}\left[1 \pm \sqrt{1-2p+2p^2}\right], \tag{5.63}$$

while the normalized eigenvectors are sufficiently messy that I prefer to not write them down.

Thus, in this case, the description of the density matrix in terms of non-orthogonal states in Eq. (5.60) is simpler than the description in terms of orthogonal states. Note that the factors of p and $(1-p)$ in Eq. (5.60) are *not* the eigenvalues. These have to be determined in an orthogonal basis and are given by Eq. (5.63).

5.6 Conclusions

We have seen that the density matrix is useful when studying the properties of a system composed of two subsystems A and B. More precisely, it can be used to:

- Determine the properties of one of the subsystems A without explicitly having to include the degrees of freedom of the other subsystem B. This is particularly useful if B contains a very large number of degrees of freedom. An example of a large "subsystem" is the environment, with which, unfortunately, the qubits of a quantum computer unavoidably interact.
- If the combined AB system is in a single state, the properties of the subsystem density matrices tell us whether that state is a product state with respect to the A-B partition or whether, on the other hand, it is a mixed state in which the two subsystems are entangled.

Appendices

5.A Schmidt Decomposition

(This is more advanced material which is not required for the course.)

It can be shown [NC00] that a state $|\psi_{AB}\rangle$ can be written as

$$|\psi_{AB}\rangle = \sum_{\alpha} c_\alpha \, |\phi_{\alpha,A}\rangle \otimes |\sigma_{\alpha,B}\rangle, \tag{5.64}$$

where the number of terms is less than or equal to the smaller of N_A and N_B, and the $|\phi_\alpha\rangle$ are mutually orthogonal as are the $|\sigma_\alpha\rangle$. The c_α are real, non-negative numbers called Schmidt coefficients. It is always possible to make the c_α real and non-negative because the phases of $|\phi_{\alpha,A}\rangle$ and $|\sigma_{\alpha,B}\rangle$ can be chosen independently. The sum in Eq. (5.64) is known as a Schmidt decomposition. An example of a Schmidt decomposition is shown in Eq. (5.56).

Using the definition of ρ_A given in Eq. (5.7) and working in the $|\phi_\alpha\rangle$ basis for the states of A and the $|\sigma_\alpha\rangle$ basis for the states of B, one has

$$\begin{aligned} \rho^A &= \mathrm{Tr}_B \, |\psi_{AB}\rangle\langle\psi_{AB}| \\ &= \sum_{\alpha} c_\alpha^2 \, |\phi_{\alpha,A}\rangle \, \langle\phi_{\alpha,A}|. \end{aligned} \tag{5.65}$$

This shows that ρ_A has non-zero eigenvalues $p_\alpha = c_\alpha^2$ with corresponding eigenvectors $|\phi_\alpha\rangle_A$.

Similarly one has

$$\begin{aligned}\rho^B &= \mathrm{Tr}_A\,|\psi_{AB}\rangle\langle\psi_{AB}| \\ &= \sum_\alpha c_\alpha^2\,|\sigma_{\alpha,B}\rangle\ \langle\sigma_{\alpha,B}|,\end{aligned} \tag{5.66}$$

which shows that ρ_B has non-zero eigenvalues $p_\alpha = c_\alpha^2$ (the same as for ρ^A) with corresponding eigenvectors $|\sigma_\alpha\rangle_B$. The number of non-zero Schmidt coefficients (the c_α) is called the Schmidt number (or Schmidt rank). If the Schmidt number is 1 the state is a product state, while if it is greater than 1, the state is entangled (mixed).

5.B Change in the density matrix under a unitary transformation

If qubit A (more generally subsystem A) is acted by a unitary transformation U^A then we show now that the density matrix for subsystem A changes from ρ^A to $\rho^{'A}$ where:

$$\rho^{'A} = U^A \rho^A \left(U^A\right)^\dagger. \tag{5.67}$$

To see this, note that $|\psi_{AB}\rangle$ in Eq. (5.9) goes to $|\psi'_{AB}\rangle$ where

$$|\psi'_{AB}\rangle = \sum_{i,j} c'_{ij} |i_A\rangle \otimes |j_B\rangle \tag{5.68}$$

in which

$$c'_{ij} = \sum_k U^A_{ik}\, c_{kj} \tag{5.69}$$

describes the change in amplitudes produced by the action of U^A. Note that the second index j on the amplitude c_{ij} refers to subsystem B and is not changed. Hence

$$\begin{aligned}\rho^{'A}_{i,i'} &= \sum_j c'_{ij} c'^*_{i'j} \\ &= \sum_{j,k_1,k_2} U^A_{ik_1}\, c_{k_1 j}\, U^{A*}_{i'k_2} c^*_{k_2 j} \\ &= \sum_{k_1,k_2} U^A_{ik_1} \left(\sum_j c_{k_1 j} c^*_{k_2 j} \right) U^{A*}_{i'k_2} \\ &= \sum_{k_1,k_2} U^A_{ik_1} \rho^A_{k_1,k_2} \left(U^A_{k_2 i'}\right)^\dagger \\ &= \left(U^A \rho^A \left(U^A\right)^\dagger \right)_{i,i'},\end{aligned} \tag{5.70}$$

so we obtain Eq. (5.67).

Note that the most general operation that can be applied to the combined AB system is a unitary transformation acting on the *whole system*, not just on subsystem A. One can show that if one performs such a *general* unitary operation on the combined system, and then recomputes the density matrix of subsystem A, the new density matrix is *not in general related to the old one by a unitary transformation*. This is how irreversible processes can occur in a subsystem which is coupled to the environment. A more detailed discussion of this is beyond the scope of the course but the interested student is referred to Nielsen and Chuang [NC00] and Rieffel and Polak [RP14].

Chapter 6

Coherent Superposition Versus Incoherent Addition of Probabilities

The purpose of this chapter is to clarify the distinction between a *coherent* superposition of *amplitudes* in quantum mechanics and an *incoherent* (classical) addition of *probabilities*.

6.1 Coherent Linear Superposition: 1 qubit

To illustrate coherent superposition, consider one qubit in the following state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{6.1}$$

where $|\alpha|^2 + |\beta|^2 = 1$. We denote $|\alpha^2|$ by p. Evidently $|\psi\rangle$ is a linear superposition of basis states $|0\rangle$ and $|1\rangle$. We say it is a *coherent* superposition because there is a well defined phase relationship between the pieces in the superposition, which means that there can be *interference* between these pieces in subsequent operations.

If we measure $|\psi\rangle$ in the computational basis we get

$$\begin{aligned} &|0\rangle \text{ with probability } |\alpha|^2 = p, \\ &|1\rangle \text{ with probability } |\beta|^2 = 1 - p. \end{aligned} \tag{6.2}$$

To show the effects of interference we apply a Hadamard gate, defined in Eq. (2.34), before doing the measurement. The result is

$$|\psi'\rangle = H|\psi\rangle = \frac{\alpha}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) + \frac{\beta}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = \left(\frac{\alpha+\beta}{\sqrt{2}}\right)|0\rangle + \left(\frac{\alpha-\beta}{\sqrt{2}}\right)|1\rangle. \tag{6.3}$$

If we do a measurement in the computational basis *after* applying the Hadamard, the results are

$$\begin{aligned} &|0\rangle \text{ with probability } \tfrac{1}{2}|\alpha+\beta|^2 = \tfrac{1}{2}\left(1 + \alpha\beta^* + \alpha^*\beta\right), \\ &|1\rangle \text{ with probability } \tfrac{1}{2}|\alpha-\beta|^2 = \tfrac{1}{2}\left(1 - \alpha\beta^* - \alpha^*\beta\right). \end{aligned} \tag{6.4}$$

The factor $\alpha\beta^* + \alpha^*\beta$ comes from *interference* between the two pieces in the linear combination of $|\psi\rangle$ in Eq. (6.1). In particular, if $\alpha = \beta = \frac{1}{\sqrt{2}}$, so $p = \frac{1}{2}$, we get

$$\begin{aligned} &|0\rangle \text{ with probability } 1, \\ &|1\rangle \text{ with probability } 0, \end{aligned} \tag{6.5}$$

showing that there is *zero* probability of getting state $|1\rangle$ in this case if we measure after performing a Hadamard. The vanishing probability of getting $|1\rangle$ is due to destructive interference between the two pieces of the superposition in state $|\psi\rangle$ in Eq. (6.1).

6.2 Incoherent (Classical) Addition of Probabilities: 2 qubits

An example of a situation with classical probabilities is measuring a single qubit in the presence of external noise. Suppose the qubit starts out in state $|\psi\rangle$ in Eq. (6.1) but is then acted on by noise which will randomise the phases of the two parts of the superposition. After the noise has acted for some time, we can write the state in terms of a global phase θ and a relative phase ϕ as

$$|\psi\rangle = e^{i\theta}\left(\alpha|0\rangle + e^{i\phi}\beta|1\rangle\right). \tag{6.6}$$

If we apply a Hadamard this state becomes

$$e^{i\theta}\left(\frac{\alpha+e^{i\phi}\beta}{\sqrt{2}}\right)|0\rangle + e^{i\theta}\left(\frac{\alpha-e^{i\phi}\beta}{\sqrt{2}}\right)|1\rangle. \tag{6.7}$$

If we then measure, we will get state $|0\rangle$ with probability

$$\frac{1}{2}e^{i\theta}\left(\alpha+e^{i\phi}\beta\right)e^{-i\theta}\left(\alpha^\star+e^{-i\phi}\beta^\star\right) = \frac{1}{2}\left(|\alpha|^2+|\beta|^2+e^{-i\phi}\alpha\beta^\star+e^{i\phi}\alpha^\star\beta\right). \tag{6.8}$$

We need to average over the relative phase ϕ. After some time the external noise will have completely randomized the phases so each value of ϕ will be equally probable. Since $\int_0^{2\pi} e^{i\phi}\,d\phi = 0$ the interference terms disappear, so the probability of getting state $|0\rangle$ is $\frac{1}{2}\left(|\alpha|^2+|\beta|^2\right) = \frac{1}{2}$. In other words measuring the qubit after acting with a Hadamard one finds

$$\begin{aligned} &|0\rangle \text{ with probability } \tfrac{1}{2}, \text{ and similarly} \\ &|1\rangle \text{ with probability } \tfrac{1}{2}. \end{aligned} \tag{6.9}$$

The probabilities in Eq. (6.9) differ from those in Eq. (6.4), which is for the case of a coherent superposition, by the absence of the factors of $\alpha\beta^* + \alpha^*\beta$ which came from interference. Interference does not happen here because the phase relation between the $|0\rangle$ and $|1\rangle$ parts of the qubit state has been erased by noise.

As another example of the incoherent addition of probabilities, consider two qubits in the following entangled state

$$|\psi_2\rangle = \alpha|00\rangle + \beta|11\rangle, \tag{6.10}$$

where we again denote $|\alpha|^2$ by p. If $\alpha = \pm\beta = \frac{1}{\sqrt{2}}$ this is a Bell state. Let us write $|\psi_2\rangle$ more explicitly as

$$|\psi_2\rangle = \alpha|0_A\rangle\otimes|0_B\rangle + \beta|1_A\rangle\otimes|1_B\rangle. \tag{6.11}$$

If we focus on qubit A, say, then state $|\psi_2\rangle$ looks rather similar to the 1-qubit state $|\psi\rangle$ in Eq. (6.1), in that there is a piece where qubit A is $|0\rangle$ with amplitude α and a piece where qubit A is $|1\rangle$ with amplitude β. However, for $|\psi_2\rangle$, unlike for $|\psi\rangle$, each of these pieces goes with a different state for qubit B (i.e. $|\psi_2\rangle$ is entangled). Because of this entanglement, we do *not* get interference between the pieces of $|\psi_2\rangle$ if we perform operations on qubit A followed by a measurement of that qubit, as we now show.

If we measure qubit A before doing any operation on it we get

$$\begin{aligned} &|0\rangle \text{ with probability } p\ (=|\alpha|^2), \\ &|1\rangle \text{ with probability } 1-p\ (=|\beta|^2), \end{aligned} \tag{6.12}$$

which is the same as in Eq. (6.2) for a single qubit in a coherent superposition.

However, a difference appears if we perform a unitary transformation on qubit A before measuring it. Here we apply a Hadamard as we did in Sec. 6.1. After the Hadamard the state is given by

$$\begin{aligned}
|\psi_2'\rangle &= H_A|\psi_2\rangle \\
&= \alpha\left(H_A|0_A\rangle\right)\otimes|0_B\rangle + \beta\left(H_A|1_A\rangle\right)\otimes|1_B\rangle \\
&= \frac{\alpha}{\sqrt{2}}\left(|0_A0_B\rangle + |1_A0_B\rangle\right) + \frac{\beta}{\sqrt{2}}\left(|0_A1_B\rangle - |1_A1_B\rangle\right) \\
&= \frac{1}{\sqrt{2}}\Big[\,|0_A\rangle\otimes\left(\alpha|0_B\rangle + \beta|1_B\rangle\right) + |1_A\rangle\otimes\left(\alpha|0_B\rangle - \beta|1_B\rangle\right)\Big] \\
&= \frac{1}{\sqrt{2}}\Big[\,|0_A\rangle\otimes|\phi_{0,B}\rangle + |1_A\rangle\otimes|\phi_{1,B}\rangle\,\Big],
\end{aligned} \tag{6.13}$$

where

$$\begin{aligned}
|\phi_{0,B}\rangle &= \alpha|0_B\rangle + \beta|1_B\rangle \\
|\phi_{1,B}\rangle &= \alpha|0_B\rangle - \beta|1_B\rangle.
\end{aligned} \tag{6.14}$$

According to the generalized Born hypothesis discussed in Sec. 3.9, if one measures qubit A *after* acting with the Hadamard one observes

$$\begin{aligned}
&|0_A\rangle\otimes|\phi_{0,B}\rangle \text{with probability } \tfrac{1}{2}, \\
&|1_A\rangle\otimes|\phi_{1,B}\rangle \text{with probability } \tfrac{1}{2}.
\end{aligned} \tag{6.15}$$

Hence one observes qubit A to be in state $|0\rangle$ with probability $1/2$ and in state $|1\rangle$ with probability $1/2$. One could also obtain Eq. (6.15) by computing the density matrix for qubit A, see Chapter 5. Again, the probabilities differ from those in Eq. (6.4), which is for the case of a coherent superposition, by the absence of the factors of $\alpha\beta^* + \alpha^*\beta$ which came from interference.

Intuitively, interference terms do not appear when the qubit being investigated (qubit A here) is entangled with another qubit because there is then no well defined phase relation between the two parts of the superposition ($|0_A\rangle$ and $|1_A|\rangle$).

6.3 Summary

For a coherent superposition, to compute probabilities one sums the amplitudes and then squares, e.g.

$$\frac{1}{2}|\alpha+\beta|^2, \tag{6.16}$$

while for an incoherent addition of probabilities one squares and then sums, e.g.

$$\frac{1}{2}\left(|\alpha|^2 + |\beta|^2\right). \tag{6.17}$$

Chapter 7

Einstein-Podolsky-Rosen (EPR), Bell's inequalities, and *Local Realism*

7.1 Introduction

In classical physics, objects have definite properties irrespective of whether we measure them or not. This is called *objective reality*. A measurement just *reveals* a property which *already existed*.

However, this is not the case in quantum mechanics. To see this, suppose that a qubit is initially in the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right). \tag{7.1}$$

If we measure the qubit (i.e. measure Z) the Born rule states that we get $|0\rangle$ (i.e. eigenvalue $+1$) with probability $\frac{1}{2}$ and $|1\rangle$ (i.e. eigenvalue -1) with probability $\frac{1}{2}$. However, we can not infer from this that, *before* the measurement, the qubit was in state $|0\rangle$ with probability $\frac{1}{2}$ and $|1\rangle$ with probability $\frac{1}{2}$, for this leads to a contradiction as we will now see.

If we apply the Hadamard operator,

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{7.2}$$

to $|\psi\rangle$ we get

$$H|\psi\rangle = |0\rangle. \tag{7.3}$$

Hence, according to the Born rule if we measure a qubit in state $H|\psi\rangle$, i.e. after applying the Hadamard, we get $|0\rangle$ with probability 1.

However, suppose we assume that, *before* the measurement, the qubit in state $|\psi\rangle$ corresponds to being in state $|0\rangle$ with probability $\frac{1}{2}$ and $|1\rangle$ with probability $\frac{1}{2}$, then the action of H on $|\psi\rangle$ produces either $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ or $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, again with equal probability, and so, in either case, measurement of the qubit gives $|0\rangle$ or $|1\rangle$, again with equal probability. This is in contradiction to Eq. (7.3), which states that the measurement gives $|0\rangle$ with probability 1. Hence we can *not* assume that state $|\psi\rangle$ in Eq. (7.1) corresponds to being in $|0\rangle$ with probability $\frac{1}{2}$ and $|1\rangle$ with probability $\frac{1}{2}$.

One person who did not like that quantum mechanics describes a world *without* objective reality (and that quantum mechanics involves probabilities at a fundamental level), was Albert Einstein[1]. In 1935 he wrote a famous paper with Podolsky and Rosen (now called EPR), in which they simply *asserted* that nature has the property of objective reality. According to this picture of the world, the

[1]He reputedly claimed to Niels Bohr that "God does not play dice with the universe". Bohr's reported reply, which may be apocryphal, was "Albert, you shouldn't tell God what to do".

reason that, in general, measurements do not give a definite answer but give different results with various probabilities, is that quantum mechanics, as we have it, is *incomplete*. Rather, there is a deeper level of structure, which we don't have access to at present, with extra, hidden, variables, such that if we could access those variables, the measurement would be deterministic and would just reveal the state of the system which existed previously, i.e. we would have objective reality. The fact that measurements on a quantum state do not give a unique result is, in this picture, because the hidden variables have different values when the different measurements are done.

The classical, EPR picture is called *local realism*:

1. **Realism**. The measured values of each particle are objectively real. They have definite values before measurement and irrespective of whether or not a measurement is made.

2. **Locality**. A measurement of A does not affect B instantaneously. More precisely, the measurement of A has no effect on B if A and B are spatially separated, i.e. $|\vec{r}_A - \vec{r}_B| > ct$ where t is the time between measurements and c is the speed of light. This is just special relativity, one of Einstein's greatest insights.

7.2 An EPR Experiment

In this chapter we will describe an experiment in which quantum mechanics gives different results from *any* local realistic theory. Such experiments have been done and found to be in agreement with quantum mechanics and in disagreement with local realism.

EPR examined a thought experiment with entangled particles. We shall consider a simpler version of the EPR thought experiment due to Bohm. For this experiment we will derive a condition (an inequality) which any theory with local realism must have, but which is violated by quantum mechanics. This is one of many inequalities of a similar nature, the first of which was discovered by John Bell. Hence they are known quite generally as Bell's inequalities.

We suppose that an experimenter prepares pairs of 2-state particles (qubits) in the following entangled Bell state

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right). \tag{7.4}$$

In experiments the qubits will be photons. He sends one particle of the pair to Alice and the other, in the opposite direction, to Bob, see Fig. 7.1. He then repeats this for many pairs. Suppose that Alice and Bob measure the particles in the computational (Z) basis. If Alice measures $|0\rangle$ (for which the eigenvalue of Z is $+1$) then Bob must measure the opposite, i.e. $|1\rangle$ (for which the eigenvalue of Z is -1).

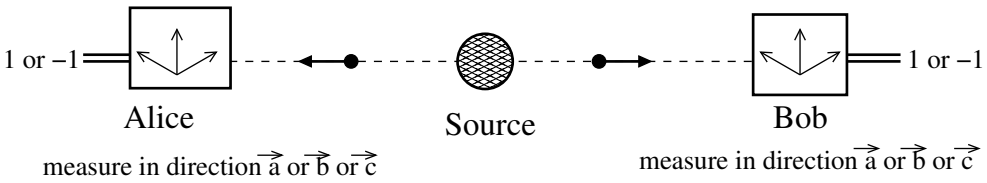


Figure 7.1: Sketch of the experimental setup for the version of the EPR experiment discussed in the text. The source emits pairs of qubits (in practice photons) in the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ given in Eq. (7.4). For each pair Alice and Bob decide independently and randomly which of the three non-orthogonal directions, $\vec{a}, \vec{b}$ or $\vec{c}$ to measure along. The result in each case is $+1$ or -1. The double lines indicate that the result of the measurement is a *classical* bit.

Now consider a general basis. As discussed in Chapter 4 a general qubit state $|0_{\hat{n}}\rangle$ is characterized by two parameters, θ and ϕ, which are the polar and azimuthal angles of a point in direction $\hat{n}$ on the unit sphere, known, in this context, as the Bloch sphere, see Fig. 4.1. The state on the antipodal point on the sphere is denoted by $|1_{\hat{n}}\rangle$. The connection between $|0_{\hat{n}}\rangle$ and $|1_{\hat{n}}\rangle$ and the basis states in the computational basis, $|0\rangle$ and $|1\rangle$, is given by Eqs. (4.11).

It is shown in Eq. (7.33) in Appendix 7.B, that Eq. (7.4) can equivalently be written as

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|0_{\hat{n}}\, 1_{\hat{n}}\rangle - |1_{\hat{n}}\, 0_{\hat{n}}\rangle \right), \tag{7.5}$$

ignoring an overall phase, for *any* direction $\hat{n}$. Hence the state in Eq. (7.4) has the interesting property that Alice and Bob will always get opposite results as long as they measure in the same basis[2] *no matter what that basis is.*

The results of the measurements of Alice and Bob are therefore strongly correlated. Of course, one can also have correlations between experimental results in classical systems. However, we will show below that the quantum correlations in entangled states like that in Eq, (7.5) are different from classical correlations.

In the experiment that we will consider, Alice and Bob can each choose to measure in one of three[3] distinct, non-orthogonal directions $\vec{a}, \vec{b}$ and $\vec{c}$. Every time they receive a particle they separately choose at random one of these three directions and record whether they get $+1$ or -1

The timing of the measurements is important. **They must be done a causally disconnected manner** so information about the direction that Alice has chosen can not have reached Bob when he makes his measurement, and vice versa.

The setup is sketched in Fig. 7.1.

7.3 Bells' Inequality

If Alice and Bob choose the same direction we know that they will get opposite results. Next consider in some detail what happens when Alice and Bob do not choose the same direction.

Firstly let us see what happens in a **classical picture with objective reality**.

The qubits then have a well defined state prior to the measurement. The reason that we don't always get the same result for measurements along a given direction must be that the qubit pairs are not all emitted in the same state each time. Rather, each possible result of the measurements corresponds to a particular type of initial state. There are three directions, for each of which Alice and Bob get one of two possible results. Let's first consider the results that Alice might get. For each of the three directions she gets one of two possible results, ± 1. With objective reality, the result of the measurement is pre-ordained before the measurement takes place, it just depends on the state of the photon. Furthermore, even though only one measurement direction is used for each photon, assuming objective reality it makes sense to talk about the results that Alice *would* have got if she had measured in one of the other directions. For example, there are photons where Alice would find $+1, +1, +1$ in the three directions. Let call these photons type 1. Since there are $2^3 = 8$ possible results for the three directions, there are eight possible types of photon, as far as Alice is concerned.

Now we incorporate Bob's results with those of Alice. Assuming that Bob's results are not affected by the measurement direction chosen by Alice, which is the case if the measurements are done in a causally disconnected manner, Bob's results are also determined only by the state of his photon when

[2]Note: when we say "measure the qubit in the $\hat{n}$ basis" we mean measure $\vec{\sigma} \cdot \hat{n}$, where, as discussed in Sec. 2.4, the $\sigma_\alpha, (\alpha = x, y, z)$ are just another notation for the Pauli operators X, Y and Z.

[3]In the simpler setup of two directions, one finds that there is no incompatibility between quantum mechanics and local realism. Three directions is the minimum needed to derive an inequality which is violated by quantum mechanics.

	Alice			Bob		
Population	$\vec{a}$	$\vec{b}$	$\vec{c}$	$\vec{a}$	$\vec{b}$	$\vec{c}$
N_1	+	+	+	−	−	−
N_2	+	+	−	−	−	+
N_3	+	−	+	−	+	−
N_4	+	−	−	−	+	+
N_5	−	+	+	+	−	−
N_6	−	+	−	+	−	+
N_7	−	−	+	+	+	−
N_8	−	−	−	+	+	+

Table 7.1: The eight types of qubit pairs give different results when measured along the $\vec{a}, \vec{b}$ and $\vec{c}$ directions. Note that Alice and Bob get opposite results if they measure in the same direction, so Bob's side of the table is precisely the opposite of Alice's. Hence there are 2^3 possible sets of outcomes.

emitted by the source. In this case, if he and Alice measure in the same direction we know that they must get opposite results[4]. For example if Alice receives a photon which would give $+1, +1, +1$ in the three directions, then Bob's photon would give $-1, -1, -1$. Hence, including both Alice and Bob's results, there are still only eight possible types of photon pair that we need consider, and these are shown in Table 7.1. For the i-th type, N_i pairs will be generated where

$$N = \sum_{i=1}^{8} N_i, \tag{7.6}$$

is the total number of pairs.

Let us discuss next some examples taken from Table 7.1. For a qubit pair in population 4, Alice will get $+1$ if she measures in direction $\vec{a}$, and Bob will get $+1$ if he measures in direction $\vec{b}$. Similarly for population 7, Alice will get -1 if she measures in direction $\vec{a}$ and Bob will get $+1$ if he measures in direction $\vec{b}$. In all cases, if Alice and Bob measure in the same direction they get opposite results.

We now make some simple observations. (Each observation is simple but one needs to focus to follow the thread of the argument to the end.) Clearly $N_i \geq 0$, so it must be true that

$$\frac{N_3 + N_4}{N} \leq \frac{N_2 + N_4}{N} + \frac{N_3 + N_7}{N}, \tag{7.7}$$

since N_2 and N_7, which can not be negative, have been added on the RHS.

- $\boldsymbol{(N_3, N_4)}$ Let's suppose that Alice measures along $\vec{a}$ and Bob along $\vec{b}$. The appropriate columns of Table 7.1 are collected in Table 7.2. According to Table 7.2 only for populations 3 and 4 would Alice and Bob both get $+1$. None of the other populations give this. Hence, among the times that Alice measures along $\vec{a}$ and Bob along $\vec{b}$, the probability that they both get $+1$ is $(N_3 + N_4)/N$. Let's call this $P(+\vec{a}; +\vec{b})$, in which the first argument refers to Alice and the second to Bob, i.e.

$$\frac{N_3 + N_4}{N} = P(+\vec{a}; +\vec{b}). \tag{7.8}$$

[4]The state $|\psi\rangle$ in Eq. (7.4), is known as a "spin singlet" state in the physics literature and has zero total spin angular momentum. Assuming that the initial state of the source, before the qubits are omitted, has zero angular momentum, then conservation of angular momentum *requires* that the qubits be emitted in state $|\psi\rangle$ and therefore that Alice and Bob must get opposite results if they measure in the same direction.

	Alice	Bob
Population	$\vec{a}$	$\vec{b}$
N_1	$+$	$-$
N_2	$+$	$-$
N_3	$+$	$+$
N_4	$+$	$+$
N_5	$-$	$-$
N_6	$-$	$-$
N_7	$-$	$+$
N_8	$-$	$+$

Table 7.2: The columns of Table 7.1 for the case when Alice measures along $\vec{a}$ and Bob along $\vec{b}$.

- $\boldsymbol{(N_2, N_4)}$. Following similar arguments, only for populations 2 and 4 would Alice get $+1$ measuring along $\vec{a}$ and Bob get $+1$ measuring along $\vec{c}$. Hence

$$\frac{N_2 + N_4}{N} = P(+\vec{a}; +\vec{c}). \tag{7.9}$$

- $\boldsymbol{(N_3, N_7)}$. Similarly, only for populations 3 and 7 would Alice get $+1$ measuring along $\vec{c}$ and Bob get $+1$ measuring along $\vec{b}$. Hence

$$\frac{N_3 + N_7}{N} = P(+\vec{c}; +\vec{b}). \tag{7.10}$$

Combining Eqs. (7.7)–(7.10), we have[5]

$$P(+\vec{a}; +\vec{b}) \leq P(+\vec{a}; +\vec{c}) + P(+\vec{c}; +\vec{b})\,. \tag{7.11}$$

In the simple case that all the populations are equal, each probability is $1/4$ so the inequality is trivially satisfied. Equation (7.11) is an example of a Bell's inequality. **It is satisfied by any theory with local realism.** Note that there is nothing sophisticated about this Bell's inequality; it is just bookkeeping. I emphasize that Eq. (7.11) has nothing to do with quantum mechanics. In fact, we will now see that it is *violated* by quantum mechanics for a broad range of measurement directions $\vec{a}, \vec{b}, \vec{c}$.

We therefore now consider what **quantum mechanics** has to say.

The 2-qubit state generated by the source is given by Eq. (7.5) for any direction $\hat{n}$, where $|0_{\hat{n}}\rangle$ and $|1_{\hat{n}}\rangle$ are given by Eqs. (7.27). We take the $\theta = \phi = 0$ direction to be that of $\vec{a}$. so we write

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0_{\vec{a}}\rangle_1 |1_{\vec{a}}\rangle_2 - |1_{\vec{a}}\rangle_1 |0_{\vec{a}}\rangle_2\right), \tag{7.12}$$

where we indicate on the RHS which qubit is meant (1 for Alice's and 2 for Bob's).

We now compute $P(+\vec{a}; +\vec{c})$ according to quantum mechanics. We need the probability amplitude for the state in Eq. (7.12) to have eigenvalue $+1$ along $\vec{a}$ for Alice and eigenvalue $+1$ along $\vec{c}$ for Bob, i.e. $|0_{\vec{a}}\rangle_1 |0_{\vec{c}}\rangle_2$. Hence, to get $P(+\vec{a}; +\vec{c})$ we compute first the amplitude

$$\left({}_1\langle 0_{\vec{a}}| \; {}_2\langle 0_{\vec{c}}| \right) |\psi\rangle = \frac{1}{\sqrt{2}}\Big(\langle 0_{\vec{a}}|0_{\vec{a}}\rangle_1 \langle 0_{\vec{c}}|1_{\vec{a}}\rangle_2 - \langle 0_{\vec{a}}|1_{\vec{a}}\rangle_1 \langle 0_{\vec{c}}|0_{\vec{a}}\rangle_2 \Big), \tag{7.13}$$

[5]Recall what we mean by these probabilities. $P(+\vec{a}; +\vec{b})$, for example, means that, *out of the times when Alice measures along $\vec{a}$ and Bob measures along $\vec{b}$*, this is the probability that they both get $+1$. The sum of the probabilities for the different measurement results for these fixed directions must add to 1, i.e. $P(+\vec{a}; +\vec{b}) + P(+\vec{a}; -\vec{b}) + P(-\vec{a}; +\vec{b}) + P(-\vec{a}; -\vec{b}) = 1$.

Now $\langle 0_{\vec{a}}|0_{\vec{a}}\rangle = 1$ and $\langle 0_{\vec{a}}|1_{\vec{a}}\rangle = 0$, so

$$\langle 0_{\vec{a}}\, 0_{\vec{c}}|\psi\rangle = \frac{1}{\sqrt{2}}\langle 0_{\vec{c}}|1_{\vec{a}}\rangle. \tag{7.14}$$

If $\vec{c}$ is at angles (θ_{ac}, ϕ_{ac}) relative to $\vec{a}$, then, according to Eq. (7.27a),

$$\frac{1}{\sqrt{2}}\langle 0_{\vec{c}}|1_{\vec{a}}\rangle = \frac{1}{\sqrt{2}} e^{i\phi_{ac}} \sin\frac{\theta_{ac}}{2}, \tag{7.15}$$

so

$$P(+\vec{a};+\vec{c}) = |\langle 0_{\vec{a}}\, 0_{\vec{c}}|\psi\rangle|^2 = \frac{1}{2}\left| e^{i\phi_{ac}} \sin\frac{\theta_{ac}}{2}\right|^2 = \frac{1}{2}\sin^2\left(\frac{\theta_{ac}}{2}\right). \tag{7.16}$$

We recall that out of the times when Alice measures along $\vec{a}$ and Bob measures along $\vec{c}$, this is the probability that they both get $+1$. For these same directions there are three other possibilities. It is straightforward to check that $P(-\vec{a};-\vec{c}) = P(+\vec{a};+\vec{c})$, and a calculation shows that

$$P(+\vec{a};-\vec{c}) = P(-\vec{a};+\vec{c}) = \frac{1}{2}\cos^2\left(\frac{\theta_{ac}}{2}\right). \tag{7.17}$$

Hence the sum of the probabilites for the four different (± 1) results when Alice measures along $\vec{a}$ and Bob measures along $\vec{c}$ adds up to 1 as required.

A check on Eq. (7.16) is that it predicts $P(+\vec{a};+\vec{c}) \to 0$ if $\vec{a}$ and $\vec{c}$ are in the same direction. This result is correct because when Alice and Bob measure in the same direction they must get *different* results because of the nature of $|\psi\rangle$, see Eq. (7.5).

Similarly one has

$$P(+\vec{a};+\vec{b}) = \frac{1}{2}\sin^2\left(\frac{\theta_{ab}}{2}\right), \tag{7.18}$$

$$P(+\vec{c};+\vec{b}) = \frac{1}{2}\sin^2\left(\frac{\theta_{cb}}{2}\right). \tag{7.19}$$

Hence Bell's inequality, Eq. (7.11), when applied to quantum mechanics, gives

$$\sin^2\left(\frac{\theta_{ab}}{2}\right) \le \sin^2\left(\frac{\theta_{ac}}{2}\right) + \sin^2\left(\frac{\theta_{cb}}{2}\right). \tag{7.20}$$

As we shall now see, it is easy to find cases where this is violated.

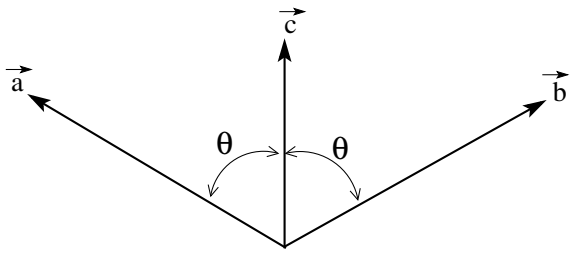


Figure 7.2: A possible choice of directions for which the Bell's inequality in Eq. (7.20) is violated.

Consider the situation in Fig. 7.2 where $\theta_{ac} = \theta_{cb} = \theta$, so $\theta_{ab} = 2\theta$, and take $\theta = \pi/3$. We have

$$\sin^2\left(\frac{\theta_{ac}}{2}\right) = \sin^2\left(\frac{\theta_{cb}}{2}\right) = \sin^2\left(\frac{\theta}{2}\right) = \sin^2(\frac{\pi}{6}) = \frac{1}{4}, \tag{7.21}$$

and

$$\sin^2\left(\frac{\theta_{ab}}{2}\right) = \sin^2\theta = \sin^2\left(\frac{\pi}{3}\right) = \frac{3}{4}. \tag{7.22}$$

Hence the LHS of Eq. (7.20) is 3/4 while the RHS is 1/2 so the inequality is violated. For general θ in Fig. 7.2, the inequality in Eq. (7.20) can be written

$$\sin\theta \leq \sqrt{2}\sin\left(\frac{\theta}{2}\right), \tag{7.23}$$

which is violated for the broad range $0 < \theta < \pi/2$, as shown graphically in Fig. 7.3.

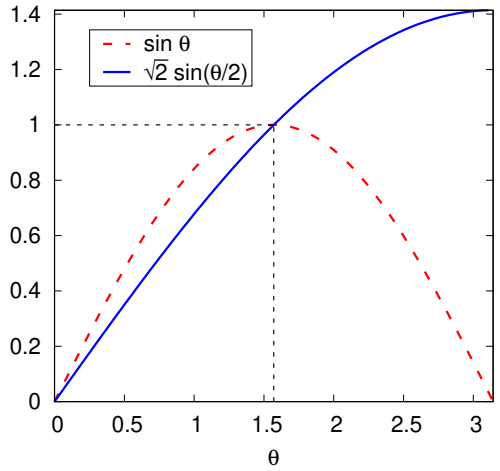


Figure 7.3: A graph showing that the inequality in Eq. (7.23) is violated for $0 < \theta < \pi/2$.

7.4 Summary

Hence quantum mechanics violates Bell's inequalities. These inequalities are satisfied by any theory with local realism. Experiments along the lines of that sketched in Fig. 7.1 have been done, using polarized photons. **These experiments agree with quantum mechanics and disagree with local realism.** See `https://physics.aps.org/articles/v8/123` for a brief discussion of these experiments. Among the different experiments there are variations in the initial state of the entangled qubits and in which Bell's inequality is being tested, but they are all equivalent. The more sophisticated experiments choose (randomly) the polarization directions while the photons are in flight. This makes it impossible for a measurement at one polarizer to depend on the chosen orientation of the other, since relativistic causality (i.e. the fact that information can not travel faster than the speed of light) prevents one polarizer from "knowing" the orientation of the other at the time of a measurement. This feature of the experiment is necessary to show that no *local* hidden variable theory can explain the data.

Bell's inequalities characterize quantum correlations between *two* entangled qubits, which are different from classical correlations. Very recently non-classical correlations, *distinct* from those of Bell, have been found in experiments with three sources of pairs of entangled photons and three detectors in the shape of a triangle, see `https://physics.aps.org/articles/v12/106`. Thus the study of non-classical correlations in quantum mechanics, stimulated by EPR in the 1930s, made precise by Bell in the 1960s, and studied experimentally since the 1970s, remains an active field up to the present day.

Although the experimentally found violations of Bell's inequalities rule out local theories with objective reality, they do not, in principle, rule out *non-local*[6] theories with objective reality. However, these would violate special relativity. Hence very few physicists think that a non-local theory of quantum mechanics will turn out to be the correct theory of nature.

In an EPR-like experiment the entangled state changes when one qubit is measured. We can ask whether any information is instantaneously transmitted to the other qubit at the moment of measurement. Since the two qubits in an entangled state are correlated, naively one might imagine that this occurs. If so, special relativity, one of the cornerstones of modern physics, would be violated. Fortunately, no information is transmitted at the moment of measurement, as we show in Appendix 7.A, so special relativity *is* preserved.

To conclude, we see that quantum mechanics is **strange**:

- Unlike in classical physics, probabilities enter in a *fundamental* way.
- Unlike in classical physics, we do not have objective reality. Reality is an *emergent* concept for bigger systems when we go over to a description in terms of classical physics.

Many physicists feel uncomfortable with these aspects of quantum mechanics, and hope that a better insight will emerge. But, in the 85 years since the EPR paper this has not happened, so we will probably have to continue living with the strange world of quantum mechanics as we now understand it.

Can we use the differences between the strange quantum world and the familiar classical world to do more efficient computation, at least for some problems? This question will be the focus of the rest of the course.

Appendix

7.A Information does not propagate faster than the speed of light

Consider a pair of entangled qubits A and B which are widely separated. If a measurement is done on qubit B then the state of the system changes, the final state depending on the result of the measurement. This change in state happens instantaneously. Does this mean that *information* is transmitted instantaneously to qubit A? If so, this would violate special relativity. We shall now see that this is not the case, no information is transferred at the moment of measurement, and therefore quantum mechanics does not violate special relativity.

Since qubits A and B are entangled we have to describe qubit A by a density matrix. To see its form we separate out the parts of the entangled state corresponding to B being in state $|0\rangle$ and B being in state $|1\rangle$. Referring to our discussion of the generalized Born rule in Sec. 3.9, we write the state of the two qubits *before* the measurement as

$$|\psi_{AB}\rangle = \alpha|\psi_{0,A}\rangle|0_B\rangle + \beta|\psi_{1,A}\rangle|1_B\rangle, \tag{7.24}$$

where $|\psi_{0,A}\rangle$ and $|\psi_{1,A}\rangle$ are normalized (but not necessarily orthogonal) states of qubit A, and $|\alpha|^2 + |\beta|^2 = 1$. As stated in Eq. (5.6) in Sec. 5.2, the density matrix of the two qubits in a well-defined quantum state is

$$\begin{aligned}\rho^{AB} &= |\psi_{AB}\rangle\langle\psi_{AB}| \\ &= \left(\,\alpha|\psi_{0,A}\rangle|0_B\rangle + \beta|\psi_{1,A}\rangle|1_B\rangle\,\right)\left(\,\alpha^*\langle\psi_{0,A}|\langle 0_B| + \beta^*\langle\psi_{1,A}|\langle 1_B|\,\right),\end{aligned} \tag{7.25}$$

[6]The term non-local refers to information propagating faster than the speed of light.

and the density matrix of qubit A alone is

$$\begin{aligned}\rho^A &= \mathrm{Tr}_B\, \rho^{AB} \\ &= \langle 0_B|\rho^{AB}|0_B\rangle + \langle 1_B|\rho^{AB}|1_B\rangle \\ &= |\alpha|^2|\psi_{0,A}\rangle\langle\psi_{0,A}| + |\beta|^2|\psi_{1,A}\rangle\langle\psi_{1,A}|. \end{aligned} \tag{7.26}$$

Note that Eq. (7.26) is a representation of the density matrix in terms of non-orthogonal states. Another example involving non-orthogonal states was described in Sec. 5.5. As discussed in Sec. 5.2.2, Eq. (7.26) implies that that qubit-A is in state $|\psi_{0,A}\rangle$ with probability $|\alpha|^2$ and is in state $|\psi_{0,B}\rangle$ with probability $|\beta|^2$.

Now consider the situation *after* the measurement on qubit B. According to the generalized Born rule discussed in Sec. 3.9, for the state of the combined AB system in Eq. (7.24), there is probability $|\alpha|^2$ that qubit B is measured to be in state $|0_B\rangle$ while qubit A is left in state $|\psi_{0,A}\rangle$, and there is probability $|\beta|^2$ that qubit B is measured to be in state $|1_B\rangle$ while qubit A is left in in state $|\psi_{1,A}\rangle$. For qubit A this situation is exactly the same as we found *before* the measurement, see Eq. (7.26).

Hence the density matrix for qubit A, which determines the probabilities of measurements on A, is *unchanged* by the measurement of the distant qubit B, even though the two qubits are entangled. Thus, although our *description* of the state of the two qubits does change instantaneously at the moment of measurement, *information* is not propagated instantaneously by the measurement and so special relativity is satisfied.

7.B The spin-singlet state is isotropic

We showed in Eqs. (4.11) of Chapter 4 that the eigenstate of spin in a direction specified by polar angles (θ, ϕ) with eigenvalue $+1$ is given by

$$|0_{\hat{n}}\rangle = \cos\tfrac{\theta}{2}\,|0\rangle + e^{i\phi}\,\sin\tfrac{\theta}{2}\,|1\rangle\,, \tag{7.27a}$$

see Fig. 4.1. A similar calculation gives the eigenstate corresponding to eigenvalue -1 to be

$$|1_{\hat{n}}\rangle = -\sin\tfrac{\theta}{2}\,|0\rangle + e^{i\phi}\,\cos\tfrac{\theta}{2}\,|1\rangle\,, \tag{7.27b}$$

which is the antipodal point where $\theta \to \pi - \theta, \phi \to \phi + \pi$.

From Eqs. (7.27a) and (7.27b), we see that the unitary matrix which transforms from the Z basis to the $\hat{n}$ basis is

$$U = \begin{pmatrix} \cos\frac{\theta}{2} & e^{i\phi}\,\sin\frac{\theta}{2} \\ -\sin\frac{\theta}{2} & e^{i\phi}\,\cos\frac{\theta}{2} \end{pmatrix}. \tag{7.28}$$

The inverse transformation is given by U^{-1}, but since U is unitary we have

$$U^{-1} = U^\dagger \equiv \left(U^T\right)^\star = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ e^{-i\phi}\,\sin\frac{\theta}{2} & e^{-i\phi}\,\cos\frac{\theta}{2} \end{pmatrix}, \tag{7.29}$$

so

$$|0\rangle = \cos\tfrac{\theta}{2}\,|0_{\hat{n}}\rangle - \sin\tfrac{\theta}{2}\,|1_{\hat{n}}\rangle \tag{7.30a}$$

$$|1\rangle = e^{-i\phi}\sin\tfrac{\theta}{2}\,|0_{\hat{n}}\rangle + e^{-i\phi}\,\cos\tfrac{\theta}{2}\,|1_{\hat{n}}\rangle\,. \tag{7.30b}$$

Hence the entangled Bell state $|\psi\rangle$ in Eq. (7.4), (which is called the spin-singlet state in the physics literature) can be written in the $\hat{n}$ basis as

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right) \tag{7.31}$$

$$\begin{aligned} &= \frac{1}{\sqrt{2}}\Big[\left(\cos\tfrac{\theta}{2}\,|0_{\hat{n}}\rangle_1 - \sin\tfrac{\theta}{2}\,|1_{\hat{n}}\rangle_1\right)\left(e^{-i\phi}\sin\tfrac{\theta}{2}\,|0_{\hat{n}}\rangle_2 + e^{-i\phi}\cos\tfrac{\theta}{2}\,|1_{\hat{n}}\rangle_2\right) - \\ &\qquad \left(e^{-i\phi}\sin\tfrac{\theta}{2}\,|0_{\hat{n}}\rangle_1 + e^{-i\phi}\cos\tfrac{\theta}{2}\,|1_{\hat{n}}\rangle_1\right)\left(\cos\tfrac{\theta}{2}\,|0_{\hat{n}}\rangle_2 - \sin\tfrac{\theta}{2}\,|1_{\hat{n}}\rangle_2\right)\Big] \end{aligned} \tag{7.32}$$

$$= \frac{e^{-i\phi}}{\sqrt{2}}\left(|0_{\hat{n}}1_{\hat{n}}\rangle - |1_{\hat{n}}0_{\hat{n}}\rangle\right)\,, \tag{7.33}$$

where, in the middle expression, we indicated by a subscript, e.g. $|\cdots\rangle_1$, whether the state is that of the first or second qubit. Apart from the unimportant overall phase factor of[7] $e^{-i\phi}$, Eq. (7.33) is the same form that state takes in the computational (Z) basis, Eq. (7.31). Hence if two qubits in the entangled Bell state in Eq. (7.4) are observed in the same basis (see footnote 2 on page 59), no matter which one, the results of the two measurements will always be opposite, one giving $+1$ and the other -1.

[7] Note that $e^{-i\phi}$ is just the determinant of the transformation matrix from the computational basis to the $\hat{n}$ basis given in Eq. (7.29). Quite generally, if the "singlet" state $|\psi\rangle$ in Eq. (7.31) is acted on by a unitary transformation V then one can show that $V|\psi\rangle = \det V|\psi\rangle$. Since V is unitary its determinant can only be a pure phase.

Chapter 8

Classical and Quantum Gates

Now, finally, we get to computation!

The elementary circuit elements which acts on the data in a computer are called gates. In this chapter we will first discuss classical gates and then go on to describe quantum gates.

8.1 Classical Gates

Data in a classical digital computer is in the form of bits, x, which take values 0 or 1. The only operation involving a single classical bit, i.e. the only 1-bit classical gate, is NOT which takes 0 to 1 and vice versa.

Of particular interest are 2 bit gates, the most common ones being

	In	Out	
AND	00	0	$x \wedge y$
	01	0	
	10	0	
	11	1	
	In	Out	
OR	00	0	$x \vee y$
	01	1	
	10	1	
	11	1	
	In	Out	
XOR	00	0	$x \oplus y$
	01	1	
	10	1	
	11	0	

(8.1)

These have two input bits and one output bit. For the AND gate the result is 0 unless both inputs are 1. For the OR gate the result is 0 unless one or both of the inputs are 1. The XOR gate only differs from the OR gate in giving zero if *both* the inputs are 1.

Note that AND gives the same results as multiplication of the bits xy. The XOR operation is equivalent to addition of the bits modulo 2, i.e. $x + y \,(\mathrm{mod}\ 2)$. To see this, note that the modulo

operation gives the remainder after integer division. For example, since $13 = (5 \times 2) + 3$ we have $13 (\mod 5) = 3$. Referring to the XOR gate consider the case $x = y = 1$, so we have $1 + 1 (\mod 2) = 0$, which is the value of XOR in this case. It is trivial to see that XOR is also addition modulo 2 for the other values of x and y. For convenience of notation $x + y (\mod 2)$ is written as $x \oplus y$.

One can show that the AND, NOT and OR gates form a **universal set** which means that any logical operation on a arbitrary number of bits can be expressed in terms of these gates. Thus, classically, we only need 1-bit and 2-bit gates to perform any operation.

However, we cannot directly take over gates like AND, OR and XOR to a quantum computer for the following reason. A gate in a quantum computer will be implemented by a unitary operator acting on a small number of qubits. A unitary operator has the property that $U^{-1} = U^\dagger$. Now U^{-1} performs the inverse operation, and since $U^\dagger$ is well defined the inverse operation must exist. **Thus, quantum gates must be reversible.**

However, AND, OR and XOR can not be reversible because they have a different number of outputs and inputs. Suppose, for example, we know that the output from an OR gate is 1, and want to know what is the input. We can't say because there are three possible inputs, 01, 10 and 11, which give this output.

Thus, a major change in going from classical to quantum computing will be having to deal with *reversible* computation. Next we will consider reversible *classical* computation before doing the quantum case.

Clearly a necessary condition for a gate to be reversible is that it has the same number of input and output bits. The 1-bit NOT gate has one input and one output, and is reversible since acting twice gives back the original bit, i.e. $(\text{NOT})^2 = \text{IDENTITY}$, so $(\text{NOT})^{-1} = \text{NOT}$, i.e. NOT is its own inverse.

We will now consider a reversible, classical, 2-bit gate, the quantum analog of which will play an important role in quantum computing. This is the controlled-NOT, or CNOT gate. It is similar to XOR except that it has a second output bit, which is equal to one of the input qubits, i.e. this qubit is unchanged on output. As we shall see, this simple modification, namely keeping one of the input bits as part of the output, suffices to make the CNOT gate a reversible version of XOR.

One way of representing the action of CNOT is

$$\begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow \begin{pmatrix} x \\ x \oplus y \end{pmatrix}. \tag{8.2}$$

The first (upper) bit is called the control bit. This is unchanged by the action of CNOT. The second (lower) bit is called the target bit, and the effect of the XOR operation $x \oplus y$ is to flip y if $x = 1$ and to leave y alone if $x = 0$. Hence, as far as the target bit is concerned, the gate is indeed a controlled NOT, since the NOT acts if x, the control bit, is 1, and does not act if $x = 0$. The truth table is as follows:

x	y	x'	y'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

. (8.3)

It is useful to represent the CNOT gate by a diagram, as shown in Fig. 8.1. The input is on the left and the output on the right. The upper line is the control bit, and has value x on input, while the lower line is the target bit and has value y on input. On output, the control qubit is unchanged and the target qubit is the exclusive or (XOR) of x and y.

It is easy to see that CNOT is reversible since, if we act twice, we get back the original input

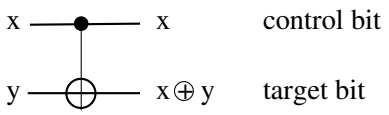


Figure 8.1: The CNOT gate. The input is on the left and the output on the right.

because

$$\begin{pmatrix} x \\ y \end{pmatrix} \stackrel{\text{CNOT}}{\longrightarrow} \begin{pmatrix} x \\ x \oplus y \end{pmatrix} \stackrel{\text{CNOT}}{\longrightarrow} \begin{pmatrix} x \\ x \oplus x \oplus y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}. \tag{8.4}$$

The last line follows because $x \oplus x = 0$ since $0 + 0 = 0$ and $1 + 1 = 0 \,(\mathrm{mod}\, 2)$. Thus CNOT is its own inverse. Hence, as mentioned earlier, it can therefore be regarded as a reversible version of XOR.

Note that to be reversible it is not required that the inverse operator is the same as the original operator, only that the inverse operator exists. However, it turns out that *most* quantum gates we consider will be their own inverse.

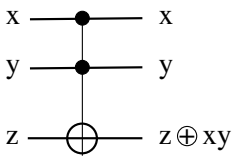


Figure 8.2: The Toffoli gate. This has two control bits x and y and one target bit z. On output the control bits are unchanged and the target bit is flipped if both control bits are 1, so $z \to z \oplus xy$.

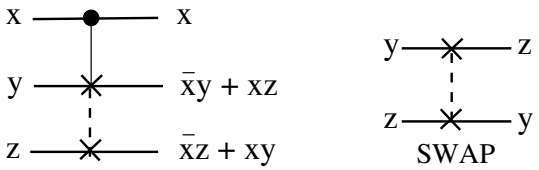


Figure 8.3: Left: the Fredkin gate. This is a controlled-swap gate. If the upper (control) bit is 1 then the two lower (target) bits are swapped, and otherwise the target bits are unchanged. $\overline{x} \equiv 1 - x$ is the complement of x. Right: the elemental SWAP gate.

We mentioned above that the 1-bit (NOT) gate and a set of irreversible 2-bit gates (AND and OR) together form universal set for a classical computer, which means that any logical operation on an arbitrary number of bits can be constructed out of these gates. The question we now ask is whether 1-bit and 2-bit *reversible* classical gates are universal. The answer is no. Classically one also needs a 3-bit gate such as the Toffoli gate shown in Fig. 8.2 or the Fredkin gate shown in Fig. 8.3.

Amazingly we shall see that 3-qubit gates are *not* needed quantum mechanically. In fact it is possible build the Toffoli gate, for example, out of 1-qubit and 2-qubit gates, and we will discuss how to do this in Sec. 12.B. We shall see that quantum mechanics allows for a big range of 1-qubit gates, whereas we have already noted that classically the only 1-bit gate is NOT. It is this wide range of possibilities for 1-qubit gates that allows us to construct a quantum mechanical Toffoli gate out of 1-qubit and 2-qubit gates, whereas no such construction is possible using classical gates.

8.2 Quantum Gates

Following David Deutsch we represent the action of quantum gates by a circuit. The circuit comprises a set of qubits in some initial state, acted on by gates and ending up in a final state. Each qubit is represented by a line in the circuit diagram and time runs from left to right, see e.g. Fig. 8.4.

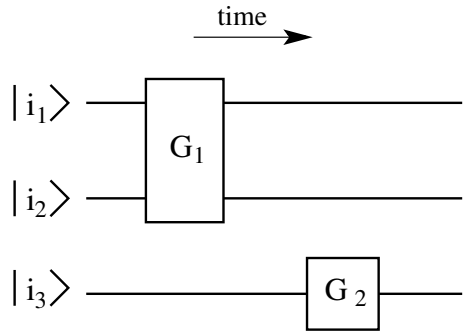
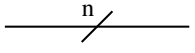


Figure 8.4: A schematic circuit with three qubits and two gates. Time runs from left to right. The initial state of the qubits is $|i_1\rangle \otimes |i_2\rangle \otimes |i_3\rangle$.

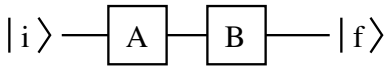
Sometimes we will indicate a set of n qubits (called a register) compactly by a single line with a slash through it as follows: 

.

Quantum circuits have the following properties:

- There are no loops, because qubits can't go back in time.
- Lines can't splay out (fan out) because of the no-cloning theorem.
- Similarly lines can't merge.
- Gates and circuits are *linear*. We evaluate the effect of the circuit on an initial state which is a computational basis state. However, if the initial qubits are in a superposition of computational basis states, then the final state of the qubits, after the circuit has acted, is easily computed since it is the corresponding linear superposition of outputs for each of the computational basis state inputs.

Circuits have several gates acting in succession on a qubit and it is important to understand the order in which they act. Unfortunately, this can be confusing. By convention, in diagrams time is from left to right, so in the diagram



A (the leftmost gate) acts first then B. However, when writing operator expressions, these work from right to left, so, the above diagram corresponds to

$$|f\rangle = BA|i\rangle, \tag{8.5}$$

in which A is on the *right*. You simply have to get used to this reversal of order when going from circuit diagrams to operator expressions.

Now we describe some commonly used quantum gates, recalling that quantum gates are unitary operators and so must be reversible.

Firstly we consider 1-qubit gates.

- NOT, i.e. bit-flip (corresponds to the Pauli X matrix)

$$\begin{array}{l} X|0\rangle = |1\rangle \\ X|1\rangle = |0\rangle \end{array}, \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{8.6}$$

- Phase flip (corresponds to the Pauli Z matrix)

$$\begin{array}{l} Z|0\rangle = |0\rangle \\ Z|1\rangle = -|1\rangle \end{array}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{8.7}$$

  In the physics literature X and Z are called Pauli spin matrices. There is also a third Pauli spin matrix, Y, where

$$Y = iXZ = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{8.8}$$

  which corresponds to a combined bit- and phase-flip. The Pauli Y-matrix will appear again when we do quantum error correction in Chapter 19.

- Hadamard
  The Hadamard gate H will be very important.

$$H = \frac{1}{\sqrt{2}}(X+Z) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{8.9}$$

  Note that $H^2 = \mathbb{1}$, and similarly $X^2 = Y^2 = Z^2 = \mathbb{1}$.

  Now a matrix which squares to the identity has eigenvalues ± 1. To see this note that if $\vec{x}$ is an eigenvector of A with eigenvalue λ then

$$A^2\vec{x} = A\left(A\vec{x}\right) = A\lambda\vec{x} = \lambda A\vec{x} = \lambda^2\vec{x}. \tag{8.10}$$

  But if $A^2 = \mathbb{1}$ then it follows that $\lambda^2 = 1$ and so $\lambda = \pm 1$.

  We need to become familiar with the action of H on computational basis states. This is:

$$\begin{aligned} H|0\rangle =& |+\rangle \equiv \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \\ H|1\rangle =& |-\rangle \equiv \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right). \end{aligned} \tag{8.11}$$

  Combining these two equations, the action of H on a computational basis state $|x\rangle$ is seen to be

$$H|x\rangle = \frac{1}{\sqrt{2}}\left(\,|0\rangle + (-1)^x|1\rangle\,\right), \tag{8.12}$$

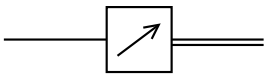
  for both values of x, namely 0 and 1.

A crucial point is that these gates are linear, and so they act in the same way on a superposition. For example:

$$H\left[\,\alpha|0\rangle + \beta|1\rangle\,\right] = \frac{\alpha}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) + \frac{\beta}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) = \left(\frac{\alpha+\beta}{\sqrt{2}}\right)|0\rangle + \left(\frac{\alpha-\beta}{\sqrt{2}}\right)|1\rangle. \tag{8.13}$$

We also need to consider measurement gates, in which a classical measurement of a qubit takes place. By convention, measurements are made in the computational basis. The Pauli spin matrices are defined such that Z is diagonal, so we can also call the computational basis the Z-basis.

The result of the measurement is a classical bit. In the circuit diagrams we indicate a classical bit by a double line, and so a measurement gate is indicated as follows:



A measurement gate

The measurement apparatus acting on a qubit determines the value of Z for that qubit, obtaining either $+1$, in which case the qubit is left in state $|0\rangle$, or -1, in which case the qubit is left in state $|1\rangle$. If one wants to measure the value of some other quantity one needs to perform an appropriate unitary transformation first. For example, to determine the value of X one acts with a Hadamard before the measurement, since the Hadamard converts the X-basis to the Z- basis and vice-versa.

Next we consider 2-qubit gates, the most important of which by far is the CNOT. We already met the classical CNOT gate in Fig. 8.1. In the quantum case, if initially the qubits are in a computational basis state, then the action of the CNOT is the same as classically, i.e. as shown in Fig. 8.5.

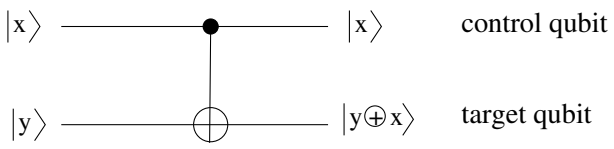


Figure 8.5: A quantum CNOT gate. If the initial state of the qubits (on the left) is a computational basis state, then the action of the quantum CNOT gate is the same as that of the classical CNOT shown in Fig. 8.1. The upper line represents the control qubit and the lower line the target qubit.

The CNOT gate has the matrix representation

$$U_{\rm CNOT} = \begin{array}{c} \\ \langle 00| \\ \langle 01| \\ \langle 10| \\ \langle 11| \end{array} \begin{array}{c} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} . \tag{8.14}$$

In this tensor product the control qubit is the one to the left. The target qubit (to the right) is flipped if the control qubit is 1 (so, relative to the identity matrix, columns 3 and 4 are interchanged). We can also write $U_{\rm CNOT}$ in terms of 2×2 blocks as follows

$$U_{CNOT} = \begin{pmatrix} \mathbb{1} & 0 \\ 0 & X \end{pmatrix} . \tag{8.15}$$

The quantum aspect appears if we input (on the left) a linear combination of basis states. Suppose, for example, we set the target (lower) qubit to $|0\rangle$. Then if the control qubit is initially $|0\rangle$ the final state of the 2-qubit system is $|00\rangle$, because the target qubit is not flipped and stays as $|0\rangle$ (we take the control qubit to be the left one). If the control qubit is initially $|1\rangle$ then the final state of the 2-qubit system is $|11\rangle$ because the target qubit *is* flipped from $|0\rangle$ to $|1\rangle$. Hence, by linearity, if the initial state of the control qubit is the superposition $\alpha|0\rangle + \beta|1\rangle$, then the final state of the 2-qubit system is $\alpha|00\rangle + \beta|11\rangle$, see Fig. 8.6.

Note that if $\alpha = 0$ (so $\beta = 1$ since $|\alpha|^2 + |\beta|^2 = 1$) or $\alpha = 1$ ($\beta = 0$), the final state is a clone of the initial state of the control qubit. However, for a general input state, the final state of the two qubits, $\alpha|00\rangle + \beta|11\rangle$, is not a clone of the initial state of the control qubit which would be $(\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle) = \alpha^2|00\rangle + \alpha\beta(|01\rangle + |10\rangle) + \beta^2|11\rangle$. Hence there is no violation of the no-cloning theorem which states that a *general*, unknown quantum state can not be cloned.

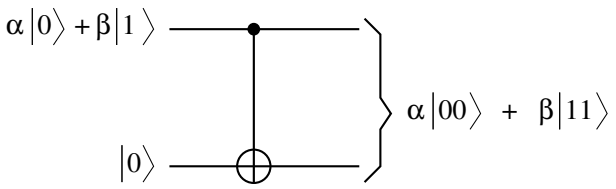


Figure 8.6: The action of the CNOT gate when the upper (control) qubit is initially in a superposition $\alpha|0\rangle + \beta|1\rangle$, and the lower (target) qubit is initially $|0\rangle$. By linearity, the final state is α times the result of inputting $|0\rangle$ in the control qubit plus β times the result of inputting $|1\rangle$, i.e. $\alpha|00\rangle + \beta|11\rangle$. We see that the final state is *entangled*.

In this course, we will specify the action of a gate by its action on an initial computational basis state. If we denote a qubit by a Latin letter, e.g. $|x\rangle$, we mean that this is a computational basis state and x takes values 0 or 1. General quantum states, i.e. superpositions of computational basis states, will be indicated by Greek letters, e.g. $|\psi\rangle$.

As already mentioned above, we do not need 3-qubit gates for quantum computing. More precisely, the statement is that one can generate an arbitrary unitary transformation (to a specified level of accuracy) on a arbitrary number of qubits, using only CNOT and single-qubit gates. I do not prove this result but refer interested students to a more advanced text [NC00]. It is fortunate that we don't need 3-qubit gates given the difficulty of making quantum circuits.

It is useful to mention here that one has to be careful when dealing with superpositions, and one's initial intuition as to the final result may be incorrect. As an example, consider the circuit in Fig. 8.7.

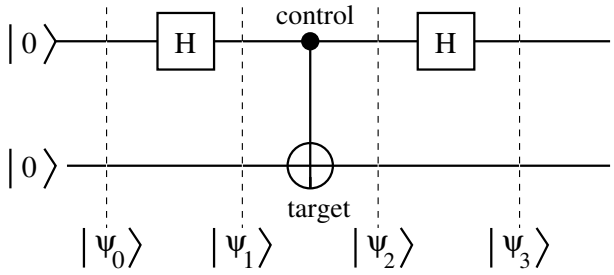


Figure 8.7: The initial state of both qubits is $|0\rangle$. What is the final state $|\psi_3\rangle$? Equation (8.16) gives the state of the two qubits at each stage. The end result is that the two qubits are entangled and, in contrast to what one might have thought, the control (upper) qubit has a non-zero amplitude to be flipped relative to its initial state, i.e. to be in state $|1\rangle$.

Since $H^2 = \mathbb{1}$ and the CNOT gate doesn't change the control (upper) qubit, one might think that the final state of the control qubit would be the same as the initial state, i.e. $|0\rangle$. However this is not correct because the control and target qubits become entangled. Let's go through each stage of the circuit using the notation in Fig. 8.7, and taking the left-hand qubit in the formulae to be the control

qubit:

$$
\begin{aligned}
|\psi_0\rangle &= |00\rangle \\
|\psi_1\rangle &= \frac{1}{\sqrt{2}}\left(\,|00\rangle + |10\rangle\,\right) \\
|\psi_2\rangle &= \frac{1}{\sqrt{2}}\left(\,|00\rangle + |11\rangle\,\right) \\
|\psi_3\rangle &= \frac{1}{2}\left(\,|00\rangle + |10\rangle + |01\rangle - |11\rangle\,\right) \\
&= \frac{1}{\sqrt{2}}\left[\,|0\rangle_c \otimes \left(\frac{|0\rangle_t + |1\rangle_t}{\sqrt{2}}\right) + |1\rangle_c \otimes \left(\frac{|0\rangle_t - |1\rangle_t}{\sqrt{2}}\right)\right],
\end{aligned}
\tag{8.16}
$$

where in the last expression we indicate explicitly which qubit is the control qubit ("c"), and which the target qubit ("t"). We see that, contrary to what one might have initially guessed, there is an amplitude for the control qubit to be in state $|1\rangle$ because of its entanglement with the target qubit.

We have noted that the Pauli operators X, Y and Z, and the Hadamard operator have eigenvalues ± 1. Later in the course, when we consider the important topic of quantum error correction, we will encounter combinations of these operators on different qubits which also have ± 1 eigenvalues. We will now describe a convenient way of measuring such operators. Let us denote the operator by U. It will have matrix elements given by

$$
U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix} \tag{8.17}
$$

and eigenvalue $+1$ with eigenvector $|\psi_+\rangle$ and an eigenvalue -1 with eigenvector $|\psi_-\rangle$. We would like to investigate the qubit (or qubits) to determine which eigenstate of U it is in, or, if it is in a linear superposition, to project by measurement on to one of the eigenstates, and know which one.

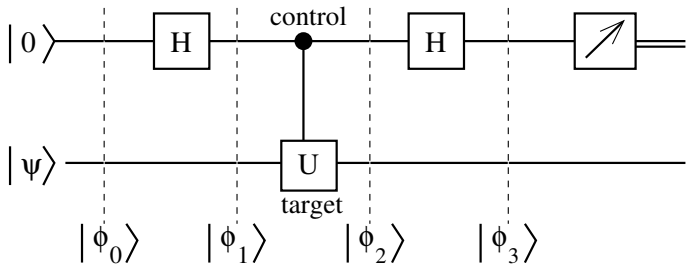


Figure 8.8: A circuit with a control-U gate in which the control (upper) qubit is surrounded by Hadamards. U is an operator with eigenvalues ± 1 and corresponding eigenvectors $|\psi_+\rangle$ and $|\psi_-\rangle$. As shown in the text, if a measurement of the upper qubit gives $|0\rangle$ then the lower qubit will be in state $|\psi_+\rangle$, and if the measurement gives $|1\rangle$ then the lower qubit will be in state $|\psi_-\rangle$. The states $|\phi_i\rangle$ $(i = 0, 1, 2, 3)$ are described in the text.

A convenient way is to use the circuit shown in Fig. 8.8, which has a control-U gate[1]. The matrix

[1] Apart from the absence of the final measurement gate, Fig. 8.7 is a special case of Fig. 8.8 with $U = X$ (since the NOT gate is effected by the X operator).

representation of control-U is

$$\text{control}-U = \begin{array}{c} \\ \langle 00| \\ \langle 01| \\ \langle 10| \\ \langle 11| \end{array} \begin{array}{c} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix} \end{array} = \begin{pmatrix} \mathbb{1} & 0 \\ 0 & U \end{pmatrix}, \tag{8.18}$$

where the last expression is written in terms of 2×2 blocks. If the control qubit is 1 the effect on the target qubit can be represented in terms of eigenstates of U by

$$U|\psi_+\rangle = |\psi_+\rangle, \quad U|\psi_-\rangle = -|\psi_-\rangle. \tag{8.19}$$

If the control qubit is 0 then the target qubit is unchanged.

The lower (target) qubit is initially in state $|\psi\rangle$, which can be written as a linear combination of the two eigenvectors

$$|\psi\rangle = \alpha_+|\psi_+\rangle + \alpha_-|\psi_-\rangle\,, \tag{8.20}$$

and so, including the upper (control) qubit which is initially in state $|0\rangle$, the initial state of the circuit (on the left of Fig. 8.8) is

$$|\phi_0\rangle = \alpha_+|0\,\psi_+\rangle + \alpha_-|0\,\psi_-\rangle\,. \tag{8.21a}$$

In labeling the states, we put the state of the control qubit to the left and that of the target qubit to the right. After the first Hadamard on the upper qubit the state is

$$|\phi_1\rangle = \frac{\alpha_+}{\sqrt{2}}\,(\,|0\,\psi_+\rangle + |1\,\psi_+\rangle\,) + \frac{\alpha_-}{\sqrt{2}}\,(\,|0\,\psi_-\rangle + |1\,\psi_-\rangle\,)\,. \tag{8.21b}$$

The effect of the control-U gate on the target qubit is given by Eq. (8.19) when the control qubit is 1 and has no effect if the control qubit is 0. Hence, after the control-U gate, the state is

$$|\phi_2\rangle = \frac{\alpha_+}{\sqrt{2}}\,(\,|0\,\psi_+\rangle + |1\,\psi_+\rangle\,) + \frac{\alpha_-}{\sqrt{2}}\,(\,|0\,\psi_-\rangle - |1\,\psi_-\rangle\,)\,. \tag{8.21c}$$

Applying the righthand Hadamard in Fig. 8.8 to the upper (control) qubit we get

$$|\phi_3\rangle = \alpha_+|0\,\psi_+\rangle + \alpha_-|1\,\psi_-\rangle\,. \tag{8.21d}$$

Hence if a measurement of the upper qubit gives $|0\rangle$ (which it does with probability $|\alpha_+|^2$) the lower qubit will be in state $|\psi_+\rangle$, and if the measurement gives $|1\rangle$ (probability is $|\alpha_-|^2$) the lower qubit will be in state $|\psi_-\rangle$. We see that measuring the control qubit projects the target qubit onto an eigenstate of U and tells us which one.

We will return to the circuit in Fig. 8.8 in Chapter 19 when we discuss quantum error correction.

Chapter 9

Generating and measuring Bell States

Entangled states play an important role in quantum computing. The most-studied entangled states are so-called Bell states which involve two qubits. They are named in honor of the physicist John Bell who clarified the Einstein-Podolsky-Rosen (EPR) paradox, discussed in Chapter 7, and whose inequalities demonstrated that the description of nature provided by quantum mechanics is fundamentally different from the classical description. The Bell states are defined by

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\left(\,|00\rangle + |11\rangle\,\right), \tag{9.1a}$$

$$|\beta_{01}\rangle = \frac{1}{\sqrt{2}}\left(\,|01\rangle + |10\rangle\,\right), \tag{9.1b}$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}\left(\,|00\rangle - |11\rangle\,\right), \tag{9.1c}$$

$$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}\left(\,|01\rangle - |10\rangle\,\right). \tag{9.1d}$$

These four equations can be combined as follows:

$$|\beta_{xy}\rangle = \frac{1}{\sqrt{2}}\left(\,|0y\rangle + (-1)^x|1\overline{y}\rangle\,\right), \tag{9.2}$$

where $\overline{y}$ is the complement of y, i.e. $\overline{y} = 1 - y$.

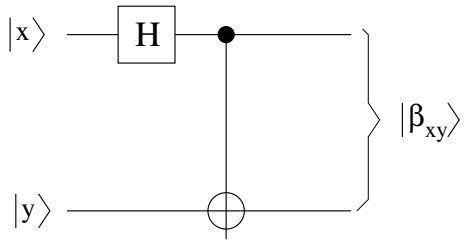


Figure 9.1: Circuit to create the Bell states defined by Eqs. (9.1). In the CNOT gate the upper qubit $|x\rangle$ is the control qubit and the lower qubit $|y\rangle$ is the target qubit.

The Bell states are clearly entangled. They can be created out of two (unentangled) qubits in computational basis states $|xy\rangle$ by the circuit shown in Fig. 9.1. To see this note that, according to Eq. (8.12), after the Hadamard the state is

$$|xy\rangle \to \frac{1}{\sqrt{2}}\left(\,|0y\rangle + (-1)^x|1y\rangle\,\right). \tag{9.3}$$

The effect of the CNOT gate is to flip y in the second term (since $x = 1$ there) and so we get Eq. (9.2)

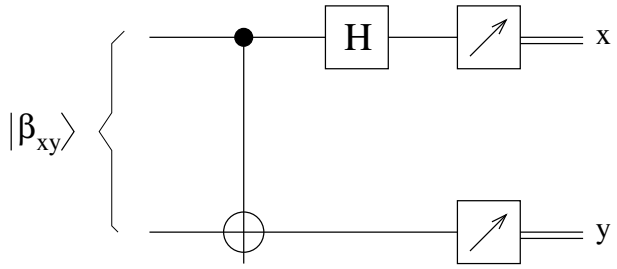


Figure 9.2: Circuit for Bell measurements. This will be used later in the course when we discuss teleportation.

The circuit in Fig. 9.1 converts the computational basis to the Bell basis. The reverse of this circuit can be used to convert the Bell basis back to the computational basis as shown in Fig. 9.2. The measured values of x and y tell us which Bell state we started with. This is called a *Bell Measurement*. To see that this works note that after the CNOT gate the state of the two qubits in Fig. 9.2 is[1]

$$\frac{1}{\sqrt{2}}\left[\,|0y\rangle + (-1)^x|1y\rangle\,\right], \tag{9.4}$$

which is separable and so can be written as

$$\frac{1}{\sqrt{2}}\left[\,|0\rangle + (-1)^x|1\rangle\,\right] \otimes |y\rangle. \tag{9.5}$$

Recall that the left-hand qubit is the upper (control) qubit in Fig. 9.2 and the right hand qubit is the lower (target) qubit. Acting with the Hadamard has the effect

$$H\frac{1}{\sqrt{2}}\left[\,|0\rangle + (-1)^x|1\rangle\,\right] = |x\rangle\,, \tag{9.6}$$

so the final state in Fig. 9.2 is $|xy\rangle$ as desired.

Note that the Bell states $|\beta_{xy}\rangle$ provide a basis for two qubits, see Appendix 4.A in Chapter 4, since they are normalized, mutually orthogonal and linearly independent. Consequently, if the state inputted into the Bell measurement circuit in Fig. 9.2 is not a single Bell state, but rather a linear combination,

$$|\psi_{\rm in}\rangle = \sum_{x,y=0}^{1} \alpha_{xy}|\beta_{xy}\rangle, \tag{9.7}$$

with $\sum_{x,y} |\alpha_{xy}|^2 = 1$, then the probability that the measurements obtain a particular set of values for x and y is $|\alpha_{xy}|^2$.

[1]The reason that $\overline{y}$ in the Bell state, Eq. (9.2), changes to y in the second term in Eq. (9.4) is because $x = 1$ and so the y (target) qubit is flipped.

Chapter 10

Quantum Functions

10.1 An elementary quantum function

In computation we need to evaluate functions. How can we do this in a quantum computer where functions are determined by unitary transformations which are reversible?

Let us first consider the simplest case, where the argument of the function, x, is a single bit, and the result of the function, $f(x)$, is also a single bit. In other words, x takes only the values 0 and 1, and the same for $f(x)$. We need to have a qubit for x and an *additional* qubit[1] which contains information on the function $f(x)$.

The function $f(x)$ will be implemented by a unitary operator U_f acting on two qubits such that

$$U_f|x\rangle|y\rangle = |x\rangle|f(x) \oplus y\rangle. \tag{10.1}$$

Note the similarity with the CNOT gate, which is precisely of this form with $f(x) = x$. It is easy to see that $U_f^2 = \mathbb{1}$ since

$$U_f^2|x\rangle|y\rangle = U_f|x\rangle|f(x) \oplus y\rangle = |x\rangle|f(x) \oplus f(x) \oplus y\rangle = |x\rangle|y\rangle \tag{10.2}$$

since, as discussed earlier in the course, $f(x) \oplus f(x) = 0$. Hence U_f has an inverse, which is U_f itself.

The corresponding circuit diagram is shown in Fig. 10.1

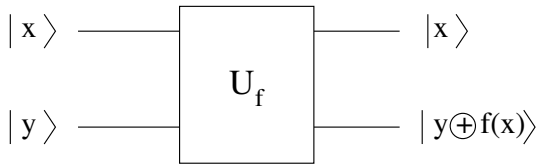


Figure 10.1: Schematic diagram of a unitary transformation U_f for a function $f(x)$ in which both the argument x and the function just take two values, 0 and 1.

For a general function, the range of inputs can be represented by n bits, say, and the range of outputs by m bits. Thus we need a total of $n + m$ qubits both in the initial state and final state. The unitary transformation is

$$U_f|x\rangle_n|y\rangle_m = |x\rangle_n|f(x) \oplus y\rangle_m, \tag{10.3}$$

[1]We need to have two qubits in both the initial and final states in order that the function is reversible, just as we needed two qubits in the final state as well as the initial state to make the CNOT gate which is a reversible generalization of the XOR gate, see Chapter 8.

where the modulo 2 addition, indicated by $\oplus$, applies separately to each of the m bits of $f(x)$ and y. The proof that U_f is its own inverse is the same as that in Eq. (10.2). The circuit diagram corresponding to Eq. (10.3) is shown in Fig. 10.2.

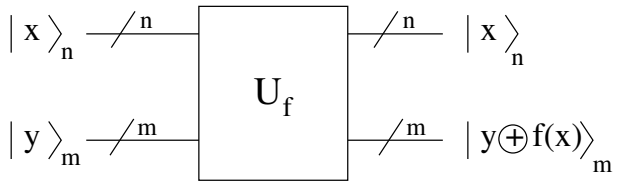


Figure 10.2: Schematic diagram of a general unitary transformation U_f for an n-bit input x and an m-bit output $f(x)$. The upper register in the figure has n qubits and contains the input value x. The lower register has m qubits and contains information about the function value $f(x)$. The registers are shown as single lines. To ensure the transformation is reversible there are $n + m$ qubits in both the initial state (to the left) and final state (to the right).

One sometimes calls the upper register in Fig. 10.2 the "input" register, because it contains the input, x, and the lower register the "output register" because it contains information on the function $f(x)$. However, since both registers are present in the initial state (on the left) and the final state (on the right) this terminology can be confusing.

Note that if $y = 0$ the lower register contains precisely the function $f(x)$.

10.2 Quantum Parallelism

Things get interesting if we feed in a superposition. We can generate a uniform superposition by acting with Hadamards on $|0\rangle_n$. Note that for one qubit

$$H|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \tag{10.4}$$

and similarly applying a Hadamard to each of two qubits

$$\begin{aligned} H|0\rangle \otimes H|0\rangle &= \frac{1}{2}\left(|0\rangle + |1\rangle\right) \otimes \left(|0\rangle + |1\rangle\right) \\ &= \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right) \\ &= \frac{1}{2}\left(|0\rangle_2 + |1\rangle_2 + |2\rangle_2 + |3\rangle_2\right) = \frac{1}{2}\sum_{x=0}^{3}|x\rangle_2. \end{aligned} \tag{10.5}$$

Here we have used the convenient notation that $|x\rangle_n$ is an n-qubit state in which the values of the individual qubits are the bits of the n-bit integer x. Generalizing we have

$$H^{\otimes n}|0\rangle_n = \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x\rangle_n. \tag{10.6}$$

Now lets consider the circuit shown in Fig. 10.3. The initial state is

$$|\phi_0\rangle = |0\rangle_n|0\rangle_m, \tag{10.7}$$

so the state fed into the unitary operator U_f is the superposition

$$|\phi_1\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n |0\rangle_m. \tag{10.8}$$

Noting that the lower register is initialized to $|0\rangle$, then by linearity, according to Eq. (10.3), the final state must be

$$|\phi_2\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n |f(x)\rangle_m. \tag{10.9}$$

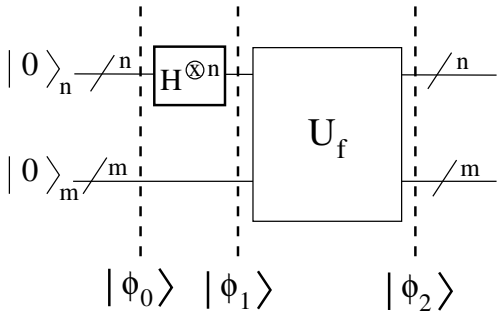


Figure 10.3: Because of the Hadamards, the input to U_f is now the uniform superposition of all computational basis states in Eq. (10.6). The output from U_f is given by Eq. (10.9).

This is an astonishing result. The final state contains the function values for *all* 2^n possible values of the input x. They have been evaluated in parallel, a feature of quantum mechanics called, naturally enough, "quantum parallelism". For $n = 100$ we have $2^{100} \simeq 10^{30}$ function evaluations in parallel.

A speedup of 10^{30} seems to good to be true, and, unfortunately, it is. What's the catch? The catch is that the only way one can access the information contained in the state is to do a measurement. This does not give 10^{30} results but just one result, the probabilities of the different results being the square of the amplitudes (which are all equal here so there is a probability $1/2^n$ of getting each of the 2^n possible states). So, it seems that we have achieved nothing. We have found the value of the function for one value of its argument, which we could have got much more easily on a classical computer. However, for some problems, one can gain enough useful information to get a "quantum speedup" by doing clever pre-processing *before* the measurement. How to achieve this in practice will occupy us for most of the rest of the course.

Philosophers, and some physicists, debate whether one can really state that all 2^n values of the function have been evaluated since one can not observe them. Most physicists would argue that the only "real" quantities are those that can be observed, and, in particular, the quantum mechanical state itself is not real. From this point of view, it is not valid to claim that all 2^n values of the function have actually been evaluated.

Now we have done enough preliminaries to study our first quantum algorithm! This will be described in the next chapter.

Chapter 11

Deutsch's Algorithm

11.1 Introduction

We now turn to our first algorithm, due to David Deutsch [Deu85] which is generally felt to have started the field of quantum computing.

As we shall see the problem is very trivial. It concerns functions which takes a 1-qubit argument and give a 1-qubit output. The problem is clearly contrived and is of no practical interest. However, it does show a quantum speedup, and this arises from the same features of quantum circuits, namely *quantum parallelism* and *interference*, used in more sophisticated and useful quantum algorithms such as that of Shor.

Since the input takes one of two values, 0 and 1, as does the output, there are only four distinct functions as shown in the table.

	$x = 0$	$x = 1$
f_1	0	0
f_2	0	1
f_3	1	0
f_4	1	1

Table 11.1: The four functions which have a 1-qubit input and a 1-qubit output.

You see that f_1 and f_4 gave the same result for each input, they are *constant*. On the other hand, f_2 and f_3 give *different* results for the two inputs. This is analogous to a coin toss. The two values of x correspond to the two physical sides of the coin, the upper and the lower sides. The function values correspond to what is represented on those sides, heads or tails. If the two sides of the coin give different results (one heads and the other tails), corresponding to a non-constant function, the coin is honest. However, if the two sides of the coin give the same result (both heads or both tails) the coin is dishonest, corresponding to a constant function.

We are given a "black box"[1] function $f(x)$ and we want to learn about it. Of course we could just feed in $x = 0$ and $x = 1$ and observe the results. Suppose, however, we only want to know whether the function is constant (satisfied by f_1 and f_4) or not constant (satisfied by f_2 and f_3). On a classical computer the only thing to do is to evaluate the function for both values of x and compare them, i.e. we need to make *two* calls to the function. However, we shall see that we can answer this question on a quantum computer with only *one* call to the function. We get less information than classically,

[1]The term "black box" implies that the only information we can get about the function is by evaluating it for different inputs. We can't open up the box to see what is inside. A black box function is often called an "oracle".

because we don't determine the individual values of $f(0)$ and $f(1)$, but we do determine whether or not f is constant. Hence Deutsch's problem may be thought of as determining whether a coin to be tossed is honest or not *with just one toss of the coin.*

As we discussed in Chapter 10, a quantum function f is implemented by a unitary operator U_f as shown in Fig. 11.1. You will recall that we need to have the same number of qubits in the initial and finite states in order to make the function reversible. The function U_f in Fig. 11.1 is its own inverse as can be seen since $U_f^2 = \mathbb{1}$, the identity. If this is not obvious note that $y \oplus f(x) \oplus f(x) = y$. (Adding 0 to 0 gives 0 and adding 1 to 1 mod 2 also gives 0.)

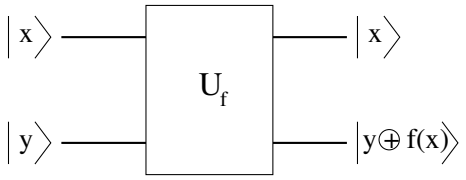


Figure 11.1: The blackbox routine U_f for a function $f(x)$ which takes a 1-qubit input x and computes a 1-qubit function $f(x)$. Here x and y are computational basis states $|0\rangle$ or $|1\rangle$. However, to gain a quantum speedup, we will input superpositions, generated by Hadamard gates, as shown in Fig. 11.2. We obtain the result of inputing a superposition from the results of inputing computational basis states by using linearity. Recall that time runs from left to right in circuit diagrams.

In order to take advantage of quantum parallelism we insert Hadamard gates before the black box function U_f on both the upper (input) and lower (output) qubits, and to take advantage of quantum interference of the results we will also put Hadamards on both qubits after U_f has acted[2], see Fig. 11.2. We initialize the upper qubit to be $|0\rangle$ and the lower qubit to be $|1\rangle$. The upper qubit could be initialized to either $|0\rangle$ or $|1\rangle$ but it is essential to initialize the lower qubit to $|1\rangle$ as we shall see.

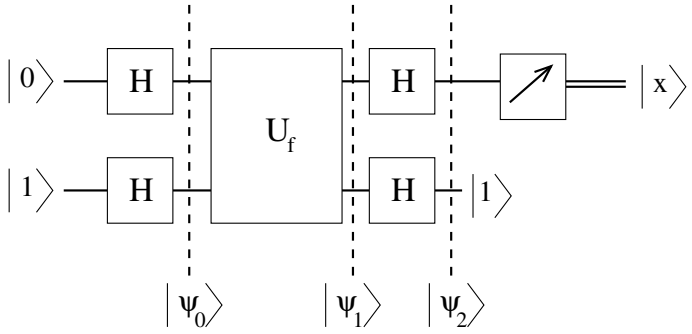


Figure 11.2: Circuit for Deutsch's algorithm. The initial state (on left) has $|0\rangle$ in the upper (input) qubit and $|1\rangle$ in the lower (output) qubit. Hadamard gates are applied to both qubits both before and after the function U_f (which we assume to be an unknown black box). In the final state the lower qubit is unchanged at $|1\rangle$. A measurement is made of the final value (on right) of the upper qubit. If this is unchanged, i.e. $x = 0$ in this case, then the function is constant, while if the upper qubit has flipped, then the function is not constant. One could equivalently start with the upper qubit as $|1\rangle$ and find the same conclusion: namely if the upper qubit is unchanged the function is constant whereas if it has flipped the function is not constant.

[2]This is actually an improved version of Deutsch's original algorithm. The improved version works every time, whereas the original version only worked half the time.

Recalling that

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \tag{11.1}$$

we find that after the first Hadamards the state in Fig. 11.2 is

$$\begin{aligned} |\psi_0\rangle &= \frac{1}{2}((|0\rangle_u + |1\rangle_u) \otimes (|0\rangle_l - |1\rangle_l)\,, \\ &= \frac{1}{2}|0\rangle_u \otimes (|0\rangle_l - |1\rangle_l) + \frac{1}{2}|1\rangle_u \otimes (|0\rangle_l - |1\rangle_l)\,, \end{aligned} \tag{11.2}$$

where, in the tensor product, the the upper qubit (labeled "u") is to the left and the lower qubit (labeled "l") is to the right.

The function U_f is then applied. Recall from Fig. 11.1 that if the state of the upper qubit is x, then the final state of the lower qubit is $f(x)$ if its initial state is zero, and the complement $\overline{f(x)}$ if its initial state is one. Hence, after U_f has been applied, the state is

$$|\psi_1\rangle = \frac{1}{2}|0\rangle_u \otimes (\,|f(0)\rangle_l - |\overline{f(0)}\rangle_l\,) + \frac{1}{2}|1\rangle_u \otimes (\,|f(1)\rangle_l - |\overline{f(1)}\rangle_l\,) \tag{11.3}$$

It is helpful to note that

$$\begin{aligned} |f(x)\rangle_l - |\overline{f(x)}\rangle_l &= \begin{cases} |0\rangle_l - |1\rangle_l & \text{if } f(x) = 0, \\ |1\rangle_l - |0\rangle_l & \text{if } f(x) = 1, \end{cases} \\ &= (-1)^{f(x)}(\,|0\rangle_l - |1\rangle_l\,)\,. \end{aligned} \tag{11.4}$$

Hence whether or not $f(x) = 0$ or $f(x) = 1$ just changes the overall sign of the state. To get this effect it was necessary to prepare the lower qubit in state $|1\rangle$ rather than $|0\rangle$. Vathsan [Vat16] calls Eq. (11.4) "*phase kickback*". Consequently we can write $|\psi_1\rangle$ as

$$|\psi_1\rangle = \frac{(-1)^{f(0)}|0\rangle_u + (-1)^{f(1)}|1\rangle_u}{\sqrt{2}} \otimes \frac{|0\rangle_l - |1\rangle_l}{\sqrt{2}}\,. \tag{11.5}$$

Now we run both qubits through Hadamards (those to the right of U in Fig. 11.2). It is easy to see that action on the lower qubit (right hand one in the tensor product) is to convert $\frac{1}{\sqrt{2}}(|0\rangle_l - |1\rangle_l)$ back to $|1\rangle_l$. The action of H on the upper qubit is to give

$$\frac{1}{2}\Big[\,(-1)^{f(0)}(|0\rangle_u + |1\rangle_u) + (-1)^{f(1)}(|0\rangle_u - |1\rangle_u)\,\Big] \tag{11.6}$$

which can be written as

$$\frac{1}{2}|0\rangle_u\Big[\,(-1)^{f(0)} + (-1)^{f(1)}\,\Big] + \frac{1}{2}|1\rangle_u\Big[\,(-1)^{f(0)} - (-1)^{f(1)}\,\Big]\,. \tag{11.7}$$

Clearly this is $\pm|0\rangle_u$ if $f(0) = f(1)$ (where the plus sign is for $f(0) = f(1) = 0$ and the minus sign for $f(0) = f(1) = 1$), and is $\pm|1\rangle_u$ if $f(0) \neq f(1)$ (where the sign depends on whether $f(0) = 1, f(1) = 0$ or vice versa). Hence the state to the right of the Hadamards in Fig. 11.2 is

$$|\psi_2\rangle = \begin{cases} \pm|0\rangle_u \otimes |1\rangle_l & \text{if } f(1) = f(0)\,, \\ \pm|1\rangle_u \otimes |1\rangle_l & \text{if } f(1) \neq f(0)\,. \end{cases} \tag{11.8}$$

Consequently, measuring the upper qubit in Fig. 11.2 (left in the tensor product) tells us that $f(0) = f(1)$ if it is unchanged[3] from the initial state, i.e. if we get $|0\rangle$, and that $f(0) \neq f(1)$ if the upper qubit

[3]It does not matter whether we initialize the upper qubit to be 0 or 1, the conclusion is the same. Namely, if the upper qubit is unchanged, then the function is constant, whereas if it is flipped the function is non-constant.

is flipped, i.e. if we get $|1\rangle$. We do this with *one* call to the function so we have achieved a "quantum speedup" of 2, which is admittedly not spectacular but it is interesting that we get any speedup at all. We will get more impressive speedups in later algorithms.

If we could measure the sign of the state we could determine the values of $f(0)$ and $f(1)$ separately but the sign of the state (more generally its phase) has no effect and can not be determined.

A crucial role has been played by the Hadamards. Those which act before U is called generate a superposition state with both inputs $x = 0$ and 1 present. Looking at Eq. (11.3) it "seems" that U has computed $f(x)$ for both values of x with just one call to it. This is "*quantum parallelism*". If we do a measurement directly after the application of U we only get one value. However, for certain problems like this one, if we do some additional post-processing (in this case acting with Hadamards again), we can use "*quantum interference*" between the different pieces in the superposition to set to zero the probability of getting certain results (in this case all possible results bar one are suppressed). Consequently it is possible to get useful information (in this case whether the function is constant or not) when the measurement is *subsequently* done.

Note that the Deutsch algorithm is not probabilistic: **it succeeds with probability 1.** This shows that quantum algorithms don't necessarily have to be probabilistic (though many are). In this case, quantum interference transforms the state to be measured into an eigenstate of the computational basis. As we know, if we measure an eigenstate we always get the same answer (the eigenvalue) and there is no uncertainty.

Appendices

11.A An alternative derivation

To familiarize ourselves with quantum circuits we will obtain Eq. (11.8) in a different way by explicitly writing down circuits for the four functions f_1 to f_4, see Fig. 2.1 of Mermin [Mer07]. Noting that the function flips the lower (output) qubit if the result of the function is 1 but leaves it alone if the function gives 0, we can represent the four functions in Table 11.1 by the circuits shown in Fig. 11.3. Explanations of why each circuit is equivalent to the corresponding function are given in the figure caption. We sandwich each of these functions between Hadamards to carry out the Deutsch algorithm, as shown in Fig. 11.2, and prepare the qubits in the initial state $|x\rangle \otimes |1\rangle$. The results are shown in Fig. 11.4. It is important to understand each of the diagrams in this figure.

- $\mathbf{f_1}$:
  This follows simply because U_{f_1} makes no change, see Fig. 11.3, and $H^2 = \mathbb{1}$ (the identity), see Fig. 11.5(a) in Appendix 11.B, so the final qubits are the same as the initial qubits, $|x\rangle \otimes |1\rangle$, see Fig. 11.4. In particular x is unchanged indicating, correctly that the function is constant.

- $\mathbf{f_2}$:
  The function U_{f_2} has a CNOT gate in which the upper qubit is the control and the lower qubit is the target, see Fig. 11.3. The result of sandwiching a CNOT between Hadamards is, perhaps surprisingly, to interchange the role of the target and control qubits. This is shown in Appendix 11.B, see Fig. 11.5(f). Hence we see that x is flipped because the lower qubit is set to $|1\rangle$, see Fig. 11.4. This is correct because the function is not constant.

- $\mathbf{f_3}$:
  The circuit for U_{f_3} is shown in Fig. 11.3. Noting that $H^2 = \mathbb{1}$, one can insert two Hadamards between the two X gates in the circuit for U_{f_3} in Fig. 11.3. As we noted for $\mathbf{f_2}$, the effect of putting Hadamards on either side of the CNOT gate is to interchange the role of the target and

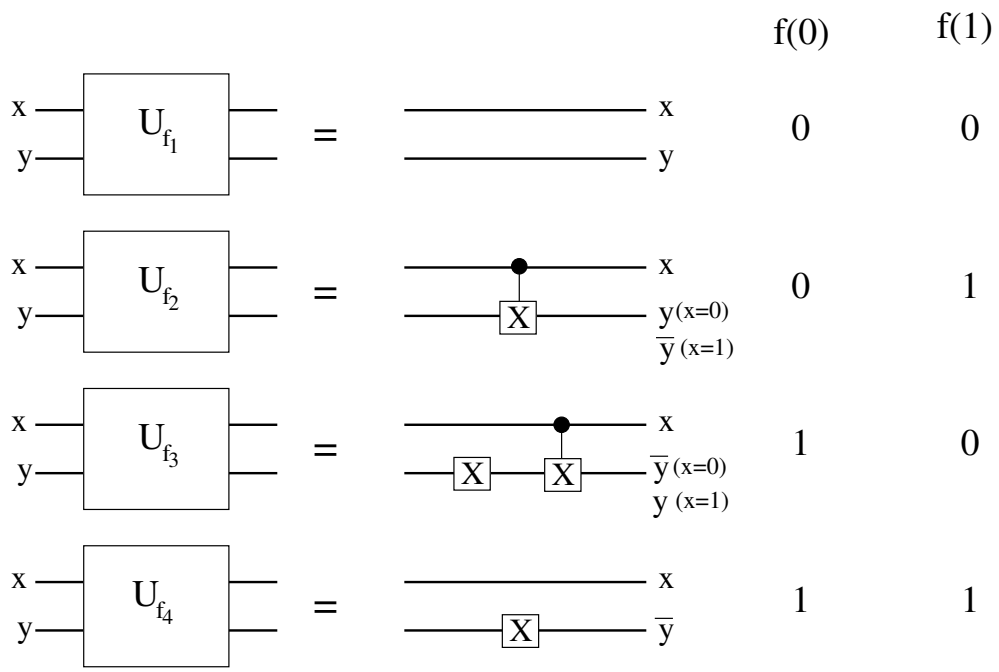


Figure 11.3: Circuit diagrams for each of the four functions $f_1, \cdots, f_4$ in Table 11.1. As seen in Fig. 11.1, the function flips the lower (output) qubit if the result of the function is 1 but leaves it alone if the function gives 0. If $f(x) = 0$, y is unchanged but if $f(x) = 1$ then y is flipped and becomes $\overline{y}$, the complement. Note that x is always unchanged. For example, with f_1 (top diagram), nothing happens. For f_4, y is always flipped which is done with the X gate on the lower qubit. For f_2, y is only flipped if $x = 1$ which is done by the CNOT gate as shown. For f_3, y is only flipped if $x = 0$ which can be accomplished by the extra X gate on the y-qubit.

control qubits. In addition, we have $HXH = Z$, see Fig. 11.5(b) in the Appendix 11.B. Hence x is flipped and there is a sign change, see Fig. 11.4. We can't measure the sign change but the fact that x is flipped correctly indicates that the function is not constant.

- **f_4**:
  The function U_{f_4} has an X gate on the lower qubit, see Fig. 11.3, and again we have $HXH = Z$. Hence x remains unchanged and there is a sign change, see Fig. 11.4. Again we cannot measure the sign change and the fact that x is not flipped indicates correctly that the function is constant.

11.B Derivation of some useful identities in quantum circuits

We have

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{11.9}$$

By direct calculation it is easy to see that $X^2 = \mathbb{1}, Z^2 = \mathbb{1}$, and

$$H^2 = \mathbb{1}\,, \tag{11.10}$$

where $\mathbb{1}$ is the identity

$$\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{11.11}$$

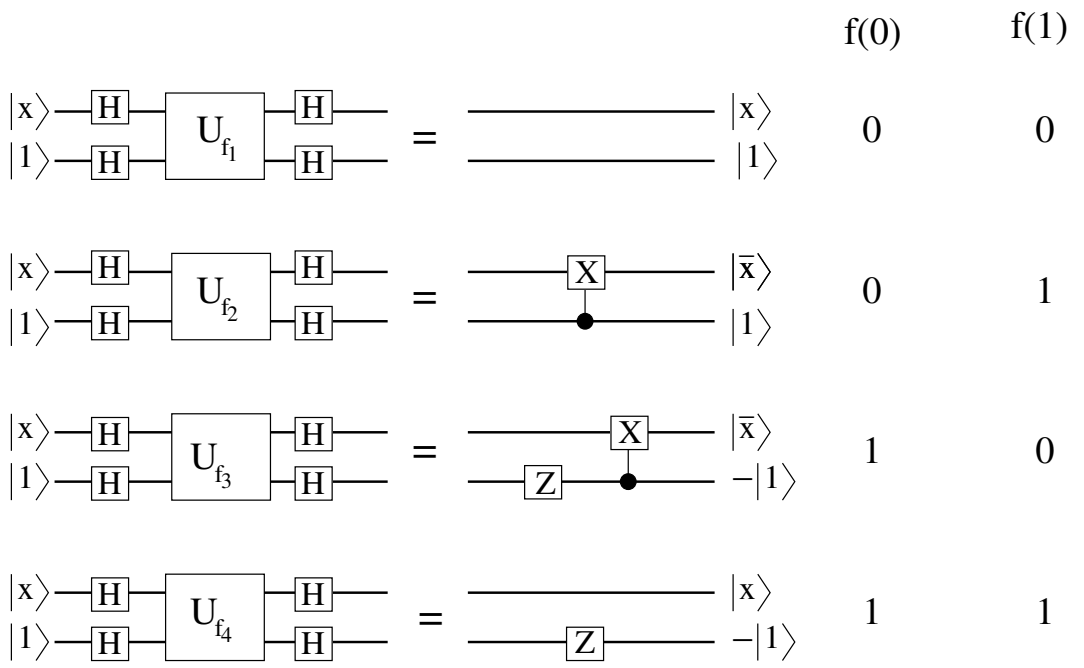


Figure 11.4: The circuits for the four functions $f_1, \cdots, f_4$ given in Fig. 11.3 when sandwiched between Hadamards in order to perform the Deutsch algorithm. The upper qubit is initialized in either of computational basis states, $|x\rangle$ with $x = 0$ or 1, while the lower qubit is initialized to be $|1\rangle$. The derivations of the equivalent circuits shown are given in the text. One sees that the upper qubit is flipped for those functions which are not constant, and is not flipped for the constant functions.

Equation (11.10) is represented graphically by Fig. 11.5(a) Also by direct calculation, we have $XH = HZ$. Hence multiplying on the left by H gives

$$HXH = Z\,, \tag{11.12}$$

see Fig. 11.5(b) for a graphical illustration, and multiplying on the right by H gives

$$HZH = X\,, \tag{11.13}$$

which is illustrated graphically in Fig. 11.5(c)

The NOT part of the CNOT gate is performed by the X operator. Hence it is sometimes convenient to denote a CNOT gate as a control-X gate, see Fig. 11.5(d). We will also meet the control-Z gate, in which the target qubit is acted upon by Z if the control qubit is 1, and otherwise the target qubit is unchanged. As with the control-X gate, there is no change in the control qubit. With a bit of thought, we see that the only effect of the control-Z gate is to change the overall sign of the state if both the target and control qubits are one. Thus the distinction between target and control is non-existent, and control and target qubits can be interchanged, see Fig. 11.5(e).

Now consider a CNOT (control-X) gate sandwiched between Hadamards as shown in Fig. 11.5(f). Consider the target (lower) qubit. If the control qubit does not act on it, the target qubit is just acted on by the two Hadamards which is equivalent to the identity, see Fig. 11.5(a). If the control qubit does act on the target qubit, the target qubit is acted on by the succession of gates HXH which is equivalent to Z, see Fig. 11.5(b). Both these possibilities are taken care of by the equivalent circuit in Fig. 11.5(f)(i), which is control-Z gate. As illustrated in Fig. 11.5(e), the target and control qubits in a control-Z gate can be interchanged so Fig. 11.5(f)(i) is equivalent to Fig. 11.5(f)(ii). Now the

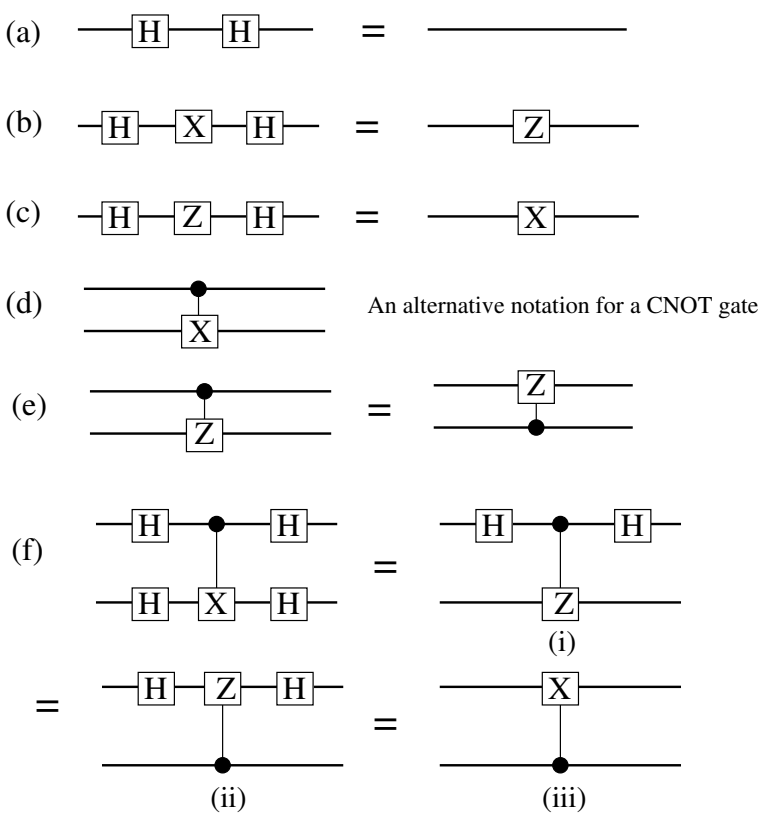


Figure 11.5: Some useful identities in quantum circuits. Of particular note is identity (f) which shows that putting Hadamards around a CNOT gate is equivalent to a CNOT gate without Hadamards, but with the control and target qubits interchanged.

target qubit is the upper one, and has the sequence of gates H Ctrl-ZH acting on it. Similar to the argument that showed Fig. 11.5(f) is equivalent to Fig. 11.5(f)(i), this is equivalent to Ctrl-X because of the identities in Fig. 11.5(a) and Fig. 11.5(c). Hence Fig. 11.5(f) is equivalent to Fig. 11.5(f)(iii).

So we see that a CNOT surrounded by Hadamards is equivalent to a CNOT gate without Hadamards but with the control and target qubits interchanged, a quite surprising result.

One could also derive this result by multiplying 4×4 matrices which is more tedious. However, for completeness we will do it here. The CNOT gate has the matrix representation

$$\begin{matrix} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix} \tag{11.14}$$

$$U_{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{11.15}$$

In this tensor product the control qubit is to the left. The target qubit (to the right) is flipped if the control qubit (to the left) is 1 (so, relative to the identity matrix, columns 3 and 4 are interchanged). In a CNOT gate with target and control qubits swapped, the left hand qubit is flipped if the right

hand qubit is 1 (so columns 2 and 4 are interchanged). Hence we have

$$\begin{matrix} & |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix} \tag{11.16}$$

$$U_{CNOT\_SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{11.17}$$

The tensor product $H^{\otimes 2}$ is given by

$$H^{\otimes 2} = \frac{1}{\sqrt{2}} \begin{pmatrix} H & H \\ H & -H \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \tag{11.18}$$

One can check by working out the matrix multiplication that

$$U_{CNOT\_SWAP} = H^{\otimes 2}\, U_{CNOT}\, H^{\otimes 2}\,, \tag{11.19}$$

in agreement with Fig. 11.5(f). This is a bit tedious so I used *Mathematica*. It is more straightforward to use the circuit identities shown in Fig. 11.5.

Chapter 12

The Bernstein-Vazirani Algorithm

12.1 The Algorithm

Like the Deutsch algorithm, the Bernstein-Vazirani algorithm finds information about a black box function, but has a bigger speedup. It is very similar to the Deutsch-Josza algorithm which is set as a homework problem in the appendix at the end of this chapter.

Consider a function

$$f(x) = a \cdot x \tag{12.1}$$

where a and x have n bits while the function itself, f, has one bit. The dot indicates a bitwise inner product with modulo 2 addition:

$$a \cdot x \equiv a_0 x_0 \oplus a_1 x_1 \oplus \cdots \oplus a_{n-1} x_{n-1} \,. \tag{12.2}$$

The problem is to determine a.

Let's make sure that we understand the "dot". We have $a_i x_i = 0$ or 1. Hence

$$a \cdot x = a_0 x_0 \oplus a_1 x_1 \oplus \cdots \oplus a_{n-1} x_{n-1} \tag{12.3}$$

$$= \begin{cases} 1 & \text{if an odd number of terms is 1} \\ 0 & \text{if an even number of terms is 1} \end{cases} \tag{12.4}$$

For example for $n = 4$, if the bits of a are 1101 and the bits of x are 1110 (recall that the zeroth bit is the least significant, i.e. the rightmost one) then[1]

$$a \cdot x = (1 \times 0) + (0 \times 1) + (1 \times 1) + (1 \times 1) \mod 2 = 0 + 0 + 1 + 1 \mod 2 = 2 \mod 2 = 0. \tag{12.5}$$

Hence, for these values of a and x, $f(x) = 0$. If we take $x = 1000$ then $f(x) = 0 + 0 + 0 + 1 \bmod 2 = 1$.

Classically we can only determine the bits of a one at a time. The k-th bit of a can be determined by feeding in $x = 2^k$. To see this, consider the binary representations of a and x:

$$\begin{aligned} a &= a_0 + a_1 2^1 + \cdots + a_k 2^k + \cdots a_{n-1} 2^{n-1} \,, \\ x &= x_0 + x_1 2^1 + \cdots + x_k 2^k + \cdots x_{n-1} 2^{n-1} \,. \end{aligned} \tag{12.6}$$

Hence if $x = 2^k$ then $x_k = 1$ while, for $l \neq k, x_l = 0$, so $a \cdot x = a_k$. Consequently $f(2^k) = a_k$. We have to do this for each bit, $k = 0, 1, 2, \cdots, n-1$, so it requires n calls of the function.

[1] One can either do the mod 2 operation after each addition or add up in the normal way and apply the mod 2 operation at the end. In either case, the result is 0 if an even number of terms in the sum are 1, and 1 if an odd number of terms are 1.

The quantum algorithm succeeds in determining a with just *one* call!

A schematic diagram of a general reversible unitary transformation which takes an n-bit input x in the upper register and generates an m-bit output $f(x)$ in the lower register is shown in Fig. 10.2. For the Bernstein-Vazirani Algorithm there are n qubits in the upper register but only 1 qubit in the lower register. In addition, the unitary U_f is surrounded by Hadamards, as shown in Fig. 12.1. The upper register is set to $|0\rangle_n$ and the lower qubit to $|1\rangle$. This is the same circuit as for the Deutsch-Josza algorithm, see Appendix 12.A.

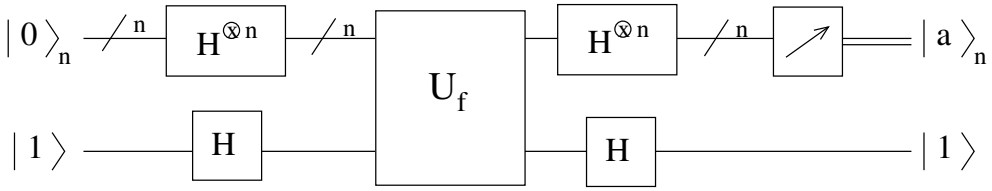


Figure 12.1: Circuit diagram for the Bernstein-Vazirani algorithm. In the final state the upper (input) register contains $|a\rangle$ while the lower (output) qubit reverts to its initial state $|1\rangle$. The desired value of a can therefore be read off by measuring the upper register.

Acting with H on $|0\rangle$ gives an equal linear superposition of the two basis states. Similarly acting with $H^{\otimes n}$ on $|0\rangle_n$ gives an equal superposition of the 2^n basis states. Hence, including the lower register, the state inputted to U_f is

$$H^{\otimes n}|0\rangle_n \otimes H|1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} |x\rangle_n \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \,. \tag{12.7}$$

For each term in the superposition, the function U_f acts in the same way as for the Deutsch algorithm described in Chapter 11. The lower qubit is flipped if $f(x) = 1$, which is the same as changing the sign of the state. If $f(x) = 0$ there is no change. Hence each term in the superposition acquires a factor of $(-1)^{f(x)}$, so the state of the system immediately after the action of U_f is

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} (-1)^{f(x)} |x\rangle_n \otimes \frac{(\,|0\rangle - |1\rangle\,)}{\sqrt{2}} \,. \tag{12.8}$$

Next consider the effect of the Hadamards acting after U_f. The action on the lower qubit is to convert $(\,|0\rangle - |1\rangle\,)/\sqrt{2}$ to $|1\rangle$. However, the effect of $H^{\otimes n}$ acting on an arbitrary computational basis state $|x\rangle_n$ needs more thought. Consider first just one qubit. Then

$$H|x\rangle = \frac{1}{\sqrt{2}} (\,|0\rangle + (-1)^x |1\rangle\,) = \frac{1}{\sqrt{2}} \sum_{y=0}^{1} (-1)^{xy} |y\rangle \,. \tag{12.9}$$

Hence the effect of applying $H^{\otimes n}$ on an n-qubit computational basis state is

$$\begin{aligned} H^{\otimes n}|x\rangle_n &= \frac{1}{2^{n/2}} \sum_{y_{n-1}=0}^{1} \cdots \sum_{y_1=0}^{1} \sum_{y_0=0}^{1} (-1)^{\sum_{j=0}^{n-1} x_j y_j} |y_{n-1}\rangle \cdots |y_1\rangle |y_0\rangle \,, \\ &= \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} (-1)^{x\cdot y} |y\rangle_n \,, \end{aligned} \tag{12.10}$$

where $x \cdot y$ is the bitwise inner product with modulo 2 addition defined in Eq. (12.2), and we have used the fact that we only need to know whether $\sum_{j=0}^{n-1} x_j y_j$ is even or odd. Hence, combining Eqs. (12.8) and (12.10), the amplitude to find the upper register in state $|y\rangle_n \equiv |y_{n-1}\rangle \cdots |y_1\rangle |y_0\rangle$ is

$$\begin{aligned} c_y =& \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)+x\cdot y} \\ =& \frac{1}{2^n} \prod_{j=0}^{n-1} \left[\sum_{x_j=0}^{1} (-1)^{(a_j+y_j)x_j} \right] . \end{aligned} \tag{12.11}$$

Let us evaluate this for the state where $y_j = a_j$ for all j, in which case $a_j + y_j = 2$ or 0. If $x_j = 0$ then $(-1)^{(a_j+y_j)x_j} = 1$ and if $x_j = 1$ we also get $(-1)^{(a_j+y_j)x_j} = 1$, so $\sum_{x_j=0}^{1} (-1)^{(a_j+y_j)x_j} = 2$, i.e. the two terms add up in phase. Hence, from Eq. (12.11), we have $c_a = 1$. Since the total probability must add up to 1 this means that all the other amplitudes must be zero. To see that this is indeed the case, note that for each qubit where $y_j \neq a_j$, $a_j + y_j = 1$ and so the sum over x_j for these qubits gives zero. The final result in Eq. (12.11) is a product over terms for each qubit and so we get zero, as required.

Including the lower (output) qubit, the final state is

$$|a\rangle_n \otimes |1\rangle \,, \tag{12.12}$$

and a measurement of the upper register in Fig. 12.1 gives a, *with probability one*, even though we made just *one* call to the function.

Since a classical computation of a requires n function calls, we have obtained a "*quantum speedup*" of n. Note that the procedure is analogous to Deutsch's algorithm. The first set of Hadamards generates a superposition of inputs to the gate U_f which "evaluates"[2] the function for all 2^n inputs using *quantum parallelism*, and then the second set of Hadamards destroys all the outputs apart from a, using quantum *interference*.

12.2 An Alternative Derivation

Following Mermin [Mer07] and Vathsan [Vat16] it is useful to give an alternative derivation of how the circuit in Fig. 12.1 works, by giving an *explicit* construction of the black box U_f. It is convenient to illustrate by a specific example. We take $n = 5$ and $a = 11010$ so $x_0 = 0, x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$ (recall we read the bits from right, the least significant, to left, the most significant). The function $a \cdot x$ can be implemented by the gates shown in Fig. 12.2.

To incorporate U_f into the Bernstein-Vazirani algorithm, we sandwich it in between Hadamards, see Fig. 12.1, and note that the Hadamards interchange control and target qubits in the CNOT (control-X) gates, see Fig. 11.5(f) in Chapter 11. As before, the initial upper register is $|0\rangle_n$ and the lower register is $|1\rangle$. We see immediately from Fig. 12.3 that a is *directly* imprinted in the final state of the input register. There does not appear to be any parallelism and interference.

Hence these two explanations of the Bernstein-Vazirani algorithm are quite different. To quote Mermin [Mer07]:

> "The first applies U_f to the quantum superposition of all possible inputs and then applies operations which leads to perfect destructive interference of all states in the superposition except for the one in which the upper (input) register is in the state $|a\rangle$. The second suggests a specific mechanism for representing the subroutine that executes U_f and then shows that sandwiching such a mechanism between Hadamards *automatically* (my italics)

[2] To understand the reason for the quotation marks see the discussion at the end of Sec. 10.2.

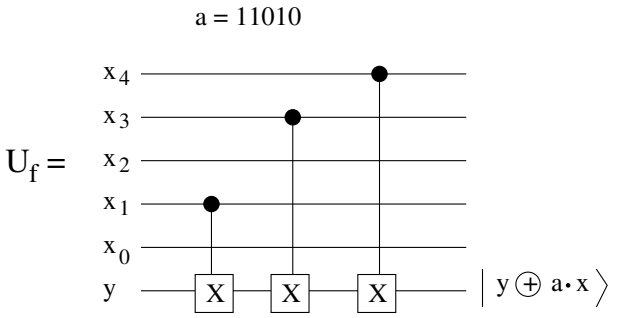


Figure 12.2: A circuit diagram for $n = 5$ to implement the function $f(x) = a \cdot x$ with $a = 11010$, i.e. $f(x) = x_1 + x_3 + x_4 \mod 2$. The circuit flips the output qubit, the lowest one, initialized to y, whenever x_1 or x_3 or x_4 is 1. (Note that flipping y is equivalent to adding 1 to y mod 2.) Hence the final value of the output qubit is $y \oplus (a \cdot x) = y \oplus x_1 \oplus x_3 \oplus x_4$ as required.

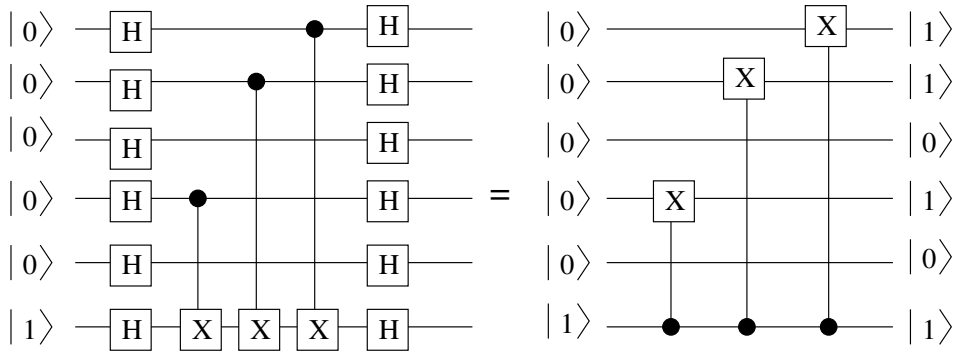


Figure 12.3: Sandwiching the circuit for U_f in Fig. 12.2 between Hadamards, and realizing that the effect of the Hadamards is to interchange the control and target qubits in the CNOT (control-X) gates, we see immediately that the final state of the upper (input) register contains $a = 11010$.

> imprints a on the upper register. Interestingly, quantum mechanics appears in the second method only because it allows the reversal of the control and target qubits of a cNOT operation solely by means of 1-qubit (Hadamard) gates."

(I have used the conventional spelling of "qubit" rather than Mermin's idiosyncratic "Qbit".)

Appendices

12.A Homework Problem on the Deutsch-Josza Algorithm

This is an extension of the Deutsch algorithm discussed in Chapter 11. Recall that in Deutsch's algorithm the input is one bit and the output is also one bit. In the Deutsch-Josza algorithm, the output is still one bit but the input has n bits, so there are 2^n distinct inputs. We are told that *either* the function is "constant" (in which case the function outputs the same value for all 2^n inputs) *or* is "balanced" (in which case an equal number of inputs give the results 1 and 0). Clearly this is a very artificially constructed problem.

The circuit for the Deutsch-Josza algorithm, shown in Fig. 12.4, is the same as for the Bernstein-Vazirani algorithm.

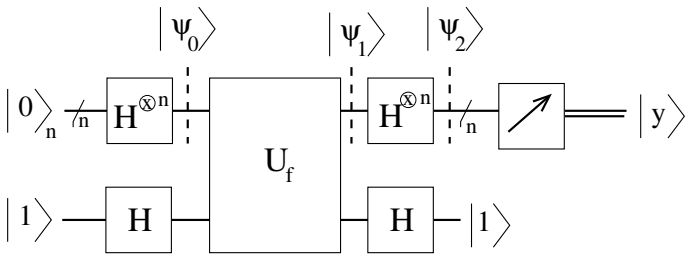


Figure 12.4: The circuit for the Deutsch-Josza algorithm. It is the same as that for the Bernstein-Vazirani algorithm shown in Fig. 12.1. Here we label the states at different points in the circuit.

The function U_f acts as follows on computational basis states $|x\rangle_n$ and $|z\rangle$:

$$U_f|x\rangle_n|z\rangle = |x\rangle_n|z \oplus f(x)\rangle\,, \tag{12.13}$$

where x is an n-bit integer, $|x\rangle$ is the state of the n-qubit upper register in the figure, z and $f(x)$ are 1-bit integers, and $|z\rangle$ is the lower qubit in the figure. The lower qubit is initialized to $|1\rangle$ and the upper register is initialized to $|0\rangle_n$.

(i) Show that

$$|\psi_0\rangle_n = H^{\otimes n}|0\rangle_n = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle_n\,, \tag{12.14}$$

so the input to the function U_f is the uniform superposition of all 2^n basis states.

(ii) Show that after the action of U_f the state of the upper register is

$$|\psi_1\rangle_n = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle_n. \tag{12.15}$$

(iii) Show that after the action of the second set of Hadamards on the n-qubit register, the state of that register is

$$|\psi_2\rangle_n = H^{\otimes n}|\psi_1\rangle_n = \frac{1}{2^n}\sum_{x,y=0}^{2^n-1}(-1)^{[f(x)+x\cdot y]}|y\rangle_n, \tag{12.16}$$

where $x \cdot y$ is the bitwise inner product of x and y with modulo 2 addition:

$$x \cdot y = x_0y_0 \oplus x_1y_1 \oplus \ldots \oplus x_{n-1}y_{n-1}\,. \tag{12.17}$$

(iv) The upper register is then measured, and an n-bit integer y is obtained. Show that if the function is a constant then $y = 0$ with probability 1. Show also that if the function is balanced then one must get a non-zero value of y. Hence the Deutsch-Josza algorithm succeeds with *just one* function call.

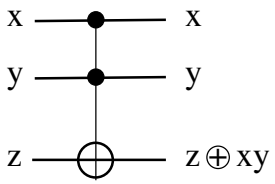
(v) How does this compare with a classical approach? The only thing one can do classically is keep computing $f(x)$ for different values of x and seeing if one gets more than one value for the output. If the function is balanced, one would probably get different outputs quite quickly. If the function is constant one would need to evaluate half the inputs (plus 1), i.e. $2^{n-1}+1$, to be 100% sure that the function is not balanced. This is exponentially (in n) worse than the quantum algorithm.

However, this is arguably not fair. We may well be content to establish that the function is constant with some high probability[3], a bit less than one. If the function is constant, how many function calls would you need classically to rule out the possibility that it is balanced with a probability of error of no more than (i) 10^{-3} and (ii) 10^{-6}.
Note: For simplicity, assume that the number of function calls is much less than $2^{n/2}$, the number of values of x which give the same result if the function is balanced. This means that for a balanced function the probability of getting the result 0 or 1 is 1/2 each time, independent of results obtained previously from other values of x.

12.B Homework problem on making a Toffoli Gate out of 1-qubit and 2-qubit gates

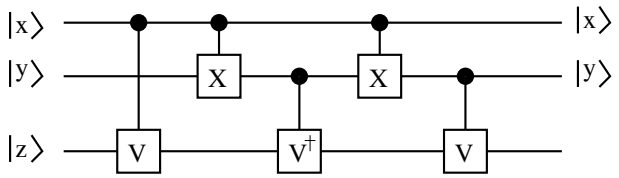
We noted in Sec. 8.1 that for *classical* reversible computation we need three-bit gates, such as the Toffoli gate, in addition to 1-bit and 2-bit gates, to be able to perform universal computation. However, three qubit gates are, fortunately, not needed in quantum computation because the necessary three-bit gates can be constructed out of 1-qubit and 2-qubit gates.

Here we consider the quantum Toffoli gate, which is a control-control-NOT (C-C-NOT) gate. The target qubit z is flipped if both the control qubits, x and y, are 1 and is otherwise unchanged.



We present here the method of constructing the Toffoli gate out of 1-qubit and 2-qubit gates as a homework problem (including help).

(i) Consider the following circuit for an arbitrary unitary operator V:



Show that it acts with V^2 on $|z\rangle$ if both x and y are 1 and otherwise does nothing.
Hint: One possible way of approaching this question (though not the most elegant way) is to consider separately what happens for the four possible input values of the control qubits x and y, 00, 01, 10, and 11. Alternatively, and more elegantly, one can note that if $|z\rangle$ is the target and $|x\rangle$ is the control, then the control-V gate acts like V^x on $|z\rangle$. (Recall, too, that $V^\dagger = V^{-1}$ since we are told that V is unitary.)

(ii) Now take V to be the following 1-qubit gate:

$$V = (1-i)\frac{(\mathbb{1} + i\,X)}{2}\,. \tag{12.18}$$

[3]For later quantum algorithms we will only be able to solve the problem with high probability. Since we have to give up 100% certainty in the quantum case, we should not insist on 100% certainty here from the classical algorithm.

Show that $V^\dagger V = \mathbb{1}$, and hence V is unitary. Show also that $V^2 = X$ and hence the above circuit is a quantum Toffoli gate.
Note:

- One sometimes says that V is the "square root of X".
- Unlike other unitary operators considered in this course, V is not its own inverse.
- Vathsan [Vat16] discusses how to construct a general Controlled-V gate in her Sec. 7.4.1.

Chapter 13

Simon's Algorithm

So far we have studied Deutsch's algorithm in Chapter 11 which gave a quantum speedup of a factor of 2, and the Bernstein-Vazirani algorithm in Chapter 12, which gave a speedup of n, where n is the size of the problem. Next we consider a problem, due to Daniel Simon, which gives an *exponential* speedup in n. Like the previous algorithms it has an artificial character and is not of practical use, but it has features in common with the vastly more useful algorithm of Shor for factoring integers, which we shall spend a substantial amount of time on in the next few chapters. Like Shor's algorithm, Simon's is of a probabilistic nature.

In Simon's problem we are given a black box function which takes an n-bit input and has the property that

$$f(x \oplus a) = f(x), \tag{13.1}$$

where a is a non-zero n-bit integer and $\oplus$ means bitwise addition modulo 2. Note that each bit is treated separately, so if the integer x is represented in binary notation by bits $x_{n-1}x_{n-2}\cdots x_1x_0$, and similarly for a then $x \oplus a$ is an integer y with binary representation $y_{n-1}y_{n-2}\cdots y_1y_0$ where $y_j = x_j \oplus a_j$.

Adding a twice to x (modulo 2) gives back x, i.e.

$$x \oplus a \oplus a = x \tag{13.2}$$

since adding a bit to itself gives $0 \,(\mod 2)$ irrespective of whether that bit is 0 or 1. Hence

$$f(x) = f(x \oplus a) = f(x \oplus a \oplus a) \tag{13.3}$$

and so on, so $f(x)$ is *periodic*, with period a, under bitwise mod 2 addition. We are told that for every x there is only one other input to the function, $x \oplus a$, which gives the same output, so there are 2^{n-1} distinct values of f. Hence we assume that we can represent f by $n-1$ qubits.

The problem is to determine the period a with the least number of function calls.

If we input different values of x and find a repeated output, i.e. if $f(x_i) = f(x_j)$, then $x_j = x_i \oplus a$. If we add x_i to both sides (bitwise addition modulo 2) we get

$$a = x_i \oplus x_j \,. \tag{13.4}$$

so we obtain a if we can find two values of x which give the same function value.

An example of a function with the desired property is shown in Table 13.1.

Classically this problem is hard, by which we mean that the number of function calls grows *exponentially* with n. All one can do is call the function with different values of x until one finds a repeated output, i.e. $f(x_i) = f(x_j)$, which gives us a from Eq. (13.4). After m calls to the function we have compared $m(m-1)/2$ pairs. For a reasonable chance of success we need $\frac{1}{2}m(m-1) \sim 2^n$, so $m = O(2^{n/2})$, i.e. exponential in the number of bits n.

x	0	1	2	3	4	5	6	7
$f(x)$	3	2	2	3	0	1	1	0

Table 13.1: An example with $n = 3$ bits of the type of function that is considered in Simon's algorithm. The function satisfies $f(x) = f(x \oplus a)$ for some non-zero a. To determine a we look for repetitions. An example is $f(4) = f(7) = 0$. Hence, according to Eq. (13.4), $a = 4 \oplus 7 = 100 \oplus 111 = 011 = 3$. The other repetitions satisfy this same condition as you can check.

The circuit to solve this problem **quantum mechanically** is similar to that in the Bernstein-Vazirani algorithm except that the lower register has enough qubits to contain the function values, i.e. $n-1$. Also the phase kickback is not used, so the lower register is initialized to $|0\rangle_{n-1}$ rather than $|1\rangle$ and we do not have Hadamards on the lower register. A final difference is that we measure first on the lower register rather than the upper one. The circuit diagram is shown in Fig. 13.1.

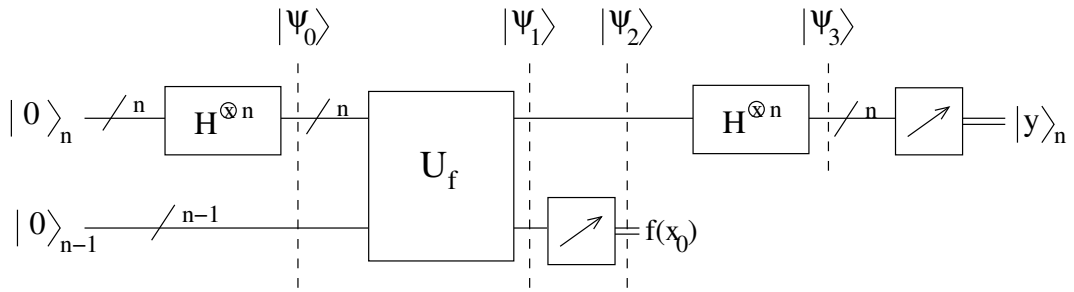


Figure 13.1: Circuit diagram for Simon's algorithm. The upper register has n qubits and contains the x values, while the lower register has $n-1$ qubits and contains the values of the function $f(x)$.

After the first Hadamards in the upper register the state of the system is

$$|\psi_0\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n \otimes |0\rangle_{n-1} \,. \tag{13.5}$$

The function call makes the transformation $|x\rangle_n \otimes |y\rangle_{n-1} \to |x\rangle_n \otimes |y \oplus f(x)\rangle_{n-1}$, see Fig. 12.1 in Chapter 12. Here $y = 0$ so, after the function call the state becomes

$$|\psi_1\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n \otimes |\, f(x)\,\rangle_{n-1} \,. \tag{13.6}$$

A measurement is then done on the lower register which will record some value of the function, f_0 say, for which there are two values of x which we denote by x_0 and $x_0 \oplus a$. Hence, immediately after the measurement, the state of the system is

$$|\psi_2\rangle = \frac{|x_0\rangle_n + |x_0 \oplus a\rangle_n}{\sqrt{2}} \otimes |\, f_0\,\rangle_{n-1} \,. \tag{13.7}$$

If we were now to measure the upper register, we would get *either* x_0 or $x_0 \oplus a$. At first glance, this might seem like progress since we appear to be halfway there. If we could just get the other number, we would have a. However there is no way to get both. If we could clone the state several times and measure each clone then, with high probability, we would be able to determine both of them. However,

the no-cloning theorem says that we can't clone an arbitrary, unknown state. Also, repeating the whole procedure doesn't help because, with high probability, we would get a different function value, $\tilde{f}_0$, and hence a different pair $\tilde{x}_0$ and $\tilde{x}_0 \oplus a$, from which again we would not be able to extract a.

As in Deutsch's algorithm and the Bernstein-Vazirani algorithm, we must do some processing *before* the final measurement. As we showed in Eq. (12.10) in Chapter 12 on the Bernstein-Vazirani algorithm, the effect of Hadamards on n-qubit register which is in a computational basis state $|x\rangle_n$, is given by

$$H^{\otimes n}|x\rangle_n = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} (-1)^{x\cdot y}|y\rangle_n \,, \tag{13.8}$$

where $x \cdot y$ is the bitwise inner product modulo 2,

$$x \cdot y \equiv x_0 y_0 \oplus x_1 y_1 \oplus \cdots \oplus x_{n-1} y_{n-1} \mod 2, \tag{13.9}$$

discussed in Sec. 12.1. Hence, applying Hadamards to the n-qubit upper register in state $|\psi_2\rangle$ in Eq. (13.7), the state of that register becomes

$$|\psi_3\rangle_n = \frac{1}{\sqrt{2}} \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} \Big[(-1)^{x_0\cdot y} + (-1)^{(x_0\oplus a)\cdot y} \Big] |y\rangle_n \,. \tag{13.10}$$

Now[1] $(x_0 \oplus a)\cdot y = (x_0 \cdot y) \oplus (a \cdot y)$ and we note that $a \cdot y = 0$ or 1. If $a \cdot y = 1$ then $(-1)^{(x_0\oplus a)\cdot y} = (-1)^{x_0\cdot y}(-1)^{a\cdot y} = -(-1)^{x_0\cdot y}$, so the two terms in Eq. (13.10) cancel. Hence the only terms with a non-zero amplitude are those with $a \cdot y = 0$.

A measurement on the upper register then gives, with equal probability, *one* value of y with $a\cdot y = 0$. This is a linear equation for the a_i, the bits of a. If we can find n linearly-independent equations for the a_i, we can obtain the solution. Hence we have to repeat the procedure, each time getting a value for f_0 and y. As discussed in Appendix G of Mermin [Mer07] one needs to run the algorithm *a little more* than n times because the equations one gets each time for the a_i are not necessarily linearly independent. The result is that if one runs $n + x$ times, then the probability of getting n linearly dependent equations (and hence the solution for the a_i) is is greater than

$$1 - \frac{1}{2^{x+1}} \,. \tag{13.11}$$

Hence there is less than one chance in a million of failure if one calls the function $n + 20$ times. A crucial point in this expression is that the number of calls beyond n needed to find a solution with some high probability *does not depend on* n.

The occurrence of probability, and some arcane mathematical arguments to prove that one does get the solution with high probability within the specified number of runs, is characteristic of several quantum algorithms including Shor's.

In the case of Simon's problem, the classical algorithm takes of order $2^{n/2}$ function calls whereas the quantum algorithm finds the answer with high probability with little more than n calls[2]. This is an *exponential* speedup[3].

[1] This is the mod 2 version of the usual distributive rule for addition and multiplication: $a \times (b + c) = (a \times b) + (a \times c)$.

[2] In the interests of full disclosure I should state that one also needs to solve n linear equations on a classical computer, which takes of order n^3 steps. A algorithm which takes a time proportional to a power of the problem size n is said to be *polynomial*. Since classical hardware is cheap it is not clear if one should include this time using a classical computer in the computational cost of Simon's algorithm. However, since n^3 is polynomial, even if one does include this time the comparison is still between a polynomial quantum (+classical) algorithm and an exponential purely classical algorithm, which is still an exponential speedup, see footnote 3.

[3] An algorithm which takes a time proportional to a power of the problem size is said to have *polynomial complexity*, while if the time increases exponentially with size (or exponentially with a power of the size) it is said to have *exponential complexity*. If one algorithm has polynomial complexity and another has exponential complexity then the former is said to have an exponential speedup compared with the latter.

Finally a few words of anticipation for Shor's algorithm which we will do next. Simon's problem considers a function which is periodic under bitwise modulo 2 addition, see Eq. (13.3). Shor's algorithm investigates functions which are periodic under *ordinary* addition: $f(x+a) = f(x)$, which is much more useful. In Simon's problem, the action of the n-Hadamards in Eq. (13.8) can be written

$$H^{\otimes n}|x\rangle_n = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} e^{i\pi x\cdot y}|y\rangle_n \,, \tag{13.12}$$

Since $x \cdot y$ is the bitwise inner product modulo 2, it only takes values 0 and 1, so the phases in the complex exponential are just 0 and π. The core of Shor's algorithm is a quantum Fourier transform (QFT), where an essential difference from Eq. (13.12) is that the bitwise inner product is replaced by ordinary multiplication. Hence the QFT generates many different phases, with the result that, unlike Simon's algorithm, it cannot, in general, be constructed entirely out of 1-qubit gates. Fortunately, it *can* be constructed entirely out of 1- and 2-qubit gates. All this and more will be discussed in Chapter 18.

Chapter 14

Factoring and RSA (Rivest-Shamir-Adleman) Encryption

Shor's famous quantum algorithm, to be discussed in detail in Chapter 18, factors large integers much more efficiently than any known classical algorithm. Factoring is not just of interest to mathematicians, however, because the difficulty of factoring is at the heart of the popular RSA method of encrypting sensitive information sent via the internet (or some other public channel). While RSA is not the only method use to encrypt information, my understanding is that some version of Shor's algorithm can be used to crack other encryption methods such as Diffie-Hellman. RSA stands for the names of its inventors, Rivest, Shamir and Adleman.

This chapter is a LaTeX copy of a *Mathematica* notebook, the original of which is available at `https://young.physics.ucsc.edu/150/rsa.nb`. In it, the RSA algorithm is implemented, parameters are chosen, and random messages are generated. These are encrypted, the encrypted messages are decrypted, and a check is made that the original message is recovered. It you have *Mathematica* you can run the notebook version and verify that the RSA algorithm works.

Suppose that Bob wants to receive a message from Alice on the internet (a public channel). Anything sent on a public channel can be intercepted by others. How can Bob and Alice agree on a coding scheme and then send each other coded messages which can be decoded by the other person but not by anyone "sniffing" on the internet? This has to be accomplished by only sending messages down the public channel.

We will now describe the RSA encryption scheme for doing this. It uses a result of number theory which we will quote but not prove. To receive the message from Alice, Bob picks two large prime numbers p and q, and sends to Alice, on the public channel, their product

$$N = pq, \tag{14.1}$$

but not p and q separately. N is taken to be large enough, typically a few thousand bits, that it cannot be factored on a classical computer. You might ask how can one choose the large prime numbers p and q. If one selects a large integer N at random it can be shown that the probability that it is prime is about $1/\ln N$. Hence, even if N has, say, 400 digits (around 1000 bits) you only have to take test a few hundred to a thousand random integers to typically find a prime number. But can one efficiently test if a number is prime? It turns out that one can, even though, if the number is found to be not prime, there is no known efficient classical algorithm to determine the prime factors. The website `http://mathworld.wolfram.com/PrimalityTest.html` explains how the test for primality is done in *Mathematica*.

Bob also sends a large "encoding number" c which has no factors in common with $(p-1)(q-1)$. If there are no factors in common then the greatest common divisor (GCD) is 1. The GCD of two

integers is easily determined by Euclid's algorithm discussed in Sec. 14.A. According to Appendix J of Mermin [Mer07], the probability that two large random integers have no common factors is greater that 1/2, so it is not difficult to find a suitable value for c.

Hence the public key (available to everyone) is N and c.

Since Bob knows both p and q, and hence $(p-1)(q-1)$, he can also determine the integer d such that

$$c\,d = 1(\text{mod } (p-1)(q-1)). \tag{14.2}$$

Let us remind ourselves of this mod function. The value of a mod b is the result after one subtracts (or adds) the appropriate multiple of b to a to get a value which lies in the range 0 to $b-1$. If a is positive, things are simple, one subtracts a multiple of b (possibly 0) so the mod function is just the remainder after integer division. Hence, for example, 9 mod 5 = 4 because 9/5 = 1 remainder 4. If a is negative one has to add a multiple of b, so, for example, (-13) mod 5 = 2 (since $-13+(3\times 5)=2$).

The above equation, $c\,d = 1$ mod (something), looks strange at first. If c is an integer we would normally think that its inverse should be a fraction. However, here d is also an integer, and the product of two integers *can* give 1 if we use modular arithmetic. For example if $c = 5$ and $d = 3$ then $cd = 15$, and cd mod 7 = 1 (since $15 = (7\times 2)+1$).

The algorithm for computing d in Eq. (14.2) is efficient and an extension of Euclid's algorithm. It is given in Appendix 14.B and in Appendix J of Mermin [Mer07]. It turns out that d is unique. Hence Alice, and anyone else sniffing on the public channel, knows N and c (but not p, and q, and hence not d).

The private key (known only to Bob) is p and q (and hence d).

Alice breaks up her message into chunks of each containing a number of bits less than the number of bits of the integer N. Each chunk is then a binary number less than N. Let's denote by a the numerical value of one chunk.

$$a \text{ is the original message.}$$

Using the values of N and c that Bob has sent, she computes

$$b = a^c \text{ (mod } N) \quad \text{the encoded message.} \tag{14.3}$$

The encoded message b is another large integer, and is sent down the public channel from Alice to Bob.

Bob knows not only c and N, but also the value of d. Here number theory kicks in and shows that the original (unencoded) message a is given by

$$a = b^d \text{ (mod } N) \quad \text{(the original message is recovered).} \tag{14.4}$$

For a proof of this result see the book by Mermin [Mer07]. Bob can compute a because he knows d, but anyone sniffing on the public channel does not know d. However, if a third person, traditionally called Eve, listening on the public channel, could factor N (which is sent down the public channel) into its factors p and q, she would then have $(p-1)(q-1)$ and, since c is also sent down the public channel, she could determine d where $c\,d = 1$ (mod $(p-1)(q-1)$) using the extension of the Euclid algorithm mentioned above. Hence she could find the original unencrypted message a from Eq. (14.4).

Let's do a simple example. We will take

$$p = 7,\ q = 13,\ \text{so } N = 91. \tag{14.5}$$

For the encoding integer we take $c = 11$, which has no factors in common with $(p-1)(q-1) = 6\times 12 = 72$. As shown in Appendix 14.B, using the extended Euclid algorithm one finds that $d = 59$. (Let's verify this: $cd = 11\times 59 = 649 = (9\times 72)+1$ so cd mod $(p-1)(q-1) = 1$, as desired.) The

Mathematica code below sets these values, checks that p and q are prime while N is not, and that $cd = 1 \pmod{(p-1)(q-1)}$. (Note: in *Mathematica* commands I use n rather than N because N has a special meaning in *Mathematica*.) The code then generates a message a by computing a random integer between 0 and $N-1$, and next computes the encoded message b from $b = a^c \pmod{N}$. It then computes $b^d \pmod{N}$ and checks that it gives back the original message a. If you have *Mathematica* you can run the code several times (each time a different random value for the message a will be generated) and see that the original message is always returned.

```
In[1]:=  p=7; q=13; c=11; d=59; n=p*q
Out[1]=  91
```

We check that p and q are prime. The Mathematica command PrimeQ[p] returns "True" if p is prime and "False" if it is not.

```
In[2]:=  PrimeQ[p]
Out[2]=  True
In[3]:=  PrimeQ[q]
Out[3]=  True
In[4]:=  PrimeQ[n]
Out[4]=  False
```

We check that $cd = 1 \bmod ((p-1)(q-1))$.

```
In[5]:=  Mod[c * d, (p-1)(q-1)]
Out[5]=  1
```

We generate a random message.

```
In[6]:= mess = Random[Integer, n - 1]
Out[6]=  51
```

We compute the encoded message.

```
In[7]:= encodedmess = Mod[mess^c, n]
Out[7]=  25
```

We decode the encoded message and check that we recover the original message.

```
In[8]:= recoveredmess = Mod[encodedmess^d, n]
Out[8]=  51
In[9]:= recoveredmess == mess
Out[9]=  True
```

Hence the message was successfully decoded.

Appendices

14.A The Euclidean Algorithm

We want to efficiently find the Greatest Common Divisor (GCD) of two integers. This is the largest factor that they have in common. As a simple example, the GCD of 24 and 9 is 3 since $24 = 2^3 \times 3$ and $9 = 3^2$.

Suppose we want the GCD of two numbers a_0 and b_0 with $a_0 > b_0$. We proceed iteratively. At each stage, the new value of a is equal to the old value of b, and the new value of b is equal to the remainder when the old value of a is divided by the old value of b, i.e.

$$\begin{aligned} a_{n+1} &= b_n \\ b_{n+1} &= a_n - [a_n/b_n]b_n \quad \text{which is the same as } b_{n+1} = a_n \bmod b_n, \end{aligned} \tag{14.6}$$

where $[\cdots]$ means the integer part of the quantity in brackets. Now a_n and b_n decrease at successive iterations and maintain the inequality $a_n > b_n$ (since the largest value that a number can have mod b_n is $b_n - 1$, so $b_{n+1} < b_n\,(= a_{n+1})$). Also a_n and b_n have the same common factors as a_0 and b_0, because a_{n+1} and b_{n+1} are linear combinations of the values at the previous stage, a_n and b_n, and so any common factor is preserved. Eventually we get to a stage where $b_{n+1} = 0$. This means that a_n is divisible by b_n so b_n is the greatest common divisor.

As an example we take $a_0 = 24, b_0 = 9$,

n	a_n	b_n	
0	24	9	(the initial values)
1	9	6	(since $24 = 2 \times 9 + 6$)
2	6	3	(since $9 = 6 \times 1 + 3$)
3	3	0	(since $6 = 3 \times 2 + 0$) .

Hence the GCD of 24 and 9 is b_2 $(= 3)$.

14.B Extension of the Euclidean Algorithm to find an inverse modulo an integer

Given a and c which have no common factors, and $a > c$, we want to find d where

$$c\,d = 1 \bmod a. \tag{14.7}$$

The greatest common divisor of c and a is 1 since, by assumption, they have no common factors. We go through the Euclid algorithm

$$\begin{aligned} a_{n+1} &= c_n \\ c_{n+1} &= a_n - [a_n/c_n]c_n \end{aligned} \tag{14.8}$$

until we get to the stage where $c_n = 1$, the greatest common divisor. One can then obtain d by working backwards through the iterations. This is best shown by an example. We take $p = 7, q = 13$, as in example above, so we have $a = (p-1)(q-1) = 72$ and hence we initialize $a_0 = 72$. We also take $c = 11$ (again as in the example) which has no factors in common with a, and so initialize $c_0 = 11$. Hence the Euclid algorithm proceeds as follows

n	a_n	c_n	
0	72	11	$a_0 = a,\ \ c_0 = c$ (the initial values)
1	11	6	$a_1 = c_0,\ c_1 = a_0 - 6c_0 = 6$
2	6	5	$a_2 = c_1,\ c_2 = a_1 - c_1 = 5$
3	5	1	$a_3 = c_2,\ c_3 = a_2 - c_2 = 1$ ($c_3 = 1$ so we stop).

Hence working backwards,

$$1 = a_2 - c_2 = c_1 - (a_1 - c_1) = 2c_1 - a_1 = 2(a_0 - 6c_0) - c_0 = 2a_0 - 13c_0\ (= 2a - 13c). \tag{14.9}$$

We want to take this (mod a). Now $2a \pmod a = 0$. Since $-13c$ is negative we need to make it positive by adding $a\,c$ (which is zero (mod a)). Hence

$$1 = 2a - 13c \pmod a = -13c \pmod a = (-13 + a)c \pmod a = 59c \pmod a, \tag{14.10}$$

where we used that $a = 72$ to get the last equality. Hence $d = 59$ as stated in the above example.

Chapter 15

Using Period Finding to Factor an Integer

In this chapter, we explain how finding the period of a certain function will enable us to factor integers. We will also illustrate the technique with a simple example. This will probably seem a strange approach for factoring, and is not the preferred method on a classical computer, but it is the method used by Shor in his quantum algorithm.

We take two large primes p and q and form the product

$$N = p\,q\,. \tag{15.1}$$

The goal is to find the factors p and q given only the product N. This is a problem which is hard classically. For applications in cryptography p and q may have around 600 digits (around 2000 bits). We proceed by choosing a random integer a which has no factors in common with N. Whether or not a and N have a common factor can be determined efficiently using Euclid's algorithm, which was described in Sec. 14.A. In the very unlikely event that a and N do have a common factor we have found a factor of N and the problem is solved. Otherwise we compute the following function

$$f(x) \equiv a^x \ (\mathrm{mod}\ N\,) \tag{15.2}$$

for $x = 1, 2, \cdots$. As stated, a and N have no common factors, and for this case one can show that eventually we will get $f(x) = 1$ for some value, $x = r$ say, so

$$a^r \ (\mathrm{mod}\ N) \equiv 1\,. \tag{15.3}$$

The function then repeats since

$$f(x+r) \equiv a^{x+r} \ (\mathrm{mod}\ N\,) \equiv a^x \ (\mathrm{mod}\ N\,) \times \ a^r \ (\mathrm{mod}\ N\,) \equiv a^x \ (\mathrm{mod}\ N\,) = f(x)\,, \tag{15.4}$$

using Eq. (15.3). Hence r is the period of the function.

We we illustrate with a simple example,

$$N = p\,q = 91, \quad \text{with factors } p = 13, q = 7. \tag{15.5}$$

We also take $a = 4$, which has no factors in common with 91. We plot $f(x) \equiv 4^x \ (\mathrm{mod}\ 91\,)$ in Fig. 15.1. The periodic nature is clear, and the period is found to equal 6 by inspection. Let's make sure we

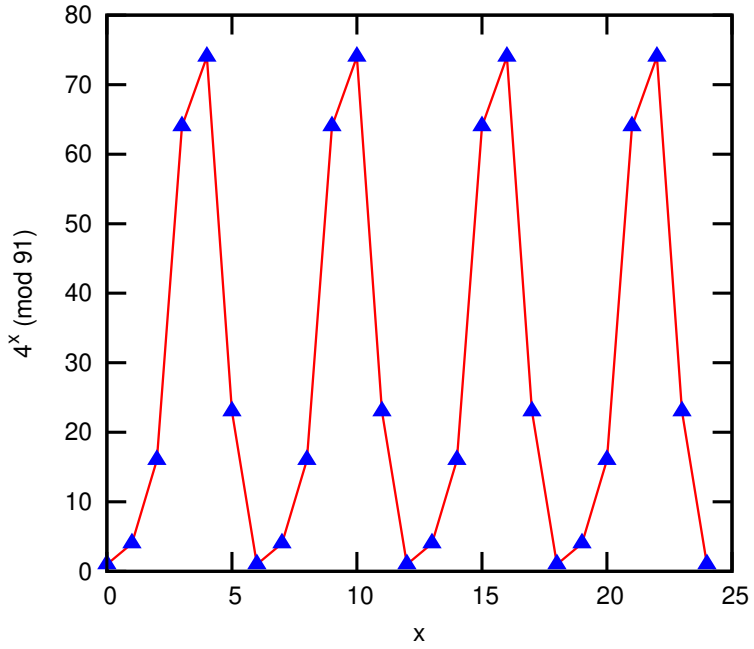


Figure 15.1: The function $f(x) \equiv 4^x \pmod{91}$. The period is seen by inspection to equal 6.

understand how this figure is obtained by working out the values of $4^x \pmod{91}$ for $x = 1, 2, \cdots, 6$.

$$x = 1, \quad 4^x = 4\,, \tag{15.6a}$$
$$x = 2, \quad 4^x = 16\,, \tag{15.6b}$$
$$x = 3, \quad 4^x = 64\,, \tag{15.6c}$$
$$x = 4, \quad 4^x \equiv 64 \times 4 = 256 = 2 \times 91 + 74 \equiv 74 \ (\bmod 91)\,, \tag{15.6d}$$
$$x = 5, \quad 4^x \equiv 74 \times 4 = 296 = 3 \times 91 + 23 \equiv 23 \ (\bmod 91)\,, \tag{15.6e}$$
$$x = 6, \quad 4^x \equiv 23 \times 4 = 92 = 91 + 1 \equiv 1 \ (\bmod 91)\,. \tag{15.6f}$$

The plot in Fig. 15.1 seems to have a fairly regular behavior, but such smooth behavior is exceptional and occurs here only because of the particularly simple choice of parameters. Figure 15.2 shows a plot for the same value of N but with $a = 19$. This is a much more random looking figure, as is typical. In this case the period is $r = 12$. The apparently random shape of $f(x)$ means that one can not estimate the period by taking a few nearby values of x and extrapolating.

We now need to be lucky in two respects:

(i) The period r must be even. This means that $r/2$ is an integer and so is $a^{r/2}$. Hence we can write

$$0 \equiv a^r - 1 \equiv (a^{r/2} - 1)(a^{r/2} + 1) \ (\bmod \ pq)\,. \tag{15.7}$$

(ii) We need that

$$a^{r/2} + 1 \not\equiv 0 \ (\bmod pq)\,. \tag{15.8}$$

It is automatically true that $a^{r/2} - 1 \not\equiv 0 \pmod{pq}$ because, by defintion, $x = r$ is the smallest power for which $a^x - 1 \equiv 0 \pmod{pq}$. Hence, if Eq. (15.8) is true, neither $a^{r/2} + 1$ nor $a^{r/2} - 1$ is divisible by $N = pq$ but, according to Eq. (15.7), their product is, i.e. $\left(a^{r/2} + 1\right)\left(a^{r/2} - 1\right) =$

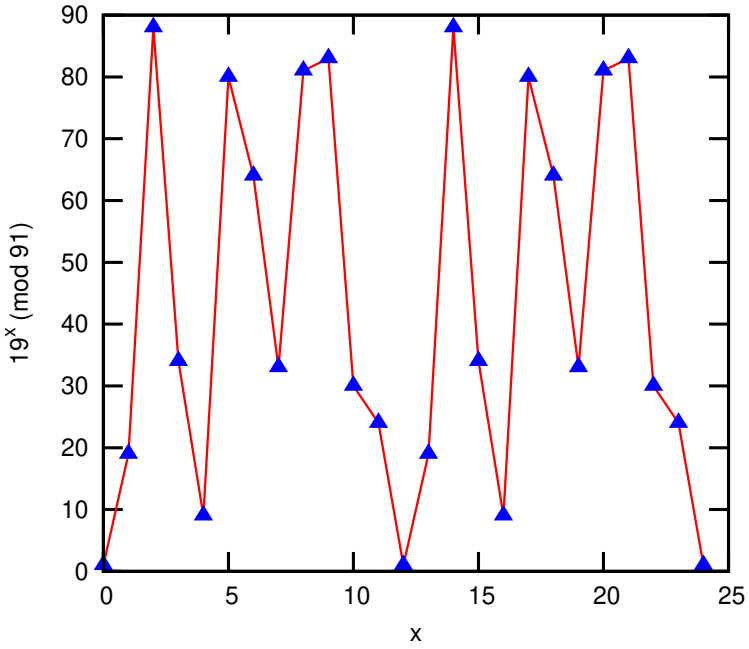


Figure 15.2: The function $f(x) \equiv 19^x \pmod{91}$. The period is seen by inspection to equal 12.

const. pq. Since p and q are primes (and neither $\left(a^{r/2}+1\right)$ nor $\left(a^{r/2}-1\right)$ are multiples of pq), this is only possible if $a^{r/2}+1$ is a multiple of one of the factors, p say, i.e. $a^{r/2}+1 = Cp$, and $a^{r/2}-1$ is a multiple of the other one q, i.e. $a^{r/2}-1 = C'q$ (C and C' are constants). Consequently p is the greatest common divisor of $N\,(=p\,q)$ and $a^{r/2}+1\,(=Cp)$, and q is the greatest common divisor of $N\,(=p\,q)$ and $a^{r/2}-1\,(=C'q)$. We can therefore find p and q using the Euclidean algorithm mentioned earlier.

What are the odds that we will be doubly lucky in this way. According to Appendix M in Mermin [Mer07] the probability is greater than 0.5 for large N. If one *is* unlucky one tries a different choice for a. Since the probability of success is quite high at each attempt, one does not have to repeat the process very many times to succeed with very high probability.

Back to our example. For $N = 91, a = 4$ we found $r = 6$. Indeed we are lucky! This is even. Also $a^{r/2}+1 = 65 \not\equiv 0 \pmod{91}$. So we are doubly lucky! However, this is not remarkable. As noted above the probability of this double luck is greater than 0.5 (at least for large N).

Hence one of the factors is the greatest common divisor (GCD) of 91 and $a^{r/2}+1 = 65$. The other factor is the greatest common divisor of 91 and $a^{r/2}-1 = 63$.

Applying Euclid's algorithm, described in Sec. 14.A, to $f_0 = 91, g_0 = 65$:

$$\begin{aligned} f_1 &= 65, \\ g_1 &= 91 - [91/65]\,65 = 91 - 65 = 26, \\ f_2 &= 26, \\ g_2 &= 65 - [65/26]\,26 = 65 - 52 = 13, \\ f_3 &= 13, \\ g_3 &= 26 - [26/13]\,13 = 26 - 26 = 0. \end{aligned} \tag{15.9}$$

Hence the GCD is $g_2 = 13$, which is indeed one of the factors of 91. By the same process the GCD of 63 and 91 is found to be 7, the other factor of 91.

Chapter 16

The Fourier Transform and the Fast Fourier Transform (FFT)

16.1 Introduction

The standard Fourier Transform concerns a *continuous* function, $x(t)$ say. For descriptive purposes it will be convenient to think of t as time, but this is not essential. In the Fourier transform we decompose $x(t)$ into its components at different "frequencies" $y(\omega)$ as follows:

$$y(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\omega t}\, x(t)\, dt\,. \tag{16.1}$$

If $x(t)$ comprises oscillations at frequencies at ω_0, say, (i.e. has a period T equal to $2\pi/\omega_0$), so $x(t) \sim e^{-i\omega_0 t}$, then $y(\omega)$ will be sharply peaked at $\omega = \omega_0$ (or equivalently at $\omega = 2\pi/T$). Note the inverse relation between the period T and the position of the peak in the Fourier Transform. The larger the period, the smaller the value of ω at the peak.

There is also an inverse Fourier transform,

$$x(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-i\omega t}\, y(\omega)\, d\omega\,, \tag{16.2}$$

which has almost the same form as the original (forward) transform, apart from the sign of i in the exponential. It is shown in standard mathematics texts that substituting for $y(\omega)$ from Eq. (16.1) into the RHS of Eq. (16.2) does give back $x(t)$.

This chapter is concerned with the discrete analog of Eqs. (16.1) and (16.2) in which the data x_m is at a set of N equally spaced "times", and the Fourier transform y_k is at a set of N equally spaced "frequencies". In addition, in the discrete Fourier Transform, the data only covers a finite range, whereas the data in the original, continuous Fourier Transform extends to $\pm\infty$.

16.2 The Discrete Fourier Transform

If we have a set of N data points $x_m\,(m = 0, 1, \cdots, N-1)$, the discrete Fourier transform (FT) is a set of N new values y_k given by

$$y_k = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \exp(2\pi i\, km/N)\, x_m\,, \tag{16.3}$$

evaluated for $k = 0, 1, \cdots, N-1$. We don't need to consider k values outside this range because $y_{k+N} = y_k$ (so the y_k are actually periodic with period N). Equation (16.3) corresponds to a discretized

and finite-range version of Eq. (16.1) with m corresponding to t and $2\pi k/N$ corresponding to ω. If x_m is a periodic function of m with period T, i.e. $x_m \sim e^{-2\pi i m/T}$, then y_k will be peaked for k around N/T since the terms in Eq. (16.3) then add up in phase. This corresponds, in the continuous Fourier Transform, to a peak for ω at around $2\pi/T$.

The inverse Fourier transform has almost the same form; one just needs to take the complex conjugate of the exponential, i.e.

$$x_m = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp(-2\pi i\, km/N)\, y_k\,. \tag{16.4}$$

To see this we substitute Eq. (16.3) into Eq. (16.4) so

$$\begin{aligned} x_m &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp(-2\pi i\, km/N)\, \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} \exp(2\pi i\, kl/N)\, x_l \\ &= \frac{1}{N} \sum_{l=0}^{N-1} x_l \left[\sum_{k=0}^{N-1} \exp(2\pi i\, k(l-m)/N) \right] \\ &= \frac{1}{N} \sum_{l=0}^{N-1} x_l \left[\frac{1-\exp(2\pi i\,(l-m))}{1-\exp(2\pi i\,(l-m)/N)} \right], \end{aligned} \tag{16.5}$$

where, in the last expression, we summed up the geometric series. The numerator in the brackets is always zero. The denominator is only zero if $l = m$. Hence, as long as $l \neq m$ the sum is zero. However, if $l = m$ we get $0/0$, which is undefined, and so, to get the answer, we either evaluate it as the limit $l \to m$ or go back the start and put $l = m$ from the beginning. In either method one finds that the term in rectangular brackets is equal to N. Hence the RHS of Eq. (16.5) is x_m, showing that the inverse transform in Eq. (16.4) does give back the original dataset x_m as claimed.

Note that $x_{m+N} = x_m$, so the x-values obtained from the inverse Fourier transform are actually a periodic repetition of the original data (i.e. the x_m for $m = 0, \cdots, N-1$) with period N.

The discrete Fourier transform can be conveniently written as

$$y_k = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \omega^{km} x_m\,, \tag{16.6}$$

where

$$\omega = \exp(2\pi i/N)\,. \tag{16.7}$$

To determine the FT, each application of Eq. (16.6) requires N additions and N multiplications for each of the N values of k, so the operation count is $O(N^2)$.

In the appendices of this chapter we describe the fast Fourier transform (FFT) which is a much more efficient way to calculate a discrete Fourier transform. We don't need the FFT for this course, but I include a description of it here in the appendices partly to stimulate students' interest in it (since it is a gem of computer science), and partly because it bears a strong resemblance to Shor's quantum Fourier transform (QFT), see Chapter 17, which is the heart of his factoring algorithm. We shall show this connection in the appendices of Chapter 17.

The Fast Fourier Transform (FFT) requires an operation count of only $N \log_2 N$ compared with N^2 which is needed for a straightforward evaluation of Eq. (16.6) for all k. This reduction (which is considerable for large N) is possible because ω^n is a periodic function of n with period N and so ω^{km} takes only N distinct values, even though $k\,m$ runs over $O(N^2)$ values.

The FFT is discussed in the appendices which now follow. As mentioned above, this material is not required for the rest of the course and can be omitted.

Appendices

16.A The Fast Fourier Transform; an example with $N = 8$

We will understand the Fast Fourier Transform (FFT) by first working out in detail a simple example. The number of data points N must be a power of 2. If it's not a power of 2 then one pads the data with zeroes to make it so. We will take $n = 3$, i.e. $N = 8$. Written out explicitly, the Fourier Transform for $N = 8$ data points is

$$y_0 = \tfrac{1}{\sqrt{8}}\left(x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \right], \tag{16.8a}$$
$$y_1 = \tfrac{1}{\sqrt{8}}\left(x_0 + \omega\, x_1 + \omega^2 x_2 + \omega^3 x_3 + \omega^4 x_4 + \omega^5 x_5 + \omega^6 x_6 + \omega^7 x_7 \right), \tag{16.8b}$$
$$y_2 = \tfrac{1}{\sqrt{8}}\left(x_0 + \omega^2 x_1 + \omega^4 x_2 + \omega^6 x_3 + x_4 + \omega^2 x_5 + \omega^4 x_6 + \omega^6 x_7 \right), \tag{16.8c}$$
$$y_3 = \tfrac{1}{\sqrt{8}}\left(x_0 + \omega^3 x_1 + \omega^6 x_2 + \omega\, x_3 + \omega^4 x_4 + \omega^7 x_5 + \omega^2 x_6 + \omega^5 x_7 \right), \tag{16.8d}$$
$$y_4 = \tfrac{1}{\sqrt{8}}\left(x_0 + \omega^4 x_1 + x_2 + \omega^4 x_3 + x_4 + \omega^4 x_5 + x_6 + \omega^4 x_7 \right), \tag{16.8e}$$
$$y_5 = \tfrac{1}{\sqrt{8}}\left(x_0 + \omega^5 x_1 + \omega^2 x_2 + \omega^7 x_3 + \omega^4 x_4 + \omega\, x_5 + \omega^6 x_6 + \omega^3 x_7 \right), \tag{16.8f}$$
$$y_6 = \tfrac{1}{\sqrt{8}}\left(x_0 + \omega^6 x_1 + \omega^4 x_2 + \omega^2 x_3 + x_4 + \omega^6 x_5 + \omega^4 x_6 + \omega^2 x_7 \right), \tag{16.8g}$$
$$y_7 = \tfrac{1}{\sqrt{8}}\left(x_0 + \omega^7 x_1 + \omega^6 x_2 + \omega^5 x_3 + \omega^4 x_4 + \omega^3 x_5 + \omega^2 x_6 + \omega\, x_7 \right), \tag{16.8h}$$

where the x_j are the original data, the y_j are the Fourier transformed data,

$$\omega = \exp(2\pi i/8) = \frac{1}{\sqrt{2}}(1 + i)\,, \tag{16.9}$$

and we note that

$$\omega^8 = 1 \ = \omega^0\,, \tag{16.10}$$

so we have reduced all the powers of ω to be between 0 and 7 ($= N - 1$). We also note that

$$\omega^2 = i,\ \omega^4 = -1\,. \tag{16.11}$$

To evaluate Eqs. (16.8) efficiently the FFT proceeds recursively. We firstly define Fourier transforms of length 2:

$$u_0 = \tfrac{1}{\sqrt{2}}(x_0 + x_4) \qquad = \tfrac{1}{\sqrt{2}}(x_0 + \omega^{4k} x_4)\ (k = 0)\,, \tag{16.12a}$$
$$u_1 = \tfrac{1}{\sqrt{2}}(x_1 + x_5) \qquad = \tfrac{1}{\sqrt{2}}(x_1 + \omega^{4k} x_5)\ (k = 0)\,, \tag{16.12b}$$
$$u_2 = \tfrac{1}{\sqrt{2}}(x_2 + x_6) \qquad = \tfrac{1}{\sqrt{2}}(x_2 + \omega^{4k} x_6)\ (k = 0)\,, \tag{16.12c}$$
$$u_3 = \tfrac{1}{\sqrt{2}}(x_3 + x_7) \qquad = \tfrac{1}{\sqrt{2}}(x_3 + \omega^{4k} x_7)\ (k = 0)\,, \tag{16.12d}$$
$$u_4 = \tfrac{1}{\sqrt{2}}(x_0 - x_4) \qquad = \tfrac{1}{\sqrt{2}}(x_0 + \omega^{4k} x_4)\ (k = 1)\,, \tag{16.12e}$$
$$u_5 = \tfrac{1}{\sqrt{2}}(x_1 - x_5) \qquad = \tfrac{1}{\sqrt{2}}(x_1 + \omega^{4k} x_5)\ (k = 1)\,, \tag{16.12f}$$
$$u_6 = \tfrac{1}{\sqrt{2}}(x_2 - x_6) \qquad = \tfrac{1}{\sqrt{2}}(x_2 + \omega^{4k} x_6)\ (k = 1)\,, \tag{16.12g}$$
$$u_7 = \tfrac{1}{\sqrt{2}}(x_3 - x_7) \qquad = \tfrac{1}{\sqrt{2}}(x_3 + \omega^{4k} x_7)\ (k = 1)\,. \tag{16.12h}$$

Pairs of quantities in Eqs. (16.12) are combined into Fourier Transforms of length 4:

$$v_0 = \tfrac{1}{\sqrt{2}}(u_0 + u_2) \qquad = \tfrac{1}{\sqrt{2}}(u_0 + \omega^{2k} u_2)\ (k = 0)\,, \tag{16.13a}$$
$$v_1 = \tfrac{1}{\sqrt{2}}(u_1 + u_3) \qquad = \tfrac{1}{\sqrt{2}}(u_1 + \omega^{2k} u_3)\ (k = 0)\,, \tag{16.13b}$$
$$v_2 = \tfrac{1}{\sqrt{2}}(u_4 + iu_6) \qquad = \tfrac{1}{\sqrt{2}}(u_4 + \omega^{2k} u_6)\ (k = 1)\,, \tag{16.13c}$$
$$v_3 = \tfrac{1}{\sqrt{2}}(u_5 + iu_7) \qquad = \tfrac{1}{\sqrt{2}}(u_5 + \omega^{2k} u_7)\ (k = 1)\,, \tag{16.13d}$$
$$v_4 = \tfrac{1}{\sqrt{2}}(u_0 - u_2) \qquad = \tfrac{1}{\sqrt{2}}(u_0 + \omega^{2k} u_2)\ (k = 2)\,, \tag{16.13e}$$
$$v_5 = \tfrac{1}{\sqrt{2}}(u_1 - u_3) \qquad = \tfrac{1}{\sqrt{2}}(u_1 + \omega^{2k} u_3)\ (k = 2)\,, \tag{16.13f}$$
$$v_6 = \tfrac{1}{\sqrt{2}}(u_4 - iu_6) \qquad = \tfrac{1}{\sqrt{2}}(u_4 + \omega^{2k} u_6)\ (k = 3)\,, \tag{16.13g}$$
$$v_7 = \tfrac{1}{\sqrt{2}}(u_5 - iu_7) \qquad = \tfrac{1}{\sqrt{2}}(u_5 + \omega^{2k} u_7)\ (k = 3)\,, \tag{16.13h}$$

and finally pairs of quantities in Eqs. (16.13) are combined to form the Fourier Transform in Eqs. (16.8):

$$y_0 = \tfrac{1}{\sqrt{2}}(v_0 + v_1) \qquad = \tfrac{1}{\sqrt{2}}(v_0 + \omega^{k} v_1)\ (k = 0)\,, \tag{16.14a}$$
$$y_1 = \tfrac{1}{\sqrt{2}}(v_2 + \omega\, v_3) \qquad = \tfrac{1}{\sqrt{2}}(v_2 + \omega^{k} v_3)\ (k = 1)\,, \tag{16.14b}$$
$$y_2 = \tfrac{1}{\sqrt{2}}(v_4 + iv_5) \qquad = \tfrac{1}{\sqrt{2}}(v_4 + \omega^{k} v_5)\ (k = 2)\,, \tag{16.14c}$$
$$y_3 = \tfrac{1}{\sqrt{2}}(v_6 + \omega^3 v_7) \qquad = \tfrac{1}{\sqrt{2}}(v_6 + \omega^{k} v_7)\ (k = 3)\,, \tag{16.14d}$$
$$y_4 = \tfrac{1}{\sqrt{2}}(v_0 - v_1) \qquad = \tfrac{1}{\sqrt{2}}(v_0 + \omega^{k} v_1)\ (k = 4)\,, \tag{16.14e}$$
$$y_5 = \tfrac{1}{\sqrt{2}}(v_2 - \omega v_3) \qquad = \tfrac{1}{\sqrt{2}}(v_2 + \omega^{k} v_3)\ (k = 5)\,, \tag{16.14f}$$
$$y_6 = \tfrac{1}{\sqrt{2}}(v_4 - iv_5) \qquad = \tfrac{1}{\sqrt{2}}(v_4 + \omega^{k} v_5)\ (k = 6)\,, \tag{16.14g}$$
$$y_7 = \tfrac{1}{\sqrt{2}}(v_6 - \omega^3 v_7) \qquad = \tfrac{1}{\sqrt{2}}(v_6 + \omega^{k} v_7)\ (k = 7)\,, \tag{16.14h}$$

Equations (16.12)–(16.14) are represented graphically by Fig. 16.1.

We see that the FFT, specified by Eqs. (16.12)–(16.14), requires 8×3 ($= N \log_2 N$ for $N = 8$) additions and multiplications, whereas a direct evaluation of the FT according to Eq. (16.8) takes 8×8 ($= N^2$) additions and multiplications. For large N, the speedup factor, $N/\log_2 N$, in using the FFT rather than direct evaluation of the FT is considerable.

Let's check that this works by evaluating y_1. We have

$$y_1 = \tfrac{1}{\sqrt{2}}\,(v_2 + \omega\, v_3)\ , \tag{16.15a}$$
$$= \tfrac{1}{2}\,(\,u_4 + iu_6 + \omega(u_5 + iu_7)\,) = \tfrac{1}{2}\,\big(\,u_4 + \omega^2 u_6 + \omega\, u_5 + \omega^3 u_7\,\big)\ , \tag{16.15b}$$
$$= \tfrac{1}{\sqrt{8}}\,\big(\,x_0 - x_4 + \omega^2(x_2 - x_6) + \omega(x_1 - x_5) + \omega^3(x_3 - x_7)\,\big)\ , \tag{16.15c}$$
$$= \tfrac{1}{\sqrt{8}}\,\big(\,x_0 + \omega\, x_1 + \omega^2 x_2 + \omega^3 x_3 + \omega^4 x_4 + \omega^5 x_5 + \omega^6 x_6 + \omega^7 x_7\,\big)\ , \tag{16.15d}$$

which agrees with Eq. (16.8b). We have used Eq. (16.14b) to get Eq. (16.15a), Eqs. (16.13c) and (16.13d) to get Eq. (16.15b), and Eqs. (16.12e), (16.12g), (16.12f) and (16.12h) to get Eq. (16.15c). Equation (16.15d) is the same as Eq. (16.15c) with powers of ω written out explicitly using Eq. (16.11).

It is instructive to write the linear transformations in Eqs. (16.8), (16.12), (16.13) and (16.14) in matrix form. Equation (16.8) is written in matrix formulation as

$$\vec{y} = U\vec{x}\,, \tag{16.16}$$

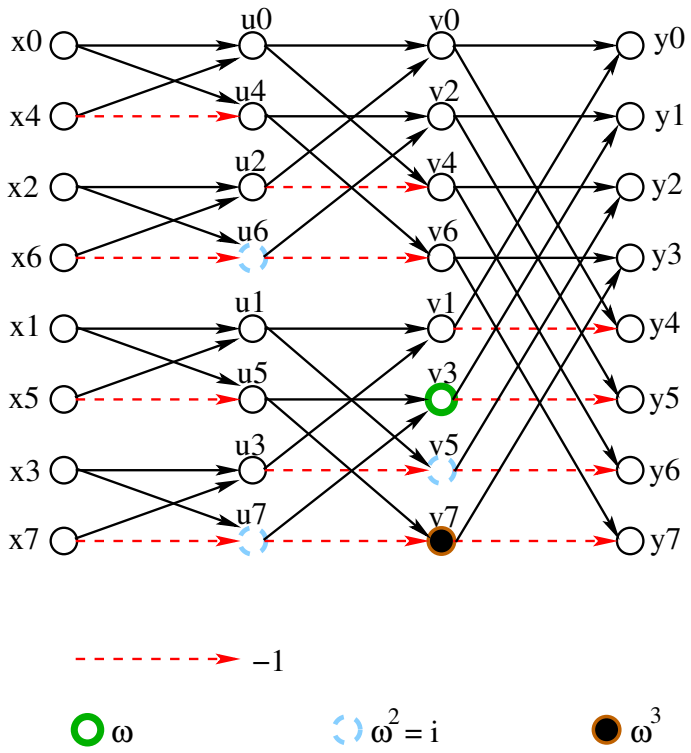


Figure 16.1: A graphical representation of Eqs. (16.12)–(16.14), which is the FFT for $N = 8\,(= 2^n$ with $n = 3)$. The original data are the x_j and the Fourier transformed data are the y_j. The dashed (red) lines have a factor of -1 and the solid lines have a factor of 1. The thick (green) circle transmits a factor of ω to the right, the dashed (blue) circles transmit a factor of ω^2 $(= i)$ to the right, and the (brown) filled-in circle transmits a factor of ω^3 to the right. In Sec. 16.C we will change to a notation applicable for general n, as follows: $y_j \equiv x_j^{(0)}, v_j \equiv x_j^{(1)}, u_j \equiv x_j^{(2)}$, and $x_j = x_j^{(3)}$. (Adapted from R. Vathsan *Introduction to Quantum Physics and Information Processing*.)

where

$$U = \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega \end{pmatrix} . \tag{16.17}$$

Equation (16.12) in matrix form is

$$\vec{u} = D\vec{x}\,, \tag{16.18}$$

where

$$D = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & \omega^4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \omega^4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \omega^4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega^4 \end{pmatrix} . \tag{16.19}$$

Equation (16.13) in matrix form is

$$\vec{v} = E\vec{u}\,, \tag{16.20}$$

where

$$E = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \omega^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^2 \\ 1 & 0 & \omega^4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \omega^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \omega^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^6 \end{pmatrix} . \tag{16.21}$$

Equation (16.14) in matrix form is

$$\vec{y} = F\vec{v}\,, \tag{16.22}$$

where

$$F = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \omega & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \omega^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega^3 \\ 1 & \omega^4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \omega^5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \omega^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega^7 \end{pmatrix} . \tag{16.23}$$

Notice that D, E and F, which describe the FFT, are very sparse, they have only two entries in each row and column, so they can be multiplied very efficiently, whereas the matrix U, which describes the original Fourier transform, is dense. With some tedious matrix manipulations one can verify that

$$U = F\,E\,D\,, \tag{16.24}$$

as required. (I used *Mathematica*.)

16.B Beyond $N = 8$

Now we discuss how we obtained Eqs. (16.12)–(16.14). For a general value n, with $N = 2^n$, the FT is defined by

$$y_k = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \omega^{km} x_m\,, \qquad (k = 0, 1, \cdots, N-1) \tag{16.25}$$

with ω given by Eq. (16.7). We can break Eq. (16.25) into even and odd terms as follows:

$$y_k = \frac{1}{\sqrt{N}}\left[\sum_{m=0}^{N/2-1}\omega^{2km}x_{2m} + \sum_{m=0}^{N/2-1}\omega^{k(2m+1)}x_{2m+1}\right],$$
$$= \frac{1}{\sqrt{2}}\left[\sqrt{\frac{2}{N}}\sum_{m=0}^{N/2-1}(\omega^2)^{km}x_{2m} + \omega^k\sqrt{\frac{2}{N}}\sum_{m=0}^{N/2-1}(\omega^2)^{km}x_{2m+1}\right], \qquad (k = 0, 1, \cdots, N-1). \quad (16.26)$$

Noting that ω^2 is the complex exponential factor analogous to Eq. (16.7) which figures in a Fourier Transform with $N/2$ points, we see that the first term in Eq. (16.26) is a FT for the $N/2$ even points and the second term is the FT for the $N/2$ odd points. We can write Eq. (16.26) as

$$\boxed{y_k = \tfrac{1}{\sqrt{2}}\left[v_{2k} + \omega^k v_{2k+1}\right], \quad (k = 0, 1, \cdots, N-1),} \qquad (16.27)$$

where

$$v_{2k} = \sqrt{\frac{2}{N}}\sum_{m=0}^{N/2-1}(\omega^2)^{km}x_{2m}, \qquad (16.28a)$$

$$v_{2k+1} = \sqrt{\frac{2}{N}}\sum_{m=0}^{N/2-1}(\omega^2)^{km}x_{2m+1}, \quad (k = 0, 1, \cdots, N-1). \qquad (16.28b)$$

Here k runs over the range $0, 1, \cdots, N-1$ so the indices on the v_j in Eqs. (16.28) run from 0 to $2N-1$. However, since $\omega^N = 1$, see Eq. (16.7), it follows from the definition of the v_j in Eq, (16.28) that $v_{j+N} = v_j$. Hence the index j, of the v_j is to be evaluated modulo N. This applies in an obvious way to other quantities as well, such as the u_j, and, in Sec. 16.C, to the lower index on the $x_j^{(\ell)}$.

For $N = 8$ please check that Eq. (16.27) corresponds to our Eqs. (16.14) for $k = 0, 1, 2, \cdots 7$ and that, according to Eqs. (16.28), the expressions for the v_k in terms of the original data x_m are

$$v_0 = \tfrac{1}{2}\sum_{m=0}^{3}x_{2m}, \qquad v_2 = \tfrac{1}{2}\sum_{m=0}^{3}(\omega^2)^m x_{2m}, \qquad v_4 = \tfrac{1}{2}\sum_{m=0}^{3}(\omega^2)^{2m}x_{2m}, \qquad v_6 = \tfrac{1}{2}\sum_{m=0}^{3}(\omega^2)^{3m}x_{2m}, \qquad (16.29a)$$

$$v_1 = \tfrac{1}{2}\sum_{m=0}^{3}x_{2m+1}, \quad v_3 = \tfrac{1}{2}\sum_{m=0}^{3}(\omega^2)^m x_{2m+1}, \quad v_5 = \tfrac{1}{2}\sum_{m=0}^{3}(\omega^2)^{2m}x_{2m+1}, \quad v_7 = \tfrac{1}{2}\sum_{m=0}^{3}(\omega^2)^{3n}x_{2m+1}, \qquad (16.29b)$$

so v_0, v_2, v_4 and v_6 are the FT of the 4 even points for $k = 0, 1, 2$ and 3 respectively, while v_1, v_3, v_5 and v_7 are the FT of the 4 odd points for $k = 0, 1, 2$ and 3 respectively.

We can again separate each of Eqs. (16.28) into even and odd terms by analogy with Eq. (16.26). We have

$$v_{2k} = \sqrt{\frac{2}{N}}\left[\sum_{m=0}^{N/4-1}(\omega^4)^{km}x_{4m} + \left(\omega^2\right)^k\sum_{m=0}^{N/4-1}(\omega^4)^{km}x_{4m+2}\right], \qquad (16.30a)$$

$$v_{2k+1} = \sqrt{\frac{2}{N}}\left[\sum_{m=0}^{N/4-1}(\omega^4)^{km}x_{4m+1} + \left(\omega^2\right)^k\sum_{m=0}^{N/4-1}(\omega^4)^{km}x_{4m+3}\right]. \qquad (16.30b)$$

We can write these equations as

$$v_{2k} = \tfrac{1}{\sqrt{2}}\left[u_{4k} + \left(\omega^2\right)^k u_{4k+2}\right], \tag{16.31a}$$

$$v_{2k+1} = \tfrac{1}{\sqrt{2}}\left[u_{4k+1} + \left(\omega^2\right)^k u_{4k+3}\right], \quad (k = 0, 1, \cdots, N/2-1), \tag{16.31b}$$

where

$$u_{4k} = \sqrt{\frac{4}{N}} \sum_{m=0}^{N/4-1} (\omega^4)^{km} x_{4m}, \qquad u_{4k+1} = \sqrt{\frac{4}{N}} \sum_{m=0}^{N/4-1} (\omega^4)^{km} x_{4m+1}, \tag{16.32a}$$

$$u_{4k+2} = \sqrt{\frac{4}{N}} \sum_{m=0}^{N/4-1} (\omega^4)^{km} x_{4m+2}, \qquad u_{4k+3} = \sqrt{\frac{4}{N}} \sum_{m=0}^{N/4-1} (\omega^4)^{km} x_{4m+3}. \tag{16.32b}$$

Note that the two equations in Eqs. (16.31) can be combined as

$$\boxed{v_{2k+p} = \tfrac{1}{\sqrt{2}}\left[u_{4k+p} + \left(\omega^2\right)^k u_{4k+p+2}\right], \quad (p = 0, 1), (k = 0, 1, \cdots, N/2-1).} \tag{16.33}$$

Again, the index j on the u_j is to be evaluated modulo N.

For $N = 8$ please check that Eq. (16.33) corresponds to our Eqs. (16.13) for $p = 0, 1$, and $k = 0, 1, 2$ and 3, and that, according to Eqs. (16.32), the explicit expressions for the u_j are

$$u_0 = \tfrac{1}{\sqrt{2}} \sum_{m=0}^{1} x_{4m} = \tfrac{1}{\sqrt{2}}(x_0 + x_4), \qquad u_1 = \tfrac{1}{\sqrt{2}} \sum_{m=0}^{1} x_{4m+1} = \tfrac{1}{\sqrt{2}}(x_1 + x_5), \tag{16.34a}$$

$$u_2 = \tfrac{1}{\sqrt{2}} \sum_{m=0}^{1} x_{4m+2} = \tfrac{1}{\sqrt{2}}(x_2 + x_6), \qquad u_3 = \tfrac{1}{\sqrt{2}} \sum_{m=0}^{1} x_{4m+3} = \tfrac{1}{\sqrt{2}}(x_3 + x_7), \tag{16.34b}$$

$$u_4 = \tfrac{1}{\sqrt{2}} \sum_{m=0}^{1} (\omega^4)^m x_{4m} = \tfrac{1}{\sqrt{2}}(x_0 - x_4), \qquad u_5 = \tfrac{1}{\sqrt{2}} \sum_{m=0}^{1} (\omega^4)^m x_{4m+1} = \tfrac{1}{\sqrt{2}}(x_1 - x_5), \tag{16.34c}$$

$$u_6 = \tfrac{1}{\sqrt{2}} \sum_{m=0}^{1} (\omega^4)^m x_{4m+2} = \tfrac{1}{\sqrt{2}}(x_2 - x_6), \quad u_7 = \tfrac{1}{\sqrt{2}} \sum_{m=0}^{1} (\omega^4)^n x_{4m+3} = \tfrac{1}{\sqrt{2}}(x_3 - x_7). \tag{16.34d}$$

Equations (16.34) agree with the expressions in Eq. (16.12). They can be written as a single equation as

$$\boxed{u_{4k+p} = \tfrac{1}{\sqrt{2}}[x_p + (-1)^k x_{p+4}], \quad (p = 0, 1, 2, 3), (k = 0, 1).} \tag{16.35}$$

Thus we have seen that the FFT for $N = 8\,(= 2^n$ with $n = 3)$, which is written out explicitly in Eqs. (16.12)–(16.14), corresponds to firstly doing the Fourier transforms of length 2 in Eq. (16.35), followed by two applications of the iterative procedure, the first shown in Eq. (16.33) and the second shown in Eq. (16.27).

16.C The General Case

So far we have unsystematically labeled the results at each stage of iteration by a different symbol, $x \to u \to v \to y$, see Fig. 16.1. When writing a code applicable for $N = 2^n$ data points for arbitrary

n, one would use a common symbol but add a second index, so

$$x_j \equiv x_j^{(n)}\,, \tag{16.36a}$$

$$\vdots$$

$$u_j \equiv x_j^{(2)}\,, \tag{16.36b}$$

$$v_j \equiv x_j^{(1)}\,, \tag{16.36c}$$

$$y_j \equiv x_j^{(0)}\,. \tag{16.36d}$$

Note that since $\omega = \exp(2\pi i/2^n)$ we have

$$\omega^{2^n} = \exp(2\pi i) = 1, \quad \omega^{2^{n-1}} = \exp(\pi i) = -1. \tag{16.37}$$

The ℓ-th iteration, analogous to Eqs. (16.33), (16.27) and (16.35) is

$$\boxed{x^{(\ell-1)}_{2^{\ell-1}k+p} = \tfrac{1}{\sqrt{2}}\left[x^{(\ell)}_{2^\ell k+p} + (\omega^{2^{\ell-1}})^k x^{(\ell)}_{2^\ell k+p+2^{\ell-1}}\right]\,,} \tag{16.38}$$

with

$$p = 0, 1, \cdots, 2^{\ell-1}-1, \quad k = 0, 1, \cdots, 2^{n-\ell+1}-1\,. \tag{16.39}$$

Sorry that the notation is messy but I can't see how to improve it; one just has to keep track of the indices and the powers of ω. Recall that the lower index j on the $x_j^{(\ell)}$ is to be evaluated modulo 2^n.

Let's see how this works.

- We start with $\ell = n$, for which $x_j^{(\ell)} \equiv x_j$, the original data points.
  Equation (16.38) is then

  $$x^{(n-1)}_{2^{n-1}k+p} = \tfrac{1}{\sqrt{2}}\left[x_p + (-1)^k x_{p+2^{n-1}}\right]\,, \quad (p = 0, 1, \cdots, 2^{n-1}-1),\ (k = 0, 1)\,. \tag{16.40}$$

  For $n = 3\,(N = 8)$ this corresponds to Eq. (16.35) with $x_j^{(n-1)} \equiv u_j$.

- We then iterate Eq. (16.38) for $\ell = n-1, n-2, \cdots, 2, 1$.
  At the next to the last iteration, $\ell = 2$, we have

  $$x^{(1)}_{2k+p} = \tfrac{1}{\sqrt{2}}\left[x^{(2)}_{4k+p} + (\omega^2)^k x^{(2)}_{4k+p+2}\right]\,, \quad (p = 0, 1),\ (k = 0, 1, \cdots, 2^{n-1}-1)\,, \tag{16.41}$$

  which corresponds to Eq. (16.33) with, $x_j^{(1)} \equiv v_j, x_j^{(2)} \equiv u_j$. At the last iteration, $\ell = 1$, we obtain

  $$y_k = \tfrac{1}{\sqrt{2}}\left[x^{(1)}_{2k} + \omega^k x^{(1)}_{2k+1}\right]\,, \qquad (k = 0, 1, 2, \cdots, 2^n-1)\,, \tag{16.42}$$

  which is Eq. (16.27). (Recall that $x_j^{(0)} \equiv y_j$, the Fourier transformed data, and $x_j^{(1)} \equiv v_j$.)

Note that the iterations are evaluated in reverse, starting with $\ell = n$ and working down to $\ell = 1$.

Chapter 17

The Quantum Fourier Transform (QFT)

17.1 Introduction

This chapter introduces the quantum Fourier transform (QFT), which is at the heart of Shor's algorithm for period finding, and hence for factoring. Shor's algorithm will be discussed in Chapter 18. The appendices of this chapter make a detailed comparison with the (classical) Fast Fourier Transform(FFT). The FFT is not part of the course so if you are not interested in this comparison you can ignore the appendices.

The QFT can be defined as follows. Starting with n qubits in a single computational basis state $|x\rangle_n$, where x is an n-bit integer, one generates the following superposition:

$$|x\rangle_n \xrightarrow{\text{QFT}} |\psi_x\rangle_n = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} \exp[2\pi i x y/2^n] |y\rangle_n \tag{17.1}$$

where y is also an n-bit integer. The real power of the QFT arises, of course, because it acts *in parallel* if one inputs a superposition $\sum_{x=0}^{2^n-1} a_x |x\rangle_n$, i.e.

$$\sum_{x=0}^{2^n-1} a_x |x\rangle_n \xrightarrow{\text{QFT}} \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} a_x \sum_{y=0}^{2^n-1} \exp[2\pi i x y/2^n] |y\rangle_n = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} \left[\sum_{x=0}^{2^n-1} a_x \exp[2\pi i x y/2^n] \right] |y\rangle_n. \tag{17.2}$$

The circuit to perform the QFT, the derivation of which is the main topic of this chapter and which is shown below in Fig. 17.5, takes no more time to act on the superposition in Eq. (17.2) than on the single basis state in Eq. (17.1). This is where the power of the QFT lies.

Note that the effect of the QFT acting on a superposition, given in Eq. (17.2), can be written as

$$\sum_{x=0}^{2^n-1} a_x |x\rangle_n \xrightarrow{\text{QFT}} \sum_{y=0}^{2^n-1} a'_y |y\rangle_n, \tag{17.3}$$

where the transformed amplitudes a'_y are related to the original amplitudes a_x by

$$a'_y = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} \exp[2\pi i x y/2^n]\, a_x \tag{17.4}$$

which is a discrete Fourier transform on the *amplitudes*. This transformation of the amplitudes is a useful alternative way of defining a QFT, and is equivalent to Eq. (17.1).

17.2 QFT with two qubits

To help us derive the circuit which generates the transformation in Eq. (17.1) we start with just $n = 2$ qubits.

The Quantum Fourier Transform (QFT)in Eq. (17.1) for $n = 2$ qubits is

$$|\psi_x\rangle_2 = \frac{1}{2}\sum_{y=0}^{3} \exp\left[2\pi i x y/2^2\right]\ |y\rangle_2, \tag{17.5}$$

where $|x\rangle_2 \equiv |x_1 x_0\rangle$ and $|y\rangle_2 \equiv |y_1 y_0\rangle$. Note this has the same structure as a standard Fourier Transform, but here we are performing a change of basis of a quantum system, rather than transforming a set of data. The $|\psi_x\rangle_2$ form a basis just as the $|x\rangle_2$ form a basis because one can show that they are orthonormal, i.e.

$$_2\langle\psi_x|\psi_{x'}\rangle_2 = \delta_{x,x'}\,. \tag{17.6}$$

Noting that $y = y_0 + 2y_1$ and $x = x_0 + 2x_1$ we can simplify the argument of the exponential:

$$\frac{2\pi i x y}{2^2} = \frac{2\pi i (x_0 + 2x_1)(y_0 + 2y_1)}{2^2} = 2\pi i \left\{ y_0 \left(\frac{x_0}{4} + \frac{x_1}{2}\right) + y_1 \left(\frac{x_0}{2} + x_1\right)\right\}. \tag{17.7}$$

Now $\exp(2\pi i y_1 x_1) = 1$ so the factor $y_1 x_1$ above can be neglected. Hence Eq. (17.5) becomes

$$|\psi_x\rangle_2 = \left(\frac{1}{\sqrt{2}}\sum_{y_0=0}^{1} \exp\left[2\pi i y_0 \left(\frac{x_0}{4} + \frac{x_1}{2}\right)\right]\right)\ \left(\frac{1}{\sqrt{2}}\sum_{y_1=0}^{1} \exp\left[2\pi i y_1 \frac{x_0}{2}\right]\right) |y_1 y_0\rangle. \tag{17.8}$$

Next we will explain how to perform the operations in Eq. (17.8) using quantum gates.

Consider the second factor on the RHS of Eq. (17.8), which involves a sum over y_1. If $x_0 = 1$ the exponential is 1 for $y_1 = 0$ and is -1 for $y_1 = 1$. If $x_0 = 0$ the exponential is always 1. This functionality is provided by a Hadamard gate H, since

$$H|x_0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{x_0}|1\rangle\right) = \frac{1}{\sqrt{2}}\sum_{y_1=0}^{1}(-1)^{x_0 y_1}|y_1\rangle = \frac{1}{\sqrt{2}}\sum_{y_1=0}^{1}\exp[2\pi i y_1 x_0/2]|y_1\rangle\,, \tag{17.9}$$

where we denote by y_1 the dummy summation variable in order to correspond with the notation in the second factor on the RHS of Eq. (17.8). We therefore see that the second factor on the RHS of Eq. (17.8) is generated by the Hadamard gate shown in Fig. 17.1.

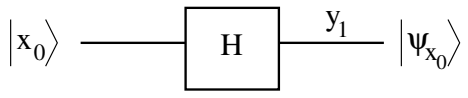


Figure 17.1: The output from the Hadamard gate is $|\psi_{x_0}\rangle = \frac{1}{\sqrt{2}}(|y_1{=}0\rangle + (-1)^{x_0}|y_1{=}1\rangle)$. This can be expressed as $\frac{1}{\sqrt{2}}\sum_{y_1=0}^{1}(-1)^{x_0 y_1}|y_1\rangle = \frac{1}{\sqrt{2}}\sum_{y_1=0}^{1}\exp[2\pi i y_1 x_0/2]|y_1\rangle$. Recall that x_0 takes a fixed value 0 or 1.

What about the first factor on the RHS of Eq. (17.8) which involves y_0? There are two pieces in the exponential. The one involving $2\pi i y_0 x_1/2$ can be dealt with by a Hadamard, similar to Fig. 17.1 but with the left hand qubit being x_1 and the right hand qubit being labeled by y_0. However, the piece

involving $2\pi i y_0 x_0/4$ is different. It induces a phase shift of $e^{i\pi/2}$ for $y_0 = 1$ provided that x_0 is also 1. This requires a controlled phase gate. We define a phase gate R_d by[1]

$$R_d = \begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i/2^d} \end{pmatrix} . \tag{17.10}$$

Acting on $|0\rangle, R_d$ makes no change, while acting on $|1\rangle$ R_d changes the phase by $\pi/2^d$. Note that R_0 is just the Z gate. Here we need R_1.

Hence the exponential in the first term on the RHS of Eq. (17.8) can be generated by a Hadamard followed by a controlled R_1 gate as shown for the top qubit in Fig. 17.2, in which the R_1 gate on the upper qubit is controlled by the lower qubit, x_0. Including the Hadamard on the lower qubit, Fig. 17.2 generates both factors on the RHS of Eq. (17.8).

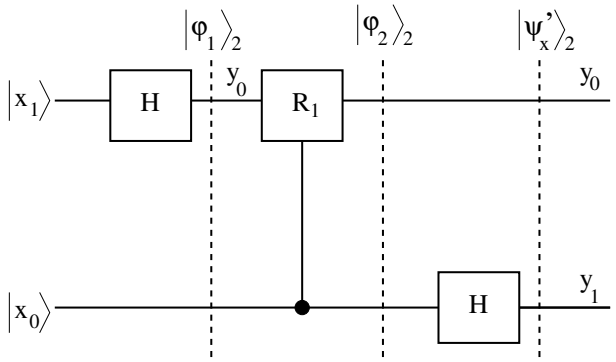


Figure 17.2: The initial state on the left is the single quantum state $|x\rangle_2 \equiv |x_1 x_0\rangle$ in the computational basis. The final state on the right is the superposition $|\psi'_x\rangle_2 = (1/2)\sum_{y=0}^{3} \exp(2\pi i x y/2^2)|y_0 y_1\rangle$, which is almost $|\psi_x\rangle_2$, the QFT of $|x\rangle_2 \equiv |x_1 x_0\rangle$ given in Eq. (17.8), except that the order of the bits in the final state is the reverse of what it should be according to Eq. (17.8). This can be corrected by a swap gate as shown in Fig. 17.3. Note the controlled-R_1 phase gate. This acts if the control qubit, x_0, is 1, and changes the phase of the state if the target qubit, y_0, is also equal to 1. The general phase gate R_d is defined in Eq. (17.10).

To make sure we understand we understand what is happening in the circuit in Fig. 17.2 we now write down the state at each of the steps shown in the figure. The initial state is

$$|x\rangle_2 = |x_1 x_0\rangle. \tag{17.11a}$$

After the first Hadamard the state is

$$|\phi_1\rangle_2 = \frac{1}{\sqrt{2}} \sum_{y_0=0}^{1} e^{2\pi i y_0 x_1/2} |y_0 x_0\rangle. \tag{17.11b}$$

After the controlled-R_1 gate we have

$$|\phi_2\rangle_2 = \frac{1}{\sqrt{2}} \sum_{y_0=0}^{1} e^{2\pi i y_0 x_1/2}\, e^{2\pi i y_0 x_0/4} |y_0 x_0\rangle. \tag{17.11c}$$

[1]This is the definition of R_d given in Vathsan's book, and I find it convenient. Some other authors adopt a slightly different definition with a factor of $e^{2\pi i/2^d}$ instead of $e^{\pi i/2^d}$.

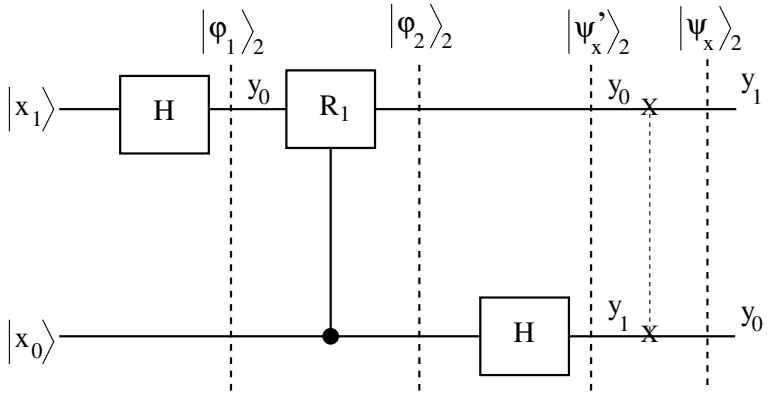


Figure 17.3: The same as Fig. 17.2 but with the addition of a swap gate on the right (the dashed line with crosses at the ends). The final state is now precisely $|\psi_x\rangle_2 = (1/2)\sum_{y=0}^{3}\exp(2\pi ixy/2^2)|y_1y_0\rangle$, the QFT given in Eq. (17.8).

The final state after the Hadamard on the lower qubit is therefore

$$|\psi'_x\rangle_2 = \left(\frac{1}{\sqrt{2}}\sum_{y_0=0}^{1} e^{2\pi i y_0 x_1/2}\, e^{2\pi i y_0 x_0/4}\right)\left(\frac{1}{\sqrt{2}}\sum_{y_1=0}^{1} e^{2\pi i y_1 x_0/2}\right)|y_0y_1\rangle. \tag{17.11d}$$

$|\psi'_x\rangle$ is almost the desired QFT in Eq. (17.8), except that the order of the qubits on in the final state on the right has been reversed. This can be compensated for by adding a swap gate on the right as shown in Fig. 17.3. Hence Fig. 17.3 displays the circuit to implement the QFT for 2 qubits written out in Eq. (17.8).

17.3 QFT with three or more qubits

We next do another special case, this time with $n = 3$ qubits. After this, we will be able to see the structure of the circuit for *general* n.

The QFT analogous to Eq. (17.8) is

$$|\psi_x\rangle_3 = \frac{1}{2^{3/2}}\sum_{y=0}^{7}\exp[2\pi ixy/2^3]|y\rangle_3 \tag{17.12}$$

$$= \left(\frac{1}{\sqrt{2}}\sum_{y_0=0}^{1}\exp\left[2\pi i y_0\left(\frac{x_0}{8}+\frac{x_1}{4}+\frac{x_2}{2}\right)\right]\right)\left(\frac{1}{\sqrt{2}}\sum_{y_1=0}^{1}\exp\left[2\pi i y_1\left(\frac{x_0}{4}+\frac{x_1}{2}\right)\right]\right) \times\left(\frac{1}{\sqrt{2}}\sum_{y_2=0}^{1}\exp\left[2\pi i y_2\frac{x_0}{2}\right]\right)|y_2y_1y_0\rangle, \tag{17.13}$$

where we have again replaced factors of $\exp(2\pi i \times \text{integer})$ by unity.

Following along the lines in the previous section, the circuit diagram which will perform this is shown in Fig. 17.4. To make sure we understand this circuit we will write down the state at each stage indicated on the figure. (Although these expressions look rather complicated is useful to make the effort to understand them.) The initial state is

$$|x\rangle_3 = |x_2x_1x_0\rangle, \tag{17.14a}$$

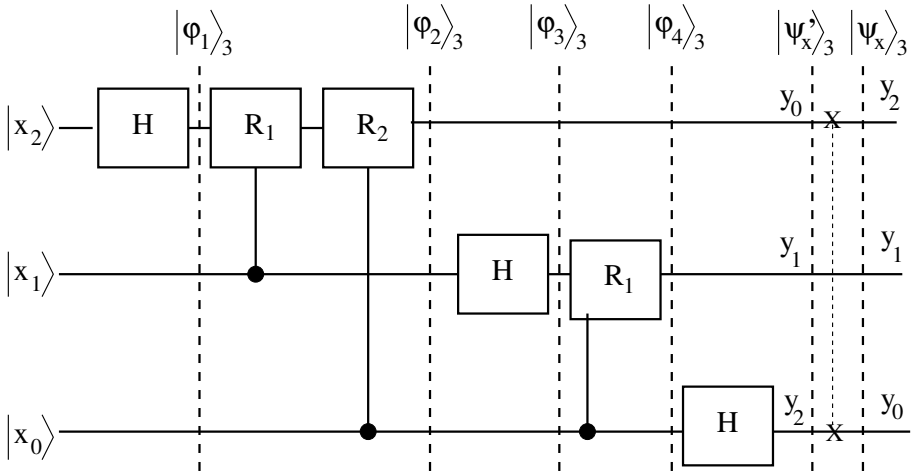


Figure 17.4: Circuit diagram for performing the QFT with $n = 3$ qubits. It generates the transformation shown in Eq. (17.13). The initial state is $|x\rangle_3 = |x_2x_1x_0\rangle$ and the subsequent states are given in Eqs. (17.14). The phase gates, R_d, are defined in Eq. (17.10). The dashed line with crosses at the ends indicates a swap gate between qubits 0 and 2. This serves to reverse the order of the qubits.

and the subsequent states, labeled in Fig. 17.4, are

$$|\phi_1\rangle_3 = \left(\frac{1}{\sqrt{2}}\sum_{y_0=0}^{1}\exp\left[2\pi i y_0\left(\frac{x_2}{2}\right)\right]\right)|y_0x_1x_0\rangle, \tag{17.14b}$$

$$|\phi_2\rangle_3 = \left(\frac{1}{\sqrt{2}}\sum_{y_0=0}^{1}\exp\left[2\pi i y_0\left(\frac{x_0}{8}+\frac{x_1}{4}+\frac{x_2}{2}\right)\right]\right)|y_0x_1x_0\rangle, \tag{17.14c}$$

$$|\phi_3\rangle_3 = \left(\frac{1}{\sqrt{2}}\sum_{y_0=0}^{1}\exp\left[2\pi i y_0\left(\frac{x_0}{8}+\frac{x_1}{4}+\frac{x_2}{2}\right)\right]\right)\left(\frac{1}{\sqrt{2}}\sum_{y_1=0}^{1}\exp\left[2\pi i y_1\left(\frac{x_1}{2}\right)\right]\right)|y_0y_1x_0\rangle, \tag{17.14d}$$

$$|\phi_4\rangle_3 = \left(\frac{1}{\sqrt{2}}\sum_{y_0=0}^{1}\exp\left[2\pi i y_0\left(\frac{x_0}{8}+\frac{x_1}{4}+\frac{x_2}{2}\right)\right]\right)\left(\frac{1}{\sqrt{2}}\sum_{y_1=0}^{1}\exp\left[2\pi i y_1\left(\frac{x_0}{4}+\frac{x_1}{2}\right)\right]\right)|y_0y_1x_0\rangle, \tag{17.14e}$$

$$|\psi'_x\rangle_3 = \left(\frac{1}{\sqrt{2}}\sum_{y_0=0}^{1}\exp\left[2\pi i y_0\left(\frac{x_0}{8}+\frac{x_1}{4}+\frac{x_2}{2}\right)\right]\right)\left(\frac{1}{\sqrt{2}}\sum_{y_1=0}^{1}\exp\left[2\pi i y_1\left(\frac{x_0}{4}+\frac{x_1}{2}\right)\right]\right)\times\left(\frac{1}{\sqrt{2}}\sum_{y_2=0}^{1}\exp\left[2\pi i y_2\frac{x_0}{2}\right]\right)|y_0y_1y_2\rangle. \tag{17.14f}$$

$|\psi'_x\rangle$ is almost the desired QFT in Eq. (17.13), except that the order of the qubits on in the final state on the right has been reversed. This can be compensated for by adding a swap gate between qubits 1 and 3. Hence $|\psi_x\rangle$ in the figure is the desired QFT for 3 qubits given in Eq. (17.13).

Intuitively, the reason that for the reverse order of the qubits in the final state before the swaps, is the following. The Hadamards generate the superpositions, i.e. the sums over the y_j. They also produce the factors in the exponential involving $2\pi i/2$. From the straightforward generalization of Eq. (17.13)

to arbitrary n, one sees that the factors generated by the Hadamards are $(2\pi i/2)\sum_{j=0}^{n-1} x_j y_{n-j}$. Here x_j is the label of the j-th physical qubit in its initial state, and y_{n-j} is the dummy label for the state of the same physical qubit in its final state. Because it is the label y_{n-j} (rather than y_j) which occurs on the same physical qubit as x_j, the qubits in the final state are in reverse order.

Comparing with the case for two qubits shown in Fig. 17.3, and that for three qubits in Fig. 17.4, the generalization to an arbitrary number of qubits can be deduced and is shown in Fig. 17.5. Note that the controlled phase gate between qubits x_i and x_j is $R_{|i-j|}$, which makes the structure fairly simple.

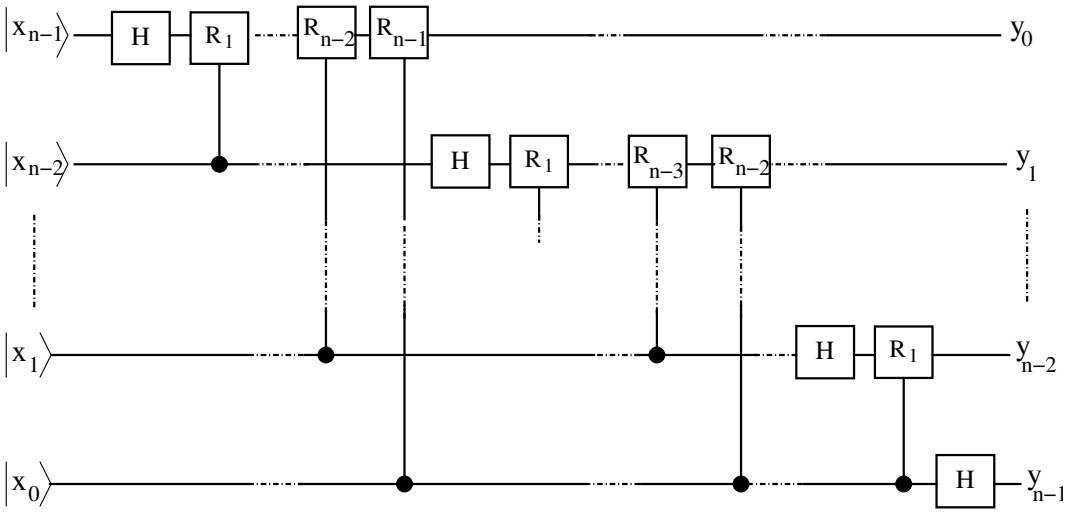


Figure 17.5: Circuit diagram for performing the QFT with an arbitrary number of qubits. For clarity the final swaps are not shown, so the input states on the left, x_i, and the output states on the right, y_i, are in opposite order. Note that the controlled phase gate between qubits x_i and x_j is $R_{|i-j|}$, which makes the structure fairly simple. If the state inputted in on the left is a single computational basis state $|x\rangle_n$, then, if we include the final swaps, the state outputted on the right is the superposition in Eq. (17.1).

For an n-qubit QFT one needs n Hadamard gates. The number of controlled phase gates is $1 + 2 + \cdots + n - 1 = n(n-1)/2$. Also $[n/2]$ swaps are required, where $[k]$ denotes the largest integer less than or equal to k. The circuit therefore provides an algorithm for performing the QFT in $O(n^2)$ steps. By contrast the FFT requires $O(n2^n)$ steps which is exponentially greater.

However, we cannot obtain the 2^n Fourier amplitudes from the QFT since a measurement will just give one of the basis states with a probability proportional to the square of the absolute value of its Fourier amplitude. However, the QFT does give useful information if the input state is a linear combination $\sum_x a_x|x\rangle$, see Eq. (17.2), in which the a_x are *periodic* in x with some period r. As we shall see in Chapter 18 the Fourier amplitudes are then strongly peaked at values of y which are multiples of $2^n/r$, so there is a high probability that a measurement of y will give a value which is equal, or close, to a multiple of $2^n/r$. As we shall see in Chapter 18, from this information one can then deduce the period r with high probability. Hence the QFT is very useful for period finding.

As we saw in Chapter 15, period finding can be used to factor integers. If one could factor large integers, one would be able to decode messages sent down the internet which have been encoded with the standard RSA encryption method. We discussed RSA encryption in Chapter 14.

Another application of the QFT is to estimate the phase of the eigenvalues of a unitary matrix. This is discussed in section 17.4.

17.4 The Phase Estimation Algorithm

The eigenvalue of a unitary operator U must be a pure phase, i.e. $\lambda = e^{i\theta}$. The reason is that U preserves the norm of states, so if $|\psi'\rangle = U|\psi\rangle$, we have

$$\langle\psi'|\psi'\rangle = \langle U\psi|U\psi\rangle = \langle\psi|U^\dagger U|\psi\rangle = \langle\psi|\psi\rangle = 1, \tag{17.15}$$

since $U^\dagger U = \mathbb{1}$. If $|\psi\rangle$ is an eigenstate of U, i.e. $|\psi'\rangle = \lambda|\psi\rangle$ this last equation becomes

$$\langle\psi'|\psi'\rangle = \langle\lambda\psi|\lambda\psi\rangle = \langle\psi|\lambda^\star\lambda|\psi\rangle = |\lambda|^2\langle\psi|\psi\rangle = |\lambda|^2, \tag{17.16}$$

so $|\lambda|^2 = 1$, and hence $\lambda = e^{i\theta}$ for some θ.

The objective of this section is to determine an eigenvalue of a unitary matrix, which is equivalent to determining its (complex) phase (since, as we just showed, its modulus is 1). Hence this problem is called "phase estimation".

Let us write

$$\theta = 2\pi\phi \tag{17.17}$$

so $0 \le \phi < 1$. The result for the phase will be encoded as an integer (formed from the values of the measured qubits) and let's suppose we want to determine ϕ correct to n bits of precision. The procedure is to compute an n-bit integer ϕ', related to ϕ and θ by

$$\phi' = 2^n\phi \ \left(= 2^n\frac{\theta}{2\pi}\right), \qquad \text{so} \ \ \theta = 2\pi\frac{\phi'}{2^n}. \tag{17.18}$$

The possible values of ϕ' are $0, 1, 2, \cdots, 2^n - 1$.

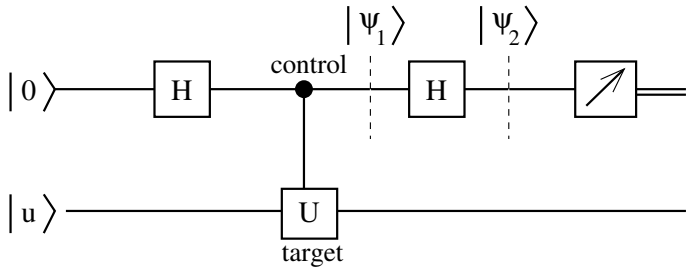


Figure 17.6: The circuit for phase estimation for 1 bit of precision.

We start with a simple example in which we only require 1 bit accuracy, so $\phi' = 0$ or 1. We will see that circuit in Fig. 17.6 does the trick. Figure 17.6 is essentially the same as Fig. 8.8 in Chapter 8. Here we assume that $|u\rangle$ is an eigenstate of U with eigenvalue $\exp(2\pi i\phi'/2)$. Following the discussion after Fig. 8.8 we find that

$$\begin{aligned} |\psi_1\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\phi'/2}|1\rangle\right), \\ |\psi_2\rangle &= \frac{1}{2}\left[\left(1 + e^{2\pi i\phi'/2}\right)|0\rangle + \left(1 - e^{2\pi i\phi'/2}\right)|1\rangle\right] \end{aligned} \tag{17.19}$$

We see that if $\phi' = 0$ the measurement of the upper qubit gives $|0\rangle$ and if $\phi' = 1$, the measurement of the upper qubit gives $|1\rangle$. Hence a measurement of the upper qubit in Fig. 17.6 determines the phase to one bit of precision.

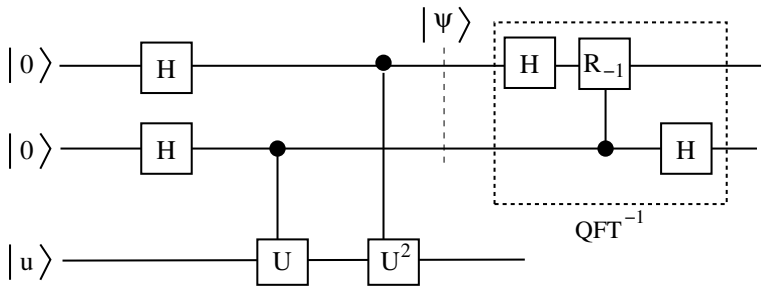


Figure 17.7: The circuit for phase estimation for two bits of precision. In their final state the two upper qubits contain the two bits of ϕ', which is related to the phase θ by Eq. (17.18).

We note that the right hand Hadamard on the upper qubit in Fig. 17.6 is just the QFT for 1 qubit, see Fig. 17.1. In fact, one can obtain ϕ' to an arbitrary accuracy of n-bits by using the n-bit QFT (strictly speaking the inverse QFT).

To see this we proceed gently by considering the circuit in Fig. 17.7 which is for two qubits. Both of the upper qubits are acted on by a Hadamard, after which one of them is the control for a control-U gate and the other is the control for a control-U^2 gate. The state $|\psi\rangle$ is given by

$$\begin{aligned}
|\psi\rangle &= \frac{1}{\sqrt{2}}\Big(|0\rangle + e^{2\pi i\phi'/2^2}|1\rangle\Big)\frac{1}{\sqrt{2}}\Big(|0\rangle + e^{4\pi i\phi'/2^2}|1\rangle\Big) \\
&= \frac{1}{2}\Big(|00\rangle + e^{2\pi i\phi'/2^2}|01\rangle + e^{4\pi i\phi'/2^2}|01\rangle + e^{6\pi i\phi'/2^2}|11\rangle\Big) \\
&= \frac{1}{2}\Big(|0\rangle_2 + e^{2\pi i\phi'/2^2}|1\rangle_2 + e^{4\pi i\phi'/2^2}|2\rangle_2 + e^{6\pi i\phi'/2^2}|3\rangle_2\Big) \\
&= \frac{1}{2}\sum_{k=0}^{3} e^{2\pi ik\phi'/2^2}|k\rangle_2. \qquad (17.20)
\end{aligned}$$

This is just the QFT of $|\phi'\rangle$ which can be undone by an inverse QFT, i.e.

$$|k\rangle_2 \to \frac{1}{2}\sum_{y=0}^{2} e^{-2\pi iyk/2^3}|y\rangle_2, \qquad (17.21)$$

since

$$\begin{aligned}
|\psi\rangle &\to \frac{1}{2}\sum_{k=0}^{2} e^{2\pi ik\phi'/2^2}\ \frac{1}{2}\sum_{y=0}^{2} e^{-2\pi iyk/2^2}|y\rangle_2 \\
&= \frac{1}{2^2}\sum_{y=0}^{2}\left[\sum_{k=0}^{3} e^{2\pi i(\phi'-y)k/2^2}\right]|y\rangle_2 \\
&= \frac{1}{2^2}\sum_{y=0}^{2} 2^2\,\delta_{y,\phi'}\,|y\rangle_2 \\
&= |\phi'\rangle_2. \qquad (17.22)
\end{aligned}$$

Hence the final measurement gives the 2-bit integer ϕ' from which the eigenvalue is given by $\lambda = e^{2\pi i\phi'/2^2}$.

This generalizes to the case of n bits of precision. We need n qubits to act as control-U, control-U^2, control-$U^4, \cdots,$ control-$U^{2^{n-1}}$ gates on the qubit containing $|u\rangle$. After the control-U^{2^l} gates have acted,

we run the qubits through the inverse Fourier transform to get $|\phi'\rangle_n$, from which $\theta = 2\pi\phi'/2^n$. The circuit is shown in Fig. 17.8.

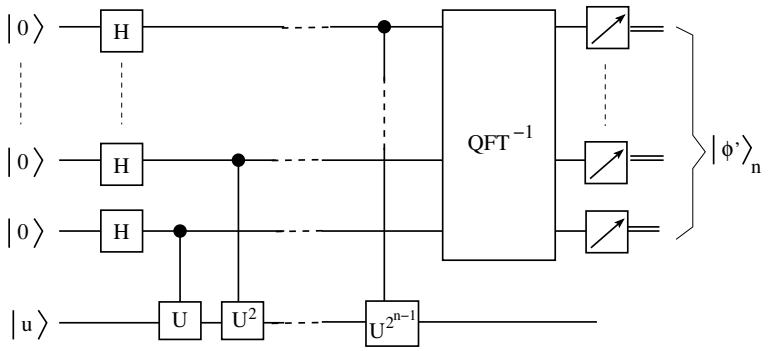


Figure 17.8: The circuit for phase estimation. The values of the n measured qubits form an integer ϕ', related to an eigenvalue $\lambda = e^{i\theta}$ of the unitary operator U by $\theta = 2\pi\phi'/2^n$.

What happens if $|u\rangle$ is not a single eigenstate of U as we have been assuming up to now, but a superposition? After the inverse QFT, the state of the n qubits will be a superposition of computational basis states $|\phi'\rangle$ for *each* of the eigenvalues present in the decomposition of $|\psi\rangle_n$ into its eigenstates. Measurement will then project on to the value of ϕ' of *one* of the eigenvalues.

Appendices

17.A Fast Fourier Transform (FFT) for $N = 4$

In this and the subsequent appendices in this chapter, we describe the connection between the QFT and FFT. This material is not necessary for the rest of the course and can be skipped.

For comparison with the QFT we write out the Fast Fourier Transform (FFT) for $N = 4$. The Fourier transform for this case is

$$y_0 = \tfrac{1}{2}\left(x_0 + x_1 + x_2 + x_3\right), \tag{17.23a}$$
$$y_1 = \tfrac{1}{2}\left(x_0 + ix_1 + i^2x_2 + i^3x_3\right), \tag{17.23b}$$
$$y_2 = \tfrac{1}{2}\left(x_0 + i^2x_1 + x_2 + i^2x_3\right), \tag{17.23c}$$
$$y_3 = \tfrac{1}{2}\left(x_0 + i^3x_1 + i^2x_2 + ix_3\right), \tag{17.23d}$$

where the x_j are the original data, the y_j are the Fourier transformed data, and we have used that

$$\exp(2\pi i/4) = i\,. \tag{17.24}$$

To evaluate Eqs. (17.23) efficiently, the FFT proceeds recursively. We firstly define Fourier transforms of length 2:

$$u_0 = \tfrac{1}{\sqrt{2}}(x_0 + x_2) \qquad = \tfrac{1}{\sqrt{2}}(x_0 + i^{2k}x_2)\ (k = 0)\,, \tag{17.25a}$$
$$u_1 = \tfrac{1}{\sqrt{2}}(x_1 + x_3) \qquad = \tfrac{1}{\sqrt{2}}(x_1 + i^{2k}x_3)\ (k = 0)\,, \tag{17.25b}$$
$$u_2 = \tfrac{1}{\sqrt{2}}(x_0 - x_2) \qquad = \tfrac{1}{\sqrt{2}}(x_0 + i^{2k}x_2)\ (k = 1)\,, \tag{17.25c}$$
$$u_3 = \tfrac{1}{\sqrt{2}}(x_1 - x_3) \qquad = \tfrac{1}{\sqrt{2}}(x_1 + i^{2k}x_3)\ (k = 1)\,, \tag{17.25d}$$
$$\tag{17.25e}$$

Pairs of quantities in Eqs. (17.25) are combined to form the Fourier Transform in Eqs. (17.23):

$$y_0 = \tfrac{1}{\sqrt{2}}(u_0 + u_1) \qquad = \tfrac{1}{\sqrt{2}}(u_0 + i^k u_1)\ (k=0)\,, \tag{17.26a}$$
$$y_1 = \tfrac{1}{\sqrt{2}}(u_2 + i\,u_3) \qquad = \tfrac{1}{\sqrt{2}}(u_2 + i^k u_3)\ (k=1)\,, \tag{17.26b}$$
$$y_2 = \tfrac{1}{\sqrt{2}}(u_0 - u_1) \qquad = \tfrac{1}{\sqrt{2}}(u_0 + i^k u_1)\ (k=2)\,, \tag{17.26c}$$
$$y_3 = \tfrac{1}{\sqrt{2}}(u_2 - iu_3) \qquad = \tfrac{1}{\sqrt{2}}(u_2 + i^k u_3)\ (k=3)\,, \tag{17.26d}$$
$$\tag{17.26e}$$

Let's check that this works by evaluating y_1. We have

$$y_1 = \tfrac{1}{\sqrt{2}}\,(u_2 + i\,u_3)\ , \tag{17.27a}$$
$$= \tfrac{1}{2}\,(\,x_0 - x_2 + i(x_1 - x_3)\,) = \tfrac{1}{2}\,\left(\,u_0 + ix_1 + i^2 x_2 + i^3 x_3\right)\ , \tag{17.27b}$$
$$\tag{17.27c}$$

which agrees with Eq. (17.23b).

It is instructive to write the linear transformations in Eqs. (17.23), (17.25), and (17.26) in matrix form. Equation (17.23) is written in matrix formulation as

$$\vec{y} = U\vec{x}\,, \tag{17.28}$$

where

$$U = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & 1 & i^2 \\ 1 & i^3 & i^2 & i \end{pmatrix}\ . \tag{17.29}$$

Equation (17.25) in matrix form is

$$\vec{u} = U_1\vec{x}\,, \tag{17.30}$$

where

$$U_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & i^2 & 0 \\ 0 & 1 & 0 & i^2 \end{pmatrix}\ . \tag{17.31}$$

Equation (17.26) in matrix form is

$$\vec{y} = U_2\vec{u}\,, \tag{17.32}$$

where

$$U_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & i \\ 1 & i^2 & 0 & 0 \\ 0 & 0 & 1 & i^3 \end{pmatrix}\ . \tag{17.33}$$

With some matrix manipulations one can verify that

$$U = U_2\,U_1\,, \tag{17.34}$$

as required. (I used *Mathematica*.)

17.B Comparison between FFT and the QFT for $N = 4$

The QFT is generated by the unitary matrix U in Eq. (17.29). The classical FFT is written as the product of two sparse matrices, $U = U_2U_1$, see Eq. (17.34), where U_1 is given in Eq. (17.31) and U_2 is given in Eq. (17.33). We will now see that there is a close connection between the FFT and the QFT, in particular, that the transformations U_1 and U_2 correspond to different parts of the diagram in Fig. 17.3.

The swap gate interchanges states $|01\rangle$ and $|10\rangle$, so it has the matrix representation

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} . \tag{17.35}$$

The Hadamard gate acting on the lower qubit of Fig. 17.3 was shown in Eq. (11.10). Including now also the (unchanged) upper qubit, the matrix representation of the transformation induced by this gate is

$$H_l = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} . \tag{17.36}$$

The Hadamard on the upper qubit has a similar representation, except that the two qubits have been interchanged, i.e.

$$H_u = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} . \tag{17.37}$$

The controlled R_1 phase gate gives a multiplicative factor of i if y_0 and x_0 are both 1, i.e. state $|3\rangle$. Hence

$$R_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix} . \tag{17.38}$$

The total effect of the quantum circuit in Fig. 17.3, reading from left to right on the circuit, is given by the matrix product $SH_lR_1H_u$. Note that one reads from from right to left in a product of operators because the operators act on the right. It can be confusing that the direction of time in the circuit diagram is opposite to that in an expression of operators. Multiplying out these matrices using *Mathematica* one gets the expected result,

$$SH_lR_1H_u = U\,, \tag{17.39}$$

where U is the Fourier transform, shown in Eq. (17.29). Recall that S is the swap, H_l is the Hadamard on the lower qubit, R_1 is the controlled phase gate, and H_u is the Hadamard on the upper qubit. Hence the gates in the quantum circuit in Fig. 17.3 do indeed affect a Fourier transform for 2 qubits.

In the FFT we decomposed U into a product of two sparse matrices, $U = U_2U_1$, see Eq. (17.34). We can also make a connection between the individual matrices U_2 and U_1 of the FFT and the individual matrices S, H_l, H_u and R_1 of the QFT. One finds

$$U_1 = H_u\,, \tag{17.40a}$$

$$U_2 = SH_lR_1\,. \tag{17.40b}$$

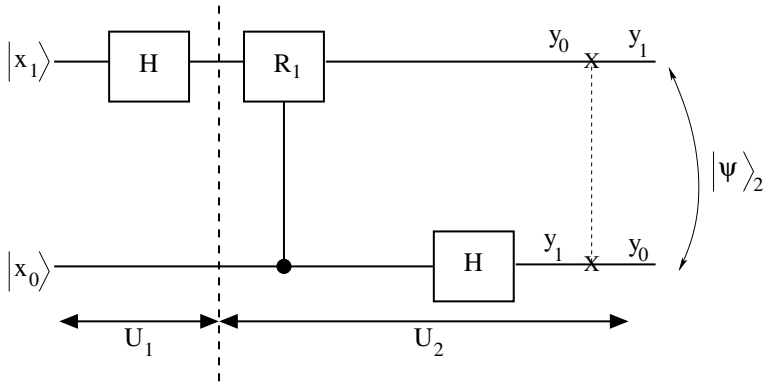


Figure 17.9: The same as Fig. 17.3 but also showing the correspondence with the breakup of the FFT into two operations $U = U_2 U_1$, see Eqs. (17.40).

The first is obtained by inspection and the second I checked with *Mathematica*. Hence the first operation U_1 in the FFT for $N = 4$ corresponds, in the QFT, to the Hadamard on the upper qubit in Fig. 17.3, while the second operation U_2 in the FFT corresponds to the remaining operations in the QFT: the controlled phase gate on the upper qubit, the Hadamard on the lower qubit, and the swap. This breakup is shown in Fig. 17.9.

To conclude this section, we have seen that there is close connection between the breakup used in the FFT and that used in the QFT. This should not be a surprise. In the FFT we iteratively divide the FT into two FTs of half the length, while in the QFT we have a binary representation of the states and treat each bit in turn, so clearly these are related. For $N = 4$, this connection is expressed in Eqs. (17.40).

17.C Comparison of FFT and QFT for $N = 8$

In this appendix we show how the breakup of the FFT for $N = 8$ is related to the circuit for the QFT. Our final result will be Fig. 17.10, which is the analog of Fig. 17.9 for $N = 4$.

As shown in Chapter 16, the FFT for $N = 8$ can be written as

$$U^{(8)} = U_3^{(8)} U_2^{(8)} U_1^{(8)} \tag{17.41}$$

where

$$U^{(8)} = \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega \end{pmatrix}, \tag{17.42}$$

$$U_1^{(8)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & \omega^4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \omega^4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \omega^4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \omega^4 \end{pmatrix}, \tag{17.43}$$

$$U_2^{(8)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \omega^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^2 \\ 1 & 0 & \omega^4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \omega^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \omega^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^6 \end{pmatrix}, \tag{17.44}$$

and

$$U_3^{(8)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \omega & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \omega^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega^3 \\ 1 & \omega^4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \omega^5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \omega^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega^7 \end{pmatrix}. \tag{17.45}$$

One can verify by doing the matrix multiplication (using *Mathematica* helps) that Eq. (17.41) is satisfied.

One can see from Fig. 17.4 that the QFT can be written as[2]

$$U^{(8)} = S_{02}^{(8)}\, H_l^{(8)}\, R_{1,m}^{(8)}\, H_m^{(8)}\, R_{2,u}^{(8)}\, R_{1,u}^{(8)}\, H_u^{(8)}\,, \tag{17.46}$$

in a fairly obvious notation, where

$$S_{02}^{(8)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{17.47}$$

[2]Recall that we work from right to left in operator equations like Eq. (17.46) but from left to right in circuit diagrams such as Fig. 17.4.

$$H_l^{(8)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}, \tag{17.48}$$

$$R_{1,m}^{(8)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & i \end{pmatrix}, \tag{17.49}$$

$$H_m^{(8)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{pmatrix}, \tag{17.50}$$

$$R_{2,u}^{(8)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega \end{pmatrix}, \tag{17.51}$$

$$R_{1,u}^{(8)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & i \end{pmatrix}, \tag{17.52}$$

$$H_u^{(8)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \end{pmatrix}, \tag{17.53}$$

One may verify Eq. (17.46) using *Mathematica*. Note that S_{02} swaps qubits 0 and 2, as required to reverse the order of the qubits.

Can we make a connection between the individual matrices, $U_1^{(8)}, U_2^{(8)}$, and $U_3^{(8)}$, in the FFT, Eq. (17.41), and the individual matrices, $S_{02}^{(8)}, H_l^{(8)}, R_{1,m}^{(8)}, H_m^{(8)}, R_{2,u}^{(8)}, R_{1,u}^{(8)}$, and $H_u^{(8)}$, in the QFT, Eq. (17.46)?

One immediately sees that $U_1^{(8)} = H_u^{(8)}$. However to make a connection between the other parts of the FFT, $U_2^{(8)}$ and $U_3^{(8)}$, we introduce the swap operator between qubits 1 and 2:

$$S_{12}^{(8)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{17.54}$$

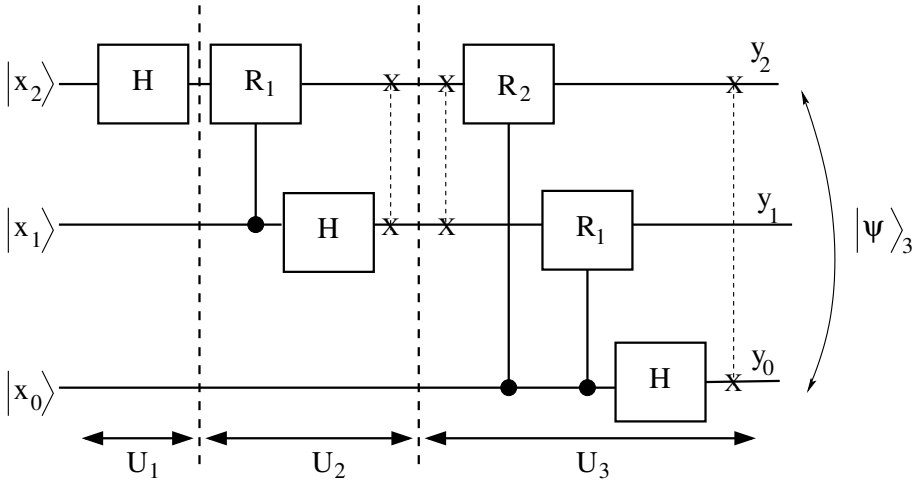


Figure 17.10: Like Fig. 17.4 except that the R_2 gate has been moved to the right of the Hadamard on the middle qubit (which has no effect) and that a pair of reversals of the order of qubits 1 and 2 have been added (which also has no effect). The reversal is accomplished by a swap gate. Note that the final reversal of the order of all three qubits (on the right of the diagram) is also accomplished by a single swap gate. The correspondence with the breakup of the FFT ($U = U_3 U_2 U_1$) is indicated, see Eqs. (17.56). To see this correspondence it is necessary to include the pair of reversals of the order of qubits 1 and 2.

We also need to realize that we can move the R_2 gate in Fig. 17.4 to the right as long as it does not cross the Hadamard on the lowest qubit (since this is the control qubit). Hence we can also write Eq. (17.46) as

$$U^{(8)} = S_{02}^{(8)}\, H_l^{(8)}\, R_{1,m}^{(8)}\, R_{2,u}^{(8)}\, H_m^{(8)}\, R_{1,u}^{(8)}\, H_u^{(8)}\,, \tag{17.55}$$

where we have moved $R_{2,u}^{(8)}$ to the *left*. We then find that

$$U_1^{(8)} = H_u^{(8)}\,, \tag{17.56a}$$

$$U_2^{(8)} = S_{12}^{(8)}\, H_m^{(8)}\, R_{1,u}^{(8)}, \tag{17.56b}$$

$$U_3^{(8)} = S_{02}^{(8)}\, H_l^{(8)}\, R_{1,m}^{(8)}\, R_{2,u}^{(8)}\, S_{12}^{(8)}\,, \tag{17.56c}$$

which agrees with Eqs. (17.55) and (17.41) since $\left(S_{12}^{(8)}\right)^2$ is the identity (swapping twice makes no change). This breakup is shown in Fig. 17.10. Apart from the reversals of qubit order, the correspondence between the QFT and the FFT is straightforward to see.

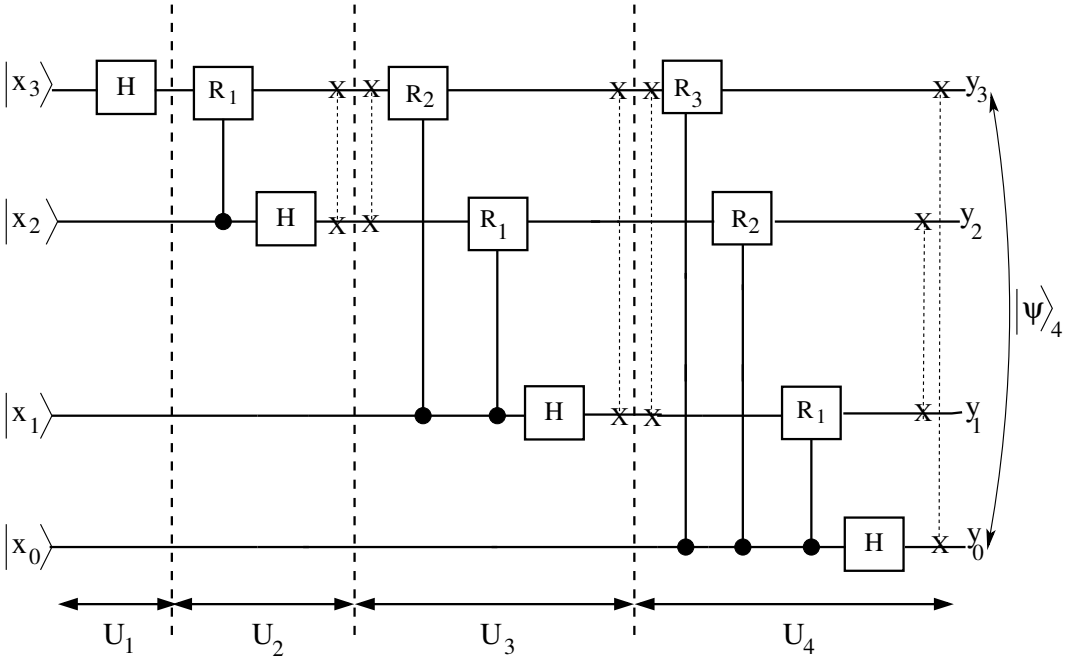


Figure 17.11: The generalization of Figs. 17.10 and 17.9 to the case of four qubits. The correspondence with the breakup of the FFT ($U = U_4U_3U_2U_1$) is indicated.

Following the structure of Fig. 17.9 for two qubits, and Fig. 17.10 for three qubits the generalization to four qubits is shown in Fig. 17.11. The correspondence with the FFT is clear, the only complication being that, in order to show the correspondence, pairs of reversals of the order of the qubits (which cancel each other out) have to be introduced, with one reversal being in one stage of the FFT and the other reversal in the next stage of the FFT. Reading Fig. 17.11 from left to right, the first reversal pair reverses qubits 2 and 3 (which needs a single swap gate between qubits 2 and 3), the next reversal reverses qubits 1, 2 and 3 (which only needs a single swap gate between qubits 1 and 3), and the last reversal (not a pair because this is the last one so there is no additional stage to compensate it) reverses all 4 qubits (which needs two swap gates, one between qubits 0 and 3 and the other between qubits 1 and 2).

We see that there is a close parallel between the breakup of the FFT and circuit of the QFT. The details are slightly complicated because one needs reversals of the order of the qubits to make the correspondence precise. Note that Fig. 1 in `https://arxiv.org/pdf/1005.3730.pdf` is related to the results presented here.

Chapter 18

Shor's Algorithm

When computers we build become quantum,
Then spies of all factions will want 'em.
Our codes will all fail,
They'll hack our email,
But crypto that's quantum will daunt 'em.

This is a slightly modified version of a limerick by Peter and Jennifer Shor. (The original version is printed in the book by Nielsen and Chuang [NC00].) Continuing in a literary vein, on p. 453 of Nielsen and Chuang is a very well-crafted (Shakespearean) sonnet by Daniel Gottesman on quantum error correction. It seems that quantum computing brings out latent literary qualities in some scientists who work on it (unfortunately not me!).

18.1 Introduction

Consider an integer N composed of two prime factors p and q, i.e. $N = p\,q$. In Chapter 15 we showed how to determine the factors of N from the period r of the function

$$f(x) \equiv a^x \,(\mod N)\,, \tag{18.1}$$

where a is some number less than N and which has no factors in common with N. Since $a^0 = 1$, the period is the smallest value $x = r$ such that

$$a^r \,(\mod N) = 1. \tag{18.2}$$

In 1994 Peter Shor [Sho94] developed a famous quantum algorithm for period finding which is much more efficient for factoring large integers than any known algorithm running on a classical computer. The ability to factor a large integer can be used to decode messages sent down a public channel (such as the internet) which have been encrypted with the RSA scheme. The first four lines of the above limerick refer to this[1]. The RSA encryption scheme is described in Chapter 14.

Here we describe in detail Shor's algorithm to determine the period of the function $f(x)$ in Eq. (18.1). Useful references are [Mer07, NC00, Vat16].

We denote by n_0 the number of bits needed to contain N, so N is comparable to 2^{n_0}. In cryptography, N may have of order 600 digits (so $n_0 \sim 2000$ bits).

18.2 Modular Exponentiation

In Shor's algorithm the period is found by a Quantum Fourier transform of the function in Eq. (18.1) evaluated for $x = 0, 1, 2, \cdots, 2^n - 1$. What do we take for n? Now the period may be comparable to

[1] The last line of the limerick refers to quantum key distribution (QKD) which will be discussed in Chapter 21.

N and, according to Mermin [Mer07], in general we need at least N periods in the data, i.e. $2^n > N^2$, and so set $n = 2n_0$. We will see why the doubling of the number of qubits is necessary in Sec. 18.5. Hence, if $n_0 \sim 2000$ we have $n \sim 4000$.

It would seem to be a formidable (nay, impossible) task to calculate $a^x \pmod N$ for a value of x of order 2^{4000}. However, it can be done as follows. First compute $a, a^2, a^4, \cdots a^{2^n} \pmod N$ by successively squaring. This only takes n multiplications and so can be done on a classical or quantum computer. Let the binary expansion of x be

$$x = x_{n-1}x_{n-2}\cdots x_2x_1x_0. \tag{18.3}$$

Then we have

$$a^x = \prod_{j=0}^{n-1} \left(a^{2^j}\right)^{x_j} . \tag{18.4}$$

For example for $n = 4$, $x = 10$, the binary expansion of x is 1010 (note the least significant bit is to the right) so

$$a^{10} = \left(a^8\right)^1 \left(a^4\right)^0 \left(a^2\right)^1 \left(a^1\right)^0 . \tag{18.5}$$

The use of Eq. (18.4) to compute a^x for huge values of x is called "*modular exponentiation*".

On a quantum computer each digit of x is represented by one qubit, and, as we shall see, Eq. (18.4) can be computed efficiently for *all* x between 1 and $2^n - 1$ on a quantum computer using quantum parallelism.

A schematic circuit diagram for doing modular exponentiation is shown in Fig. 18.1. There are n upper or "input" qubits, which contain the values of x, and n_0 lower or "output" qubits, which contain the function values $f(x)$. As discussed above, we usually take $n = 2n_0$. We will call the *set* of input qubits the "input register", and similarly denote the output qubits as the "output register". The notation "input" and "output", though often used, can be rather confusing since both input and output registers are present in the initial state (left edge of the circuit diagram in Fig. 18.1) and in the final state (right edge of the circuit).

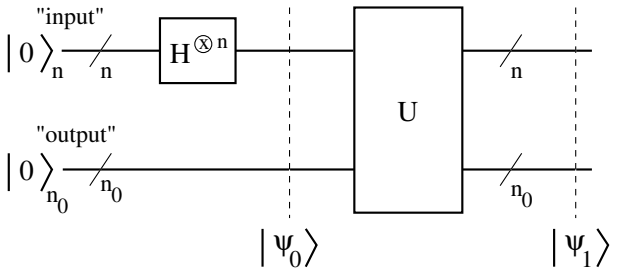


Figure 18.1: Schematic circuit diagram for performing the modular exponentiation. The workings of the black box U are described in the text. The state entering U is $|\psi_0\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n |0\rangle_{n_0}$ and the state exiting from U is $|\psi_1\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n |f(x)\rangle_{n_0}$.

Both the input and output qubits are initialized to $|0\rangle$. The input qubits are each run through a Hadamard gate. As shown in Sec. 10.2, Hadamards acting on n qubits gives the symmetric sum of all 2^n basis states. Hence before entering into the box U shown in Fig. 18.1, the state of the system is

$$|\psi_0\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n |0\rangle_{n_0} . \tag{18.6}$$

On exiting the box U, the state of the system has the values of $f(x)$ in the output register, i.e.

$$|\psi_1\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n \, |f(x)\rangle_{n_0} \, . \tag{18.7}$$

Note that, in general, if the lower (output) qubits were initialized to $|y\rangle$, then after the function acted they would be in state $|y \oplus f(x)\rangle$, but here $y = 0$.

How does the function-evaluating box U work? We start with a certain value for $x \equiv x_{n-1}x_{n-2}\cdots x_2x_1x_0$ in the input register and $1 \equiv 000\cdots001$ in the output register. We also need an additional work register with n_0 qubits, whose contents we will denote by w, with initial value $w = a$. The following steps compute $a^x \pmod N$ using Eq. (18.4):

- (a) Multiply the output register by w if $x_0 = 1$.
- (b) Replace w by its square $w \to w^2$.
- (a') Repeat (a) but for x_1.
- (b') Repeat (b)
- Continue repeating (a) (with successive bits of x) and (b).

Since all possible values of x occur in the linear superposition in $|\psi_0\rangle$ in Eq. (18.6), (which is inputted to U) linearity of the operations in U ensures that the state outputted by U is the linear superposition in Eq. (18.7), with $f(x)$ computed for *all* x.

How many operations does this require? If we consider (b) we need to do n squares of an n_0-bit number. Multiplying two n_0 bit numbers in the simplest way[2] takes $O(n_0^2)$ operations. Since $n = 2n_0$ we see that the operation count for (b) is $O(n^3)$. The operation count for (a) is similar, so the total operation count for modular exponentiation is $O(n^3)$.

18.3 Quantum Fourier Transform (QFT)

A schematic of the full circuit for period finding is shown in Fig. 18.2.

The first (left) part of the algorithm is the modular exponentiation also shown in Fig. 18.1. A measurement is then made of the result in the output (lower) register from the modular exponentiation routine U. This measurement is indicated by the lower box with an arrow in Fig. 18.2. The measurement will yield some value for $f(x)$, say f_0. According to the extended Born hypothesis, the input (upper) register will then contain a superposition of those basis states for which $f(x) = f_0$. Since $f(x)$ is periodic with period r, the possible values of x are of the form $x_0 + kr$, so, after the measurement on the output register, the state of the input register becomes

$$|\psi_2\rangle = \frac{1}{\sqrt{Q}} \sum_{k=0}^{Q-1} |x_0 + kr\rangle_n \, . \tag{18.8}$$

Here $0 \le x_0 \le r-1$, $x_0 + (Q-1)r \le 2^n - 1$, $f(x_0 + kr) = f_0$, and the number states in the sum is

$$Q = \left[\frac{2^n}{r}\right] \, , \tag{18.9}$$

where $[\cdots]$ denotes the integer part. Thus $P_x(x)$, the probability of of measuring state $|x\rangle$ in the input register, consists of Q delta functions at positions $x_0 + kr, k = 0, 1, \cdots, Q-1$, see Fig. 18.3.

[2] As mentioned in Nielsen and Chuang [NC00], there are more sophisticated methods of multiplying n-bit numbers which only take $n\,(\ln n)\,(\ln\ln n)$ operations rather than n^2. This gives a total operation count for modular exponentiation of $O(n^2 \ln n \, \ln\ln n)$, hardly more than $O(n^2)$.

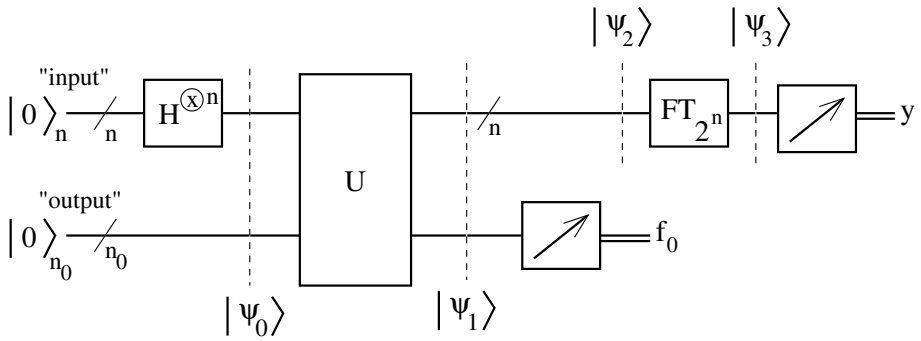


Figure 18.2: Schematic circuit diagram for Shor's algorithm for period finding on a quantum computer. The black box U does the modular exponentiation as described in the text, see also Fig. 18.1. The state inputted to U is given by $|\psi_0\rangle$ in Eq. (18.6) and the state outputted from U is given by $|\psi_1\rangle$ in Eq. (18.7). A measurement (indicated by the box with the arrow) is performed on the output (lower) register, giving some value f_0. The double lines indicate that the measurement gives classical bits which take values 0 *or* 1. The state of the input (upper) register is then given by $|\psi_2\rangle$ in Eq. (18.8), the equally weighted superposition of all values of x for which $f(x) = f_0$. The n input qubits then go through the quantum Fourier transform the result of which is given by $|\psi_3\rangle$ in Eq. (18.10). A measurement of the input qubits then gives a result y which is close to an integer multiple of $2^n/r$, where r is the period, as discussed in the text.

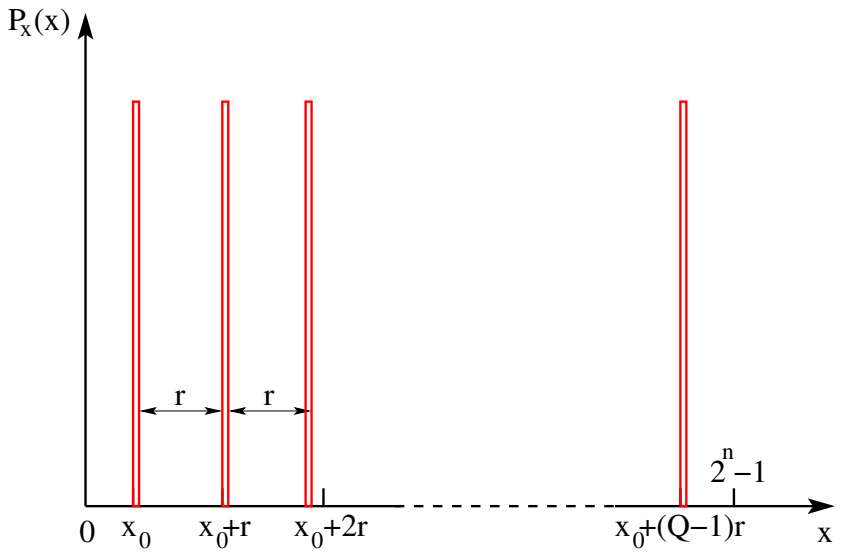


Figure 18.3: The probability of getting state x in the input register if a measurement were performed *before* doing the Quantum Fourier Transform. There are Q delta functions, where $Q = [2^n/r]$, each with weight $1/Q$ separated by r, the period. The values of x where these delta functions appear, $x_0 + kr, k = 0, 1, \cdots, Q-1$, are those values for which $f(x) = f_0$ the result obtained from the measurement of the output register. A measurement would get a value for $x_0 + kr$ for some k but since we don't know x_0 this is no help in determining the period r. Hence measuring the input register at this point is not useful. We need to Fourier transform the state of the input register before measuring it, in order to determine the period.

If we were to measure $|\psi_2\rangle$ we would just get one value of x_0+kr, which, because of the dependence on the unknown quantity x_0, does not give any information from which we might be able to determine the period r. In order to extract information on r, we have perform a quantum Fourier transform on the states in Eq. (18.8) before measuring. This gives

$$|\psi_3\rangle = \sum_{y=0}^{2^n-1} \left(\frac{1}{\sqrt{2^n Q}} \sum_{k=0}^{Q-1} e^{2\pi i(x_0+kr)y/2^n} |y\rangle_n \right) . \tag{18.10}$$

The quantum circuit which performs the quantum Fourier transform is described in Chapter 17. An example for $n = 4$ qubits is shown in Fig. 18.4. The controlled phase gates act on the target qubit according to Eq. (17.10) if the control qubit is 1, and otherwise do nothing. Like the controlled Z gate, the controlled phase gate is symmetric between the control and target qubits (the phase is changed only if both qubits are $|1\rangle$), so the control and target qubits can be exchanged. We will use this in Appendix 18.A when we see how to actually eliminate these 2-qubit gates.

Generalizing the diagram in Fig. 18.4 to the case of n qubits we see that controlled phase gates R_d are required for $d = 1, 2, \cdots, n-1$. Hence, in total, we need n Hadamard gates and $1+2+\cdots+n-1 = n(n-1)/2$ controlled phase gates. However, as discussed in Sec. 3.9 of Mermin [Mer07], and in Appendix 18.C, it is both impossible to contruct gates giving a phase change which is exponentially small in n, and also not necessary to do this to obtain the QFT with the required precision. Mermin shows that one only needs controlled phase gates R_d for $d < \log_2(\text{const.}\, n)$, where the constant Mermin gives is large but independent of n. Thus the number of controlled phase gates needed *in practice* is of order $n \log_2 n$ which is considerably less than $O(n^2)$ if n is of order several thousand.

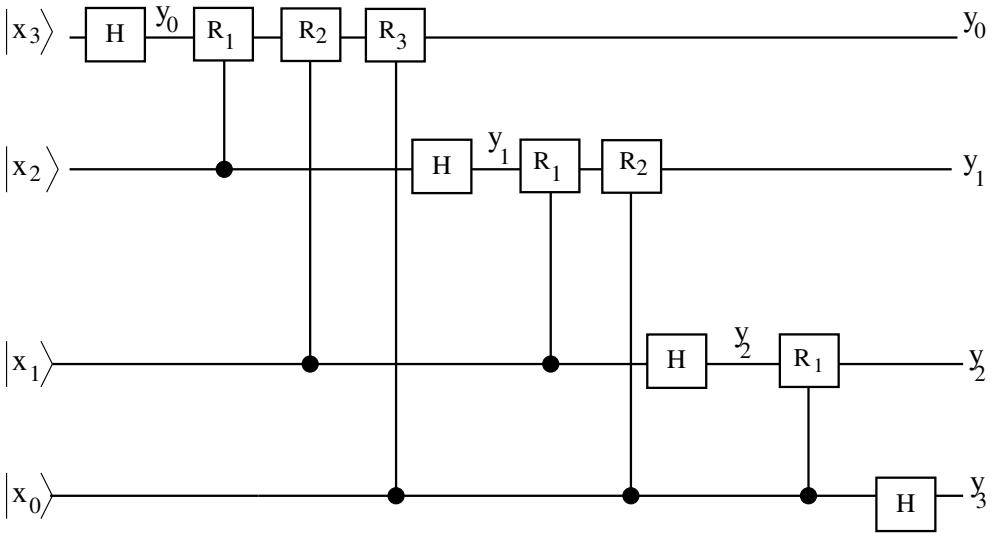


Figure 18.4: The circuit for the quantum Fourier transform for $n = 4$ qubits. The controlled phase gates act on the target qubit according to Eq. (17.10) if the control qubit is 1 and otherwise does nothing. The final swap gates to reverse the order of the qubits outputted on the right are not included here. Note that the controlled phase gate between qubits x_i and x_j is $R_{|i-j|}$ which makes the structure of the circuit quite simple to understand.

In fact we can eliminate the 2-qubit controlled phase gates by measuring each qubit immediately after the gates of the QFT have acted on it, rather than after completion of the QFT. This is discussed in Appendix 18.A.

After the quantum Fourier transform we measure the input register, see Fig. 18.2, obtaining a value for y. The probability of getting a particular state y is given by the square of the absolute value of the amplitude of $|y\rangle$ in Eq. (18.10), i.e.

$$P(y) = \frac{1}{2^n Q} \left| \sum_{k=0}^{Q-1} e^{2\pi i k r y/2^n} \right|^2 . \tag{18.11}$$

Note that the dependence on x_0, which was troublesome before doing the Fourier transform, and appears just as a phase factor after the Fourier transform, Eq. (18.10), now drops out completely when we take the square of the absolute value to get the probabilities in Eq. (18.11).

If y could take real values, the exponentials would add up precisely in phase (and so there would be a peak in the probability for y), when $yr/2^n$ is an integer, i.e. for $y = y_m$ where

$$y_m = m \frac{2^n}{r}, \tag{18.12}$$

in which m is an integer. Note that there are r values of m, from 0 to $r-1$ since y runs over a range of 2^n values. We emphasize that y_m is not an integer in general, but the measured values of y *are* are integers, so there will be peaks in $P(y)$ at integer values *close to* the y_m in Eq. (18.12), see the sketch in Fig. 18.5. Precise values of $P(y)$ for a particular case will be calculated in Sec. 18.5. Hence there is a high probability that we will obtain an integer y close to an integral multiple of $2^n/r$.

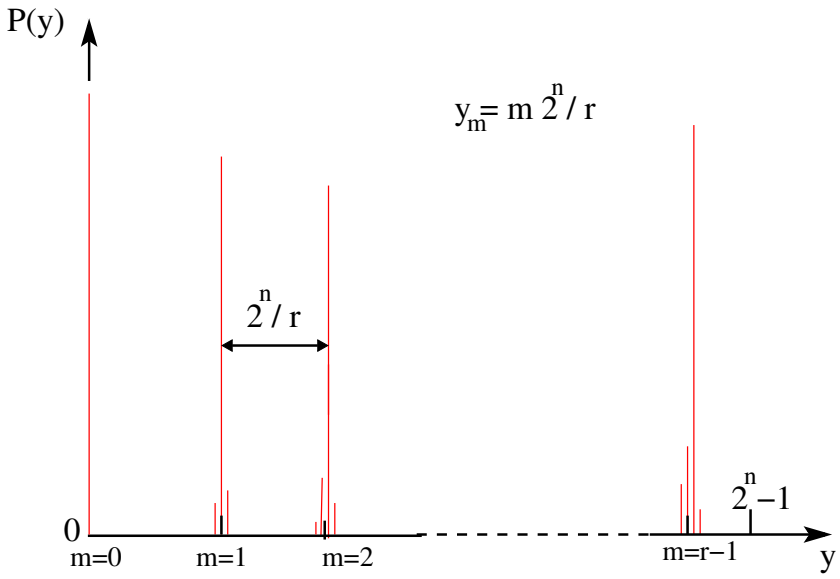


Figure 18.5: A sketch of the probability of getting state y in the input register *after* the Quantum Fourier Transform. There are r peaks at $y_m = m\,2^n/r$ for $m = 0, 1, 2, \cdots, r-1$. Note that $[2^n/r] = Q$ so the *separation* between the peaks in $P(y)$ is no more than 1 away from Q, the *number* of peaks in the distribution $P_x(x)$ for the state *before* the quantum Fourier transform, see Fig. 18.3. Precise values of $P(y)$ will be calculated in Sec. 18.5 for a particular case and the resulting values of $P(y)$ will be shown in Fig. 18.8.

To summarize this part, $P(y)$ has r peaks separated by $2^n/r$. We recall that r is the period, which is what we want to compute.

18.4 A special case: the period is a power of 2.

In some special cases the period r will be a power of 2, in which case an integer number of periods fits *exactly* into the range of x-values (2^n). An example discussed by Mermin [Mer07] is if both p and q are both primes of the form $2^\ell + 1$ (e.g. the commonly studied case of $N = 15$). In this situation we will not need n to be as big as $2n_0$ (where n_0 is the number of bits needed to contain N). Rather, we will see that we just need 2^n to be big enough to contain some integer number[3] of periods for us to exactly determine an integer multiple of $2^n/r$. Since the period might be as large as N, when r is a power of 2 we need

$$\begin{aligned} 2^n &= \text{const. } 2^{n_0} \quad \text{rather than} \\ 2^n &= 2^{2n_0} \quad \text{in the general case.} \end{aligned} \tag{18.13}$$

Here we go through this special case because the mathematics is simpler than the generic case which we will study in the next section.

First of all we check for $N = 15$ that the period *is* a power of 2 as stated above. Let's take $a = 7$ which has no factors in common with 15:

$$x = 1, \quad a^x = 7\,, \tag{18.14a}$$
$$x = 2, \quad a^x = 7 \times 7 = 49 \equiv 4 \ (\text{ mod } 15)\,, \tag{18.14b}$$
$$x = 3, \quad a^x = 7 \times 4 = 28 \equiv 13 \ (\text{ mod } 15)\,, \tag{18.14c}$$
$$x = 4, \quad a^x = 7 \times 13 = 91 \equiv 1 \ (\text{ mod } 15)\,, \tag{18.14d}$$

so the period is $r = 4$, i.e. a power of 2 as claimed.

Now, we perform the sum in Eq. (18.11). Since r is a power of 2 here, and $2^n \geq r$, it follows that $2^n/r$ is an integer, so Q, the number of terms in the sum in Eq. (18.11), is given *exactly* by

$$Q = \frac{2^n}{r}\,. \tag{18.15}$$

From Eq. (18.15), we see that Eq. (18.11) becomes

$$P(y) = \frac{1}{r}\left|\frac{1}{Q}\sum_{k=0}^{Q-1} e^{2\pi i k y/Q}\right|^2 . \tag{18.16}$$

Firstly suppose that $y = mQ$ for integer m. It is trivial to see that all the exponentials in Eq. (18.16) are unity so

$$P(y = mQ) = \frac{1}{r}. \tag{18.17}$$

Note that there are r distinct values of m, $m = 0, 1, 2, \cdots, r-1$ since y runs over a range of 2^n values and $Q = 2^n/r$, see Eq. (18.15). Hence the sum of the probabilities for the set of values $y = mQ$ is unity. Since the *total* probability must be unity there can be no probability for other values of y, as we will now verify.

The sum in Eq. (18.16) is a geometric series, which can be summed to give

$$\sum_{k=0}^{Q-1} e^{2\pi i k y/Q} = \frac{1 - e^{2\pi i y}}{1 - e^{2\pi i y/Q}}\,. \tag{18.18}$$

[3] Even one period is sufficient, i.e. $2^n = r$.

The numerator is zero for all y (recall that y is an integer), but for $y \neq mQ$ the denominator is non-zero, so

$$P(y \neq mQ) = 0\,, \tag{18.19}$$

as required. Thus, with probability 1, the measured value of y is an integer multiple of $2^n/r$. This is shown in Fig. 18.6. Superficially, this may look similar to the situation before the QFT shown in Fig. 18.3. The difference is that the unknown quantity x_0 does not appear in Fig. 18.6. Rather, the delta functions occur at positions y_m where $y_m/2^n = m/r$ from which one can determine r.

Notice the reciprocal relation between the period r in the original data in Fig. 18.3 and the period in the Fourier transformed data which is the size of the dataset, 2^n, *divided* by r. To use terminology from sound waves and frequencies, quite generally, if the original dataset is a periodic function of "time" with period r, the Fourier transform will have a peak at the "fundamental frequency", $2^n/r$, and in addition can have peaks at "higher harmonics" ($m\,2^n/r$ for $m > 1$). It can also have a component at zero "frequency" ($y = 0$) if the average of the original data is non-zero. The special nature of the original dataset here (uniformly spaced delta functions, see Fig. 18.3) leads to the Fourier transform having *equal* weight in all non-zero Fourier components.

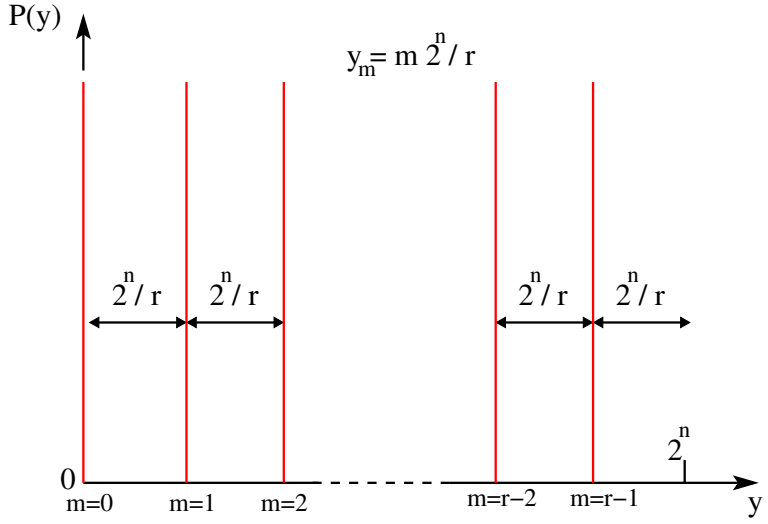


Figure 18.6: The probability of getting state y in the input register after the Quantum Fourier Transform for the special case where r is a power of 2 so there are an exact number of periods in the interval 2^n. There are r delta functions of equal weight at exactly $y_m = m\,2^n/r$, for $m = 0, 1, \cdots, Q-1$.

Let us give a simple example so we can see in detail how to extract the period r from this knowledge. We take our previous example of $N = 15, a = 7$, for which we found in Eq. (18.14) that the period is $r = 4$. This means that $7^4 \equiv 1 \mod 15$. We will assume that we have $n = 5$ qubits, so $2^n = 32$. The only possible results of a measurement of y are an integer multiple of $Q = 2^n/r\,(= 8)$, so here we have $y = 0, 8, 16$ or 24, each with equal probability 1/4, see Table 18.1.

From the measurement of y we determine the numerator and denominator of the fraction $y/2^n (= m/r)$ for some m, but with any common factor c (shown in the last column of Table 18.1) divided out. We therefore write $y/2^n$ as m_0/r_0 with $m = cm_0, r = cr_0$. To determine $r = cr_0$, where c is generally a small integer, we compute the function $a^{cr_0} \mod N$ for the first few values of $c = 1, 2, \cdots$ and see for what value of c we obtain 1, the result if $cr_0 = r$, see Eq. (18.2). The common ratio c is unlikely to be large. For example if m is odd, which occurs with probability 1/2, then c must equal 1. Similarly there is probability 1/4 that m is even but not a multiple of 4 in which case c cannot be greater than 2.

y	m	$\frac{y}{2^n}\left(=\frac{m_0}{r_0}\right)$	$c=\frac{r}{r_0}$
0	0	0	–
8	1	1/4	1
16	2	1/2	2
24	3	3/4	1

Table 18.1: The possible results of a measurement of y for the case of $N = 15, a = 7, 2^n = 32$ for which $r = 4$. A measurement gives both the numerator and denominator of the fraction $y/2^n (= m/r)$ for some m, but with any common factor c, shown in the last column, divided out, so we write $y/2^n$ as m_0/r_0 with $m = cm_0, r = cr_0$. Hence we obtain r_0 (and m_0), but not c. We determine c by computing the function a^{cr_0} for $c = 1, 2, \cdots$ until we get the value 1. There is a probability 1/2 that $c = 1$ works, and it is extremely unlikely that a large value of c will be needed.

Proceeding in this vein we see that it is very unlikely that c is large. In the rare case that the common ratio c *is* large, we would stop after the first few values of c and restart the quantum computation (the steps shown in Fig. 18.2).

In Table 18.1 we see that the value $y = 0$ does not give useful information but, since the number of possible results is equal to r and each result is equally probable, the probability of getting $y = 0$ is small if the period r is large (the situation if one needs a quantum computer).

In this section, we have seen that in the rare situation that the period is a power of 2, the measurement of y gives an integer multiple of $2^n/r$ *with probability one.* Hence $y/2^n = m/r$ with integer m *exactly.* However, in the general case, which we discuss in the next section, the measurement of y will give, with a probability which is high but less than one, a value such that $y/2^n$ is *close to* (but not equal to) m/r. The continued fraction method in Appendix 18.B is then needed to determine m/r. For the continued fraction method to work it turns out that we need to have at least N periods in the range of values of x, and so we will take $n = 2n_0$.

18.5 The generic case: the period is not a power of 2.

We now evaluate the sum in Eq. (18.11) for the generic case when r is not a power of 2 so we do not have an exact integer number of periods in the range of x-values, 2^n, over which $f(x)$ is calculated. As discussed after Eq. (18.11), $P(y)$ has r peaks in the vicinity of y_m given by Eq. (18.12). We set

$$\begin{aligned} y &= y_m + \delta_m \,, \\ &= m\frac{2^n}{r} + \delta_m \,, \end{aligned} \tag{18.20}$$

We will assume that δ_m is small, so we are close to the m-th peak, but $2^n, r$ and m are large, since we only need the quantum algorithm when these numbers are large. (Recall that y, the measured value is an integer, whereas y_m and δ_m are not.)

Equation (18.11) involves a geometric series which can be summed as follows:

$$\begin{aligned}\sum_{k=0}^{Q-1} e^{2\pi i k r y/2^n} &= \sum_{k=0}^{Q-1} e^{2\pi i k m} e^{2\pi i k r \delta_m/2^n}\,, \\ &= \sum_{k=0}^{Q-1} e^{2\pi i k r \delta_m/2^n}\,, \\ &= \frac{1-e^{2\pi i Q r \delta_m/2^n}}{1-e^{2\pi i r \delta_m/2^n}}\,, \\ &= \frac{e^{\pi i Q r \delta_m/2^n}\sin\left(\pi Q r \delta_m/2^n\right)}{e^{\pi i r \delta_m/2^n}\sin\left(\pi r \delta_m/2^n\right)}\,. \end{aligned} \tag{18.21}$$

Inserting Eq. (18.21) into Eq. (18.11) the phase factors drop out and we get

$$P(y) = \frac{1}{2^n Q}\,\frac{\sin^2\left(\pi Q r \delta_m/2^n\right)}{\sin^2\left(\pi r \delta_m/2^n\right)}\,. \tag{18.22}$$

Now Q is within an integer of $2^n/r$ and Q is also large so so we can replace $Qr/2^n$ by 1 with negligible error. Also $r/2^n$ is very small, since we take n to be big enough that there are many periods within the range of x computed, so the sine in the denominator can be replaced by its argument. Hence we get, to a good approximation,

$$\boxed{P(y) = \frac{1}{r}\left(\frac{\sin \pi\delta_m}{\pi\delta_m}\right)^2\,,} \tag{18.23}$$

for y in the vicinity of y_m. Recall that the relation between δ_m and y is given in Eq. (18.20). The function in Eq. (18.23) is plotted in Fig. 18.7. The area under the curve is 1, and most of the weight is in the peak centered at 0.

To find the period we would like to get the integer y which is *closest* to $m\,2^n/r$ for some integer m i.e. $|\delta_m| < 1/2$. Writing $\pi\delta_m = x$, this corresponds to $|x| < \pi/2$, and in this region

$$\frac{\sin x}{x} > \frac{2}{\pi}\,, \tag{18.24}$$

so, according to Eq. (18.23), the probability of getting the nearest integer to y_m is greater than

$$\frac{1}{r}\,\frac{4}{\pi^2} \simeq \frac{0.40}{r}, \tag{18.25}$$

see Fig. 18.7. There are r distinct values[4] of m so the total probability of getting the closest integer to *one* of the y_m is greater than 40%.[5]

So, with fairly high probability, we have obtained the nearest integer to $m\,2^n/r$ for some integer m (which we don't know). How can we determine r from this information? We need some post-processing which will be done on a *classical* computer.

In deriving Eq. (18.23) we just needed that the range of x studied contains many periods, i.e. $2^n \gg r$. Since r can not be bigger than N we needed $2^n \gg N$. However, to actually extract r we need a stronger condition, $2^n > N^2$, as we shall now see.

[4] One of these is for $m = 0$ which doesn't give useful information but since we are interested in situations where r is large, the difference between r and $r-1$ is negligible.

[5] In fact, according to Mermin [Mer07], Appendix L, when N is the product of two primes (as we have here) the period is not only less than N but less than $N/2$. As a result, still using $n = 2n_0$ qubits in the input register, the algorithm will provide a result for r not only if the measured value of y is the closest integer to $m\,2^n/r$, but also if it is the second, third or fourth closest. This increases the probability of a successful run to about 0.9.

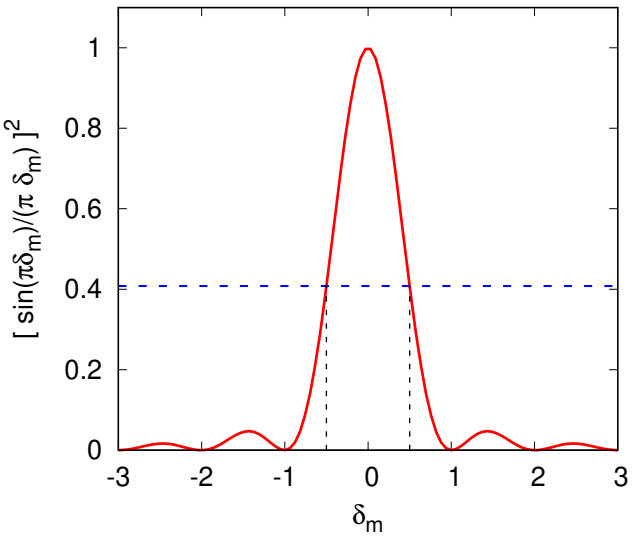


Figure 18.7: A plot of the function in Eq. (18.23), neglecting the factor of $1/r$ where r is the number of peaks. The area under the curve is 1. The result of a measurement will be one of a series uniformly spaced possible values of δ_m separated by 1. For example, if $y_m+0.3$ is an integer, the possible measured values of δ_m would be $\cdots, -1.7, -0.7, 0.3, 1.3, \cdots$. One of these values for δ_m must be within 1/2 of 0 and the figure shows that the probability for this is greater than $4/\pi^2$, the dashed horizontal line. (The dashed vertical lines are at $\delta_m = \pm 1/2$). An example of real data is shown in Fig. 18.9.

We assume now that we have been successful and found a y which is within 1/2 of $2^n m/r$. Dividing by 2^n we have

$$\left|\frac{y}{2^n} - \frac{m}{r}\right| < \frac{1}{2^{n+1}}\,, \tag{18.26}$$

so $y/2^n$, our estimate for m/r, is off by no more than $1/(2\,2^n)$.

The value of m/r can then be obtained using continued fractions. A continued fraction representation of a number x has the form

$$x = c_0 + \cfrac{1}{c_1 + \cfrac{1}{c_2 + \cfrac{1}{c_3 + \cdots}}}, \tag{18.27}$$

where the c_i are integers known as the continued fraction coefficients. If we stop after a certain number of iterations and ignore the remainder we have a "partial sum", which is an approximation for x. If x is a rational number (ratio of two integers) the continued fraction will eventually terminate. If x is irrational (like π) the continued fraction will go on for ever. More details about continued fractions are given in described in Appendix 18.B.

The crucial result of continued fractions which we need is theorem A4.16 in Appendix 4 of Ref. [NC00], which states that if

$$\left|\frac{y}{2^n} - \frac{m}{r}\right| < \frac{1}{2r^2} \tag{18.28}$$

then m/r is one of the partial sums in the continued fraction representation of $y/2^n$. Here $r < N \sim 2^{n_0} = 2^{n/2}$ so we see from Eq. (18.26) that the theorem applies[6]. Hence m/r will appear as one of the

[6]It is at this point that we need the data to contain at least N periods.

partial sums in the continued fraction representation of $y/2^n$. Since $r < N$ this must be a partial sum with denominator less than N. Successive partial sums get more and more accurate, so we want the one with the *largest denominator* less than[7] N.

As we already noted for the special case when r is a power of 2 (Sec. 18.4), if m and r have a common factor, c say, then the continued fraction representation will divide this out and give m_0/r_0 where $m_0 = m/c, r_0 = r/c$. Thus we actually get r_0 which is a divisor of r. However, we may be lucky and still get r straight away. As shown in Appendix J of Mermin [Mer07], the probability that two large numbers chosen at random have no common factors is greater than 1/2. Thus, with probability greater than 1/2, we get r directly. We can check if r_0 is the period r by computing, on a classical computer, $a^{r_0}(\mod N)$ and seeing if we get 1. If we do not, we would try simple multiples, $r = 2r_0, 3r_0, 4r_0, \cdots$, since it is very unlikely that the common factor is large. If we *are* very unlucky, and the common factor *is* large, we could start again from the beginning, get another value for m/r and hence get another value for r_0, and compute $a^{r_0}(\mod N)$. If this is not 1, then again we try $r = 2r_0, 3r_0, 4r_0, \cdots$. There is also a chance that the measured value of y is not close enough to one of the y_m to get the period from continued fractions. Again, if this happens we need to repeat the whole procedure. However, we will not have to repeat very many times because the probability of success in one run is quite high.

The probabilistic nature of Shor's algorithm, with the resultant need to run the algorithm several times (usually not very many), is a quite common feature of quantum algorithms.

18.6 An example

The last section was probably hard going, so we will try to clarify things by discussing a simple example. Consider the following, which was also discussed in Chapter 15, $N = 91, a = 4$. As shown in Eq. (15.6), the period is $r = 6$. Since the period is not a power of 2 this is a generic example, as discussed in the previous two sections.

order (m)	peak position ($y_m = m\,2^n/r$)	nearest integer	P(nearest int.)
0	0	0	0.167
1	2730.67	2731	0.114
2	5461.33	5461	0.114
3	8192	8192	0.167
4	10922.67	10923	0.114
5	13653.33	13653	0.114

Table 18.2: The peak positions in the Fourier transform for the example discussed in this chapter. The output is at integer values of y and the nearest integers to the peaks are shown along with the probability at those nearest integer values. Neglecting the zeroth order peak at $y = 0$, which doesn't give useful information, the sum of the other probabilities at the nearest integers is 0.623, so we have a greater than 60% probability of obtaining the nearest integer to a non-zero multiple of $2^n/r$, from which one can deduce r using continued fractions, as discussed in the text and Appendix 18.B.

[7] If we have two approximants for $y/2^n$, p/q and p'/q' say, then

$$\left|\frac{p}{q} - \frac{p'}{q'}\right| = \frac{|pq' - p'q|}{2qq'} > \frac{1}{N^2}$$

(since q and q' are less than N) unless the two approximants are equal, so there is at most one approximant with denominator less than N which satisfies Eq. (18.26). Since successive approximants give better approximations, the unique partial fraction that we want must be the one with the *largest* denominator less than N.

One needs $n_0 = 7$ bits to represent N so we take $n = 2n_0 = 14$. Hence

$$\frac{2^n}{r} = 2730.67 \tag{18.29}$$

so

$$Q = 2730\,. \tag{18.30}$$

Hence there are 2730 (and two thirds) periods in our data. As discussed in Mermin [Mer07] and Sec. 18.5 we need at least $N\,(= 91)$ periods so 2730 is something of an overkill. The peaks in the Fourier transform, which are at integers next to multiples of $2^n/r$ as discussed above, are shown in Table 18.2.

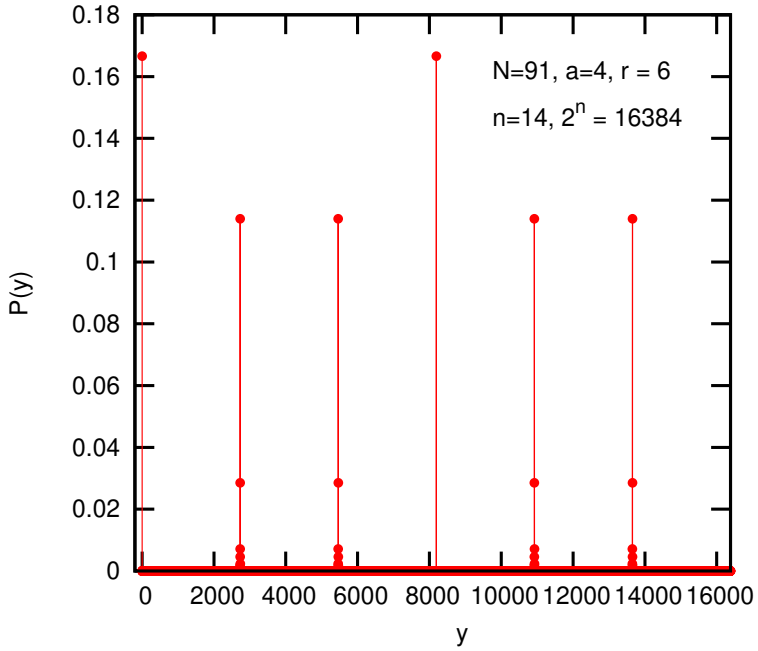


Figure 18.8: Probabilities for the different components of the Fourier transformed state for the example studied with $N = 91, a = 4$ for which the period is $r = 6$. There are six sharp peaks near $y_m = m\,2^n/r$, for $m = 0, 1, \cdots, 5$. The one at $y = 0\,(m = 0)$ doesn't give useful information. However, the probability of hitting the highest point of one of the other five peaks, i.e. the nearest integer to a non-zero multiple of $2^n/r$, is greater than 60%, see Table 18.2. If, as is likely, the measurement gives one of these results, it can then be used to determine the period r, as discussed in the text and Appendix 18.B. A blowup of the $m = 2$ peak is shown in Fig. 18.9.

I have evaluated $P(y)$ numerically from Eq. (18.11) and the results are shown in Fig. 18.8. There are $r = 6$ peaks at values close to $y_m = m\,2^2/r$. There is a trivial peak at exactly $y = 0\,(m = 0)$ but this can not give any useful information about the period r. For the other 5 peaks, the peaks are not, in general, centered at exactly integer values, so the possible observed (integer) values of y are a set of discrete values around each peak, as shown in the histogram in Fig. 18.9 which blows up the region around the $m = 2$ peak.

As discussed in Sec. 18.5, the sum in Eq. (18.11) can be evaluated, and is given, to a good approximation, by Eq. (18.23) in the region of the m-th peak, where y is given by Eq. (18.20), and y_m, given by Eq. (18.12), indicates the peak position. (Recall that y itself is an integer.) The function in Eq. (18.23) is plotted for continuous y as the solid curve in Fig. 18.9. When evaluated at integer

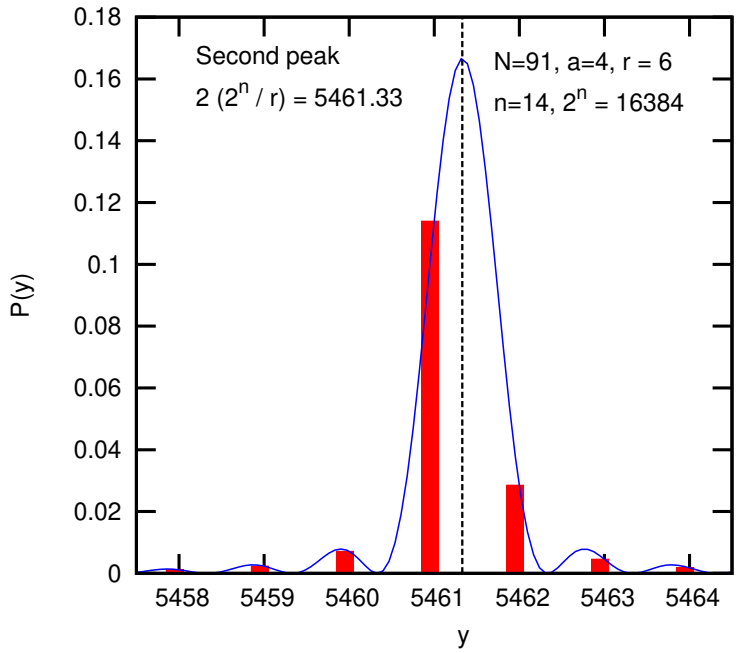


Figure 18.9: A blowup of the region around the $m = 2$ peak in Fig. 18.8 (see also Table 18.2). The histogram is obtained from numerical evaluation of Eq. (18.11). The probability is dominated by the biggest bar, which is at $y = 5461$ the nearest integer to $y_2 = 2 \times (2^n/r) = 5461.33$ (indicated by the vertical dashed line). The solid curve is the expression shown in Eq. (18.23), with y $(= y_m + \delta_m)$ considered to be a continuous variable.

y, it agrees very well with the values numerically computed from Eq. (18.11) which are shown as the histogram in Fig. 18.9.

Note that δ_m in Eq. (18.23) is defined in Eq. (18.20) and can be written as

$$\delta_m = \epsilon + \ell \tag{18.31}$$

where ℓ is an integer and $|\epsilon| < 0.5$. Note too that

$$\sum_{\ell=-\infty}^{\infty} \left(\frac{\sin(\pi(\epsilon+\ell))}{\pi(\epsilon+\ell)} \right)^2 = 1, \tag{18.32}$$

for arbitrary ϵ (recent versions of Mathematica know this). Hence, according to Eqs. (18.23), (18.31), and (18.32), the weight around each of the peaks in Fig. 18.8 is equal to $1/r$ $(= 1/6$ here). There are r peaks so the total probability is $r \times (1/r) = 1$ as required. Referring to Fig. 18.9, the weight in the largest bar is 0.114 which is 68% of 1/6, the total weight in all the bars for this $(m = 2)$ peak.

From Table 18.2 we see that the probability of getting the nearest integer to an integral multiple of $2^n/r$ is greater than 60%. Let's suppose we get one of these. In fact, lets suppose we get the large bar at $y = 5461$ in Fig. 18.9. (Recall that Fig. 18.9 is a blowup of the $m = 2$ peak in Fig. 18.8.) Given the measured value, $y = 5461$, we will now see how to determine the period r using continued fractions.

We define $x = y/2^n$. This is close to m/r, where r, the period, is what we want to determine. Since r is no greater than N, as discussed in Sec. 18.5, the best guess for x is the partial sum having the largest denominator less than N. As stated above we assume in this example that the measurement gave the value $y = 5461$, the highest histogram for the peak in Fig. 18.9. We therefore determine the

continued fraction representation for $x = 5461/16384$ (since $n = 14$ we have $2^n = 16384$). Since this is a rational fraction the continued fraction terminates.

We use the methods of Appendix 18.B to determine coefficients as follows. We have $c_0 = [x] = 0$ (note: $[\cdots]$ means the integer part of what is in the brackets). We subtract c_0 from x and call the inverse of the remainder x_1, so $x_1 = 16384/5461$. c_1 is the integer part of x_1 so $c_1 = 3$. Subtract c_1 from x_1 and call the inverse of the remainder x_2. Since $x_1 - c_1 = 1/5416$, we have $x_2 = 5461$. Since this is an integer, the continued fraction terminates at this point. Hence the coefficients are

$$c_0 = 0,\ c_1 = 3,\ c_2 = 5461, \tag{18.33}$$

and the corresponding partial sums are

$$\begin{aligned} c_0 &= 0, \\ c_0 + \frac{1}{c_1} &= \frac{1}{3}, \\ c_0 + \cfrac{1}{c_1 + \cfrac{1}{c_2}} &= \frac{5416}{16384}. \end{aligned} \tag{18.34}$$

The last result has a denominator bigger than N $(= 91)$ so we neglect it and conclude that[8]

$$\frac{m}{r} = \frac{1}{3}\,. \tag{18.35}$$

It is possible that m and r have a common factor, i.e. $m = c, r = 3c$ for some integer c. We try some small values for c. Starting with $c = 1$, so $r = 3$, we compute $a^3 \pmod{91}$ and find that it is not 1, see Eq. (15.6c). However, we find that $c = 2$ does work, since $a^6 \equiv 1 \pmod{91}$, see Eq. (15.6f). Hence the period r is equal to 6, the desired result.

18.7 Summary

What is the operation count for Shor's period finding algorithm?

To factor an integer with n bits, the QFT requires, in principle, $O(n^2)$ operations, as shown in section 18.3. Note, however, as discussed there, in Appendix 18.C, and in Mermin [Mer07], in practice one only needs of order $n \log_2 n$ gates to perform the QFT to within the necessary precision.

The computation of the function values using modular exponentiation takes $O(n^3)$ operations, as shown in section 18.2 (but see footnote 2 on page 143 which states that the operation count is $O(n^2 \log n \log\log n)$, not much more than $O(n^2)$, if one uses a sophisticated method for multiplying two large numbers).

What about the continued fraction part, which is, of course, done on a classical computer? Each division of an n-bit number takes of order n^2 operations if the division is done in a simple way. In fact, division can be rewritten as several multiplications, see `https://en.wikipedia.org/wiki/Division_algorithm`, so the operation count can be reduced to that for multiplication, i.e. $O(n \log n \log\log n)$. The depth of the continued fraction where the denominator is $O(N)$ is $O(\log N)$, since the coefficients in the continued fraction multiply to get the numerator and denominator. This is $O(n)$ since N contains no more than $n/2$ bits. Hence the operation count for the continued fraction post-processing is $O(n^3)$,

[8] In this case, where there are many more than N periods in the intervals 2^n, one gets the right answer if the measurement gives one of the other nearby y values. For example, if we get $y = 5460$ (the third closest to the peak), the continued fraction coefficients are $0, 3$, and 1365 which give the partial sums $0, 1/3, 1365/4096$. The last value has a denominator greater than N, so we ignore it and take the previous partial sum, again getting $m/r = 1/3$.

but recall that this is done on a classical computer. Again the count is not much more than $O(n^2)$ if one uses a sophisticated method for dividing two large numbers. Hence, the overall operation count of Shor's algorithm is[9] $O(n^3)$.

Shor's algorithm for factoring integers therefore runs in polynomial time as a function of n, the number of bits in N. For comparison, no polynomial time classical algorithm for factoring integers is known. The fastest classical algorithm at present, the general number field sieve (GNFS), takes a time $\exp(\text{const.}\, n^{1/3} \log^{2/3} n)$. It is currently not known whether there exists a yet to be determined polynomial time classical algorithm for factorization.

Even though the power of n in the exponent of the GNFS algorithm is less than one, it still much slower for large n than Shor's polynomial-time algorithm. Hence, if the considerable technical difficulties could be overcome, and a quantum computer with a sufficiently large number of qubits built with the error rate made sufficiently low, then such a device could decode encrypted messages currently being sent down the internet which are currently impossible to decode on a classical computer.

Appendices

18.A Eliminating the two-qubit gates

It is possible to replace the 2-qubit gates by 1-qubit gates which act or not depending on the result of a measurement. This is important from a technological point of view since 1-qubit gates are much easier to implement than 2-qubit gates. The point is that we measure the final state of the QFT anyway, and we will see that we can eliminate the 2-qubit gates by measuring each qubit *immediately after all the gates of the QFT have acted on it* rather than waiting until the QFT is completed. We now see how to do this.

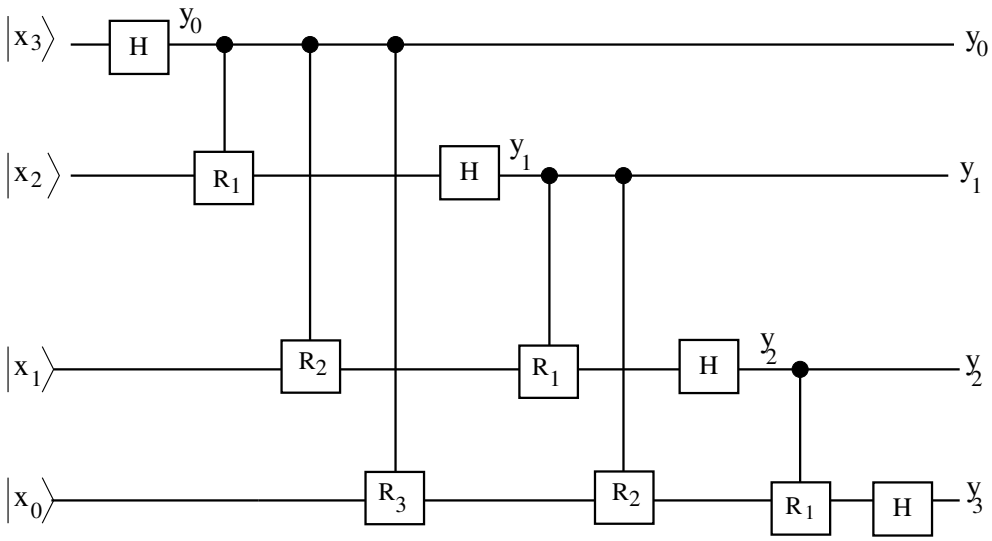


Figure 18.10: Circuit equivalent to Fig. 18.4 but with the target and control qubits interchanged on the controlled phase gates.

First of all we note that, similar to the control-Z gate, the target and control qubits in the controlled phase gates can be interchanged. Hence Fig. 18.4 is equivalent to Fig. 18.10.

[9] This can be reduced to $O(n^2 \log n \log\log n)$ using sophisticated methods for multiplying and dividing large numbers.

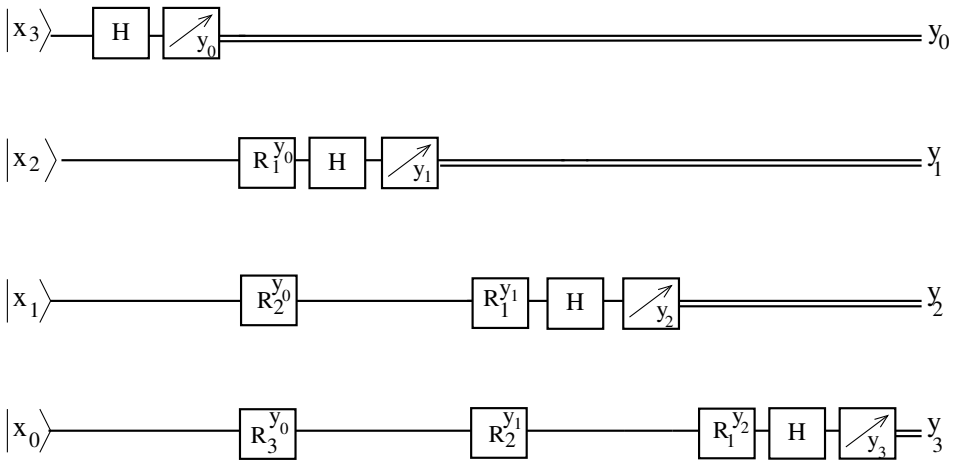


Figure 18.11: Circuit for the QFT with 4 qubits equivalent to Fig. 18.10 but in which each qubit is measured immediately after the Hadamard gate. Subsequent phase gates (on qubits lower in the diagram) are controlled by classical circuits (not shown) which use the values of the already measured qubits. Note that $R_1^{y_0}$ means R_1 to the power y_0. Since y_0 is 0 or 1 this gives R_1 if $y_0 = 1$ and 1 if $y_0 = 0$. Hence we obtain the required control, but done by a classical circuit rather than the 2-qubit controlled phase gates in Fig. 18.10.

In Fig 18.10 we see that, for each qubit, once the phase gates Hadamard and have acted the qubit doesn't change, so it could be measured at this point. (Recall that time flows from left to right in circuit diagrams). Consider the top qubit x_3 which, on output, is y_0. We measure it immediately after the Hadamard. If the result is $y_0 = 1$ then the R_1 phase gate for x_2 is activated, as well as the R_2 phase gate for x_1 and the R_3 phase gate for x_0. However, if the result is $y_0 = 0$ then those phase gates are not activated. Since y_0 has been measured, this control can be done by a *classical* circuit, which is much simpler to implement than a 2-qubit quantum gate. Similarly we measure x_2, which is y_1 on output, immediately after its Hadamard. Hence the R_1 gate on x_1 and the R_2 gate on x_0 can be activated classically if $y_1 = 1$. We can proceed in this way for the whole circuit, measuring the qubit after the Hadamard, and using the result to phase change other qubits, or not, using classical control. The circuit is shown in Fig. 18.11.

18.B Continued Fractions

Continued fractions are a convenient way of finding a simple rational approximation to a number.

The continued fraction representation of a number x is obtained as follows. If there is an integer part of x call this c_0. Subtract c_0 from x and call the inverse of the remainder x_1, so

$$x = c_0 + \frac{1}{x_1}. \tag{18.36}$$

Let the integer part of x_1 be c_1. Subtract c_1 from x_1 and call the inverse of the remainder x_2 so $x_1 = c_1 + 1/x_2$. Continuing in the same way for c_2 and x_3 etc. we get

$$x = c_0 + \cfrac{1}{c_1 + \cfrac{1}{x_2}} = c_0 + \cfrac{1}{c_1 + \cfrac{1}{c_2 + \cfrac{1}{x_3}}} \cdots = c_0 + \cfrac{1}{c_1 + \cfrac{1}{c_2 + \cfrac{1}{c_3 + \cdots}}}. \tag{18.37}$$

To evaluate continued fractions we start at the bottom. For example if we wish to evaluate

$$x = \cfrac{1}{2 + \cfrac{1}{5 + \cfrac{1}{4}}} \tag{18.38}$$

we determine first that

$$5 + \frac{1}{4} = \frac{21}{4} \tag{18.39}$$

and then that

$$2 + \frac{4}{21} = \frac{46}{21} \tag{18.40}$$

so

$$x = \frac{21}{46}. \tag{18.41}$$

If we stop after a certain number of iterations and ignore the remainder we have a "partial sum", which is an approximation for x. After each iteration the approximation improves. If x is a rational number (ratio of two integers) the continued fraction will eventually terminate. If x is irrational (like π) the continued fraction will go on for ever. The first few continued fraction coefficients $c_i (i = 0, 1, 2 \cdots)$ for $\pi = 3.141592654\ldots$ are

$$3, 7, 15, 1, 292, 1, \cdots . \tag{18.42}$$

It is a property of continued fractions, which you can verify, that if a relatively large coefficient appears at some point, stopping the continued fraction at the previous coefficient gives an accurate approximation to the number. For, example, omitting 15 and subsequent coefficients in Eq. (18.42) gives the well known approximation[10]

$$\mathbf{3} + \frac{1}{\mathbf{7}} = \frac{22}{7} = 3.14286\ldots , \tag{18.43}$$

which has an error of about 10^{-3} (the continued fraction coefficients are in bold).

In the present case we are interested in the continued fraction representation of $y/2^n$, which is a rational fraction so the continued fraction will eventually terminate. As discussed in the text, the value of $y/2^n$ is close to m/r where r is no bigger than N (N can be represented by n_0 qubits with $n_0 = n/2$). So we are interested in a continued fraction *approximation* to $y/2^n$ with a denominator no bigger than N. (Recall that $2^n = (2^{n_0})^2$ which is greater than N^2.)

Consider the example described in this chapter which has $N = 91, a = 4$ and $n = 14$ so $2^n = 16384$. The most probable results for y are those in the column labeled "nearest integer" in Table 18.2. Suppose the measurement of y gives the nearest integer for $m = 5$, i.e. 13653. The continued fraction

[10] A much more accurate result is obtained by omitting 292 and subsequent terms, which gives a value $355/113 = 3.141592920\ldots$, which has an error of a bit less than 3×10^{-7}. This rational approximation to π was apparently first obtained by a Chinese mathematician Zu Chouygzhi about 1500 years ago.

representation of 13653/16384 is obtained as follows:

$$\begin{aligned} x &= \frac{13653}{16384}, \\ c_0 &= [x] = 0, \qquad x_1 = (x-c_0)^{-1} = \frac{16384}{13653} \\ c_1 &= [x_1] = 1, \qquad x_2 = (x_1-c_1)^{-1} = \frac{13653}{2731} \\ c_2 &= [x_2] = 4, \qquad x_3 = (x_2-c_2)^{-1} = \frac{2731}{2729} \\ c_3 &= [x_3] = 1, \qquad x_4 = (x_3-c_3)^{-1} = \frac{2729}{2} \\ c_4 &= [x_4] = 1364, \quad x_5 = (x_4-c_4)^{-1} = 2 \\ c_5 &= [x_5] = 2, \end{aligned} \tag{18.44}$$

and the series terminates since x_5 is an integer. Hence the exact continued fraction coefficients of 13653/16384 are

$$0, 1, 4, 1, 1364, 2\,. \tag{18.45}$$

Successive partial sums are $0, 1, 4/5, 5/6, 6824/8189$ and 13653/16384. We want the partial sum with the largest denominator less than N (= 91), which is 5/6. This tells us, if m and r have no common factors, that $m = 5$ and $r = 6$.

We check if $r = 6$ works by directly calculating $4^6(\mod 91)$. We find that it is equal to 1, see Eq. (15.6f), so the period is indeed 6. According to Appendix M in Mermin [Mer07] the probability of two large randomly chosen numbers not having a common factor is greater than 1/2. If we are unlucky and the assumption of no common factor does not work, then usually we would only have to try a few values for the common factor i.e. $2, 3, 4, \cdots$, before succeeding. If we are *really* unlucky, and the common factor is very large, we would give up at some point, start again and get a different value for y. In the related example studied in detail in Sec. 18.6, where the measurement is assumed to give the nearest integer to the second peak, the common factor turns out to be 2.

18.C Unimportance of Small Phase Errors

The action of the controlled-phase gate is given by Eq. (17.10) and the QFT requires, in principle, these gates for $d = 1, 2, \cdots, n-1$. The total number of controlled phase gates is therefore $1+2+\cdots+n-1 = O(n^2)$. However, it is clearly impossible to accurately construct a phase gate for a phase which is exponentially small in n if n is large. For factoring, n would typically be several thousand.

Fortunately it is not necessary to include controlled phase gates with such small phase changes. Mermin [Mer07] shows that one can generate the closest integer to a multiple of $2^n/r$ within almost the same probability as when one includes all gates (reduced by at most 1%) if one neglects controlled phase gates with $d > d^\star = \log_2(Cn)$, where the constant C is quite large (500π) but independent of n. Hence, in practice, one only needs of order $d^\star n$ controlled phase gates ($\sim n \log_2 n$) to obtain the desired result, rather than $O(n^2)$ which would be needed if one includes all the gates with d up to n. Hence the size of the circuit does not grow much faster than n which is a huge improvement compared with $O(n^2)$ if n is several thousand.

Chapter 19

Quantum Error Correction

19.1 Introduction

Quantum error correction has developed into a huge topic, so here we will only be able to describe the main ideas.

Error correction is essential for quantum computing, but appeared at first to be impossible, for reasons that we shall soon see. The field was transformed in 1995 by Shor [Sho95] and Steane [Ste96] who showed that quantum error correction *is* feasible. Before Shor and Steane, the goal of building a useful quantum computer seemed clearly unattainable. After those two papers, while building a quantum computer obviously posed enormous experimental challenges, it was not *necessarily* impossible.

Some general references on quantum error correction are Refs. [Mer07, NC00, Vat16, RP14].

Let us start by giving a simple discussion of classical error correction which will motivate our study of quantum error correction. Classically, error correction is not necessary for computation. This is because the hardware for one bit is huge on an atomic scale and the states 0 and 1 are so different that the probability of an unwanted flip is tiny. However, error correction is needed classically for transmitting a signal over large distances where it attenuates and can be corrupted by noise.

To perform error correction one needs redundancy. One simple way of doing classical error correction is to encode each *logical* bit by three *physical* bits, i.e.

$$|0\rangle \to |\overline{0}\rangle \equiv |0\rangle|0\rangle|0\rangle \equiv |000\rangle\,, \tag{19.1a}$$
$$|1\rangle \to |\overline{1}\rangle \equiv |1\rangle|1\rangle|1\rangle \equiv |111\rangle\,, \tag{19.1b}$$

(for convenience we are using Dirac notation here even though these are classical bits for now.) The sets of three bits, $|000\rangle$ and $|111\rangle$, are called *codewords*. One monitors the codewords to look for errors. If the bits in a codeword are not all the same one uses "majority rule" to correct. For example

$$\begin{aligned} &|010\rangle \text{ is corrected to } |000\rangle \\ &|110\rangle \text{ is corrected to } |111\rangle. \end{aligned} \tag{19.2}$$

This works if no more than one bit is corrupted and so the error rate must be sufficiently low that the probability of two or more bits in a codeword being corrupted is negligible.

In quantum error correction one also uses multi-qubit codewords and monitoring. However, there are several major differences compared with classical error correction:

(i) *Error correction is essential.* Quantum computing requires error correction. This is because the physical systems for a single qubit are very small, often on an atomic scale, so any small outside interference can disrupt the quantum state.

(ii) *Measurement destroys quantum information.* In contrast to the classical case checking for errors is problematic. Monitoring means measuring, and measuring a general quantum state alters it. Thus it seems that any attempt at error correction must destroy important quantum information.

(iii) *More general types of error can occur.* Bit flips are not the only possible errors. For example one can have phase errors where $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{i\phi}|1\rangle)$.

(iv) *Errors are continuous.* Unlike all-or-nothing bit flip errors for classical bits, errors in qubits can grow continuously out of the uncorrupted state.

One might imagine that point (ii), in particular, would be fatal. Amazingly this is not so as we shall see.

19.2 Correcting bit flip errors

We start our discussion of quantum error correction by considering how one can correct for just bit flip errors. If the error rate is low we might hope to correct them by tripling the number of bits as in the classical case, Eq. (19.1).

The tripling of the qubits can be accomplished by the circuit in Fig. 19.1. To see how this works suppose that the input qubit, $|x\rangle$, is $|0\rangle$. Then none of the CNOT gates act on their target qubit so all three qubits are $|0\rangle$ at the end (i.e. on the right). However, if the input qubit $|x\rangle$ is $|1\rangle$ then the CNOT gates act so all three qubits are 1 at the end.

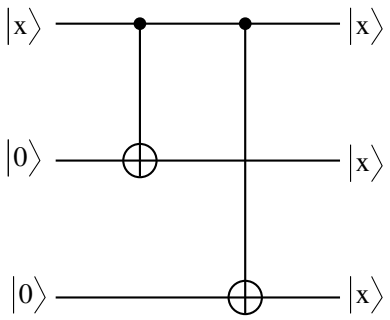


Figure 19.1: Circuit to encode the 3-qubit bit-flip code. Here $|x\rangle$ is $|0\rangle$ or $|1\rangle$ in the computational basis. The effect of this circuit on a linear combination of $|0\rangle$ and $|1\rangle$ is shown in Fig. 19.2.

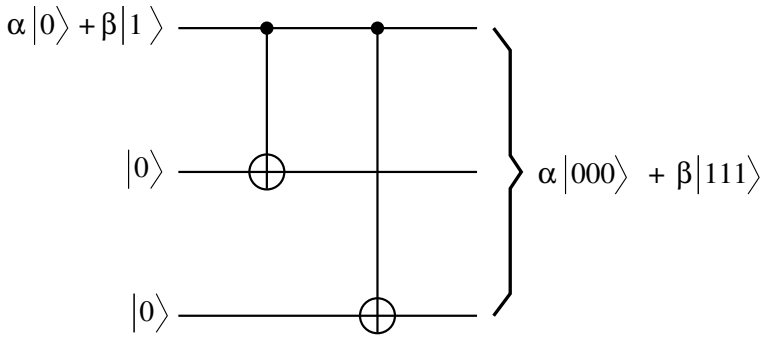


Figure 19.2: Circuit to encode the 3-qubit bit-flip code acting on a linear combination of $|0\rangle$ and $|1\rangle$.

By linearity a linear combination of $|0\rangle$ and $|1\rangle$ is transformed as we want:

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|000\rangle + \beta|111\rangle\,, \tag{19.3}$$

see Fig. 19.2. Note that this is not a clone of the input state which would be

$$(\,\alpha|0\rangle + \beta|1\rangle\,)^{\otimes 3} = \alpha^3|000\rangle + \alpha^2\beta\,(\,|001\rangle + |010\rangle + |100\rangle\,) + \alpha\beta^2\,(\,|110\rangle + |101\rangle + |011\rangle\,) + \beta^3|111\rangle\,. \tag{19.4}$$

We recall that cloning an arbitrary unknown state is impossible according to the no-cloning theorem.

Next an aside on notation. The CNOT gate is usually written with a $\oplus$ symbol (as in Figs. 19.1 and 19.2) to indicate the XOR operation but it is often more illuminating to write it, in an equivalent way, with an X symbol (in a square) to indicate that the NOT (i.e. bit-flip) operation is performed with the operator X (recall that in a CNOT gate the target qubit is flipped if the control qubit is 1.) From now on in this chapter we shall use the symbol X inside a square when drawing a CNOT gate and call it Control-X, see e.g. Fig. 19.3.

Now we have to check if any of the three qubits generated by the circuit in Fig. 19.2 are flipped so the situation is that shown in Fig. 19.3. We assume that no more than one has been flipped, which is a reasonable approximation if the error rate is small.

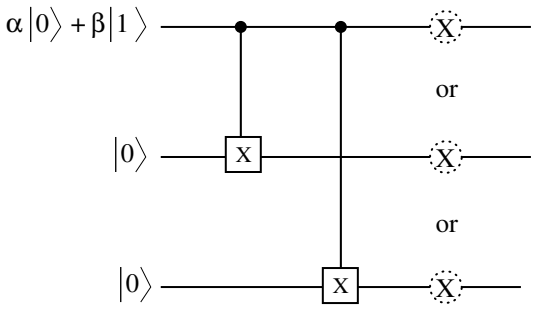


Figure 19.3: Circuit indicating that at most one of the three bits generated by the circuit in Fig. 19.2 has flipped due to an error. The goal will be to determine whether any have flipped, if so which one, and then correct the error. Note that the Control-X gates here are identical to the CNOT gates in Figs. 19.1 and 19.2. (Control-X and CNOT are just different ways of describing the same gate.)

We have therefore one uncorrupted state and three corrupted states:

$$|\,\psi\,\rangle = \alpha|000\rangle + \beta|111\rangle\,, \tag{19.5a}$$

$$|\psi_1\rangle = \alpha|100\rangle + \beta|011\rangle = X_1|\,\psi\,\rangle \qquad \text{(qubit 1 flipped)}\,, \tag{19.5b}$$

$$|\psi_2\rangle = \alpha|010\rangle + \beta|101\rangle = X_2|\,\psi\,\rangle \qquad \text{(qubit 2 flipped)}\,, \tag{19.5c}$$

$$|\psi_3\rangle = \alpha|001\rangle + \beta|110\rangle = X_3|\,\psi\,\rangle \qquad \text{(qubit 3 flipped)}\,. \tag{19.5d}$$

These four states are called the "syndromes". Note that we denote the left hand qubit as the first qubit, the one to its right as the second qubit, and so on, e.g. $|x_1x_2x_3\rangle$. Hence in Eq. (19.5) $|\psi_i\rangle$ refers to the state in which qubit i is flipped relative to the uncorrupted state $|\psi\rangle$.

Classically, to determine if one of the bits is flipped we just have to look at them. However, quantum mechanically, if we measure $|\psi\rangle$, say, we get $|000\rangle$ with probability $|\alpha|^2$ and $|111\rangle$ with probability $|\beta|^2$, which destroys the coherent superposition. It might therefore seem that quantum error correction is impossible.

Amazingly this is not so. The secret is to couple the codeword qubits to ancillary qubits and measure only these. This will give enough information to determine which syndrome the state is in *without destroying the coherent superposition.*

Here we need two ancillary qubits. The circuit including them is shown in Fig. 19.4. The three codeword qubits are at the bottom and the ancillary qubits are at the top. The ancillary qubits are measured and give values x and y. We shall now see that each of the four possible pairs of values for x and y corresponds to one of the syndrome states in Eq. (19.5).

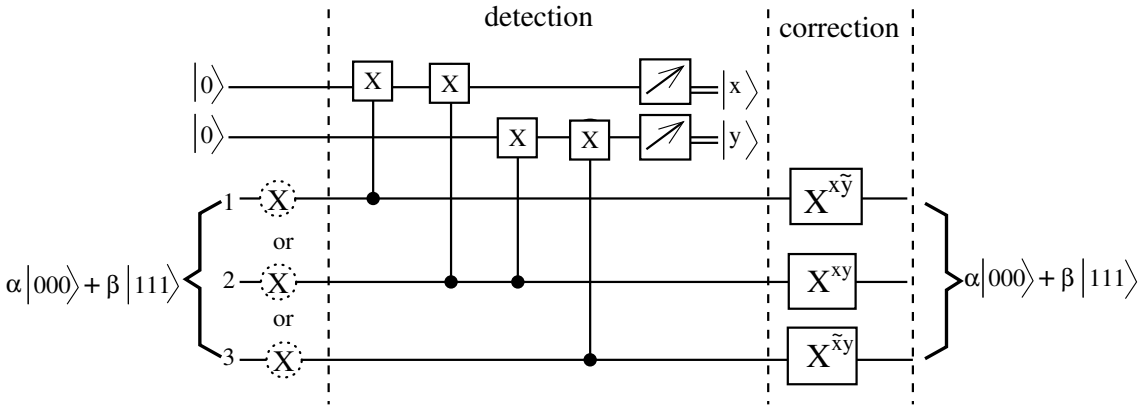


Figure 19.4: Circuit to determine the syndrome for the 3-qubit bit-flip code, and correct if necessary. A box with an arrow denotes a measurement. The double lines indicate that the result of a measurement is a classical bit.

Both ancillas are targeted by two of the codeword qubits.

1st (upper) ancilla (x) is targeted by codeword qubits 1 and 2.
2nd (lower) ancilla (y) is targeted by codeword qubits 2 and 3.

Let's see what happens for the four syndrome states.

$|\psi\rangle$ Codeword $|000\rangle$. No ancilla flipped so $x = 0, y = 0$.
Codeword $|111\rangle$. Both ancillas are flipped twice so again $x = 0, y = 0$.
Note that the result of the measurement is the same for both the $|000\rangle$ and $|111\rangle$ parts of the state $|\psi\rangle$. Hence the coherent superposition of $|\psi\rangle$ is not destroyed by the measurement on the ancillas. If the result of the measurement were different for the different parts of the superposition, then only the piece corresponding to the measured value would survive and the superposition would be broken.

$|\psi_1\rangle$ Codeword $|100\rangle$. x is flipped once, and y is not flipped, so $x = 1, y = 0$.
Codeword $|011\rangle$. x is flipped once and y is flipped twice so again $x = 1, y = 0$.
Recall that the qubits are ordered such that qubit 1 is on the left.

$|\psi_2\rangle$ Codeword $|010\rangle$. x and y are both flipped once so $x = 1, y = 1$.
Codeword $|101\rangle$. x and y are both flipped once so again $x = 1, y = 1$.

$|\psi_3\rangle$ Codeword $|001\rangle$. x is not flipped and y is flipped once so $x = 0, y = 1$.
Codeword $|110\rangle$. x is flipped twice and y is flipped once so again $x = 0, y = 1$.

Hence we get the table of results shown in Table 19.1. Note that in all cases the coherent superposition of the syndrome state is not destroyed by the measurement of the ancillas.

Hence by measuring the auxiliary qubits we can determine which if any of the codeword qubits have flipped and then apply a compensating flip if necessary. The X-gates which perform these compensating flips are shown at the right of Fig. 19.4. For example the $X^{x\tilde{y}}$ gate on qubit 1 indicates that a flip

syndrome	bit flipped	x	y
$\vert\,\psi\,\rangle$	none	0	0
$\vert\psi_1\rangle$	1	1	0
$\vert\psi_2\rangle$	2	1	1
$\vert\psi_3\rangle$	3	0	1

Table 19.1: Results of measurement of the ancillary qubits for the different syndromes of the codeword qubits.

is done by acting with the X operator on qubit 1 only if $x = 1$ and $y = 0$, which corresponds to the second entry in the Table 19.1 ($\tilde{y}$ means the complement of y).

We have assumed up to now that the state of the system has had a bit flipped with probability one. However, as already noted, errors in quantum circuits can arise continuously from zero, and we are concerned with the situation in which the error rate is small (otherwise we can not error correct). Consider therefore a state $|\psi\rangle$ which has a small amplitude less than one to have a bit flipped, i.e.

$$|\psi\rangle \to [1 + (\,\epsilon_1 X_1 + \epsilon_2 X_2 + \epsilon_3 X_3\,)]\,|\psi\rangle\,, \tag{19.6}$$

where the ϵ_i may be complex, $|\epsilon_k| \ll 1$, we have only indicated terms to first order in the ϵ_k, and ignored the normalization. The probability of qubit k being flipped is $|\epsilon_k|^2$ to leading order. When we measure the ancilla qubits we project on to either the uncorrupted state or one of the three corrupted states which have one qubit flipped.

Since the ϵ_k are small, the *probability* that a corrupted state is detected is small, so the most probable situation is that projection is on to the uncorrupted state so no correction is needed. However, there is a small probability that the projection will be on to one of the corrupted syndromes. The corrupted syndromes differ *substantially* from the uncorrupted state. They are further, in fact, from the uncorrupted state than the original state in Eq. (19.6). This might, at first, seem like a retrograde step but it is not because the corrupted state is *known precisely* so it is possible to correct it back to to the uncorrupted state.

To summarize this part, quantum error correction is feasible, even though errors arise continuously, because possibly corrupted states are projected on to one of a *discrete* set of states which can be corrected if necessary. We will discuss this important point again in Sec. 19.5 when we consider how general errors arise.

It should be noted that in classical analog computers, where errors also arise continuously, no such projection can be done, and hence error correction can not be performed. This is why we don't have classical analog computers.

Going back to the discussion of Fig. 19.4, one can avoid explicitly measuring the qubits and instead correct any bit-flip error coherently and automatically by having the ancillas interact back on the codeword qubits as shown in Fig. 19.5. In that figure, the rightmost three controlled gates have the same effect as the NOT (i.e. X) gates in the right of Fig. 19.4 which depend on the result of measurements of the x and y ancillary qubits. The rightmost gate in Fig. 19.5 has two control qubits and three target qubits. This gate flips all the target qubits if *both* control qubits are 1. It is a generalization of the Toffoli gate T which has two control qubits, and one target qubit which is flipped if both control qubits are 1, i.e. $T|x\rangle|y\rangle|z\rangle = |x\rangle|y\rangle|z \oplus x\,y\rangle$. If we denote by T^* the rightmost gate in Fig. 19.5 then $T^*|x\rangle|y\rangle|z\rangle|u\rangle|v\rangle = |x\rangle|y\rangle|z \oplus x\,y\rangle|u \oplus x\,y\rangle|v \oplus x\,y\rangle$. Note that this gate is equivalent to three separate Toffoli gates, in which the two ancilla qubits are the controls, qubit 1 is the target for the first Toffoli, qubit 2 for the second Toffoli, etc. After the error on the computational bits has been corrected the ancilla qubits have to be reinitialized to zero.

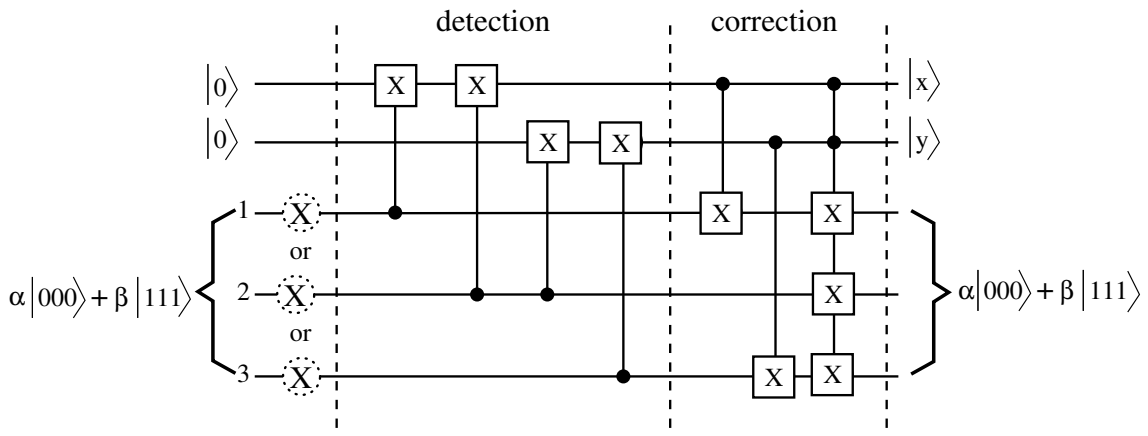


Figure 19.5: Automation of the error correction procedure of Fig. 19.4. The three controlled gates on the right have the same effect as the NOT (i.e. X) gates on the right of Fig. 19.4 which depend on the result of measurements of the x and y ancillary qubits. The rightmost gate, with two control qubits and three target qubits, is discussed in the text. The values of the control bits x and y at the end depend on which of the four syndromes is present (i.e. which if any of the X gates on the left of the figure have acted) according to Table 19.1. Before this circuit can be used again, the ancillary qubits have to be reinitialized to 0.

It is instructive to show for the different syndromes in Eq. (19.5) that the circuits in Figs. 19.4 and 19.5 give the same result, i.e. the end product is the uncorrupted state $|\psi\rangle$. The results from the circuit of Fig. 19.4 have already been discussed above. For the circuit in Fig. 19.5 we just consider the case of $|\psi_2\rangle$ (so qubit 2 has been flipped), and we have $x = 1, y = 1$ according to Table 19.1. Consider the rightmost three gates in Fig. 19.5 (these are the ones that do the error correction). For $x = 1, y = 1$, the rightmost gate is active and flips all three codeword qubits. Hence, between them, the rightmost three gates flip codeword qubit 1 twice, flip codeword qubit 2 once, and flip codeword qubit 3 twice. The net result is that only codeword qubit 2 is flipped so we recover the uncorrupted state $|\psi\rangle$. It is useful to check that the circuit in Fig. 19.5 also works to correct $|\psi_1\rangle$ and $|\psi_3\rangle$.

19.3 Stabilizer formalism

In order to conveniently generalize the ideas in the previous section to arbitrary errors we need to reformulate them.

For reasons that will shortly become clear, consider the two Hermitian[1] operators Z_1Z_2 and Z_2Z_3. Because $Z_i^2 = \mathbb{1}$ (the identity) and different $Z's$ commute we have

$$(Z_1Z_2)^2 = \mathbb{1}, \quad (Z_2Z_3)^2 = \mathbb{1}\,. \tag{19.7}$$

An operator whose square is unity has eigenvalues equal to ± 1, since acting twice with the operator on an eigenvector gives the eigenvector, so the square of the eigenvalue is 1. We also we know that Z_1Z_2 and Z_2Z_3 commute with each other *and hence have the same eigenvectors.*

One can verify that the syndrome states in Eq. (19.5) are eigenvectors of Z_1Z_2 and Z_2Z_3 according to Table 19.2. In general we use the term "stabilizers" to denote operators like operators Z_1Z_2 and Z_2Z_3 whose eigenvalues distinguish the different syndromes. As we will see below, each of the stabilizers

[1] As discussed in Chapter 3 it is an axiom of quantum mechanics that measurable quantities are represented by Hermitian operators.

syndrome		Z_1Z_2	Z_2Z_3	x	y
$\mid \psi \rangle$		1	1	0	0
$\|\psi_1\rangle$	$X_1\|\psi\rangle$	-1	1	1	0
$\|\psi_2\rangle$	$X_2\|\psi\rangle$	-1	-1	1	1
$\|\psi_3\rangle$	$X_3\|\psi\rangle$	1	-1	0	1

Table 19.2: The eigenvalues of the stabilizers Z_1Z_2 and Z_2Z_3 for the four syndromes for the 3-qubit bit-flip code, and a comparison with the measurements of the ancillary qubits x and y used to measure them, see Fig. 19.7. The uncorrupted state has eigenvalue $+1$ for both stabilizers. This is an important property that stabilizers must have in general. Note that $Z_1Z_2 = 1$ corresponds to $x = 0$, and $Z_1Z_2 = -1$ corresponds to $x = 1$. There is a similar connection between Z_2Z_3 and y, so $Z_1Z_2 = (-1)^x, Z_2Z_3 = (-1)^y$. The second column shows how the corrupted state is generated from the uncorrupted state.

is measured by an ancilla qubit, $|x\rangle$ for Z_1Z_2 and $|y\rangle$ for Z_2Z_3, see Fig 19.7 below. The ancilla state $|x = 0\rangle$ corresponds to $Z_1Z_2 = +1$, and $|x = 1\rangle$ corresponds to $Z_1Z_2 = -1$, or in other words, $Z_1Z_2 = (-1)^x$, and similarly $Z_2Z_3 = (-1)^y$.

Below we will discuss the circuit with which we measure the stabilizers, but first we show a more straightforward way to determine whether the eigenvalue of a stabilizer in a syndrome is $+1$ or -1 than simply acting with the stabilizer on the syndrome.

We note first that the eigenvalue of all the stabilizers is $+1$ in the uncorrupted syndrome $|\psi\rangle$. This is an essential property that stabilizers must have. Also note that the operators for the stabilizers will be built out of the single-qubit operators Z_i and X_i. For the 3-qubit, bit-flip code we only have the Z_i but the X_i will also be needed to correct for general errors. Furthermore the syndromes with a single qubit error are obtained by acting on the uncorrupted syndrome with the X_i, Y_i and Z_i operators.[2] Again, for our simple example above, we only had the X_i, but the other operators will be used when we deal with general errors.

The operators, X_i, Y_i, Z_i, have the property that they commute for different qubits i, whereas different operators on the same qubit anti-commute, where the anti-commutator of A and B is defined by $\{A, B\} \equiv AB + BA$. Hence we have, for example,

$$[X_i, Y_j] \equiv X_iY_j - Y_jX_i = 0 \quad (i \neq j)\,, \tag{19.8a}$$

$$\{X_i, Y_i\} \equiv X_iY_i + Y_iX_i = 0\,. \tag{19.8b}$$

(Verify the anti-commutation relations like Eq. (19.8b) by explicitly working out some cases.)

Consequently, if we consider a general stabilizer A_α and a syndrome state $|\psi_\beta\rangle = B_\beta|\psi\rangle$ then A_α either commutes or anti-commutes with B_β. Note that B_β only involves a single Pauli operator (which, in general, can be an X or a Y or a Z) whereas A_α involves a product of Pauli operators, which, in the general case, can be made up of X's and Z's. We will now show that if A_α commutes with B_β the eigenvalue of the stabilizer A_α in state $|\psi_\beta\rangle$ is $+1$ and if they anti-commute the eigenvalue is -1.

Firstly, if A_α commutes with B_β then

$$A_\alpha|\psi_\beta\rangle = A_\alpha B_\beta|\psi\rangle = B_\beta A_\alpha|\psi\rangle = B_\beta|\psi\rangle = |\psi_\beta\rangle\,, \tag{19.9}$$

where we used that the eigenvalues of all the stabilizers A_α are $+1$ in the uncorrupted state $|\psi\rangle$ to get the third equality. Hence the eigenvalue of A_α in state $|\psi_\beta\rangle$ is $+1$ if A_α commutes with B_β. Similarly

[2] Recall that the Pauli operators X, Y and Z are given by $X \equiv \sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y \equiv \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z \equiv \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and so $Y = iXZ$.

if A_α anti-commutes with B_β then

$$A_\alpha|\psi_\beta\rangle = A_\alpha B_\beta|\psi\rangle = -B_\beta A_\alpha|\psi\rangle = -B_\beta|\psi\rangle = -|\psi_\beta\rangle\,, \tag{19.10}$$

so the eigenvalue is -1.

We emphasize that the syndromes must be eigenstates of *all* the stabilizers which means that the *stabilizers must commute with each other.*

Next we will see how to determine efficiently if a stabilizer commutes or anti-commutes with the operator which generates a corrupted syndrome out of the uncorrupted state.

For the case of the 3-qubit, bit-flip code discussed so far the stabilizers are

$$Z_1Z_2 \text{ and } Z_2Z_3\,, \tag{19.11}$$

and the operators which generate the corrupted syndrome from the uncorrupted state are

$$X_1, X_2 \text{ and } X_3. \tag{19.12}$$

As an example, we see that X_1 commutes with Z_2Z_3 because there are no sites in common, so the eigenvalue of Z_2Z_3 for $|\psi_1\rangle$ must be $+1$ which agrees with Table 19.2. On the other hand X_2 has one site in common with Z_2Z_3 so

$$X_2\, Z_2Z_3 = -Z_2X_2Z_3 = -Z_2Z_3X_2\,, \tag{19.13}$$

and the operators anticommute, so the eigenvalue of Z_2Z_3 for $|\psi_2\rangle$ must be -1, which again agrees with Table 19.2.

> The point is that every time we have to interchange the order of two different operators acting on the same qubit we pick up a minus sign.

Hence it is straightforward to deduce the overall sign. Note that operators of the same type, e.g. the Z_i, always commute.

As a more complicated example, which occurs in a scheme for full error correction, consider the stabilizer $Z_3X_4X_5Z_1$. For the syndrome which has been corrupted by Z_4 the eigenvalue is -1, the minus sign coming from interchanging the order of X_4 and Z_4. However, for the syndrome which was corrupted by X_4 the eigenvalue is $+1$ since, for the qubit in common, (qubit 4), both operators are X and so commute. As another example, for the syndrome which was corrupted by X_2 the eigenvalue is $+1$, because X_2 and the stabilizer commute since they have no qubits in common.

To summarize, in the stabilizer formalism we need to construct a mutually commuting set of Hermitian operators (the stabilizers) which square to 1 and for which (i) the syndromes are eigenstates, (ii) the uncorrupted syndrome has eigenvalue $+1$ for all stabilizers, and (iii) the set of ±1 eigenvalues of the stabilizers uniquely specifies the syndrome. Whether the eigenvalue is $+1$ or -1 is easily determined from the commutation properties of the stabilizer with respect to the operator which generates the corruption in the syndrome. In Sec. 19.6 we will describe an example with full error correction which has codewords with 9 qubits and needs 8 stabilizers.

Next we describe the circuit which will measure the eigenvalues of the stabilizers and hence determine which syndrome has occurred. Consider the circuit in Fig. 19.6 which includes a control-U gate in which the control qubit is sandwiched between Hadamards. Here U is an operator, which, like the stabilizers, has eigenvalues ±1. If the control qubit is 1 the effect on the target qubit is

$$U|\psi_+\rangle = |\psi_+\rangle, \quad U|\psi_-\rangle = -|\psi_-\rangle, \tag{19.14}$$

where $|\psi_+\rangle$ and $|\psi_-\rangle$ are the eigenvectors with eigenvalue $+1$ and -1 respectively. If the control qubit is 0 then the target qubit is unchanged. We discussed this circuit in Chapter 8 and found that the states

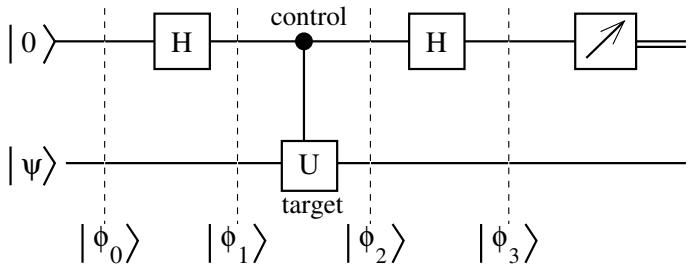


Figure 19.6: A circuit with a control-U gate in which the control (upper) qubit is surrounded by Hadamards. U is an operator with eigenvalues ± 1 and corresponding eigenvectors $|\psi_+\rangle$ and $|\psi_-\rangle$. As shown in the text, if a measurement of the upper qubit gives $|0\rangle$ then the lower qubit will be in state $|\psi_+\rangle$, and if the measurement gives $|1\rangle$ then the lower qubit will be in state $|\psi_-\rangle$. The states $|\phi_i\rangle\,(i = 0, 1, 2, 3)$ are described in the text. Note that this figure is identical to Fig. 8.8 and was discussed in Chapter 8.

$|\phi_i\rangle$, $(i = 0, 1, 2, 3)$ are given by Eqs. (8.21). In particular, the final state $|\phi_3\rangle$, before the measurement of the upper qubit, is given by

$$|\phi_3\rangle = \alpha_+|0\,\psi_+\rangle + \alpha_-|1\,\psi_-\rangle\,. \tag{19.15}$$

Hence if a measurement of the upper qubit gives $|0\rangle$ (which it does with probability $|\alpha_+|^2$) the lower qubit will be in state $|\psi_+\rangle$, and if the measurement gives $|1\rangle$ (probability is $|\alpha_-|^2$) the lower qubit will be in state $|\psi_-\rangle$. Hence we see that measuring the control qubit tells us which eigenstate of U the target qubit is in.

Stabilizers involve more than one codeword qubit so the gates we need will have several target qubits. For the 3-qubit, bit-flip code, the circuit equivalent to Fig. 19.4 is shown in Fig. 19.7. We see that the x ancilla is the control qubit for a control-Z_1Z_2 gate which is sandwiched between Hadamards, and similarly the y ancilla is the control qubit for a control-Z_2Z_3 gate. Hence if $x = 0$ the state of the codeword bits has $Z_1Z_2 = +1$, whereas if $x = 1$ the state of the codeword bits has $Z_1Z_2 = -1$. There is an analogous correspondence between y and Z_2Z_3.

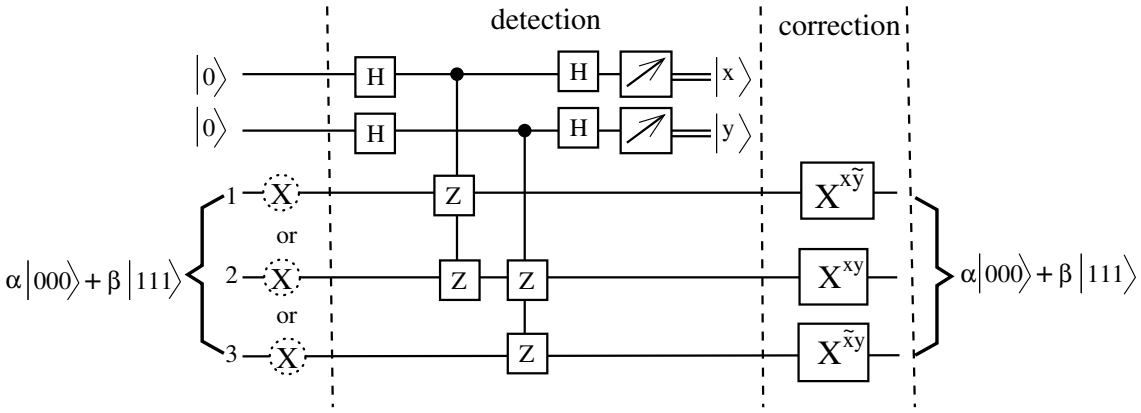


Figure 19.7: Circuit equivalent to that in Fig. 19.4 but in the stabilizer formalism. In this circuit x measures Z_1Z_2, and y measures Z_2Z_3. In other words, if $x = 0$ the state of the codeword bits has $Z_1Z_2 = +1$, whereas if $x = 1$ the state of the codeword bits has $Z_1Z_2 = -1$, with an analogous correspondence between y and Z_2Z_3. Note that Z_1Z_2 and Z_2Z_3 have eigenvalues ± 1 and commute with each other.

The equivalence of the circuits in Figs. 19.4 and 19.7 can also be understood from the simpler case of the equivalences shown in Fig. 19.8 in which the left-hand equality comes from the fact that the target and control qubits can be exchanged in a control-Z gate,[3] and the right-hand equality is because $HZH = X$ and $H^2 = \mathbb{1}$ (the identity).

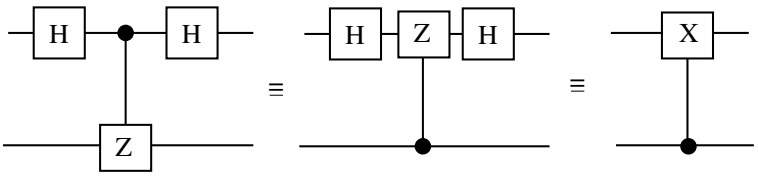


Figure 19.8: The equalities in this figure are helpful to understand the equivalence of Figs. 19.4 and 19.7. The left-hand equality comes from the fact that the target and control qubits can be exchanged in a control-Z gate, and the right-hand equality is because $HZH = X$ and $H^2 = \mathbb{1}$.

The stabilizer formalism will be convenient when devising circuits for full error correction rather than just correcting bit flips as we have done up to now.

19.4 Phase Flip Code

Before discussing how to correct general errors, we will briefly mention another special case, a phase flip, which has no classical equivalent since classical bits don't have any property corresponding to phase. In this error model, with some probability p, the relative phase of $|0\rangle$ and $|1\rangle$ is flipped so

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \;\rightarrow\; \alpha|0\rangle - \beta|1\rangle\,. \tag{19.16}$$

Phase flips are generated by the Z operator since

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \rightarrow Z\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}\alpha\\ -\beta\end{pmatrix} \quad \text{(computational basis)}. \tag{19.17}$$

The phase-flip error model can be turned into the already-studied bit-flip model by transforming to the $\pm$ basis (also called the X-basis because it is the basis in which X is diagonal) where

$$|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \quad |-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right), \tag{19.18}$$

One transforms between the $\pm$ basis and the computational basis using Hadamards:

$$H|0\rangle = |+\rangle, \quad H|1\rangle = |-\rangle, \tag{19.19a}$$
$$H|+\rangle = |0\rangle, \quad H|-\rangle = |1\rangle. \tag{19.19b}$$

In the $\pm$ basis the roles of X and Z are interchanged since

$$X|0\rangle = |1\rangle, \quad X|1\rangle = |0\rangle, \quad Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle, \tag{19.20a}$$
$$Z|+\rangle = |-\rangle, \quad Z|-\rangle = |+\rangle, \quad X|+\rangle = |+\rangle, \quad X|-\rangle = -|-\rangle. \tag{19.20b}$$

Thus we shall find in Sec. 19.6 that stabilizers to detect phase errors involve X operators, as opposed to those used to detect bit-flip errors which involve Z operators (see Fig. 19.7).

The encoding circuit for the 3-qubit phase-flip code is obtained from that for the 3-qubit bit-flip code in Fig. 19.2 by adding Hadamards to the circuit, with the result shown in Fig. 19.9. We shall use this circuit in Sec. 19.6 as part of the encoding circuit in Fig. 19.10 for a code (due to Shor) which corrects general 1-qubit errors.

[3]Because the only effect of the gate is to change the sign of the state if both target and control qubits are 1.

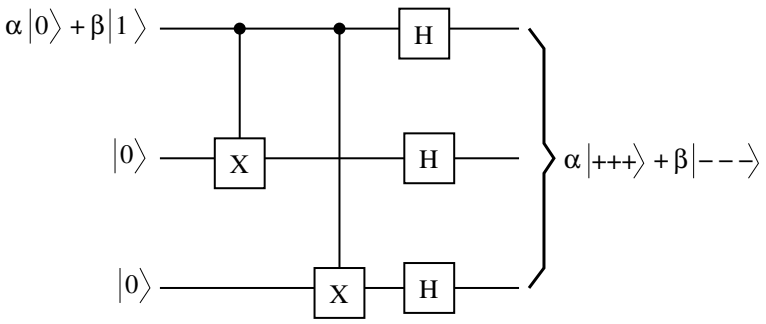


Figure 19.9: Encoding circuit for the 3-qubit phase-flip code.

19.5 General Errors and the Effects of the Environment

In our discussion of errors we have so far implicitly assumed that the errors occur because of some malfunction in the circuit. The state has underdone a unitary transformation, but not exactly the right one. Another, and very important, source of error is interaction between the qubits and the environment, which is unavoidable even though quantum computer engineers work very hard to reduce it to a minimum. This can lead to errors due to a *non-unitary* change in the computational qubits (though the combined system of qubits plus environment undergoes unitary time development.) In this section we include the effects of the environment and also consider the most general type of single qubit error.

Consider a single qubit $|x\rangle$, and call the environment $|e\rangle$. Unlike the state of the qubit, the state of the environment is in a space of very many dimensions. Ideally $|x\rangle$ evolves under the effects of the gates only, independent of the environment. However, interactions with the environment cannot be avoided which leads to a corruption of the qubit and an entangling of the qubit with the environment.

The most general such form of these effects is

$$|e\rangle\,|0\rangle \to |e_0\rangle\,|0\rangle + |e_1\rangle\,|1\rangle, \tag{19.21a}$$
$$|e\rangle\,|1\rangle \to |e_2\rangle\,|0\rangle + |e_3\rangle\,|1\rangle, \tag{19.21b}$$

where $|e_i\rangle$ $(i = 0, \cdots, 3)$ are possible final states of the environment. The environment states are not normalized, and not orthogonal either. However, the two states on the right hand side of Eqs. (19.21) must be orthogonal since the time evolution of the combined qubit-environment system is unitary. In other words

$$\langle e_2|e_0\rangle + \langle e_3|e_1\rangle = 0\,. \tag{19.22}$$

The corruption of the computation by the environment indicated in Eq. (19.21) is called "decoherence". It is the main source of difficulty in building a practical quantum computer.

In previous sections we have neglected entanglement with the environment. Rather, errors were assumed to occur because of mistakes made in the circuit itself. This corresponds to a special case of Eqs. (19.21), where all the environment states are the same, apart from normalization, i.e. $|e_i\rangle = c_i|e\rangle$, for $i = 0, \cdots, 3$.

We are interested in the case where the probability of an error is small (otherwise we would not be able to correct for it), i.e.

$$\langle e|e\rangle = 1, \quad \langle e_0|e_0\rangle \simeq 1, \quad \langle e_3|e_3\rangle \simeq 1, \quad \langle e_1|e_1\rangle \ll 1, \quad \langle e_2|e_2\rangle \ll 1. \tag{19.23}$$

Equations (19.21) can be combined into one as

$$|e\rangle\,|x\rangle \to \left\{ \left(\frac{|e_0\rangle + |e_3\rangle}{2} \right) \mathbb{1} + \left(\frac{|e_0\rangle - |e_3\rangle}{2} \right) Z + \left(\frac{|e_2\rangle + |e_1\rangle}{2} \right) X + \left(\frac{|e_2\rangle - |e_1\rangle}{2} \right) (iY) \right\} |x\rangle, \tag{19.24}$$

where $x = 0$ or 1 and, as usual,[4]

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad iY = ZX = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad \mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{19.25}$$

Please evaluate Eq. (19.24) separately for $x = 0$ and 1 to verify that it is equivalent to Eqs. (19.21). There is nothing special about these environment states so we can write

$$|e\rangle\,|x\rangle \to (\,|d\rangle\mathbb{1} + |a\rangle X + |b\rangle(iY) + |c\rangle Z\,)\,|x\rangle. \tag{19.26}$$

Equation (19.25) applies to both $x = 0$ and $x = 1$. Since time evolution of the combined qubit-environment system follows quantum mechanics and so is unitary and linear, it also applies to a linear superposition $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ so

$$|e\rangle\,|\psi\rangle \to (\,|d\rangle\mathbb{1} + |a\rangle X + |b\rangle(iY) + |c\rangle Z\,)\,|\psi\rangle. \tag{19.27}$$

We see that the effects of the environment on the uncorrupted state of a single qubit can be expressed entirely in terms of the Pauli operators, $X, (iY)$ and Z. These are characterized as follows:

- X corresponds to a bit-flip error,
- Z corresponds to a phase-flip error, and
- $iY(= ZX)$ corresponds to combined bit-flip and phase-flip errors.

Intuitively, the reason that the new state can be expressed in terms of the Pauli operators and the identity, is that any 2×2 matrix can be written as a linear combination of these operators, see Eq. (2.25).

We remind the reader that the environment states are not normalized, and so, in the important case where the initial state is close to the final state, we have

$$\langle a|a\rangle \ll 1, \quad \langle b|b\rangle \ll 1, \quad \langle c|c\rangle \ll 1, \tag{19.28}$$

in Eq. (19.27),

We now extend this discussion to the situation where we have expanded a single qubit into an n-qubit codeword which we write as $|\psi\rangle_n$. In this course we just consider how to correct single-qubit errors, so we neglect the possibility that two or more of the qubits in the codeword are corrupted. From Eq. (19.27), we see that all single qubit errors are incorporated by

$$|e\rangle\,|\psi\rangle_n \to \left(|d\rangle\mathbb{1} + \sum_{k=1}^{n} |a_k\rangle X_k + \sum_{k=1}^{n} |b_k\rangle(iY_k) + \sum_{k=1}^{n} |c_k\rangle Z_k \right) |\psi\rangle_n. \tag{19.29}$$

Based on Eq. (19.29), single qubit quantum error correction involves the following steps:

- Expand the logical qubit to an n-qubit codeword.

[4] I prefer to write equations like (19.24) in terms of $iY(= ZX)$ rather than Y to avoid having explicitly complex coefficients in the matrices. Many texts on quantum computing write ZX rather than iY. Note that $iY\,(= ZX)$ is not Hermitian (though Y is) but we do not need the Hermitian property here. More importantly, iY, like X, Y and Z is unitary.

- Project the possibly corrupted state to *one* of the $3n+1$ states (syndromes) on the right hand side of Eq. (19.29), with information indicating *which* one.

- Correct, if necessary, the 1-qubit error by acting with the appropriate X_k, Y_k or Z_k.

Please note the following important points:

(i) The whole *continuum* of errors can be represented by a finite set of *discrete* errors. Errors emerge continuously from the uncorrupted state by increasing from zero the size of the terms in Eq. (19.29) involving X_i, Y_i and Z_i, which are characterized by $\langle a_i|a_i\rangle^{1/2}, \langle b_i|b_i\rangle^{1/2}$ and $\langle c_i|c_i\rangle^{1/2}$ respectively. However, the projection is always to one of the $3n+1$ discrete states. If the amplitude of the error is small then, with high probability, the projection will be to the uncorrupted state (which needs no correction) but with small but non-zero probability the projection will be to one of the $3n$ corrupted states (which do need correction).

(ii) An *arbitrary* error on a single qubit will be corrected, not just bit-flip (X), or phase-flip (Z), or combined bit- and phase-flip (iY) errors but also *any combination of them.* For example, suppose that the k-th qubit has been reinitialized to zero, i.e.

$$|0_k\rangle \to |0_k\rangle, \ |1_k\rangle \to |0_k\rangle. \tag{19.30}$$

The matrix which accomplishes this transformation is[5]

$$\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \tag{19.31}$$

which can be written as

$$\frac{\mathbb{1} + X_k + iY_k + Z_k}{2}. \tag{19.32}$$

Hence the state of the codeword qubits and environment has been transformed as follows:

$$|e\rangle\,|\psi\rangle_n \to |e'\rangle\,|\psi'\rangle_n = |e'\rangle\,\frac{1}{2}\,(\mathbb{1} + X_k + iY_k + Z_k)\,|\psi\rangle_n. \tag{19.33}$$

The codeword qubits are now in a linear combination of four syndromes, corresponding to the four terms in this equation. A general syndrome measuring circuit, such as the Shor 9-qubit code discussed in the next section, will detect these syndromes and obtain a *unique* set of values for the ancilla qubits for each of them. Hence, even for this non-unitary error, measuring the ancillas will project on to one of the syndromes which can then be corrected if necessary.

(iii) A full discussion of how the entanglement of qubits with the environment generates errors and how they can subsequently be corrected, requires a detailed treatment of the density matrix, see Chapter 5. This advanced material is discussed in Refs. [NC00, RP14] but is beyond the scope of the present course.

[5]The reader will notice that the transformation in Eqs. (19.31), which involves a *linear combination* of X, iY and Z on a single qubit, are not unitary. Now the evolution of an isolated (closed) system *is* unitary, However, qubits are coupled to the environment. If we consider a system coupled to the environment (called an open system), and subject the combined system+environment to a unitary transformation, and finally consider the behavior of just the system by tracing out over the environment, the resulting transformation of the system is not necessarily unitary [NC00, RP14].

19.6 Correcting Arbitrary Errors: the 9-qubit Shor code

In the section we discuss a code, due to Peter Shor [Sho95], for correcting arbitrary 1-qubit errors. This code needs code words of nine qubits to represent one logical qubit. It is not the most efficient code, there are others which use smaller code words and so don't need as many physical qubits, but the structure of Shor's code follows quite naturally from the discussion we have already given of 1-qubit bit-flip, and 1-qubit phase-flip errors, so will discuss it here.

Essentially it combines bit-flip (X) and phase-flip (Z) codes, which turns out to then automatically correct for combined bit-flip, phase-flip (iY) errors. As discussed in the previous section, it then also corrects *arbitrary* 1-qubit errors.

We first encode for phase flip errors:

$$|0\rangle \to |++\!+\rangle, \quad |1\rangle \to |--\!-\rangle, \tag{19.34}$$

and then encode for bit-flip errors

$$|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \to \frac{1}{\sqrt{2}}\left(|000\rangle + |111\rangle\right), \quad |-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \to \frac{1}{\sqrt{2}}\left(|000\rangle - |111\rangle\right). \tag{19.35}$$

The final result is the 9-qubit encoding

$$|0\rangle \to |\overline{0}\rangle = \frac{1}{2^{3/2}}\left(|000\rangle + |111\rangle\right)\left(|000\rangle + |111\rangle\right)\left(|000\rangle + |111\rangle\right), \tag{19.36a}$$

$$|1\rangle \to |\overline{1}\rangle = \frac{1}{2^{3/2}}\left(|000\rangle - |111\rangle\right)\left(|000\rangle - |111\rangle\right)\left(|000\rangle - |111\rangle\right). \tag{19.36b}$$

These two equations can be combined as

$$|x\rangle \to |\overline{x}\rangle = \frac{1}{2^{3/2}}\left(|000\rangle + (-1)^x|111\rangle\right)\left(|000\rangle + (-1)^x|111\rangle\right)\left(|000\rangle + (-1)^x|111\rangle\right), \tag{19.37}$$

or more concisely as

$$|\overline{x}\rangle = \frac{1}{2^{3/2}}\left(|000\rangle + (-1)^x|111\rangle\right)^{\otimes 3}. \tag{19.38}$$

Such a code is called a *concatenated* code. The circuit to achieve this encoding is obtained by concatenating the phase flip and the bit flip encodings as shown in Fig. 19.10. Note the labeling of the qubits. The qubits in each of the three blocks in Eq. (19.36) have labels $123, 456$ and 789.

The form of the 1-qubit corruption in Eq. (19.29) simplifies a little here because if $|\psi\rangle$ is a linear combination of the codeword states in Eq. (19.36) then

$$Z_1|\psi\rangle = Z_2|\psi\rangle = Z_3|\psi\rangle, \tag{19.39a}$$

$$Z_4|\psi\rangle = Z_5|\psi\rangle = Z_6|\psi\rangle, \tag{19.39b}$$

$$Z_7|\psi\rangle = Z_8|\psi\rangle = Z_9|\psi\rangle. \tag{19.39c}$$

The reason is that, for example, changing the first of the $+$ signs in Eq. (19.36a) into a $-$ sign, and the first $-$ sign in Eq. (19.36b) into a $+$ sign, can be accomplished by acting with either Z_1, Z_2 or Z_3.

Hence, the general form of a 1-qubit corruption contains only 22 independent syndromes rather than $28 = (3 \times 9) + 1$:

$$|e\rangle\,|\psi\rangle \to \left(|d\rangle I + |c\rangle Z_1 + |c'\rangle Z_4 + |c''\rangle Z_7 + \sum_{i=1}^{9}|a_i\rangle X_i + \sum_{i=1}^{9}|b_i\rangle iY_i\right)|\psi\rangle. \tag{19.40}$$

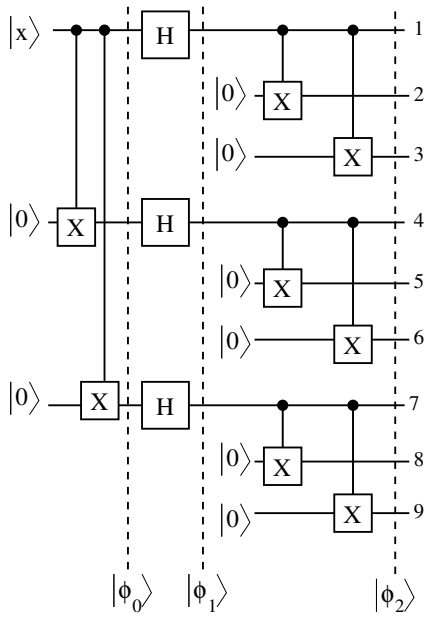


Figure 19.10: Encoding for the Shor 9-qubit code. The initial state at the top left, $|x\rangle$, is equal to $|0\rangle$ or $|1\rangle$ in the computational basis, so $\phi_0 = |xxx\rangle$ and $\phi_1 = 2^{-3/2}(|0\rangle + (-1)^x|1\rangle)(|0\rangle + (-1)^x|1\rangle)(|0\rangle + (-1)^x|1\rangle)$ since $H|x\rangle = 2^{-1/2}(|0\rangle + (-1)^x|1\rangle)$. We see by comparison with Fig. 19.1, that if $x = 0$ then the final state $|\phi_2\rangle$ is equal to $|\overline{0}\rangle$ given by Eq. (19.36a), while if $x = 1$ the final state is $|\overline{1}\rangle$ given by Eq. (19.36b). If the initial state at the top left is a linear combination $\alpha|0\rangle + \beta|1\rangle$ then, by linearity, the final state at the right is $\alpha|\overline{0}\rangle + \beta|\overline{1}\rangle$. The numbers at the right are the labels of the nine qubits. Note that this circuit is a concatenation of the encoding circuit for phase-flips shown in Fig. 19.9, and that for bit-flips in Fig. 19.1.

The eight stabilizers which we use to diagnose the error are

$$M_1 = Z_1Z_2, \quad M_2 = Z_2Z_3, \quad M_3 = Z_4Z_5, \quad M_4 = Z_5Z_6, \quad M_5 = Z_7Z_8, \quad M_6 = Z_8Z_9,$$
$$M_7 = X_1X_2X_3X_4X_5X_6, \quad M_8 = X_4X_5X_6X_7X_8X_9\,. \tag{19.41}$$

Note that the nine qubits can conveniently be grouped into three blocks of three, containing qubits $123, 456$ and 789 respectively. M_1 and M_2 act entirely on the first block, and do so in the same way as the stabilizers of the 3-qubit, bit flip code shown in Fig. 19.7. Similarly M_3 and M_4 act on the second block and M_5 and M_6 act on the third block. M_7 acts on all qubits in blocks 1 and 2, while M_7 acts on all qubit in blocks 2 and 3.

The circuit for determining the syndrome eigenvalues is shown in Fig. 19.11.

We will now show that the M_i have the desired properties:

- They all square to unity (since each of the Z's and X's square to unity and the X's commute amongst each other as do the Z's). Hence their eigenvalues are ± 1.

- They mutually commute. The six Z-stabilizers trivially commute with each other as do the two X-stabilizers. Comparing the indices on the Z-stabilizers with the X-stabilizers one sees that either they have none in common, in which case this X-stabilizer and Z-stabilizer trivially commute, or they have two in common, in which case there are two minus signs when one pulls one of the stabilizers through the other, so the overall sign is positive and again the X-stabilizer and the Z-stabilizer commute).

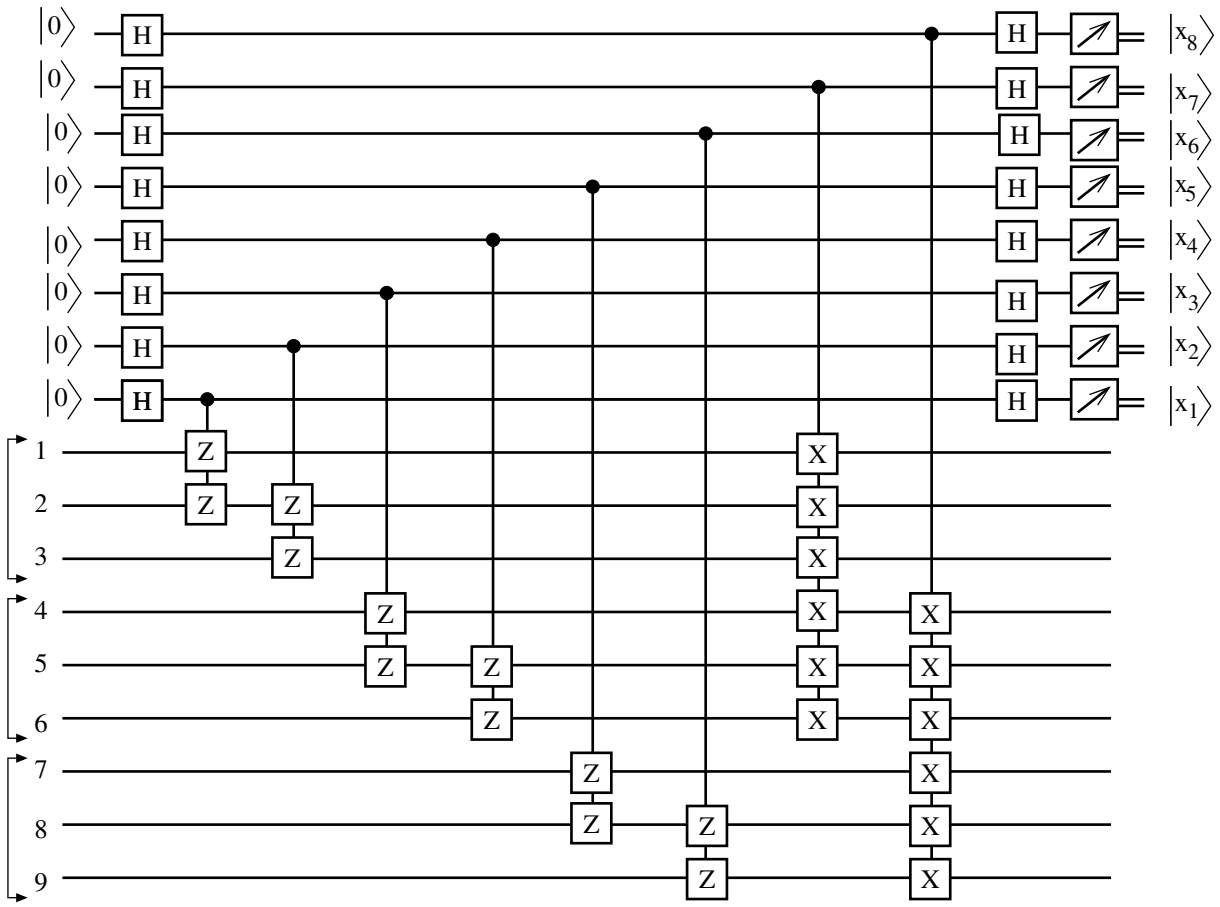


Figure 19.11: A circuit to measure the error syndrome for the Shor 9-qubit code. The nine codeword qubits are at the bottom and the eight ancillary qubits at the top. The ancillary qubits determine the values of the eight, mutually commuting stabilizers in Eq. (19.41), $M_1 = Z_1Z_2, M_2 = Z_2Z_3, M_3 = Z_4Z_5, M_4 = Z_5Z_6, M_5 = Z_7Z_8, M_6 = Z_8Z_9, M_7 = X_1X_2X_3X_4X_5X_6$ and $M_8 = X_4X_5X_6X_7X_8X_9$. The nine codeword qubits can be conveniently grouped into three groups of three as indicated. The measured value of the i-th ancilla x_i ($= 0$ or 1), is related to the value of the corresponding stabilizer M_i by $M_i = (-1)^{x_i}$. The values of the M_i determine which syndrome in Eq. (19.40) is projected out, as discussed in the text and Table 19.3. If a corrupted syndrome is found, it can be corrected back to the uncorrupted state by acting with the appropriate X_i, Y_i or Z_i.

- The eigenvalue of the uncorrupted codewords $|\overline{0}\rangle$ and $|\overline{1}\rangle$ is $+1$ for all stabilizers.

  This is trivially seen for M_1–M_6 which involve pairs of Z operators, since, for each pair, both qubits are 0 or both are 1 in the codewords. Note that the pairs are entirely within the blocks of three adjacent qubits in Eq. (19.36), see Fig. 19.10.

  Next consider M_7 and M_8 which involve a product of six X operators, each spanning two of the three blocks shown in Fig. 19.10. For example, M_7 is a product of the X operators for the qubits

in the first two blocks. We have

$$\begin{aligned} M_7|\overline{0}\rangle = X_1X_2X_3X_4X_5X_6\frac{1}{2^{3/2}}\left(|000\rangle+|111\rangle\right)\left(|000\rangle+|111\rangle\right)\left(|000\rangle+|111\rangle\right) \\ =\frac{1}{2^{3/2}}\left(|111\rangle+|000\rangle\right)\left(|111\rangle+|000\rangle\right)\left(|000\rangle+|111\rangle\right) \\ =\frac{1}{2^{3/2}}\left(|000\rangle+|111\rangle\right)\left(|000\rangle+|111\rangle\right)\left(|000\rangle+|111\rangle\right) \\ =|\overline{0}\rangle, \end{aligned} \tag{19.42}$$

and

$$\begin{aligned} M_7|\overline{1}\rangle = X_1X_2X_3X_4X_5X_6\frac{1}{2^{3/2}}\left(|000\rangle-|111\rangle\right)\left(|000\rangle-|111\rangle\right)\left(|000\rangle-|111\rangle\right) \\ =\frac{1}{2^{3/2}}\left(|111\rangle-|000\rangle\right)\left(|111\rangle-|000\rangle\right)\left(|000\rangle-|111\rangle\right) \\ =\frac{1}{2^{3/2}}\left(|000\rangle-|111\rangle\right)\left(|000\rangle-|111\rangle\right)\left(|000\rangle-|111\rangle\right) \\ =|\overline{1}\rangle, \end{aligned} \tag{19.43}$$

so M_7 has eigenvalue $+1$ for both uncorrupted codewords. The argument for M_8 goes along the same lines.

- The ±1 eigenvalues of the stabilizers allow one to determine which of the 22 syndromes in Eq. (19.40) the system has projected on to. Recalling the discussion in Sec. 19.3, the eigenvalue is $+1$ if the stabilizer commutes with the operator which caused the 1-qubit corruption, and is -1 if it anti-commutes. Each time two different operators on the same qubit are pulled through each other to perform the commutation one generates a minus sign. The operators which generate the corruption are the 21 X_i, Y_i and Z_i in Eq. (19.40). A table of the eigenvalues of the stabilizers for all 22 syndromes is given in Table 19.3.

Let's make sure that we understand how the syndrome-detection circuit in Fig. 19.11 works. Firstly we remind the reader that if the measurement of an auxiliary qubit, x_i say, is 0, then the value of the corresponding stabilizer M_i is $+1$, while if the measurement is 1, then the value of M_i is -1. Thus we can say that x_i measures M_i, see the discussion of Fig. 19.6 on page 169. Next we discuss how each of the stabilizers works.

- We consider first M_1–M_6, the stabilizers involving Z operators.
  The ancilla qubits x_1 and x_2 measure $M_1 = Z_1Z_2$ and $M_2 = Z_2Z_3$ respectively, and so detect a bit-flip error in the first group of three qubits in the 9-qubit encoding of Eq. (19.36), in exactly the same way as for the 3-qubit, bit-flip code shown in Fig. 19.7. Similarly x_3 and x_4 detect a bit-flip error in the second group of three qubits (qubits 4–6), and x_5 and x_6 detect a bit-flip error in the third group of three qubits (qubits 7–9).

- Next we consider M_7 and M_8, the stabilizers involving X operators.
  The ancilla x_7 measures $M_7 = X_1X_2X_3X_4X_5X_6$ and the ancilla x_8 measures $M_8 = X_4X_5X_6X_7X_8X_9$. These detect phase flips. M_7 acts on the first two groups of three qubits (qubits 1–6) while M_8 acts on the second and third groups of three qubits (qubits 4–9).

  Let's suppose that there is a phase flip in one of the qubits in the first group (it doesn't matter which one; the resulting state is the same). In other words

$$|\psi\rangle = \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle \tag{19.44}$$

Syndrome	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
$\mathbb{1}$	+	+	+	+	+	+	+	+
X_1	−	+	+	+	+	+	+	+
X_2	−	−	+	+	+	+	+	+
X_3	+	−	+	+	+	+	+	+
X_4	+	+	−	+	+	+	+	+
X_5	+	+	−	−	+	+	+	+
X_6	+	+	+	−	+	+	+	+
X_7	+	+	+	+	−	+	+	+
X_8	+	+	+	+	−	−	+	+
X_9	+	+	+	+	+	−	+	+
Y_1	−	+	+	+	+	+	−	+
Y_2	−	−	+	+	+	+	−	+
Y_3	+	−	+	+	+	+	−	+
Y_4	+	+	−	+	+	+	−	−
Y_5	+	+	−	−	+	+	−	−
Y_6	+	+	+	−	+	+	−	−
Y_7	+	+	+	+	−	+	+	−
Y_8	+	+	+	+	−	−	+	−
Y_9	+	+	+	+	+	−	+	−
$Z_1 (= Z_2 = Z_3)$	+	+	+	+	+	+	−	+
$Z_4 (= Z_5 = Z_6)$	+	+	+	+	+	+	−	−
$Z_7 (= Z_8 = Z_9)$	+	+	+	+	+	+	+	−

Table 19.3: The eigenvalues of the 8 stabilizers defined in Eq. (19.41) for the 22 syndromes of Shor's 9-qubit error correcting code. The left column indicates the operator which generates for syndrome from the uncorrupted state. A + sign indicates eigenvalue +1 and a − sign indicates eigenvalue −1. Each stabilizer M_i is measured by an ancilla qubit x_i, see Fig. 19.11, such that if $M_i = +1$ then $x_i = 0$ and if $M_i = -1$ then $x_i = 1$. An essential feature is that each of the 22 rows, i.e. syndromes, has a unique pattern of + and − signs.

has been transformed to

$$\begin{aligned} |\psi'\rangle = &\frac{\alpha}{2^{3/2}} \left(|000\rangle - |111\rangle \right) \left(|000\rangle + |111\rangle \right) \left(|000\rangle + |111\rangle \right) + \\ &\frac{\beta}{2^{3/2}} \left(|000\rangle + |111\rangle \right) \left(|000\rangle - |111\rangle \right) \left(|000\rangle - |111\rangle \right). \end{aligned} \tag{19.45}$$

Acting with the product of the three X_i in a group has the effect

$$\begin{aligned} &|000\rangle + |111\rangle \to |111\rangle + |000\rangle = |000\rangle + |111\rangle \\ &|000\rangle - |111\rangle \to |111\rangle - |000\rangle = -\left(|000\rangle - |111\rangle \right). \end{aligned} \tag{19.46}$$

Recalling that we are considering here a phase flip in the first group, on which M_7 acts but M_8 does not, it follows that

$$\begin{aligned} M_7|\psi'\rangle &= X_1X_2X_3X_4X_5X_6|\psi'\rangle = -|\psi'\rangle \\ M_8|\psi'\rangle &= X_4X_5X_6X_7X_8X_9|\psi'\rangle = |\psi'\rangle. \end{aligned} \tag{19.47}$$

Hence M_7 has eigenvalue −1 and M_8 has eigenvalue +1. These values are shown in Table 19.3.

Similarly, if the phase-flip is in the second group, both M_7 and M_8 have eigenvalue -1 whereas if phase-flip is in the third group, M_7 has eigenvalue $+1$ and M_8 has eigenvalue -1. These results are also shown in Table 19.3.

We now illustrate in more detail how Table 19.3 was obtained by working through a few cases. (Eigenvalues are taken to be $+1$ unless otherwise stated.)

(a) $\boldsymbol{Z_2}$: Clearly Z_2 commutes with all the Z-stabilizers. It anticommutes with M_7 (because it has one qubit in common and X and Z anticommute) and commutes with M_8 because it has no qubits in common. Hence M_7 has eigenvalue -1 while all other stabilizers have eigenvalue $+1$.

(b) $\boldsymbol{Z_4}$: Both M_7 and M_8 have eigenvalue -1 since they have one qubit in common with Z_4 (and X and Z anticommute).

(c) $\boldsymbol{X_4}$: Clearly X_4 commutes with both X-stabilizers. It anticommutes with M_3 because it has one qubit in common (and Z and X anticommute). Hence M_3 has eigenvalue -1.

(d) $\boldsymbol{Y_5}$: We note that Y anticommutes with both X and Z so we have to consider all the stabilizers. Y_5 has a qubit in common with M_3, M_4, M_7 and M_8 so these stabilizers have eigenvalue -1.

Table 19.3 shows that each syndrome gives rise to a unique set of $+1$ and -1 eigenvalues of the stabilizers as required. Thus, measuring the eigenvalues of the eight stabilizers in Eq. (19.41) projects the corrupted state on to one of the 22 syndromes in Eq. (19.40), and the set of eigenvalues determines which one it is. One then applies an appropriate unitary transformation to correct the state if necessary. Note that the Shor code is *explicitly* designed to detect and correct bit-flip (X) and phase-flip (Z) errors, but then *automatically* detects and corrects combined bit-flip and phase-flip ($ZX \equiv iY$) errors.

Not only that, it also corrects *arbitrary* errors on a single qubit, which, as discussed in Sec. 19.5, can be expressed as *linear combinations* of bit-flip, phase-flip, and combined bit- and phase-flip errors. As an example consider the situation mentioned in Eq. (19.33) in Sec. 19.5 in which a qubit has been reset to $|0\rangle$. This is an example of a *non-unitary*[6] operation on the qubit. Let's take it to be qubit 1 and indicate the codeword qubits by putting the first on the left, the last on the right (we will use the same ordering below for the ancilla qubits). In other words

$$|\psi\rangle = \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle \tag{19.48}$$

has been transformed to

$$\begin{aligned} |\psi'\rangle =& \frac{\alpha}{2^{3/2}} \left(|000\rangle + |011\rangle \right) \left(|000\rangle + |111\rangle \right) \left(|000\rangle + |111\rangle \right) + \\ & \frac{\beta}{2^{3/2}} \left(|000\rangle - |011\rangle \right) \left(|000\rangle - |111\rangle \right) \left(|000\rangle - |111\rangle \right). \end{aligned} \tag{19.49}$$

According to Eq. (19.33) this can be written as

$$|\psi'\rangle = \frac{1}{2} \left(\mathbb{1} + X_1 + iY_1 + Z_1 \right) |\psi\rangle, \tag{19.50}$$

[6] In footnote 5 we noted that, while a transformation of the combined system+environment *is* unitary, if the system is coupled to the environment, then a unitary operation applied to system+environment followed by a trace over the environment leaves the system in a new state which is not, in general, related by a unitary transformation to its initial state.

where

$$|\psi\rangle = \alpha\,(\quad |000\rangle + |111\rangle\,)\,(\cdots)_+\,(\cdots)_+ \;+\beta\,(\quad |000\rangle - |111\rangle\,)\,(\cdots)_-\,(\cdots)_- \tag{19.51a}$$
$$X_1|\psi\rangle = \alpha\,(\quad |100\rangle + |011\rangle\,)\,(\cdots)_+\,(\cdots)_+ \;+\beta\,(\quad |100\rangle - |011\rangle\,)\,(\cdots)_-\,(\cdots)_- \tag{19.51b}$$
$$iY_1|\psi\rangle = \alpha\,(-|100\rangle + |011\rangle\,)\,(\cdots)_+\,(\cdots)_+ \;+\beta\,(-|100\rangle - |011\rangle\,)\,(\cdots)_-\,(\cdots)_- \tag{19.51c}$$
$$Z_1|\psi\rangle = \alpha\,(\quad |000\rangle - |111\rangle\,)\,(\cdots)_+\,(\cdots)_+ \;+\beta\,(\quad |000\rangle + |111\rangle\,)\,(\cdots)_-\,(\cdots)_-\,, \tag{19.51d}$$

in which

$$\begin{aligned}(\cdots)_+ &\equiv (\,|000\rangle + |111\rangle\,)\\ (\cdots)_- &\equiv (\,|000\rangle - |111\rangle\,)\,.\end{aligned} \tag{19.52}$$

One can verify that adding Eqs. (19.51) (and dividing by 2 according to Eq. (19.50)) does indeed give Eq. (19.49).

Equation (19.50) is the input to the syndrome measurement circuit. According to Table 19.3, after the syndrome measurement circuit in Fig. 19.10 has acted, the state of the system is

$$\frac{1}{2}\,[\,|\psi\rangle\,|00000000\rangle_A + (X_1|\psi\rangle)\,|10000000\rangle_A + (iY_1|\psi\rangle)\,|10000010\rangle_A + (Z_1|\psi\rangle)\,|00000010\rangle_A\,]\,, \tag{19.53}$$

where $|\cdots\rangle_A$ denotes the ancillas, which are ordered from 1 on the left to 8 on the right. Measuring the ancillas will project the computational qubits on to one of the four syndromes, $|\psi\rangle, X_1|\psi\rangle, iY_1|\psi\rangle, Z_1|\psi\rangle$. Since the measurements of the ancillas tell us which syndrome the state has been projected on to, the computational qubits can then be corrected if necessary.

Thus, Shor's 9-qubit code, and other codes designed to correct both bit-flip and phase-flip errors, actually correct *arbitrary* 1-qubit errors. I find this amazing.

19.7 Other error-correcting codes

The Shor code uses nine physical qubits to encode one logical qubit. What is the minimum number of physical qubits needed to correct all 1-qubit errors? If we encode using n qubits the dimension of the space of states is 2^n. This must be sufficient to contain $3n+1$ mutually orthogonal 2-d subspaces for the syndromes (the 1 is for the uncorrupted state and there are n possible corruptions with each of the X, iY or Z operators). Hence we need

$$2^n \geq 2(3n+1)\,, \tag{19.54}$$

so the smallest value is $n = 5$ which satisfies this condition as an equality.

There *is* a 5-qubit code, but it turns out to be difficult to construct the necessary gates. A more popular choice is a 7-qubit code due to Steane [Ste96]. The Shor code, which has 9-qubit codewords, is now mainly of pedagogical interest.

19.7.1 The 5-qubit code

We now state, without much discussion, the codewords and stabilizers for the 5-qubit code. Further details are in Mermin [Mer07].

For the 5-qubit code we have $(3 \times 5) + 1 = 16$ mutually orthogonal, two-dimensional subspaces, i.e. 16 syndromes. There are four stabilizers and, since they each have two eigenvalues (± 1), the

number of distinct sets of eigenvalues is $2^4 = 16$ which is just enough to distinguish the syndromes. These stabilizers are

$$M_1 = Z_2X_3X_4Z_5, \tag{19.55a}$$
$$M_2 = Z_3X_4X_5Z_1, \tag{19.55b}$$
$$M_3 = Z_4X_5X_1Z_2, \tag{19.55c}$$
$$M_4 = Z_5X_1X_2Z_3. \tag{19.55d}$$

The circuit to measure the M_i is shown in Fig. 19.12.

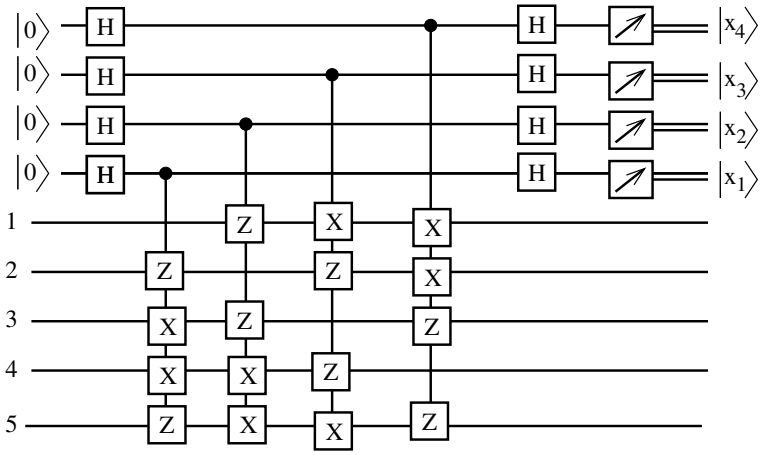


Figure 19.12: A circuit to measure the error syndrome for the 5-qubit code. The five codeword qubits are at the bottom and the four ancillary qubits at the top. The ancillary qubits determine the values of the four, mutually commuting stabilizers in Eq. (19.55), $M_1 = Z_2X_3X_4Z_5, M_2 = Z_3X_4X_5Z_1, M_3 = Z_4X_5X_1Z_2, M_4 = Z_5X_1X_2Z_3$.

The 5-qubit codewords are most conveniently expressed in terms of the M_i:

$$|\overline{0}\rangle = \frac{1}{4}(1+M_1)(1+M_2)(1+M_3)(1+M_4)|00000\rangle, \tag{19.56a}$$
$$|\overline{1}\rangle = \frac{1}{4}(1+M_1)(1+M_2)(1+M_3)(1+M_4)|11111\rangle. \tag{19.56b}$$

Note that $|\overline{0}\rangle$ is composed of the 16 basis states with an even number of 1's, while $|\overline{1}\rangle$ is composed of the 16 basis states with an odd number of 1's, so the two codewords are orthogonal. It is not completely trivial to generate these codewords, see Mermin [Mer07] for details.

Furthermore the M_i square to unity, are mutually commuting and each has eigenvalue $+1$ for the uncorrupted codewords in Eq. (19.56). Each of them commutes or anti-commutes with the X_i, Y_i and Z_i error operators, so the 15 corrupted syndromes and the uncorrupted state are distinguished by the set of ± 1 eigenvalues of the M's, as shown in Table 19.4.

19.7.2 The Steane 7-qubit code

Next I describe briefly the 7-qubit Steane code.

There are 6 stabilizers which are

$$\begin{aligned} M_1 &= X_1X_5X_6X_7, \qquad & N_1 &= Z_1Z_5Z_6Z_7, \\ M_2 &= X_2X_4X_6X_7, \qquad & N_2 &= Z_2Z_4Z_6Z_7, \\ M_3 &= X_3X_4X_5X_7, \qquad & N_3 &= Z_3Z_4Z_5Z_7. \end{aligned} \tag{19.57}$$

Syndrome	$M_1 = Z_2X_3X_4Z_5$	$M_2 = Z_3X_4X_5Z_1$	$M_3 = Z_4X_5X_1Z_2$	$M_4 = Z_5X_1X_2Z_3$
$\mathbb{1}$	+	+	+	+
X_1	+	−	+	+
X_2	−	+	−	+
X_3	+	−	+	−
X_4	+	+	−	−
X_5	−	+	+	−
Y_1	+	−	−	−
Y_2	−	+	−	−
Y_3	−	−	+	−
Y_4	−	−	−	+
Y_5	−	−	−	−
Z_1	+	+	−	−
Z_2	+	+	+	−
Z_3	−	+	+	+
Z_4	−	−	+	+
Z_5	+	−	−	+

Table 19.4: The table shows whether the four stabilizers M_i for the 5-qubit error correcting code commute (+) or anti-commute (−) with the 15 operators X_i, Y_i and Z_i, $i = 1, 2, \cdots, 5$ (which generate a corruption of the codeword) as well as with the identity. Each of the 16 rows has a unique pattern of + and − signs. A + sign corresponds to an eigenvalue +1 while a − sign indicates an eigenvalue −1.

The circuit to detect errors is shown in Fig. 19.13. The 7-qubit codewords are given by

$$|\overline{0}\rangle = \frac{1}{\sqrt{8}}(1+M_1)(1+M_2)(1+M_3)|0\rangle_7,$$
$$|\overline{1}\rangle = \frac{1}{\sqrt{8}}(1+M_1)(1+M_2)(1+M_3)\overline{X}|0\rangle_7, \tag{19.58}$$

where

$$\overline{X} = X_1X_2X_3X_4X_5X_6X_7, \tag{19.59}$$

so

$$|1111111\rangle = \overline{X}|0000000\rangle. \tag{19.60}$$

It is instructive for the student to show the following:

(a) The stabilizers mutually commute and square to the identity.

(b) The two states in Eq. (19.58) are orthogonal.

(c) The two states in Eq. (19.58) are normalized.
Hint: You will need to use that the M_i square to the identity, as does $\overline{X}$, and that $\overline{X}$ commutes with the M_i.

(d) The codewords $|\overline{0}\rangle$ and $|\overline{1}\rangle$ are eigenstates of each of the stabilizers with eigenvalue +1.
Hint: Note that $M_i(1+M_i) = 1+M_i$ (why?), that the N_j commute with $\overline{X}$ (explain why), and that $|0\rangle_7$ is an eigenstate of the N_i with eigenvalue 1.

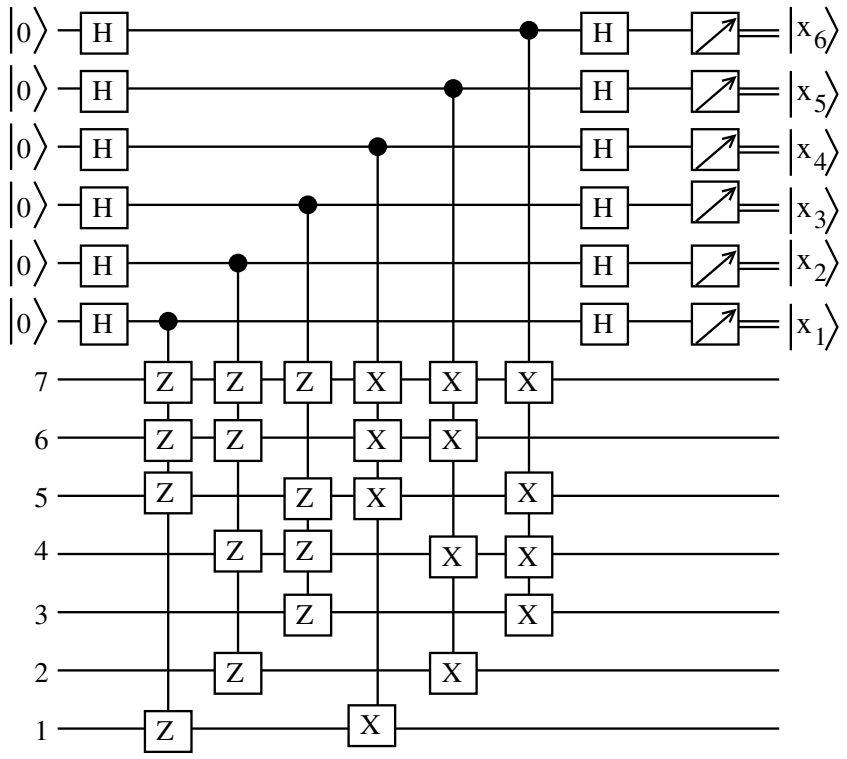


Figure 19.13: The circuit of Steane's 7-qubit code to detect errors in the computational qubits, (labeled 1–7 in the figure). There are also six ancilla qubits (at the top) each of which is associated with one of the stabilizers as follows: N_1-N_3 correspond to x_1-x_3 respectively, and M_1-M_3 correspond to x_4-x_6 respectively, in the usual way, e.g. $N_1 = (-1)^{x_1}, M_1 = (-1)^{x_4}$.

19.7.3 Surface Codes

A different approach to quantum error correction, but one that seems the most promising, is to use "surface codes" in which the physical qubits are arranged in a square array and the values of the logical qubits are encoded in complicated entangled states of the square array. Unfortunately, I have not been able to find a simple introduction to this topic.

19.8 Fault Tolerant Quantum Computing

So far we have assumed that an error has occurred in some way and that we can correct it by *perfect* gates which do not introduce any further errors. This is, of course unreasonable since all aspects of quantum computing can introduce errors: acting with gates, measurements, or simply waiting. Looking at the number of gates for Shor's 9-qubit syndrome-detection code in Fig. 19.11 we might imagine that this circuit could introduce more errors than it corrects. Of particular importance is that a circuit does not spread an error initially in one qubit into multiple qubits which would then be much harder to correct. A circuit which does not spread errors is said to be "fault tolerant".

An important result in quantum error correction is the "threshold theorem" which states that if the intrinsic error rate in an individual gate in a fault tolerant circuit is less than a critical value p_c then the overall error rate in the circuit can be reduced to arbitrary low levels by quantum error correction. This means that errors are being corrected faster than they are being generated. However, since error correction requires duplication, getting the error rate down to an acceptable level will require that the

number of physical qubits is much greater than the number of logical qubits (those that appear in the algorithm).

To see how one might reduce errors to an arbitrarily low level suppose that the intrinsic error rate is p and we have a fault tolerant error correction scheme which corrects 1-qubit errors. This means that the error rate after error correction is[7] cp^2 for some constant c. If $pc < 1$ then we have decreased the errors, so the threshold error rate is $p_c = 1/c$.

How can we go decrease the errors further? Suppose the error correction procedure requires n physical qubits for each logical qubit, so, for example, $n = 9$ for the Shor code and $n = 7$ for the 7-qubit Steane code. We can then take each of the n qubits and error correct these with the same code. This procedure is known as *concatenation*. We then have n^2 physical qubits and the error rate is $c(cp^2)^2 = c^{-1}(cp)^{2^2}$. Generalizing, if we concatenate l times, then the number of qubits is n^l while the resulting error rate is $c^{-1}(cp)^{2^l}$. Note that while the number of qubits increases exponentially with the level of concatenation l, the error rate decreases *doubly* exponentially with l. As an example, to get a feel for what this means, consider the case $p = 1/8, c = 2$, so $cp = 1/4$ and also suppose that $n = 7$ (corresponding to the Steane code). Then successive concatenations give the numbers in Table 19.5.

no. of concatenations (l)	error rate (formula)	error rate (numeric)	no. of qubits
0	p	$1/2^3 = 0.125$	1
1	$cp^2 = c^{-1}(cp)^2$	$1/2^5 = 0.03125$	n $(= 7)$
2	$c(cp^2)^2 = c^{-1}(cp)^{2^2}$	$1/2^9 = 1.953 \times 10^{-3}$	n^2 $(= 49)$
3	$c((cp^2)^2)^2 = c^{-1}(cp)^{2^3}$	$1/2^{17} = 7.629 \times 10^{-6}$	n^3 $(= 343)$
4	$c(c((cp^2)^2)^2)^2 = c^{-1}(cp)^{2^4}$	$1/2^{33} = 1.164 \times 10^{-10}$	n^4 $(= 2401)$
5	$c(c(c((cp^2)^2)^2)^2)^2 = c^{-1}(cp)^{2^5}$	$1/2^{65} = 2.711 \times 10^{-20}$	n^5 $(= 16807)$

Table 19.5: Parameters for the concatenation of a fault tolerant circuit with an (artificial) choice of parameters discussed in the text.

These numbers are not realistic. They correspond to a threshold value of $p_c = 1/c = 1/2$ and any realistic circuit would have a much smaller value. However, they do show, and this is the main point, that the error rate goes down much faster than the number of physical qubits goes up. Of course, the number of physical qubits per logical qubit will still have to be very large to get the error rate down to an acceptable value for computation.

Various calculations have estimated the threshold for 7-qubit Steane code at around 10^{-5}. To perform error correction one would need individual circuit elements with an error rate significantly less than this, which, to my knowledge, is not feasible at present. Surface codes, which were briefly mentioned above, are estimated to have a higher threshold, of around 10^{-2}, and it does seem feasible to make gates with a lower error rate than this. For example, at the end of a very long and technical paper, Ref. [FMMC12] estimates that to factor, using Shor's algorithm, an integer which is too large to be factored on a classical computer (2000 bits), would require no less than around 220×10^6 qubits with then state-of-the-art superconducting qubits using quantum error correction with surface codes. At present, quantum computers (using the "gate" model of quantum computing which is the topic of this course) have at most a few tens of qubits, so a huge increase in scale will be required. However, who is to say that this cannot happen in a few decades? An example of a comparable increase in scale which has *already* happened is the number of transistors on a modern chip compared with the number on early integrated circuits.

[7] The crucial point is that the new error rate is proportional to the *square* of the old error rate. I don't think it's obvious that one can design a circuit with this property, but a detailed study indicates that one can [NC00, RP14]. Unfortunately, I have not been able to find a simple explanation of this result.

Thus, in my view, in the next few years, we may see quantum computers with a modest number of logical qubits which perform error correction. However, quantum computers with error correction *having enough logical qubits to outperform classical computers* for some *useful* problem such as integer factorization are for the distant future, if ever.

I thank Eleanor Rieffel for a helpful email exchange on quantum error correction.

Chapter 20

Grover's Search Algorithm

20.1 Introduction

Grover's algorithm discussed in this chapter is of a different type from Shor's algorithm. Whereas Shor's (and related algorithms like Simon's) depend on a quantum Fourier transform (of some sort), Grover's algorithm involves a different approach, *amplitude amplification.*

To motivate Grover's algorithm consider looking up someone in a phone directory. It is straightforward to lookup a person's phone number in a directory if one is given the name, because names are in alphabetic order. To locate the name systematically one would go to the midpoint of the list, see which half the name is in, divide that half in two, again see which half the number is in, and so on. One continues this procedure until the size of the region containing the desired entry is just one. For a directory with N entries, this *bisection* method takes $\log_2 N$ operations (rounded up to the nearest integer if N is not a power of 2) since one halves the range over which the special entry could be at each stage.

By contrast, suppose one is given the number and asked which person has that number. Since the numbers are not ordered, all one can do is go through the entries one at a time and see if each one has the desired name. On average this would take $N/2$ operations before success was achieved.

If N is large this is a huge difference. For example if $N = 10^6$ then $\log_2 N \simeq 20$, to be compared with $N/2 = 5 \times 10^5$. Note that if the N possible values are represented by the configurations of n qubits then

$$N = 2^n. \tag{20.1}$$

The quantum search algorithm algorithm discussed here, due to Grover, is often presented as such a search of an unstructured database.[1] Grover's algorithm requires a quantum computer running a subroutine for which the input is a number corresponding to an entry in the database, and which performs a test to see if this is the special value being searched for. For large N it will determine the special value, with probability close to 1, by calling the subroutine only $(\pi/4)\sqrt{N}$ times. This is a *quadratic* speedup compared with a classical computer. While less spectacular than the exponential speedup of Shor's algorithm, it can potentially be applied to a wide variety of problems[2].

[1] Though it is doubtful it would ever be used in this way since it would be a very extravagant use of a precious resource to use qubits to store classical information.

[2] However, most applications of practical interest have some structure, whereas Grover is designed for problems with no structure. In most cases that Grover could potentially be applied, the structure of the problem allows an efficient classical algorithm which outperforms Grover. Thus it is debated whether the Grover algorithm would be of practical utility, even if one could overcome the severe experimental difficulties of building a large quantum computer.

20.2 The Black Box (Oracle)

To formulate the problem we consider n-bit integers, one of which, a, is special. The goal is to find a. We need a subroutine which outputs 1 if the input value x is equal to a and outputs 0 otherwise, i.e.

$$\begin{aligned} f(x) &= 0, \quad (x \neq a), \\ f(a) &= 1\,. \end{aligned} \tag{20.2}$$

As usual, the function will be determined from a unitary transformation acting on an n-qubit "input" register and a 1-qubit "output" register which is flipped or not flipped depending on whether x is the special number a or not:

$$U|x\rangle_n|y\rangle_1 = |x\rangle_n|y \oplus f(x)\rangle_1\,. \tag{20.3}$$

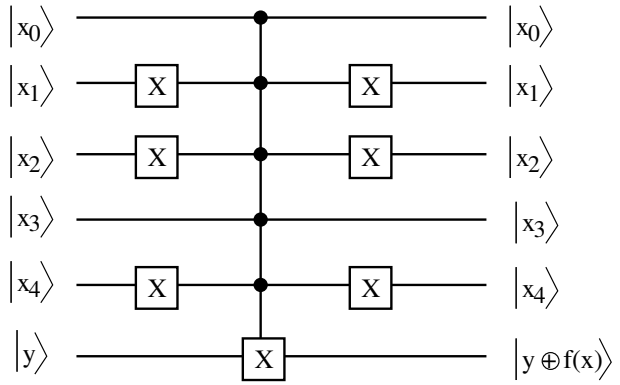


Figure 20.1: A black box circuit that executes the first part of a Grover iteration, Eq. (20.3), in which $f(x) = 0$ if $x \neq a$ and $f(a) = 1$, for the case of $n = 5$ qubits and where the special number a is 01001. The 6-qubit gate in the center is a five-fold-controlled-NOT gate which acts to flip the target qubit y only if all the control qubits are 1. The X gates on the left flip qubits x_1, x_2 and x_4. Hence the target qubit is flipped if and only if $x_0 = 1, x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 0$, which are the bits of a. The X-gates on the right flip back those qubits which had previously been flipped, thus leaving the "input" register, the $\{|x_i\rangle\}$, unchanged. The lower "output" qubit, which is initialized to $|y\rangle$, contains information on the function $f(x)$ in its final state.

A simple example of such a function for $n = 5$ and $a = 01001$ is shown in Fig. 20.1. Recall that x_0 is the least significant (i.e. right-hand) bit. The target bit is flipped only if all five of the control bits are one, which requires $x_0 = 1, x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 0$ (the bits of a). How to construct such a five-fold-controlled-NOT gate out of 1-qubit and 2-qubit elementary gates is discussed in Mermin [Mer07] §4.2.

Such a black box function is often called an oracle. The oracle can give a yes or no answer as to whether the input is the special number. For the implementation in Fig. 20.1 you might object and say that surely we *already* know the answer since it is built into the quantum device by placing the X-gates only on those qubits where the special number has a 0 bit. Can't we just open up the black box and look? In this case the answer is "yes". However, the implementation of the black box in Fig. 20.1 is just a simple example. The Grover algorithm can also be applied in more useful situations where the value of $f(x)$ is *not* built in explicitly but has to be calculated in a non-trivial way. Examples are discussed in Mermin [Mer07] and Nielsen and Chuang [NC00].

It is useful to initially set the "output" bit y to be 1 and then apply a Hadamard gate before applying U. The "output" bit is then

$$H|1\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle \right) . \tag{20.4}$$

If the result of U is $f(x) = 0$ then the "output" bit is unchanged. If the result is $f(x) = 1$ then $|0\rangle \to |1\rangle$ and vice-versa, so the "output" bit changes sign. Consequently

$$U \left(|x\rangle_n \otimes H|1\rangle_1 \right) = (-1)^{f(x)} |x\rangle_n \otimes H|1\rangle_1 . \tag{20.5}$$

We can associate the possible sign change with the "input" register" in which case the "output" bit remains unchanged. Hence, for simplicity, the "output" bit will be ignored in what follows. Thus we consider the following unitary operator $\hat{O}$ acting only on the n-qubit "input" register[3]:

$$\hat{O}|x\rangle = (-1)^{f(x)}|x\rangle = \begin{cases} |x\rangle, & x \neq a, \\ -|a\rangle, & x = a. \end{cases} \tag{20.6}$$

Since U, and hence $\hat{O}$, are linear, acting with $\hat{O}$ on a superposition changes the sign of the component along $|a\rangle$ but leaves the component perpendicular to $|a\rangle$ unchanged. Hence if

$$|\psi\rangle = \sum_x c_x |x\rangle, \tag{20.7}$$

then

$$|\psi'\rangle \equiv \hat{O}|\psi\rangle = \sum_{x \neq a} c_x |x\rangle - c_a |a\rangle = \sum_x c_x |x\rangle - 2c_a |a\rangle = |\psi\rangle - 2|a\rangle\langle a|\psi\rangle \tag{20.8}$$

since $c_a = \langle a|\psi\rangle$. You should check that $\langle a|\psi'\rangle = -\langle a|\psi\rangle \, (= -c_a)$ and, for $x \neq a$, that $\langle x|\psi'\rangle = \langle x|\psi\rangle \, (= c_x)$, as required. You should also verify that $|\psi'\rangle$ is correctly normalized if $|\psi\rangle$ and $|a\rangle$ are.

We initialize the n-qubit input register into a uniform superposition of all basis states by acting with n Hadamards on $|0\rangle$:

$$|\psi_0\rangle = H^{\otimes n}|0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle . \tag{20.9}$$

We can also write $|\psi_0\rangle$ as

$$|\psi_0\rangle = \frac{1}{\sqrt{N}}|a\rangle + \sqrt{\frac{N-1}{N}}|a_\perp\rangle, \tag{20.10}$$

where $|a_\perp\rangle$ is a normalized, *uniform* superposition of all basis states perpendicular to $|a\rangle$, i.e.

$$|a_\perp\rangle = \frac{1}{\sqrt{N-1}} \sum_{\substack{x=0 \\ (x \neq a)}}^{N-1} |x\rangle . \tag{20.11}$$

We shall see that all the subsequent states generated during the Grover algorithm can also be written as a linear combination of $|a\rangle$ and $|a_\perp\rangle$. These can be conveniently drawn as vectors in the 2-dimensional space spanned by these two basis vectors, see Fig. 20.2.

Hence $|\psi_0\rangle$ makes an angle θ_0 with the $|a_\perp\rangle$ axis where $\sin\theta_0 = \langle a|\psi_0\rangle$, or

$$\sin\theta_0 = \frac{1}{\sqrt{N}} , \tag{20.12}$$

[3]We omit the subscript n on the states from now on since we will only be dealing with n-qubit states.

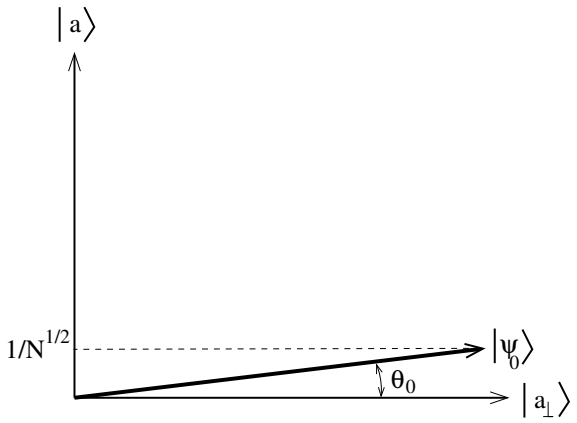


Figure 20.2: Projection of the 2^N-dimensional space on to a 2-dimensional space spanned by $|a\rangle$ and $|a_\perp\rangle$, the latter being a (normalized) equal linear combination of all basis states except for $|a\rangle$ itself, see Eq. (20.11). The vector in bold is the initial state $|\psi_0\rangle$, an equal linear combination of all basis states, see Eq. (20.9). The vector $|\psi_0\rangle$ has a projection $1/\sqrt{N}$ on to $|a\rangle$, so $\sin\theta_0 = 1/\sqrt{N}$, where θ_0 is the angle between $|\psi_0\rangle$ and $|a_\perp\rangle$.

so we can express $|\psi_0\rangle$ in Eq. (20.10) as

$$|\psi_0\rangle = \sin\theta_0\,|a\rangle + \cos\theta_0\,|a_\perp\rangle. \tag{20.13}$$

Note that $|\psi_0\rangle, |a_\perp\rangle$ and $|a\rangle$ are all normalized.

From Eq. (20.13) we see that if we were to measure $|\psi_0\rangle$ now, we would get $|a\rangle$ with probability $\sin^2\theta_0\,(= 1/N)$, which is *very small* for large N. (Of course we can also see that the probability is $1/N$ directly from Eq. (20.9).) The goal of the Grover algorithm is to iteratively rotate the vector representing the state of the input register from its initial direction, that of $|\psi_0\rangle$ (which is close to the $|a_\perp\rangle$ axis), to a direction close to the $|a\rangle$ axis, because a measurement of it will then give a with *high probability*. This is called *amplitude amplification*.

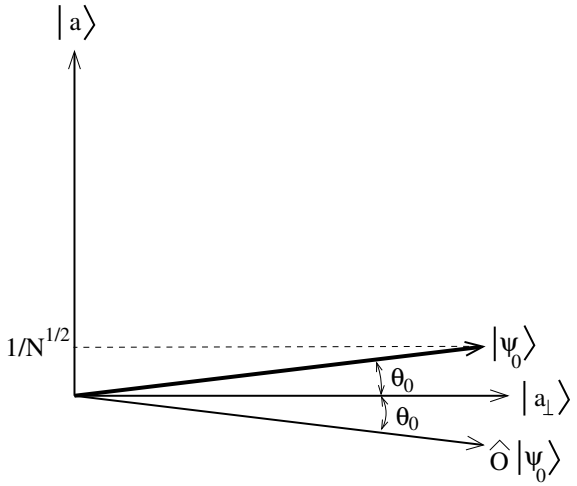


Figure 20.3: Figure showing that the action of the operator $\hat{O}$ is to reflect the state it is acting on, in this case $|\psi_0\rangle$, about the $|a_\perp\rangle$ axis.

As shown in Eq. (20.8) the action of $\hat{O}$ is to invert the component along $|a\rangle$ of the vector it acts on, while keeping the component perpendicular to $|a\rangle$ unchanged. The net effect is to *reflect* about the $|a_\perp\rangle$ axis. Figure 20.3 shows the effect of $\hat{O}$ on the initial state $|\psi_0\rangle$. To rotate the direction of the state towards the $|a\rangle$ axis we will need a second unitary operation that is discussed in the next section.

20.3 The second step of the Grover iteration

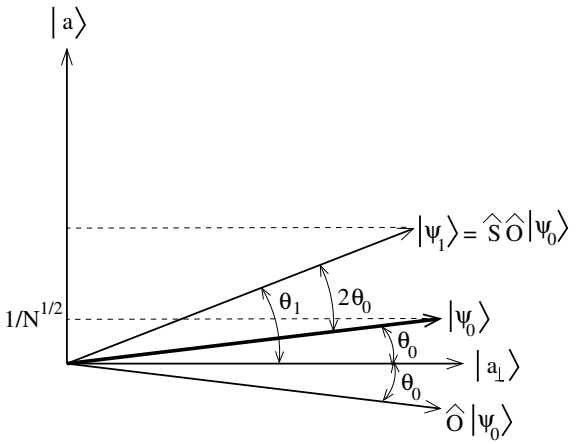


Figure 20.4: Figure showing that the action of the operator $\hat{S}$ is to reflect the state it is acting on, in this case $\hat{O}|\psi_0\rangle$, about the direction of $|\psi_0\rangle$ which is defined in Eq. (20.9). The net result of the two operations, $\hat{O}$ followed by $\hat{S}$, is to rotate the direction of $|\psi_0\rangle$ by $2\theta_0$ in an anti-clockwise direction. We will call the new state $|\psi_1\rangle$. It is at an angle $\theta_1 = \theta_0 + 2\theta_0$ to the $|a_\perp\rangle$ axis.

The second stage of a single Grover iteration is independent of the special number a. It changes the sign of the component perpendicular to the initial state $|\psi_0\rangle$ and keeps unchanged the component along $|\psi_0\rangle$. Denoting this operation by $\hat{S}$ we have

$$|\phi\rangle \to |\phi'\rangle = \hat{S}|\phi\rangle = 2|\psi_0\rangle\langle\psi_0|\phi\rangle - |\phi\rangle, \tag{20.14}$$

where $|\phi\rangle$ is an arbitrary state. You should check that $\langle\psi_0|\phi'\rangle = \langle\psi_0|\phi\rangle$, so the component along $|\psi_0\rangle$ is unchanged, and for a state $|\mu\rangle$ which is orthogonal to $|\psi_0\rangle$, $\langle\mu|\phi'\rangle = -\langle\mu|\phi\rangle$, showing that the component perpendicular to $|\psi_0\rangle$ has the sign changed. The net result if to reflect $|\phi\rangle$ about the direction of $|\psi_0\rangle$.

Figure 20.4 shows the effects of $\hat{S}$ acting on the state generated by $\hat{O}|\psi_0\rangle$. The combined effect of $\hat{O}$ followed by $\hat{S}$ is to rotate the initial state $|\psi_0\rangle$ by $2\theta_0$ in an anti-clockwise direction, i.e. $2\theta_0$ towards the desired direction of the $|a\rangle$ axis. The combination of these two operations is called a Grover iteration, implemented by the Grover operator

$$\hat{G} = \hat{S}\hat{O}. \tag{20.15}$$

The effect of the first Grover iteration, therefore, is to take the initial state $|\psi_0\rangle$ and rotate it anti-clockwise by $2\theta_0$. We will call the resulting state $|\psi_1\rangle$. It is at an angle θ_1 to the $|a_\perp\rangle$ axis, where

$$\theta_1 = \theta_0 + 2\theta_0, \tag{20.16}$$

see Fig. 20.4.

20.4 Subsequent iterations

Subsequent Grover iterations perform the same two steps: $\hat{O}$ which reflects about $|a_\perp\rangle$ followed by $\hat{S}$ which reflects about $|\psi_0\rangle$. The overall circuit implementing the Grover algorithm is shown in Fig. 20.5.

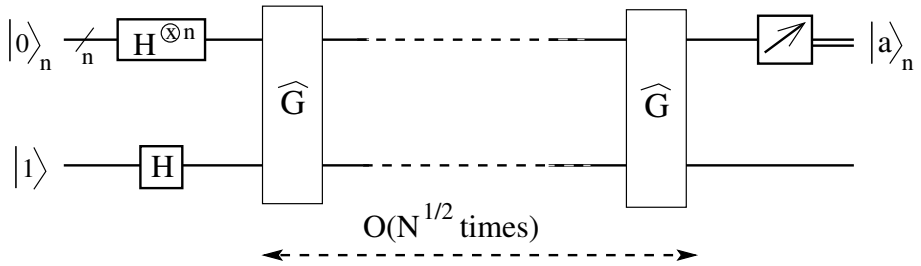


Figure 20.5: Circuit implementing the Grover algorithm. $\hat{G}$ is the Grover operator, given by $\hat{G} = \hat{S}\hat{O}$ where $\hat{O}$ and $\hat{S}$ are given by Eqs. (20.8) and (20.14) respectively. It acts only on the n input qubits (the upper line). The output qubit (the lower line) remains unchanged by $\hat{G}$. After $O(\sqrt{N})$ iterations of the Grover operator, the result of a measurement on the input qubits is the special value a with high probability.

If m iterations have already been done, so the current state is $|\psi_m\rangle$, Fig. 20.6 shows the effect of doing an additional iteration. The state $|\psi_m\rangle$ makes an angle θ_m with the $|a_\perp\rangle$ axis, so $\hat{O}$ rotates the direction by $2\theta_m$ clockwise, while $\hat{S}$ rotates it by $2(\theta_m + \theta_0)$ anti-clockwise. The net result is a rotation by $2\theta_0$ (independent of θ_m) anti-clockwise, which is towards the desired direction, $|a\rangle$, i.e.

$$\theta_{m+1} = \theta_m + 2\theta_0, \tag{20.17}$$

which gives

$$\theta_m = (2m+1)\theta_0 \tag{20.18}$$

The relationship between $|\psi_m\rangle, |a\rangle$ and $|a_\perp\rangle$ is

$$|\psi_m\rangle = \cos\theta_m |a_\perp\rangle + \sin\theta_m |a\rangle. \tag{20.19}$$

According to Eq. (20.19), the amplitude for $|\psi_m\rangle$ to be measured in state $|a\rangle$, i.e. $\langle a|\psi_m\rangle$, is $\sin\theta_m = \sin[(2m+1)\theta_0]$, the projection on to the vertical axis in Fig. 20.6. This increases as m increases up to the point where $\theta_m = \pi/2$ but then decreases. One therefore takes the number of Grover iterations, m, to be such that $\theta_m \simeq \pi/2$. From Eqs. (20.18) and (20.12) we see that we need

$$\theta_m = (2m+1)\theta_0 = (2m+1)\sin^{-1}\frac{1}{\sqrt{N}} = \frac{\pi}{2}, \tag{20.20}$$

which, for large N, gives

$$m = \frac{\pi}{4}\sqrt{N}. \tag{20.21}$$

When $\theta_m \simeq \pi/2$ measuring the state gives a with high probability.

We do not have to get the number of iterations precisely right. After m iterations, the probability that a measurement gives a is $\sin^2\theta_m = \sin^2[(2m+1)\theta_0]$. Any value of θ_m in the range

$$\frac{\pi}{4} < \theta_m < \frac{3\pi}{4} \tag{20.22}$$

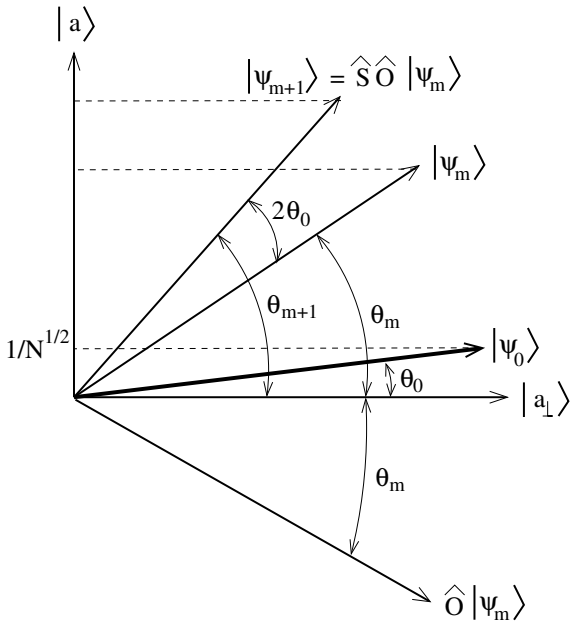


Figure 20.6: After the m-th iteration of the Grover algorithm, the state $|\psi_0\rangle$ has been rotated to $|\psi_m\rangle$, which makes an angle θ_m with the $|a_\perp\rangle$ axis. At the next iteration of the Grover algorithm, firstly the action of $\hat{O}$ reflects $|\psi_m\rangle$ about the $|a_\perp\rangle$ axis as shown. This is equivalent to a clockwise rotation by $2\theta_m$ so $\hat{O}|\psi_m\rangle$ is at an angle θ_m below the $|a_\perp\rangle$ axis. Secondly, the state $\hat{O}|\psi_m\rangle$ is acted on by $\hat{S}$ which reflects about the direction of $|\psi_0\rangle$. This is equivalent to an anti-clockwise rotation by $2(\theta_m + \theta_0)$. The net effect of the two operations is to rotate $|\psi_m\rangle$ by an angle $2\theta_0$ in an anti-clockwise direction. Hence the new state $|\psi_{m+1}\rangle$ is at an angle $\theta_{m+1} = \theta_m + 2\theta_0$ to the $|a_\perp\rangle$ axis. The amplitude for the state $|\psi_m\rangle$ to be $|a\rangle$ is the projection on to the vertical axis, which increases with m up to the point where $\theta_m = \pi/2$.

will get determine a correctly with a probability greater than $1/2$. For large N this corresponds to

$$\frac{\pi}{8}\sqrt{N} < m < \frac{3\pi}{8}\sqrt{N}. \tag{20.23}$$

Note that the probability *decreases* for $m > (\pi/4)\sqrt{N}$, unlike many algorithms where increasing the number of iterations progressively improves the probability of success.

The operation count of the Grover algorithm is $O(\sqrt{N})$ which is a quadratic speedup compared with the $O(N)$ count on a classical computer. The quantum speedup comes, of course, from quantum parallelism; all $N = 2^n$ values of $f(x)$ are evaluated in parallel, so naively it looks as though we should be able to get a speedup by a factor of N, i.e. an operation count of $O(1)$. However, if one measured directly after computing the function, one would just get one value of x and the corresponding $f(x)$, which is no better than on a classical computer. It requires additional operations, in the form of the Grover operator $\hat{G}$ applied iteratively, to extract a speedup, which in this case only reduces the operation count to $O(\sqrt{N})$ not $O(1)$. One can show that the $O(\sqrt{N})$ operation count of the Grover algorithm is optimal. An operation count of $O(1)$ is *proved* to be impossible.

20.5 Extensions

20.5.1 More than one special value

In the standard implementation of the Grover algorithm it is assumed that there is only one special value. If there are M solutions, $a_i, i = 1, \cdots, M$ then, proceeding along the lines of the derivation for one solution, one finds [NC00, Mer07, Vat16, RP14]:

(a) The states generated by the Grover algorithm can be written as a linear combination of a uniform superposition of all the special states,

$$|a\rangle = \frac{1}{\sqrt{M}} \sum_{x \in \{a_i\}} |x\rangle, \tag{20.24}$$

and a uniform superposition of all the other states,

$$|a_\perp\rangle = \frac{1}{\sqrt{N-M}} \sum_{x \notin \{a_i\}} |x\rangle. \tag{20.25}$$

We see that $|a\rangle$ and $|a_\perp\rangle$ are normalized.

(b) The initial state, $|\psi_0\rangle$, the uniform superposition of *all* states given in Eq. (20.9), can be written in terms of $|a\rangle$ and $|a_\perp\rangle$ as

$$|\psi_0\rangle = \sqrt{\frac{M}{N}}\,|a\rangle + \sqrt{\frac{N-M}{N}}\,|a_\perp\rangle. \tag{20.26}$$

Since $|a\rangle$ and $|a_\perp\rangle$ are normalized it follows that $|\psi_0\rangle$ is also normalized. Hence $|\psi_0\rangle$ makes an angle θ_0 with the $|a_\perp\rangle$ axis where $\sin\theta_0 = \langle a|\psi_0\rangle$, or

$$\sin\theta_0 = \sqrt{\frac{M}{N}}, \tag{20.27}$$

rather than Eq. (20.12). Consequently we can write Eq. (20.26) in terms of θ_0 in the same way as for $M = 1$, namely Eq. (20.13).

(c) Subsequent iterations rotate the direction of the state by an angle $2\theta_0$ towards the $|a\rangle$ axis and so, after m iterations, the angle θ_m is given by Eq. (20.18), and the state $|\psi_m\rangle$ is given by Eq. (20.19). Hence the effect of each Grover iteration, when expressed in terms of θ_0, is the same as for $M = 1$, and the only difference compared with $M = 1$ is that θ_0 is given by Eq. (20.27) rather than (20.12).

(d) Assuming $M \ll N$, then θ_m is approximately $\pi/2$ when the number of iterations m is given by

$$m = \frac{\pi}{4}\sqrt{\frac{N}{M}}. \tag{20.28}$$

After this number of iterations of the Grover operator, with high probability a measurement of the state will give *one* of the special values a_i with equal likelihood.

The student is advised to check these steps.

20.5.2 Quantum Counting

The results of the previous subsection are only useful if we *know* in advance how many special values, M, there are. If we have no prior knowledge of M, how can we determine it? We saw that the Grover operator $\hat{G}$ rotates vectors in the $|a\rangle$–$|a_\perp\rangle$ plane by an angle $2\theta_0$, where θ_0 is given by Eq. (20.27) and so depends on M. In other words, in the space of $|a\rangle$ and $a_\perp\rangle$, the Grover operator has the standard form of a rotation matrix

$$\hat{G} = \begin{pmatrix} \cos 2\theta_0 & -\sin 2\theta_0 \\ \sin 2\theta_0 & \cos 2\theta_0 \end{pmatrix}. \tag{20.29}$$

The eigenvalues of $\hat{G}$ are easily found to be $\exp(\pm 2i\theta_0)$. (It is a general property of unitary matrices that their eigenvalues are a pure phase.) We already showed in section 17.4 in Chapter 17 that the phase of the eigenvalue of a unitary matrix can be determined from the phase estimation algorithm using Shor's quantum Fourier transform.

Consequently, we can determine θ_0 (and hence M), and also get one of the special values a_i, by combining the Quantum Fourier Transform with Grover's algorithm. In fact this "quantum counting" algorithm will even tell us whether or not a special value exists at all, i.e. whether or not $M = 0$. The interested student can find more details in advanced texts such as Refs. [NC00, RP14].

Chapter 21

Quantum Protocols Using Photons

There are several problems of interest where qubits can be considered one at a time, without needing any qubit-qubit interactions. Photons are ideal qubits for this because their interactions with each other are immeasurably weak, and they can be propagated down optical fibres for a big distance with little attenuation while preserving their polarization. You will recall from Sec. 1.4 that it is the polarization of the photon which characterizes the qubit, e.g.:

$$\begin{aligned} |0\rangle &\equiv |\leftrightarrow\rangle, \qquad (\text{left}-\text{right}) \\ |1\rangle &\equiv |\updownarrow\rangle, \qquad (\text{up}-\text{down}) \\ |+\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) &\equiv |\nearrow\!\!\swarrow\rangle, \qquad (\text{one of the diagonals}) \\ |-\rangle = H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) &\equiv |\nwarrow\!\!\searrow\rangle, \qquad (\text{the other diagonal}). \end{aligned} \tag{21.1}$$

The connection between the polarization of photons and qubit states was described in more detail in Sec. 4.1.

Several quantum protocols involving photons have been successfully implemented. Here we will discuss applications to cryptography and "teleportation". Some references are [NC00, Vat16, Mer07].

21.1 Quantum Key Distribution

Cryptography is concerned with transmitting secret messages. There are two main approaches:

- **Public Key**
  An example is the RSA scheme which we already met in Chapter 14 in the context of Shor's algorithm for factoring integers. Let us briefly review the basic idea. Suppose Bob wants to send a message to Alice. Alice sends her public key down an open channel to Bob who uses this to encrypt his message. Alice decodes the encrypted message using her private key. The private key is not shared, only the public key. Security depends on the difficulty of decoding the message without the private key. In the case of RSA we recall that this required factoring a large integer.

- **Private key (or symmetric key)**. (Note: public key encryption is not symmetric between sender and receiver.)
  Alice and Bob share a private key, which has been generated and shared in advance. This must be as long as the message and, as we shall explain later, can only be use once. But how do Alice and Bob share the private key securely? Perhaps Alice could put it in a box and send it to Bob by FedEx. This is not convenient which is why internet transactions use public key encryption instead.

We shall now see that quantum mechanics can help with securely sharing private keys, using what is called **Quantum Key Distribution** (QKD).

The idea of QKD is to create a one-time codepad which Alice and Bob share. By using quantum mechanics, Alice and Bob will be able to detect whether an eavesdropper whom, following tradition, we shall call Eve, is trying to intercept their messages when they share the codepad.

The codepad is a shared random string of bits R, which must be at least as long as the message. Alice encodes the message M by bit-wise XOR-ing it with the random string, i.e.

$$\text{Alice}: \qquad M \longrightarrow M \oplus R \ (= M'). \tag{21.2}$$

Bob decodes the encoded message M' by also XOR-ing it with R, i.e.

$$\text{Bob}: \qquad M' \longrightarrow M' \oplus R = M. \tag{21.3}$$

This works because $M \oplus R \oplus R = M$, as we have discussed several times before in the course.

We now explain why this codepad can only be used once securely. Suppose we send two encoded messages using the same codepad, i.e.

$$\begin{aligned} M_1' &= M_1 \oplus R \\ M_2' &= M_2 \oplus R. \end{aligned} \tag{21.4}$$

Anyone intercepting the message can XOR the two messages with the result

$$M_1' \oplus M_2' = M_1 \oplus R \oplus M_2 \oplus R = M_1 \oplus M_2, \tag{21.5}$$

so the random string has dropped out. The eavesdropper can then use standard methods (e.g. letter frequency) to decrypt. This is harder than for a single message since one has to extract both messages, but feasible. Hence the great security[1] coming from using a random bit string has been lost.

How do Alice Bob know that their random bit string R was not intercepted by Eve as they were sharing it? This is where quantum mechanics comes into play.

21.1.1 BB84 protocol

We describe here the method proposed by Bennett and Brassard in 1984 (BB84). Alice sends Bob a long string of photons. Each photon is in one of the four polarization states in Eq. (21.1). The polarization states corresponding to qubits $|0\rangle$ and $|1\rangle$ we will call Z-basis qubits (since this is the basis in which Z is diagonal). The polarization states corresponding to $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ we will call X-basis qubits (since this is the basis in which X is diagonal). To decide in which basis to send a photon Alice generates a random integer taking values 0 and 1. If she gets 0 she sends a Z-basis photon, and if she gets 1 she sends an X-basis photon. Within each basis-type there are two states, which Alice chooses by generating a second random integer, again taking values 0 and 1. If she gets 0 she sends $|0\rangle$ if the Z-basis were chosen and $H|0\rangle$ if the X-basis were chosen. If she gets 1 for the second random number, she sends $|1\rangle$ or $H|1\rangle$, depending on whether the Z-basis or X-basis was chosen. An example of a set of photons sent to Bob is

basis	Z	X	X	X	Z	Z	X	Z	X	$\cdots$
state	0	1	0	1	1	0	1	0	0	$\cdots$

(21.6)

Bob receives these qubits and decides randomly whether to measure in the Z-basis or the X-basis. Note that the photons are individually identifiable by the sequence in which they arrive.

[1] If the bit string is truly random it is impossible to decrypt the message without knowing the string.

If the *basis* in which Alice sends a photon (Z or X) is the same as that in which Bob measures it, then the *state* which Bob measures, 0 or 1, must be the same as the state that Alice sent. However if the bases for sending and measuring are different, then Bob will only find the same state as Alice about half the time. Alice tells Bob over an insecure channel which photons were in the Z basis and which in the X-basis, but not the state. Bob then tells Alice over an insecure channel for which of the photons he measured in the same basis as she sent it in. They keep these and discard the others (about 1/2 on average).

The onetime codepad is the set of random bits corresponding to the state of the qubits for which Alice and Bob measured in the same basis. Note that this information was *not* send down the insecure channel.

Let's complete the above example with a possible set of measurements that Bob made.

Alice	basis	Z	X	X	X	Z	Z	X	Z	X	$\cdots$
	state	$0^\star$	1	0	1	1	$0^\star$	1	0	0	$\cdots$
Bob	basis	X	X	X	Z	Z	X	Z	Z	Z	$\cdots$
	state	$1^\star$	1	0	1	1	$1^\star$	1	0	0	$\cdots$

(21.7)

For the photons where Alice's and Bob's bases agree, the information is boxed. For these photons, the state that Alice generated and that which Bob measured agree. For the other photons, the states agree only half the time on average. The cases where the states disagree are starred (in this example, the states differ for 2 out of the 5 cases where the bases differ).

The codepad which Alice and Bob have shared is the set of states for which their bases agree, i.e.

$$R = 1010\cdots . \tag{21.8}$$

How can Alice Bob know if Eve is interrupting the photons? Consider the "good" photons, those where Alice and Bob used the same basis. If Eve is not interrupting them, then Alice and Bob agree on the state with 100% probability. However, if Eve measures the photons and sends them on to Bob, then Alice and Bob will have different states some of the time, as we now show. Like Alice and Bob, Eve will have to choose a random basis for each photon. There is probability 1/2 that she will choose a different basis from the common basis of Alice and Bob. We remind ourselves that

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), & H|1\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right), \\ |0\rangle &= \frac{1}{\sqrt{2}}\left(H|0\rangle + H|1\rangle\right), & |1\rangle &= \frac{1}{\sqrt{2}}\left(H|0\rangle - H|1\rangle\right). \end{aligned} \tag{21.9}$$

Hence, out the times when Eve chooses a different basis from the common basis of Alice and Bob, there is a probability 1/2 that Eve's intervention will result in her sending on to Bob a photon in the opposite state from the one which Alice sent. Hence, for the photons where Alice and Bob used the same basis, Eve's intervention results in Alice and Bob having different states about 1/4 of the time[2].

To see if this is happening, Alice and Bob sacrifice some fraction of the good photons by sending their values for the state down an insecure channel. If about 1/4 of the states disagree, then they know that the photons are being intercepted. If only a small fraction disagree, Alice and Bob would have needed to decide beforehand up to what fraction of disagreements they would consider an acceptable risk in order to still send the message.

In summary, a quantum key distribution protocol is able to detect an eavesdropper because measurements in quantum mechanics change the state in general.

[2]There is a probability 1/2 that Eve measures in a different basis and for those qubits there is a probability 1/2 that her measurement changes the state.

21.1.2 BB92 protocol

There is a later version, also due to Bennet and Brassard, from 1992 (BB92), in which only two polarizations are used: $\leftrightarrow$ and $\swarrow\!\!\nearrow$. Note that these states are not orthogonal. Lack of orthogonality is essential for the method to work. If only orthogonal states are used then there is only one basis, so if Eve knows what this is she can measure the states of the photons in this basis and send then on to Bob without being detected.

The BB92 protocol works as follows. To decide in which state to send the k-th photon, Alice generates a random bit, k_i, which is 0 or 1. If she gets 0 she sends $|\leftrightarrow\rangle \equiv |0\rangle$ a Z-type photon, whereas if she gets 1 she sends $|\swarrow\!\!\nearrow\rangle \equiv H|0\rangle$, an X-type photon.

If Bob were to always measure in the same basis as the one Alice used, i.e. the Z basis for Z-type photons, and the X basis for X-type photons, he would always get 0. However, he doesn't know which basis Alice used, so, for each photon, he chooses a random basis by generating a random bit l_i. As Alice also did, Bob chooses the Z-basis if l_i is 0 and the X basis if the $l_i = 1$. He notes for which photon he *measures* 1 and sends this information to Alice on a public channel. This only happens when they use different bases, i.e. they generate complementary random bits, $k_i = 1 - l_i$, since if they use the same basis Bob must get 0. The shared key is then the set $\{l_i\}$ for which Bob measures 1. Alice just has to take the complement of her bits for the same photons to get the same key as Bob.

The effect of an eavesdropper is the same as for the BB84 protocol.

21.2 Homework Problem On Teleportation

Another quantum protocol which has been successfully implemented using photons is teleportation, in which the state of a qubit, though not the physical qubit itself, is transported (teleported) from one location to another. The best reference for this is Mermin [Mer07].

Here we give an introduction to teleportation in the form of a homework problem.

Suppose that Alice has a qubit in a state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{21.10}$$

The values of α and β are unknown to her and can not be determined without destroying the superposition. The no-cloning theorem means that we can't do repeated measurements on copies of this state. This qubit may be the result of a (possibly complicated) quantum computation which Alice would like to send on to Bob to continue the computation. Bob is far away and Alice can not physically transport the qubit to Bob but wants to send the state.

Now Alice and Bob are able to

- share an entangled qubit
$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\left(\,|0\rangle_a|0\rangle_b + |1\rangle_a|1\rangle_b\,\right), \tag{21.11}$$
where a stands for Alice's qubit and b stands for Bob's,
- and communicate over a classical channel (e.g. by phone).

Hence, together they have a 3-qubit state,

$$|\phi_0\rangle = \frac{1}{\sqrt{2}}\left(\,\alpha|0\rangle_a + \beta|1\rangle_a\,\right) \otimes \left(\,|0\rangle_a|0\rangle_b + |1\rangle_a|1\rangle_b\,\right) \tag{21.12}$$

$$= \frac{1}{\sqrt{2}}\left(\,\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle\,\right), \tag{21.13}$$

where the leftmost two qubits refer to Alice and the rightmost qubit to Bob.

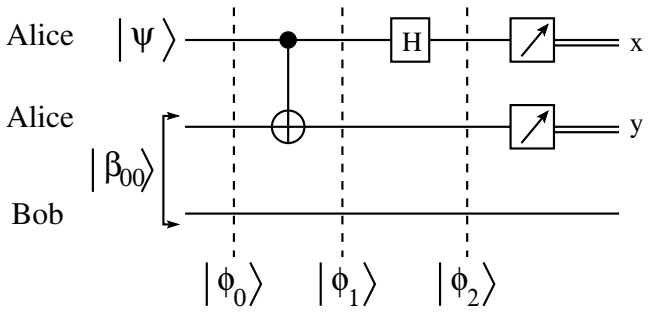


Figure 21.1:

Alice now applies a Bell measurement (discussed in Chapter 9) to the two qubits in her possession, see Fig. 21.1.

(i) Determine the states $|\phi_1\rangle$ and $|\phi_2\rangle$ shown in the figure.

(ii) Alice then measures the two qubits in her possession, obtaining results x and y as shown. She then calls up Bob and tells him the result of her measurements.
Explain what Bob needs to do, depending on the results of Alice's measurements, for his qubit to be in state

$$|\psi\rangle = \alpha|0\rangle_b + \beta|1\rangle_b, \tag{21.14}$$

i.e. the state that was originally in Alice's possession.

Note:

- The state, but not the physical qubit, has been transported. This is called *teleportation*.
- This procedure doesn't violate relativity (according to which information can not be transmitted faster than the speed of light) since classical communication between Alice and Bob is required.
- It does not violate the no-cloning theorem because, at the end, Alice doesn't have her original state $|\psi\rangle$, only two classical bits x and y. There is never more than one copy of $|\psi\rangle$ in existence.

Final Comment:
There are claims that teleportation has been verified experimentally which I will now discuss briefly. One would like to show the following:

- Alice stores state $|\psi\rangle$.
- The state $|\psi\rangle$ is transported to Bob who is far away.
- Bob stores state $|\psi\rangle$.

To transport qubits over a long distance one needs photons. One can teleport photons over a large distance while retaining their polarization, but at present one can not conveniently store them. One can store other types of qubits, e.g. trapped ions, but can't entangle them over large distances, so they can be teleported only locally. Hence, in my view, a complete demonstration of teleportation, incorporating all three bullet points above, has not yet been achieved.

Chapter 22

Epilogue: Quantum Simulators

We are currently in the middle of what is called the "second quantum revolution". The first quantum revolution was the development of quantum mechanics in the 1920's and subsequent applications to devices like integrated circuits. These applications treated the information, i.e. the bits, classically. However, in the second quantum revolution, the information is treated according to the rules of quantum mechanics.

In this book we have discussed what is called the circuit model (or gate model) of a quantum computer. The qubits are initialized, and then acted on by a series of discrete unitary transformations to solve the problem at hand. This sort of quantum device is what people are normal refer to when they talk about a "quantum computer". The circuit model quantum computer was proposed initially by David Deutsch [Deu85].

However, other types of quantum device are being developed as part of the second quantum revolution, many of which can be termed "quantum simulators". It is anticipated that we will have new results from quantum simulators, i.e. results which could not be obtained by a classical computer, in the next few years. By contrast, the ability to get new results from a circuit model quantum computer, for example by decoding information sent down the internet using Shor's algorithm (which requires factoring a huge integer) will be very far in the future, if ever.

The idea of a quantum simulator is to use an artificial quantum device to simulate the quantum system which we want to understand. It was first proposed by Feynman[Fey82]. For example a quantum chemist might want to understand the properties of a certain molecule, or a condensed matter physicist might want to understand a material with unusual magnetic or superconducting behavior. Properties of these materials are, of course, determined by quantum mechanics. Many problems in nature are not amenable to analytic (i.e. pencil and paper) calculations and need to be simulated. Although many problems can be simulated efficiently on a classical computer, there remain problems of interest where the quantum aspects cause serious difficulties for classical simulations. As an example, we learn in quantum mechanics classes that particles of a certain type (e.g. electrons or protons or π mesons) are (i) all identical and (ii) are in one of two classes, bosons or fermions. For bosons, the state of the system (wave function) does not change if the two particles are interchanged, whereas for fermions the state does change sign under particle interchange. This sign change for fermions creates great difficulty when trying to simulate fermions on a classical computer.

By and large quantum simulators are *analog* devices. The reason is that, in order for the system of qubits to model the problem of interest, there must be interactions between the qubits. In a classical (digital) computer they would be represented by floating point numbers with typically 16 digits or precision[1]. In a quantum computer, however, interactions are induced by turning some "knob" on the

[1]For many purpose this can be considered exact but, in any case, the interactions are represented by a precisely *known* string of bits.

experimental apparatus, the nature of the knob depending on the hardware used for the qubits. For example, in the case of superconducting qubits, interactions would be determined by the value of a magnetic field threading superconducting loops. The magnetic field takes a continuous range of values (i.e. is analog) and can only be set within a certain level of precision.

Above I stated that we will probably have new results from a quantum simulator before we have new results from a (circuit model) quantum computer. Why is this? A quantum computer uses quantum parallelism to get its quantum speedup. This depends on accurately preserving phase relations between the different pieces of the state. These phase relations are destroyed by noise, an effect called decoherence. Present-day qubits are quite noisy. In principle one can include error correction, but this requires a huge number of physical qubits for each logical qubit. Thus, in the near future, we will have to live with noisy qubits. However, as stated, noise is a disaster for circuit model quantum computers.

Is a modest amount of noise as big a disaster for a quantum simulator? The answer is "probably not". For example, suppose we want to simulate the temperature dependence of the behavior of a material which goes superconducting. A non-zero temperature means that there is noise due to thermal fluctuations. One might hope that a bit of extra (non-thermal) noise from the qubits would not change the results all that much, so the results would, nonetheless, be useful. The next paragraph discuses another example for which there is also reason to believe that some noise is not disastrous.

A particular type of quantum simulator is one used to solve "optimization" problems, where we need to find the maximum (or minimum) of some "objective function" with constraints. Let's assume for concreteness that we want the minimum. Optimization problems are very important in science and engineering, two widely used applications being speech recognition and image recognition. Optimization problems are hard when there is "frustration", i.e. competition, between different pieces of the function that one has to minimize. In these cases, if one locally minimizes individual pieces, one will end up in a "local minimum" rather than the global minimum. It has been proposed to use "quantum annealing"[2] to try to find the global minimum. We recall from Chapter 3 that if we have two operators which don't commute then one or both of them must have an uncertainty in any quantum state. Thus non-commuting operators generate fluctuations. By making the (classical) objective function become quantum, by adding a non-commuting "driver" piece to it, one induces fluctuations, which can help get one out of a local minimum. In such a "quantum annealer" the qubits simulate the "objective function plus driver function". By letting the driver piece tend to zero at the end of the simulation, the model simulated at the end is just the objective function, and we anticipate that the set of qubits will then be close to the ground state. Quantum annealing has been pioneered by a company called D-Wave, which has manufactured machines with around 5000 qubits. These 5000 qubits do not maintain coherence during the time of the simulation, but it is anticipated that, despite some noise, the induced quantum fluctuations will help to find the ground state.

To summarize, in the near future qubits will be noisy and we won't be capable of assembling a huge number of them together. Hence, in the short and intermediate term, we will only have "Noisy Intermediate-Scale Quantum" (NISQ) devices. I expect that in the next few years we will be able to get interesting new[3] results from NISQ *simulators*, but probably not from NISQ circuit model *quantum computers*.

[2]It was earlier proposed to add thermal fluctuations to solve optimization problems. This approach is called "thermal annealing" or "simulated annealing". Whether quantum annealing is more efficient in finding ground states than classical algorithms such as simulated annealing is hotly debated at present.

[3]By "new" I mean results that would be impossible to obtain on a classical computer.

Bibliography

[Deu85] D. Deutsch. Quantum theory, the Church-Turing Principle and the Universal Quantum Computer. *Proc. Roy. Soc. London*, 400:97, 1985.

[Fey82] R. P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21:467, 1982.

[FLS64] R. P. Feynman, R. B. Leighton, and M Sands. *The Feynman Lectures on Physics.* Addison Wesley, New York, 1964. Available online at https://www.feynmanlectures.caltech.edu/.

[FMMC12] A.G. Fowler, M. Mariantoni, J.M. Martinis, and A.N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, 2012.

[Gri05] D. J. Griffiths. *Introduction to Quantum Mechanics.* Addison-Wesley, Boston, 2005.

[Mer07] N. D. Mermin. *Quantum Computer Science.* Cambridge University Press, Cambridge, 2007.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, Cambridge, England, 2000.

[RP14] E. Rieffel and I. Polak. *Quantum Computing; A Gentle Introduction.* MIT Press, Massachusetts Institute of Technology, 2014.

[Sho94] P. W. Shor. Algorithms for quantum computing: discrete logarithms and factoring. In S. Goldwasser, editor, *Proc. 35th Symp. on Foundations of Computer Science*, page 124, Los Alamitos CA, 1994. IEEE Computer Society Press.

[Sho95] P. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:2493, 1995.

[Ste96] A. M. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, 77:793, 1996.

[Vat16] R. Vathsan. *Introduction to Quantum Physics and Information Processing.* CRC Press, Boca Raton, 2016.

Index