Quantum Numerical Algorithms: An Introduction Part I

Anthony Williams

March 16, 2019

1 Why Quantum Computing?

In the early 1980's Richard Feynman (1982) [3] asked the question "what kind of computer are we going to use to simulate physics?". He concluded that, since the world is fundamentally quantum mechanical, the essential problem is the simulation of quantum systems. It makes sense that the best way to simulate quantum physics is by using a 'quantum' computer which takes advantage of the effects of quantum mechanics.

The Church-Turing thesis of classical computing, concerning computable functions, was extended by David Deutsch [2] in 1985 by stating not only that a function is computable by a human being if and only if it is computable by a Turing machine but that every realisable physical process can be perfectly simulated by a universal computing device. A Turing machine operating on the natural numbers can only represent computable numbers which can be used in place of the real numbers in many situations. However classical physics, which relies upon the concept of real numbers, cannot be simulated by a Turing machine. Deutsch (1985) [2] showed that quantum computers can simulate classical physics assuming that every physical process can be described by the laws of quantum physics.

Interest in quantum computing has grown dramatically in recent decades due to the perceived possibility that quantum computers offer a means of performing computations of great complexity that are infeasible on classical machines. The possible power of quantum computation has been well documented, see Simon (1997) [9], hence the possible returns from investing in quantum computing technology could be astronomical. The power of quantum computation comes from the way that information is represented in a quantum computer. In a classical computer the basic unit of information is the 'bit' which can exist in one of two distinct states 0 and 1. In a quantum computer not only can information exist as a 0 or a 1 but also as superposition of both simultaneously. In a quantum machine the basic unit of information is called a 'qubit'. An operation acting upon a qubit effectively acts upon both 0 and 1 at the same time so by performing one operation on the qubit the operation has been performed on two different values. It follows that a two-qubit system performs the operation on four values, a three-qubit system on eight values and generally an n qubit system performs the operation on 2^n values. As the number of qubits increases the 'quantum parallelism' of the system increases exponentially. This 'quantum parallelism' is what allows certain problems to solved significantly quicker on a quantum computer than on a classical machine since exponentially large amounts of information may be processed in polynomial time. One key example of the power of quantum parallelism is the factoring of large numbers using Shor's algorithm which could break current cryptographic systems in seconds if a large enough quantum computer could be constructed.

The mechanisms which give quantum computers their greatest strength also provide one of the most significant obstacles to creating a scalable quantum computing system. A qubit in a coherent state, existing as a superposition of the two classical states, is very sensitive to interactions with its surroundings. Any measurable interaction with the environment causes the qubit to decohere into one of the two classical states. This is a significant problem as this decoherence removes the parallelism which gives a quantum computer its power. This sensitivity is a real issue when trying to control the machine or when attempting to obtain results from the machine as the coherence necessary for the quantum parallelism can easily be destroyed. This is why it is so difficult to scale quantum machines to large numbers of qubits since adding more qubits increases the sensitivity of the system to environmental interactions. The success of a hardware implementation of a quantum computer is measured in terms of its decoherence error caused by the coupling between the artificial quantum system and the unpredictable external environment.

There are many plausible physical implementations of a quantum computer, the large number of possibilities shows the relative immaturity of quantum computing as a field. The most popular implementations at present are the trapped ion quantum computer and the superconducting quantum computer. The trapped ion quantum computer works by using lasers to change the spin states of supercooled ions trapped in an electromagnetic field. Superconducting quantum computing is implemented by using small superconducting circuits (Josephson junctions). The vast array of physical hardware implementations of quantum computers and the technical details for describing them are beyond the scope of this work. Here we are concerned only with the 'software' required for simulating physical systems and performing complex calculations on quantum computers. Large scale general purpose quantum computers are a long way off realisation however small scale quantum computers might soon be used by classical computers to perform certain calculations more quickly. One such use of these small scale quantum computers could be simulating chemical or biological systems or image processing. It is likely that quantum computing will become increasingly ubiquitous so it is important that the tools required to use them effectively are developed alongside the necessary hardware.

As mentioned earlier quantum computers have many possible applications with far reaching consequences. One of the first algorithms to bring quantum computing to a wider audience was Peter Shor's algorithm [8] for factoring large numbers. Shor's algorithm, implemented on a large quantum computer, would essentially break RSA public key encryption which is currently used throughout the world. Luckily quantum computing also allows for the possibility of quantum cryptography, using quantum key distribution, which would be far more se-

cure than traditional cryptographic methods. Another important algorithm is Lov Grover's database search algorithm [4] which finds a specified entry in an unordered database, Grover's algorithm gives a polynomial time speed up when compared with traditional algorithms. There are many proposed quantum algorithms which exist to model physical systems and solve complicated mathematical problems such as Poisson's equation [1] or linear systems of equations [5]. In order to understand how these algorithms work it necessary to introduce some of the notation and mathematical tools that are used to describe quantum numerical algorithms.

2 Mathematical Preliminaries

The qubit is the basic unit of information in quantum computation and exists in the mathematical abstraction of a vector space. Quantum states behave in an analogous way to physical vectors and share the basic properties that vectors have. A vector space V is a nonempty set with elements $\mathbf{u}$, $\mathbf{v}$ for which vector addition and scalar multiplication are defined i.e. $\mathbf{u} + \mathbf{v} = \mathbf{w} \in V$ and $\alpha\mathbf{u} \in V$ where α is a scalar. The usual axioms hold for the vector space V including associativity of addition, existence of a zero vector, additive inverses and commutativity of addition etc.

A particularly important vector space in quantum computation is the Hilbert space $\mathbb{C}^n$, which is the vector space of n-tuples of complex numbers. A Hilbert space is essentially a vector space with an inner product and a norm defined by that inner product. The elements of $\mathbb{C}^n$ are labelled by $|u\rangle$, $|v\rangle$, $|w\rangle$ etc. An element of this vector space is written down as an n-dimensional column vector in the following way:

$$|v\rangle = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}. \tag{1}$$

The notation $|v\rangle$ is called 'bra-ket' or Dirac notation. The column vector $|v\rangle$ is referred as 'ket-v' and the dual vector of $|v\rangle$, given by

$$\langle v| = \overline{\mathbf{v}^T} = \begin{bmatrix} \overline{v_1} & \overline{v_2} & \dots & \overline{v_n} \end{bmatrix} \tag{2}$$

is referred to as 'bra-v' where $\overline{\mathbf{v}}$ is the complex conjugate of $\mathbf{v}$. In general the components of the vector v_i are complex.

Let α_i be a set of complex coefficients and $|v_i\rangle$ be a set of vectors then a linear combination of these vectors is given by

$$\alpha_1 |v_1\rangle + \alpha_2 |v_2\rangle + \ldots + \alpha_n |v_n\rangle = \sum_{i=1}^{n} \alpha_i |v_i\rangle . \tag{3}$$

If a given set of vectors $|v_1\rangle$, $|v_2\rangle$, ..., $|v_n\rangle$ can be used to represent any vector $|u\rangle$ in the vector space V then the set $\{|v_i\rangle\}$ spans the vector space. If

$$\alpha_1 |v_1\rangle + \alpha_2 |v_2\rangle + \ldots + \alpha_n |v_n\rangle = 0, \tag{4}$$

and at least one of the $\alpha_i \neq 0$, then the set $\{|v_i\rangle\}$ is linearly dependent i.e. one vector can be written as a linear combination of the other vectors in the set.

A spanning set of vectors for a given vector space V is not unique. For example consider the complex vector space $\mathbb{C}^2$, an arbitrary column vector with two elements may be written in the following ways:

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \alpha \begin{bmatrix}1\\ 0\end{bmatrix} + \beta \begin{bmatrix}0\\ 1\end{bmatrix} = \left(\frac{\alpha+\beta}{2}\right)\begin{bmatrix}1\\ 1\end{bmatrix} + \left(\frac{\alpha-\beta}{2}\right)\begin{bmatrix}1\\ -1\end{bmatrix}. \tag{5}$$

So the set $\left\{\begin{bmatrix}1 & 0\end{bmatrix}^T, \begin{bmatrix}0 & 1\end{bmatrix}^T\right\}$ and the set $\left\{\begin{bmatrix}1 & 1\end{bmatrix}^T, \begin{bmatrix}1 & -1\end{bmatrix}^T\right\}$ both span the vector space $\mathbb{C}^2$.

A set of linearly independent vectors which span a vector space form a basis of that space. Any vector in the space V may be expressed in terms of a linear expansion on a basis set. The dimension of a vector space V is equal to the number of elements in the basis set. Both of the sets given above form a basis set for $\mathbb{C}^2$ as both sets span the vector space and the two vectors in each are linearly independent.

The vector space $\mathbb{C}^n$ is a Hilbert space as it has an inner product and a norm defined by that inner product. The inner product of a vector space assigns a scalar value to each pair of vectors in the space. Using the Dirac notation the inner product of two vectors is given by

$$\langle u|v\rangle = \overline{\mathbf{u}^T}\mathbf{v} = \begin{bmatrix}\overline{u_1} & \overline{u_2} & \ldots & \overline{u_n}\end{bmatrix}\begin{bmatrix}v_1\\ v_2\\ \vdots\\ v_n\end{bmatrix} = \overline{u_1}\cdot v_1 + \overline{u_2}\cdot v_2 + \ldots + \overline{u_n}\cdot v_n. \tag{6}$$

The inner product satisfies the conditions:

1. $\langle v|v\rangle \geq 0$, with $\langle v|v\rangle = 0$ if and only if $|v\rangle = \mathbf{0}$.

2. $\langle u|v\rangle = \overline{\langle v|u\rangle}$ for all $|u\rangle$, $|v\rangle$ in the vector space.

3. $\langle u|\alpha v + \beta w\rangle = \alpha \langle u|v\rangle + \beta \langle u|w\rangle$,
   $\langle \alpha u + \beta v|w\rangle = \alpha^* \langle u|w\rangle + \beta^* \langle v|w\rangle$ where α and β are scalars and α^* and β^* are their complex conjugates.

If the inner product between two vectors is zero, $\langle u|v\rangle = 0$, then we say that $|u\rangle$ and $|v\rangle$ are orthogonal to one another. The inner product may be used to define a norm on the vector space by computing the inner product of a vector with itself such that the norm of a vector $|v\rangle$ is defined by

$$\||v\rangle\| = \sqrt{\langle v|v\rangle}. \tag{7}$$

When the norm of a vector is unity then the vector is said to be normalised, so when $\langle v|v\rangle = 1$ the vector $|v\rangle$ is normalised. A vector can be normalised by computing the norm and dividing the vector by it. If each vector in a set of vectors is normalised and the vectors are orthogonal to one another then the set is orthonormal.

2.1 Qubits and quantum states

The basic unit of information in a classical computer is the bit which takes either the value 0 or 1. In an analogous manner the basic unit of information in quantum computing is the quantum bit or qubit. A qubit, like a bit, can be in one of two states. A qubit is represented as two-dimensional state space in $\mathbb{C}^2$ with orthonormal basis vectors $|0\rangle$ and $|1\rangle$. The vectors in the basis set $\{|0\rangle, |1\rangle\}$ are defined by

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{8}$$

This is the most common basis used in quantum computing and is called the computational basis. A qubit can exist in the state $|0\rangle$ or the state $|1\rangle$ or a superposition state which is a linear combination of the two. The superposition state $|\psi\rangle$ is written as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{9}$$

and the Hermitian conjugate (dual vector) $\langle\psi| = (|\psi\rangle)^\dagger$ is given by

$$\langle\psi| = \alpha^*\,\langle 0| + \beta^*\,\langle 1|\,. \tag{10}$$

Here α is the complex scalar amplitude of measuring $|0\rangle$ and β the amplitude of measuring $|1\rangle$. These amplitudes are essentially 'quantum probabilities', they represent the chance that a given quantum state will be observed when the superposition is collapsed. Unlike traditional probabilities, described by real numbers, these amplitudes are represented by complex numbers. The probability of finding $|\psi\rangle$ in state $|0\rangle$ is $|\alpha|^2$ and in state $|1\rangle$ is $|\beta|^2$.

In order to simplify reasoning about quantum computational systems it is assumed that $|\psi\rangle$ is a normalised unit vector. This means that $\||\psi\rangle\| = \langle\psi|\psi\rangle = 1$. Since both $|0\rangle$ and $|1\rangle$ are orthonormal it follows that

$$\begin{aligned} 1 =& \langle\psi|\psi\rangle \\ =& (\alpha^* \langle 0| + \beta^* \langle 1|) \cdot (\alpha |0\rangle + \beta |1\rangle) \\ =& \alpha^*\alpha \langle 0|0\rangle + \alpha^*\beta \langle 0|1\rangle + \beta^*\alpha \langle 1|0\rangle + \beta^*\beta \langle 1|1\rangle \\ =& |\alpha|^2 + |\beta|^2, \end{aligned} \tag{11}$$

as $\langle 0|0\rangle = \langle 1|1\rangle = 1$ and $\langle 0|1\rangle = \langle 1|0\rangle = 0$. In order to work with qubits to produce meaningful results it is necessary to understand the notion of operators.

2.2 Matrices and operators

An operator $\hat{A}$ is a mathematical rule that transforms a ket $|\psi\rangle$ into another ket $|\phi\rangle$ such that

$$\hat{A} |\psi\rangle = |\phi\rangle . \tag{12}$$

Operators can also act on bras with the result being another bra i.e. $\langle\mu| \hat{A} = \langle\nu|$. An operator is said to be linear if

$$\hat{A} (\alpha |\psi_1\rangle + \beta |\psi_2\rangle) = \alpha\hat{A} |\psi_1\rangle + \beta\hat{A} |\psi_2\rangle , \tag{13}$$

where α and β are complex numbers and $|\psi_1\rangle$ and $|\psi_2\rangle$ are state vectors. The simplest operator is the identity operator $\hat{I}$ which leaves a state unchanged such that $\hat{I} |\psi\rangle = |\psi\rangle$.

In quantum theory observables are dynamical variables such as position, momentum, angular momentum or energy. Observables are measured quantities which characterise the quantum state of a particle. An important postulate of quantum theory is that each physical observable has a corresponding operator.

A set of operators that are of fundamental importance to quantum computation are the Pauli operators. There are four Pauli operators, including the identity operator, denoted either by $\{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$, $\{\sigma_0, \sigma_x, \sigma_y, \sigma_z\}$ or $\{I, X, Y, Z\}$.

The identity operator $\sigma_0 = I$ acts on the computational basis state as previously described:

$$\sigma_0 |0\rangle = |0\rangle , \quad \sigma_0 |1\rangle = |1\rangle . \tag{14}$$

The next Pauli operator, denoted by $\sigma_1 = \sigma_x = X$, acts as follows:

$$\sigma_1 |0\rangle = |1\rangle , \quad \sigma_1 |1\rangle = |0\rangle , \tag{15}$$

this operator is often referred to as the quantum NOT operator. Then the operator $\sigma_2 = \sigma_y = Y$ acts on the computational basis in the following way:

$$\sigma_2 |0\rangle = i |1\rangle , \quad \sigma_2 |1\rangle = -i |0\rangle . \tag{16}$$

Finally the $\sigma_3 = \sigma_z = Z$ operators acts as:

$$\sigma_3 |0\rangle = |0\rangle , \quad \sigma_3 |1\rangle = -|1\rangle . \tag{17}$$

The outer product of two vectors, denoted $|v\rangle \langle u|$ or $|v\rangle \otimes \langle u|$, is the tensor product of $|v\rangle$ with the conjugate transpose of $|u\rangle$. Unlike the inner product which produces a scalar, the outer product is an operator which may be represented as a matrix:

$$|v\rangle \langle u| = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \begin{bmatrix} \overline{u_1} & \overline{u_2} & \dots & \overline{u_m} \end{bmatrix} = \begin{bmatrix} v_1\overline{u_1} & v_1\overline{u_2} & \dots & v_1\overline{u_m} \\ v_2\overline{u_1} & v_2\overline{u_2} & \dots & v_2\overline{u_m} \\ \vdots & \vdots & \ddots & \vdots \\ v_n\overline{u_1} & v_n\overline{u_2} & \dots & v_n\overline{u_m} \end{bmatrix} . \tag{18}$$

The outer product is a linear transformation between vector spaces. It is often more convenient not to write the outer product in matrix form, instead the transformation from a Hilbert space U to another Hilbert space V on a vector $|w\rangle$ in U may be described by

$$(|v\rangle \langle u|) |w\rangle = |v\rangle \langle u|w\rangle = \langle u|w\rangle |v\rangle , \tag{19}$$

since $\langle u|w\rangle$ is a scalar value.

It is often convenient to be able to convert between an operator in Dirac notation to a matrix representation. A useful tool in this regard is the closure relation which states that given a basis set $\{|v_i\rangle\}$ in n dimensions, the identity operator may be written as

$$\sum_{i=1}^{n} |v_i\rangle \langle v_i| = \hat{I}. \tag{20}$$

So an arbitrary state $|\psi\rangle$ can be expanded in terms of the basis $\{|v_i\rangle\}$ in the following way:

$$|\psi\rangle = \hat{I} |\psi\rangle = \left(\sum_{i=1}^{n} |v_i\rangle \langle v_i| \right) |\psi\rangle = \sum_{i=1}^{n} |v_i\rangle \langle v_i|\psi\rangle = \sum_{i=1}^{n} \alpha_i |v_i\rangle , \tag{21}$$

where $\alpha_i = \langle v_i|\psi\rangle$ are constant inner products. Qubits exist in a two-dimensional Hilbert space so the identity operator may be written, using the computational basis, as

$$\hat{I} = |0\rangle \langle 0| + |1\rangle \langle 1| . \tag{22}$$

If we know the action of an operator $\hat{A}$ on a basis set $\{|v_i\rangle\}$ then the matrix elements of the operator may be found using the closure relation:

$$\hat{A} = \hat{I}\hat{A}\hat{I} = \left(\sum_{i} |v_i\rangle \langle v_i| \right) \hat{A} \left(\sum_{j} |v_j\rangle \langle v_j| \right) = \sum_{i,j} \langle v_i|\hat{A}|v_j\rangle |v_i\rangle \langle v_j| . \tag{23}$$

The term $\langle v_i|\hat{A}|v_j\rangle = \hat{A}_{ij}$ represents the matrix element of the operator $\hat{A}$ at row i and column j with respect to the basis $\{|v_i\rangle\}$. The operator $\hat{A}$ is written is matrix form as:

$$\hat{A} = \begin{pmatrix} \langle v_1|\hat{A}|v_1\rangle & \langle v_1|\hat{A}|v_2\rangle & \dots & \langle v_1|\hat{A}|v_n\rangle \\ \langle v_2|\hat{A}|v_1\rangle & \langle v_2|\hat{A}|v_2\rangle & \dots & \langle v_2|\hat{A}|v_n\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle v_n|\hat{A}|v_1\rangle & \langle v_n|\hat{A}|v_2\rangle & \dots & \langle v_n|\hat{A}|v_n\rangle \end{pmatrix}. \tag{24}$$

In this way the Pauli operators may be represented both in Dirac notation and in matrix form. The identity operator may be written, in Dirac notation, as $\sigma_0 = |0\rangle\langle 0| + |1\rangle\langle 1|$ or in matrix form as

$$\sigma_0 = \begin{pmatrix} \langle 0|\sigma_0|0\rangle & \langle 0|\sigma_0|1\rangle \\ \langle 1|\sigma_0|0\rangle & \langle 1|\sigma_0|1\rangle \end{pmatrix} = \begin{pmatrix} \langle 0|0\rangle & \langle 0|1\rangle \\ \langle 1|0\rangle & \langle 1|1\rangle \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{25}$$

Similarly the X Pauli operator may be written, in both Dirac and matrix form, as

$$\sigma_1 = |0\rangle\langle 1| + |1\rangle\langle 0| = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{26}$$

Then the Y Pauli operator may be written as

$$\sigma_2 = -i\,|0\rangle\langle 1| + i\,|1\rangle\langle 0| = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \tag{27}$$

Finally the Z Pauli operator may be written as

$$\sigma_3 = |0\rangle\langle 0| - |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{28}$$

The Pauli operators are very important, they are Hermitian along with many of the other operators required for quantum computation.

2.2.1 Hermitian, unitary and normal operators

The adjoint of an operator $\hat{A}$ is denoted by $\hat{A}^\dagger$ and is defined as follows:

$$\langle a|\hat{A}^\dagger|b\rangle = \langle b|\hat{A}|a\rangle^* \tag{29}$$

where * denotes the complex conjugate as usual. When computing the adjoint of an expression all constants are replaced with their complex conjugate, all kets by bras and bras by kets and all operators are replaced with their adjoint. For

the adjoint of a product of operators the order of the operators must be reversed. These rules are exemplified below:

$$\left(\alpha\hat{A}\right)^\dagger = \alpha^* \hat{A}^\dagger, \tag{30a}$$

$$\left(|\psi\rangle\right)^\dagger = \langle\psi|, \tag{30b}$$

$$\left(\langle\psi|\right)^\dagger = |\psi\rangle, \tag{30c}$$

$$\left(\hat{A}\hat{B}\right)^\dagger = \hat{B}^\dagger \hat{A}^\dagger, \tag{30d}$$

$$\left(\hat{A}\,|\psi\rangle\right)^\dagger = \langle\psi|\,\hat{A}^\dagger. \tag{30e}$$

$$\left(\hat{A} + \hat{B} + \hat{C}\right)^\dagger = \hat{A}^\dagger + \hat{B}^\dagger + \hat{C}^\dagger \tag{30f}$$

For an operator in outer product form $\hat{A} = |\psi\rangle\langle\phi|$ the rules (30b), (30c) and (30d) may be combined to give $\hat{A}^\dagger = |\phi\rangle\langle\psi|$.

An operator $\hat{A}$ is Hermitian if it is equal to its own adjoint i.e. $\hat{A} = \hat{A}^\dagger$. It is clear to see that the Pauli operators are all Hermitian for example the adjoint of the Pauli Y operator is given by

$$Y^\dagger = \left(-i\,|0\rangle\langle 1| + i\,|1\rangle\langle 0|\right)^\dagger = i\,|1\rangle\langle 0| - i\,|0\rangle\langle 1| = Y. \tag{31}$$

In quantum mechanics operators that represent physical observables are Hermitian.

An operator U is said to be unitary if its adjoint is equal to its inverse such that

$$UU^\dagger = U^\dagger U = I, \tag{32}$$

where I is the identity operator. The Pauli operators are both Hermitian and unitary. Unitary operators are useful for describing the time evolution of a quantum state. A unitary operator is also normal since an operator A is said to be normal if $AA^\dagger = A^\dagger A$.

2.2.2 Eigenvalues and eigenvectors

The usual definition of eigenvalues and eigenvectors from basic linear algebra translates to the Dirac notation used in quantum computation. A vector is said to be an eigenvector of an operator A if

$$A\,|\psi\rangle = \lambda\,|\psi\rangle, \tag{33}$$

where λ is a scalar called an eigenvalue of the operator A. The eigenvalues of an operator A are found by solving the characteristic equation $\det(A - \lambda I) = 0$.

The spectral decomposition theorem states when an operator in a vector space has a diagonal matrix representation with respect to some basis of that vector space. If an operator A satisfies the spectral decomposition theorem for some basis $|u_i\rangle$ then the operator may be written in the form

$$A = \sum_{i=1}^{n} \lambda_i |u_i\rangle \langle u_i| , \tag{34}$$

where λ_i are the eigenvalues of the operator. The Pauli $Z = |0\rangle \langle 0| - |1\rangle \langle 1|$ operator is written in this form since the Z operator is diagonal in the computational basis.

2.2.3 Functions of operators

Operators are useful for describing quantum systems therefore there are many functions of those operators which provided useful information about a given quantum system. The trace of an operator, when represented by a matrix, is the sum of the diagonal elements. When an operator is written as an outer product the trace is taken by summing over inner products with the basis vectors. For the basis vectors $|u_i\rangle$ the trace of an operator A is defined as

$$\mathrm{tr}(A) = \sum_{i=1}^{n} \langle u_i|A|u_i\rangle . \tag{35}$$

The trace has a number of useful properties such as being cyclic, $\mathrm{tr}(ABC) = \mathrm{tr}(CAB) = \mathrm{tr}(BCA)$. The trace of an outer product is given by an inner product, $\mathrm{tr}(|\phi\rangle \langle \psi|) = \langle \psi|\phi\rangle$. The trace is also basis independent and is equal to the sum of the eigenvalues of the operator.

Another useful operator function is the expectation value of an operator which gives the mean value of that operator with respect to a specified quantum state. For a quantum state $|\psi\rangle$ which is measured many times, using the operator A, the average of the measurements is given by the expectation value

$$\langle A\rangle = \langle \psi|A|\psi\rangle . \tag{36}$$

The expectation value may be calculated for repeated applications of an operator such as $\langle A^2\rangle$. The standard deviation or uncertainty of an operator is defined as

$$\Delta A = \sqrt{\langle A^2\rangle - \langle A\rangle^2}. \tag{37}$$

A general function of an operator may be determined by calculating its Taylor expansion

$$f(A) = \sum_{n=0}^{\infty} a_n A^n, \tag{38}$$

where a_n are determined coefficients. For example an operator in the argument of an exponential function may be written as

$$e^A = I + A + \frac{1}{2!}A + \ldots + \frac{a^n}{n!}A^n + \ldots . \tag{39}$$

If an operator A has a spectral expansion given by (34) then

$$f(A) = \sum_{i=1}^{n} f(\lambda_i) \left|u_i\right\rangle \left\langle u_i\right| . \tag{40}$$

If H is a Hermitian operator then

$$U = e^{i\epsilon H}, \tag{41}$$

where ϵ is a scalar and U is a unitary operator. If the Hermitian operator $H = \sum_i \phi_i \left|u_i\right\rangle \left\langle u_i\right|$ then we can write

$$U = e^{i\epsilon H} = \sum_i e^{i\epsilon\phi_i} \left|u_i\right\rangle \left\langle u_i\right| . \tag{42}$$

By taking the Taylor expansion of (41), retaining only the first two terms, the infinitesimal unitary transformation

$$U = I + i\epsilon H \tag{43}$$

is obtained. The operator H is the 'generator' of the this transformation.

2.2.4 Unitary transformations

The matrix representation of an operator can be transformed from one basis to another by means of a unitary transformation. In the two-dimensional vector space $\mathbb{C}^2$ the change of basis matrix, from a basis $|u_i\rangle$ to a basis $|v_i\rangle$, is given by

$$U = \begin{pmatrix} \langle v_1|u_1\rangle & \langle v_1|u_2\rangle \\ \langle v_2|u_1\rangle & \langle v_2|u_2\rangle \end{pmatrix} . \tag{44}$$

A state vector $|\psi\rangle$ in the basis $|u_i\rangle$ may be given in terms of the basis $|v_i\rangle$ as

$$\left|\psi'\right\rangle = U \left|\psi\right\rangle . \tag{45}$$

An operator A in the basis $|u_i\rangle$ is written in terms of the $|v_i\rangle$ basis as

$$A' = UAU^\dagger . \tag{46}$$

2.2.5 Projection operators

An operator formed by writing the outer product using a single ket is called a projection operator. Given a state $|\psi\rangle$ there is a corresponding projection operator

$$P = |\psi\rangle \langle\psi| , \tag{47}$$

which is Hermitian. If the state $|\psi\rangle$ is normalised then $P = P^2$. If the projection operators P_1 and P_2 commute, $P_1P_2 = P_2P_1$, then the product P_1P_2 is also a projection operator. An operator A may be written in terms of projection operators using the spectral decomposition theorem (34). The projection operator $P_i = |u_i\rangle \langle u_i|$ projects onto the subspace defined by the eigenvalue λ_i. A given measurement for the operator A is represented by the eigenvalue λ_i. Projection operators represent a measurement described by quantum theory. The spectral decomposition theorem may be rewritten in terms of projection operators as

$$A = \sum_{i=1}^{n} \lambda_i P_i. \tag{48}$$

Suppose that a system is in the state

$$|\psi\rangle = \sum_{i=1}^{n} c_i |u_i\rangle , \tag{49}$$

where the vectors $|u_i\rangle$ form an orthonormal basis. The probability of finding the ith outcome when a measurement is made on the system is given by

$$Pr(i) = |P_i |\psi\rangle|^2 = \langle\psi|P_i^\dagger P_i|\psi\rangle = \langle\psi|P_i^2|\psi\rangle = \langle\psi|P_i|\psi\rangle , \tag{50}$$

since the projection operator is Hermitian, $P_i^\dagger = P_i$, and $P_i^2 = P_i$. So it follows that the probability of the ith outcome is given by

$$\begin{aligned} Pr(i) = \langle\psi|P_i|\psi\rangle &= \left(\sum_{j=1}^{n} c_j^* \langle u_j| \right) (|u_i\rangle \langle u_i|) \left(\sum_{k=1}^{n} c_k |u_k\rangle \right) \\ &= \sum_{j=1}^{n} c_j^* \langle u_j|u_i\rangle \sum_{k=1}^{n} c_k \langle u_i|u_k\rangle = \sum_{j=1}^{n} c_j^* \delta_{ij} \sum_{k=1}^{n} c_k \delta_{ik} \\ &= c_i^* c_i = |c_i|^2. \end{aligned} \tag{51}$$

2.2.6 Commutator algebra

The commutator of two operators A and B is defined as

$$[A, B] = AB - BA. \tag{52}$$

The operators A and B are said to commute when $[A, B] = 0$. When operators commute the order of the operators is interchangeable. In general $[A, B] \neq 0$ so the order of operators is important. The commutator is antisymmetric and linear such that

$$[A, B] = -[B, A] \tag{53a}$$

and

$$[A, B + C] = [A, B] + [A, C]. \tag{53b}$$

The commutator also follows the distributive rule

$$[A, BC] = [A, B]C + B[A, C]. \tag{54}$$

An operator A is normal if $\left[A, A^\dagger\right] = 0$. If two operators commute then they possess a set of common eigenvectors. The anti-commutator of two operators A and B is defined as

$$\{A, B\} = AB + BA. \tag{55}$$

For two operators A and B the product of their uncertainties, defined by (37), satisfies

$$\Delta A \Delta B \geq \frac{1}{2} |\langle [A, B] \rangle|. \tag{56}$$

This states that there is a limit to the precision with which the values of two incompatible observables can be known simultaneously. The inequality (56) sets a lower bound on the precision that can be obtained from measurements. If $[A, B] \neq 0$, as the uncertainty in A is reduced the uncertainty in B must increase. This is a generalisation of the famous Heisenberg uncertainty principle.

2.2.7 Polar decomposition

For an operator A, with a nonsingular matrix representation, the operator can be decomposed into a unitary operator U and a positive semidefinite Hermitian operator P such that $A = UP$ (left polar decomposition). The operator P is given by

$$P = \sqrt{A^\dagger A} \tag{57}$$

and the operator U is determined by

$$U = AP^{-1} = A\left(\sqrt{A^\dagger A}\right)^{-1}. \tag{58}$$

There also exists the right polar decomposition $A = QU$ where $Q = \sqrt{AA^\dagger}$.

2.2.8 Quantum states and observables

The qubits and operators, describing quantum states and observables respectively, can be combined into a framework describing the physical theory of quantum mechanics. The time varying state of a quantum system is a vector $|\psi(t)\rangle$ in a Hilbert space, which contains all the information about the system. The state vectors are normalised such that $\langle\psi|\psi\rangle = 1$. A qubit is a state vector in a complex two-dimensional vector space $\mathbb{C}^2$ defined by $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ normalised such that $|\alpha|^2 + |\beta|^2 = 1$.

There is a corresponding operator A for every dynamical variable A that is a physically measurable quantity. The eigenvectors of A form a complete orthonormal basis of the vector space and the operator A is Hermitian. When measuring a dynamical variable A the possible results of the measurement are the eigenvalues λ_n of the operator A corresponding to that variable. The operator A can be written in terms of it's eigenvalues and projection operators $P_n = |u_n\rangle\langle u_n|$ as $A\sum_n \lambda_n P_n$ using the spectral decomposition theorem. The probability of measuring the result λ_n is given by

$$Pr(\lambda_n) = \langle\psi|P_n|\psi\rangle = \mathrm{tr}(P_n\,|\psi\rangle\langle\psi|). \tag{59}$$

The amplitude $c_n = \langle u_n|\psi\rangle$ gives the probability of obtaining the result λ_n as

$$Pr(\lambda_n) = \frac{|\,\langle u_n|\psi\rangle\,|^2}{\langle\psi|\psi\rangle} = \frac{|c_n|^2}{\langle\psi|\psi\rangle} = |c_n|^2, \tag{60}$$

since the state is normalised. Taking a measurement causes a collapse of the wavefunction, this means that obtaining a measurement result λ_n leaves the system in the state $|u_n\rangle$. The state of the system postmeasurement may be written in terms of the projection operator P_n as

$$|\psi\rangle \to \frac{P_n\,|\psi\rangle}{\sqrt{\langle\psi|P_n|\psi\rangle}}. \tag{61}$$

The Schrödinger equation governs the time evolution of a physically isolated quantum system. In Dirac notation the Schrödinger equation is given by

$$i\hbar\frac{\partial}{\partial t}\,|\psi\rangle = H\,|\psi\rangle\,, \tag{62}$$

where H is the Hamiltonian operator of the system and $\hbar$ is the reduced Planck constant. The eigenvalues of the Hamiltonian operator determine the possible energies the system can have. At a later time t the state of the system is given by

$$|\psi(t)\rangle = e^{-iHt/\hbar}\,|\psi(0)\rangle = U\,|\psi(0)\rangle\,, \tag{63}$$

where $U = e^{-iHt/\hbar}$ is a unitary operator. The above describes the time evolution of a single particle quantum state however it often necessary to consider multi-particle states.

2.3 Tensor products

It is difficult to do any interesting computation with a single qubit. Quantum computers tend to work with registers made up of multiple qubits. In order to understand multiparticle quantum systems mathematically it is necessary to be able to create a composite Hilbert space made up of the Hilbert spaces that are associated with each particle. This composition is done using the tensor product.

The tensor product is a way of combining vector spaces into larger vector spaces. Consider the two particle case where H_1 and H_2 are two Hilbert spaces of dimension N_1 and N_2. Combining these two Hilbert spaces, using the tensor product, forms the larger Hilbert space

$$H = H_1 \otimes H_2. \tag{64}$$

The dimension of this Hilbert space is the product of the dimensions of H_1 and H_2 such that

$$\dim(H) = N_1 N_2. \tag{65}$$

Suppose that $|\phi\rangle \in H_1$ and $|\xi\rangle \in H_2$ are two vectors that belong to the Hilbert spaces which are composed to construct H. A vector $|\psi\rangle \in H$ can be constructed, using the tensor product, as

$$|\psi\rangle = |\phi\rangle \otimes |\xi\rangle = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_m \end{bmatrix} \otimes \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix} = \begin{bmatrix} \phi_1 \cdot \xi_1 \\ \phi_1 \cdot \xi_2 \\ \vdots \\ \phi_1 \cdot \xi_n \\ \phi_2 \cdot \xi_1 \\ \vdots \\ \phi_{m-1} \cdot \xi_n \\ \phi_m \cdot \xi_1 \\ \vdots \\ \phi_m \cdot \xi_n \end{bmatrix}. \tag{66}$$

The tensor product of two vectors $|\phi\rangle \otimes |\xi\rangle$ is a very common operation so it is often written more simply as $|\phi\rangle\,|\xi\rangle$ or $|\phi\xi\rangle$. The tensor product of two vectors is linear such that

$$|\phi\rangle \otimes (|\xi_1\rangle + |\xi_2\rangle) = |\phi\rangle \otimes |\xi_1\rangle + |\phi\rangle \otimes |\xi_2\rangle\,, \tag{67a}$$

$$(|\phi_1\rangle + |\phi_2\rangle) \otimes |\xi\rangle = |\phi_1\rangle \otimes |\xi\rangle + |\phi_2\rangle \otimes |\xi\rangle \tag{67b}$$

and

$$|\phi\rangle \otimes (\alpha\,|\xi\rangle) = \alpha\,|\phi\rangle \otimes |\xi\rangle\,. \tag{67c}$$

A basis for the larger Hilbert space H is constructed by forming the tensor product of the basis vectors of the spaces H_1 and H_2. If the basis vectors of H_1 and H_2 are denoted by $|u_i\rangle$ and $|v_i\rangle$ respectively then a basis for $H = H_1 \otimes H_2$ can be constructed by forming the tensor product

$$|w_i\rangle = |u_i\rangle \otimes |v_i\rangle . \tag{68}$$

Also a vector $|\phi\rangle$ undergoing the tensor product with itself n times is denoted by $|\phi\rangle^{\otimes n}$. A simple example is the composition of two single qubits to form a two qubit systems. If H_1 and H_2 are two Hilbert spaces for single qubits, with each having the basis $\{|0\rangle , |1\rangle\}$, then the basis for $H = H_1 \otimes H_2$ is given by the composition of each basis vector with each other i.e. $\{|0\rangle \otimes |0\rangle , |0\rangle \otimes |1\rangle , |1\rangle \otimes |0\rangle , |1\rangle \otimes |1\rangle\}$.

Let $|\phi\rangle$ and $|\xi\rangle$ be vectors from H_1 and H_2 such that

$$|\phi\rangle = \sum_i \alpha_i |u_i\rangle \quad \text{and} \quad |\xi\rangle = \sum_i \beta_i |v_i\rangle ,$$

where $|u_i\rangle$ is the basis of H_1 and $|v_i\rangle$ is the basis of H_2. The vector $|\psi\rangle = |\phi\rangle \otimes |\xi\rangle$, belonging to the larger Hilbert space H, may be written as the products of the individual terms such that

$$|\psi\rangle = \sum_{i,j} \alpha_i \beta_j |u_i\rangle \otimes |v_j\rangle . \tag{69}$$

2.3.1 Inner products

It is simple to compute the inner product of two vectors belonging to the larger Hilbert space H by taking the inner products of the vectors belonging to H_1 and H_2. Suppose that $|\psi_1\rangle , |\psi_2\rangle \in H$ such that

$$\begin{aligned} |\psi_1\rangle &= |\phi_1\rangle \otimes |\xi_1\rangle , \\ |\psi_2\rangle &= |\phi_2\rangle \otimes |\xi_2\rangle . \end{aligned}$$

Then the inner product of the two vectors is given by

$$\langle\psi_1|\psi_2\rangle = (\langle\phi_1| \otimes \langle\xi_1|)(|\phi_2\rangle \otimes |\xi_2\rangle) = \langle\phi_1|\phi_2\rangle \langle\xi_1|\xi_2\rangle . \tag{70}$$

2.3.2 Operators and tensor products

Let A be an operator that acts on vectors $|\phi\rangle \in H_1$ and let B be an operator that acts on the vectors $|\xi\rangle \in H_2$ where H_1 and H_2 are the Hilbert spaces used to construct the larger Hilbert space $H = H_1 \otimes H_2$. An operator $A \otimes B$ may be created that acts on the vectors $|\psi\rangle \in H$ as follows:

$$(A \otimes B)|\psi\rangle = (A \otimes B)(|\phi\rangle \otimes |\xi\rangle) = (A|\phi\rangle) \otimes (B|\xi\rangle) . \tag{71}$$

For example, given that $\sigma_1 |0\rangle = |1\rangle$ and $\sigma_3 |1\rangle = -|1\rangle$ and if $|\psi\rangle = |0\rangle \otimes |1\rangle$ then

$$\begin{aligned}\sigma_1 \otimes \sigma_3 |\psi\rangle &= (\sigma_1 \otimes \sigma_3)(|0\rangle \otimes |1\rangle) \\ &= (\sigma_1 |0\rangle) \otimes (\sigma_3 |1\rangle) \\ &= -|1\rangle \otimes |1\rangle .\end{aligned}$$

If both A and B are Hermitian then the operator $A \otimes B$ is also Hermitian. Similarly if A and B are both projection operators or unitary operators then $A \otimes B$ is a projection operator or a unitary operator respectively.

The computation of the tensor (Kronecker) product of two matrices is frequently required in quantum computing as matrices represent operators. If an operator A is represented by a $m \times n$ matrix and an operator B is represented by a $p \times q$ matrix then the operator $A \otimes B$ is represented by an $mp \times nq$ matrix such that

$$\begin{aligned}A \otimes B &= \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix} \\ &= \begin{pmatrix} a_{11}b_{11} & \dots & a_{11}b_{1q} & \dots & a_{1n}b_{11} & \dots & a_{1n}b_{1q} \\ \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & \dots & a_{11}b_{pq} & \dots & a_{1n}b_{p1} & \dots & a_{1n}b_{pq} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{m1}b_{11} & \dots & a_{m1}b_{1q} & \dots & a_{mn}b_{11} & \dots & a_{mn}b_{1q} \\ \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & \dots & a_{m1}b_{pq} & \dots & a_{mn}b_{p1} & \dots & a_{mn}b_{pq} \end{pmatrix} .\end{aligned} \tag{72}$$

For example the tensor products of operators acting on the two-dimensional Hilbert space $\mathbb{C}^2$ produce an operator that acts on the four-dimensional Hilbert space $\mathbb{C}^4$. So if

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

are the matrix representation of two operators then

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{pmatrix} . \tag{73}$$

It is worth noting that $\mathrm{tr}(A \otimes B) = \mathrm{tr}(A)\,\mathrm{tr}(B)$. Understanding these mathematical concepts allows for the construction of algorithms which can be carried out by quantum computers, as shown in the next section.

3 Quantum Algorithms

Quantum computers, like classical computers, use registers made up of multiple qubits. When measured, so the system is in a collapsed state, quantum registers are strings of bits just like in classical computers. When each qubit in the register is in a superposition however the register of n qubits is in a superposition of all 2^n possible bit strings that could be represented using n bits. For a quantum register of size n the state space is a linear combination of n basis vectors, each of length 2^n such that

$$|\psi_n\rangle = \sum_{i=1}^{2^n} \alpha_i |i\rangle , \tag{74}$$

where i is the base-10 representation of a length n number in base-2. A register containing three qubits would be given by the state vector

$$\begin{aligned} |\psi_3\rangle =& \alpha_1 |000\rangle + \alpha_2 |001\rangle + \alpha_3 |010\rangle + \alpha_4 |011\rangle \\ &+ \alpha_5 |100\rangle + \alpha_6 |101\rangle + \alpha_7 |110\rangle + \alpha_8 |111\rangle . \end{aligned}$$

In vector form, using the computational basis, this would be

$$|\psi_3\rangle = \alpha_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_3 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_5 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_6 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \alpha_7 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \alpha_8 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Each possible bit configuration is denoted by the tensor product of each qubit i.e.

$$\begin{aligned} |110\rangle &= |1\rangle \otimes |1\rangle \otimes |0\rangle \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T . \end{aligned} \tag{75}$$

In a similar manner to single qubits, the probability of observing a given bit string upon collapsing the register is the square of the absolute value of the amplitude associated with a given bit string i.e. $Pr(i) = |\alpha_i|^2$. Also the squares of the absolute values of the amplitudes of all 2^n possible bit configurations sum to unity such that

$$\sum_{i=1}^{2^n} |\alpha_i|^2 = 1. \tag{76}$$

3.1 Quantum logic gates

In a classical computer it is possible to manipulate information either by moving it around or by doing some sort of processing using logic gates. Logic gates can be connected together to form complex circuits. The idea of logic gates and circuits has an equivalent notion in quantum computing. In a quantum computer information is processed using gates which are represented abstractly by unitary operations. Unlike classical logic gates a quantum logic gate must be a reversible operation where the inputs to the gate may be determined uniquely from the outputs of the gate.

3.1.1 Single qubit gates

Operators can be represented by matrices therefore a quantum gate with n inputs and outputs can be represented by a matrix of degree 2^n. For a single qubit a matrix of degree $2^1 = 2$ is required, so a quantum gate acting on a single qubit will be a 2×2 unitary matrix. Any unitary transformation is a valid elementary operation on a quantum computer however there are a small number of frequently used logic gates which are important in quantum computing.

The three Pauli gates discussed earlier are important single-qubit gates for quantum computation. The Pauli-X gate, also known as the quantum NOT gate, swaps the amplitudes of $|0\rangle$ and $|1\rangle$:

$$\boxed{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |1\rangle\langle 0| + |0\rangle\langle 1|. \tag{77}$$

The action of a NOT gate on an arbitrary state $|j\rangle$ can be written using the XOR operation as $X|j\rangle = |j \oplus 1\rangle$ since the XOR operation returns 1 if one of the inputs is 1 but 0 otherwise. The Pauli-Y matrix is also valid single-qubit gate since it is defined by a 2×2 unitary matrix:

$$\boxed{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = i|1\rangle\langle 0| - i|0\rangle\langle 1|. \tag{78}$$

The Pauli-Z gate, sometimes called the phase flip gate, is defined by

$$\boxed{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|. \tag{79}$$

The action of the Pauli-Z gate can be represented generally by $Z|j\rangle = (-1)^j|j\rangle$.

The Pauli-Z gate, which alters the phase of the system only, is a special case of the more general phase shift gate given by

$$\boxed{R_\theta} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} = |0\rangle\langle 0| + e^{i\theta}|1\rangle\langle 1|, \tag{80}$$

where θ is the phase shift. The phase shift gate alters the relative phase of the amplitudes of a qubit. This gate changes the amplitude of $|1\rangle$ by a factor of $e^{i\theta}$ but leaves the amplitude of $|0\rangle$ unchanged.

Another special case of the general phase shift gate, when $\theta = \pi/2$, is the phase or S gate, which is given by

$$\boxed{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = |0\rangle\langle 0| + i\,|1\rangle\langle 1|\,. \tag{81}$$

Similarly when $\theta = \pi/4$ we have the T or $\pi/8$ gate

$$\begin{aligned} \boxed{T} = &\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} = |0\rangle\langle 0| + e^{i\pi/4}\,|1\rangle\langle 1| \\ = &e^{i\pi/8}\begin{bmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{bmatrix} = e^{i\pi/8}\left(e^{-i\pi/8}\,|0\rangle\langle 0| + e^{i\pi/8}\,|1\rangle\langle 1|\right). \end{aligned} \tag{82}$$

Another important gate is the single-qubit Hadamard operator given by

$$\boxed{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}}\left(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|\right). \tag{83}$$

When applied to a qubit with the value $|0\rangle$ or $|1\rangle$ the Hadamard operator induces an equal superposition of the states $|0\rangle$ and $|1\rangle$ so there is an equal probability of the qubit being in the state $|0\rangle$ or $|1\rangle$ when observed. Although the probabilities are the same when applying the Hadamard operator to $|0\rangle$ or $|1\rangle$ the amplitudes are different. This can be seen by applying the Hadamard operator to each basis state:

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

The probability of finding the qubit in the states $|0\rangle$ or $|1\rangle$ is $1/2$ in both cases but the amplitudes are not the same. Applying the Hadamard operator on these states again will return the qubit to its original state. The Hadamard operator is used on each qubit in a register at the start of many quantum algorithms to give each of the 2^n possible configurations the same probability of 2^{-n} of being observed when the system is measured.

3.1.2 The Bloch sphere

Unitary transformations performed on a single qubit may be visualized as rotations and reflections about the x, y, and z axes of the Bloch sphere shown in figure 1.

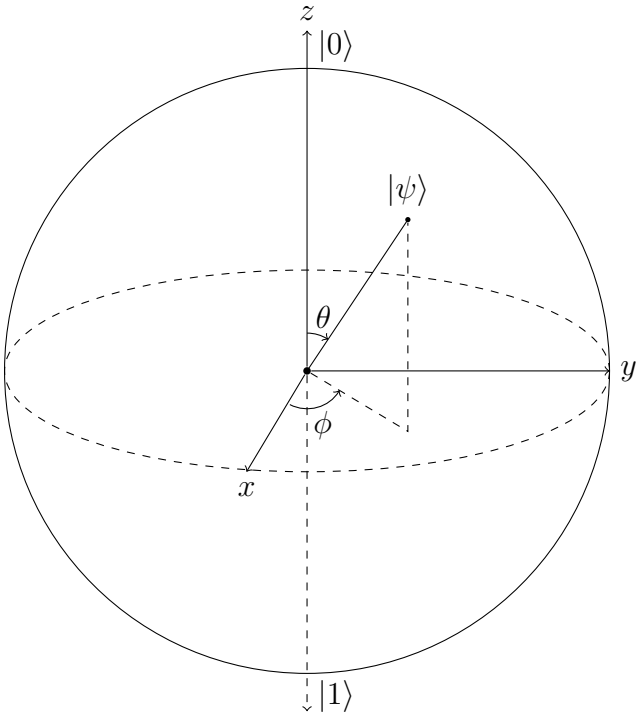

Figure 1: The Bloch sphere representation of a qubit

All the possible states $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ in $\mathbb{C}^2$ correspond to the points (θ, ϕ) on the surface of the unit sphere where

$$\begin{aligned} |\psi\rangle =& \alpha\,|0\rangle + \beta\,|1\rangle \\ =& \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\,. \end{aligned} \tag{84}$$

The Bloch sphere is an intuitive visual framework for understanding the effects of unitary transformations on quantum states.

More single-qubit gates can be constructed using exponentiation. If U is a unitary, Hermitian matrix, then

$$\exp\left(-i\theta U\right) = \cos\theta I - i\sin\theta U. \tag{85}$$

For example by exponentiating the Pauli matrices rotation operators can be created to represent rotation about the x, y and z axes on the Bloch sphere. These

rotation matrices are given by

$$R_x(\gamma) = e^{-i\gamma X/2} = \begin{pmatrix} \cos\left(\frac{\gamma}{2}\right) & -i\sin\left(\frac{\gamma}{2}\right) \\ -i\sin\left(\frac{\gamma}{2}\right) & \cos\left(\frac{\gamma}{2}\right) \end{pmatrix}, \tag{86a}$$

$$R_y(\gamma) = e^{-i\gamma Y/2} = \begin{pmatrix} \cos\left(\frac{\gamma}{2}\right) & -\sin\left(\frac{\gamma}{2}\right) \\ \sin\left(\frac{\gamma}{2}\right) & \cos\left(\frac{\gamma}{2}\right) \end{pmatrix}, \tag{86b}$$

$$R_z(\gamma) = e^{-i\gamma Z/2} = \begin{pmatrix} e^{-i\gamma/2} & 0 \\ 0 & e^{i\gamma/2} \end{pmatrix}. \tag{86c}$$

Given a single-qubit operator U, real numbers α, β, γ and δ can be found such that

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta), \tag{87}$$

this is the Z–Y decomposition.

3.1.3 Basic circuit diagrams

Circuit diagrams can be used to represent the action of a quantum gate on a qubit. Each gate is represented by a block with lines used to represent inputs and outputs. Figure 2 shows the circuit diagram representation for the three Pauli gates, the phase shift, S and T gates and the Hadamard gate. Figure 2 also shows the effect of each of the gates on the amplitudes of a single qubit.

Measurement in a circuit is represented by an encircled M, as shown in figure 3. Figure 3 indicates that the probability of each measurement result is given by the corresponding square amplitude. Other representations of a measurement may be different to this one but it is sufficient for our purposes at present.

3.1.4 Controlled gates

Controlled operations are often used in quantum computing, these operations use multiple qubits and change the state of a qubit based upon the values of other qubits. The controlled gate allows for an implementation of an if-else type construct with a quantum gate. Controlled unitary gates work by using a control qubit to determine whether or not a specified unitary action is applied to a target qubit.

For two-qubit controlled gates the operator representing the gate acts with respect to two-qubit states of the form $|a\rangle \otimes |b\rangle$ or more succinctly $|ab\rangle$. The matrix representation of an operator acting on a two-qubit state $|ab\rangle$ is calculated using

$$U = \begin{pmatrix} \langle 00|U|00\rangle & \langle 00|U|01\rangle & \langle 00|U|10\rangle & \langle 00|U|11\rangle \\ \langle 01|U|00\rangle & \langle 01|U|01\rangle & \langle 01|U|10\rangle & \langle 01|U|11\rangle \\ \langle 10|U|00\rangle & \langle 10|U|01\rangle & \langle 10|U|10\rangle & \langle 10|U|11\rangle \\ \langle 11|U|00\rangle & \langle 11|U|01\rangle & \langle 11|U|10\rangle & \langle 11|U|11\rangle \end{pmatrix}. \tag{88}$$

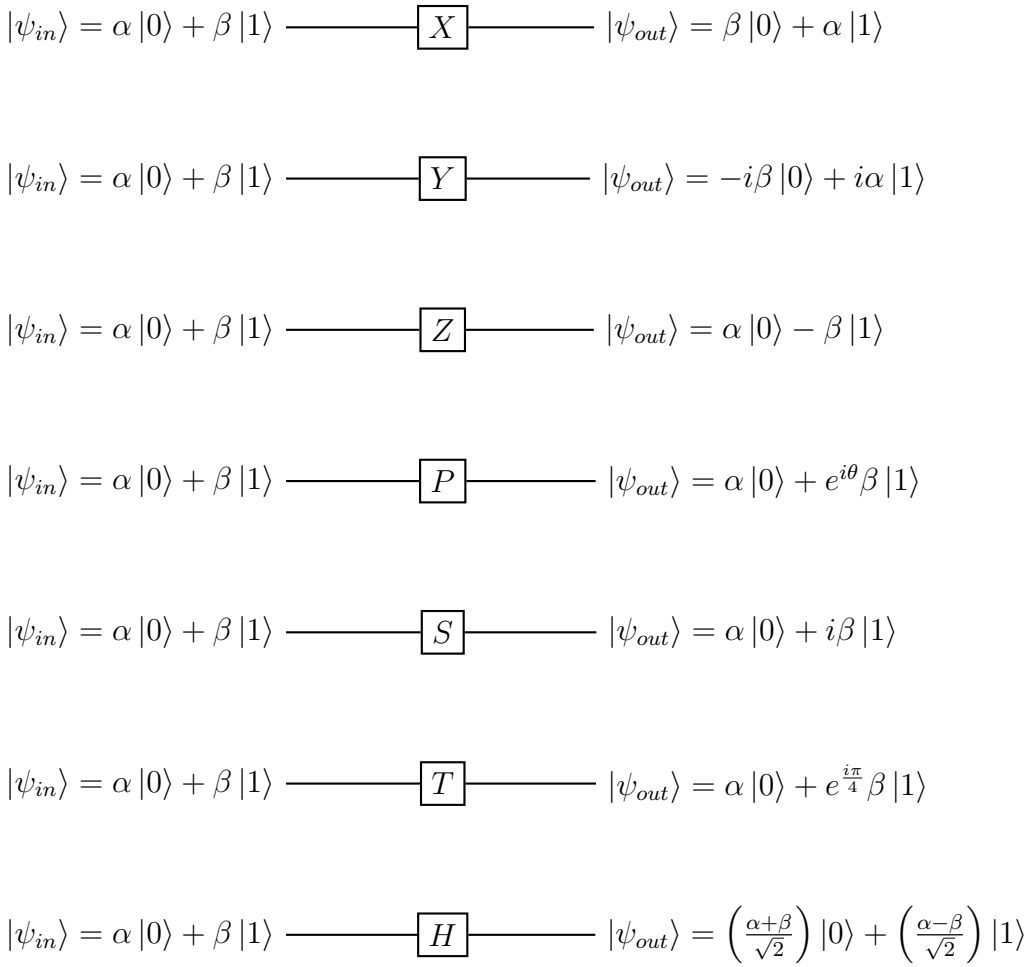

Figure 2: Circuit diagram representations of the Pauli, phase shift, S, T and Hadamard gates and their effect on a single general qubit.

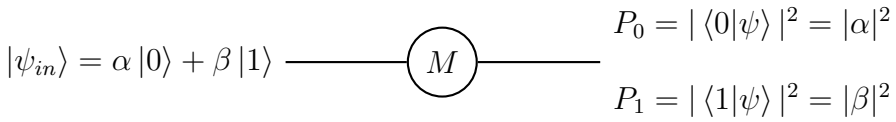

Figure 3: Representation of a measurement in a quantum circuit.

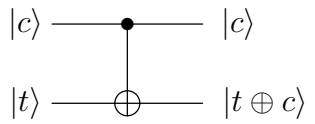

Figure 4: Circuit diagram representation of a controlled NOT gate.

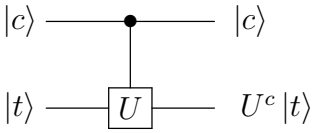

Figure 5: Circuit diagram representation of a general controlled gate.

The quantum controlled-NOT or CNOT gates swaps the amplitudes of the $|0\rangle$ and $|1\rangle$ basis states of a qubit only if the controlling qubit has the value $|1\rangle$. A CNOT gate can be described in terms of the XOR operation as

$$|c, t\rangle \to |c, t \oplus c\rangle\,. \tag{89}$$

If the control qubit c is $|0\rangle$ then nothing happens to the target qubit $|t\rangle$. If the control qubit is $|1\rangle$ then the NOT or Pauli-X matrix is applied to the target qubit. The action of the CNOT gate on the possible input states are

$$|00\rangle \mapsto |00\rangle\,, \tag{90a}$$
$$|01\rangle \mapsto |01\rangle\,, \tag{90b}$$
$$|10\rangle \mapsto |11\rangle\,, \tag{90c}$$
$$|11\rangle \mapsto |10\rangle\,. \tag{90d}$$

A circuit representation of the CNOT gate is shown in figure 4. The matrix representation of the controlled NOT gate is given by

$$C_{NOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{91}$$

Then in Dirac notation it is

$$C_{NOT} = |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 11| + |11\rangle\langle 10|\,. \tag{92}$$

It is not only the NOT gate that can be controlled in a quantum circuit; any unitary operation may be controlled in a similar manner. A circuit diagram of this general controlled unitary operator is shown in figure 5. If the control qubit is $|0\rangle$ then nothing happens to the target qubit ($U^0 = I$) however if the control qubit is $|1\rangle$ then the unitary operator U is applied to the target qubit. The gate U

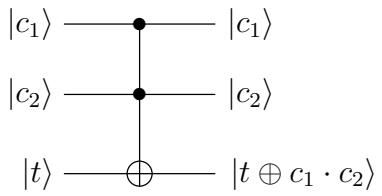

Figure 6: Circuit diagram representation of a Toffoli gate.

could be any unitary operator such as the Hadamard gate for example. A general controlled gate of this form has the matrix representation

$$C_U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{pmatrix}. \tag{93}$$

This can be written in Dirac notation as

$$C_U = |00\rangle\langle 00| + |01\rangle\langle 01| + U_{00}|10\rangle\langle 10| + U_{01}|10\rangle\langle 11| + U_{10}|11\rangle\langle 10| + U_{11}|11\rangle\langle 11|. \tag{94}$$

For example the controlled Hadamard gate may be written in matrix form as

$$C_H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \tag{95}$$

or in Dirac notation as

$$C_H = |00\rangle\langle 00| + |01\rangle\langle 01| + \frac{1}{\sqrt{2}}\left(|10\rangle\langle 10| + |10\rangle\langle 11| + |11\rangle\langle 10| - |11\rangle\langle 11|\right). \tag{96}$$

Controlled operations are possible with any number of control qubits and any unitary operator acting on any number of target qubits. One particularly interesting gate that may be implemented on a quantum computer is the Toffoli which has two control bits and one target bit with the unitary operator being the NOT or Pauli-X operator. A circuit diagram for the Toffoli gate can be seen in figure 6. If both control qubits are $|1\rangle$ then the NOT operator is applied to the target qubit otherwise nothing happens. The Toffoli gate is of particular interest because it can perform the classical computing operations AND, XOR, NOT and $FANOUT$ and so it can perform all the operations that a classical computer can perform. This shows that classical computing is a subset of quantum computing and that quantum computation is a least as powerful as classical computing. As the Toffoli

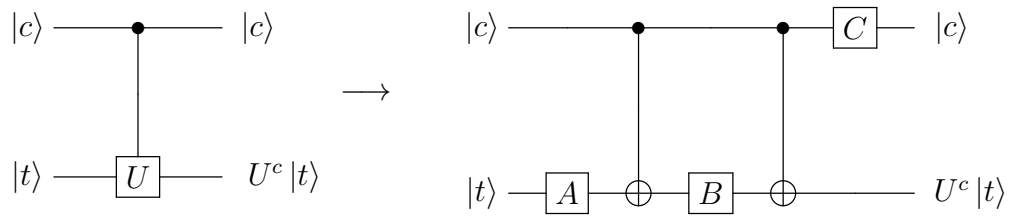

Figure 7: Decomposition of a controlled unitary operator into controlled NOT gates and the single qubit gates A, B and C.

gate acts upon three qubits it is represented by the 8×8 matrix

$$C_{\text{Toffoli}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{97}$$

3.1.5 Gate decomposition

In a quantum circuit an arbitrary controlled unitary operation U can be decomposed into a series of single-qubit operations and controlled NOT gates. This is shown schematically in figure 7 where a controlled unitary operation is transformed into two controlled NOT gates and three single-qubit gates A, B and C. This sort of decomposition is useful as it allows complex operators to be rewritten in terms of simple, single-qubit operators.

3.2 Basic quantum algorithms

Quantum logic gates can be combined in a multitude of different ways to produce a set of operations which perform some specified task. Some algorithms perform certain tasks much better on a quantum computer than on a classical computer. The properties of quantum computers, such as superposition and interference of qubits, allows computations to be performed in parallel in a way that would never be possible on a classical computer. Simultaneous parallel computations can occur in a quantum system because the system can exist in a linear combination of states. Parallel computations can be performed on a quantum computer using only a single circuit, for example a quantum algorithm is able to evaluate the function $f(x)$ at multiple values of x simultaneously.

This quantum parallelism due to superposition is lost as soon as a qubit is measured. Reading off n bits of data from a classical computer has no effect on the values of the bits however reading the contents of n qubits in a quantum computer collapses the superposition and the outcome of the measurement is not predictable. In a quantum algorithm it is assumed that the superposed states exist simultaneously even though it is only possible to actually measure one of those states. The amplitudes of each of the basis states indicate the probability of find the superposed states in a particular state when measured.

Quantum algorithms use superposition to their advantage whilst trying to avoid measurements which destroy the superposition before a result is required. The probability of reading out a useful result varies so it is incumbent upon the designer of the algorithm to try and change the state vector within the algorithm in order to significantly increase the likelihood of measuring something useful. Unitary transformations are able to influence all of the superposed states simultaneously without measuring the system and causing the superposition to collapse. Superposed quantum states and unitary transformations are the building blocks of quantum algorithms.

3.2.1 Hadamard transform

In many quantum algorithms the Hadamard transform is an important initial step as it creates a superposition of states each with equal probability of being found if the system were to be measured. Applying the Hadamard gate twice, in series, to an arbitrary qubit returns the qubit to its original state. Applying Hadamard gates in parallel however leads to very different behaviour. For example consider the action of two Hadamard gates, applied in parallel, to the two-qubit product state $|0\rangle|0\rangle = |00\rangle$, this gives the superposition of states

$$\begin{aligned}(H \otimes H)|0\rangle|0\rangle &= (H|0\rangle)(H|0\rangle) = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)\,. \end{aligned} \tag{98}$$

The parallel application of n Hadamard gates on n qubits is called a Hadamard transform. The application of the Hadamard transform $H^{\otimes n}$ to a register of n qubits, all initialised to $|0\rangle$, is given by

$$H^{\otimes n}\left(|0\rangle^{\otimes n}\right) = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle\,. \tag{99}$$

Here $x \in \{0,1\}^n$ means that x is one of the 2^n possible n qubit states i.e. if $x \in \{0,1\}^3$ then x is one of the three qubit states $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$,

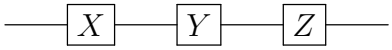

Figure 8: Circuit diagram representation of three gates applied in series.

$|101\rangle$, $|110\rangle$, $|111\rangle$. A quantum register if often initialised by setting all the qubits to $|0\rangle$ then applying the Hadamard transform.

The Hadamard gate, applied to an arbitrary qubit, gives an example of quantum interference. Applying the Hadamard gate to an arbitrary qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ gives

$$H|\psi\rangle = \left(\frac{\alpha+\beta}{\sqrt{2}}\right)|0\rangle + \left(\frac{\alpha-\beta}{\sqrt{2}}\right)|1\rangle . \tag{100}$$

The Hadamard gate exhibits both positive and negative interference in which amplitudes add constructively or destructively to increase or decrease the amplitudes after the operation. Quantum interference plays an important role in many quantum algorithms by allowing information to be obtained about a function $f(x)$ that depends on the function being evaluated at many values of x.

3.2.2 Matrix representation of operations

For examining quantum algorithms it is often useful to break the circuit down into matrix operations. If a set of operations is performed in series then the operation as a whole is represented by a matrix product. The matrix representation of a sequence of operations is given by multiplying the matrices in reverse order such that for the operations shown in figure 8 the operation is written as ZYX.

For operations that are performed in parallel the matrix representation is given by the tensor product of the individual matrices. For example the matrix representation of $X \otimes Y$ is

$$X \otimes Y = \begin{pmatrix} 0Y & 1Y \\ 1Y & 0Y \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \\ 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \end{pmatrix} . \tag{101}$$

3.2.3 Function evaluation (Deutsch's algorithm)

Quantum parallelism, which gives the ability to evaluate a function $f(x)$ at many values of x simultaneously, was exploited by Deutsch (1985) [2] to show how a particular problem may be solved faster on a quantum computer than on a classical computer. Consider a function that accepts a single bit as an input and produces a single bit as an output such that the function maps $\{0, 1\}$ into $\{0, 1\}$. Only four

such functions exist namely the constant functions $f(x) = 0$ and $f(x) = 1$, the identity function

$$f(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x = 1, \end{cases} \tag{102}$$

and the bitflip function

$$f(x) = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x = 1. \end{cases} \tag{103}$$

The first two functions are said to be constant since $f(0) = f(1)$ and the identity and bit flip functions, with $f(0) \neq f(1)$, are said to be balanced. Deutsch's algorithm uses quantum interference to determine whether or not the given function is constant or balanced i.e. the condition $f(0) = f(1)$. It is equivalent check $f(0) \oplus f(1)$ (which can be implemented as a controlled NOT gate) if $f(0) \oplus f(1) = 0$ then f is constant otherwise f is balanced.

Let us propose the unitary operation, acting on two qubits, U_f such that

$$U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle . \tag{104}$$

The operation U_f leaves the first qubit unchanged and produces the XOR of the second qubit with the function evaluated with the first qubit as the argument. Now since $|x\rangle$ is a qubit it can exist in a superposition state which can be done by applying the Hadamard transform to a given initial state.

Firstly Deutsch's algorithm applies the Hadamard transform to the input state $|0\rangle |1\rangle$ to produce the superposition state

$$(H \otimes H) \left|0\right\rangle \left|1\right\rangle = \frac{1}{2} \left(\left|00\right\rangle - \left|01\right\rangle + \left|10\right\rangle - \left|11\right\rangle\right). \tag{105}$$

The unitary function U_f is then applied to the product state to give

$$\begin{aligned} U_f \left(H \otimes H\right) \left|0\right\rangle \left|1\right\rangle =& \frac{1}{2} \left(\left|0\right\rangle \left|0 \oplus f(0)\right\rangle - \left|0\right\rangle \left|1 \oplus f(0)\right\rangle \right. \\ & \left. + \left|1\right\rangle \left|0 \oplus f(1)\right\rangle - \left|1\right\rangle \left|1 \oplus f(1)\right\rangle\right) \\ =& \frac{1}{2} \left((-1)^{f(0)} \left|0\right\rangle \left(\left|0\right\rangle - \left|1\right\rangle\right) + (-1)^{f(1)} \left|1\right\rangle \left(\left|0\right\rangle - \left|1\right\rangle\right)\right) \\ =& \frac{1}{2} \left((-1)^{f(0)} \left(\left|00\right\rangle - \left|01\right\rangle\right) + (-1)^{f(1)} \left(\left|10\right\rangle - \left|11\right\rangle\right)\right). \end{aligned} \tag{106}$$

To get the output state the Hadamard gate is applied to the first qubit, leaving the second one alone, by applying $H \otimes I$ to give $|\psi\rangle = (H \otimes I) U_f (H \otimes H) |0\rangle |1\rangle$.

Now since $(H \otimes I)\,|ab\rangle = H\,|a\rangle \otimes |b\rangle$ if follows that

$$\begin{aligned}
|\psi\rangle =& (H \otimes I)\, U_f\, (H \otimes H)\, |0\rangle\, |1\rangle \\
=& (H \otimes I)\, \frac{1}{2} \left((-1)^{f(0)} (|00\rangle - |01\rangle) + (-1)^{f(1)} (|10\rangle - |11\rangle) \right) \\
=& \frac{1}{2\sqrt{2}} \left[\left((-1)^{f(0)} + (-1)^{f(1)} \right) |00\rangle + \left((-1)^{f(0)} - (-1)^{f(1)} \right) |10\rangle \right. \\
& \left. - \left((-1)^{f(0)} + (-1)^{f(1)} \right) |01\rangle - \left((-1)^{f(0)} - (-1)^{f(1)} \right) |11\rangle \right] \\
=& \frac{(-1)^{f(0)}}{2\sqrt{2}} \left[\left(1 + (-1)^{f(1)-f(0)} \right) (|00\rangle - |01\rangle) + \left(1 - (-1)^{f(1)-f(0)} \right) (|10\rangle - |11\rangle) \right] \\
=& \frac{(-1)^{f(0)}}{2\sqrt{2}} \left[\left(1 + (-1)^{f(1)-f(0)} \right) |0\rangle + \left(1 - (-1)^{f(1)-f(0)} \right) |1\rangle \right] (|0\rangle - |1\rangle) \,. \quad (107)
\end{aligned}$$

If the function is constant, such that $f(0) = f(1)$, then

$$|\psi\rangle = \frac{(-1)^{f(0)}}{\sqrt{2}} |0\rangle\, (|0\rangle - |1\rangle) \,. \quad (108)$$

If the function is balanced, such that $f(0) \neq f(1)$, then

$$|\psi\rangle = \frac{(-1)^{f(0)}}{\sqrt{2}} (\pm |1\rangle)\, (|0\rangle - |1\rangle) \,. \quad (109)$$

So if the function is constant then the first qubit is zero but if the function is balanced then the first qubit is one. A classical computer would have to evaluate the function for both inputs to prove if f is constant or balanced but Deutsch's algorithm computes both at once because of the quantum superposition. This algorithm is of little practical use but it is an example of the exponential speed up offered by quantum computers in certain situations.

3.2.4 Deutsch-Josza algorithm

Deutsch's algorithm may be generalised to allow for functions with multiple input values, this is knowns as the Deutsch-Jozsa algorithm. If the function $f(x)$ is constant then the output is the same for all values of x but if the function is balanced then $f(x) = 0$ for half of the inputs and $f(x) = 1$ for half the inputs. Again the algorithm starts with the Hadamard transform of the initial state which this time consists of n qubits in the state $|0\rangle$ and one qubit in the state $|1\rangle$ such that

$$\begin{aligned}
|\psi'\rangle =& \left(H^{\otimes n} \right) \left(|0\rangle^{\otimes n} \right) \otimes (H\,|1\rangle) \\
=& \left(\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) . \quad (110)
\end{aligned}$$

Next the unitary function U_f is applied to the product state where the values of x are the first n qubits and the last qubit is the value of y. The output state is

$$\begin{aligned}|\psi''\rangle =& U_f \left(H^{\otimes n}\right)\left(|0\rangle^{\otimes n}\right) \otimes \left(H|1\rangle\right) \\ =& \left(\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle\right) \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right). \end{aligned} \tag{111}$$

The Hadamard gate is applied to the first n qubits, leaving the last qubit alone, to give the output state

$$|\psi_{out}\rangle = \frac{1}{2^n} \sum_{y\in\{0,1\}^n} \left[\sum_{x\in\{0,1\}^n} (-1)^{x\cdot y + f(x)}\right] |y\rangle, \tag{112}$$

where $x \cdot y = x_0 y_0 \oplus x_1 y_1 \oplus \ldots \oplus x_{n-1} y_{n-1}$. The first n qubits are then measured, if they are all zero then the function $f(x)$ is constant but if at least one qubit is one then the function is balanced. The probability of measuring $|0\rangle^{\otimes n}$ in the first n qubits is

$$\left|\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}\right|^2 \tag{113}$$

which is one if $f(x)$ is constant and zero if $f(x)$ is balanced.

3.2.5 Quantum Fourier transform

The quantum Fourier transform is a transformation acting on n qubits which is analogous to the discrete Fourier transform of classical computing. The quantum Fourier transform operaor U_F acts on an arbitrary basis state $|x\rangle = |x_{n-1}x_{n-2}\ldots x_0\rangle$, where $x_i \in \{0,1\}$, and maps it to a quantum state

$$\begin{aligned} U_F|x\rangle =& \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i x y/2^n} |y\rangle \\ =& \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i (y_0(0.x_0x_1\ldots x_{n-1}) + y_1(0.x_0x_1\ldots x_{n-2}) + \ldots + y_{n-1}(0.x_0))} |y\rangle. \end{aligned} \tag{114}$$

Here the fractional binary notation

$$(0.x_0x_1\ldots x_n) = \sum_{k=1}^{n} x_k 2^{-k} \tag{115}$$

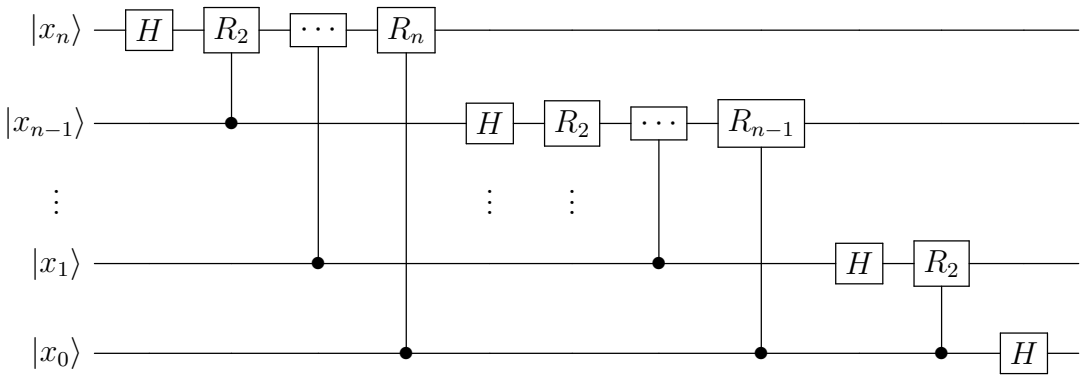

Figure 9: Circuit diagram representation of the quantum Fourier transform.

is useful to simplify things. The quantum Fourier transform can be represented as a tensor product state by

$$U_F \left|x\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i y_0 (0.x_0 x_1 \ldots x_{n-1})} \left|y_0\right\rangle \otimes \ldots \otimes e^{2\pi i y_0 (0.x_0)} \left|y_{n-1}\right\rangle, \tag{116}$$

where each $y_k \in \{0, 1\}$ and $|y\rangle = |y_0\rangle \otimes |y_1\rangle \otimes \ldots \otimes |y_{n-1}\rangle$. It is clear to see that if $y_k = 0$ then $e^{2\pi i y_k (0.x_0 x_1 \ldots x_j)} = e^0 = 1$ but if $y_k = 1$ then $e^{2\pi i y_k (0.x_0 x_1 \ldots x_{n-1})} = e^{2\pi i (0.x_0 x_1 \ldots x_{n-1})}$. The quantum Fourier transform combines superposition states and discrete phase shifts and is therefore achieved by application of the Hadamard gate and the discrete phase gate

$$\boxed{R_k} = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix} = \left|0\right\rangle \left\langle 0\right| + e^{2\pi i/2^k} \left|1\right\rangle \left\langle 1\right|. \tag{117}$$

Figure 9 shows a circuit diagram for the quantum Fourier transform using Hadamard and discrete phase shift gates. The quantum Fourier transform is effectively split into n single-qubit operations with the other qubits acting as controls. The first term requires a Hadamard gate and $n-1$ controlled phase gates and the second term requires a Hadamard gate and $n-2$ controlled phase gates etc. For this implementation $n(n+1)/2 = O(n^2)$ gates are required however there are algorithms that require only $O(n \log n)$ gates to find an efficient approximation. The classical discrete Fourier transform requires $O(n2^n)$ gates which is exponentially more than its quantum counterpart.

3.2.6 Grover's algorithm

Another example of a basic quantum algorithm which improves significantly on its classical counterpart is Grover's algorithm for database searching. Grover's

algorithm searches over a set of $N = 2^n$ items to find a unique element based on some specified criterion. Grover's algorithm requires $O(\sqrt{N})$ operations whereas a classical search of unordered data requires $O(N)$ operations.

Grover's algorithm begins by creating an equal superposition of all possible 2^n states of the n-qubit register. So every configuration has an equal amplitude of $1/\sqrt{2^n}$ and an equal probability of existing when the system is measured of $1/2^n$. Each state corresponds to an entry in the database so because of the superposition every element in the database may be considered at once. After this Grover's algorithm seeks to manipulate the amplitudes of the states to increase the likelihood that the correct database entry is returned when the system is measured. The technique of exploiting the qualities of quantum amplitudes is called amplitude amplification and it relies upon selective phase shifting of a particular state that satisfies some condition. This phase selection is only possible by using quantum amplitudes rather than just the probabilities.

Firstly the system is put into an equal superposition of states by applying the Hadamard transform to a register of n qubits all initialised to $|0\rangle$ such that

$$|\psi\rangle = H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \,. \tag{118}$$

After this so-called 'Grover iteration', which performs amplitude amplification, is applied at least $\pi\sqrt{2^n}/4$ times. Each Grover iteration consists of a call to a 'quantum oracle' and the diffusion transform. An oracle is a function which recognises if the system is in the correct state and modifies it without collapsing the superposition. The oracle rotates the phase of the correct state by π radians and does nothing to any incorrect states. The oracle $\mathcal{O}$ produces the effect

$$|x\rangle \to (-1)^{f(x)} |x\rangle \,, \tag{119}$$

where $f(x) = 1$ is x is in the correct state and $f(x) = 0$ otherwise. The implementation of $f(x)$ is dependent upon the database and the search criteria.

The next stage of each Grover iteration is the diffusion transform which modifies the amplitude of each state so that it is a far above the average as it was below the average before the transformation and vice versa. The diffusion transform consists of an application of the Hadamard transform, then a conditional phase shift (shifts every state by -1 except $|0\rangle$) and then anther application of the Hadamard transform. So the diffusion transform is given by

$$H^{\otimes n} \left[2 |0\rangle \langle 0| - I\right] H^{\otimes n} = 2H^{\otimes n} |0\rangle \langle 0| H^{\otimes n} - I = 2 |\psi\rangle \langle \psi| - I. \tag{120}$$

A single Grover iteration may be written as

$$\left[2 |\psi\rangle \langle \psi| - I\right] \mathcal{O}. \tag{121}$$

The Grover iteration is applied $K = \pi\sqrt{2^n}/4$ times to given the output state

$$|\psi_{out}\rangle = \left([2\,|\psi\rangle\,\langle\psi| - I]\,\mathcal{O}\right)^K |\psi\rangle \tag{122}$$

when the register is measured. The probability of returning the correct answer is $O(1)$ and the algorithm becomes more accurate as the size of the database becomes larger.

There are many other algorithms which could be mentioned to demonstrate the power of quantum systems, such as Shor's algorithm for factoring numbers or Simon's algorithm for period finding, but there are too many to mention in this brief introduction. There are however a few specific algorithms which are worth discussing in the context of applying quantum computers to the modelling and simulation of physical systems. However the discussion of numerical analysis on quantum computers will be left for part II of this introduction to quantum numerical algorithms.

References

[1] Cao, Y., Papageorgiou, A., Petras, I., Traub, J. and Kais, S., 2013. Quantum algorithm and circuit design solving the Poisson equation. *New Journal of Physics*, 15(1), p.013021.

[2] Deutsch, D., 1985. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A*, 400(1818), pp.97-117.

[3] Feynman, R.P., 1982. Simulating physics with computers. *International journal of theoretical physics*, 21(6-7), pp.467-488.

[4] Grover, L.K., 1996, July. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (pp. 212-219). ACM.

[5] Harrow, A.W., Hassidim, A. and Lloyd, S., 2009. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15), p.150502.

[6] McMahon, D., 2007. *Quantum computing explained*. John Wiley & Sons.

[7] Pittenger, A.O., 2000. *An introduction to quantum computing algorithms*. Progress in Computer Science and Applied Logic, (Vol. 19). Birkhäuser

[8] Shor, P.W., 1994, November. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on* (pp. 124-134). Ieee.

[9] Simon, D.R., 1997. On the power of quantum computation. *SIAM journal on computing*, 26(5), pp.1474-1483.