**Part IIC** (January 2019 version) **Lent term 2019**

# QUANTUM INFORMATION AND COMPUTATION

## Lecture notes

**Richard Jozsa, DAMTP Cambridge**
rj310@cam.ac.uk

## CONTENTS

**Some useful references:**

M. Nielsen and I. Chuang "Quantum computation and information". CUP.

B. Schumacher and M. Westmoreland, "Quantum processes, systems and information". CUP 2010.

S. Leopp and W. Wootters, "Protecting information: from classical error correction to quantum cryptography". Academic press 2006.

John Preskill's notes for Caltech course on quantum computation.
Available at http://www.theory.caltech.edu/people/preskill/ph229/notes/book.ps

# 1 Introduction: why *quantum* computation and information?

Let us begin by asking: "*What is information?*" Well, intuitively we "get information" when we acquire knowledge of an alternative that we did not know before. In classical computing and communication, information is conventionally represented by a *classical bit* (or string of bits) viz. a Boolean variable that can take values 0 or 1. It represents the "elementary unit of information" giving the result of a single binary decision e.g. a yes/no question. More elaborate kinds of information are represented by bit strings to provide distinct labels for more than two *a priori* possible answers.

Having our notion of information we can then go on to introduce *computation* as *information processing* i.e. the updating of a bit string by a prescribed sequence of steps (the "program"). These steps (at the basic bit string level) are normally taken to be local Boolean gates – in each step one or two of the bits (at prescribed locations in the string) are updated by applying a prescribed Boolean function, or gate, to them. Typical examples of such gates are the 1-bit NOT operation and the 2-bit operations called AND and OR; and it can be shown that this small basic set is "universal" in the sense that it suffices for the construction of any Boolean function (with inputs and outputs being bit strings of any length).

All the above will probably be unsurprising to many readers but we can go further – if information is represented by bits then: "*What is a bit?*" Above we have associated it with a Boolean variable, an abstract mathematical concept. But this cannot be our answer because when we acquire information, we need to actually receive "something real", not just entertain an abstract mathematical concept. The key point here is well expressed by the quote (R. Landauer 1996) "Information is not a disembodied abstract entity; it is always tied to a physical representation". Indeed the Boolean values 0 and 1 serve only to provide two recognisably different labels. So our answer to "What is a bit?" is: a bit is given by any two different physical states (of some physical system) that can be reliably distinguished (by a physical measurement). The Boolean values 0 and 1 are just two distinguishable patterns of physical ink on a page; when we ask a question and hear 'yes' or 'no', we are just using our ears as a physical device to distinguish two different soundwave forms in the air; and in a computer memory, bits can be represented by two different voltage levels in a material etc. The key message here is: "No information without representation!"

That message may by now in retrospect seem almost self evident, but its consequences, gentle reader, are truly awesome: if information is represented in physical states or degrees of freedom of some physical system, then any possible act of computation, or information processing, must correspond to a physical evolution of that physical system. For example, any actual computer is always a physical device whose operation must *ipso facto* obey the laws of physics.

More generally we see that:
*the possibilities and limitations of information storage, processing (i.e. computation) and communication must all rest on the laws of physics and cannot be determined by abstract thought or mathematics alone!*
Thus there must be a deep fundamental connection between physics and computation, and *that* is why we need *quantum* information and computation. Or as Feynman put it more succinctly: because "Nature isn't classical, dammit...". And indeed it can be argued that our conventional generally accepted model of computation (in any of its equivalent forms, e.g. based on Turing machines or viewing computations in terms of Boolean gates etc.) amounts to the computational possibilities allowed by the laws of classical physics.

Quantum physics differs dramatically from classical physics in the way it represents the physical world and the kinds of processes that it allows (as we'll see in detail shortly). Quantum computation and information is the study of the possible applications and exploitation of these novel quantum features in issues of information representation, computation, computational complexity, cryptography and communication. The subject is a remarkable synthesis of theoretical computer science, classical information theory and cryptography with quantum physics, promising a series of benefits of much practical significance, beyond the remit (even in principle) of conventional (classical) computing and information technology.

The subject emerged in the mid-1980's and it is currently one of the most active areas of all scientific research internationally. Here we'll just briefly highlight some of the key issues for its benefits:

*Computing power - computational complexity issues.*
As we'll see later in the course, a quantum computer cannot compute any computational task that's not already computable in principle on a classical computer. However the key issue here is not 'computability in principle' but 'computability in practice' i.e. that of computational *hardness*. In computational complexity theory the 'hardness' of a computational task is measured by the amount of computational resources needed to compute it; the resources considered are 'time' i.e. the number of computational steps, and 'space' i.e. the amount of computer memory workspace needed.

For example, think of the task of factoring a given integer with $n$ digits, and how hard this is to do, as a function of $n$. Here $n$ is called the 'input size' for the computation. For any proposed algorithm we ask: does the time (i.e. number of steps) grow polynomially with $n$ (so-called poly-time algorithms) or exponentially with $n$ (faster than any polynomial growth), as $n$ gets larger and larger. Poly-time algorithms are regarded as "feasible in practice" whereas a task with only exponential time algorithms, while computable in principle, is regarded as infeasible, or effectively uncomputable, in practice – because for relatively modest input sizes, the time required would exceed any reasonably available period (e.g. exceeding the age of the universe).

This is where quantum computing has a major impact: we'll see that the formalism of quantum theory leads to new kinds of "non-classical" modes of computation (new kinds of computational steps for information processing), providing remarkable new possibilities for computational algorithms. In some cases these possibilities are able to cross

the boundary between poly-time and exponential time algorithms i.e. there are some computational tasks for which no known classical poly-time algorithm exists but which can be solved in poly-time on a quantum computer i.e. these tasks, which are effectively uncomputable in practice on a classical computer, become computable in practice on a quantum computer.

The most famous example is the computational task of integer factorisation. In classical computation there is no known algorithm that runs in polynomial time (in the number of digits) but in 1994 Peter Shor discovered a poly-time *quantum* algorithm for factorisation. We emphasise that this exponential speedup in time is achieved not by an increase in clock speed of steps on the computer, but by exploiting entirely new (quantum) kinds of computational steps (and needing exponentially fewer of them) that are simply not available to classical computers.

We'll see a variety of examples of such quantum computational benefits (including factoring) in the second half of the course.

*Communication and security issues - quantum states as information carriers.*
Intrinsically quantum (i.e. non-classical) features of quantum states (including the possibilities of quantum superposition, entanglement and principles of quantum measurement theory) can be exploited to provide novel possibilities (beyond what's achievable with classical physics) for information communication and security. These include the so-called process of quantum teleportation, and a variety of important cryptographic issues such as the ability to implement provably secure communication. In this course we'll discuss quantum teleportation, the Bennett-Brassard quantum scheme for secure communication, and some further features of quantum states when viewed as information carriers.

*Technological issues.*
Historically in computer science (before the advent of quantum computing) there was a phenomenon known as Moore's law viz. that since 1965 there has been a steady rate of miniaturisation of computer components, by approximately a factor of 4 every 3.5 years. With this trend we have now effectively reached the atomic scale where classical physics fails completely and quantum effects are dominant – components begin to malfunction in 'bizarre' quantum ways. To deal with this, we could either aim to re-design our components to stamp out the new effects to provide the same functionality as before, or else we could embrace the new quantum effects, aiming to exploit them in new kinds of computational ways. Our discussion of computational complexity above shows that the latter is surely the way to go!

However this involves immense technological challenges: it turns out that quantum states and processes are intrinsically more fragile and difficult to control cleanly, than their classical counterparts. Inspired by the theoretical technological possibilities on offer, in recent years there has be a huge effort devoted to developing the needed quantum technological capability. To date, some quantum cryptographic protocols have been implemented (including secure communication), even to the level of being commercially available. However these require quantum processing of only relatively small systems (thus within our current quantum technological capabilities) and similarly some quantum algorithms

on very small input instances have been demonstrated. But the ultimate 'holy grail' of a working scalable universal quantum computer is currently beyond our quantum technological capability. Some of the world's leading information technology companies (including IBM, Google, Microsoft) have mounted huge research and development efforts with just that aim. They have announced their expectation of having a working quantum computer in 2018 (for the first time) that can reliably process around 50 qubits of quantum information (a qubit being the quantum analogue of a classical bit). This may not sound like a large number e.g. what useful computation can you carry out with just 50 classical bits!? But this comparison is completely misguided: astonishingly it turns out that the extraordinary possibilities of quantum effects (that we'll see) imply that even a quantum computer of only around 50 qubits should already be able to perform some computational tasks that are beyond the dedicated use of all classical computing power on earth today.

In the popular science press, quantum computing (and quantum information technology more broadly) has been the subject of some sensationalisation. In that spirit one could indeed say that (in view of the above expected technological developments) it is now widely accepted that we are presently on the cusp of the next major revolution in technology, in a venerable tradition that perhaps began with the stone age, and later the iron age, continuing to steam and mechanical engineering, electrical, then electronic, and now quantum technology (including more broadly nano-technology), for the first time embracing fully the technological possibilities of quantum physics.

# 2 Principles of quantum mechanics and the Dirac bra ket notation

We will begin by setting out the basic principles of quantum mechanics (as four basic principles (QM1) - (QM4)) while simultaneously introducing and explaining the formalism of Dirac notation, which we will use to express their mathematical content.

Dirac notation is nothing more than an alternative notation for basic linear algebra which is widely used in quantum mechanics. We will use it for essentially all aspects of this course so it will be important for you to master it at the outset!

## 2.1 Quantum states and unitary operations

### Dirac notation: bra and ket vectors

Let $V$ be a (finite dimensional) complex vector space of dimension $m$ with an inner product. Vectors in $V$ will be written as $|v\rangle$ and called *ket vectors* or just *kets*. Thus we will use a curious asymmetrical bracket notation rather than a more conventional notation such as $\underline{v}$. In this course we'll often work with a 2-dimensional space $V_2$ with a chosen orthonormal basis denoted $\{|0\rangle, |1\rangle\}$ i.e. the basis vectors are labelled by the Boolean bit values 0 and 1, and this basis will be called the computational basis or standard basis. (All of our constructions and formulae can be easily generalised to arbitrary finite dimensional spaces). Ket vectors $|v\rangle = a|0\rangle + b|1\rangle$ are always written in components as *column* vectors

$$|v\rangle = \begin{pmatrix} a \\ b \end{pmatrix}.$$

The conjugate transpose $|v\rangle^\dagger$ (denoted by a dagger) is called a *bra vector* and is written using a mirror image bracket

$$\langle v| = |v\rangle^\dagger = a^* \langle 0| + b^* \langle 1| = (a^* \ b^*).$$

Thus in components bra vectors are always written as row vectors. If $|w\rangle = c|0\rangle + d|1\rangle$ is another ket then the inner product of $|v\rangle$ and $|w\rangle$ is written by juxtaposing brackets

$$\langle v|w\rangle = |v\rangle^\dagger |w\rangle = (a^* \ b^*) \begin{pmatrix} c \\ d \end{pmatrix} = a^*c + b^*d.$$

Indeed the whole Dirac notation formalism is motivated by the bracket notation $(\underline{v}, \underline{w})$ for inner products commonly used in mathematics, hence the terms "bra" and "ket" vectors, giving the inner product as a "bra-ket". Orthonormality of the basis $\{|0\rangle, |1\rangle\}$ is equivalent to the condition $\langle i|j\rangle = \delta_{ij}$ (the Kronecker delta). In more abstract terms, bra vectors are a notation for elements of the dual space $V^*$ viz. $\langle v|$ is the linear functional whose value on any ket $|w\rangle$ is the inner product $\langle v|w\rangle$.

### Dirac notation: tensor products of vectors

If $V$ and $W$ are vector spaces of dimensions $m$ and $n$ with bases $\{|e_1\rangle, \ldots, |e_m\rangle\}$ and $\{|f_1\rangle, \ldots, |f_n\rangle\}$ respectively, then the tensor product space $V \otimes W$ has dimension $mn$ and can be regarded as consisting of all formal linear combinations of the symbols $|e_i\rangle \otimes |f_j\rangle$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. There is a natural bilinear embedding $V \times W \to V \otimes W$ defined as follows. If $|\alpha\rangle = \sum_i a_i |e_i\rangle$ and $|\beta\rangle = \sum_j b_j |f_j\rangle$ are general vectors in $V$ and $W$ respectively then

$$(|\alpha\rangle, |\beta\rangle) \mapsto |\alpha\rangle \otimes |\beta\rangle = \sum_{ij} a_i b_j |e_i\rangle \otimes |f_j\rangle \qquad (*)$$

obtained by formally "multiplying out" the brackets in $(\sum_i a_i |e_i\rangle)(\sum_j b_j |f_j\rangle)$. Alternatively, this is just the outer product of tensors with components $a_i$ and $b_j$ to obtain a tensor $T_{ij} = a_i b_j$ in the larger vector space of two index tensors.

**Product vectors and entangled vectors**
Any vector $|\alpha\rangle \otimes |\beta\rangle$ in $V \otimes W$ is called a *product vector*. We often write the product vector $|\alpha\rangle \otimes |\beta\rangle$ simply as $|\alpha\rangle |\beta\rangle$ (omitting the $\otimes$). The mapping in eqn. (*) above is not surjective – vectors in $V \otimes W$ that are not product vectors are called *entangled vectors*. We will see that in the formalism of quantum mechanics, physical states are represented by vectors of unit length, and entangled states will play a very important role in quantum computation and information. We will introduce and define them again later, and have a lot more to say about them in due course!

We will be mostly concerned with tensor products of the two-dimensional space $V_2$ with itself (multiple times). For the $k$-fold tensor power we write $\otimes^k V_2 = V_2 \otimes \ldots \otimes V_2$ which is a space of dimension $2^k$ with basis $|i_1\rangle \otimes \ldots \otimes |i_k\rangle$ $(i_1, \ldots i_k = 0, 1)$ labelled by the $2^k$ $k$-bit strings $i_1 \ldots i_k$. We often write $|i_1\rangle \otimes \ldots \otimes |i_k\rangle$ simply as $|i_1 \ldots i_k\rangle$. This basis is also called the computational (or standard) basis of $\otimes^k V_2$.

**Example.** For $k = 2$ if $|v\rangle = a|0\rangle + b|1\rangle$ and $|w\rangle = c|0\rangle + d|1\rangle$ , we have $|v\rangle |w\rangle \in V_2 \otimes V_2$. By formal multiplication we get

$$|v\rangle \otimes |w\rangle = (a|0\rangle + b|1\rangle)(c|0\rangle + d|1\rangle) = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

and in terms of components we have

$$|v\rangle \otimes |w\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} = \begin{pmatrix} a \begin{pmatrix} c \\ d \end{pmatrix} \\ b \begin{pmatrix} c \\ d \end{pmatrix} \end{pmatrix}.$$

Note how the last expression gives the pattern for how to get the final tensor product components from those of the individual vectors: take each numerical component of the first vector in turn and "expand it up (doubling it to two numbers)" by multiplying it by the components of the second vector taken in order, and then list all these in order in a column. Note also that this illustrates that the tensor product is not commutative i.e. $|v\rangle \otimes |w\rangle \neq |w\rangle \otimes |v\rangle$ in general. $\square$

**Example.** The vector $|00\rangle + |11\rangle \in V_2 \otimes V_2$ is entangled i.e. it is not a product vector. To see this in an elementary way, suppose that it is a product i.e.

$$|00\rangle + |11\rangle = a|0\rangle + b|1\rangle)(c|0\rangle + d|1\rangle) = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

for some $a, b, c, d$. Then comparing the first and last expressions, we must have $ad = 0$ (and also $bc = 0$), so either $a = 0$ or $d = 0$. Thus respectively either $|00\rangle$ or $|11\rangle$ has coefficient zero too, which is a contradiction.
This argument may be generalised to show that an arbitrary vector $\alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle \in V_2 \otimes V_2$ is entangled iff $\alpha\delta - \beta\gamma \neq 0$ (see exercise sheet 1). But beware: this simple single-equation characterisation no longer suffices if the component spaces have dimension greater than 2. Indeed in that general case, a vector $\sum_{i,j} A_{ij}\,|i\rangle\,|j\rangle$ is a product vector iff the matrix $A_{ij}$ of coefficients has rank one. □

**Inner product in $V \otimes W$**
The inner products on $V$ and $W$ give a natural inner product on $V \otimes W$ defined "slot-wise" (the slots being the component spaces). Thus for product vectors we have the inner product of $|\alpha_1\rangle\,|\beta_1\rangle$ with $|\alpha_2\rangle\,|\beta_2\rangle$ being $\langle\alpha_1|\alpha_2\rangle\langle\beta_1|\beta_2\rangle$. This extends to general (entangled) vectors by linearity since general vectors are always linear combinations of product vectors (e.g. of the product basis vectors $|e_i\rangle\,|f_j\rangle$). More explicitly, (and here using the summation convention for repeated indices) if $|A\rangle = a_{ij}\,|e_i\rangle\,|f_j\rangle$ and $|B\rangle = b_{ij}\,|e_i\rangle\,|f_j\rangle$ are two vectors in $V \otimes W$ then

$$\langle A|B\rangle = \big(\, a^*_{ij}\,\langle e_i|\,\langle f_j|\,\big)\;\big(\, b_{pq}\,|e_p\rangle\,|f_q\rangle\,\big) = a^*_{ij} b_{pq}\,\langle e_i|e_p\rangle\,\langle f_j|f_q\rangle = a^*_{ij} b_{ij}$$

where we have used the basis orthonormality relations $\langle e_i|e_p\rangle = \delta_{ip}$ and $\langle f_j|f_q\rangle = \delta_{jq}$.

Thus inner products are calculated by contracting indices on the vector components (relative to an orthonormal product basis), also with a complex conjugation for the left vector. Later (preceeding (QM4) below) we will introduce a useful 'partial inner product' construction for $|A\rangle \in \otimes^m V_2$ and $|B\rangle \in \otimes^n V_2$ with $m \leq n$ in which the (complex conjugated) components of $|A\rangle$ are contracted against a specified set of $m$ indices of $|B\rangle$ to result in a vector in $\otimes^{(n-m)} V_2$.

We often write the bra vector of a product ket $|\alpha\rangle\,|\beta\rangle \in V \otimes W$ as $\langle\beta|\,\langle\alpha|$ ("reflecting" the symbols) with order of spaces reversed. But however we write it, it will generally be important to remain aware of which vector space corresponds to which slot. If needed, we can make this explicit by using subscripts to denote the names of the spaces e.g. writing the bra vector of $|\alpha\rangle_V\,|\beta\rangle_W$ as ${}_W\langle\beta|\;{}_V\langle\alpha|$ or ${}_V\langle\alpha|\;{}_W\langle\beta|$.

## Quantum principles (QM1) and (QM2)

Our description of quantum mechanics below may at first sight look a little different from standard textbook presentations but in fact it's equivalent. Here we focus on quantum mechanics of physical systems with *finite* dimensional state spaces (multi-qubit systems, cf below) and unitary matrices representing finite time evolutions, whereas quantum physics textbooks traditionally begin with the infinite dimensional case viz. wavefunctions, and Schrödinger's wave equation giving infinitesimal time evolution via a Hamiltonian. We will also emphasise *ab initio* the quantum measurement formalism, which will be of crucial significance for us.

**(QM1) (physical states)**: the states of any (isolated) physical system are represented by unit vectors in a complex vector space with an inner product. □

By slight abuse of terminology we will often say that "a system has state space $V$ (of

some dimension $d$)" when its states are the unit vectors in the vector space $V$.

The simplest non-trivial quantum system has a 2 dimensional vector space. Choosing a pair of orthonormal vectors and labelling them $|0\rangle$ and $|1\rangle$, the general state can be written

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle \qquad |a|^2 + |b|^2 = 1.$$

We say that $|\psi\rangle$ is a *superposition* of states $|0\rangle$ and $|1\rangle$ with *amplitudes* $a$ and $b$.

**Qubits**: any quantum system, with a 2 dimensional state space and a chosen orthonormal basis (which we write as $\{|0\rangle, |1\rangle\}$) is called a *qubit*. The basis states $|0\rangle, |1\rangle$ are called computational basis states or standard basis states. They will be used to represent the two corresponding classical bit values as qubit states, and then general qubit states can be thought of as superpositions of the classical bit values (cf later for how we can think of a superposition as a kind of "simultaneous or parallel physical existence" of bit values 0 and 1). There are many real physical systems that can embody the structure of a qubit, for example the spin of an electron, the polarisation of a photon, superpositions of two selected energy levels in an atom etc.

**Example**. For a single qubit, the orthonormal states $|0\rangle$ and $|1\rangle$ give a quantum representation of the classical bit values 0 and 1. Another pair of orthonormal states that we will frequently encounter in applications is the following pair, labelled by plus and minus signs:

$$|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \qquad |-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right).$$

They are "equally weighted superpositions" in the sense that the squared amplitudes of 0 and 1 are equal in each state. The basis $\{|+\rangle, |-\rangle\}$ is called the conjugate basis (and the states themselves are called the conjugate basis states).

**(QM2) (composite systems)**: if system $S_1$ had state space $V$ and system $S_2$ has state space $W$ then the joint system obtained by taking $S_1$ and $S_2$ together, has states given by arbitrary unit vectors in the *tensor product* space $V \otimes W$. $\square$

**Fundamental example**: ($n$ qubits) A system comprising $n$ qubits thus has state space $\otimes^n V_2$ of dimension $2^n$. An $n$-qubit state $|\psi\rangle$ is a called a *product state* if it is the product of $n$ single-qubit states $|\psi\rangle = |v_1\rangle\,|v_2\rangle \ldots |v_n\rangle$ and $|\psi\rangle$ is called *entangled* if it is not a product state.

As we've mentioned previously, the *computational basis* or *standard basis* for $n$ qubits is given by the tensor products of $|0\rangle$'s and $|1\rangle$'s in each slot, giving the $2^n$ orthonormal vectors $|i_1\rangle\,|i_2\rangle \ldots |i_n\rangle$ where each $i_1, \ldots i_n$ is 0 or 1. Thus the basis vectors are labelled by $n$-bit strings and we often write $|i_1\rangle\,|i_2\rangle \ldots |i_n\rangle$ simply as $|i_1 i_2 \ldots i_n\rangle$.

We note the significant fact that as the number of qubits grows *linearly*, the full state description (given as the full list of amplitudes) grows *exponentially* in its complexity. However the description of any product state grows only linearly with $n$ (each successive $|v_i\rangle$ is described by two further amplitudes) so this exponential complexity of state description is intimately related to the phenomenon of entanglement that arises for tensor products of spaces. With this in mind, it is especially interesting to contrast (QM2) with

its classical counterpart – for *classical* physics, the state space of a composite system is the *cartesian* product of the state spaces of the constituent parts. Thus if classical system $S$ requires $K$ parameters for its state description then a composite of $n$ such systems will require only $nK$ parameters i.e. a linear growth of description, in contrast to the exponential growth for quantum systems.

## Dirac notation: linear maps

To illustrate the notation and formalism, we'll consider the case of linear maps on $V_2$ and its tensor powers. With $|v\rangle = a\,|0\rangle + b\,|1\rangle$ and $|w\rangle = c\,|0\rangle + d\,|1\rangle$ in $V_2$, standard matrix multiplication for the "ket-bra" product gives

$$M = |v\rangle\,\langle w| = \begin{pmatrix} a \\ b \end{pmatrix} (c^*\ d^*) = \begin{pmatrix} ac^* & ad^* \\ bc^* & bd^* \end{pmatrix} \tag{1}$$

which is a linear map on $V$ (acting by matrix multiplication on column vectors). In fact for any $|x\rangle \in V_2$ we have $M\,|x\rangle = (|v\rangle\,\langle w|)\,|x\rangle = |v\rangle\,\langle w|x\rangle$, i.e. the vector $|v\rangle$ multiplied by scalar $\langle w|x\rangle$. Such ket-bra products do not give all linear maps from $V_2$ to $V_2$ but only rank 1 mappings (the kernel being the subspace of vectors orthogonal to $|w\rangle$).

For general mappings on $V_2$ note that

$$|0\rangle\,\langle 0| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad |0\rangle\,\langle 1| = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \text{etc.}$$

so if $A : V_2 \to V_2$ is any linear map with matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

then we can write

$$A = a\,|0\rangle\langle 0| + b\,|0\rangle\langle 1| + c\,|1\rangle\langle 0| + d\,|1\rangle\langle 1|\,.$$

More formally, this just expresses the fact that $\{|0\rangle\langle 0|\,, |0\rangle\langle 1|\,, |1\rangle\langle 0|\,, |1\rangle\langle 1|\}$ is a basis for the vector space $V_2 \otimes V_2^*$ of linear maps on $V_2$.

Not also from eqn. (1) the calculational fact that an inner product (bra-ket) can be expressed as a trace (of the corresponding ket-bra):

$$\langle w|v\rangle = \text{trace}\ |v\rangle\,\langle w|\,.$$

### Projection operators

An important special case of eq. (1) is when $|v\rangle = |w\rangle$ and $|v\rangle$ is normalised (i.e. $\langle v|v\rangle = 1$). Then $\Pi_v = |v\rangle\,\langle v|$ is the operator of *projection onto* $|v\rangle$, satisfying $\Pi_v\Pi_v = \Pi_v$. The latter property can be seen very neatly in Dirac notation: $\Pi_v\Pi_v = (|v\rangle\,\langle v|)(|v\rangle\,\langle v|) = |v\rangle\,\langle v|v\rangle\,\langle v| = |v\rangle\,\langle v| = \Pi_v$ as $\langle v|v\rangle = 1$. If $|a\rangle$ is any vector orthogonal to $|v\rangle$ then $\Pi_v\,|a\rangle = |v\rangle\,\langle v|a\rangle = 0$. It then easily follows that for any vector $|x\rangle$, $\Pi_v\,|x\rangle$ is the vector obtained by projection of $|x\rangle$ into the one dimensional subspace spanned by $|v\rangle$. Similarly for any vector space $W$ (of any dimension), if $|w\rangle$ is any normalised vector in $W$, then

$\Pi_w = |w\rangle\langle w|$ is the linear operation of projection into the one-dimensional subspace spanned by $|w\rangle$.

More generally, if $E$ is any linear subspace of a vector space $V$ and $\{|e_1\rangle, \ldots, |e_d\rangle\}$ is any orthonormal basis of $E$ (which thus has dimension $d$), then $\Pi_E = |e_1\rangle\langle e_1| + \ldots + |e_d\rangle\langle e_d|$ is the operator of projection into $E$. This property is easily checked by extending the given basis of $E$ to a full orthonormal basis of the whole space $V$. Then by writing any vector $|\psi\rangle$ in $V$ in terms of this basis we readily see that $\Pi_E|\psi\rangle$ is indeed its projection into $E$.

Finally we point out a possible notational confusion: if $|x\rangle = A|v\rangle$ then the corresponding bra vector is given by $\langle x| = (A|v\rangle)^\dagger = |v\rangle^\dagger A^\dagger = \langle v|A^\dagger$. This follows from the fact that taking adjoints of matrix products reverses the product order $(MN)^\dagger = N^\dagger M^\dagger$. Thus for example in the bra-ket inner product construction we can write $\langle a|M|b\rangle$ as $\langle a|x\rangle$ or as $\langle y|b\rangle$ where $|x\rangle = M|b\rangle$ but $|y\rangle = M^\dagger|a\rangle$ (so $\langle y| = \langle a|M$) i.e. the central $M$ in $\langle a|M|b\rangle$ acts as $M$ if viewed as acting to the right, but acts as $M^\dagger$ if viewed as acting to the left i.e. on the ket $|a\rangle$ before it is turned into a bra vector.

**Tensor products of maps**
If

$$B = \begin{pmatrix} p & q \\ r & s \end{pmatrix}$$

is a second linear map on $V_2$ then the tensor product of maps $A \otimes B : V_2 \otimes V_2 \to V_2 \otimes V_2$ is defined by its action on the basis $|i\rangle|j\rangle \to A|i\rangle B|j\rangle$ for $i$ and $j$ being 0,1. Extending this linearly defines $A \otimes B$ on general vectors in $V_2 \otimes V_2$. In particular for product vectors we get $(A \otimes B)(|v\rangle|w\rangle) = A|v\rangle \otimes B|w\rangle$.

The $4 \times 4$ matrix of components of $A \otimes B$ has a simple block form, as can be seen by writing down its action on basis states in components (giving the columns of the matrix of $A \otimes B$). We get the following pattern (similar to our previous pattern for components of tensor products of vectors):

$$A \otimes B = \begin{pmatrix} aB & bB \\ cB & dB \end{pmatrix} = \begin{pmatrix} ap & aq & bp & bq \\ ar & as & br & bs \\ cp & cq & dp & dq \\ cr & cs & dr & ds \end{pmatrix}.$$

Important special cases of tensor product maps are $A \otimes I$ and $I \otimes A$, being the action of $A$ on the first (resp. second) component space of $V_2 \otimes V_2$, leaving the other space "unaffected".

**Example**: for $|\psi\rangle = |00\rangle + |11\rangle$ and $A$ as above, we have

$$\begin{aligned} A \otimes I|\psi\rangle &= (A|0\rangle)|0\rangle + (A|1\rangle)|1\rangle = (a|0\rangle + c|1\rangle)|0\rangle + (b|0\rangle + d|1\rangle)|1\rangle \\ &= a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle. \end{aligned}$$

On the other hand $I \otimes A|\psi\rangle = |0\rangle(A|0\rangle) + |1\rangle(A|1\rangle)$ giving $a|00\rangle + c|01\rangle + b|10\rangle + d|11\rangle$, which is different. □

## Quantum principle (QM3)

**(QM3) (physical evolution of quantum systems)**: any physical (finite time) evolution of an (isolated) quantum system is represented by a *unitary* operation on the corresponding vector space of states. □

Recall that a linear operation $U$ on any vector space is unitary if its matrix has $U^{-1} = U^\dagger$ (where dagger is conjugate transpose). We have the following equivalent characterisations (useful for recognising unitary operations). $U$ is unitary:
iff $U$ maps an orthonormal basis to an orthonormal set of vectors;
iff the columns (or rows) of the matrix of $U$ form an orthonormal set of vectors.

Below (after (QM4)) we will introduce a number of particular unitary operations on one and two qubits that will be frequently used.

## Dirac notation: partial inner products for vectors in $V \otimes W$

For expressing our final quantum principle (QM4) it will be useful to introduce a 'partial inner product' operation on tensor product spaces. Any ket $|v\rangle \in V$ defines a linear map $V \otimes W \to W$ which we call "*partial inner product* with $|v\rangle$". It is defined on the basis of $V \otimes W$ by the formula $|e_i\rangle\,|f_j\rangle \to \langle v|e_i\rangle\,|f_j\rangle$, and on general vectors in $V \otimes W$ by linear extension of its basis action. Similarly for any $|w\rangle \in W$ we get a partial inner product mapping $V \otimes W$ to $V$. If $V$ and $W$ are instances of the same space (e.g. we will often have them both being $V_2$) then it is important to specify (e.g. with a subscript label on the kets) which of the two spaces is supporting the bra-ket construction of the inner product.

**Example.** For $|v\rangle \in V$ and $|\xi\rangle \in V \otimes V$ we can form the partial inner product on the first or second space. To make the position explicit we'll introduce subscripts to label the slots, writing $V \otimes V$ as $V_{(1)} \otimes V_{(2)}$, and writing ${}_1\langle v|\xi\rangle_{12} \in V_{(2)}$ for partial inner product on the first slot, and ${}_2\langle v|\xi\rangle_{12} \in V_{(1)}$ for partial inner product on the second slot.
Thus for example, if $V = V_2$ and $|\xi\rangle = a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle$ then the orthonormality relations $\langle i|j\rangle = \delta_{ij}$ give ${}_1\langle 0|\xi\rangle_{12} = a\,|0\rangle + b\,|1\rangle$ and ${}_2\langle 0|\xi\rangle_{12} = a\,|0\rangle + c\,|1\rangle$ i.e. we just pick out the terms of $|\xi\rangle$ that contain 0 in the first, respectively second, slot. □

The partial inner product operation is in fact just the familiar operation of contraction of tensor indices (with a complex conjugation). In index notation (with components always relative to an orthonormal product basis), if $|\xi\rangle \in V \otimes V$ and $|v\rangle \in V$, have components $\xi_{ij}$ and $v_k$ respectively then the partial inner products of $|v\rangle$ with $|\xi\rangle$ on the two slots are respectively $v_i^* \xi_{ij}$ and $v_j^* \xi_{ij}$ (with the summation convention applied i.e. repeated indices are summed).

## 2.2 Quantum measurements

## Quantum principle (QM4) – quantum measurements and the Born rule

In classical physics the state of any given physical system can always in principle be fully determined by suitable measurements on a single instance of the system, while leaving the original state intact. In quantum theory the corresponding situation is bizarrely

different – quantum measurements generally have only probabilistic outcomes, they are "invasive", generally unavoidably corrupting the input state, and they reveal only a rather small amount of information about the (now irrevocably corrupted) input state identity. Furthermore the (probabilistic) change of state in a quantum measurement is (unlike normal time evolution) not a unitary process. Here we outline the associated mathematical formalism, which is at least, easy to apply.

**The basic Born rule – complete projective (or von Neumann) measurements**
Suppose we are given a (single physical instance of a) quantum state for a system with state space $V$ of dimension $n$. Let $\mathcal{B} = \{|e_1\rangle, \ldots, |e_n\rangle\}$ be any orthonormal basis of $V$ and write $|\psi\rangle = \sum a_i |e_i\rangle$. Then we can make a *quantum measurement of* $|\psi\rangle$ *relative to the basis* $\mathcal{B}$. This is sometimes called a (complete) von Neumann measurement or projective measurement. The possible outcomes are $j = 1, \ldots, n$ corresponding to the basis states $|e_j\rangle$. The probability of obtaining outcome $j$ is

$$\mathrm{pr}(j) = |\langle e_j|\psi\rangle|^2 = |a_j|^2.$$

If outcome $j$ is seen then after the measurement the state is no longer $|\psi\rangle$ but has been "collapsed" to $|\psi_{\mathrm{after}}\rangle = |e_j\rangle$ i.e. the basis state corresponding to the seen outcome. Stated alternatively: the probability is the squared length of the projection of $|\psi\rangle$ onto the basis state, and the post-measurement state is that projected vector, renormalised to have length 1. Since $|\psi_{\mathrm{after}}\rangle = |e_j\rangle$ corresponding to the seen outcome $j$, if we were to apply the measurement again we will simply see the *same* $j$ with certainty, and not be able to sample the probability distribution $p_i = |a_i|^2$ again.

The qualifier "complete" in "complete projective measurement" refers to the fact that the projections here are into *one*-dimensional orthogonal subspaces (defined by the orthonormal basis). The notion of incomplete projective measurement generalises this to arbitrary decompositions of the state space into orthogonal subspaces (of arbitrary dimension, summing to the dimension of the full space).

**Incomplete projective measurements**
Let $\{E_1, \ldots, E_d\}$ be any decomposition of the state space $V$ into $d$ mutually orthogonal subspaces i.e. $V$ is the direct sum $E_1 \oplus \ldots \oplus E_d$. Let $\Pi_i$ be the operation of projection into $E_i$. Thus $\Pi_i\Pi_i = \Pi_i$ (property of any projection operator) and by orthogonality we have $\Pi_i\Pi_j = 0$ for all $i \neq j$. Then the *incomplete measurement of any state* $|\psi\rangle$ *relative to the orthogonal decomposition* $\{E_1, \ldots, E_d\}$ is the following quantum operation: the measurement outcomes $i = 1, \ldots d$ have probabilities given by the squared length of the projection of $|\psi\rangle$ into $E_i$:

$$\mathrm{prob}\,(i) = \langle\psi|\,\Pi_i\,\Pi_i\,|\psi\rangle = \langle\psi|\,\Pi_i\,|\psi\rangle$$

and the post-measurement state $|\psi_i\rangle$ for outcome $i$ is the ("collapsed") projected vector renormalised to unit length:

$$|\psi_i\rangle = \Pi_i\,|\psi\rangle\,/\sqrt{\mathrm{prob}\,(i)}.$$

A complete projective measurement is thus clearly a special case in which all the subspaces have dimension one. Any incomplete measurement (with orthogonal decomposition $\{E_1, \ldots, E_d\}$) can be refined to a complete one by choosing an orthonormal basis of

the state space that is consistent with the $E_i$'s i.e. each $E_i$ is spanned by a subset of the basis vectors. Then by performing this complete measurement (instead of the incomplete one) we can recover the output probabilities of the incomplete measurement outcomes by summing all the probabilities corresponding to basis vectors in each subspace $E_i$. However the post-measurement states will be different for the incomplete measurement and its refinement.

**Example**. (parity measurement). The parity of a 2-bit string $b_1b_2$ is the mod 2 sum $b_1 \oplus b_2$. The parity measurement on two qubits is the incomplete measurement on the four dimensional state space with two outcomes (labelled 0 and 1), which on the computational basis states corresponds to the parity of the state label. Thus the corresponding orthogonal decomposition is $E_0 = \text{span}\{|00\rangle, |11\rangle\}$ and $E_1 = \text{span}\{|01\rangle, |10\rangle\}$. Upon measurement, the state $|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$ will give outcome 0 with probability $p_0 = |a|^2 + |d|^2$ and the post-measurement state would then be $|\psi_1\rangle = (a|00\rangle + d|11\rangle)/\sqrt{p_0}$. □

**Measurement of "quantum observables"**

In textbooks we often read of measurement of a *quantum observable* $\mathcal{O}$ which is just a slight variation of the above notion of complete or incomplete projective measurement. A quantum observable is defined to be a Hermitian operator $\mathcal{O}$ on $V$. Recall that a Hermitian matrix always has real eigenvalues $\lambda_i$ and the corresponding eigenspaces $\Lambda_i$ (with dimension given by the multiplicity of the corresponding eigenvalue) give an orthogonal decomposition of $V$. The measurement of the quantum observable $\mathcal{O}$ is then just the incomplete measurement relative to that orthogonal eigenspace decomposition of $V$ and in which we label the outcomes by the corresponding eigenvalue $\lambda_i$ (rather than just $i$ or some other label). If the eigenvalues of $\mathcal{O}$ are all non-degenerate, then the measurement will be a complete projective measurement. For the purposes of obtaining information about the state identity, the actual choice of naming of the distinct outcomes is of no real consequence, so we prefer to base our notion of quantum measurement on the underlying orthogonal decomposition of the state space rather than the hermitian observable itself. However, for other purposes the physical observable is important: physical implementation of a measurement involves a physical interaction between the system and a "measuring apparatus" and if for example, the basic $|0\rangle$ and $|1\rangle$ states of our qubit physically are spin-Z eigenstates or photon polarisations or two chosen energy energy levels in a Calcium atom (with corresponding quantum observables being spin, polarisation or energy respectively), this knowledge will have a crucial effect on how the measurement interaction for a standard basis measurement is actually implemented.

**The extended Born rule**

We will often consider measurement of only some part of a composite system, which is in fact just a particular kind of incomplete measurement. The associated formalism for probabilities and post-measurement states is called the extended Born rule and we give an explicit description here (as it will be often used). Suppose $|\psi\rangle$ is a quantum state of a composite system $S_1S_2$ with state space $V \otimes W$. Let $\mathcal{B} = \{|e_1\rangle, \ldots, |e_n\rangle\}$ be an orthonormal basis of $V$. Note that $|\psi\rangle$ can be expanded uniquely as $|\psi\rangle = \sum_i |e_i\rangle |\xi_i\rangle$ with $|\xi_i\rangle$ being vectors in $W$ (not generally normalised nor orthogonal). Indeed orthonormality of the basis gives the $|\xi_k\rangle$'s as the partial inner products $|\xi_k\rangle = \langle e_k|\psi\rangle$. Alternatively if

$\{|f_1\rangle, \ldots, |f_m\rangle\}$ is a basis of $W$ then writing $|\psi\rangle = \sum_{ij} a_{ij} |e_i\rangle |f_j\rangle$ in the product basis of $V \otimes W$, we see that $|\xi_k\rangle = \sum_j a_{kj} |f_j\rangle$ i.e. we just pick out all terms of $|\psi\rangle$ that involve $|e_k\rangle$ in the $V$-slot.

Now we can make a measurement of $|\psi\rangle \in V \otimes W$ relative to the basis $\mathcal{B}$ of $V$. This amounts to an incomplete measurement in $V \otimes W$ relative to its decomposition into mutually orthogonal subspaces $E_i$ given by $E_i = \text{span}\{|e_i\rangle \otimes |\psi\rangle \text{ for all } |\psi\rangle \in W\}$. The extended Born rule asserts the following:
(a) the probability of outcome $k = 1, \ldots, n$ is $\text{pr}(k) = \langle \xi_k | \xi_k \rangle$ i.e. the squared length of the partial inner product $\langle e_k | \psi \rangle$;
(b) if the outcome $k$ is seen then the post-measurement state is the product state

$$|\psi_{\text{after}}\rangle = |e_k\rangle |\xi_k\rangle / \sqrt{\text{pr}(k)}$$

i.e. the $V$-slot is "collapsed" to the seen outcome $|e_k\rangle$ and the $W$-slot retains only the associated vector $|\xi_k\rangle$ but renormalised by $\sqrt{\text{pr}(k)}$ to have unit length. Note that the basic Born rule is just a special case of (a) and (b) with $W$ having dimension 1 and $|\xi_k\rangle = \langle e_k | \psi \rangle$ is just the complex number $a_k$ (viewed as a 1-dimensional vector).

**Fixed choice of basis**:
note that a measurement relative to any general basis $\mathcal{C}$ can be performed by a measurement relative to any a priori fixed basis $\mathcal{B}$ together with some unitary operations; indeed for any two orthonormal bases $\mathcal{B} = \{|e_1\rangle, \ldots, |e_n\rangle\}$ and $\mathcal{C} = \{|e'_1\rangle, \ldots, |e'_n\rangle\}$ there is a unitary transformation $U$ with $|e'_i\rangle = U |e_i\rangle$ for all $i$. Thus to perform a measurement on $|\psi\rangle$ relative to $\mathcal{C}$ we first apply $U^{-1}$ to $|\psi\rangle$, then perform a measurement relative to $\mathcal{B}$, then finally apply $U$ to the resulting post-measurement state, to obtain the same probabilities and post-measurement states as would have been obtained from a $\mathcal{C}$ measurement.

**Standard measurement on multi-qubit systems**
Recall that any $k$-qubit system comes equipped with a standard or computational basis $\mathcal{B}$ of orthonormal states labelled by $k$-bit strings. In this course our measurements will often be restricted to being only relative to this standard basis for some subset of $k$ qubits of an $n$-qubit system.

**Example**. Consider the 3-qubit state

$$|\phi\rangle = \frac{i}{2} |000\rangle + \frac{12 + 5i}{26} |001\rangle - \frac{1}{2} |101\rangle + \frac{3}{10} |110\rangle - \frac{2i}{5} |111\rangle .$$

Computing the partial inner product with $|1\rangle$ on the first qubit we get

$$|\alpha\rangle = {}_1\langle 1 | \phi \rangle = -\frac{1}{2} |01\rangle + \frac{3}{10} |10\rangle - \frac{2i}{5} |11\rangle$$

and its squared length is $\langle \alpha | \alpha \rangle = 1/2$. Hence if we make a standard measurement on the first qubit, the probability of seeing outcome 1 is half, and in that case the state of the three qubits after the measurement will be $|1\rangle \otimes$"$|\alpha\rangle$ normalised" $= \sqrt{2} |1\rangle |\alpha\rangle$. $\square$

**Remark**
According to (QM4), states with guaranteed different measurement outcomes always lie

in orthogonal subspaces of the state space. Consequently two states are reliably physically distinguishable iff the corresponding vectors are orthogonal. Here distinguishability means that there is a measurement which respectively outputs two distinct results, say 0 or 1, *with certainty* when applied respectively to the two states. We will explore consequences of this important non-classical feature much more later! – but we emphasise here that in contrast, in classical physics *any* two different states of a system are in principle distinguishable. □

**Remark**
If $|\psi\rangle$ is an $n$-dimensional state then for all the measurements we have described, there are at most $n$ outcomes. But we can associate properties to $|\psi\rangle$ with more than $n$ outcomes by enlarging the space as follows. We introduce a second quantum system (called an ancilla) of any dimension $m$, in some fixed state $|A\rangle$ independent of $|\psi\rangle$. Then performing a complete projective measurement on the joint system $|\psi\rangle\,|A\rangle$ we will obtain an outcome depending only on $|\psi\rangle$ and having $mn$ possible values. Such measurements have the curious physical feature that generally there will be no states $|\psi\rangle$ having any one of the outcomes with certainty. (why?)

**Remark (optional)**
In the most general form of quantum measurement theory there is a notion of measurement that is more general than the projective measurements we have described. It is called a positive operator valued measurement (or POVM) and it can have any number of outcomes. We will not describe the formalism of POVMs in this course but just say that it can be shown (the so-called Stinespring dilation theorem) that any such POVM is equivalent to a projective measurement in an enlarged space, constructed exactly as in the previous remark, and as such, does not provide any genuinely new features that are not already accessible by projective measurements.

**Remark (optional): global and relative phases.**
If $|v\rangle$ is any unit vector then the states $|v\rangle$ and $e^{i\alpha}\,|v\rangle$ will have the same measurement probabilities (for any basis or orthogonal decomposition), independent of $\alpha$ (since probabilities always depend on squared moduli of amplitudes.) Also under unitary (hence linear) evolution the phase $e^{i\alpha}$ just persists unchanged as a coefficient scalar multiplier. Here $\alpha$ is called a *global phase*. Thus $|v\rangle$ and $e^{i\alpha}\,|v\rangle$ represent identical physical situations and in (QM1) we should (more correctly) have said that states of a physical system correspond to unit vectors *up to an (irrelevant) global phase*. Note also that the projection operator $\Pi_v = |v\rangle\langle v|$ is independent of the choice of global phase for $|v\rangle$ and hence it can also be used to uniquely represent distinct physical systems (not having the global phase ambiguity).
On the other hand $\theta$ in $|0\rangle + e^{i\theta}\,|1\rangle$ is called a *relative* phase and it is a crucially important parameter for the qubit state. Indeed for example, we can think of any unitary operation as evolving $|0\rangle$ and $|1\rangle$ separately and combining the results with relative phase $\theta$ which will affect the way that the two terms interfere (cf below). A notable illustrative example is the pair of states $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. These differ only by a relative phase of $\pi$ but they're easily seen to be orthogonal, so can be distinguished with certainty by a suitable measurement. □

## 2.3 Some basic unitary operations for qubits

Unitary operations on qubits are also called *quantum gates*. Matrices given below are always relative to the standard basis $|0\rangle, |1\rangle$. The following notations for commonly occurring gates will be used throughout the course, and you should memorise them.

### One-qubit gates

$$\textbf{Hadamard gate} \qquad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Thus we have $H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$ and $HH = I$. As an orthogonal transformation in the real Euclidean plane $\mathbb{R}^2$, $H$ is reflection in the mirror line at angle $\pi/8$ to the $x$-axis.

Next, introduce the 1-qubit gates

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad Y = ZX = -XZ = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

$X$ is sometimes also called the (quantum) NOT-gate because it interchanges the kets $|0\rangle$ and $|1\rangle$ i.e. effects the classical NOT operation on the label.

Note that $\{|+\rangle, |-\rangle\}$ is the $X$-eigenbasis and $\{|0\rangle, |1\rangle\}$ is the $Z$-eigenbasis (in each case corresponding to eigenvalues $+1$ and $-1$ respectively). We also have the formulas

$$X|k\rangle = |k \oplus 1\rangle \qquad Z|k\rangle = (-1)^k |k\rangle \qquad \text{for } k = 0, 1.$$

The **Pauli operations** are

$$\sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = -iY = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

They have elegantly simple multiplicative properties:

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I$$
$$\sigma_x\sigma_y = -\sigma_y\sigma_x = i\sigma_z \qquad \sigma_y\sigma_z = -\sigma_z\sigma_y = i\sigma_x \qquad \sigma_z\sigma_x = -\sigma_x\sigma_z = i\sigma_y$$

(noting the cyclic shift of $x, y, z$ labels in the latter set). Note that the matrices $I, \sigma_x, \sigma_y, \sigma_z$ are all Hermitian as well as unitary (which is an unusual coincidence). Finally we have the

$$\textbf{Phase gate} \qquad P_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}.$$

### Two-qubit gates

**Controlled-$X$ (or controlled-NOT) gate**

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} (I) & (0) \\ (0) & (X) \end{pmatrix}.$$

For the four basis states we can compactly write $CX\,|i\rangle\,|j\rangle = |i\rangle\,|i \oplus j\rangle$ where $\oplus$ denotes addition modulo 2. Note that for any 1-qubit state $|\alpha\rangle$ we have

$$CX\,|0\rangle\,|\alpha\rangle = |0\rangle\,|\alpha\rangle \qquad CX\,|1\rangle\,|\alpha\rangle = |1\rangle\,X\,|\alpha\rangle$$

i.e $CX$ applies $X$ to the second qubit if the first is set to "1" and acts as the identity if the first is set to "0" (and extends by linearity if the first qubit is in a superposition of the two values etc.) Accordingly the first qubit is called the *control qubit* and the second is called the *target qubit*. Note that we get a different 2-qubit gate if we interchange the qubit roles of control and target. Thus if there is an ambiguity as to which qubit is to be the control and target we introduce labels (say 1 and 2) for the two qubit system, writing $CX_{12}$ or $CX_{21}$ with the first subscript always denoting the control qubit and the second, the target qubit. Thus for example $CX_{12}\,|0\rangle_1\,|1\rangle_2 = |0\rangle_1\,|1\rangle_2$ whereas $CX_{21}\,|0\rangle_1\,|1\rangle_2 = |1\rangle_1\,|1\rangle_2$.

The **controlled-$Z$ gate**:

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} (I) & (0) \\ (0) & (Z) \end{pmatrix}$$

i.e. as for $CX$, $CZ$ applies $Z$ to the second qubit controlled by the state of the first qubit. Note that despite this asymmetrical description, $CZ$ (unlike $CX$) is actually symmetric in its action on the two qubits.

**Example**. Although quantum operations on computational basis states generally do not correspond to just Boolean functions on their labels (e.g. the $Z$ and $H$ gates), they can serve to reveal new relationships between such operations that do (like the $CNOT$ operation), with these relations being not expressible within the formalism of Boolean algebra. For example we saw that $CNOT_{12}$ and $CNOT_{21}$ were different operations, corresponding to two Boolean functions of the basis state labels. We can now verify that

$$CNOT_{21} = (H \otimes H)(CNOT_{12})(H \otimes H)$$

i.e. we can reverse the control/target roles of the two bits by applying $H$ to each bit vector both before and after the $CNOT$ action. This relation is not possible to achieve with any classical 1-bit Boolean operations before and after the $CNOT$s.

The validity of this relation can be readily checked e.g. by computing the actions of the gates on each basis state in turn, and we omit the details here (which you can easily provide). □

## 2.4 An aside: Superposition and quantum interference

*This section could be deferred to the second half of the course.*

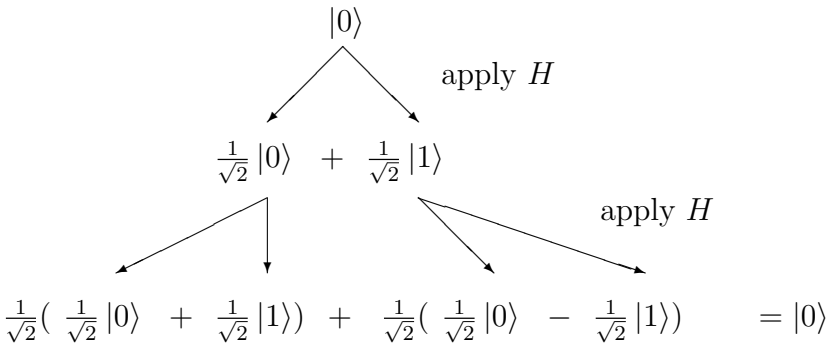
We now ask a kind of interpretational question that will be useful later to guide intuitions about quantum processes. We use the basis kets $|0\rangle$ and $|1\rangle$ as quantum representations of the classical bit values, but then how should we intuitively think of a superposition of $|0\rangle$ and $|1\rangle$?

According to the Born rule, a qubit in a superposition state $a|0\rangle + b|1\rangle$ behaves – *for the purposes of measurement* – just like a qubit which has been prepared in state $|0\rangle$ or $|1\rangle$ with probabilities $|a|^2$ and $|b|^2$ respectively. However it is important to emphasise that a superposition state *cannot* be interpreted as such a probabilistic mixture in more general quantum processes! To see an explicit example, consider the equal superposition state $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and the Hadamard gate:

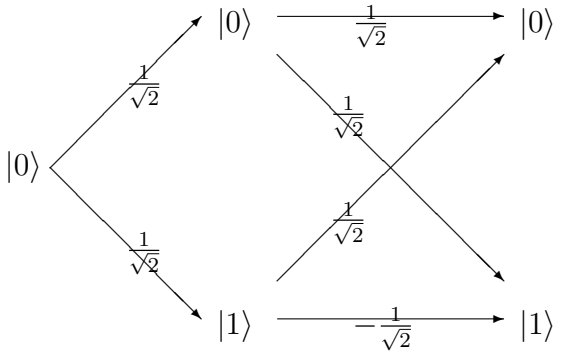
$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

which maps $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. $H$ is also its own inverse so $H$ applied to $|\psi_0\rangle$ gives $|0\rangle$. Hence if we prepare a qubit in state $|\psi_0\rangle$, apply $H$ and then measure it, we will see outcome 0 with certainty. On the other hand if we prepare either $|0\rangle$ or $|1\rangle$ and conduct the same process then in each case we will see outcome 0 and 1 with probabilities half i.e. the equal *probabilistic mixture* of $|0\rangle$ and $|1\rangle$ will behave differently from the equal *superposition* state. In an intuitive sense we think of $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ as "simultaneously having *both* $|0\rangle$ and $|1\rangle$ present" rather than just a probabilistic choice of one or the other. When we apply $H$ to $|\psi_0\rangle$, $|0\rangle$ (starting with amplitude $a = 1/\sqrt{2}$) is transformed to $|0\rangle$ with amplitude $c = 1/\sqrt{2}$ and to $|1\rangle$ with amplitude $c$ whereas $|1\rangle$ (starting with amplitude $a = 1/\sqrt{2}$ also) is transformed to $|0\rangle$ with amplitude $c$ and to $|1\rangle$ with amplitude $-c$. Thus the total amplitude to go from $|\psi_0\rangle$ to $|0\rangle$ is made up of two "paths": $a(c) + a(c) = 1$ which add. On the other hand the total amplitude to go to $|1\rangle$ also has two contributions but these cancel: $a(c) + a(-c) = 0$. In the first case we say that the two paths *interfere constructively* whereas in the latter case we say that they *interfere destructively*.

These notions of "interfering paths" form the basis of **Feynman's sum-over-paths description of quantum mechanics** which we now briefly outline. To illustrate the formalism consider the simple quantum process of applying $H$ twice to $|0\rangle$:




We think of this as a process of "transitions between basis states $|0\rangle$ and $|1\rangle$ with prescribed amplitudes and we depict it as a branching tree, just like a probabilistic process but the branches are labelled by amplitudes, not probabilities:




The rules for accumulating amplitudes are just like those for probabilities:
(i) each path has an amplitude given by the **product** of numbers along the path;
(ii) each final basis state has an amplitude given by the **sum over all paths** from the start to it;
(iii) the probability for the transition from initial $|0\rangle$ to a final basis state is the **modulus square of the sum** over all paths to it.

This is the Feynman sum-over-paths formulation of quantum processes. It turns out to be an alternative equivalent description of the calculations involved in multiplying gate matrices and applying the Born rule for a measurement on the final state of the process. As an illustration, in our example above there are two paths to go from initial $|0\rangle$ to final $|0\rangle$ or final $|1\rangle$. For final $|0\rangle$ the two paths interfere constructively: $|\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}|^2 = 1$ whereas for final $|1\rangle$ the two paths interfere totally destructively: $|\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}|^2 = 0$ and this transition is forbidden.

For *probabilistic* trees we can simulate the whole process by walking through some *single* path of the tree so long as we make an appropriate probabilistic choice of direction at each node along the way. But for quantum amplitude trees this is not possible! In the above example there are non-zero single paths from initial $|0\rangle$ to final $|1\rangle$ yet this transition is forbidden. The process "needs to know about" *all* paths and how they interfere!

Just as in the example above, any quantum process of applying a sequence of unitary gates to an initial starting state of say, $n$ qubits in state $|0\rangle \ldots |0\rangle$ can be represented as a branching tree of amplitudes. The nodes are now labelled by $n$-bit strings organised in columns, such that each column contains each $n$-bit string once. The quantum state at the $k^{\text{th}}$ stage of the process corresponds to the nodes in the $k^{\text{th}}$ column of the tree. The lines connecting column $k$ to column $k+1$ are labelled by the amplitudes of the $k^{\text{th}}$ unitary gate acting on the corresponding basis state in the $k^{\text{th}}$ column. The Feynman sum-over-paths rules then correctly reproduce the calculation of matrix multiplication of the corresponding unitary matrices to determine the amplitudes in the final state.

An important feature of the $n$ qubit process (with depth say of order $n$ too) is that the tree will generally have exponentially many nodes and exponentially many paths from the root to any final node. Suppose we are given a description of the process i.e. the sequence of operations to be applied, that generates the tree. If it were a *classical probabilistic* branching tree, we would still be able to sample the output distribution quite easily – by just "walking through the tree" once, along a single path, making probabilistic choices at each node as we go. But for a tree of *quantum amplitudes*, because of interference as above, to sample the output distribution by classical means (given the description of the process), it does not suffice to choose some single pass through the tree – we need to know about *all* (generally exponentially many!) paths from the root to the chosen final node, in order to be able to sample its probability distribution. On the other hand the quantum process itself perform only one *single* physical pass through the tree, albeit in quantum superposition over all the nodes at each depth. In this sense the quantum physical process achieves something that requires massive classical effort to mimic or simulate i.e. sampling the output distribution by quantum physical means requires exponentially less physical effort (physically just one quantum pass) than by classical means (needing to know the results for exponentially many passes). This feature, expressed intuitively here, is in fact a key effect for the power of quantum computing (compared to classical computing) that we'll be studying in the second half of the course.

# 3 Quantum states as information carriers

Recall from the introduction that information is represented by distinguishable states of a physical system. In classical physics it is axiomatic that any two different states are perfectly distinguishable by a measurement, but in quantum physics, quantum measurements can reliably (i.e. with probability 1) distinguish alternative states only if they lie in orthogonal subspaces of the state space. Hence two (or more) states cannot be reliably distinguished unless they are (pairwise) orthogonal. A quantum system with a $d$-dimensional state space, despite having a continuous infinity of different pure states, can represent at most $d$ reliably distinguishable messages, and a qubit is the simplest quantum system that can be used to represent a classical bit.

**The idea of quantum information**
Pure states are the most definite kind of state that a quantum system can have, and we can reliably *prepare* any desired pure state. But if we receive such a state (of unknown identity) we cannot identify it with certainty yet we are still receiving a kind of definite signal or message, albeit unreadable in the classical sense. We use the term *quantum information* to refer to what we acquire when we receive a quantum state. This turns out to be a very useful concept with many intriguing properties (some of which we will soon see) if thought of as a kind of quantum analogue of familiar classical information. It is a useful intuition being the "stuff" that's processed in quantum computations and communicated over quantum channels. Quantum measurement provides a link back to classical information. A rich theory of its properties and applications has been developed, called quantum Shannon theory, inspired by the formalism and applications of classical information as given in Shannon's classical information theory.

Classical information can be thought of (or represented) as a special case of quantum information in which all quantum states are required to be drawn from a fixed orthonormal set (e.g. computational basis states of a qubit) and its tensor powers.

Given a unknown quantum state $|\psi\rangle$ i.e. some quantum information, there are three basic kinds of operations that we can inflict upon it:

**(Ancilla)**: we can take a fixed (known) quantum state $|A\rangle$ of a second quantum system and adjoin it so $|\psi\rangle$ becomes $|\psi\rangle\,|A\rangle$. $|A\rangle$ is called an *ancilla*. This has the useful effect of embedding our quantum information into a larger dimensional space, that of the joint system, having dimension $d_1 d_2$ where $d_1$ resp. $d_2$ are the dimensions for $|\psi\rangle$ resp. $|A\rangle$.

**(Unitary)**: we can apply a unitary operation $U$ of our choice so $|\psi\rangle$ becomes $U\,|\psi\rangle$.

**(Measure)**: we can perform a measurement on (possibly only part of) $|\psi\rangle$, record the result, and retain the post-measurement state for further processing. The output here will generally be a probabilistic mixture of the possible post-measurement states, with probabilities as given by the Born rule.

The most general quantum action we can apply to $|\psi\rangle$ is then represented by a sequence of these three basic operations.

## 3.1 The no-cloning theorem

We now give our first tantalising property of quantum information which is not shared by classical information: quantum information cannot be copied or 'cloned'!

The copying of classical information is a very familiar process (e.g photocopying) and it is common to view the possibility of copying as obvious and unremarkable. In fundamental terms, given some classical information represented in the state of a classical physical system $A$ (e.g. text ink and paper page) we can take another similar system $B$, initially in some fixed ("blank") state independent of $A$ (blank sheet of paper). We can then carry out a physical operation (operation of the photocopier) on the joint system $AB$ to reliably measure or "read" the state of $A$ (without any corruption) and evolve the state of $B$ to that measured state, to achieve cloning.

Now let us consider similar ideas for quantum information to establish what we'd mean by *quantum* copying or cloning. Our process will involve three quantum subsystems $A$, $B$ and $M$. $A$ will contain our quantum information to be copied; $B$ is system (with state space of same dimension as $A$) which should finally contain the copy that we seek; $M$ will represent any extra "machinery" or physical objects that are needed in the cloning process (like the photocopy machine for classical copying).

Initially $A$ will contain $|\psi\rangle$ and $B$ will contain some standard "blank" state denoted $|0\rangle$; $M$ will also be in some fixed starting state, denoted $|M_0\rangle$ representing the initial "ready" state of the machine and any materials that it may require. A crucially important point here is that the initial state $|0\rangle\,|M_0\rangle$ of $BM$ should be independent of $|\psi\rangle$.

The cloning process will be a fixed quantum physical evolution of $ABM$ achieving the following transformation:

$$|\psi\rangle_A\,|0\rangle_B\,|M_0\rangle_M \longrightarrow |\psi\rangle_A\,|\psi\rangle_B\,|M_\psi\rangle_M$$

i.e. the quantum information is copied into $B$ (while also remaining intact in $A$) and the final state of $M$ is allowed to be arbitrary and even depend possibly on $|\psi\rangle$ (as indicated). The cloning process may be required to work for all states $|\psi\rangle$ of $A$ or alternatively only for some restricted subset of states.

**No-cloning theorem.** Let $\mathcal{S}$ be any set of states of $A$ that contains at least one non-orthogonal pair of states. Then no unitary cloning process exists that achieves cloning for all states in $\mathcal{S}$. □

**Remark.**
The no-cloning theorem actually remains true for *arbitrary* prospective cloning processes, not just unitary ones i.e. even if the further operations of (Ancilla) and (Measure) are allowed to be included. If (Measure) is used then all probabilistic branches are required to lead to perfect cloning. The use of these extra operations can in fact be reduced to the fully unitary case, and in this course we will prove only the unitary case.

For (Ancilla) it is clear that we can just include any needed ancilla into the initial state of $M$ to be used when required.

The case of (Measure) is rather more complicated (optional!):
suppose that during a process we make a (generally incomplete) measurement with $d$ outcomes $i = 1, \ldots, d$ and subsequently apply unitary operations $U_i$ that can even be chosen adaptively to depend on the earlier measurement outcome. Then it can be shown that this gives a final result that is totally equivalent to the following fully unitary substitute:
Let $C$ be an extra ancilla system with $d$-dimensional state space and orthonormal basis $|i\rangle$ labelled by the $d$ measurement outcomes. Let $\Lambda_i$ for $i = 1, \ldots, d$ be the orthogonal subspaces of ABM corresponding to the outcomes of the (generally incomplete) measurement. Let $\{|k\rangle\}_{k\in K}$ be an orthonormal basis of $ABM$ (labelled by elements of some set $K$) that is consistent with the orthogonal decomposition into the $\Lambda_i$'s i.e. each $\Lambda_i$ is spanned by a subset of the $|k\rangle$'s. And for each $k$ let $i(k)$ denote the subspace label (i.e. measurement outcome) to which the basis ket $|k\rangle$ belongs.

Let $|c_0\rangle$ be any fixed chosen state of $C$ and finally consider the mapping $V$ on the states of CABM achieving the following. If $\sum_k a_k |k\rangle$ is any state of ABM then

$$|c_0\rangle \sum_k a_k |k\rangle \longrightarrow \sum_k a_k |i(k)\rangle |k\rangle .$$

One can check that this mapping, depending only on the description of the measurement, is unitary. In words, for any state of ABM written in the $|k\rangle$ basis, $V$ attaches an extra label (in the ancilla state) to each basis state corresponding to its corresponding measurement outcome. We can view $C$ physically as a pointer system for the measurement and the state on RHS above represents the superposition of all measurement outcomes (after collecting all terms with the same $i(k)$ for each measurement outcome $i$) having the associated measurement result attached as an extra label i.e. it represents the result of the measurement before probabilistic collapse to some one outcome occurs.

Finally back to our task: to replace the measurement on ABM by a unitary process, we adjoin the ancilla $C$ and we replace the measurement by the unitary operation $V$ above. Subsequent unitary operations $U_i$ on $ABM$ that depended on the measurement outcome are then also replaced by the single larger controlled unitary operation on $CABM$, the "multiple controlled operation" (with $C$ being a $d$-dimensional control register) that for each $i$, applies $U_i$ to AMB when $C$ is in state $|i\rangle$ and linear extension to general superposition states of $CABM$. Then we can readily check that if the original process (with the intermediate measurement) achieved cloning (for all possible measurement outcomes) then our new (now fully unitary) process will also achieve cloning, thus reducing the intermediate measurement case to the unitary case. □

**Proof of the no-cloning theorem** (for unitary processes)
Let $|\xi\rangle$ and $|\eta\rangle$ be two distinct non-orthogonal states in $\mathcal{S}$. Then the cloning process must do both the following evolutions:

$$|\xi\rangle_A |0\rangle_B |M_0\rangle_M \longrightarrow |\xi\rangle_A |\xi\rangle_B |M_\xi\rangle_M$$
$$|\eta\rangle_A |0\rangle_B |M_0\rangle_M \longrightarrow |\eta\rangle_A |\eta\rangle_B |M_\eta\rangle_M$$

(for some possibly different states $|M_\xi\rangle$ and $|M_\eta\rangle$ of $M$). Now, any unitary process preserves inner products so the inner product of the two initial states must equal that of

the two final states:

$$\langle\xi|\eta\rangle\,\langle 0|0\rangle\,\langle M_0|M_0\rangle = \langle\xi|\eta\rangle\,\langle\xi|\eta\rangle\,\langle M_\xi|M_\eta\rangle \qquad (*)$$

Taking absolute values in eqn. (*) and using $\langle 0|0\rangle = \langle M_0|M_0\rangle = 1$ we get

$$|\langle\xi|\eta\rangle| = |\langle\xi|\eta\rangle|^2\,|\langle M_\xi|M_\eta\rangle|.$$

Since $|\xi\rangle \neq |\eta\rangle$ and $|\xi\rangle$ is not orthogonal to $|\eta\rangle$ we have $0 \lneqq |\langle\xi|\eta\rangle| \lneqq 1$ and cancelling it gives

$$1 = |\langle\xi|\eta\rangle|\,|\langle M_\xi|M_\eta\rangle|$$

which is a contradiction since $|\langle\xi|\eta\rangle| \lneqq 1$ and $|\langle M_\xi|M_\eta\rangle| \leq 1$. □

**Example.** (cloning and superluminal signalling)
The no-cloning theorem was proved in 1982 independently by D. Dieks and by W. Wootters & W. Zurek. The theorem also appears (at least implicitly) in earlier work of D. Park of 1970 but this went completely unnoticed until recently. The 1982 work arose in response to a proposal by F. Herbert for a method of superluminal (in fact instantaneous) communication using quantum methods. If Herbert's result were correct it would have cast serious doubt on quantum theory as an acceptable physical theory! Fortunately there was an error in Herbert's argument – he had taken for granted without justification or discussion that quantum states could be cloned!

Herbert's method was as follows. Suppose Alice and Bob are distantly separated and they share the entangled state

$$\left|\phi^+\right\rangle = \frac{1}{\sqrt{2}}(\,|0\rangle\,|0\rangle + |1\rangle\,|1\rangle\,)$$

i.e. they each hold one qubit of this 2-qubit state. Note that we can also write (as is easily directly checked):

$$\left|\phi^+\right\rangle = \frac{1}{\sqrt{2}}(\,|+\rangle\,|+\rangle + |-\rangle\,|-\rangle\,)$$

where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ are the Pauli $X$ eigenstates.

Alice wants to communicate a yes/no decision instantaneously to Bob at noon. She does the following.
*Alice's action*:
at noon, for 'yes' she measures her qubit in the standard (i.e. Pauli $Z$) basis $\{|0\rangle, |1\rangle\}$, and for 'no' she measures her qubit in the $X$ basis $\{|+\rangle, |-\rangle\}$.
According to the Born rule, after her 'yes' action, Bob's qubit will be in state $|0\rangle$ or $|1\rangle$ with 50/50 probability; and after her 'no' action, Bob's qubit will be in state $|+\rangle$ or $|-\rangle$ with 50/50 probability (as is immediately clear from the second formula for $|\phi^+\rangle$ above).

*Fact*:
these 'yes' and 'no' preparations of Bob's qubit are completely indistinguishable by any local action (measurement) on Bob's qubit i.e. they each give exactly the same probability distribution of outcomes for any measurement (and also in fact the same as that for Alice

having done no action at noon!) Indeed if $\Pi_i$ is the projection operator for outcome $i$ of a measurement by Bob, then in the 'yes' case, his probability of seeing $i$ is

$$\text{prob}_{\text{yes}}(i) = \frac{1}{2}\langle 0|\,\Pi_i\Pi_i\,|0\rangle + \frac{1}{2}\langle 1|\,\Pi_i\Pi_i\,|1\rangle = \text{trace}\,\Pi_i(\frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2})$$

(since $\Pi_i^2 = \Pi_i$ and $\langle A|B\rangle = \text{trace}\,(|B\rangle\langle A|)$.
On the other hand, in the 'no' case we similarly get

$$\text{prob}_{\text{no}}(i) = \text{trace}\,\Pi_i(\frac{|+\rangle\langle +| + |-\rangle\langle -|}{2}).$$

Now (as can easily be checked by computing the $2 \times 2$ matrices of components) we have

$$\frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2} = \frac{|+\rangle\langle +| + |-\rangle\langle -|}{2}$$

so Bob cannot detect any effect of Alice's attempted signalling!

*But*: Suppose Bob can clone quantum states. Then he proceeds as follows.

*Bob's action*:
Immediately after noon he clones his qubit to make many copies (say one million copies). Then he measures them all in the standard basis to obtain a bit string B of length one million, of the measurement results.

In the 'yes' case, all the qubits will be $|0\rangle$ or all $|1\rangle$ so B will be the string of all 0's or all 1's.

In the 'no' case all qubits will be $|+\rangle$ or all $|-\rangle$. However both of these give a 50/50 outcome of 0 and 1 upon standard basis measurement so B will now have a uniformly random bit string of length one million. This is easily distinguishable from the 'yes' case except with negligibly small probability $2/2^{10^6}$, so with arbitrarily high probability Bob can instantaneously get Alice's message.

The 'Fact' above is actually a special case of the so-called quantum no-signalling principle which we will discuss later. □

Now moving on from no-cloning, we'll consider another property of quantum information:

## 3.2 Distinguishing non-orthogonal states

Suppose we are given an unknown quantum state $|\psi\rangle$ that is promised to be one of two non-orthogonal states $|\alpha_i\rangle$ for $i = 0, 1$, and we wish to determine which one it is i.e. the value of subscript $i$. We have seen that this is impossible to do *with certainty* (which would then e.g. also provide a method for cloning the states!) but can we still obtain *some* information about $i$, and then, how much? Since quantum measurement outputs are generally probabilistic we ask if we can identify the state while allowing some probability of error in the answer, or failure of the process. For example we could just do nothing with the state and randomly guess $i = 0$ or 1, which would always be

correct with probability half. But we can do better than this by performing a quantum measurement on $|\psi\rangle$ to guide our answer.

More formally we consider a state estimation process of the following general kind. Given $|\psi\rangle$ we first adjoin an ancillary system in some fixed state $|A\rangle$ (independent of $|\psi\rangle$) which has the effect of enlarging the state space that we can work in. Then we apply a unitary operation to the joint system and finally a measurement with two outcomes labelled 0 and 1 corresponding to our guess of $|\psi\rangle$ having been $|\alpha_0\rangle$ or $|\alpha_1\rangle$.

**Remark.** (optional)
Using the constructions outlined in the remark after the statement of the no-cloning theorem, it can be shown that the triple of operations above viz. (Ancilla) followed by (Unitary) followed by (Measure) is in fact equivalent to any general process i.e any arbitrary sequence of our three basic operations. Indeed the construction in the previous Remark can be used to replace intermediate measurements by unitary operations and all measurements can be postponed to a single all-encompassing measurement at the end. We will omit the technical details here. □

We can simplify the mathematical description of our process as follows. Adjoining the ancilla state $|A\rangle$ amounts to just converting the discrimination problem from $|\alpha_0\rangle$ vs. $|\alpha_1\rangle$ to $|\alpha_0\rangle|A\rangle$ vs. $|\alpha_1\rangle|A\rangle$ i.e. just another example of two non-orthogonal states, which in fact even have the same inner product. Secondly applying a unitary $U$ to any state $|\xi\rangle$ before a measurement with orthogonal projectors $\Pi_0$ and $\Pi_1$, is equivalent to just performing only a measurement with $U$-rotated orthogonal projectors $\Pi_i' = U^\dagger \Pi_i U$ as these give the same outcome probabilities:

$$\text{prob}\,(i) = (\langle\xi| U^\dagger)\,(\Pi_i)\,(U|\xi\rangle) = \langle\xi|\,(U^\dagger \Pi_i U)\,|\xi\rangle\,.$$

Hence we can recast our state estimation process as just a single measurement: given one of two possible non-orthogonal states $|\alpha_0\rangle$ and $|\alpha_1\rangle$ (which are now the $|\alpha_i\rangle$'s with the ancilla adjoined), perform a single two outcome measurement with projectors $\Pi_0$ and $\Pi_1$ (which are now the $U$-rotated versions of the original measurement).

Some measurements will be better than others by providing the correct answers with higher probability. To formalise this we introduce a definition of success probability $P_S$ for the process to quantify how good it is, and we'll seek to optimise this. In the absence of any prior knowledge about which of the two states $|\alpha_0\rangle$ or $|\alpha_1\rangle$ we will receive we assume a prior probability of half for each. Then the success probability is defined by

$$\begin{aligned} P_S \;=\; & \tfrac{1}{2}\,\text{prob}\,\{\text{process outputs 0 given } |\alpha_0\rangle \text{ was sent}\} + \\ & \tfrac{1}{2}\,\text{prob}\,\{\text{process outputs 1 given } |\alpha_1\rangle \text{ was sent}\} \end{aligned}$$

which by the Born rule becomes

$$P_S = \frac{1}{2}\left(\langle\alpha_0|\,\Pi_0\,|\alpha_0\rangle + \langle\alpha_1|\,\Pi_1\,|\alpha_1\rangle\right).$$

Since $\Pi_0 + \Pi_1 = I$ (the identity operator on the full space) we have $\Pi_1 = I - \Pi_0$ and so

$$P_S = \frac{1}{2} + \frac{1}{2}\text{trace}\;\Pi_0\left(|\alpha_0\rangle\langle\alpha_0| - |\alpha_1\rangle\langle\alpha_1|\right). \qquad (*)$$

The optimal choice of measurement $\{\Pi_0, I - \Pi_0\}$ will be the one that maximises $P_S$ (for the given known pair of states $|\alpha_i\rangle$).

To explicitly identify the optimal measurement let's look in detail at the operator

$$D = |\alpha_0\rangle \langle\alpha_0| - |\alpha_1\rangle \langle\alpha_1| .$$

It has the following properties:
(i) it is hermitian so has a complete basis of eigenstates (and all eigenvalues real).
(ii) For any $|\beta\rangle$ orthogonal to both $|\alpha_0\rangle$ and $|\alpha_1\rangle$ we have $D|\beta\rangle = 0$. Hence $D$ has at most two non-zero eigenvalues, whose eigenvectors must lie in the span of $|\alpha_0\rangle$ and $|\alpha_1\rangle$.
(iii) trace $D = 0$ so the two non-zero eigenvalues sum to 0. Writing them as $+\delta$ and $-\delta$, and corresponding normalised eigenvectors as $|p\rangle$ and $|m\rangle$ respectively, we have

$$D = \delta |p\rangle \langle p| - \delta |m\rangle \langle m| .$$

(iv) We can determine $\delta$ in terms of $|\alpha_0\rangle$ and $|\alpha_1\rangle$ as follows. Working in the 2 dimensional subspace spanned by $|\alpha_0\rangle$ and $|\alpha_1\rangle$, choose a unit vector $\left|\alpha_0^\perp\right\rangle$ orthogonal to $|\alpha_0\rangle$ and write $|\alpha_1\rangle = c_0 |\alpha_0\rangle + c_1 \left|\alpha_0^\perp\right\rangle$. Thus in components relative to the $\{|\alpha_0\rangle, \left|\alpha_0^\perp\right\rangle\}$ basis we have

$$|\alpha_0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |\alpha_1\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \quad \text{with } |c_0|^2 + |c_1|^2 = 1.$$

So

$$D = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \begin{pmatrix} c_0^* & c_1^* \end{pmatrix} = \begin{pmatrix} |c_1|^2 & -c_0 c_1^* \\ -c_1 c_0^* & -|c_1|^2 \end{pmatrix}.$$

A straightforward calculation shows that $\det(D - \delta I) = 0$ has solutions $\delta = \pm|c_1| = \pm \sin\theta$, where we have introduced $\theta$ defined by $\cos\theta = |\langle\alpha_0|\alpha_1\rangle|$.

Now, finally returning to our formula eq. (*) for $P_S$ and substituting our expression for $D$, we get

$$\begin{aligned} P_S &= \tfrac{1}{2} + \tfrac{\delta}{2} \text{trace } \Pi_0 (|p\rangle \langle p| - |m\rangle \langle m|) \\ &= \tfrac{1}{2} + \tfrac{\delta}{2} (\langle p| \Pi_0 |p\rangle - \langle m| \Pi_0 |m\rangle) . \end{aligned}$$

For any projector $\Pi$ and state $|\xi\rangle$ we have $0 \leq \langle\xi| \Pi |\xi\rangle \leq 1$ (since $\langle\xi| \Pi |\xi\rangle$ is the squared length of the projected state $\Pi|\xi\rangle$). Thus we see that $P_S$ achieves its maximum value of $(1 + \delta)/2$ if $\Pi_0$ is chosen to be any subspace that contains $|p\rangle$ (so $\Pi_0 |p\rangle = |p\rangle$) and is orthogonal to $|m\rangle$ (so $\Pi_0 |m\rangle = 0$); and then $\Pi_1 = I - \Pi_0$ will have $\Pi_1 |m\rangle = |m\rangle$. Such a choice of $\Pi_0$ is always possible since $|p\rangle$ and $|m\rangle$ are always orthogonal.

In particular for example if $|\alpha_0\rangle$ and $|\alpha_1\rangle$ are qubit states then we can just work entirely in their two dimensional space and an optimal measurement to discriminate them will be the measurement relative to the eigenbasis of the hermitian operator $D = |\alpha_0\rangle \langle\alpha_0| - |\alpha_1\rangle \langle\alpha_1|$.

The achievable optimal success probability above is known as the Helstrom-Holevo bound for discriminating pure states.

In summary, we have proven

**Theorem** (Helstrom-Holevo bound for pure states): Given one of two equally likely states $|\alpha_0\rangle$ and $|\alpha_1\rangle$ with $|\langle\alpha_0|\alpha_1\rangle| = \cos\theta$, the probability $P_S$ of correctly identifying the state by any quantum measurement process is bounded by $P_S \leq \frac{1}{2}(1 + \sin\theta)$ and the bound is tight (i.e. achieved by a particular choice of measurement). $\square$

**Remark (unambiguous state discrimination)**
Finally we mention that by "changing the rules of the game" we can formulate other interesting kinds of state discrimination tasks, such as so-called *unambiguous state discrimination*. For this task we are again given an unknown one of two states $|\alpha_0\rangle$ and $|\alpha_1\rangle$ and we want to construct quantum measurement process with *three* outcomes called 0, 1 and 'fail' with the following properties:
(i) if measurement outcome 0 occurs then the state was certainly $|\alpha_0\rangle$;
(ii) if measurement outcome 1 occurs then the state was certainly $|\alpha_1\rangle$;
(iii) if measurement outcome 'fail' occurs then our process has failed and we have generally irretrievably lost all information about the given state.
In this scenario we would seek to minimise the average probability of obtaining the third outcome.

Both scenarios fall short of providing reliably perfect discrimination of non-orthogonal states albeit in interestingly different ways: in the first we always get an answer 0 or 1 with the caveat that it may be incorrect, whereas in the second the 0 and 1 answers are always certainly correct but the catch now is that the process sometimes fails, and destroys the state (so we cannot try again!) See exercise sheet 1 for an example of an actual unambiguous state discrimination process.

## 3.3 The no-signalling principle

Consider two parties Alice and Bob separated distantly in space, each holding their own local quantum systems $A$ and $B$ respectively. Suppose they possess a (generally) entangled) joint quantum state $|\phi\rangle_{AB}$ of the joint system (sometimes called a *bipartite state* since there are two subsystems). They each have access only to their respective part of the bipartite state, which they can manipulate locally by quantum actions. Suppose Alice performs a complete measurement on her subsystem $A$. According to the Born rule, for each measurement outcome the state of Bob's system will change instantaneously. If Bob could notice this change then they would be able to communicate instantaneously!

**Example**.
Suppose that the shared bipartite state is the entangled state

$$\left|\phi^+\right\rangle_{AB} = \frac{1}{\sqrt{2}}\left(|00\rangle_{AB} + |11\rangle_{AB}\right).$$

If Alice performs a standard basis measurement on $A$ then Bob's system will be "collapsed" into pure state $|0\rangle$ or $|1\rangle$ corresponding to Alice's outcome, each occurring with probability half. Before the measurement however, $B$ was not in a pure state (but was part of an entangled state). Can Bob notice this change by just local actions on his subsystem? Note also that he does not know Alice's outcome - acquisition of that knowledge would require communication from Alice!

Suppose Bob performs a complete measurement in basis $\{|b_i\rangle : i = 0, 1\}$. By the Born rule, after Alice's measurement he will get

$$\text{prob}(i) = \frac{1}{2}|\langle 0|b_i\rangle|^2 + \frac{1}{2}|\langle 1|b_i\rangle|^2 = \frac{1}{2}\langle b_i|b_i\rangle = \frac{1}{2}$$

(where we have averaged over Alice's two possible outcomes using their probabilities of half each). However before Alice's measurement he would have had

$$\text{prob}(i) = ||\ {}_B\langle b_i|\phi^+\rangle_{AB}\ ||^2 = \frac{1}{2} \quad \text{too.}$$

Thus even though each individual outcome of Alice's measurement will give noticeably different probabilities of $i$ for Bob (viz. $|\langle 0|b_i\rangle|^2$ and $|\langle 1|b_i\rangle|^2$), if Bob does not know Alice's outcome he must average over their probabilities and his ability to notice any change is lost! $\square$

This turns out to be true in full generality: for any bipartite state $|\phi\rangle_{AB}$, no local actions by Alice can be noticed by Bob locally i.e. for any local measurement by Bob, the output probability distribution is always unaffected by any local action by Alice. This is the quantum no signalling principle. It seems bizarrely remarkable that quantum theory appears to involve non-local effects (at the level of state descriptions viz. post-measurement states arising from local actions on composite systems) yet the full quantum formalism conspires to prevent us from being able to harness this nonlocality for communication!

We now give a more formal formulation and proof of the no-signalling principle.

**Local operations on a composite system**

(loc-Unitary): a local unitary operation $U$ by Alice resp. Bob on a bipartite system is mathematically represented as the operator $U_A \otimes I_B$ resp. $I_A \otimes U_B$ on the full state of $AB$ (and here $I$ is the identity operation). Note that any two local unitary operations on disjoint subsystems always commute (as $(U \otimes I)(I \otimes V) = (I \otimes V)(U \otimes I) = U \otimes V$)).

(loc-Ancilla): Alice and Bob can adjoin local ancillary systems $A'$ and $B'$ which simply enlarge their locally held systems.

(loc-Measure): Let $\mathcal{H}_A$, $\mathcal{H}_B$ and $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$ denote the state spaces of $A$, $B$ and $AB$ respectively. If Alice performs a (generally incomplete) local measurement on $A$ corresponding to the decomposition of $\mathcal{H}_A$ into the orthogonal subspaces $E_a$'s labelled by outcomes $a$, then mathematically on the full state space this is represented by the measurement with the orthogonal decomposition $E_a \otimes \mathcal{H}_B$'s of $\mathcal{H}_{AB}$. In particular even a complete measurement on $A$ will be an incomplete measurement of the full system.

If $F_b$'s are the orthogonal subspaces of a local measurement by Bob on $B$ with outcomes $b$, then one can check from the Born rule that the joint probabilities $\text{prob}(a, b)$ obtained from performing both measurements is independent of whether $A$ or $B$ goes first, or whether they measure 'simultaneously', corresponding to the global measurement with orthogonal subspaces $E_a \otimes F_b$ for all pairs $(a, b)$.

**No-signalling theorem**: Suppose Alice and Bob have access to subsystems $A$ and $B$ respectively of a joint state $|\phi\rangle_{AB}$. Then Alice cannot convey any information to Bob by

performing local operations i.e. no local action by Alice can change the output probability distribution of any local quantum process by Bob.

**Proof**: Consider first the basic case of Bob performing a complete measurement on $B$ relative to a basis $\{|b\rangle\}$ labelled by the outcomes $b$. We use this basis of $B$ to express $|\phi\rangle_{AB}$ as

$$|\phi\rangle_{AB} = \sum_b |\xi_b\rangle_A |b\rangle_B . \qquad (*)$$

Here the $|\xi_b\rangle_A$'s are subnormalised states of $A$ given by the partial inner products

$$|\xi_b\rangle_A = {}_B\langle b|\phi\rangle_{AB}.$$

They are sometimes called conditional states of $A$ given $B$'s measurement outcomes $b$, or relative states of $A$, relative to basis states $|b\rangle$ of $B$. By the Born rule, the probability of Bob seeing outcome $b$ in the absence of any local operation by Alice, will be squared length:

$$\text{prob}(b) = \langle \xi_b|\xi_b\rangle = |\, {}_B\langle b|\phi\rangle_{AB}\,|^2.$$

Now suppose Alice performs a complete measurement relative to basis $|a\rangle$'s (labelled by outcomes $a$) and subsequently Bob performs his measurement above. Then the joint probabilities are given by

$$\text{prob}(a,b) = |\, {}_A\langle a|\, {}_B\langle b|\phi\rangle_{AB}\,|^2 = |\langle a|\xi_b\rangle|^2$$

(which actually holds regardless of which time order the local measurements are performed in). Then the marginal probability distribution for $b$ i.e. the distribution that Bob will see, is

$$\begin{array}{rcl} \text{prob}(b) & = & \sum_a \text{prob}(a,b) = \sum_a |\langle a|\xi_b\rangle|^2 \\ & = & \sum_a \langle \xi_b|a\rangle\langle a|\xi_b\rangle \\ & = & \langle \xi_b| \left\{\sum_a |a\rangle \langle a|\right\} |\xi_b\rangle . \end{array}$$

But $\sum_a |a\rangle \langle a| = I_A$ as $\{|a\rangle\}$ is a basis, so $\text{prob}(b) = \langle \xi_b|\xi_b\rangle$, which is the same (cf above) as in the case of Alice not having done anything on her side.

This argument may be readily generalised to Alice and/or Bob performing incomplete measurements too e.g. using the fact that for any orthogonal decomposition $E_i$ of a state space we have $\sum_i E_i = I$. We'll omit the details here.

If Alice performs a local unitary $U$ i.e. $U_A \otimes I_B$ is applied to $|\phi\rangle_{AB}$, then by eq. (*) above this just changes the conditional states to $|\xi'_b\rangle = U|\xi_b\rangle$ and we have $\langle \xi'_b|\xi'_b\rangle = \langle \xi_b|\xi_b\rangle$ so Bob's probabilities are again unchanged.

Finally if Alice and Bob include extra local ancillas this just has the effect of enlarging their local spaces and the above arguments go through unchanged in this albeit enlarged scenario. □

**Remark** (communication complexity)
The fact that quantum theory appears to include non-local effects has a long history going back at least to the iconically influential 1935 'EPR' paper by A. Einstein, B. Podolsky and N. Rosen. Later in the 1960's J. Bell introduced what are now known as

the Bell inequalities, providing a much simplified and experimentally accessible way of demonstrating the non-local effects. But (perhaps because of the no-signalling property) these effects were largely ignored by "serious physicists" and viewed as just an awkward curiosity or inconvenience. Then early in the 1990's something remarkable happened: it was realised that if Alice and Bob shared entangled states *and were also allowed classical communication too*, then although entanglement *by itself* cannot provide communication (by no-signalling) it can nevertheless greatly *assist* (when used alongside classical communication) by greatly reducing the amount of classical communication needed to achieve some distributed tasks, involving inputs from both Alice and Bob. In some cases the amount of classical communication could be reduced by a massive exponential amount at the expense of a modest consumption of entangled states used alongside. As a result, a whole new research area, called quantum communication complexity was born.

The basic scenario is the following: Alice and Bob possess separate $n$-bit strings $x$ and $y$ and they wish to compute some joint function $f(x, y)$ of both strings. Clearly they'll need to communicate (e.g. at least the results of some intermediate local calculations) and $n$ bits of communication each way always suffices (they just exchange the information of their strings, and each can then compute $f$ locally). It is now known that for some $f$'s, if Alice and Bob share some entanglement (e.g. some $|\phi^+\rangle$ states), then $f$ can be computed using exponentially less classical communication than is possible by any method involving just classical communication. (and for other $f$'s entanglement provides no help at all). Thus a communication network having shared entanglement along its connections (a 'quantum internet') can solve some distributed computing tasks with exponentially less classical bit traffic across the network. Correspondingly entanglement is now recognised to be a preciously valuable communication resource. In so-called quantum teleportation (cf below) we'll see another communication use for entanglement, this time for the task of communicating *quantum* states. □

## 3.4 Quantum dense coding

We have seen that an individual qubit (e.g. if received as a quantum message) can reliably encode only a single classical bit, corresponding to having a maximum of two mutually orthogonal states. Quantum dense coding is a way of doubling this information capacity: a receiver can reliably extract two classical bits from a received single qubit if he is already in possession of another qubit with which the newly received qubit had previously been entangled. The protocol is very simple, and based on an important orthonormal set of 2-qubit states, the so-called Bell states, which we first introduce.

**Bell states**
The Bell states (named after the physicist J. S. Bell) are the following four orthonormal entangled states of two qubits, (usually denoted in the literature as $|\phi^{\pm}\rangle$ and $|\psi^{\pm}\rangle$):

$$\begin{aligned} |\phi^+\rangle &= \tfrac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \\ |\phi^-\rangle &= \tfrac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right) \\ |\psi^+\rangle &= \tfrac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right) \\ |\psi^-\rangle &= \tfrac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right). \end{aligned}$$

This notation is standard and you should memorise it. Orthonormality is easily checked directly. The 2-qubit basis $\mathcal{B} = \{|\phi^+\rangle, |\phi^-\rangle, |\psi^+\rangle, |\psi^-\rangle\}$ is called the *Bell basis*. A quantum measurement on two qubits relative to this basis is called a *Bell measurement*.

In physical terms, if $|0\rangle$ and $|1\rangle$ are the spin $+1/2$ and spin $-1/2$ states of a spin half particle (say in the $Z$ direction) then $|\psi^-\rangle$ is the spin-zero singlet state of two spin half particles and the other three Bell states span the 3-dimensional spin-1 triplet space.

In terms of our basic 1-qubit operators $I, X, Y, Z$ it is again easy to check that:

$$\begin{aligned} |\phi^+\rangle &= I \otimes I\,|\phi^+\rangle \\ |\phi^-\rangle &= Z \otimes I\,|\phi^+\rangle &&= I \otimes Z\,|\phi^+\rangle \\ |\psi^+\rangle &= X \otimes I\,|\phi^+\rangle &&= I \otimes X\,|\phi^+\rangle \\ |\psi^-\rangle &= Y \otimes I\,|\phi^+\rangle &&= -I \otimes Y\,|\phi^+\rangle \end{aligned}$$

and that for example, $|\phi^+\rangle$ can be prepared from a simple computational basis state via $|\phi^+\rangle = (CX)(H \otimes I)\,|00\rangle$.

**The quantum dense coding protocol**
Alice and Bob (distantly separated in space) each possess one qubit of a $|\phi^+\rangle$ state. In order to reliably communicate two classical bits to Bob by sending him only a single qubit, Alice first locally applies the operation $I, Z, X$ or $Y$ to her qubit, to represent the messages 00, 01, 10 or 11 respectively, and then sends her qubit over to Bob. On receiving Alice's qubit Bob simply performs a Bell measurement on the two qubit which he now holds, to reliably read out Alice's 2-bit message.

We will see the Bell measurement again below when we discuss quantum teleportation.

# 4 Quantum teleportation

Consider again our participants Alice and Bob who are distantly separated in space, and suppose each possesses one qubit of the entangled Bell state

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Suppose Alice has another qubit in some state $|\alpha\rangle$ and she wants to transfer this qubit state to Bob. How can she achieve this transfer? She may not even know the identity of the state $|\alpha\rangle$ and according to quantum measurement theory she is unable to learn more than a small amount of information about it before totally destroying it! She can place the (physical system embodying the) qubit state in a "box" and physically carry it across to Bob. But is there any other way? What if the space region in between A and B is a hostile and dangerous place?

Quantum teleportation provides an alternative method for state transfer that utilises the entanglement in the state $|\phi^+\rangle$. As we'll see precisely in a moment, but speaking intuitively now, the state transfer from A to B is achieved "without the state having to pass through the space in between" in the following sense: the transference is unaffected by any physical process whatever that takes place in the intervening space. Note that this is also a feature of the entanglement of $|\phi^+\rangle$: although quantum theory appears to imply the existence of some kind of "non-local connection" between entangled particles (e.g. reflected in correlations between local measurement results, cf exercise sheet 1), the entanglement ("non-local connection") itself remains entirely unaffected by any physical process occurring in the space in between; it can change only by physical actions on the particles themselves.

Let qubit 1, qubit 2 and qubit 3 denote respectively Alice's input qubit (in state $|\alpha\rangle$), Alice's qubit of $|\phi^+\rangle$ and Bob's qubit of $|\phi^+\rangle$. Using subscripts to label the qubits the starting state can be written $|\alpha\rangle_1 |\phi^+\rangle_{23}$ with 1,2 in A's possession and 3 in B's possession. If

$$|\alpha\rangle = a\,|0\rangle + b\,|1\rangle$$

then we explicitly have

$$\begin{aligned} |\alpha\rangle\,|\phi^+\rangle &= (a\,|0\rangle + b\,|1\rangle)\tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ &= \tfrac{a}{\sqrt{2}}\,|000\rangle + \tfrac{a}{\sqrt{2}}\,|011\rangle + \tfrac{b}{\sqrt{2}}\,|100\rangle + \tfrac{b}{\sqrt{2}}\,|111\rangle\,. \end{aligned} \tag{2}$$

The protocol for quantum teleportation comprises the following five steps (i) to (v).

**(i)** Alice applies $CX$ to her two qubits 1 and 2 (with 1 being the control and 2 the target).

**(ii)** Alice applies $H$ to her qubit 1.

**(iii)** Alice measures her two qubits (in the computational basis) to obtain a 2-bit string 00, 01, 10 or 11.

Note that the sequence (i), (ii), (iii) is equivalent to Alice just performing a Bell measurement on the two qubits – indeed the unitary operations in (i) and (ii) simply serve to rotate the Bell basis into the computational basis of the two qubits (as is easily checked).

By calculating the effect of these three operations on eq. (2) we see that each 2-bit string is obtained with equal probability of 1/4 (irrespective of the values of $a$ and $b$, recalling that $|a|^2 + |b|^2 = 1$). Furthermore after the measurements in (iii) we have the following post-measurement states (as you should calculate):

| mmt outcome | post-mmt state |
|---|---|
| 00 | $\|00\rangle \|\alpha\rangle$ |
| 01 | $\|01\rangle X \|\alpha\rangle$ |
| 10 | $\|10\rangle Z \|\alpha\rangle$ |
| 11 | $\|11\rangle XZ \|\alpha\rangle$ |

i.e. Bob's qubit 3 is now disentangled from 1,2 and it is in a state that's a fixed transform of $|\alpha\rangle$, the choice of transform depending only on the measurement outcome and not on the identity of $|\alpha\rangle$ (i.e. not on the $a, b$ values). In fact if the measurement outcome is $ij$ then Bob's qubit will be "collapsed" into the state $X^j Z^i |\alpha\rangle$.

**(iv)** Alice sends the 2-bit measurement outcome $ij$ to Bob (i.e. she sends him 2 bits of classical information).

**(v)** On receiving $ij$ Bob applies the unitary operation $Z^i X^j$ (i.e. the inverse of $X^j Z^i$) to his qubit which is then guaranteed to be in state $|\alpha\rangle$.

This completes the teleportation of $|\alpha\rangle$ from Alice to Bob.

Note that no remnant of any information about $|\alpha\rangle$ remains with Alice. After stage (iii) she is left with only a 2-bit string that has always been chosen uniformly at random (independent of $|\alpha\rangle$) and the 'original' state $|\alpha\rangle$ is always totally destroyed. Thus the teleportation process is fully consistent with the no-cloning theorem, as indeed it must be.

The whole protocol is shown diagrammatically in a spacetime diagram in figure 1. In figure 2 we give an alternative depiction of the protocol as a network of quantum gates. This representation is perhaps more pertinent to computation (rather than communication), using teleportation to transfer qubits between different parts of a quantum memory.

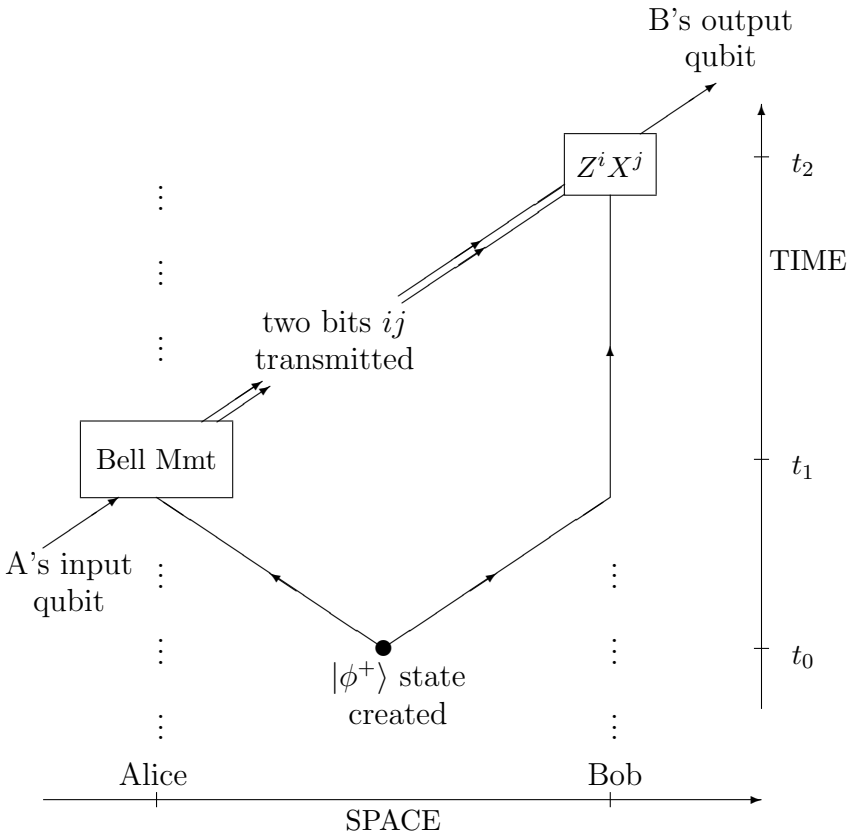



**Figure 1**: *Quantum teleportation. The figure shows a spacetime diagram with the entangled state* $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ *created at* $t_0$ *and subsequently distributed to A and B. At time* $t_1$ *Alice performs a Bell measurement on the joint state of her input qubit and her qubit from* $|\phi^+\rangle$ *and sends the outcome* $ij$ *to Bob. On reception at time* $t_2$*, Bob applies* $Z^iX^j$ *to his particle which is then guaranteed to be in the same state as Alice's original input qubit.*

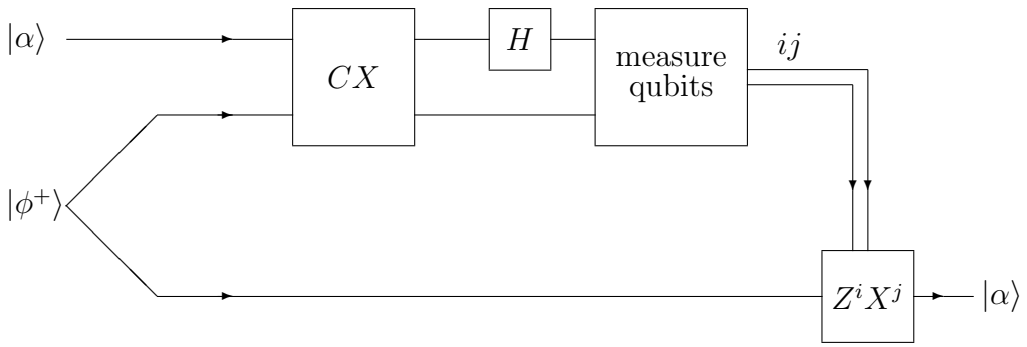



**Figure 2**: *A quantum network diagram for teleportation. The diagram is read from left to right. Horizontal lines represent qubits in a quantum memory. As a result of the above*

*sequence of operations the qubit state is transferred from the top line to the bottom line.*

We conclude this section with a few further remarks about the teleportation process (for possible discussion in lectures).

• Unlike "star-trek" teleportation, the physical system embodying $|\alpha\rangle$ is not transferred from A to B. Only the "information" of the state's identity is transferred, residing finally in a new physical system i.e. the system that was initially Bob's half of $|\phi^+\rangle$.

• Before A's measurements in (iii) Bob's qubit has preparation: "the right half of $|\phi^+\rangle$". After A's measurement Bob's qubit has preparation: " one of the four states $|\alpha\rangle$, $Z|\alpha\rangle$, $X|\alpha\rangle$ or $XZ|\alpha\rangle$ chosen uniformly at random". It can be shown (see exercise sheet 1) that for any measurement process on Bob's qubit, these two preparations give identical probability distributions of outcomes so Bob cannot notice any change at all in his qubit's behaviour as a result of A's measurements. This is just another example of the no-signalling principle in action. Bob can reliably create the qubit state $|\alpha\rangle$ only after receiving the $ij$ message from Alice.

• Figure 1 highlights one of the most enigmatic features of the quantum teleportation process. The question is this: Alice succeeds in transferring the quantum state $|\alpha\rangle$ to Bob by sending him just two bits of classical information. Clearly these two bits are vastly inadequate to specify the state (whose description depends on continuous parameters) so "how does the remaining information get across to Bob?" What carries it? What route does it take? Usually when information is transferred from one location to another, it requires a channel for its transmission! But in figure 1 there *is* clearly another route connecting Alice to Bob (apart from the channel carrying the two classical bits) and it does indeed carry a qubit – it runs *backwards in time* from Alice to the creation of the $|\psi\rangle$ state and then forwards in time to Bob. Hence it is tempting to assert that most of the quantum information of $|\alpha\rangle$ was propagated along this route, firstly backwards in time and then forwards to Bob! In this view, at times between $t_0$ and $t_1$ this part of the state's information was already well on its way to Bob even though Alice had not yet performed her measurement! Such statements may appear paradoxical but further consideration (maybe elaborate in lectures?) shows that, as an *interpretation*, this view is actually entirely consistent and sound. Whether or not you accept this view as a correct description of what *actually* happens in the real physical world, is only a matter of personal preference!

• *To add here – include some mention of practical implementations of teleportation, and maybe discuss the question: "can we teleport a human?"*

# 5 Quantum cryptography – BB84 quantum key distribution

## Introduction

The use of general quantum states to represent information and encode messages would appear, from what we have seen, to have some significant drawbacks compared to the use of classical states! Firstly (a): a received unknown quantum state cannot be reliably identified (unless it has been promised to belong to a specified orthonormal set) so the receiver cannot reliably read the message. Secondly (b): any attempt to read the message (in the context of general signal states) results in only partial information and is always accompanied by irrevocable corruption ("measurement collapse") of the quantum state and correspondingly, part of the sent message is necessarily irretrievably destroyed.

Remarkably these seemingly negative features can be used to positive effect to provide valuable benefits for some cryptographic and information security issues, which in some cases turn out to be impossible to achieve with classical signals. For example, intuitively here, in communication between distant parties, (b) implies that any attempted eavesdropping on the message en route must leave a mark on the signal which can then in principle be detected by actions of receiver in (public) discussion with the sender. It turns out that this can be used to provide communication that's provably secure against eavesdropping. Classical messages on the other hand can always be read en route and sent on to the receiver perfectly intact. Also it turns out (cf more below) that the effects of (a) for the communicators can be circumvented by a suitably clever (non-obvious) protocol involving further (public) discussion between them.

It is now known that quantum effects can provide benefits for a wide variety of cryptographic tasks beyond just secure communication. The associated subject of *quantum cryptography* is currently a highly active and flourishing area of scientific research worldwide, with evident huge practical and theoretical significance for modern society which is becoming increasingly reliant on secure information technology. In this course we will consider only one cryptographic issue, but perhaps the most fundamental of all – that of secure communication between two spacially separated parties traditionally named Alice (A) and Bob (B) with eavesdropper Eve (E). We will discuss the Bennett-Brassard protocol (the so-called BB84 protocol) for *quantum key distribution* which provides a means of communication that's provably secure against eavesdropping. The protocol itself is relatively simple to describe but the proof of unconditional security against general attacks is very involved and technical (beyond the scope of this course). Below we will describe the protocol and be content with making some remarks about its security in some restricted situations.

**How can we communicate securely?**
The issue of secure communication has a long history going back thousands of years. Circa 100 BC Julius Caesar used a cipher in which the letters of the text were simply shifted forward by three places in the alphabet. A more elaborate version of this kind of encryption method (subsequently historically used in a variety of contexts) is to apply

some more general chosen fixed permutation of the alphabet, known securely only to the sender and receiver. However such schemes are insecure (against suitably intelligent adversaries) for example by compiling a table of symbols with their occurring frequencies, and comparing this to a similar table derived from typical texts in the language.

With the development of mathematics (particularly number theory, abstract algebra, group theory, coding theory, computational complexity theory etc.) a variety of more sophisticated (classical) schemes for secure communication were invented but none of these apart from the *one time pad* (which in turn requires a method of *key distribution*) is provably secure. We will discuss the one time pad below as it is also an underpinning ingredient for quantum key distribution (QKD) schemes such as BB84. QKD schemes will be able to circumvent shortcomings of classical key distribution schemes which render them unsuitable in many common situations (cf below).

**Remark** (on public key cryptosystems).
Amongst more sophisticated schemes in common use today are the so-called public key crypto systems (Diffie-Hellman scheme, RSA, elliptic curve cryptography). The security of these schemes is not absolute but relies on (unproven but widely believed) computational hardness assumptions i.e. a belief that certain computational tasks while computable in principle require so much computing time that they are effectively uncomputable in practice. For example given two large prime numbers $p$ and $q$ (of say hundreds of digits each) it is easy to compute their product (e.g. using a very large sheet of paper and careful long multiplication we could even probably do it by hand over a rainy weekend). But conversely, given a composite number $N$ (similarly having hundreds of digits) there is no known "fast algorithm" to factorise it. In fact for $N$ having just several hundreds of digits, and using our best known classical factoring algorithms with all the classical computing power on earth today, it would generally take longer than the age of the universe to complete the task! These kinds of issues are the subject of *computational complexity theory* which we'll see more of in the second half of the course (quantum computation and quantum algorithms). More precisely here, the computational task of multiplication of $n$-digit integers can be completed in a number of steps growing polynomially (quadratically) with $n$, whereas our best classical factoring algorithms for $n$ digit integer factorisation require an exponential (super-polynomial) number of steps, exponential in the cube root of $n$, and this exponential versus polynomial growth in number of digits makes that latter task effectively uncomputable in practice for modest sizes of $n$. Public key cryptosystems exploit such asymmetric (assumed) computational hardness properties of various tasks to provide security. They also have the remarkable very useful feature that the communicating parties do not need any prior secret shared information known only to them (such as knowledge of the permutation in a Caesar-like cypher) so, for example, they need never have previously met. However there are significant drawbacks:
(a) it has not been proven that faster classical algorithms for the tasks may be discovered in the future e.g. a factoring algorithm that requires only a polynomially growing number of steps to complete.
(b) *quantum* computation provides entirely new (non-classical) modes of computing consistent with the laws of physics (as we'll see in the second half of the course). These modes lead to new kinds of algorithms which can be used to solve some computational

problems exponentially faster than any known classical method. And coincidentally, known tasks of this kind include those on which public key cryptosystems are based. So public key crypto systems in common use today could be readily broken if we had a working quantum computer! The most famous such algorithm is Shor's quantum algorithm for integer factorisation (and also computation of discrete logarithms in number theory) discovered by Peter Shor in 1994, which achieves these tasks in a number of (quantum-) computational steps growing only polynomially with $n$ i.e rendering them feasible in practice.

Thus on the one hand quantum physics (via Shor's and other quantum algorithms) allows the breaking of some classical cryptosystems that are not known to be classically breakable, while on the other hand, via quantum key distribution protocols, it offers a method for provably secure communication.

## The one time pad

We assume that our message is a bit string $M$ of length $n$ (without loss of generality e.g. we could represent letters of the alphabet and some punctuation symbols as distinct 5-bit strings).

For the one time pad Alice and Bob need to share a *secret private key* $K$ which is a uniformly random bit string of the same length $n$ as the message, and which is known only to them.

Alice encrypts her message by adding $K$ to $M$. Here addition is addition mod 2 and it is carried out separately at each bit position of the strings. This produces the cryptotext $C = M \oplus K$ which she sends to Bob (over a public classical channel).

Bob receives $C$ and computes $C \oplus K = M \oplus K \oplus K = M$ (with the last equality since $0 \oplus 0 = 1 \oplus 1 = 0$) thus decrypting the message.

**Features and remarks**

If $K$ is uniformly distributed amongst bit strings then so is $C$. Thus any potential eavesdropper Eve can learn nothing about $M$ (apart from its length) by looking at $C$. Hence this scheme cannot be broken, a feature that can be proven more formally in the context of classical information theory, introduced by Claude Shannon in the 1940s.

It is important for security that the key $K$ is used only once (hence the name "one time pad") e.g. if it would be used twice to generate $C_1 = M_1 \oplus K$ and $C_2 = M_2 \oplus K$ then $C_1 \oplus C_2 = M_1 \oplus M_2$ which would generally contain information about $M_1$ and $M_2$ and (with $C_1$ and $C_2$ available) about $K$ too, if Alice were to use $K$ again.

Thus the scheme is rather inefficient in its ongoing needed secret resource, with Alice and Bob needing fresh secret key of length equal to that of each subsequent message. But given that, all seems fine and the key question is: how can Alice and Bob acquire their secret key? It is impossible for two parties to classically generate a secure private key over a public channel. Thus they would need to meet (and carry away e.g. a private one time pad book, for later use) or else use a trusted intermediary to distribute the key. Each of these has evident limitations and potential significant security risks. Quantum

key distribution (QKD) provides a method for Alice and Bob to generate a shared secret key over public classical and quantum channels without the need to meet or to use a trusted intermediary.

In QKD schemes the quantum signals are used to generate the shared secret key rather than to encode the message itself, which is subsequently communicated using the classical one time pad scheme. A variety of quantum key distribution schemes have been proposed including:
(1) BB84 (C. Bennett and G. Brassard 1984), uses four qubit signal states that include non-orthogonal pairs;
(2) B92 (C. Bennett 1992), uses only two (non-orthogonal) quantum signal states;
(3) E91 (A. Ekert 1991), uses (one qubt of ) an entangled pair of qubits in place of the signal states of BB84 which are later created using local measurements by Alice and Bob; and others. We will discuss only BB84.

## The BB84 quantum key distribution protocol

We assume that Alice and Bob can communicate over a public classical channel and they can also send qubits over a quantum channel. Eve also has access to these channels and she wants to acquire information without being detected, about the secret key that Alice and Bob will generate. The bottom line will be that this will be impossible for Eve to achieve by any means whatever consistent with the laws of physics.

For quantum transmissions we will use the following four qubit states

$$\begin{aligned}
|\psi_{00}\rangle &= |0\rangle \\
|\psi_{10}\rangle &= |1\rangle \\
|\psi_{01}\rangle &= |+\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\
|\psi_{11}\rangle &= |-\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle)
\end{aligned}$$

making up two orthonormal qubit bases viz. $\mathcal{B}_0 = \{|\psi_{00}\rangle, |\psi_{10}\rangle\}$ and $\mathcal{B}_1 = \{|\psi_{01}\rangle, |\psi_{11}\rangle\}$. These are the computational basis (or Pauli $Z$ eigenbasis), and the diagonal basis (or Pauli $X$ eigenbasis). These bases are called *mutually unbiassed* since if any state of one basis is measured in the other basis, the outcomes are always equally likely.

We now give the BB84 protocol as a series of steps:

**BB84 Step1:**

Alice generates two uniformly random binary strings $X = x_1x_2 \ldots x_m$ and $Y = y_1y_2 \ldots y_m$, $x_i, y_i \in \{0, 1\}$. Then she prepares $m$ qubits in the states

$$|\psi_{x_1y_1}\rangle |\psi_{x_2y_2}\rangle \ldots |\psi_{x_my_m}\rangle$$

and sends these $m$ qubits over to Bob.
Here $x_i$ will represent the bit value she is trying to send and $y_i$ is her choice of quantum encoding (choice of basis) for that bit. Using such a random choice of mutually unbiassed bases for encoding each bit value is sometimes called conjugate coding.

**BB84 Step 2:**

When Bob receives the $m$ qubits they may no longer be in the states $|\psi_{x_i y_i}\rangle$ that Alice sent, since the quantum channel may have been noisy or eavesdropping may have occurred. To understand how the protocol works let us imagine first that there is no eavesdropping and that the channel is perfectly noiseless i.e. Bob receives precisely the states $|\psi_{x_i y_i}\rangle$ that Alice sent.

Bob chooses a uniformly random bit string $Y' = y'_1 y'_2 \dots y'_m$ and measures the $i^{\text{th}}$ received qubit in basis $\mathcal{B}_{y'_i}$ to get a result $x'_i$ i.e. $y'_i$ is Bob's guess at Alice's choice of encoding basis and $x'_i$ is his guess at A's bit value $x_i$. Let $X' = x'_1 x'_2 \dots x'_m$ be the string of Bob's measurement outcomes. Note that if $y'_i = y_i$ (i.e. Bob correctly guessed Alice's encoding basis) then $x'_i = x_i$ and he will learn her message bit correctly with certainty. But if $y'_i \neq y_i$ then $x'_i$ is completely uncorrelated with $x_i$ (recalling the mutually unbiassed relationship between the bases).

**BB84 Step 3:**

After the completion of Step 2 Alice and Bob publicly reveal and compare their choice of bases i.e. their strings $Y$ and $Y'$ (but they do not reveal the strings $X$ and $X'$!). They discard all bits $x_i$ and $x'_i$ for which $y_i \neq y'_i$ leaving shorter strings of expected length $m/2$. Call these strings $\tilde{X}$ and $\tilde{X}'$. Under our assumptions of no noise and no eavesdropping in the quantum channel, these bit strings would provide the desired outcome of a shared secret key.

[In lecture do an example of Steps 1 to 3.]

In reality there will always be some noise in transmissions and we need to deal with possible eavesdropping too. To address these issues the BB84 protocol concludes with the following steps 4 and 5, which we discuss further below. These are entirely issues and techniques from classical cryptography.

**BB84 Step 4 (information reconciliation):**

Alice and Bob want next to estimate the bit error rate i.e. the number of bits in $\tilde{X}'$ that are not equal to those in $\tilde{X}$. To do this they publicly compare a random sample of their strings (say half of their bits chosen at random positions), and then discard all the announced bits. They assume that the remaining bits have about the same proportion of errors as those checked. Next they want to correct these remaining errors (albeit at unknown positions) to obtain two strings that agree in a high percentage of positions with high probability. Remarkably this can be done (at the expense of sacrificing some more bits) without giving everything away, if the bit error rate is not too large (in fact less than about 11%).

**BB84 Step 5 (privacy amplification):**

From the estimated bit error rate Alice and Bob can estimate the maximum amount of information that an eavesdropper is likely to have obtained about the remaining bits. From this information estimate they use techniques of so-called privacy amplification from classical cryptography to replace their strings by even shorter strings about which the eavesdropper can have practically no knowledge whatever (with high probability).

This concludes the BB84 quantum key distribution protocol.

**Further remarks about information reconciliation and privacy amplification**

A rigorous treatment of the details of Steps 4 and 5 requires much further technical development from classical information theory, the theory of error correcting codes and classical cryptography. A full treatment is beyond the scope of this course and here we will only draw attention to some of the essential ideas.

In Step 5 the bit error rate provides an upper bound on the amount of information that an eavesdropper can have gained because (as we have previously discussed) non-orthogonal states cannot be reliably distinguished and any attempt to acquire information about the state identity certainly involves irreversible state disturbance. In the full theory one can prove information disturbance tradeoff relations that quantify the intuition that more information gain is necessarily accompanied by more disturbance. As a consequence of this fundamental property of quantum information, the amount of Eve's acquired information is reflected in the bit error rate. Of course noise in the channel also generates bit errors but Alice and Bob can reliably upper bound Eve's information by simply assuming that the whole error rate arose from eavesdropping.

There are many ways in which Eve could attempt to acquire information, such as:
(a) the intercept-resend attack: Eve can intercept each transmitted qubit separately, measure it in some chosen basis to acquire some information about it, and then send on the post-measurement state to Bob.
(b) general coherent attack: much more generally and complicatedly, Eve can introduce an auxiliary (possibly very large) probe quantum system $E$ of her own and unitarily interact $E$ with many of the passing qubits. Finally she can measure $E$ to acquire information, which now can be joint information about many of the qubits. Her measurement here can even be postponed until after she overhears Alice and Bob's public discussions in Steps 2,3,4,5 and chosen in response to what she hears.
(c) Note that the standard classical strategy for eavesdropping on classical bits viz. reading them and retaining a copy, and then sending them on perfectly intact, is not available for our quantum state bit encodings because of the no-cloning theorem and the use of non-orthogonal states in the set of encoding states!

Remarkably the privacy amplification techniques of classical information theory can be shown to provide security against any possible eavesdropping strategy that's consistent with the laws of physics.

**Example.** (an intercept-resend attack)
Assume that the quantum channel is noiseless but Eve intercepts each passing qubit and measures it in the so-called Breidbart basis:

$$\begin{aligned} |\alpha_0\rangle &= \cos\tfrac{\pi}{8}\,|0\rangle + \sin\tfrac{\pi}{8}\,|1\rangle \\ |\alpha_1\rangle &= -\sin\tfrac{\pi}{8}\,|0\rangle + \cos\tfrac{\pi}{8}\,|1\rangle\,. \end{aligned}$$

This is a good choice of basis as it lies "midway" between the two BB84 encoding bases: the squared overlaps of $|\alpha_0\rangle$ with the two states $|0\rangle$ and $|+\rangle$ used to encode bit value 0 are equal (being $\cos^2\pi/8$) and similarly for $|\alpha_1\rangle$ with $|1\rangle$ and $|-\rangle$. For any other choice of basis one of these four overlaps will be smaller and intuitively the $|\alpha_i\rangle$'s thus

provide the best (most parallel) simultaneous approximations to the two non-orthogonal states used to encode each bit value $i$; and Eve will learn each bit of $\tilde{X}$ with probability $\cos^2 \pi/8 \approx 0.85$.

Let us compute the bit error rate in the strings $\tilde{X}$ and $\tilde{X}'$ arising from Eve's intervention. For each of the four encoding states (used with probability 1/4) we compute prob($x \neq x'$). Let us denote measurement outcomes by the corresponding basis states and write for example, prob(B gets $|1\rangle \mid |\alpha_0\rangle$) to denote the probability that B's measurement result is 1 given that he received a qubit in state $|\alpha_0\rangle$ etc.
For state $|0\rangle$ sent by Alice in basis $\mathcal{B}_0$ to encode bit value 0, we know that Bob will measure in basis $\mathcal{B}_0$ (i.e. same bases for the tilde strings) but in between Eve will have measured in the Breidbart basis. Thus for this case

$$\begin{aligned}
\text{prob}(x' \neq x) &= \text{prob(B gets } |1\rangle \mid \text{A sent } |0\rangle) \\
&= \text{prob(E gets } |\alpha_0\rangle \mid |0\rangle) \cdot \text{ prob(B gets } |1\rangle \mid |\alpha_0\rangle) \\
&\quad + \text{prob(E gets } |\alpha_1\rangle \mid |0\rangle) \cdot \text{ prob(B gets } |1\rangle \mid |\alpha_1\rangle) \\
&= |\langle\alpha_0|0\rangle|^2 |\langle 1|\alpha_0\rangle|^2 + |\langle\alpha_1|0\rangle|^2 |\langle 1|\alpha_1\rangle|^2 \\
&= \cos^2 \pi/8 \sin^2 \pi/8 + \sin^2 \pi/8 \cos^2 \pi/8 \\
&= \tfrac{1}{4}.
\end{aligned}$$

The other three encoding states similarly give the same result. Thus the eavesdropping will result in a disturbance amounting to a bit error rate of 25%. This is in fact the minimal value over all choices of Eve's basis (cf exercise sheet 2). □

Having estimated the bit error rate in step 4, Alice and Bob now perform information reconciliation to correct the errors (albeit in unknown positions!) in their remaining strings. This can be achieved using techniques from the theory of error correcting codes (that we will not discuss in this course) or other methods from classical cryptography.

(Discuss a method in lectures if time permits?)

Information reconciliation leaks further limited information to Eve and the final result is a pair of shorter strings that are guaranteed to be equal in each position with high probability, but about which Eve may still have information. Finally privacy amplification is performed to produce two final strings of still shorter length, about which is guaranteed with high probability to have no significant information at all.

Eve may have different kinds of information about the string. For example she may know some specific bits, or parities of some subsets of bits, or some other Boolean function of bits, or perhaps probabilistic information e.g. that a particular bit or Boolean function has probability 2/3 of being 0 etc. It is thus prima facie remarkable that privacy amplification can be done at all, without publicly revealing the whole string. Here's an example to illustrate how it can be achieved in a very simple case.

**Example.** (optional)
Note first that if Eve knows a bit value $x$ but not $y$ (i.e. $y$ is uniformly random to her) then the Boolean sum $x \oplus y$ will be uniformly random to her. More generally she will have no knowledge of the parity of a subset of bits if she knows only some (and not all) of the bits (and nothing more).

Now suppose Alice and Bob share a 3-bit string $x = x_1x_2x_3$ and that Eve knows at most one of the bits and nothing else. Then consider the 2-bit string $y = y_1y_2$ constructed as parities of some subsets of the $x$'s:

$$y_1 = x_1 \oplus x_3 \quad y_2 = x_2 \oplus x_3.$$

We claim that Eve then knows nothing about the (shorter) string $y$. Indeed listing all eight possibilities for $x$ and their corresponding $y$'s we see that if we select the four table entries corresponding to any fixed value of any chosen $x_i$, the corresponding four $y$'s are always 00, 01, 10 and 11. Thus if Eve knows any single bit of $x$ (and even has knowledge of Alice and Bob's formulas for the $y_i$'s) she will know nothing about $y$.

The calculation of the $y_i$'s from the $x_j$'s can be written as a Boolean matrix multiplication $y = Gx$ viz.

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

with the rows of the matrix $G$ corresponding to the subsets of $x_j$'s whose parities give the $y_i$'s. More generally in coding theory one can prove that if Eve knows $k$ bits (and nothing else) of an $n$ bit string then a Boolean matrix $G$ (i.e. with $(0, 1)$ entries) mapping $n$ bit strings to $m$ bit strings ($n > m$) will produce secret $y$'s iff the minimum weight of the code generated by $G$ is strictly greater than $k$. □

These ideas can be extended to cover all possible kinds of information that Eve may have, by using the cryptographic method called universal hashing, to produce a short, very secret, string $y$ from a longer but less secret string $x$. One way to achieve this is to select a *random* Boolean matrix $G$ of size $m \times n$ (with $m < n$ determined by the bit error rate, that characterises the extent of Eve's possible information). We can also think of the choice of $G$ as that of $m$ random subsets of the $n$ bit string $x$. Then it can be shown that with high probability Eve will have no significant knowledge of the parities of these subsets.

We have here really just drawn attention to the ideas of information reconciliation and privacy amplification, and their significance for the BB84 QKD protocol. Readers interested in more details should consult the (extensive) classical cryptography literature.

## Practical implementation of quantum key distribution

*To be briefly discussed in lectures.*

# 6 Basic notions of classical computation and computational complexity

We begin by setting down the basic notions of classical computation which will later be readily generalised to provide a precise definition of quantum computation and associated notions of quantum computational complexity classes.

**Computational tasks:**
The **input** to a computation will always be taken to be a bit string. The **input size** is the number of bits in the bit string. For example if the input is 0110101 then the input size is 7. Note that strings from any other alphabet can always be encoded as bit strings (with only a linear overhead in the length of the string). For example decimal integers are conventionally represented via their binary representation.

A **computational task** is not just a single task such as "is 10111 prime?" (where we are interpreting the bit string as an integer in binary) but a whole family of similar tasks such as "given an $n$-bit string $A$ (for any $n$), is $A$ prime?" The output of a computation is also a bit string. If this is a single bit (with values variously called 0/1 or "accept/reject" or "yes/no") then the computational task is called a **decision problem**. For computational complexity considerations (cf later), we will be especially interested in how various kinds of computational resources (principally time – number of steps, or space – amount of memory needed) grow as a function of input size $n$.

Let $B = B_1 = \{0, 1\}$ and let $B_n$ denote the set of all $n$-bit strings. Let $B^*$ denote the set of all $n$-bit strings, for all $n$ i.e. $B^* = \cup_{n=1}^{\infty} B_n$. A subset $L$ of $B^*$ is called a **language**. Thus a decision problem corresponds to the recognition of a language viz. those strings for which the answer is "yes" or "accept" or 1, denoting membership of $L$. For example primality testing as above is the decision problem of recognising the language $L \subseteq B^*$ where $L$ is the subset of all bit strings that represent prime numbers in binary. More general computational tasks have outputs that are bit strings of length $> 1$. For example the task FACTOR($x$) with input bit string $x$ is required to output a bit string $y$ which is a (non-trivial) factor of $x$, or output 1 if $x$ is prime.

**Circuit model of classical computation:**
There are various possible ways of defining what is meant by a "computation" e.g. the Turing machine model, the circuit (or gate array) model, cellular automata etc. Although these look quite different, they can all be shown to be equivalent for the purposes of assessing the complexity of obtaining the answer for a computational task. Here we will discuss only the circuit model, as it provides the easiest passage to a notion of quantum computation.

For each $n$ the computation with inputs of size $n$ begins with the input string $x = b_1 \dots b_n$ extended with a number of extra bits all set to 0 viz. $b_1 \dots b_n 00 \dots 0$. These latter bits provide "extra working space" that may be needed in the course of the computation. A **computational step** is the application of a designated Boolean operation (or Boolean gate) to designated bits, thus updating the total bit string. These elementary steps should be fixed operations and for example, not become more complicated with increasing $n$. We

restrict the Boolean gates to be AND, OR or NOT. It can be shown that these operations are *universal* i.e. any Boolean function $f : B_m \to B_n$ at all can be constructed by the sequential application of just these simple operations. The output of the computation is the value of some designated subset of bits after the final step.

Then for each input size $n$ we have a so-called **circuit** $C_n$ which is just a prescribed sequence of computational steps. $C_n$ depends only on $n$ and not on the the particular input $x$ of size $n$. In total we have a **circuit family** $(C_1, C_2, \ldots, C_n, \ldots)$. We think of $C_n$ as "the computer program" or algorithm for inputs of size $n$. (There is actually an extra technical subtlety here that we will just gloss over: we also require that the descriptions of the circuits $C_n$ should be generated in a suitably simple computational way as a function of $n$, giving a so-called uniform circuit family. This prevents us from "cheating" by coding the answer of some hard computational problem into the changing structure of $C_n$ with $n$.)

**Randomised classical computations:**
It is useful to extend out model of classical computation to also incorporate classical probabilistic choices (for later comparison with outputs of quantum measurements, that are generally probabilistic). This is done in the circuit model as follows: for input $b_1 \ldots b_n$ we extend the starting string $b_1 \ldots b_n 00 \ldots 0$ to $b_1 \ldots b_n r_1 \ldots r_k 00 \ldots 0$ where $r_1 \ldots r_k$ is a sequence of bits each of which is set to 0 or 1 uniformly at random. If the computation is repeated with the same input $b_1 \ldots b_n$ the random bits will generally be different. The output is now a sample from a probability distribution over all possible output strings, which is generated by the uniformly random choice of $r_1 \ldots r_k$. (Thus any output probability must always have the form $a/2^k$ for some integer $a \leq 2^k$). Then in specific computational algorithms we normally require the output to be correct "with suitably high probability", specified according to some desired criteria. This formalism with random input bits can be used to implement probabilistic choices of gates. For example suppose we wish to apply either AND or OR at some point, chosen with probability half. Consider the 3-bit gate whose action is as follows: if the first bit is 0 (resp. 1) apply OR (resp. AND) to the last two bits. Then we use this gate with a random input to the first bit.

## Polynomial time complexity classes P and BPP

In computational complexity theory a fundamental issue is the time complexity of algorithms: how many steps (in the worst case) does the algorithm require for any input of size $n$? In the circuit model the number of steps on inputs of size $n$ is taken to mean the total number of gates in the circuit $C_n$ i.e. the *size* of the circuit $C_n$. Let $T(n)$ be the size of $C_n$, which we also interpret as a measure of the run time of the algorithm as a function of input size $n$. We are especially interested in the question of whether $T(n)$ is bounded by a polynomial function of $n$ (i.e. is $T(n) < cn^k$ for all large $n$ for some positive constants $k, c$?) or else, does $T(n)$ grow faster than any polynomial (e.g. exponential functions such as $T(n) = 2^n$ or $2^{\sqrt{n}}$ or $n^{\log n}$ have this property).

**Remark.** (Notations for growth rates in computer science (CS) literature.)
For a positive function $T(n)$ we write $T(n) = O(f(n))$ if there are positive constants $c$ and $n_0$ such that $T(n) \leq cf(n)$ for all $n > n_0$, i.e. "$T$ grows *no faster than* $f$". We

write $T(n) = O(\text{poly}(n))$ if $T(n) = O(n^k)$ for some constant $k$, i.e. $T$ grows at most polynomially with $n$.
Note that this use of big-$O$ is slightly different from common usage in say calculus, where instead of $n \to \infty$ we consider $x \to 0$ e.g. writing $e^x = 1 + x + O(x^2)$.
In CS usage if $T(n)$ is $O(n^2)$ then it is also $O(n^3)$ but in calculus $O(x^2)$ terms are *not* also $O(x^3)$.
In the CS literature we also commonly find other notations: we write $T(n) = \Omega(f(n))$ to mean $T(n) \geq cf(n)$ for all $n >$ some $n_0$ and some positive constant $c$, i.e. "$T$ grows *at least as fast as* $f$"; and we write $T(n) = \Theta(f(n))$ to mean $c_2 f(n) \leq T(n) \leq c_1 f(n)$ for all $n >$ some $n_0$ and positive constants $c_1, c_2$, i.e. $T$ is both $O(f(n))$ and $\Omega(f(n))$, i.e. "$T$ grows *at* rate $f$".
In this course we will use only the big-$O$ notation (and not $\Omega$ or $\Theta$). $\square$

Although computations with any run time $T(n)$ are computable in principle, poly time computations are regarded as "tractable" or "computable in practice". The term **efficient algorithm** is also synonymous with **poly time algorithm**. If $T(n)$ is not polynomially bounded then the computation is regarded as "intractable" or "not computable in practice" as the physical resource of time will, for fairly small $n$ values, exceed sensibly available limits (e.g. running time on any available computer may exceed the age of the universe).

We have the following standard terminology for some classes of languages (or sometimes these terms are applied to algorithms themselves, that satisfy the stated conditions):

**P** ("**p**oly time"):
class of all languages for which the membership problem has a classical algorithm that runs in polynomial time and gives the correct answer with certainty.

**BPP** ("**b**ounded error **p**robabilistic **p**oly time"):
class of all languages whose membership problem has a classical randomised algorithm that runs in poly time and gives the correct answer with probability at least 2/3 for every input.

The class **BPP** is generally viewed as the mathematical formalisation of "decision problems that are feasible on a classical computer".

**Example:** Let the problem FACTOR$(N, M)$ be the following: given an integer $N$ of $n$ digits and $M < N$, decide if $N$ has a non-trivial factor less than $M$ or not. The fastest *known* classical algorithm runs in time $\exp O(n^{\frac{1}{3}}(\log n)^{\frac{2}{3}})$ i.e. more than exponential in the cube root of the input size. Thus this problem is not known to be in **BPP**.

**Remark** (about the definition of **BPP**)
We have required the output to be correct with probability 2/3. However it may be shown that "2/3" here may be replaced by any other number $1 - \epsilon$ that's strictly greater than half without changing the contents of the class i.e. if there is a poly time algorithm for a problem that succeeds with probability $\frac{1}{2} + \delta$ (for any chosen $\delta > 0$, however small) then there is also a poly time algorithm that succeeds with probability 0.500001 or 0.99999 or indeed $1 - \epsilon$ for any $0 < \epsilon < \frac{1}{2}$ (however small). This result relies on the following fact, sometimes called the amplification lemma (proved using the Chernoff bound for repeated

Bernoulli trials cf Nielsen and Chuang p154 for a simple proof): if we have an algorithm for a decision problem that works correctly with probability $\frac{1}{2}+\delta$ then consider repeating the algorithm $K$ times and taking the majority vote of all $K$ answers as our final answer. Then this answer is correct with a probability at least $1-\exp(-2\delta^2 K)$, approaching 1 exponentially fast in $K$. Thus given any $\epsilon > 0$ this probability will exceed $1-\epsilon$ for some constant $K$, and if the original algorithm had poly running time $T(n)$ then our $K$-repetition majority vote strategy has running time $KT(n)$ which is still polynomial in $n$. $\square$

**Remark.** (Optional. Polynomial space complexity.)
If we replace the computational resource of *time* (i.e. number of gates or elementary computational steps) by that of *space* (i.e. amount of memory or number of bits needed to perform the computation) then we obtain the complexity class **PSPACE**, of all decision problems that can be solved within a polynomially bounded amount of space (as a function of input size) and no imposed restriction on time. It is easy to see that we have the inclusions $\mathbf{P} \subseteq \mathbf{BPP} \subseteq \mathbf{PSPACE}$. Indeed any poly time computation occurs in poly space since poly many one- and two-bit gates can act on at most poly many bits in total. Similarly in any randomised poly time computation, for each fixed choice of the random bits, we can perform the associated computation in poly space. Then doing this sequentially in turn (re-using the same poly space allocation) for each of the exponentially many choices of the random bits, we can keep a running total of accept and reject answers, and thus get $\mathbf{BPP} \subseteq \mathbf{PSPACE}$.
Astonishingly(!?) it is not known whether any of the preceding inclusions are equalities or strict inclusions!

## 6.1 Query complexity and promise problems

In quantum computation (cf later) and the study of its properties relative to classical computation, there is another computational scenario that is often considered. This is the formalism of "black box promise problems" with an associated measure of complexity called "query complexity".

In this scenario, instead of being given an input bit string of some length $n$, we are given as input a *black box* or *oracle* that computes some (here Boolean, but sometimes more general) function $f : B_m \to B_n$. We can query the black box by giving it inputs and this is the only access we have to the function and its values. No other use of the box is allowed. In particular we cannot "look inside it" to see its actual operation and learn information about the function $f$. Thus, at the start, it is unknown exactly which function $f$ is, but there is often an a priori *promise* on $f$ i.e. some stated a priori restriction on the possible form of $f$. Our task is to determine some desired property of $f$ e.g. some feature of the set of all values of $f$. We want to achieve this by querying the box the *least possible number of times*. In our circuits in addition to our usual gates we may use the black box as a gate, each use counting as just one step of computation. The **query complexity** of such an algorithm is simply the number of times that the oracle is used (as a function of its "size" e.g. as measured by $m+n$). In addition to the query complexity we may also be interested in the total time complexity, counting also

the number of gates used to process the answers to the queries in addition to merely the number of queries themselves.

**Example 1** *The following are examples of black box promise problems that will be especially relevant in this course.*

**The "balanced versus constant" problem**
***Input****: a black box for a Boolean function $f : B_n \to B$ (one bit output).*
***Promise****: $f$ is either (a) a constant function ($f(x) = 0$ for all $x$ or $f(x) = 1$ for all $x$) or (b) a "balanced" function in the sense that $f(x) = 0$ resp. 1 for exactly half of the $2^n$ inputs $x$.*
***Problem****: Determine whether $f$ is balanced or constant. We could ask for the answer to be correct with certainty or merely with some probability, say 0.99 in every case.*

**Boolean satisfiability**
***Input****: a black box for a Boolean function $f : B_n \to B$.*
***Promise****: no restriction on the form of $f$.*
***Problem****: determine whether there is an input $x$ such that $f(x) = 1$.*

**Search**
***Input****: a black box for a Boolean function $f : B_n \to B$.*
***Promise****: There is a unique $x$ such that $f(x) = 1$.*
***Problem****: find this special $x$.*

**Periodicity**
***Input****: a black box for a function $f : \mathbb{Z}_n \to \mathbb{Z}_n$*
*(where $\mathbb{Z}_n$ denotes the set of integers mod $n$).*
***Promise****: $f$ is periodic i.e. there is a least $r$ such that $f(x+r) = f(x)$ for all $x$ (and $+$ here denotes addition mod $n$).*
***Problem****: find the period $r$.*

*In each case we are interested in how the minimum number of queries grows as a function of the natural parameter $n$ (for quantum versus classical algorithms).*

# 7 Circuit model of quantum computation

The circuit model of classical computation above has a straightforward generalisation to the quantum setting. For inputs of size $n$ the starting string $b_1 \dots b_n 00 \dots 0$ is replaced by a sequence of qubits in the corresponding computational basis state $|b_1\rangle \dots |b_n\rangle |0\rangle |0\rangle \dots |0\rangle$. A computational step is the application of a *quantum gate* which is a prescribed unitary operation applied to a prescribed choice of qubits. We do not need any randomised input qubits here as for example, a random choice of $|0\rangle$ and $|1\rangle$ can be generated by a measurement on $H|0\rangle$.

For each input size $n$ we have a quantum circuit $C_n$ which is a prescribed sequence of such steps. The output of the computation is the result of performing a quantum measurement

(in the computational basis) on a specified subset of the qubits (this being part of the description of $C_n$).
**Remark**: More generally we could allow measurements along the way (rather than only at the end) and allow the choice of subsequent gates to depend on the measurement outcomes. However it can be shown that this further generality adds nothing extra: any such circuit can be re-expressed as an equivalent circuit in which measurements are performed at the end only. □

A quantum computation or quantum algorithm is defined by a (uniform) family of quantum circuits $(C_1, C_2, \ldots)$.

Classical or quantum circuits can be depicted pictorially as a circuit diagram. Each input bit or qubit is represented by a horizontal line running across the diagram, which is read from left to right. The applied gates are represented by labelled boxes (or other symbols attached to the relevant lines), read in order from left to right.

## Universal sets of quantum gates

In classical computation we restrict our circuits to be composed of gates chosen from a (small) universal set that act on only a few bits each. One such choice is the set {NOT, AND, OR}. Actually OR may even be deleted from this set since $b_1$ OR $b_2$ = NOT(NOT($b_1$) AND NOT($b_2$)).
**Remark** (optional): It may be shown that no sets of 2-bit *reversible* gates are universal (see Preskill p241-2) but there are 3-bit reversible gates $G$ that are universal even just by themselves i.e. any reversible Boolean function may be constructed as a circuit of $G$'s alone, so long as we have available constant extra inputs set to 0 or 1. Two examples of such gates (assuming our starting bit string can also have bits set to 1 in the extra working space) are the Fredkin gate $F(0b_2b_2) = 0b_2b_3$ and $F(1b_2b_3) = 1b_3b_2$ i.e. a controlled SWAP, controlled by the value of the first bit, and the Toffoli gate $\text{Toff}(0b_2b_3) = 0b_2b_3$ and $\text{Toff}(1b_2b_3) = 1CX(b_2b_3)$ i.e. a controlled-controlled-$X$ gate in which $X$ is applied to bit 3 iff the first two bits are 1 and Toff is the identity otherwise. □

### Approximately universal sets of quantum gates

In the quantum case all gates are reversible (unitary) by definition and there are similar universality results but the situation is a little more complicated: quantum gates are parameterised by *continuous* parameters (in contrast to classical gates which form a discrete set) so no finite set can generate them all *exactly* via (even unboundedly large) finite circuits. But many small finite sets of quantum gates are still *approximately universal* in the sense that they can generate any unitary gate with any prescribed accuracy $\epsilon > 0$. Such approximations (for suitably small $\epsilon$) will suffice for all our purposes and for clarity of discussion we will generally ignore this issue of approximability and just allow use of any exact gate that we need.

More precisely, introduce a notion of closeness of unitary operators $U$ and $V$ (on the same space) by defining $||U - V|| \leq \epsilon$ to mean that $\max ||U|\psi\rangle - V|\psi\rangle|| \leq \epsilon$. Here the maximum is taken over all normalised vectors $|\psi\rangle$ and $||..||$ in the maximum is the usual length of vectors. Then a set of quantum gates (acting on qubits) is defined to be *approximately universal* if for any unitary $W$ on any number $n$ of qubits and any $\epsilon > 0$ there is a circuit $C$ of the given gates whose overall unitary action (also denoted

$C$) satisfies $||W - C|| \leq \epsilon$. The set is called *exactly universal* if we have $\epsilon = 0$ in the preceding condition.

For either exactly or approximately universal sets of gates, the size of the circuit $C$ for $W$ will generally be exponential in the number of qubits $n$ on which $W$ acts (but in some important special cases of $W$, it can be poly-sized e.g. notably for the quantum Fourier transform on $n$ qubits, cf later! Another important issue is how the size of $C$ grows with decreasing accuracy parameter $\epsilon$. Here we just quote a fundamental result: the Solovay-Kitaev theorem asserts that if $\mathcal{G}$ is an approximately universal set of gates, then (under some further mild technical conditions on $\mathcal{G}$) the size of $C$ can be taken to be bounded by poly$(\log(1/\epsilon))$ as a function of accuracy parameter i.e. polynomial in the number of digits $(\log(1/\epsilon)$ of accuracy (For a proof see e.g. appendix in Nielsen and Chuang). The degree of the polynomial $p$ here depends (generally exponentially) on the number of qubits $n$ of $W$, but for fixed $n$, $p$ does not depend on the gate $W$ being approximated.

**Remark** (optional)
Some examples of approximately universal sets of quantum gates are the following: $\{CX, \text{all 1-qubit gates}\}$, $\{CX, H, T = \begin{pmatrix} 1 & 0 \\ 0 & \exp i\pi/4 \end{pmatrix}\}$ and $\{\text{Toffoli 3-qubit gate}, H\}$ the latter actually being universal for all gates with *real* entries, which can be shown to suffice for full universal quantum computation i.e. for any quantum circuit there is a corresponding circuit comprising only real gates, that generates the same output probability distribution for any computational basis input. The infinite set $\{CX, \text{all 1-qubit gates}\}$ is actually *exactly* universal too (with continuous parameters provided by the 1-qubit gates). For more details and proofs see Nielsen and Chuang §4.5. □

## Polynomial time quantum computations and BQP

The complexity class **BQP** (**b**ounded error **q**uantum **p**olynomial time) is defined as a direct generalisation of **BPP** viz. **BQP** is the class of languages $L$ such that there is a polynomial time quantum algorithm for deciding membership of $L$ i.e. for each input size $n$ we have a quantum circuit $C_n$ whose size is bounded by poly$(n)$ and for any input string the output answer is correct with probability at least 2/3.

**BQP** is our mathematical formalisation of "computations that are feasible on a quantum computer". From the definitions it can be shown that **BPP** $\subseteq$ **BQP** (any poly sized classical circuit can be replaced by an equivalent circuit of classical reversible gates, still of poly size and the latter is also a quantum circuit albeit comprising gates that preserve the computational basis as a set). Thus with these computational definitions the question "Is quantum computing more powerful than classical computing?" can be expressed formally as "Is **BQP** strictly larger than **BPP**?". This question remains unsolved although it is generally believed that the classes are unequal. For example the decision problem FACTOR$(M, N)$ (viz. does $M$ have a nontrivial factor less than $N$?) is in **BQP** (as we'll see in detail later) but it is not known to be in **BPP** (although we have no proof that it is not in **BPP**!)

More generally we will be especially interested in any kind of computational task that can demonstrate any kind of computational resource benefit (especially an exponential

benefit) for solution by quantum vs. classical computation. Historically the notion of query complexity and promise problems that we introduced above, provided the first source of such examples, and we'll consider some of them as our first quantum algorithms below. But before giving explicit algorithms we need a further result about black boxes (oracles) in the context of quantum vs. classical computations.

## Reversible version of any Boolean function

If $f : B_m \to B_n$ is any Boolean function it can be expressed in an equivalent *reversible* form $\tilde{f} : B_{m+n} \to B_{m+n}$ as follows. We introduce an addition operation, denoted $\oplus$, for $n$-bit strings: if $b = b_1 \dots b_n$ and $c = c_1 \dots c_n$ then $b \oplus c = (b_1 \oplus c_1) \dots (b_n \oplus c_n)$ i.e. $b \oplus c$ is the $n$-bit string obtained by adding mod 2, the corresponding bits of $b$ and $c$ in each slot separately. For example $011 \oplus 110 = 101$. Note that for any $n$-bit string we have $b \oplus b = 0 \dots 0$ where $0 \dots 0$ denotes the $n$-bits string of all zeroes.

Now for any $f : B_m \to B_n$ define $\tilde{f} : B_{m+n} \to B_{m+n}$ by

$$\tilde{f}(b, c) = (b, c \oplus f(b)) \quad \text{for any } m\text{-bit string } b \text{ and any } n\text{-bit string } c.$$

Note that $\tilde{f}$ is easily computable if we can compute $f$ and the (simple) addition operation $\oplus$ on bit strings. Conversely given $\tilde{f}$ we can easily recover $f(b)$ for any $b$ by setting $c = 0 \dots 0$ and looking at the last $n$ bits of output of $\tilde{f}$.

Furthermore we have the key property: for any $f$, $\tilde{f}$ is a *reversible* (i.e. invertibele function on $m + n$ bits. In fact $\tilde{f}$ is always self-inverse i.e. $\tilde{f}$ applied twice is the identity operation (an easy consequence of the fact that $b \oplus b = 00 \dots 0$ for any bit string $b$).

It should be intuitively clear that any classical algorithm using an oracle for $f$ can be equally well performed using an oracle for the reversible version $\tilde{f}$ instead. In quantum computation, gates are always reversible (unitary) by definition so we will always use (a quantum version of) $\tilde{f}$ for any oracle problem involving $f$. More specifically the **quantum oracle for any Boolean function** $f : B_n \to B_m$ will be the quantum gate denoted $U_f$ on $n + m$ qubits, defined by its action on basis states as follows:

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle \qquad \text{for all } x \in B_n \text{ and } y \in B_m$$

i.e. $U_f$ acts exactly like the classical function $\tilde{f}$ on the labels $(x, y) \in B_{n+m}$ of the computational basis states (and it acts on arbitrary states of $n + m$ qubits by linear extension). We sometimes refer to the $n$-qubit register $|x\rangle$ and the $m$-qubit register $|y\rangle$ as the **input** and **output registers** respectively.

**Remark.** $U_f$ as defined above is always guaranteed to be a *unitary* operation. Indeed if $g : B_k \to B_k$ is any *reversible* Boolean function on $k$ bits then it is just a permutation of all $k$-bit strings. Hence the linear map $V$ on $k$ qubits defined by $V |i_1 \dots i_k\rangle = |g(i_1 \dots i_k)\rangle$ will be represented by a permutation matrix in the computational basis i.e. each column is all 0's with a single 1 entry, and different columns have the 1 entry in different rows, so $V$ is unitary. □

## Computation by quantum parallelism

Note that as a quantum operation (in contrast to classical oracles) $U_f$ can act on (jointly) superposed inputs of both registers. Indeed if we set the input register to an equal

superposition of all $2^n$ possible $n$-bit strings we get (by linearity)

$$U_f : \frac{1}{\sqrt{2^n}} \sum_{\text{all } x} |x\rangle |0\rangle \rightarrow |f\rangle \equiv \frac{1}{\sqrt{2^n}} \sum_{\text{all } x} |x\rangle |f(x)\rangle$$

i.e. in *one run* of $U_f$ we obtain a final state which depends on *all* of the function values. Such a computation on superposed inputs is called **computation by quantum parallelism**. By further quantum processing and measurement on the state $|f\rangle$ we are able to obtain "global" information about the nature of the function $f$ (e.g. determine some joint properties of all the values) with just one run of $U_f$, and these properties may be difficult to get classically without *many* classical evaluations of $f$ (as each such evaluation reveals only one further value). This simple idea of running computations in quantum superposition is a powerful ingredient in quantum vs. classical algorithms. In Appendix 1 (at the end of the notes) we discuss some further issues relating to the interpretation of superpositions in quantum computation.

It is instructive to consider more explicitly how we can actually create the input state of a uniform superposition over all $x$ values that is needed in the above process. Recall that $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ so if we apply $H$ to each of $n$ qubits initially in state $|0\rangle$ and multiply out all the state tensor products, we get

$$\begin{aligned} H \otimes \ldots \otimes H(|0\rangle \ldots |0\rangle) &= \tfrac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle) \ldots (|0\rangle + |1\rangle) \\ &= \tfrac{1}{\sqrt{2^n}} \textstyle\sum_{x_1,x_2,\ldots,x_n=0}^{1} |x_1\rangle |x_2\rangle \ldots |x_n\rangle = \tfrac{1}{\sqrt{2^n}} \textstyle\sum_{x \in B_n} |x\rangle . \end{aligned}$$

An important feature of this process (recalling the fundamental significance of poly vs. exponential growth in complexity theory) is that we have created a superposition of *exponentially many* (viz. $2^n$) terms with only a *linear* number of elementary operations viz. application of $H$ just $n$ times.

# 8 The Deutsch-Jozsa (DJ) algorithm

Our first (and historically the first, from 1992) example of an exponential benefit of quantum over classical computation is a quantum algorithm (the so-called DJ algorithm) for the "balanced vs. constant" black-box promise problem, which we re-iterate here:
**The "balanced versus constant" problem**
***Input***: a black box for a Boolean function $f : B_n \rightarrow B$ (one bit output).
***Promise***: $f$ is either (a) a constant function ($f(x) = 0$ for all $x$ or $f(x) = 1$ for all $x$) or (b) a "balanced" function in the sense that $f(x) = 0$ resp. 1 for exactly half of the $2^n$ inputs $x$.
***Problem***: Determine (with certainty) whether $f$ is balanced or constant.
(At the end we will discuss the bounded error version of the problem i.e. requiring the correct solution only with some high probability, 0.999 say).

A little thought (hmm...) shows that classically $2^n/2 + 1$ queries (i.e. exponentially many) are necessary and sufficient to solve the problem with certainty in the worst case. Sufficiency is clear (why?). For necessity, suppose we have a deterministic classical

algorithm that purports to solve this problem with certainty in every case, for any $f$ satisfying the promise, while making $K \leq 2^n/2$ queries. Here the choice of query may even adaptively depend in any way on the results of previous queries too.
A devious adversary (having a function $f$) can force this algorithm to fail as follows (thus showing necessity): when the algorithm is applied to him, he actually hasn't a priori chosen his function $f$ yet but simply answers 0 for all queries. At the end his function has been fixed on $K$ inputs, but if $K \leq 2^n/2$ he is still free to complete the definition of his function to be either constant or balanced, and have it contradict whatever conclusion the algorithm reached.
Similarly for any *probabilistic* classical algorithm whose final output is required still to be correct with certainty (although probabilistic choices may be used along the way), each of the probabilistic branches of the algorithm must work with certainty themselves and the above argument applies to them, showing again that the number of queries (on any probabilistic branch) must be at least $2^n/2 + 1$.

We now show that in the quantum scenario, just *one* query suffices (with $O(n)$ extra processing steps) in every case! Our quantum black box is

$$U_f : |x\rangle\,|y\rangle = |x\rangle\,|y \oplus f(x)\rangle$$

where the input register $|x\rangle$ comprises $n$ qubits and the output register $|y\rangle$ comprises a single qubit. We assume that initially all $(n+1)$ qubits are in standard state $|0\rangle$. We begin by constructing an equal superposition of all $n$-bit strings in the input register (as described above) and (surprisingly!) set the output register to the state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. The latter is achieved by applying $X$ and then $H$ to the output qubit, initially in state $|0\rangle$. Thus we have the $(n+1)$-qubit state

$$\left(\frac{1}{\sqrt{2^n}} \sum_{x \in B_n} |x\rangle\right) |-\rangle\,.$$

Next we run the oracle $U_f$ on this state. To see the effect of this, consider each $x$-term separately. We have (omitting the $\sqrt{2}$ normalisation factors)

$$\begin{aligned} U_f : |x\rangle\,(|0\rangle - |1\rangle) \quad &\longrightarrow \quad |x\rangle\,(|f(x)\rangle - |f(x) \oplus 1\rangle) \\ &= \quad \begin{cases} \;\;|x\rangle\,(|0\rangle - |1\rangle) & \text{if } f(x) = 0 \\ -|x\rangle\,(|0\rangle - |1\rangle) & \text{if } f(x) = 1 \end{cases} \\ &= \quad (-1)^{f(x)}\,|x\rangle\,(|0\rangle - |1\rangle) \end{aligned} \tag{3}$$

i.e. we just get a minus sign on $|x\rangle$ if $f(x) = 1$ and no change if $f(x) = 0$. This process of obtaining the $(-1)^{f(x)}$ sign is sometimes called "phase kickback". Hence on the full superposition we get

$$U_f : \left(\frac{1}{\sqrt{2^n}} \sum_{x \in B_n} |x\rangle\right) |-\rangle \longrightarrow \left(\frac{1}{\sqrt{2^n}} \sum_{x \in B_n} (-1)^{f(x)}\,|x\rangle\right) |-\rangle\,.$$

This is a product state of the $n$-qubit input and single qubit output registers. Discarding the last (output) qubit we get the $n$-qubit state

$$|f\rangle \equiv \frac{1}{\sqrt{2^n}} \sum_{x \in B_n} (-1)^{f(x)} |x\rangle \,. \tag{4}$$

What does this state look like when $f$ is constant or balanced?

***If $f$ is constant*** then $|f\rangle = \pm\frac{1}{\sqrt{2^n}} \sum_x |x\rangle$. If we apply $H_n = H \otimes \ldots \otimes H$ we get $\pm\,|0 \ldots 0\rangle$ because $H$ is self inverse and recall that $H_n\,|0 \ldots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle$.
***If $f$ is balanced*** then the sum in eq. (4) contains an equal number of plus and minus terms, with minus signs sprinkled in some unknown locations along the $2^n$ terms. But if we take the inner product of $|f\rangle$ with $\sum_x |x\rangle$ we simply add up all the coefficients in $|f\rangle$ (as $\langle x|y\rangle = 0$ if $x \neq y$ and $= 1$ if $x = y$) and wherever the minus signs occur, the total sum is always zero i.e. if $f$ is balanced then $|f\rangle$ is orthogonal to $\sum_x |x\rangle$. Hence if we apply the unitary operation $H_n$ (which preserves inner products), $H_n\,|f\rangle$ will be orthogonal to $H_n(\frac{1}{\sqrt{2^n}} \sum_x |x\rangle) = |0 \ldots 0\rangle$. Hence $H_n\,|f\rangle$ must have the form $\sum_{x \neq 0 \ldots 0} a_x\,|x\rangle$ *having the all-zero term absent*. Generally $H_n\,|f\rangle$ will be a superposition of many basis states but for some special balanced functions $f$ it can have the form $|x\rangle \neq |0 \ldots 0\rangle$ comprising a single basis state. See exercise sheet 3 (the Bernstein-Vazirani problem) for more details.

In view of the above discussion, having constructed $|f\rangle$ (for our given black box) we apply $H_n$ and measure the $n$ qubits in the computational basis. If the result is $0 \ldots 0$ then $f$ was certainly constant and if the result is any non-zero string $x_1 \ldots x_n \neq 0 \ldots 0$ then $f$ was certainly balanced. Hence we have solved the problem with one query to $f$ and $(3n + 2)$ further operations: $(n + 1)$ $H$'s and one $X$ to make the input state for $U_f$, $n$ $H$'s on $|f\rangle$ and $n$ single qubit measurements to get the classical output string.

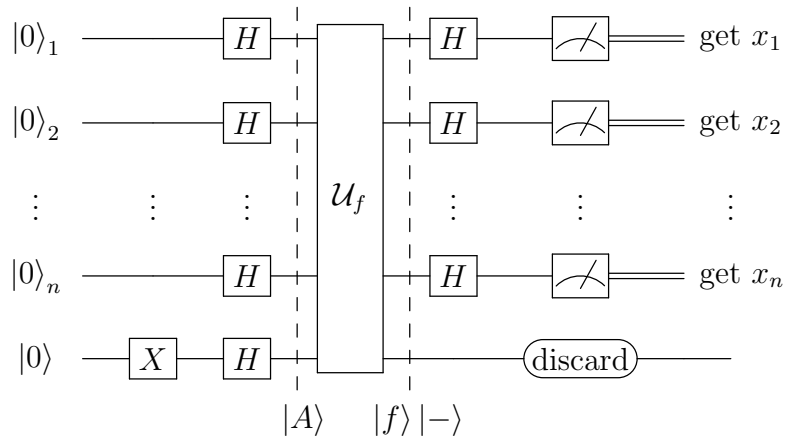



**Figure.** Circuit diagram for the DJ algorithm. The state $|A\rangle$ just before $U_f$ is the equal superposition state $\frac{1}{\sqrt{2^n}} \sum_{x \in B_n} |x\rangle$ in the first $n$ qubits and $|-\rangle$ in the last qubit. The action of $U_f$ then produces the state $|f\rangle$ as in the text above.

**The balanced versus constant problem with bounded error**
Suppose we tolerate some error i.e. require our algorithm to correctly distinguish balanced versus constant functions only with probability $> 1 - \epsilon$ for some $\epsilon > 0$. Then the above (single query) algorithm still works (as it has $\epsilon = 0$) but there is now a classical

(randomised) algorithm that solves the problem with only a constant number of queries (depending on $\epsilon$ as $O(1/\log\epsilon)$ for any $n$ and for any fixed $\epsilon > 0$). Thus we lose the all-interesting exponential gap between classical and quantum query complexities in this bounded error scenario. The classical algorithm is the following: we pick $K$ $x$ values, each chosen independently uniformly at random and evaluate the corresponding $f$ values. If they are all 0 or all 1, output "$f$ is constant". If we get at least one instance of each of 0 and 1, output "$f$ is balanced". Clearly the second output must always be correct (as a constant function can never output both values). But the first output ("$f$ is constant") can be erroneous. Suppose $f$ is a balanced function. Then each random value $f(x)$ has probability half to be 0 or 1. So the probability that $K$ random values are all 0 or all 1 is $2/2^K = 1/2^{K-1}$. This is $< \epsilon$ if $1/2^{K-1} < \epsilon$ i.e. $K > \log 1/\epsilon$ i.e. $K = O(\log 1/\epsilon)$ suffices to guarantee error probability $< \epsilon$ in every case, for all $n$. $\square$

So does the above prove conclusively that quantum computation can be exponentially more powerful (in terms of time complexity) than classical computation? We point out two important shortcomings in this claim.
(i) The first weakness is that if we allow any level of error in the result, however small, we lose the exponential separation between classical and quantum algorithm running times (as described in the previous paragraph). But the *exactly*-zero error scenario in computation is an unrealistic idealisation and for *realistic* computation we should always accept some (suitably small) level of error – physical computers never work perfectly and in the quantum case for example, gates depend on continuous parameters that cannot be physically set with infinite precision. However this weakness can be fully addressed: there exist other black box promise problems for which a provable exponential separation exists between classical and quantum query complexity even in the presence of error. An example is the so-called Simon's quantum algorithm, which we will outline below.
(ii) A second (more serious) issue is the fact that the DJ problem is only a black box problem (with the black box's interior workings being inaccessible to us) rather than a straightforward "standard" computational task with a bit string as input, and no "hidden" ingredients. To convert it to a standard task we would want a class of Boolean functions $f_n : B_n \to B$ such that the balanced/constant decision is hard classically (e.g. takes exponential time in $n$) even if we have full access to a description of the function e.g. a formula for it or a circuit $C_n$ that computes $f_n$. Note that even a constant function can be presented to us in such a perversely complicated way that its trivial action is hard to recognise! Alas, no such (*provably* hard) class of Boolean function descriptions is known.

So, are there any "standard" computational tasks for which we can prove the existence of an exponential speed-up for quantum versus classical computation? No such absolute proofs are known but the difficulty seems to be largely within the classical theory: even though many problems have only exponential-time *known* classical algorithms, they cannot be *proven* to be hard classically i.e. we cannot prove that no poly-time algorithm exists (that we have not yet discovered!) – a glaring instance of this classical theory shortcoming is the notorious fact that it is unproven that the class **NP** (cf later for more about this class) or even **PSPACE**, is strictly larger than **P**. (**PSPACE** intuitively is the class of languages that can be decided with an algorithm that uses a polynomial amount of space or memory, and can thus generally run for exponential time). However

there are problems which are *believed* to be hard for classical computation (i.e. no classical poly-time algorithm, even with bounded error, is known despite much effort) for which poly-time quantum algorithms *do* exist. A centrally important such problem is *integer factorisation*. Below we will describe *Shor's polynomial time quantum algorithm for factorisation* after we introduce the fundamentally important construct of the *quantum Fourier transform*, which is at the heart of the workings of Shor's algorithm.

## 8.1 Simon's algorithm

Consider the following black box promise problem.
**Simon's problem**
***Input***: a black box for a Boolean function $f : B_n \to B_n$ (note: $n$- bit output!).
***Promise***: $f$ is either (a) a one-to-one function or (b) a two-to-one function of the following form – there is an $n$-bit string $\xi \neq 00\ldots0$ such that

$$f(x) = f(y) \quad \text{iff} \quad y = x \oplus \xi \tag{5}$$

(where $\oplus$ is addition of $n$-bits strings as defined on page 54.)
***Problem***: Determine (with bounded error probability) whether $f$ is (a) or (b).

**Remark:** in the case (b) we can further ask for a determination of the string $\xi$. Note that for any string $\xi$ we have $\xi \oplus \xi = 00\ldots0$ so we can say in (b) that $f$ has *period* $\xi$ (relative to addition of $n$-bit strings). $\square$

Simon's quantum algorithm (see exercise sheet 3 for details of how it works!) solves this problem with only $O(n)$ queries to $f$. On the other hand we can argue that the problem is classically *hard*, requiring an exponential number of queries so (in contrast to DJ) we get here a provable exponential separation of quantum vs. classical query complexity for *bounded error* computation.

To appreciate intuitively why the problem is classically hard, suppose that (b) actually holds. Then we'll argue that we need to query $f$ an exponential number of times to have a reasonable probability of noticing that $f$ is not one-to-one. Indeed we obtain no information until we are lucky enough to choose two queries $x$ and $y$ with $f(x) = f(y)$ i.e. $x \oplus y = \xi$. Suppose for example that we choose $2^{n/4}$ queries (independently and uniformly at random). Then the number of pairs of queries is less than $(2^{n/4})^2$ and for each pair the probability that $x \oplus y = \xi$ is $2^{-n}$. Thus the probability of successfully seeing the existence of $\xi$ is less than $2^{-n}(2^{n/4})^2 = 2^{-n/2}$ i.e. even as many as $2^{n/4}$ queries cannot notice the difference between between (a) and (b) with better than an exponentially small probability and hence cannot form the basis of any *bounded* error algorithm. This argument can be made rigorous e.g. allowing arbitrary strategies for choices for queries, but we omit the technicalities here. For more details see D. Simon "On the power of quantum computation", S.I.A.M. Journal on Computing, **28**, p1474-1483 (1997) and J. Gruska "Quantum computing", chapter 3, p109-111. McGraw-Hill Publishing Company (1999).

# 9 The quantum Fourier transform and periodicities

## 9.1 Quantum Fourier transform mod $N$

The quantum Fourier transform (QFT) can be viewed as a generalisation of the Hadamard operation to dimensions $N > 2$. Later we will be especially interested in $N = 2^n$ i.e. the QFT on an $n$-qubit space. As a pure mathematical construction it is the same as the so-called discrete Fourier transform which is widely used in digital signal and image processing. It is a *unitary* matrix that arises naturally in a wide variety of mathematical situations so it fits well into the quantum formalism, providing a bridge between a quantum operation and certain mathematical problems. In fact QFT is at the heart of most known quantum algorithms that provide a significant speedup over classical computation.

Let $\mathcal{H}_N$ denote a state space with an orthonormal basis (the computational basis) $|0\rangle, |1\rangle, \ldots, |N-1\rangle$ labelled by $\mathbb{Z}_N$. The quantum Fourier transform (QFT) modulo $N$, denoted $\mathrm{QFT}_N$ (or just QFT when $N$ is clear) is the unitary transform on $\mathcal{H}_N$ defined by:

$$QFT : |x\rangle \to \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \exp(2\pi i \frac{xy}{N}) \ |y\rangle \tag{6}$$

Thus the $ab^{\text{th}}$ matrix entry is

$$[\mathrm{QFT}]_{ab} = \frac{1}{\sqrt{N}} \exp\left(2\pi i \frac{ab}{N}\right) \qquad a, b = 0, \ldots, N-1$$

(note: here we are labelling rows and columns from 0 to $N-1$ in $\mathbb{Z}_N$ rather than 1 to $N$.) If $\omega = e^{2\pi i/N}$ is the primitive $N^{\text{th}}$ root of unity then the matrix elements are all powers of $\omega$ (divided by $\sqrt{N}$) following a simple pattern:
• The initial row and column always contain only 1's.
• Each row (or column) is a geometric sequence. The $k^{\text{th}}$ row (or column) for $k = 0, \ldots, N-1$ is the sequence of powers of $\omega^k$ (starting with power 0 up to power $N-1$).

**Remark**: Note that $\mathrm{QFT}_4$ is different from $H \otimes H$ and generally $\mathrm{QFT}_{2^n}$ differs from $H_n = H \otimes \ldots \otimes H$. However in group representation theory a Fourier transform can be defined on any group, which embraces both of these constructs as special cases – on a set of 4 elements there are two (non-isomorphic) group structures viz. $\mathbb{Z}_2 \times \mathbb{Z}_2$ and $\mathbb{Z}_4$ (addition of integers mod 4), and then $H \otimes H$ and $\mathrm{QFT}_4$ are respectively the Fourier transforms for these two different group structures. In this course $\mathrm{QFT}_N$ will always mean "Fourier transform on the group $\mathbb{Z}_N$", as defined above in eq. (6). □

Many properties of QFT, including the fact that it is unitary, follow from a basic algebraic fact about roots of unity and geometric series. Recall the formula for the sum of any geometric series

$$1 + \alpha + \alpha^2 + \ldots + \alpha^{N-1} = \begin{cases} \frac{1-\alpha^N}{1-\alpha} & \text{if } \alpha \neq 1 \\ N & \text{if } \alpha = 1 \end{cases}$$

Then setting $\alpha = w^K$ (for some chosen $K$) we have $\alpha = 1$ iff $K$ is a multiple of $N$. Thus

$$1 + \omega^K + \omega^{2K} + \ldots + \omega^{(N-1)K} = \begin{cases} N & \text{if } K \text{ is a multiple of } N. \\ 0 & \text{if } K \text{ is not a multiple of } N. \end{cases} \quad (7)$$

Now to see that QFT is unitary, consider the $ab^{\text{th}}$ element of the matrix product $\text{QFT}^\dagger\text{QFT}$. This is $1/N$ times the sum of "the $a^{\text{th}}$ row of $\text{QFT}^\dagger$ lined up against the $b^{\text{th}}$ column of QFT". The latter sum is just the geometric series with $\alpha = \omega^{b-a}$, divided by $N$. So using eq. (7) we get $0/N = 0$ if $b \neq a$ and we get $N/N = 1$ if $b = a$ i.e. $\text{QFT}^\dagger\text{QFT}$ is the identity matrix and QFT is unitary.

## 9.2 Periodicity determination

A fundamental application of the Fourier transform (both classically and quantumly) is the determination of periodicity exhibited in a function or some other given data. Some important mathematical problems (such as integer factorisation, as we'll see later) can be reduced to problems of periodicity determination.

Suppose we are given (a black box for) a function $f : \mathbb{Z}_N \to Y$ (where typically $Y = \mathbb{Z}_M$ for some $M$) and it is promised that $f$ is periodic with some period $r$ i.e. there is a smallest number $r$ such that $f(x + r) = f(x)$ for all $X \in \mathbb{Z}_N$ (and $+$ is addition mod $N$). We will also assume that $f$ is one-to-one in each period i.e. $f(x_1) \neq f(x_2)$ for all $0 \leq x_1 < x_2 < r$. We want a method of determining $r$ with some constant level of probability (0.99 say) that's independent of increasing the size of $N$. It can be shown that $O(\sqrt{N})$ queries to $f$ (i.e. a number not bounded by any polynomial in $\log N$)) are necessary and sufficient to achieve this in *classical* computation with a black box for $f$. In some cases further information may be available about $f$ e.g. we may have an explicit formula for it but the periodicity determination may *still* be hard (we will see an example later), requiring a number of steps that is not bounded by any polynomial in $\log N$. In the quantum scenario we will see that $r$ can always be determined with any constant high level of probability $1 - \epsilon$ using only $O(\log \log N)$ queries and poly$(\log N)$ further processing steps i.e. exponentially faster than any classical method.

**Quantum algorithm for periodicity determination**

We begin by constructing a uniform superposition $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ and one query to $U_f$ to obtain the state $|f\rangle = \frac{1}{\sqrt{N}} \sum_{\text{all}x} |x\rangle |f(x)\rangle$. Since $f$ is periodic (with unknown period $r$) $r$ must divide $N$ exactly and we set $A = N/r$, which is the number of periods. If we measure the second register we will see some value $y = f(x_0)$ where $x_0$ is the least $x$ having $f(x) = y$. Then the first register will be projected into an equal superposition of the $A$ values of $x = x_0, x_0 + r, x_0 + 2r, \ldots, x_0 + (A-1)r$ for which $f(x) = y$ i.e. we get

$$|per\rangle = \frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |x_0 + jr\rangle$$

Here $0 \leq x_0 \leq r - 1$ has been chosen uniformly at random (by the extended Born rule, since each possible value $y$ of $f$ occurs the same number $A$ of times i.e. once in each

period.) If we measure the register of $|per\rangle$ we will see $x_0 + j_0 r$ where $j_0$ has been picked uniformly at random too. Thus we have a random period (the $j_0^{\text{th}}$ period) and a random element in it (determined by $x_0$) i.e. overall we get a random number between 0 and $N-1$, giving no information about $r$ at all. Nevertheless the state $|per\rangle$ seems to contain the information of $r$!

The resolution of this problem is to use the Fourier transform which is known even in *classical* image processing, to be able to pick up periodicities in a periodic pattern irrespective of an overall random shift of the pattern (e.g. the $x_0$ in $|per\rangle$). Applying QFT to $|per\rangle$ we get (using eq. (6) with $x$ replaced by $x_0 + jr$, and summing over $j$):

$$QFT\,|per\rangle = \frac{1}{\sqrt{NA}} \sum_{j=0}^{A-1} \left( \sum_{y=0}^{N-1} \omega^{(x_0+jr)y}\,|y\rangle \right) = \frac{1}{\sqrt{NA}} \sum_{y=0}^{N-1} \omega^{x_0 y} \left[ \sum_{j=0}^{A-1} \omega^{jry} \right] |y\rangle\,. \quad (8)$$

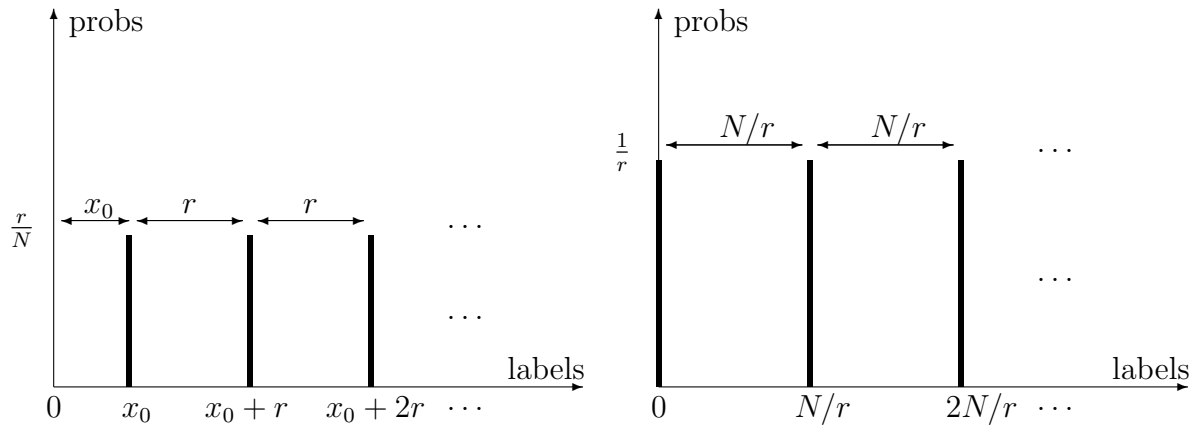
(In the last equality we have reversed the order of summation and factored out the $j$-independent $\omega^{x_o y}$ terms). Which labels $y$ appear here with nonzero amplitude? Look at the square-bracketed coefficient of $|y\rangle$ in eq. (8). It is a geometric series with powers of $\alpha = e^{2\pi i r y/N} = (e^{2\pi i/A})^y$ summed from power 0 to power $A-1$. According to eq. (7) (now applied with $A$ taking the role of $N$ there) this sum is *zero* whenever $y$ is not a multiple of $A$ and the sum is $A$ otherwise i.e. only multiples of $A = N/r$ survive as $y$ values:

$$\sum_{j=0}^{A-1} \omega^{jry} = \begin{cases} A & \text{if } y = kN/r \text{ for } k = 0, \ldots, r-1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\text{QFT}\,|per\rangle = \sqrt{\frac{A}{N}} \sum_{k=0}^{r-1} \omega^{x_0 (kN/r)}\,|kN/r\rangle\,.$$

The random shift $x_0$ has been eliminated from the labels and now occurs only in a pure phase $\omega^{x_0 kN/r}$ (whose modulus squared is 1), and the periodicity of the ket labels has been "inverted" from $r$ to $A = N/r$. Since measurement probabilities are squared moduli of the amplitudes, these probabilities are now independent of $x_0$ and depend only on $N$ (known) and $r$ (to be determined). This is represented schematically in the following diagram.




(a) for $|per\rangle$ (b) for QFT$|per\rangle$

If we now measure the label we will obtain a value $c$ which is a multiple $k_0 N/r$ of $N/r$ where $0 \leq k_0 \leq r-1$ has been chosen uniformly at random. Thus $c = k_0 N/r$ so

$$\frac{k_0}{r} = \frac{c}{N}.$$

Here $c$ and $N$ are known and $k_0$ is unknown and random, so how do we get $r$ out of this? If (by some good fortune!) $k_0$ was coprime to $r$ we could cancel $c/N$ down and read off $r$ as the denominator. If $k_0$ is not coprime to $r$ then this procedure will deliver a denominator $r'$ that is smaller than the correct $r$ so $f(x) \neq f(x + r')$ for any $x$. Thus in our process we check the output $r$ value by evaluating $f(0)$ and $f(r)$ and accepting $r$ as the correct period iff these are equal.

But $k_0$ was chosen at random so what is the chance of getting this good fortune of coprimality? We'll use (without proof) the following theorem from number theory:

**Theorem 1** *(Coprimality theorem) The number of integers less than $r$ that are coprime to $r$ grows as $O(r/\log\log r)$ with increasing $r$. Hence if $k_0 < r$ is chosen at random*

$$prob(k_0 \text{ coprime to } r) \approx O((r/\log\log r)/r) = O(1/\log\log r). \qquad \square$$

Thus if we repeat the whole process $O(\log\log r) < O(\log\log N)$ times we will obtain a coprime $k_0$ in at least one case with a constant level of probability. Here we have used the following fact from probability theory:

**Lemma 1** *If a single trial has success probability $p$ and we repeat the trial $M$ times independently then for any constant $0 < 1 - \epsilon < 1$:*

$$prob(\text{at least one success in } M \text{ trials}) > 1 - \epsilon \text{ if } M = \frac{-\log \epsilon}{p}$$

*so to achieve any constant level $1 - \epsilon$ of success probability, $O(1/p)$ trials suffice.*

**Proof of lemma** We have that the probability of at least one success in $M$ runs $= 1-$ prob(all runs fail) $= 1-(1-p)^M$. Then $1-(1-p)^M = 1-\epsilon$ if $M = \frac{-\log\epsilon}{-\log(1-p)}$. Next use the fact that $p < -\log(1-p)$ for all $0 < p < 1$ to see that $M < \frac{-\log\epsilon}{p}$ i.e. $M = O(1/p)$ repetitions suffice. $\square$

In each round we query $f$ three times (once at the start to make $|f\rangle$ and twice more at the end to check the output $r$) so we use $O(\log\log N)$ queries in all. We also need to apply the "large" unitary gate $\mathrm{QFT}_N$ (which grows with $N$) and we show in the next section that this may be implemented in $O((\log N)^2)$ elementary steps. The remaining operations are all familiar arithmetic operations on integers of size $O(N)$ (such as cancelling $c/N$ down to lowest form) that are all well known to be computable in polynomial time i.e. poly$(\log N)$ steps. Thus we succeed in determining the period with any constant level $1-\epsilon$ of probability with $O(\log\log N)$ queries and $O(\mathrm{poly}(\log N))$ further computational steps.

We have described above the quantum algorithm for periodicity determination, for periodic functions on $\mathbb{Z}_n$ which will form the core of Shor's efficient quantum algorithm for integer factorisation (cf below). But the basic problem of periodicity determination may be mathematically generalised in a natural way from $\mathbb{Z}_n$ to an arbitrary group $G$ as the so-called *hidden subgroup problem* (beyond the scope of this course). This formalism leads to a class of further important quantum algorithms of which Simon's algorithm and the above $\mathbb{Z}_n$ case are special cases.

## 9.3 Efficient implementation of QFT

*This subsection is not required for exam purposes.*

If $N = 2^n$ is an integer power of 2 then QFT mod $N$ acts on $n$ qubits. For these dimension sizes we will show how to implement QFT with a circuit of polynomial size $O(n^2)$. This is a very special property of QFT – almost all unitary transforms in dimension $2^n$ require exponential sized ( $O(\mathrm{poly}(2^n))$ sized) circuits for their implementation. For general $N$ (not a power of 2) we do not have an exact efficient (i.e. poly$(\log N)$ sized) implementation. Instead we generally approximate QFT mod $N$ by QFT mod $2^k$ where $2^k$ is near enough to $N$ to incur only an acceptably small reduction in the success probability of the algorithm.

Our efficient implementation of QFT is really just a translation of the classical fast Fourier transform formalism to the quantum scenario. We begin by showing that the $n$ qubit state

$$QFT\,|x\rangle = \frac{1}{\sqrt{2^n}} \sum_y \exp 2\pi i \frac{xy}{2^n}\ |y\rangle$$

is actually a *product* state of $n$ one-qubit states. We write $0 \le x, y \le 2^{n-1}$ in binary (as $n$ bit strings of digits):
(Warning: Take care to distinguish arithmetic mod $2^n$ in $\mathbb{Z}_{2^n}$ used here from the bitwise

arithmetic of $n$ bit strings that we used earlier!)

$$x = x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \ldots + x_1 2 + x_0$$

$$y = y_{n-1}2^{n-1} + y_{n-2}2^{n-2} + \ldots + y_1 2 + y_0$$

In $xy/2^n$ we discard any terms that are whole numbers since these make no contribution to $\exp 2\pi i xy/2^n$ and a direct calculation gives:

$$\frac{xy}{2^n} \equiv y_{n-1}(.x_0) + y_{n-2}(.x_1 x_0) + \ldots + y_0(.x_{n-1}x_{n-2}\ldots x_0) \tag{9}$$

where the factors in parentheses are binary expansions e.g.

$$.x_2x_1x_0 = \frac{x_2}{2} + \frac{x_1}{2^2} + \frac{x_0}{2^3}$$

Now

$$\sum_y \exp 2\pi i \frac{xy}{2^n} \; |y\rangle = \sum_{y_0,\ldots,y_{n-1}} \exp 2\pi i \frac{xy}{2^n} \; |y_{n-1}\rangle \, |y_{n-2}\rangle \ldots |y_0\rangle$$

and we want to insert the expression for $xy/2^n$ from eq. (9) into the exponential. Since eq. (9) is a *sum* over the different $y_i$'s, the exponential will be a *product* of these terms and hence the sum $\sum_{y_0,\ldots,y_{n-1}}$ splits up into a product of single index sums $(\sum_{y_0})(\sum_{y_1})\ldots(\sum_{y_{n-1}})$ so we get

$$\sum_y \exp 2\pi i \frac{xy}{2^n} \; |y\rangle = \sum_y \exp 2\pi i \frac{xy}{2^n} \; |y_{n-1}\rangle \, |y_{n-2}\rangle \ldots |y_0\rangle =$$

$$\left(|0\rangle + e^{2\pi i(.x_0)} \, |1\rangle\right) \left(|0\rangle + e^{2\pi i(.x_1x_0)} \, |1\rangle\right) \ldots \left(|0\rangle + e^{2\pi i(.x_{n-1}\ldots x_0)} \, |1\rangle\right). \tag{10}$$

Hence QFT$|x\rangle$ is the product of corresponding 1-qubit states obtained by taking each bracket with a $1/\sqrt{2}$ normalising factor.

This factorisation is the key to building our QFT circuit. It should map each basis (product) state $|x_{n-1}\rangle \ldots |x_0\rangle$ into the corresponding product state given in eq. (10). Before we start note that the Hadamard operation can be expressed in our binary fractional notation as

$$H \, |x\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i(.x)} \, |1\rangle\right).$$

Indeed if $x = 0$ resp. 1 then $.x$ is 0 resp. 1/2 as a decimal fraction so $e^{2\pi i(.x)}$ is 1 resp. -1, as required.

To see how the QFT circuit actually works, let's look at the example of $N = 8$ i.e. $n = 3$. We want a circuit that transforms $|x_2\rangle \, |x_1\rangle \, |x_0\rangle$ to the following states in these three registers (called $y_2, y_1, y_0$ at the output):

| $y_2$ register | | $y_1$ register | | $y_0$ register |
|---|---|---|---|---|
| $\underbrace{\frac{1}{\sqrt{2}}\left(\|0\rangle + e^{2\pi i(.x_0)}\|1\rangle\right)}$ | $\otimes$ | $\underbrace{\frac{1}{\sqrt{2}}\left(\|0\rangle + e^{2\pi i(.x_1x_0)}\|1\rangle\right)}$ | $\otimes$ | $\underbrace{\frac{1}{\sqrt{2}}\left(\|0\rangle + e^{2\pi i(.x_2x_1x_0)}\|1\rangle\right)}$ |
| STAGE 3 | | STAGE 2 | | STAGE 1 |
| $H\|x_0\rangle$. This operation depends only on $x_0$ (not $x_1, x_2$). Do it last (third) and put result on $x_0$ line. | | $H\|x_1\rangle$ followed by phase shift $e^{2\pi i 0.0x_0}$ i.e. phase shift of $e^{2\pi i 0.01}$ controlled by $x_0$ value. These operations depend on $x_1, x_0$ (not $x_2$). Do them second and accumulate result on $x_1$ line (as $x_1$ line no longer needed after this). | | $H\|x_2\rangle$ followed by phase shifts of $e^{2\pi i 0.01}$ and $e^{2\pi i 0.001}$ controlled by $x_1$ and $x_0$ respectively. These operations depend on $x_0, x_1, x_2$. Do them first and accumulate result on $x_2$ line (as $x_2$ line no longer needed after this). |

After completion of these three stages, the desired final contents of the $y_0, y_1, y_2$ lines are respectively on the $x_2, x_1, x_0$ lines. Thus finally just reverse the order of the qubits in the string (e.g. by swap operations).

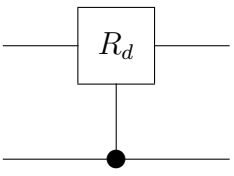
To draw an actual circuit diagram we consider the three stages in turn. In addition to the Hadamard gate $H$ we'll introduce the 1-qubit phase gate:

$$R_d = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2^d} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i(0.00\ldots01)} \end{pmatrix} \tag{11}$$

where the binary digit 1 in the last exponential is $(d+1)$ places to the right of the dot. The controlled-$R_d$ gate, denoted C-$R_d$ acts on two qubits and is defined by the following actions

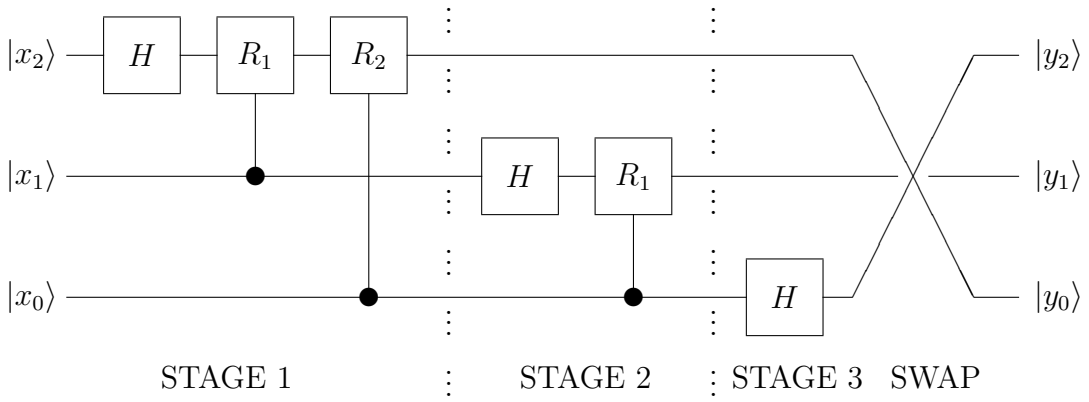
$$\text{C-}R_d\,|0\rangle\,|\psi\rangle = |0\rangle\,|\psi\rangle \qquad \text{C-}R_d\,|1\rangle\,|\psi\rangle = |1\rangle\,R_d\,|\psi\rangle$$

for any 1-qubit state $|\psi\rangle$. Diagramatically this will be denoted as




with a "blob" on the control qubit line.

In terms of all these, the circuit for $\text{QFT}_8$ is




For $N = 8 = 2^3$ we use 3 Hadamard gates (one in each stage) and $2 + 1$ controlled phase gates (in stages 1 and 2 respectively). For general $N = 2^n$ we would use $n$ Hadamard gates (one in each of $n$ stages) and $(n-1) + (n-2) + \ldots + 2 + 1 = n(n-1)/2$ controlled phase gates (in stages $1, 2, \ldots, n-1$ respectively). Overall we have $O(n^2) = O((\log N)^2)$ gates for QFT mod $N$. (In this accounting we have ignored the final swap operation to reverse the order of qubits, but this requires only a further $O(n)$ 2-qubit SWAP gates to implement).

# 10 Quantum algorithms for search problems

Searching is a fundamentally important task; many important computational problems can be thought of as searching tasks. For example factoring $N$ can be viewed as a search amongst integers less than $N$ for one that divides $N$ exactly. Before discussing Grover's quantum searching algorithm we introduce the complexity class **NP** which contains many problems of urgent practical interest, and is fundamentally related to the notion of searching.

## 10.1 The class NP and search problems

We can intuitively think of **NP** as comprising problems that are "hard to solve" (i.e. no poly time algorithm known) but if a solution (or certificate of a solution) is given then its correctness can be "easily verified" (i.e. in poly time). Typically we are faced with a search over an exponentially large space of candidates seeking a "good" candidate, and given any candidate it is easy to check if it is good or not.

**Definition of NP**
**NP** ("**n**ondeterministic **p**oly time"):
a language is in **NP** if it has a *poly-time verifier* $V$. A verifier $V$ for a language $L$ is a computation with two inputs $w$ and $c$ such that:
(i) if $w \in L$ then *for some* $c$, $V(w, c)$ halts with "accept". Any such 'good' $c$ is called a certificate of membership for $w$;
(ii) if $w \notin L$ then *for all* $c$, $V(w, c)$ halts with "reject".
$V$ is a poly-time verifier if for all inputs $(w, c)$ $V$ runs in poly$(n)$ time where $n$ is the size of $w$. (Note that in this case $c$ need only be poly$(n)$ long too, since any single step of a computation can access only a constant number of new bits).

Intuitively (i) and (ii) say that you can certify membership of $L$ (viz. (i)) in such a way that you cannot be tricked into accepting false $w$'s (viz. (ii)) and checking of certificates can be done quickly/efficiently. Note the asymmetry – we are required to certify only membership, but not non-membership.

**Alternative definition of NP**
Imagine a computer that operates "nondeterministically" i.e. instead of sequentially implementing the steps of a single algorithm, at each step the computer duplicates itself and branches into two computational paths performing two steps (possibly the same) that are performed simultaneously in parallel (in contrast to a probabilistic choice of one or other step). Thus after $m$ steps we have $2^m$ computers performing computations in parallel. We require that all paths eventually halt (with "accept" or "reject") and the running time of this nondeterministic computation is defined to be the length of the longest path.

The computation is defined to:
(i) accept its input if *at least one* path accepts; and
(ii) reject its input if *all* paths reject,

(so all inputs are either accepted or rejected as (ii) is the negation of (i)). Then we have: **Proposition: NP** is the class of languages that are decided by a nondeterministic computation with polynomial running time.

[Optional exercise: prove the proposition – given such a nondeterministic computation and input $w$, what is the verifier and certificate (if $w \in L$)? Conversely, given a verifier, what is the corresponding nondeterministic computation with acceptance conditions as above?]

Note that this notion of computation is "non-physical" for complexity considerations, in the following sense: although we have just a polynomial running time, we generally need to invest an *exponential* amount of physical resources to actually implement it viz. an exponential number of computers all running simultaneously, or alternatively, a single computer with exponential running time - being used to do an exponential number of computations i.e. all the paths, in succession.

**The satisfiability problem SAT**: given a Boolean formula $\phi(x_1, \ldots, x_n)$ with $n$ variables and single bit output, we want to decide if there is an assignment $x_1 = b_1, \ldots, x_n = b_n$ with $\phi(b_1, \ldots, b_n) = 1$. Any such assignment is called a satisfying assignment for $\phi$. A brute force evaluation of all $2^n$ possible assignments will surely decide this problem but this generally takes exponential ($O(2^n)$) time. More formally, if we encode the formula as a bit string using some specified representation of its basic symbols, each as a bit string, then inputs of size $m$ could have $O(m)$ variables and hence the brute force algorithm runs in exponential time.

It is not known whether SAT is in **P** or not but it is easily seen to be in **NP** – if $\phi$ is satisfiable then the certificate $c$ is any actual satisfying assignment and the verifier $V(\phi, c)$ simply evaluates $\phi(c)$ to check that it is 1. Clearly if $\phi$ is unsatisfiable we cannot be tricked into accepting it by this procedure!

**Relation to searching:** SAT illustrates a fundamental connection between **NP** and search problems – for any $\phi(x_1, \ldots, x_n)$ we have an exponentially large number of candidate assignments (possible certificates) and we want to know if a "good" (satisfying) assignment exists. Although it is not clear how to locate a good candidate "quickly", if we are given any prospective candidate we can check quickly if it is good or not. This is a general feature of very many practical problems e.g. scheduling/timetabling tasks, or more general simultaneous constraint satisfaction problems.

From the definitions we have the series of inclusions $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE}$ and $\mathbf{P} \subseteq \mathbf{BPP} \subseteq \mathbf{PSPACE}$, but it is not known whether either of **NP** and **BPP** is contained in the other or not. The most notoriously famous open problem of complexity theory is the question of whether **P** is equal to **NP** or not.

## The unstructured search problem

Suppose we are given a large database with $N$ items and we wish to locate a particular item. We assume that the database is entirely unstructured or unsorted but given any item we can easily check whether or not it is the one we seek. Our algorithm should locate the item with some constant level of probability (half say) independent of the

size $N$. Each access to the database is called a query and we normally regard it as one computational step.

For classical computation we may argue that $O(N)$ queries will be necessary and sufficient: the good item has completely unknown location; if we examine an item and find it bad, we gain no further information about the location of the good item (beyond the fact that it is not the current one). Hence if we examine $m$ items the probability of seeing the good one is $p = m/N$ so we must have $m = O(N)$ to have $p$ constant.

For quantum computation we will see that $O(\sqrt{N})$ queries are sufficient (and in fact that number is necessary too) to locate the good item i.e. we get a *quadratic* speedup over classical search. This speedup does not cross the polynomial vs. exponential divide (the "holy grail" of complexity theory) but it is still viewed as significant in situations where exhaustive search is the best known classical algorithm. At first sight we might have naively expected an exponential quantum speedup here: suppose $N = 2^n$ and recall that a quantum algorithm can easily access $2^n$ items in superposition (by use of only $n = \log N$ Hadamard operations) so we can look up the "goodness" of *all* items in superposition, with just *one* query! We may then hope that we could manipulate the resulting quantum state to efficiently reveal the good item. But the above-quoted result shows that this hope cannot be realised. Intuitively the good item occurs with only an exponentially small amplitude in the total superposition. If the item were re-located at another place (or reclassified as bad) then the corresponding quantum state would differ only by an exponentially small amount in the space of quantum states and it will thus be very difficult to reliably distinguish by any physical process.

Databases are often actually structured, in a way that can facilitate the search. As an example suppose our $N$ items are labelled by the numbers and we seek a particular one labelled $k$. Unstructured search (requiring $O(N)$ queries) corresponds to the database containing the numbers in some unknown random order. But if the items are structured by being presented in numerical order, then we can locate $k$ with only $O(\log N)$ queries (in fact exactly $1 \times \log N$ queries) using a binary search procedure: each query of a middle item eliminates an entire half of the remaining database. This kind of structured search is common in practice e.g. the lexicographic ordering of names in a large phone book facilitating search for a given person's number. But suppose we were given a person's number and asked to determine their name. Then we would be faced with an essentially unstructured search requiring a lot more time!

In the following we will consider quantum algorithms for only *unstructured* search, in particular Grover's quantum searching algorithm which achieves this search in $O(\sqrt{N})$ queries. The issue of understanding which kinds of structure in a database can provide a good benefit for quantum versus classical computation is still largely open and a topic of current research. (One interesting known result is that in the case of a linearly ordered database (such as the phone book above) any quantum algorithm still requires $O(\log N)$ queries but the actual number of queries now is $k \log N$ with $k$ strictly less than 1).

## 10.2 Grover's Quantum Searching Algorithm

**Reflections and projections in Dirac ket notation**
We first recall some elementary constructions from linear algebra (some of which we've already seen near the start of the course) that will be used in the discussion of Grover's algorithm. If $|\alpha\rangle$ is any unit length ket vector then

$$\Pi_{|\alpha\rangle} = |\alpha\rangle \langle\alpha|$$

is the operation of *projection onto* $|\alpha\rangle$, and

$$I_{|\alpha\rangle} = I - 2\,|\alpha\rangle \langle\alpha|$$

(with $I$ denoting the identity operation) is the operation of *reflection in the subspace that is orthogonal to* $|\alpha\rangle$ (i.e. vectors in that subspace are left unchanged and general vectors have their component along $|\alpha\rangle$ reversed in sign). For any unitary operator $U$ it is easy to check that

$$U\Pi_{|\alpha\rangle}U^\dagger = \Pi_{U|\alpha\rangle} \qquad UI_{|\alpha\rangle}U^\dagger = I_{U|\alpha\rangle}. \tag{12}$$

**Example.**
In the space of a single qubit let $\left|\alpha^\perp\right\rangle$ be any chosen unit vector orthogonal to $|\alpha\rangle$. Then any ket vector may be uniquely expressed as $|v\rangle = x\,|\alpha\rangle + y\left|\alpha^\perp\right\rangle$ and

$$\Pi_{|\alpha\rangle}\,|v\rangle = x\,|\alpha\rangle \qquad I_{|\alpha\rangle}\,|v\rangle = -x\,|\alpha\rangle + y\left|\alpha^\perp\right\rangle$$

so $I_{|\alpha\rangle}$ is reflection in the line defined by $\left|\alpha^\perp\right\rangle$. □

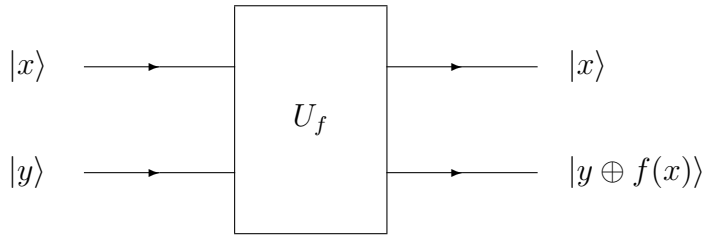
### Grover's algorithm for unstructured search

We consider the fundamental problem of unstructured search for a unique item, and describe a quantum algorithm originally due to Lov Grover in 1996 which solves the problem with only $O(\sqrt{N})$ queries. We will give a simple geometrical derivation of the algorithm (different from Grover's original algebraic approach, cf exercise sheet 4) which clarifies its workings.

It will be convenient to take the size $N$ of our search space to be a power of 2 *viz.* $N = 2^n$. Thus we can label the entries by bit strings (i.e. strings of 0's and 1's) of length $n$. Our search problem may then be phrased in terms of a black box promise problem as follows. We will replace the database by a black box which computes an $n$ bit function $f : B_n \to B$. It is promised that $f(x) = 0$ for all $n$ bit strings except exactly one string, denoted $x_0$ (the "marked" position that we seek) for which $f(x_0) = 1$. Our problem is to determine $x_0$. As usual we assume that $f$ is given as a unitary transformation $U_f$ on $n + 1$ qubits defined by

$$U_f\,|x\rangle\,|y\rangle = |x\rangle\,|y \oplus f(x)\rangle \tag{13}$$

Here the input register $|x\rangle$ consists of $n$ qubits and the output register $|y\rangle$ consists of a single qubit. The symbol $\oplus$ denotes addition modulo 2. Pictorially we have



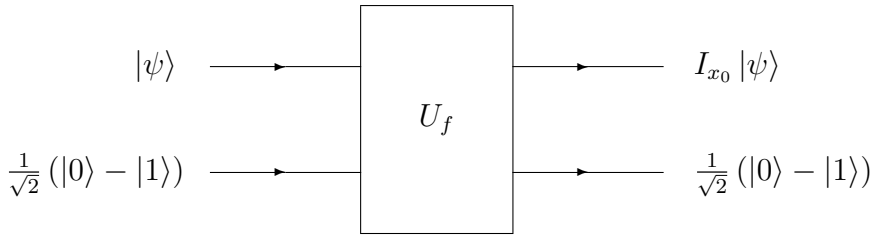

The assumption that the database is *unstructured* is formalised here as the standard oracle idealisation that we have no access to the internal workings of $U_f$ – it operates as a "black box" on the input and output registers, telling us only if the queried item is good or not.

Instead of using $U_f$ we will generally use a closely related operation denoted $I_{x_0}$ on $n$ qubits. It is defined by

$$I_{x_0}|x\rangle = \begin{cases} |x\rangle & \text{if } x \neq x_0 \\ -|x_0\rangle & \text{if } x = x_0 \end{cases} \tag{14}$$

i.e. $I_{x_0}$ simply inverts the amplitude of the $|x_0\rangle$ component and so $I_{x_0}$ is just the reflection operator $I_{|x_0\rangle}$ defined above. If $x_0$ is the $n$ bit string $00\ldots0$ then $I_{x_0}$ will be written simply as $I_0$.

A black box which performs $I_{x_0}$ may be simply constructed from $U_f$ by just setting the output register to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Then the action of $U_f$ leaves the output register in this state and effects $I_{x_0}$ on the input register. Pictorially




Our searching problem becomes the following: we are given a black box which computes $I_{x_0}$ for some $n$ bit string $x_0$ and we want to determine the value of $x_0$ using the least number of queries to the box.

We will work in a space of $n$ qubits with a standard basis $\{|x\rangle\}$ labelled by $n$-bit strings $x$. Let $\mathcal{B}_n$ denote the space of all $n$-qubit states. Let $H_n = H \otimes \ldots \otimes H$ acting on $\mathcal{B}_n$ denote the application of $H$ to each of the $n$ qubits separately.

Grover's quantum searching algorithm operates as follows. Having no initial information about $x_0$ we begin with the state

$$|\psi_0\rangle = H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle \tag{15}$$

which is an equal superposition of all possible $x_0$ values. Consider the compound operator

$Q$, called the **Grover iteration operator**, defined by

$$Q = -H_n I_0 H_n I_{x_0}. \tag{16}$$

Note that all amplitudes in $|\psi_0\rangle$ and all matrix elements of $Q$ are *real* numbers so to analyse $Q$ we will be able to use the geometrical interpretations of the projection and reflection operators described above in terms of *real* (rather than complex) Euclidean geometry.

In the next section we will explain the structure of $Q$ and show that it has a simple geometrical interpretation:

**(Q1):** In the plane $\mathcal{P}(x_0)$ spanned by (the initially unknown) $|x_0\rangle$ and $|\psi_0\rangle$, $Q$ is rotation through angle $2\alpha$ where $\sin\alpha = \frac{1}{\sqrt{N}}$.

**(Q2):** In the subspace orthogonal to $\mathcal{P}(x_0)$, $Q = -I$ where $I$ is the identity operation.

Thus by repeatedly applying $Q$ to the starting state $|\psi_0\rangle$ in $\mathcal{P}(x_0)$ we may rotate it around near to $|x_0\rangle$ and then determine $x_0$ with high probability by a measurement in the standard basis. For large $N$, $|x_0\rangle$ and $|\psi_0\rangle$ are almost orthogonal and $2\alpha \approx 2\sin\alpha = \frac{2}{\sqrt{N}}$. Thus about $\frac{\pi}{4}\sqrt{N}$ iterations will be needed. Each application of $Q$ uses one evaluation of $I_{x_0}$ and hence of $U_f$ so $O(\sqrt{N})$ evaluations are required, representing a square root speedup over the $O(N)$ evaluations needed for a classical unstructured search. More precisely we have $\langle x_0|\psi_0\rangle = \frac{1}{\sqrt{N}}$ so the number of iterations needed is the integer nearest to $(\arccos\frac{1}{\sqrt{N}})/(2\arcsin\frac{1}{\sqrt{N}})$ (which is independent of $x_0$).

**Example: searching for "one in four".**
A simple striking example is the case of $N = 4$ in which $\sin\alpha = \frac{1}{2}$ and $Q$ is a rotation through $\pi/3$. The initial state is $|\psi_0\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$ and for any marked $x_0$ the angle between $|x_0\rangle$ and $|\psi_0\rangle$ is precisely $\pi/3$ too. Hence after one application of $Q$ i.e. just *one* query, we will learn the position of any single marked item in a set of four with *certainty*! □

**The iteration operator $Q$ – reflections and rotations**

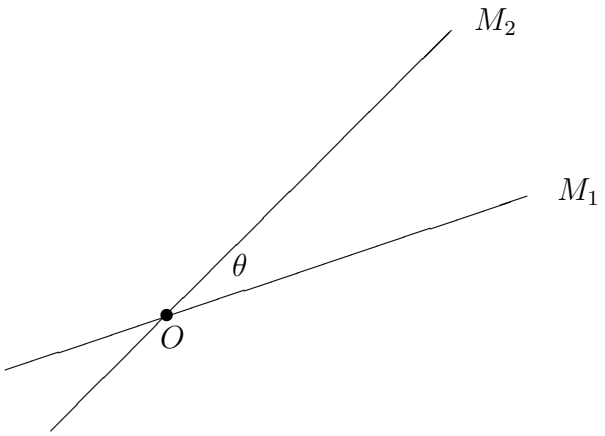
Using eq. (12) (and noting that $H = H^\dagger$) the Grover iteration operator can be written

$$Q = -I_{H_n|0\ldots0\rangle}\, I_{|x_0\rangle} = -I_{|\psi_0\rangle}\, I_{|x_0\rangle}.$$

Now for any $|\alpha\rangle$ and $|v\rangle$ we have $I_{|\alpha\rangle}\,|v\rangle = |v\rangle - 2\langle\alpha|v\rangle\,|\alpha\rangle$ i.e. $|v\rangle$ is modified by a multiple of $|\alpha\rangle$. Hence if $|v\rangle$ is in the (real) plane $\mathcal{P}(x_0)$ spanned by $|x_0\rangle$ and $|\psi_0\rangle = H_n\,|0\ldots0\rangle$ then both $I_{|x_0\rangle}\,|v\rangle$ and $I_{|\psi_0\rangle}\,|v\rangle$ will be in $\mathcal{P}(x_0)$ too – indeed $I_{|x_0\rangle}$ and $I_{|\psi_0\rangle}$ within this plane are just reflections in the lines perpendicular to $|x_0\rangle$ and $|\psi_0\rangle$ respectively. Hence $Q$ also preserves the plane $\mathcal{P}(x_0)$ and its action is given by the following fact of Euclidean geometry.

**Lemma**: let $M_1$ and $M_2$ be two mirror lines in the Euclidean plane $\mathbb{R}^2$ intersecting at a point $O$ and let $\theta$ be the angle in the plane from $M_1$ to $M_2$ (cf figure below). Then the

operation of reflection in $M_1$ followed by reflection in $M_2$ is just (anticlockwise) rotation by angle $2\theta$ about the point $O$.




**Proof of lemma**: this is immediate, for example, from standard matrix expressions for rotations and reflections in $\mathbb{R}^2$. $\square$

Using the lemma we see that the action of $I_{H_n|0\rangle} I_{|x_0\rangle} = -Q$ in $\mathcal{P}(x_0)$ is a rotation through $2\beta$ where $\cos\beta = \langle x_0| H_n |0\rangle = \frac{1}{\sqrt{N}}$. For large $N$, $\beta \approx \pi/2$ and we have a rotation of almost $\pi$. It would be possible to use this large rotation as the basis of the quantum searching algorithm but we prefer a smaller incremental motion. We could use the operator $(I_{H_n|0\rangle} I_{|x_0\rangle})^2$ but there is another solution, explaining the occurrence of the minus sign in the definition of $Q$:

**Lemma**: for any 2 dimensional real $v$ we have

$$-I_v = I_{v^\perp}$$

where $v^\perp$ is a unit vector perpendicular to $v$.

**Proof:** For any vector $u$ we write $u = av + bv^\perp$. Then $I_v$ just reverses the sign of $a$ and $-I_v$ reverses the sign of $b$. Thus the action of $-I_v$ is the same as that of $I_{v^\perp}$. $\square$

Hence $Q = -I_{H_n|0\rangle} I_{|x_0\rangle}$ acting in $\mathcal{P}(x_0)$ is a rotation through $2\alpha$ where $\alpha$ is the angle between $|x_0\rangle$ and a perpendicular state to $H_n |0\rangle$ i.e. $\sin\alpha = \langle x_0| H_n |0\rangle = \frac{1}{\sqrt{N}}$ as claimed in **(Q1)**.

To see the effect of $Q$ on states orthogonal to $\mathcal{P}(x_0)$ suppose that $|\xi\rangle \in \mathcal{B}_n$ is orthogonal to both $H_n |0\rangle$ and $|x_0\rangle$. Then from the definitions of $I_{|x_0\rangle}$ and $I_{H_n|0\ldots0\rangle}$ we see that $I_{|x_0\rangle} |\xi\rangle = I_{H_n|0\rangle} |\xi\rangle = |\xi\rangle$ so $Q = -I$ in the orthogonal complement to $\mathcal{P}(x_0)$, as claimed in **(Q2)**.

Thus even though $x_0$ is unknown (but we are given a black box for $I_{x_0}$) we can construct a rotation operator $Q$ in the plane spanned by the fixed starting state $|\psi_0\rangle$ and the unknown $|x_0\rangle$. Furthermore the angle between the starting state and $|x_0\rangle$ is independent of the value of $x_0$ (as the starting state is an equal superposition of all possible $x_0$ values)

so the number of iterations is independent of $x_0$ too.

## 10.3 Some further features of Grover's algorithm

### Optimality

Grover's algorithm achieves unstructured search for a unique good item with $\frac{\pi}{4}\sqrt{N}$ queries. Is it possible to invent an even more ingenious quantum algorithm that uses fewer queries? Alas the answer is no. We'll just state (without proof):
**Theorem** Any quantum algorithm that achieves the search for a unique good item in an unstructured database of size $N$ (with any constant level of probability, say half) must use $O(\sqrt{N})$ queries. □

Even more, it can be shown that $\frac{\pi}{4}(1-\epsilon)\sqrt{N}$ queries for any $\epsilon > 0$ are insufficient, so Grover's algorithm is optimal in a tight sense.

### Searching with multiple good items

Suppose our search space contains $r \geq 1$ good items and we wish to find any one such item. Consider first the case that $r$ is known. In this case we'll see that our previous algorithm still works; we just need to modify the number of iterations in a way that depends on $r$.

Let the good items be denoted $x_1, \ldots, x_r$ so now $f(x_i) = 1$ for $i = 1, \ldots, r$ and $f(x) = 0$ for all other $x$'s. Using the same construction that gave $I_{x_0}$ from $U_f$ in the case of a single good item, we obtain the operator $I_G$ (where $G$ stands for "good") with action:

$$I_G |x\rangle = \begin{cases} |x\rangle & \text{if } x \neq x_1, \ldots, x_r \\ -|x\rangle & \text{if } x = x_1, \ldots, x_r \end{cases}$$

and we will use the iteration operator (cf eq. (16))

$$Q_G = -H_n I_0 H_n I_G = -I_{|\psi_0\rangle} I_G.$$

Let

$$|\psi_G\rangle = \frac{1}{\sqrt{r}} \sum_{i=1}^{r} |x_i\rangle$$

be the equal superposition of all good items. We can separate out the good and bad parts of the full equal superposition $|\psi_0\rangle$ writing:

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{\text{all } x} |x\rangle = \frac{\sqrt{r}}{\sqrt{N}} |\psi_G\rangle + \frac{\sqrt{N-r}}{\sqrt{N}} |\psi_B\rangle \tag{17}$$

where $|\psi_B\rangle = \frac{1}{\sqrt{N-r}} \sum_{\text{bad } x} |x\rangle$ is the equal superposition of all bad items and $|\psi_G\rangle$ and $|\psi_B\rangle$ are orthogonal states. Note that we can write $I_G = I - 2\sum_{i=1}^{r} |x_i\rangle \langle x_i| \, x_i$ which (on arbitrary vectors) is not of the form $I_{|\alpha\rangle}$ for any single vector $|\alpha\rangle$. But we still have:

**Theorem**: let $\mathcal{P}_G$ be the plane spanned by $|\psi_0\rangle$ and $|\psi_G\rangle$. Then the action of $Q_G$ preserves this plane and within $\mathcal{P}_G$ this action is rotation through angle $2\alpha$ where

$$\sin\alpha = \langle\psi_0|\psi_G\rangle = \sqrt{\frac{r}{N}}.$$

**Proof:** Clearly $I_{|\psi_0\rangle}$ preserves $\mathcal{P}_G$ since acting on any $|\psi\rangle$ it just subtracts a multiple of $|\psi_0\rangle$. For $I_G$ we note that by eq. (17), $\mathcal{P}_G$ can also be characterised as the plane spanned by the orthogonal states $|\psi_G\rangle$ and $|\psi_B\rangle$. Now $I_G|\psi_G\rangle = -|\psi_G\rangle$ and $I_G|\psi_B\rangle = |\psi_B\rangle$ so for any state $|\psi\rangle = a|\psi_G\rangle + b|\psi_B\rangle$ in $\mathcal{P}_G$ the action of $I_G$ is to subtract a multiple of $|\psi_G\rangle$ i.e. the result lies in the plane too. This also shows that *within* $\mathcal{P}_G$, $I_G$ coincides with the operation $I_{|\psi_G\rangle}$ and $Q_G = -I_{|\psi_0\rangle}I_{|\psi_G\rangle} = I_{|\psi_0^\perp\rangle}I_{|\psi_G\rangle}$. Hence exactly as before, $Q$ is a rotation through angle $2\alpha$ where $\alpha$ is the angle between $|\psi_0^\perp\rangle$ and $|\psi_G\rangle$ i.e. $\sin\alpha = \langle\psi_0|\psi_G\rangle = \sqrt{r/N}$. $\square$

Now suppose that we start with $|\psi_0\rangle$ and repeatedly apply $Q_G$. The angle between $|\psi_0\rangle$ and $|\psi_G\rangle$ is $\beta$ where $\cos\beta = \langle\psi_0|\psi_G\rangle = \sqrt{r/N}$. Each application of $Q_G$ is a rotation through $2\alpha$ where $\sin\alpha = \sqrt{r/N}$ so we need $\beta/(2\alpha) = (\arccos\sqrt{r/N})/(2\arcsin\sqrt{r/N})$ iterations to move $|\psi_0\rangle$ very close to $|\psi_G\rangle$. If $r << N$ then $|\psi_0\rangle$ and $|\psi_G\rangle$ are almost orthogonal ($\beta \approx \pi/2$) and $\alpha \approx \sin\alpha = \sqrt{r/N}$ so we need $\frac{\pi}{4}\sqrt{N/r}$ iterations.

The basic technique and use of the operator $Q_G$ in the above result can be generalised to give the so-called principle of *amplitude amplification* (which we won't discuss in this course).

## Searching with an *unknown* number of good items

*Optional, not required for exam purposes.*

We can also adapt the algorithm to work in the case that $r$ is *unknown*. The apparent difficulty is the following: if we start with $|\psi_0\rangle$ and repeatedly apply the operator $Q$ (in either case $r = 1$ or $r > 1$) we just rotate the state round and round in the plane of $|\psi_0\rangle$ and $|\psi_G\rangle$. The trick is to know when to stop i.e. when the state lines up closely with $|\psi_G\rangle$ in this plane. But if $r$ is unknown then the rotation angle $2\alpha$ of $Q$ is unknown!

To illustrate the way around this problem we'll consider only the case where the unknown $r$ is very small $r << N$. (General $r$ values can be addressed by a more complicated argument along similar lines). We choose a number $K$ *uniformly randomly* in the range $0 < K < \frac{\pi}{4}\sqrt{N}$, apply $K$ iterations of $Q$, measure the final state and test if the result is good or not. For $r << N$ each iteration is a rotation through small angle $2\alpha \approx 2\sqrt{r/N}$ i.e. we have chosen a random angle in the range 0 to $\sqrt{r}\frac{\pi}{2}$ of $\sqrt{r}$ quadrants. Equivalently we can choose one of the $\sqrt{r}$ quadrants at random and then a random angle in it. Now think of $|\psi_0\rangle$ as the $x$-axis direction and $|\psi_G\rangle$ as the $y$ axis direction (recalling that these states are almost orthogonal for $r << N$). If the final rotation angle is within $\pm 45^\circ$ of the $y$ axis then the final state $|\psi\rangle$ has $|\langle\psi|\psi_G\rangle|^2 \geq \cos^2 45^\circ = 1/2$ i.e. we have probability at least half of seeing a good item in our final measurement. Now for every quadrant, half the angles are within $\pm 45^\circ$ of the $y$ axis so our randomised procedure above, using $O(\sqrt{N})$ queries, will locate a good item with probability at least $1/4$. Repeating the

whole procedure a constant number of times, say $M = 10$ times, thus still using $O(\sqrt{N})$ queries, we will fail to locate a good item only with tiny probability $(3/4)^M = (3/4)^{10}$.

This case of unknown $r$ is directly relevant to the consideration of computational tasks in NP, where rather than *locating* a good item we want instead to know whether a good item *exists* or not. Consider for example the task SAT: given a Boolean function $f$, does it have a satisfying assignment or not? $f$ will generally have some unknown number $r \geq 0$ of satisfying assignments. We run the above randomised version of Grover's algorithm, say 10 times, checking each output $x$ to see if $f(x) = 1$ or not. If they all fail we conclude that $f$ is not satisfiable, which will be correct with high probability $1 - (3/4)^{10}$. In this way Grover's algorithm can be applied to any NP problem to provide a quadratic speedup over classical exhaustive search.

# 11 Shor's quantum factoring algorithm

We will now describe Shor's quantum factoring algorithm. Given an integer $N$ with $n = \log N$ digits this algorithm will output a factor $1 < K < N$ (or output $N$ if $N$ is a prime) with any chosen constant level of probability $1 - \epsilon$, and the algorithm will run in polynomial time $O(n^3)$. Currently the best known classical algorithm (the so-called number field sieve algorithm) runs in time $e^{O(n^{1/3}(\log n)^{2/3})}$ i.e. there is no known polynomial time classical algorithm for this task.

We'll begin by first describing some pure mathematics (number theory) – involving no quantum ingredients at all – showing how to convert the problem of factoring $N$ into a problem of periodicity determination. Then we'll use our quantum period finding algorithm to achieve the task of factorisation. We'll encounter (and deal with) a technical complication: our function will be periodic on the infinite set $\mathbb{Z}$ of all integers so for computational purposes we need to truncate this down to a *finite* size $\mathbb{Z}_M$ for some $M$ (suitably large, depending on $N$). Since we do not know the period at the outset the restricted function will not be *exactly* periodic on $\mathbb{Z}_M$: the "last" period will generally be incomplete (as $M$ is not generally an exact multiple of the period). But we'll see that if $M$ is sufficiently large (in fact $M = O(N^2)$ will suffice) then there will be enough complete periods so that the single "corrupted" period has only a negligible effect on our period finding algorithm. We will also always choose $M$ to be a power of 2 to be able to use our explicit circuit for QFT mod $M$ for such $M$'s.

## 11.1 Factoring as a periodicity problem – some number theory

Let $N$ with $n = \log N$ digits denote the integer that we wish to factorise. We start by choosing $1 < a < N$ at random. Next using Euclid's algorithm (which is a poly-time algorithm) we compute the greatest common divisor $b = gcd(a, N)$. If $b > 1$ we are finished. Thus suppose $b = 1$ i.e. $a$ and $N$ are coprime. We will use:

**Theorem 2** *(Euler's theorem): If $a$ and $N$ are coprime then there is a least power $1 < r < N$ such that $a^r \equiv 1$ mod $N$. $r$ is called the order of $a$ mod $N$.*

We omit the proof which may be found in most standard texts on number theory.

Now consider the powers of $a$ as a function of the index i.e. the modular exponential function:

$$f : \mathbb{Z} \to \mathbb{Z}_N \qquad f(k) = a^k \bmod N \tag{18}$$

Clearly $f(k_1 + k_2) = f(k_1)f(k_2)$ and by Euler's theorem $f(r) = 1$ so $f(k + r) = f(k)$ for all $k$ i.e. $f$ is periodic with period $r$. Also since $r$ is the *least* integer with $f(r) = 1$ we see that $f$ must be one-to-one within each period.

Next *suppose* we can find $r$. (We will use our quantum period finding algorithm for this). Suppose $r$ comes out to be even. Then

$$a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \bmod N$$

i.e.

$$N \text{ exactly divides the product } (a^{r/2}-1)(a^{r/2}+1) \tag{19}$$

(and knowing $r$ we can calculate each of these terms in poly$(n)$ time).

We know $N$ does not divide $a^{r/2}-1$ (since $r$ was the *least* power $x$ such that $a^x-1$ is divisible by $N$). Thus *if* $N$ does not divide $a^{r/2}+1$ i.e. if $a^{r/2} \not\equiv -1 \bmod N$, then in eq. (19) $N$ must partly divide into $a^{r/2}-1$ and partly into $a^{r/2}+1$. Hence using Euclid's algorithm again, we compute $gcd(a^{r/2}\pm 1, N)$ which will be factors of $N$.

All this works *provided* $r$ is even and $a^{r/2} \not\equiv -1 \bmod N$. How likely is this, given that $a$ was chosen at random? We quote the following theorem.

**Theorem 3** *Suppose $N$ is odd and not a power of a prime. If $a < N$ is chosen uniformly at random with $gcd(a,N)=1$ then Prob($r$ is even and $a^{r/2} \not\equiv -1 \bmod N$) $\geq 1/2$.*

For a proof of this result see Preskill's notes page 307 et seq., Nielsen/Chuang appendix 4.3 or A. Ekert and R. Jozsa, *Reviews of Modern Physics*, vol 68, p733-753 1996, appendix B.

Hence for any $N$ which is odd and not a prime power, we will obtain a factor with probability at least half. Given any candidate factor we can check it (in poly$(n)$ time) by test division into $N$. Thus repeating the process, say 10 times, we will fail to get a factor only with tiny probability $1/2^{10}$, and succeed with any probability $1-\epsilon$ with $\log_2 1/\epsilon$ repetitions.

**Example 2** *Consider $N=15$ and choose $a=7$. Then a direct calculation shows that the function $f(k)=7^k \bmod 15$ for $k=0,1,2,\ldots$ has values 1,7,4,13,1,7,4,13,... so $r=4$. Thus $7^4-1=(7^2-1)(7^2+1)=(48)(50)$ is divisible by 15 and computing $gcd(15,48)=3$ and $gcd(15,50)=5$ gives non-trivial factors of 15.*

All of this works if $N$ is not even or a prime power. So how do we recognise and treat these latter cases? If $N$ is even (which is easy to recognise!) we immediately have a factor 2 and we are finished. If $N=p^l$ is a prime power then we can identify this case and find $p$ using the following result (which we quote without proof).

**Lemma 2** *Suppose $N=c^l$ for some integers $c,l \geq 2$. Then there is a classical polynomial time algorithm that outputs $c$.*

Running this algorithm on *any* $N$ will output *some* number $c'$ and we can check if it divides $N$ or not. If $N$ was a prime power $p^l$ then $c'$ will be $p$.

**Summarizing the process so far:** given $N$ we proceed as follows.
(i) Is $N$ even? If so, output 2 and stop.
(ii) Run the algorithm of lemma 2, test divide the output and stop if a factor of $N$ is obtained.

(iii) If $N$ is neither even nor a prime power choose $1 < a < N$ at random and compute $s = gcd(a, N)$. If $s \neq 1$ output $s$ and stop.
(iv) If $s = 1$ find the period $r$ of $f(k) = a^k \bmod N$. (We will achieve this with any desired level of constant probability $1 - \epsilon$ using the quantum algorithm described in the next section).
(v) If $r$ is odd, go back to (iii). If $r$ is even compute $t = gcd(a^{r/2} + 1, N)$, so by definition $t$ is a factor of $N$. If $t \neq 1, N$ output $t$. If $t = 1$ or $N$ go back to (iii) and try again.

According to theorem 3 any run of (iv) and (v) will output a factor with probability $> 1/2$ so $K$ repetitions of looping back to (iii) will all fail only with probability $< 1/2^K$ which can be made as small as we like.

## 11.2 Computing the period of $f(k) = a^k \bmod N$

Let $r$ denote the (as yet unknown) period of $f(k) = a^k \bmod N$ on the infinite domain $\mathbb{Z}$. We will work on the finite domain $D = \{0, 1, \ldots, 2^m - 1\}$ where $2^m$ is the least power of 2 greater than $N^2$ (see later for the reason for this choice). Let $2^m = Br + b$ with $1 < b < r$ i.e. the domain $D$ contains $B$ full periods and only the initial part up to $b$ of the next period. Using a standard application of computation by quantum parallelism we manufacture the state $\frac{1}{\sqrt{2^m}} \sum_{x \in D} |x\rangle |f(x)\rangle$ and measure the second register to obtain some value $y_0 = f(x_0)$ with $0 \leq x_0 < r$. In the first register we get the state

$$|per\rangle = \frac{1}{\sqrt{A}} \sum_{k=0}^{A-1} |x_0 + kr\rangle$$

where

$$A = \begin{cases} B + 1 = \lfloor \frac{2^m}{r} \rfloor + 1 & \text{if } x_0 < b \\ B = \lfloor \frac{2^m}{r} \rfloor & \text{if } x_0 \geq b. \end{cases} \tag{20}$$

Let

$$QFT_{2^m} |per\rangle = \sum_{c=0}^{2^m - 1} \tilde{f}(c) |c\rangle .$$

Writing $\omega = e^{2\pi i / 2^m}$ we have

$$\tilde{f}(c) = \frac{1}{\sqrt{A}\sqrt{2^m}} \sum_{k=0}^{A-1} \omega^{c(x_0 + kr)} = \frac{\omega^{cx_0}}{\sqrt{A}\sqrt{2^m}} \left[ \sum_{k=0}^{A-1} \omega^{crk} \right].$$

As before (as in eq. (8), where $c$ was called $y$) the square bracket is a geometric series with ratio $\alpha = \omega^{cr}$ and we have

$$[\ldots] = 1 + \alpha + \alpha^2 + \ldots + \alpha^{A-1} = \begin{cases} \frac{1 - \alpha^A}{1 - \alpha} & \text{for } \alpha \neq 1 \\ A & \text{for } \alpha = 1. \end{cases}$$

Let's look more closely at the ratio $\alpha = e^{2\pi i cr / 2^m}$. Previously we had $r$ dividing the denominator $2^m$ exactly and $2^m / r = A$ so if $\alpha \neq 1$ then $\alpha$ was an $A^{\text{th}}$ root of unity and the geometric series summed to *zero* in all these cases. The only $c$ values that survived

were the exact multiples of $A = 2^m/r$ having $\alpha = 1$. There were $r$ such multiples each with equal |amplitude| of $\frac{1}{\sqrt{r}}$.

In the present case $r$ does not divide $2^m$ exactly generally so $\alpha$ is not an $A^{\text{th}}$ root of unity and we don't get a lot of "exactly zero" amplitudes for $|c\rangle$'s! However we aim to show that a measurement on QFT$|per\rangle$ will yield an integer $c$-value which is *close* to a multiple of $2^m/r$ with suitably high probability.

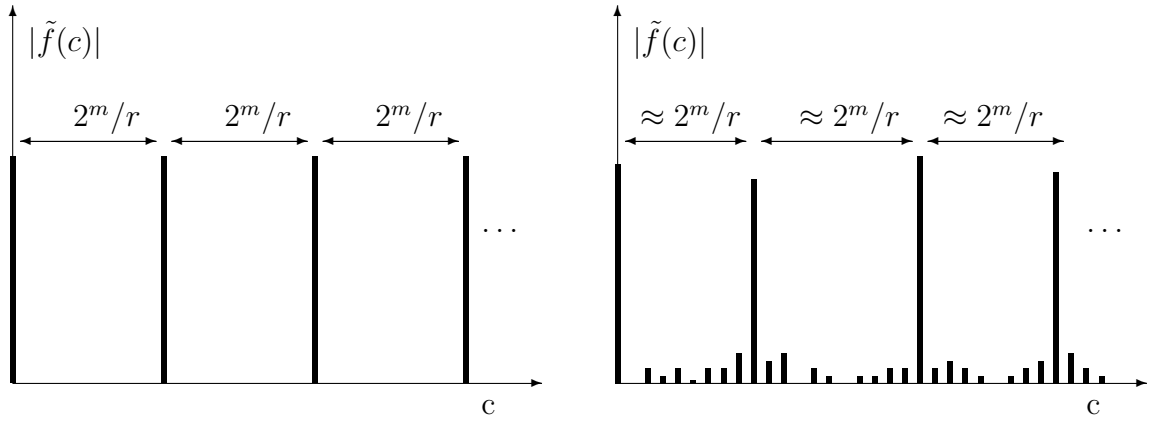
Consider the $r$ multiples of $2^m/r$ (which are now not integers necessarily!):

$$0, \frac{2^m}{r}, 2(\frac{2^m}{r}), \ldots, (r-1)(\frac{2^m}{r}).$$

Each of these is within half of a unique nearest integer. Note that $k(2^m/r)$ can never be exactly half way between two integers since $r < N$ and $2^m > N^2$, so (using 2's in $2^m$) all factors of 2 can be cancelled out of the denominator $r$. Thus we consider $c$ values ($r$ of them) such that

$$|c - k\frac{2^m}{r}| < \frac{1}{2} \qquad k = 0, 1, \ldots, (r-1). \tag{21}$$

In the previous case of exact periodicity (where $2^m/r$ was an integer) each of these $c$-values appeared with probability $1/r$ and all other $c$-values had probability zero. Here we will show that although the other $c$-values will generally have non-zero probabilities, the special ones in eq. (21) still have probability at least $\gamma/r$ for a constant $\gamma$.




(a) exact periodicity (b) inexact periodicity

Figure 11.2: Schematic depiction of amplitudes in QFT$|per\rangle$. (a) exact periodicity ($r$ divides $2^m$): we have nonzero amplitudes only at exact multiples $c = k2^m/r$. (b) non-exact periodicity: we have nonzero amplitudes for many $c$-values but the integers nearest to the multiples $k2^m/r$ still have suitably large amplitudes.

**Theorem 4** *Suppose we measure the label in QFT$|per\rangle$. Let $c_k$ be the unique integer with $|c - k\frac{2^m}{r}| < \frac{1}{2}$. Then prob$(c_k) > \gamma/r$ where $\gamma \approx 4/\pi^2$.*

**Proof:** (optional) For any $c$ we have

$$\text{prob}(c) = |\tilde{f}(c)|^2 = \frac{1}{A2^m}\left|\frac{1-\alpha^A}{1-\alpha}\right|^2$$

with $\alpha = e^{2\pi i cr/2^m} = e^{2\pi i(cr \bmod 2^m)/2^m}$. For our special $c$-values satisfying eq. (21) we have $|cr - k2^m| < r/2$ so

$$-\frac{r}{2} < cr \bmod 2^m < \frac{r}{2}. \tag{22}$$

Write $\alpha = e^{i\theta_c}$ with $\theta_c = 2\pi(cr \bmod 2^m)/2^m$ so $|\theta_c| < \pi r/2^m$. Also from eq. (20) we see that in all cases $A < 2^m/r + 1$ so

$$|A\theta_c| < \frac{\pi r}{2^m}A < \pi(1 + \frac{r}{2^m}).$$

Write $A\theta_{max} = \pi(1 + r/2^m)$. Note that for all $c$

$$0 \le |A\theta_c/2| < A\theta_{max}/2 < \pi. \tag{23}$$

To estimate prob($c$) we'll use the algebraic identity

$$\left|\frac{1-e^{iA\theta}}{1-e^{i\theta}}\right|^2 = \left(\frac{\sin A\theta/2}{\sin\theta/2}\right)^2.$$

We have

$$\begin{aligned}
\text{Prob}(c) &= \tfrac{1}{A2^m}\left(\tfrac{\sin A\theta_c/2}{\sin\theta_c/2}\right)^2 \\
&> \tfrac{1}{A2^m}\left(\tfrac{\sin A\theta_c/2}{\theta_c/2}\right)^2 \quad (\text{as } \sin x < x) \\
&= \tfrac{A}{2^m}\left(\tfrac{\sin A\theta_c/2}{A\theta_c/2}\right)^2 \\
&> \tfrac{A}{2^m}\left(\tfrac{\sin A\theta_{\max}/2}{A\theta_{\max}/2}\right)^2
\end{aligned}$$

where the last inequality follows from eq. (23) and the fact that $\frac{\sin x}{x}$ is decreasing on $0 < x < \pi$.

Next from eq. (20) we have $A > 2^m/r - 1$ so $\frac{A}{2^m} > \frac{1}{r} - \frac{1}{2^m}$. Introducing $g(x) = \left(\frac{\sin x}{x}\right)^2$ we have

$$\text{prob}(c) > (\frac{1}{r} - \frac{1}{2^m})g(A\theta_{max}/2) = \frac{1}{r}(1 - \frac{r}{2^m})g(A\theta_{max}/2) > \frac{\gamma}{r} \tag{24}$$

for a constant $\gamma$, noting that $2^m > N^2$ and $r < N$ so $r/2^m << 1$ for all large $N$. To get a proper lower bound for $\gamma$ is straightforward but a little messy. Here we will just consider the case of very large $N$ and ignore terms of order $r/2^m < 1/N$. We have $A\theta_{max}/2 = \frac{\pi}{2}(1+r/2^m) \approx \pi/2$ so $g(\pi/2) = (2/\pi)^2$ and from eq. (24) we get prob($c$) $> \gamma/r$ for $\gamma \approx 4/\pi^2$. $\square$

According to this theorem, for each $k = 0, \ldots, r-1$ we will obtain the unique $c$-value satisfying eq. (21) with probability at least $\gamma/r$. We will be especially interested in those $c$'s for which the corresponding $k$ is *coprime* to $r$ and there are $O(r/\log\log r)$ of these. Hence the total probability of obtaining such a "good" $c$-value is $O(1/\log\log r) > O(1/\log\log N)$ and with $O(\log\log N)$ repetitions we will obtain such a good $c$-value with any desired constant level of probability. To complete the determination of $r$ and hence the description of the quantum factoring algorithm, it remains to show that $r$ can be determined from a ("good") $c$-value in time poly($\log N$).

## 11.3 Getting $r$ from a good $c$ value

Suppose we have $c$ satisfying eq. (21) i.e.

$$\left|\frac{c}{2^m} - \frac{k}{r}\right| < \frac{1}{2^{m+1}}. \tag{25}$$

Recall that $r < N$ and $2^m > N^2$ so

$$\left|\frac{c}{2^m} - \frac{k}{r}\right| < \frac{1}{2N^2} \quad \text{with } r < N \tag{26}$$

and $c/2^m$ is a *known* fraction. We claim that there is at most one fraction $k'/r'$ with a denominator $r'$ less than $N$ satisfying eq. (26). Hence for given $c/2^m$, eq. (26) determines $k/r$ uniquely. To prove this claim suppose $k'/r'$ and $k''/r''$ both lie within $1/(2N^2)$ of $c/2^m$. Then

$$\left|\frac{k'}{r'} - \frac{k''}{r''}\right| = \frac{|k'r'' - r'k''|}{r'r''} \geq \frac{1}{r'r''} > \frac{1}{N^2} \tag{27}$$

But $k'/r'$ and $k''/r''$ are both within $1/(2N^2)$ of $c/2^m$ so they must be within $1/N^2$ of each other, contradicting eq. (27). Hence there is at most one $k/r$ with $r < N$ satisfying eq. (26).

This result is the reason why we chose $2^m$ to be greater than $N^2$: it guarantees that the bound on RHS of eq. (26) is $< 1/(2N^2)$ and then $k/r$ is uniquely determined from $c/2^m$.

**Example 3** *Suppose we wish to factor $N = 39$ and we have chosen $a = 7$ which is coprime to $N$. Let $r$ be the period of $f(x) = 7^x \bmod 39$. We have $N^2 = 1521$ and $2^{10} < N^2 < 2^{11} = 2048 = 2^m$ so $m = 11$. Suppose the measurement of $QFT_{2^m}|per\rangle$ yields $c = 853$. According to our theory, this number has a "reasonable" probability to be within half of a multiple $k2^{11}/r$ of $2^m/r$. If this is actually the case then our theory guarantees that the fraction $k/r$ is uniquely determined, as the unique fraction $k/r$ with denominator $< 39$ that is within $1/2^{m+1} = 1/2^{12}$ of $853/2048$. In this example we can (with a calculator) check all fractions $a/b$ with $a < b < N = 39$ to see which ones (if any) satisfy*

$$\left|\frac{a}{b} - \frac{853}{2048}\right| < \frac{1}{2^{12}}. \tag{28}$$

*There are $O(N^2)$ such fractions to try. We find that there is only one viz. $a/b = 5/12$ that satisfies eq. (28):*

$$\left|\frac{a}{b} - \frac{853}{2048}\right| = 0.000163 < \frac{1}{2^{12}} = 0.000244$$

*This result is consistent with $k = 5$ and $r = 12$ and also with $k = 10$ and $r = 24$. But our theory also guarantees that $k$ is coprime to $r$ with "reasonable" probability which in this case sets $r = 12$. We can then verify that $7^{12}$ is indeed congruent to 1 mod 39 and $7^x$ for all $x < 12$ is not congruent to 1 so $r = 12$ is the correct period.*

So far we have that $k/r$ is *uniquely determined* by $c/2^m$ but how do we actually compute $k/r$ from $c/2^m$? In the above example we were able to try out all candidate fractions $k'/r'$ with denominator less than $N$. But there are generally $O(N^2)$ such fractions to try so this method of seeking the unique one is *not efficient*, requiring at least $O(N^2)$ steps, which is exponential in $n = \log N$!

To obtain an efficient (i.e. poly($n$) time) method we invoke the elegant mathematical:

**Theory of continued fractions**

Any rational number $s/t$ (with $s < t$) may be expressed as a so-called continued fraction:

$$\frac{s}{t} = \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\cdots + \frac{1}{a_l}}}} \tag{29}$$

where $a_1, \ldots, a_l$ are positive integers. To do this we begin by writing $s/t = 1/(t/s)$. Since $s < t$ we have $t/s = a_1 + s_1/t_1$ with $a_1 \geq 1$ and $s_1 < t_1 = s$ and so

$$\frac{s}{t} = \frac{1}{a_1 + \frac{s_1}{t_1}}.$$

Then repeating with $s_1/t_1$ we get $t_1/s_1 = a_2 + s_2/t_2$, $t_2 = s_1$ and

$$\frac{s}{t} = \frac{1}{a_1 + \frac{1}{a_2 + \frac{s_2}{t_2}}}.$$

Continuing in this way we get a sequence of integers $a_k, s_k$ and $t_k$. Note that $s_k < t_k$ and $t_{k+1}$ is always given by $s_k$. Hence the sequence $t_k$ of denominators is strictly a decreasing sequence of non-negative integers and hence the process must always terminate, after some number $l$, of iterations giving the expression in eq. (29).

To avoid the cumbersome "fractions of fractions" notation in eq. (29) we will write

$$\frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\cdots + \frac{1}{a_l}}}} = [a_1, a_2, \ldots, a_l]. \tag{30}$$

For each $k = 1, \ldots, l$ we can truncate the fraction in (30) at the $k^{\text{th}}$ level to get a sequence of rational numbers

$$\frac{p_1}{q_1} = [a_1] = \frac{1}{a_1}, \quad \frac{p_2}{q_2} = [a_1, a_2] = \frac{1}{a_1 + \frac{1}{a_2}} = \frac{a_2}{a_1 a_2 + 1}, \quad \cdots$$

$$\frac{p_k}{q_k} = [a_1, \ldots, a_k], \quad \ldots \quad \frac{p_l}{q_l} = [a_1, \ldots, a_l] = \frac{s}{t}.$$

$p_k/q_k$ is called the $k^{\text{th}}$ *convergent* of the continued fraction of $s/t$.

Continued fractions enjoy the following tantalising properties.

**Lemma 3** *Let $a_1, \ldots, a_l$ be any positive numbers (not necessarily integers here). Set $p_0 = 0$, $q_0 = 1$, $p_1 = 1$ and $q_1 = a_1$.*
*(a) Then $[a_1, \ldots, a_k] = p_k/q_k$ where*

$$p_k = a_k p_{k-1} + p_{k-2} \qquad q_k = a_k q_{k-1} + q_{k-2} \qquad k \geq 2. \tag{31}$$

*Note that if the $a_k$'s are integers then so are the $p_k$'s and $q_k$'s.*
*(b) $q_k p_{k-1} - p_k q_{k-1} = (-1)^k$ for $k \geq 1$.*
*(c) If $a_1, \ldots, a_l$ are integers then $gcd(p_k, q_k) = 1$ for $k \geq 1$.*

**Proof outline** (optional):
(a) By induction on $k$. For the base case $k = 2$ direct calculation gives $[a_1, a_2] = a_2/(a_1 a_2 + 1)$ and eq. (31) correctly gives $p_2 = a_2$ and $q_2 = a_1 a_2 + 1$. Thus suppose eq. (31) holds for length $k$. For length $k+1$ we have $[a_1, \ldots a_k, a_{k+1}] = [a_1, \ldots, a_{k-1}, a_k + 1/a_{k+1}]$ where the RHS now has length $k$. Let $\tilde{p}_j/\tilde{q}_j$ be the sequence of convergents of RHS. Then $\tilde{p}_k/\tilde{q}_k = [a_1, \ldots a_k, a_{k+1}] = [a_1, \ldots, a_{k-1}, a_k + 1/a_{k+1}]$ and clearly $\tilde{p}_{k-1} = p_{k-1}$, $\tilde{p}_{k-2} = p_{k-2}$ and similarly for the $q$'s. Hence using the recurrence relation eq. (31) at length $k$ (twice) we get:

$$\frac{\tilde{p}_k}{\tilde{q}_k} = \frac{(a_k + 1/a_{k+1})p_{k-1} + p_{k-2}}{a_k + 1/a_{k+1})q_{k-1} + q_{k-2}} = \frac{p_k + p_{k-1}/a_{k+1}}{q_k + q_{k-1}/a_{k+1}} = \frac{a_{k+1}p_k + p_{k-1}}{a_{k+1}q_k + q_{k-1}}$$

i.e. eq. (31) holds for $k + 1$.

(b) is proved by induction on $k$ using the recurrence relations of (a) to express the $(k, k-1)$ expression in terms of the same expression with lower values of the subscripts.

(c) follows from (b): if $a$ divides $p_k$ and $q_k$ exactly then by (b), $a$ must divide $\pm 1$ i.e. $a = 1$. □

**Theorem 5** *Consider the continued fraction $s/t = [a_1, \ldots, a_l]$. Let $p_k/q_k = [a_1, \ldots, a_k]$ be the $k^{\text{th}}$ convergent for $k = 1, \ldots, l$. If $s$ and $t$ (cancelled to lowest terms) are $m$ bit integers then the length $l$ of the continued fraction is $O(m)$ and this continued fraction together with its convergents can be calculated in time $O(m^3)$.*

**Proof outline** (optional):
We have $a_k \geq 1$ and $p_k, q_k \geq 1$ so by the above recurrence relations, $p_k$ and $q_k$ must be increasing sequences and $p_k = a_k p_{k-1} + p_{k-2} \geq 2p_{k-2}$. Similarly $q_k \geq 2q_{k-2}$. Hence $p_k$ and $q_k$ are each $\geq 2^{\lfloor k/2 \rfloor}$ so since $p_k$ and $q_k$ are coprime and increasing, we must get $s/t$ after at most $l = O(m)$ iterations. The computation of each successive $a_k$ involves the division of $O(m)$ bit integers (and splitting off the integer parts). These arithmetic operations can be performed in $O(m^2)$ time so we can compute all $O(m)$ $a_k$'s in $O(m^3)$ time. Similarly using the recurrence relation we can compute all $p_k$'s and $q_k$'s in $O(m^3)$ time too. □

**Theorem 6** *Let $0 < x < 1$ be a rational number and suppose that $p/q$ is a rational number such that*

$$\left| x - \frac{p}{q} \right| < \frac{1}{2q^2}.$$

*Then $p/q$ is a convergent of the continued fraction of $x$.*

**Proof** (optional):
Let $p/q = [a_1, \ldots, a_n]$ be the CF expansion of $p/q$ with convergents $p_j/q_j$, so $p_n/q_n = p/q$. Introduce $\delta$ defined by

$$x = \frac{p_n}{q_n} + \frac{\delta}{2q_n^2} \tag{32}$$

so $|\delta| < 1$. We aim to show that the CF of $x$ is an extension of the CF of $p/q$ i.e. we want to construct $\lambda$ rational so that $x = [a_1, \ldots, a_n, \lambda]$. In view of lemma 3(a) define $\lambda$ by $x = (\lambda p_n + p_{n-1})/(\lambda q_n + q_{n-1})$. Using eq. (32) to replace $x$ we get

$$\lambda = 2\left(\frac{q_n p_{n-1} - p_n q_{n-1}}{\delta}\right) - \frac{q_{n-1}}{q_n}.$$

By lemma 3(b), $q_n p_{n-1} - p_n q_{n-1} = (-1)^n$. We may assume that this is the same as the sign of $\delta$ since if it is the opposite sign then from the start write $p/q = [a_1, \ldots, a_n - 1, 1]$ so the value of $n$ is increased by 1 and the sign is flipped. Thus without loss of generality we can assume that $(q_n p_{n-1} - p_n q_{n-1})/\delta$ is positive and so

$$\lambda = \frac{2}{\delta} - \frac{q_{n-1}}{q_n} > 2 - 1 > 1$$

(as $|\delta| < 1$ and $q_{n-1} < q_n$). Next let $\lambda = b_0 + \lambda'$ where $b_0$ is te integer part and $0 < \lambda' < 1$ and write $\lambda' = [b_1, \ldots, b_m]$. So $x = [a_1, \ldots, a_n, \lambda] = [a_1, \ldots, a_n, b_0, b_1, \ldots, b_m]$ i.e. $p/q$ is a convergent of the CF of $x$ as required. (In the last argument we also used the easily proven fact that the CF expansion of any number is unique, except for the above trick of splitting 1 off from the last term i.e. if $[a_1, \ldots, a_n] = [b_1, \ldots, b_m]$ and $a_n, b_m \neq 1$ then $m = n$ and $a_i = b_i$). $\square$

**Remark**: Theorem 6 actually remains true for irrational $x$ too. For an irrational number the continued fraction development does not terminate – we get an infinitely long continued fraction and corresponding infinite sequence of rational convergents $p_k/q_k$ $k = 1, 2, \ldots$. This sequence provides an efficient method of computing excellent rational approximations to an irrational recalling that $q_k$ grows exponentially with $k$ and (by theorem 6) it determines the accuracy of the approximation. $\square$

Now let us return to our problem of getting $r$ from the knowledge of $c$ and $2^m$ satisfying eq. (26):

$$\left|\frac{c}{2^m} - \frac{k}{r}\right| < \frac{1}{2N^2} \quad \text{and } r < N.$$

We know that there is (at most) a unique such fraction $k/r$ and according to theorem 6 this fraction must be a convergent of the continued fraction of $c/2^m$. Since $2^m = O(N^2)$ we have that $c$ and $2^m$ are $O(n)$ bit integers and the computation of all the convergents can be performed in time $O(n^3)$. So we do this computation and finally check through the list of $O(n)$ convergents to find the unique one satisfying eq. (26), and read off $r$ as its denominator.

**Example 4** *(Continuation of example 3).*
*Suppose we have obtained* $c = 853$ *with* $2^m = 2^{11} = 2048$. *We develop* $853/2048$ *as a continued fraction:*

$$\frac{853}{2048} = 1/(2048/853); \quad \frac{2048}{853} = 2 + \frac{342}{853}; \quad \frac{853}{243} = 2 + \frac{169}{342};$$

$$\frac{342}{169} = 2 + \frac{4}{169}; \quad \frac{169}{4} = 42 + \frac{1}{4}; \quad \frac{4}{1} = 4 + 0$$

*so*

$$\frac{853}{2048} = [2, 2, 2, 42, 4].$$

*The convergents are*

$$[2] = \frac{1}{2}; \quad [2, 2] = \frac{2}{5}; \quad [2, 2, 2] = \frac{5}{12}; \quad [2, 2, 2, 42] = \frac{212}{509}; \quad [2, 2, 2, 42, 4] = \frac{852}{2048}.$$

*Checking these five fractions we find only* $5/12$ *as being within* $1/2^{12}$ *of* $853/2048$ *and having denominator* $< 39$.

In appendix 11.4 we will reconsider all the ingredients of Shor's quantum factoring algorithm and assess its polynomial time complexity in more detail.

## 11.4 Assessing the complexity of Shor's algorithm

*This subsection is optional and not examinable.*

Let us now consider all the parts of the quantum factoring algorithm and assess the time complexity of the whole process. Recall that the best known classical algorithm to factor $N$ with $n = \log N$ digits runs in a time that's exponential in $n^{1/3}$.

Consider the case where $N$ is neither even nor a prime power and $a < N$ chosen at random is coprime to $N$. In this case we must proceed to use the quantum part of the overall algorithm summarised at the end of section 11.1 i.e. the quantum part (iv), in addition to some further classical computational steps as well.

We first need to compute the function $f(k) = a^k \mod N$ (in superposition) over a domain $0 \leq k < 2^m$ where $2^m = O(N^2)$ so $m = O(n)$. To compute $a^k$ we use repeated squaring of $a$ $\lfloor \log k \rfloor$ times. Once the exponent is close to $k$ we do a few more multiplications to reach $k$ itself. This requires $O(\log k) = O(m) = O(n)$ multiplications of integers mod $N$. Each such multiplication can be performed in $O(n^2)$ time (by the standard "long multiplication" algorithm) so the computation of $f(k)$ for any $0 \leq k < 2^m$ can be performed in $O(n^3)$ steps. To compute the uniform superposition of all inputs for this computation we need $m = O(n)$ initial Hadamard operations. Thus the state $|f\rangle = \frac{1}{\sqrt{2^m}} \sum |k\rangle\,|f(k)\rangle$ can be computed in $O(n^3)$ steps.

**Remark**
There exist algorithms for integer multiplication that are faster than $O(n^2)$ time, running

in time $O(n \log n \log\log n)$ so the above $O(n^3)$ can be improved to $O(n^2 \log n \log\log n)$. $\square$

Next we perform measurements on the output register of $O(n)$ qubits i.e. $O(n)$ single qubit measurements. Then we apply QFT mod $2^m$ to obtain the state QFT$|per\rangle$. We have seen in section 9.3 that QFT mod $2^m$ may be implemented in $O(m^2) = O(n^2)$ steps.

**Remark**
There is a further subtle issue here. To implement QFT mod $2^m$ we will need controlled $R_d$ gates (cf eq. (11)) with smaller and smaller phases $e^{i\pi/2^d}$ for $d = O(m)$, which potentially involves an implementational cost that grows with $m$. However it can be shown that we can neglect these gates for very small phases, giving an inexact but still suitably good approximation to QFT for the factoring algorithm to work, and still have implementational cost $O(n^2)$. $\square$

Next we measure the state QFT$|per\rangle$ ($O(n)$ single qubit measurements again) to obtain the value that we called $c$ in section 11.3. Thus to get such a value the number of steps is $O(n^2 \log n \log\log n) + O(n) + O(n^2) + O(n) = O(n^2 \log n \log\log n)$. To get the period $r$ we need $c$ to be a "good" $c$ value i.e. $c/2^m$ is close to a multiple $k/r$ of $1/r$ where $k$ is coprime to $r$. To achieve this with a constant level of probability, $O(\log\log N) = O(\log n)$ repetitions of the above process suffice i.e. $O(n^2 (\log n)^2 \log\log n)$ steps in all.

**Remark**
Actually it may be shown that a *constant* number of repetitions suffices here (instead of $O(\log n)$) to determine $r$. Suppose that in two repetitions we obtain $k_1/r$ and $k_2/r$ with neither $k_1$ nor $k_2$ coprime to $r$. Then we will determine $r_1$ and $r_2$ which are the denominators of $k_1/r$ and $k_2/r$ cancelled to lowest terms i.e. $r_1$ and $r_2$ will be randomly chosen factors of $r$. Then, according to a further theorem of number theory, if we compute the least common multiple $\tilde{r}$ of $r_1$ and $r_2$ we will have $\tilde{r} = r$ with probability at least $1/4$. $\square$

To get $r$ from $c$ we use the (classical) continued fractions algorithm which required $O(n^3)$ steps. Finally to obtain our factor of $N$ we (classically) compute $t = gcd(a^{r/2+1}, N)$ using Euclid's algorithm which requires $O(n^3)$ steps for $n$ digit integers. If $r$ was odd or $r$ is even but $t = 1$ then we go back to the start. But we saw that the good case "$r$ is even and $t \neq 1$" will occur with any fixed constant level of probability $1 - \epsilon$ after a constant number $O(\log 1/\epsilon)$ of such repetitions.

Hence the time complexity of the entire algorithm is $O(n^3)$ (or actually slightly better with optimized algorithms and a more careful complicated analysis). It is amusing to note that the "bottlenecks" of the algorithms performance i.e. the sections requiring the highest degree polynomial running times, are actually the *classical* processing sections and not the novel quantum parts!