De-quantisation in Quantum Computation

Alastair Avery Abbott

under the supervision of

Professor Cristian S. Calude

Honours Thesis

A thesis submitted in partial fulfilment of the requirements
for the degree of Bachelor of Science (Honours)
in Computer Science, The University of Auckland, 2009.

Abstract

Quantum computation has shown much promise at providing, at least in some cases, a significant advantage over classical computation. However, the nature of quantum computation is still far from being well understood. In order to develop quantum algorithms effectively, it is important to understand the true nature of the differences between classical and quantum computation. We investigate these differences more closely by looking at de-quantising quantum algorithms into classical counterparts which retain the benefit provided by, and thought to be intrinsic to, the quantum algorithms.

We extend a previous de-quantisation of Deutsch's problem to show that in some situations the quantum algorithm solving the Deutsch-Jozsa problem can be de-quantised into an equivalent classical one. We quantify the entanglement in this problem and show that the inability to easily extend the de-quantisation to the general case is a result of the entanglement destroying the concise classical state description needed to de-quantise such a black-box algorithm.

We further show that the quantum Fourier transform, an important process in many quantum algorithms, in its standard form is de-quantisable. This highlights key misconceptions about both the quantum Fourier transform and quantum computation itself. In such cases it is the linearity of quantum mechanics which allows constructing quantum algorithms which offer advantage over classical algorithms. The careful investigation of de-quantisation in relation to these problems allows a deeper insight into the nature of quantum computation.

Acknowledgements

I would particularly like to thank my supervisor, Cristian Calude, for his encouragement of, and enthusiasm for my work. I would also like to thank the Centre for Discrete Mathematics and Theoretical Computer Science, as well as the Department of Computer Science for providing funding to travel to the conference 'Unconventional Computation 2009' and present some of this work at the associated 'Workshop on Physics and Computation'.

Contents

List of Figures

Chapter 1

Introduction

"We always have had a great deal of difficulty in understanding the world view that quantum mechanics represents . . . every new idea, it takes a generation or two until it becomes obvious that there's no real problem. It has not yet become obvious to me that there's no real problem. I cannot define the real problem, therefore I suspect there's no real problem, but I'm not sure there's no real problem. So that's why I like to investigate things."

— Richard Feynman, 1982

Quantum mechanics is now a well-established field and physics has truly taken a step out of the classical domain. The field of computer science, however, is still far from following in the footsteps of physics. Since Feynman proposed a transition to the world of quantum computation in order to truly simulate the physical world, quantum computation has enjoyed variable amounts of attention. Potential applications such Shor's prime factorisation algorithm and quantum communication protocols have shown promise, but the shortage of known applications and difficulties in constructing quantum computational devices has dampened such efforts.

While it seems quantum computation realises the true computation of nature which classical computation failed to do, its utility for problems which fall within the classical domain is still poorly understood. In this thesis we investigate a question relating to the advantage of a quantum algorithm over a classical one for the same problem: when is it possible to *de-quantise* a quantum algorithm into a classical counterpart without loss of efficiency? This question has received surprisingly little research, even though it is not known if most the known quantum algorithms actually provide any advantage over a potential classical algorithm.

We specifically investigate two algorithms: the algorithm solving the Deutsch-Jozsa problem and the quantum Fourier transform algorithm. We show that in many cases the quantum solution, which is believed to be better than any classical solution, can in fact be de-quantised into an equivalent classical solution. This is particularly interesting in the case of the quantum Fourier transform, not only because it plays an important part in many other algorithms, but because it highlights properties that are fundamentally important to quantum computation.

Such an investigation of de-quantisation not only leads to classical algorithms which are more efficient than previously known algorithms, but also helps to highlight what properties a quantum algorithm may need if it is to pose an advantage over any classical algorithm. This illustrates that quantum computation can serve as a means to obtain new and better classical algorithms.

Chapter 2

Quantum computation

In this chapter we will outline the key concepts and definitions in the computational field of quantum computation. These ideas follow from a quantum mechanical treatment of information, and a development of quantum computation starting from quantum mechanical principles gives a more thorough overview of the field. The scope of this chapter however does not extend this far; for such a development the reader should refer to [Gru99]. We will provide a background sufficient for the area of quantum algorithms with which this thesis is concerned.

2.1 History and motivation

Quantum computation has its roots in the insight of physicist Richard Feynman. Feynman was concerned with simulating physics with computers, and argued strongly that classical computers cannot universally simulate quantum physics without avoiding an exponential explosion of computational resources as the size of the physical system increases [Fey82]. Feynman's argument was brought more strongly to the field of computer science by David Deutsch. Deutsch argued that our notion of what is computable should extend to the functions which could possibly be computed by a physical system. There are many physical process, both dynamical and quantum mechanical, which cannot be classically computed but should nonetheless be considered computable. Deutsch proposed a *physical Church-Turing principle*, which states that any finitely realisable physical system can be simulated by a finitely specified universal computational model [Deu85].

It is from these ideas that quantum computation arose. A computational model based on the fundamental laws of physics as best we understand them certainly overcomes the gap to the physical Church-Turing principle, and is a much more suitable candidate for universal computation in this broader sense. Since quantum computation was first presented by Deutsch, various algorithms and results have arisen which appear to be beyond the scope of classical computation. Most notable is Shor's efficient algorithm for prime factorisation [Sho94].

2.2 Fundamentals of quantum computation

The fundamental step towards quantum computation is the identification of a classical bit, i.e. a 0 or a 1, with a two-state quantum system, such as the spin of a particle. This is motivated by the observation that any real computational device must encode its bits in a physical medium. There is no reason that this physical medium cannot be one in which quantum mechanical behaviours are observable. The quantum mechanical laws of evolution must then apply to these quantum bits, or *qubits*, and it is from these observations that our model of quantum computation arises.

2.2.1 Qubits

We define a classical bit as an element $b \in \mathbb{B} = \{0, 1\}$. For historical reasons, the so-called ket notation, denoted $|\cdot\rangle$, is employed to denote the state of a qubit. The set of classical bits, $\mathbb{B}$, is identified with a set of qubits, $\mathcal{B} = \{|0\rangle, |1\rangle\}$. Quantum mechanics permits any arbitrary superposition of these basis-states to be a valid state of a qubit. It is often convenient to think of a qubit as a unit vector in a two-dimensional Hilbert space, $\mathcal{H}_2$, spanned by the basis $\mathcal{B}$.

Definition 1. A *qubit* $|\psi\rangle$ is an arbitrary superposition of the basis-states $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix},$$

where $\alpha, \beta \in \mathbb{C}$ are complex numbers, subject to the *normalisation condition*:

$$|\alpha|^2 + |\beta|^2 = 1.$$

The tensor product, $\otimes$, can be used to create states in higher-dimensional Hilbert spaces. This allows us to create the quantum analogue of bit strings.

Definition 2. The *tensor product* of two qubits $|\phi_1\rangle$ and $|\phi_2\rangle$ is a state in $\mathcal{H}_2 \otimes \mathcal{H}_2 = \mathcal{H}_4$:

$$\begin{aligned} |\psi\rangle &= |\phi_1\rangle \otimes |\phi_2\rangle \\ &= (\alpha_1|0\rangle + \beta_1|1\rangle)(\alpha_2|0\rangle + \beta_2|1\rangle) \\ &= \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle, \end{aligned}$$

where all the following are equivalent notations: $|ab\rangle = |a\rangle|b\rangle = |a\rangle \otimes |b\rangle$.

A classical n-bit register is a bit string $x \in \mathbb{B}^n$. Quantum registers containing n qubits can then be defined similarly to single qubits in the Hilbert space $\mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_2 = (\mathcal{H}_2)^{\otimes n} = \mathcal{H}_{2^n}$. Such a register is a superposition of states in the *computational basis*, $\mathcal{B}^n = \{|x\rangle \mid x \in \mathbb{B}^n\} = \{|i\rangle \mid i = 0, \ldots, 2^n - 1\}$, where the bitstring x is often written in decimal form as i.

Definition 3. A *register* $|\psi\rangle$ of n qubits is the state:

$$|\psi\rangle = \sum_{x \in \mathbb{B}^n} \alpha_x |x\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle ,$$

with $\alpha_x \in \mathbb{C}$, subject to the *normalisation condition*:

$$\sum_{x \in \mathbb{B}^n} |\alpha_x|^2 = 1.$$

An important note is that for a register of n qubits, any vector in $\mathcal{H}_{2^n}$ represents a valid quantum state, even those which cannot be expressed as the tensor product of n individual qubits. Such states which are not *separable* are said to be *entangled*. For a 2-qubit state the condition for separability (using the notation of Definition 3) is $\alpha_{00}\alpha_{11} = \alpha_{01}\alpha_{10}$. The most well known example of entangled states are the Bell states [Gru99]:

$$|\Phi^\pm\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle), \tag{2.1}$$

$$|\Psi^\pm\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle). \tag{2.2}$$

Entangled states, as we will see, are of much significance in quantum computation. They have the property that they exhibit correlations higher than anything achievable classically yet remain a perfectly valid physical state.

2.2.2 Operations on qubits

Measurement

One of the most important operations is one which we take for granted in classical computation, rarely even thinking of it as an operation. This operation is measurement and, unlike in the classical situation, it has an important, non-trivial role in quantum computation. The reason for this is that while a state can be in an arbitrary superposition of basis-states, measurement can only be performed on the computational basis. The effect of such a measurement irreversibly alters the measured state. This is summarised in the *Born rule*, first expressed by the physicist Max Born [Bor26]. This rule states that given an arbitrary superposition, the probability of measuring a state $|x\rangle$ is proportional to the amplitude squared, $|\alpha_x|^2$, and after measurement the superposition is destroyed and the state 'collapses' to the measured basis-state $|x\rangle$.[1] If a subset of qubits in a register are measured, the state collapses to the subspace spanned by the remaining qubits.

Born rule. *Given an arbitrary state $|\psi\rangle$, measurement has the following effect (the meter represents measurement):*

$$|\psi\rangle = \textstyle\sum_{x\in\mathbb{B}^n} \alpha_x |x\rangle \;-\!\boxed{\measuredangle}\!-\; |a\rangle\,,$$

where $a \in \mathbb{B}^n$ is obtained with probability $|\alpha_a|^2$, and the state collapses to the basis-state $|a\rangle$. In the more general case in which we wish to measure the first n bits of an $n+m$ bit register, we have [Gru99]:

$$|\psi\rangle = \textstyle\sum_{x\in\mathbb{B}^n} \sum_{y\in\mathbb{B}^n} \alpha_{xy} |x\rangle |y\rangle \;-\!\boxed{\measuredangle}\!-\; \frac{1}{\sqrt{P(a)}} \sum_{y\in\mathbb{B}^m} \alpha_{ay} |a\rangle |y\rangle\,,$$

where $P(a) = \mathbf{P_C}\left(|\psi\rangle, a, (1,n)\right) = \sum_{y\in\mathbb{B}^m} |\alpha_{ay}|^2$ is the probability of obtaining 'a' when measuring bits 1 to n of $|\psi\rangle$. The state then collapses onto the subspace spanned by the $|a\rangle|y\rangle$ basis elements.

The Born rule is important in quantum computation because early measurement can result in destroying the computation. Computations must then be designed carefully around the measurement requirement. This effect is not all negative though, as it also gives rise to the fields of quantum cryptography and teleportation in which it is of fundamental importance [Mer07]. The phenomenon of state collapse also has big implications for any potential implementation of a quantum computer because of the importance of isolating the computer from external perturbations and noise which cause the phenomenon known as decoherence [NC00].

Unitary operations

The first-order differential equation which specifies the evolution of quantum states, the Schrödinger Equation, is reversible in time [Sak94]. Along with the conservation of probability (which requires registers to be unit vectors) this means only unitary linear transformations are permitted [Gru99].

[1] For the purposes of this thesis, we will assume measurement only on the computational basis. General quantum theory and the Born rule allows for measurement and probability calculation along any complete orthonormal basis, but in quantum computation we shall only concern ourselves with the computational basis. In any cases, the state irreversibly collapses onto the measurement basis upon measurement.

Definition 4. A *unitary operator* U is one which satisfies the condition

$$UU^\dagger = U^\dagger U = I,$$

where $U^\dagger$ denotes the Hermitian adjoint operator, which is also unitary. Evidently, $U^\dagger = U^{-1}$ is the unitary inverse and thus the reversibility condition is satisfied. This definition is easily shown to be equivalent to the inner product between two states $|\psi_1\rangle$ and $|\psi_2\rangle$, $\langle\psi_1|\psi_2\rangle$, being preserved by the transformation U, ensuring probability is also conserved.

It is often convenient to specify a unitary operator U in matrix notation. The operator U acting on a n-qubit register is a $2^n \times 2^n$ matrix operating on the associated 2^n-element column-vector in $\mathcal{H}_{2^n}$. The inverse $U^\dagger = (U^*)^T$ is the complex-conjugate transpose of U.

The linearity of unitary operators means that in order to specify their effect, we need only specify the effect they have on the basis-states. This can easily be seen by noting that for an arbitrary state $|\psi\rangle$,

$$U|\psi\rangle = U \sum_{x \in \mathbb{B}^n} |x\rangle = \sum_{x \in \mathbb{B}^n} U|x\rangle .$$

Thus we need only to specify $U|x\rangle$ to specify the action of U on an arbitrary qubit.

It is interesting to note the differences between the nature of measurement compared to the unitary evolution of quantum states. The unitary evolution of a quantum state follows from basic quantum mechanical theory, while the irreversible phenomenon of measurement and the Born rule are only postulated. However, it is extremely well verified experimentally and we believe that the probabilistic nature of quantum mechanics is a fundamental property of the physical world [Sak94].

Basis for quantum computation

In order for quantum computation to be a reasonable physical computational model, we would like a finite set of unitary transformations—or gates as they are often called—to be used as a basis from which to construct all unitary transformations. This is equivalent to the set $\{AND,\ OR,\ NOT\}$ providing a universal basis for classical computation. Fortunately there are many such universal bases which can simulate any gate with arbitrarily high accuracy, and as little as one 2-qubit and two 1-qubit gates suffice [NC00]. A common example of such a basis is $\mathcal{S} = \{cNOT,\ H,\ R(\pi/4)\}$, details of which are given in Example 5. The gates of this basis are some of the most commonly used in quantum computation.

The efficiency of such a simulation varies with different bases, but has been studied extensively (see [KSV02, NC00]). We will briefly summarise some key results which hold with the basis $\mathcal{S}$ given above. To construct an arbitrary m-qubit unitary transformation out of gates in $\mathcal{S}$ requires $O(m4^m)$ gates [BBC+95]. This is only an upper bound, but there exist transformations which are proven to require at least exponentially many gates to simulate [NC00]. To avoid such an exponential blow-up, we wish to express a computation in gates which operate on a fixed number of qubits d, independent of the input size n. A result known as the *Solovay-Kitaev theorem* [KSV02] shows us that any 1-qubit and 2-qubit gate can be realised with precision ϵ using $O(\log^c(1/\epsilon))$ gates from $\mathcal{S}$, where $c \approx 2$ is a constant. Hence, if we can express a computation as consisting of $T(n)$ gates, each operating on no more than d qubits, we can realise this circuit with gates from $\mathcal{S}$ using $O\left(d4^d \log^c\left(T(n)/\epsilon\right) T(n)\right)$ gates, a poly-logarithmic increase. We are thus justified to consider computations with gates acting on a fixed number of qubits when developing quantum algorithms.

Example 5. Figure 2.1(a) shows the controlled-*NOT* (*cNOT*) gate, which flips the value of a target bit only if the control bit c is $|1\rangle$. A *cNOT* gate can be described by its action on the four basis-states:

$$\begin{aligned}
|0\rangle\,|0\rangle &\xrightarrow{cNOT} |0\rangle\,|0\rangle\,, \\
|0\rangle\,|1\rangle &\xrightarrow{cNOT} |1\rangle\,|1\rangle\,, \\
|1\rangle\,|0\rangle &\xrightarrow{cNOT} |1\rangle\,|0\rangle\,, \\
|1\rangle\,|1\rangle &\xrightarrow{cNOT} |0\rangle\,|1\rangle\,.
\end{aligned}$$

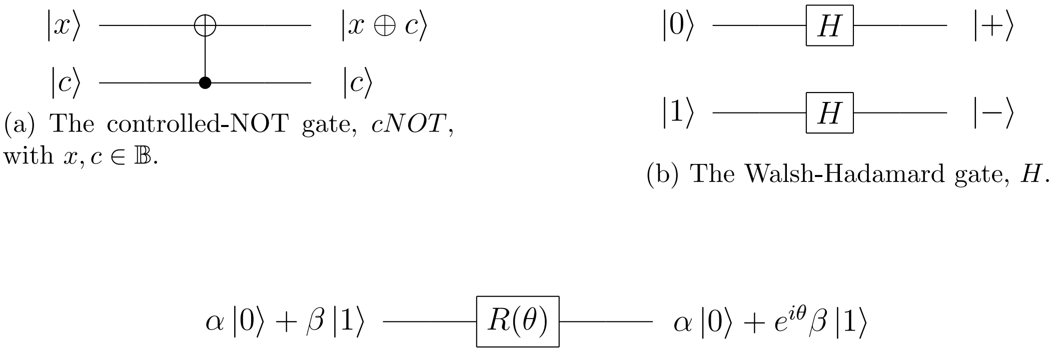
Figure 2.1(b) shows the Walsh-Hadamard gate, or Hadamard gate for short, operating on the two basis-states. The Hadamard gate creates an equal superposition of basis-states on non-superposition input. We use the notation $|+\rangle$ and $|-\rangle$ to represent symmetric and antisymmetric equal superpositions, so the the Hadamard gate has the following effect:

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

As an example of the matrix representation of gates, we can express the Hadamard gate as the matrix:

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

The final gate in this basis, the phase-shift gate $R(\theta)$, is shown in Figure 2.1(c). This gate simply introduces a complex phase factor of $e^{i\theta}$ if the qubit is $|1\rangle$, and does nothing otherwise. Note that θ is in fact a continuous parameter. In this case, our basis is no longer finite[2] but can in fact simulate any gate perfectly [NC00]. However, if we take $\theta = \pi/4$ then our gate remains universal to within arbitrary accuracy.



(a) The controlled-NOT gate, $cNOT$, with $x, c \in \mathbb{B}$.

(b) The Walsh-Hadamard gate, H.

(c) The phase-shift gate, $R(\theta)$.

Figure 2.1: The common universal basis of gates, $\mathcal{S}$.

2.2.3 Quantum computations

Quantum circuits

Loosely speaking, a quantum circuit is a sequence of unitary and measurement gates acting on a N-qubit register $|x, 0^{N-n}\rangle$, which computes a function $F : \mathcal{H}_{2^n} \to \mathcal{H}_{2^m}$ on input $|x\rangle \in \mathcal{H}_{2^n}$. Note that N is often larger than the input size n, since we may need extra bits to ensure unitarity or to act as auxiliary bits during the computation. We say that the sequence of gates $G = G_L G_{L-1} \cdots G_1$ computes F if it produces the state $|F(x), 0^{N-m}\rangle$ with high probability. Note that a unitary gate $U^{(S)}$ acting on a set of qubits S is assumed to act as the identity on all qubits not in S.

Definition 6. A *quantum circuit* $G = G_L G_{L-1} \cdots G_1$ *computes* a function $F : \mathcal{H}_{2^n} \to \mathcal{H}_{2^m}$ if for all $|x\rangle \in \mathcal{H}^n$ we have:

$$\mathbf{P_G}\left(G\,|x, 0^{N-n}\rangle = |F(x), 0^{N-m}\rangle\right) \geq 1 - \epsilon,$$

where $\epsilon < 1/2$, $\mathbf{P_G}\left(|a\rangle = |b\rangle\right)$ is the probability that $|a\rangle = |b\rangle$. We say that a circuit containing L gates has *size* L.

Note that the probability $\mathbf{P_G}$ differs from $\mathbf{P_C}$ used in the Born Rule (page 6) in that it is not the measurement probability. If measurement is intended to be done at the

[2] It is worth noting that perhaps this is not so unreasonable. A physical implementation of an $R(\theta)$ gate could easily introduce a phase dependant on how long the gate is applied. Indeed, the time evolution operator in quantum mechanics is $e^{iEt/\hbar}$ for a free state with energy E. One would need to apply such a gate to an individual qubit on demand, but in principle one physical mechanism surely can account for an infinity of gates.

end of the computation then a measurement gate should be the final one in the circuit, and the collapsed states will then be obtained with probability according to the Born rule. Thus, a circuit containing no measurement gates is deterministic and computes the function implemented by $U = U_L \cdots U_1$ with probability one.

This definition of computation extends beyond classical functions on n bits to ensure that quantum circuits, such as that computing the quantum Fourier transform (see Section 4), are valid quantum computations, even if measurement does not directly yield any classical information. However, in the case where the circuit $G = M_L U_{L-1} \cdots U_1$, i.e. $L-1$ unitary gates followed by measurement, our definition is equivalent to the standard definition for a quantum circuit implementing a classical function on n bits [KSV02].

Definition 7. The circuit $G = M_L U = M_L U_{L-1} \cdots U_1$ *computes a classical function* $F : \mathbb{B}^n \to \mathbb{B}^m$ if for all $x \in \mathbb{B}^n$ we have:

$$\mathbf{P_G}\left(G\left|x, 0^{N-n}\right\rangle = \left|F(x), 0^{N-m}\right\rangle\right) = \mathbf{P_C}\left(U\left|x, 0^{N-n}\right\rangle, F(x), (1, m)\right) \geq 1 - \epsilon,$$

where $\epsilon < 1/2$.

Quantum circuits can be diagrammatically represented in a similar fashion to classical circuits. Each line represents a qubit or register of qubits, and time moves left to right. Figure 2.2 shows a circuit to prepare the entangled Bell state in Equation 2.1.

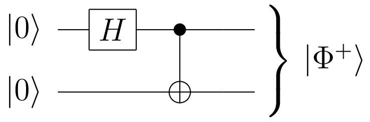


Figure 2.2: A quantum circuit generating an entangled $|\Phi^+\rangle$ Bell state.

Quantum algorithms

A quantum algorithm can be naturally defined from circuits in a similar way to classical algorithms. Given a function F operating on Hilbert space, a quantum algorithm computing F is an infinite sequence of quantum circuits $(G_0, G_1, G_2, \ldots)$ such that G_n computes F on inputs of size n, and the circuit can be generated for any n by a classical Turing machine [KSV02, Sip06].

Definition 8. A *quantum algorithm* computing the function $F : \mathcal{H} \to \mathcal{H}$ is an infinite, *uniform*, sequence of quantum circuits $(G_0, G_1, G_2, \ldots)$ such that every G_n computes the function $F_n : \mathcal{H}_{2^n} \to \mathcal{H}_{2^{m(n)}}$, and the size of the output m is a function of the input n. By uniform we mean that there exists a classical Turing machine which constructs G_n on input n. We say that an algorithm *runs in time* $T(n)$ if the circuit G_n has size $T(n)$.

Oracle quantum computation

Quantum oracle, or *black-box*, computation is the natural extension of classical oracle computation, in which we are supplied with an oracle which we may 'query' a finite number of times, providing the correct answer each time [KSV02]. In the classical case, the oracle may be viewed as a black-box into which we supply a query and the answer is supplied from the other side. However, we must have no knowledge or method of determining how the black-box performs this task and can perform no operation on it other than querying it. In quantum computation, the black-box takes the form of a given unitary gate of arbitrary complexity, which we may use a finite number of times, but have no knowledge about how it is constructed [BB94].

2.3 De-quantisation

One of the main attractions of quantum computers is the possibility of solving problems more efficiently than any classical computer can, or solving problems that no classical computer can solve. In order to construct good quantum algorithms it is important to know what features allow a quantum algorithm to be better than a classical one. Many quantum algorithms have a trivial classical counterpart: all the operations in the matrix mechanics formulation of quantum mechanics can be easily computed by classical means [EJ98], however, care must be taken. Quantum algorithms which involve uncomputable numbers,[3] or those which rely fundamentally on the true randomness of measurement, can not easily be obtained by this means. Unfortunately, in this kind of simulation the dimension of the Hilbert space grows exponentially with the number of qubits used in a quantum algorithm, so a classical counterpart obtained by the trivial means takes space and time that is exponentially larger than the quantum algorithm requires. In this thesis we examine the ability to *de-quantise* a quantum algorithm to obtain a classical counterpart which is not exponential in time or space compared to the quantum algorithm, and explore when such a de-quantisation is possible. De-quantisation was first explored by Calude in [Cal07], where Deutsch's famous problem was de-quantised. We will review this result before exploring de-quantisation in both the more general Deutsch-Jozsa problem as well as in the quantum Fourier transform.

[3] It is not known if such a gate involving uncomputable numbers exists. Kitaev et al. present an excellent discussion on this [KSV02, p. 90], and it is interesting to note that such a gate would allow computing classically uncomputable numbers. The potential existence of such transformation is intimately grounded in the nature of physics itself and is of great interest.

Chapter 3

De-quantisation in the Deutsch-Jozsa problem

In this chapter we will develop an in-depth study of de-quantisation in the Deutsch-Jozsa problem, as well as a few general results about de-quantisation, particularly with respect to black-box algorithms. We will review Deutsch's problem and the initial de-quantisation of the algorithm solving it by Calude. We will then apply the same de-quantisation technique to the more general Deutsch-Jozsa problem. The final part of this chapter explores the separability of general qubit states as well as the states in the Deutsch-Jozsa problem, before exploring some general results on de-quantisation.

3.1 Deutsch's problem

The original problem proposed by Deutsch [Deu85] is formulated as follows: consider a Boolean function $f : \mathbb{B} \to \mathbb{B}$, and suppose we are given a black-box to compute f. Deutsch's problem is to determine if f is constant (i.e. $f(0) = f(1)$) or balanced (i.e. $f(0) \neq f(1)$) in as few as possible calls to the black-box computing f.

3.1.1 Quantum solution

Here we present a standard version of the quantum solution to Deutsch's problem which uses only one black-box call and is correct with probability one. This is based on the formulation given in [CEMM97]. A traditional classical algorithm would require two calls to a classical black-box in order to determine if f is constant or balanced; it would need to evaluate both $f(0)$ and $f(1)$. The quantum black-box acts in $\mathcal{H}_4$ and can be described by the unitary operator U_f representing an f-controlled-NOT (f-$cNOT$) gate such that

$$U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle .$$

Noting that

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

we see that U_f is its own inverse and $U_f U_f = 1$. Hence U_f is unitary and our quantum black-box is valid. In order to see how the quantum solution works, it is beneficial to observe the following:

$$\begin{aligned} U_f \left|x\right\rangle \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) &= \left|x\right\rangle \frac{1}{\sqrt{2}}(\left|0 \oplus f(x)\right\rangle - \left|1 \oplus f(x)\right\rangle) \\ &= (-1)^{f(x)} \left|x\right\rangle \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle). \end{aligned} \tag{3.1}$$

Hence we see that this state is an eigenstate of U_f and the eigenvalue is a phase which is 'kicked back' in front of the state. From this observation we can formulate the quantum solution. Taking the initial state $\left|0\right\rangle \left|1\right\rangle$ and operating on it with a 2-qubit Hadamard gate $H^{\otimes 2}$:

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

Next, operating on the state with U_f:

$$\begin{aligned} \frac{1}{\sqrt{2}} U_f \left(\left|0\right\rangle + \left|1\right\rangle\right) \left|-\right\rangle &= \frac{1}{\sqrt{2}} \left((-1)^{f(0)} \left|0\right\rangle + (-1)^{f(1)} \left|1\right\rangle\right) \left|-\right\rangle \\ &= \frac{(-1)^{f(0)}}{\sqrt{2}} \left(\left|0\right\rangle + (-1)^{f(0)\oplus f(1)} \left|1\right\rangle\right) \left|-\right\rangle , \end{aligned}$$

and applying $H^{\otimes 2}$ one more time we get

$$H^{\otimes 2} \frac{(-1)^{f(0)}}{\sqrt{2}} \left(\left|0\right\rangle + (-1)^{f(0)\oplus f(1)} \left|1\right\rangle\right) \left|-\right\rangle = (-1)^{f(0)} \left|f(0) \oplus f(1)\right\rangle \left|1\right\rangle .$$

Measuring the first qubit we obtain 0 with probability one if f is constant and 1 with probability one if f is balanced.

This quantum solution is hence correct with probability one using only one call to the quantum black-box represented by U_f. An important note is that this computation involves no entanglement. This can be seen by noting that the second qubit acts as an auxiliary bit and remains unchanged by U_f, and as a result the two qubits remain separable throughout the algorithm. This is evident in the presentation of the algorithm and as a result this quantum computation gains its power only from quantum parallelism and interference.

3.1.2 Classical solutions

While the quantum algorithm makes use of quantum parallelism and interference, these qualities (unlike entanglement) are not inherently quantum mechanical, but are rather due to the two-dimensionality of qubits compared to the one-dimensionality of classical bits. Hence, a classical two-dimensional system possesses these qualities, and should be able to achieve the same result as the quantum algorithm.

The first method, presented in [Cal07], embeds classical bits in complex numbers. The set $\mathcal{C} = \left\{1, i = \sqrt{-1}\right\}$ acts as a computational basis in the same way that $\mathcal{B} = \{|0\rangle, |1\rangle\}$ does for quantum computation. It is worth noting that while we are not labelling the basis bits '0' and '1', they represent the classical bits 0 and 1 in the same way that $|0\rangle$ and $|1\rangle$ do.

An arbitrary complex number may be written as $z = a + bi$ with $a, b \in \mathbb{R}$, so a complex number z is a natural superposition of the basis elements in the same way that a qubit is. We are now given a classical black-box which operates on complex numbers and computes our function f. This black-box can be represented by an operator C_f, a direct analogue of U_f (although the requirement of unitarity is no longer necessary). The effect of C_f (in direct correspondence with U_f, although the normalisation factors and auxiliary bit are no longer necessary) is

$$C_f(a + bi) = (-1)^{f(0)} \left(a + (-1)^{f(0) \oplus f(1)} bi\right).$$

If f is constant, C_f is the identity operation to within a factor of -1 ($C_f(x) = \pm x$). If f is balanced, C_f is the conjugation operation ($C_f(x) = \pm\overline{x}$). The black-box represented by C_f rotates the input in the complex plane depending on the nature of f, which is in direct analogy of the quantum black-box U_f operating on input $|x\rangle$ in $\mathcal{H}_2$. In order to measure the output, we need a way to project our complex numbers back on to the computational basis. This is easily done by multiplying by the input so the output is either purely imaginary or purely real.

If $z = 1 + i$ (an equal superposition of basis-states),

$$\frac{1}{2} z \times C_f(z) = \begin{cases} \frac{\pm 1}{2} z^2 = \pm i & \text{if } f \text{ is constant,} \\ \frac{\pm 1}{2} z\overline{z} = \pm 1 & \text{if } f \text{ is balanced.} \end{cases}$$

In this manner, if the output is imaginary then f is constant, if it is real then f is balanced. Importantly, this is a deterministic result, and in fact the sign of the output allows to identify *which* balanced or constant function f is. This is something the quantum algorithm provably cannot do [Mer07].

This is only one method of solving the problem as efficiently as the quantum algorithm does through classical, deterministic methods. The above method places emphasis on mathematical correspondence with the quantum solution. A different solution is presented by Arvind in [Arv01] which draws physical similarities which are more evident than in the above solution.

Arvind uses the (classical) polarisation of a photon as the computational basis $\mathcal{E} = \{x\text{-pol}, y\text{-pol}\}$, and any polarisation in the xy-plane is physically valid. It is noted that all transformations in the group SU(2)[1] can be realised by two quarter-wave plates and a single half-wave plate orientated suitably. Clearly the 1-qubit transformations required to solve Deutsch's problem are included in this, so the problem can be solved classically with one photon using wave plates. Written in matrix form the solution is mathematically identical to the quantum one. This corresponds to the following physical process: preparing a photon with y-polarisation, rotating its polarisation anti-clockwise in the xy-plane by 45°, applying the optical black-box and applying the anti-clockwise rotation once more before measuring the y-polarisation of the photon.

The correspondence here relies not on embedding classical bits in a different, classical two-dimensional basis but on directly implementing the transformations used in the quantum solution through classical means. In other words, the quantum algorithm does not take advantage of non-classical effects, so the same result can be obtained through purely classical optics.

3.2 The Deutsch-Jozsa problem

Deutsch's problem is the simplest case of the more general Deutsch-Jozsa problem [DJ92] which considers balanced and constant Boolean functions on bit strings of length n. Calude noted in his original de-quantisation [Cal07] that such a de-quantised solution can be obtained for any n, but this cannot be done uniformly because of the growth in dimension of Hilbert space. Further, because the trivial matrix simulation also scales with the Hilbert space dimension (which is exponential in the number of qubits), such a solution is of little interest. In black-box algorithms such as the one solving the Deutsch-Jozsa problem, the space required in a useful de-quantisation should scale polynomially in the number of qubits used. Indeed it should not be the case that a classical $2^n \times 2^n$ matrix embedded in a black-box suffices as a valid de-quantisation, even though it would only require a single black-box call. Evidently, it would take exponential amount of time to prepare an input to such a black-box. Because of the nature of entangled states, we expect that algorithms taking advantage of entanglement to be harder to de-quantise in a way satisfying these requirements. This issue is investigated further with relation to the Deutsch-Jozsa problem.

[1] The group SU(2) is the group of 2×2 unitary matrices with determinant 1. This is exactly the set of unitary operations on a qubit permitted by quantum mechanics.

The standard formulation of the Deutsch-Jozsa problem is as follows: let $f : \mathbb{B}^n \rightarrow \mathbb{B}$, and suppose we are given a black-box computing f with the guarantee that f is either constant (i.e. $\forall x \in \mathbb{B}^n : f(x) = a,\ a \in \mathbb{B}$) or balanced (i.e. $f(x) = 0$ for exactly half of the possible inputs $x \in \mathbb{B}^n$). Such a function f is called *valid.* The Deutsch-Jozsa problem is to determine if f is constant or balanced in as few black-box calls as possible. An important note is that unlike in Deutsch's problem, where there are exactly two balanced and two constant functions f, the distribution of constant and balanced functions is asymmetrical in the Deutsch-Jozsa problem. In general, there are $N = 2^n$ possible input strings, each with two possible outputs (0 or 1). Hence, for any given n there are 2^N possible functions f. Of these, exactly two are constant and $\binom{N}{N/2}$ are balanced. Evidently the probability that our f is constant tends towards zero very quickly (recall f is guaranteed to be valid). Furthermore, the probability that any randomly chosen function of the 2^N possible functions is valid is:

$$\frac{\binom{N}{N/2} + 2}{2^N},$$

which also tends to zero as n increases. This is evidently not an ideal problem to work with, however this does not mean that we cannot gain useful information from studying it.

3.2.1 General quantum solution

The solution to the $n = 1$ problem (i.e. Deutsch's problem) can readily be extended to solve the Deutsch-Jozsa problem for any n with exactly one black-box call. A traditional classical solution would require $2^{n-1} + 1$ black-box calls in the worst case to give an answer that is guaranteed to be correct: we could obtain the same value on the first 2^{n-1} evaluations of f, and it is not till the next evaluation that we can be sure that f is balanced or constant. Once again the formulation given is based on that in [CEMM97].

We use $n + 1$ qubits $|x\rangle |y\rangle$ where $|x\rangle$ is the n-qubit input register and $|y\rangle$ is the output register for the black-box computing f:

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

as usual. The system is initially prepared in the state

$$H^{\otimes n} |00 \cdots 0\rangle |-\rangle = |+ + \cdots +\rangle |-\rangle = \frac{1}{2^{n/2}} \sum_{x \in \mathbb{B}^n} |x\rangle |-\rangle , \tag{3.2}$$

creating an equal superposition of all possible inputs for the function f. Equation 3.1, which gives the action of U_f on an arbitrary input x, holds for any n, so operating on the prepared state with U_f gives:

$$\frac{1}{2^{n/2}} U_f \sum_{x \in \mathbb{B}^n} |x\rangle |-\rangle = \frac{1}{2^{n/2}} \sum_{x \in \mathbb{B}^n} (-1)^{f(x)} |x\rangle |-\rangle . \tag{3.3}$$

Next we note that the n-qubit Hadamard gate acting on any state $|x\rangle$, with $x \in \mathbb{B}^n$, can be written in a more general form [Mer07] as

$$H^{\otimes n} |x\rangle = \sum_{y \in \mathbb{B}^n} (-1)^{x \cdot y} |y\rangle ,$$

where $x \cdot y$ is the scalar product modulo 2, i.e.

$$x \cdot y = (x_1 \wedge y_1) \oplus \cdots \oplus (x_n \wedge y_n).$$

Applying the Hadamard gate to the result of Equation 3.3, we have

$$\frac{1}{2^{n/2}} H^{\otimes n} \sum_{x \in \mathbb{B}^n} (-1)^{f(x)} |x\rangle |-\rangle = \frac{1}{2^n} \sum_{x,y \in \mathbb{B}^n} (-1)^{f(x) \oplus x \cdot y} |y\rangle |-\rangle .$$

Observe that the amplitude of the state $|00 \cdots 0\rangle$ is

$$\frac{1}{2^n} \sum_{x \in \mathbb{B}^n} (-1)^{f(x)} = \begin{cases} \pm 1 & \text{if } f \text{ is constant,} \\ 0 & \text{if } f \text{ is balanced.} \end{cases}$$

Hence, upon measuring the this qubit register, if f is constant we measure '$00 \cdots 0$' with probability one, and if f is balanced we measure $x \neq$ '$00 \cdots 0$' with probability one. Thus, assuming f is valid, we can determine the nature of f with probability one. The circuit corresponding to this algorithm is shown in Figure 3.1.

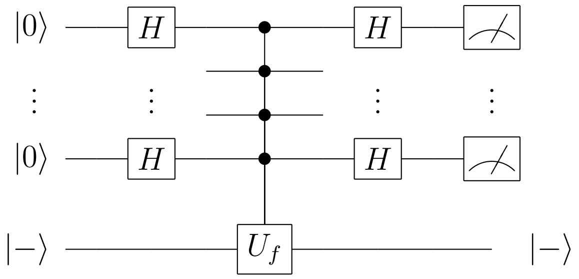


Figure 3.1: The quantum circuit solving the Deutsch-Jozsa problem with probability one.

3.2.2 The case of $n = 2$

For the Deutsch-Jozsa problem with $n = 2$ there are sixteen possible Boolean functions. Two of these are constant, another six are balanced and the remaining eight are not valid. All these possible functions are listed in Table 3.1.

$f(x)$	Constant		Balanced						Invalid							
$f(00) =$	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	1
$f(01) =$	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0
$f(10) =$	0	1	1	0	1	0	1	0	1	0	0	1	1	0	1	0
$f(11) =$	0	1	1	0	0	1	0	1	0	1	1	0	1	0	1	0

Table 3.1: All possible Boolean functions $f : \mathbb{B}^2 \to \mathbb{B}$.

Evidently, half of these functions are simply the negation of another. If we let $f'(x) = f(x) \oplus 1$, we have:

$$\begin{aligned} U_{f'} \left|x\right\rangle \left|-\right\rangle &= (-1)^{f'(x)} \left|x\right\rangle \left|-\right\rangle \\ &= -\left((-1)^{f(x)} \left|x\right\rangle \left|-\right\rangle\right) \\ &= -U_f \left|x\right\rangle \left|-\right\rangle . \end{aligned}$$

In this case the result obtains a global phase factor of -1. Since global phase factors have no physical significance to measurement (recall the Born rule), the outputs of U_f and $U_{f'}$ are physically indistinguishable.

To solve the problem, we merely need to follow the general algorithm presented on page 17. In this specific case, the action of the black-box is:

$$\begin{aligned} U_f \sum_{x\in\mathbb{B}^2} c_x \left|x\right\rangle \left|-\right\rangle &= \sum_{x\in\mathbb{B}^2} (-1)^{f(x)} c_x \left|x\right\rangle \left|-\right\rangle \\ &= \big[(-1)^{f(00)} c_{00} \left|00\right\rangle + (-1)^{f(01)} c_{01} \left|01\right\rangle \\ &\quad +(-1)^{f(10)} c_{10} \left|10\right\rangle + (-1)^{f(11)} c_{11} \left|11\right\rangle\big] \left|-\right\rangle . \end{aligned} \tag{3.4}$$

From the separability rule for 2-qubit states (page 5), we know that this state is separable if and only if $(-1)^{f(00)}(-1)^{f(11)}c_{00}c_{11} = (-1)^{f(01)}(-1)^{f(10)}c_{01}c_{10}$. While there are various initial superpositions of 2-qubit states which satisfy this condition, we only need to consider the equal superposition (3.2) that is used in this algorithm. The situation is further simplified by noting that the mapping $(-1)^{f(a)}(-1)^{f(b)} \leftrightarrow f(a) \oplus f(b)$ is a bijection. In this case, the separability condition reduces to $f(00) \oplus f(11) = f(01) \oplus f(10)$. By looking back at Table 3.1 it is clear this condition must hold for all balanced or constant functions f for $n = 2$.

With the input prepared in the state $\left|{+}{+}\right\rangle \left|-\right\rangle$, the action of U_f can be separated and written as follows:

$$U_f \left|{+}{+}\right\rangle \left|-\right\rangle = \frac{\pm 1}{2} \left(\left|0\right\rangle + (-1)^{f(00)\oplus f(10)} \left|1\right\rangle\right) \left(\left|0\right\rangle + (-1)^{f(10)\oplus f(11)} \left|1\right\rangle\right) \left|-\right\rangle . \tag{3.5}$$

Indeed,

$$\begin{aligned}
&(-1)^{f(00)}|00\rangle + (-1)^{f(01)}|01\rangle + (-1)^{f(10)}|10\rangle + (-1)^{f(11)}|11\rangle \\
&= (-1)^{f(00)}|00\rangle + (-1)^{f(00)\oplus f(10)\oplus f(11))}|01\rangle + (-1)^{f(10)}|10\rangle + (-1)^{f(11)}|11\rangle \\
&= (-1)^{f(00)}\left(|00\rangle + (-1)^{f(10)\oplus f(11)}|01\rangle + (-1)^{f(00)\oplus f(10)}|10\rangle + (-1)^{f(00)\oplus f(11)}|11\rangle\right) \\
&= \pm\left(|0\rangle + (-1)^{f(00)\oplus f(10)}|1\rangle\right)\left(|0\rangle + (-1)^{f(10)\oplus f(11)}|1\rangle\right),
\end{aligned}$$

as desired. By applying a final 3-qubit Hadamard gate to project this state onto the computational basis we obtain:

$$\begin{aligned}
&\tfrac{\pm 1}{2}H^{\otimes 3}\left(|0\rangle + (-1)^{f(00)\oplus f(10)}|1\rangle\right)\left(|0\rangle + (-1)^{f(10)\oplus f(11)}|1\rangle\right)|-\rangle \\
&\qquad\qquad = \pm\,|f(00)\oplus f(10)\rangle \otimes |f(10)\oplus f(11)\rangle\,|1\rangle\,.
\end{aligned}$$

As in the general case we only need to measure the first two qubits to determine the nature of f: if both qubits are measured as 0, then f is constant, otherwise f is balanced.

3.2.3 Classical solutions

Because the quantum solution contains no entanglement, the problem can be de-quantised in a similar way to the $n = 1$ case, but this time using two complex numbers as the input to the black-box. We extend the black-box to operate on two complex numbers, $C_f : \mathbb{C}^2 \to \mathbb{C}^2$, and define it by analogy to U_f just as we did for the $n = 1$ case. Let z_1, z_2 be complex numbers,

$$C_f\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = C_f\begin{pmatrix} a_1 + b_1 i \\ a_2 + b_2 i \end{pmatrix} = (-1)^{f(00)}\begin{pmatrix} a_1 + (-1)^{f(00)\oplus f(10)}b_1 i \\ a_2 + (-1)^{f(10)\oplus f(11)}b_2 i \end{pmatrix}. \tag{3.6}$$

It is important to note that, just as in the quantum case where the output of the black-box was two qubits that could be independently measured, the output of C_f is two complex numbers that can be independently manipulated. This is fairly intuitive because the ability to measure specific bits (of any kind) is fundamental to computation. Note, however, that in a quantum system it is impossible to measure entangled qubits independently of each other.

The analogue to applying a Hadamard gate to each qubit in order to project it onto the computational basis is to multiply each of the complex numbers that the black-box outputs by their respective inputs (in similar fashion to that in the $n = 1$ case).

If we let $z_1 = z_2 = 1 + i$, we get the following:

$$\frac{(1+i)}{2} \times C_f \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \frac{(-1)^{f(00)}}{2} \times \begin{cases} \begin{pmatrix} (1+i)(1+i) \\ (1+i)(1+i) \end{pmatrix} = \begin{pmatrix} i \\ i \end{pmatrix} & \text{if } f \text{ is constant,} \\ \\ \begin{pmatrix} (1+i)(1-i) \\ (1+i)(1+i) \end{pmatrix} = \begin{pmatrix} 1 \\ i \end{pmatrix} \\ \begin{pmatrix} (1+i)(1+i) \\ (1+i)(1-i) \end{pmatrix} = \begin{pmatrix} i \\ 1 \end{pmatrix} & \text{if } f \text{ is balanced.} \\ \begin{pmatrix} (1+i)(1-i) \\ (1+i)(1-i) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{cases}$$

By measuring both of the resulting complex numbers, we can determine whether f is balanced or constant with certainty. If both complex numbers are imaginary then f is constant, otherwise it is balanced. In fact, just as in the $n = 1$ case, the ability to determine if the output bits are negative or positive allows us to determine the value of $f(00)$ and thus which Boolean function f is.

Because the quantum solution is separable, it is possible to write the output of the black-box as a list of two complex numbers, and hence we can find a solution equivalent to the one obtained via quantum computation. Writing the output in this form would not have been possible if the state was not separable, and finding a classical solution in this fashion would have required a list of exponentially many complex numbers compared to the number of input qubits.

As with the $n = 1$ case, an alternative classical approach can be presented using two photons. If a transformation on two qubits can be written as a transformation on each bit independently (e.g. $H \otimes H$) then the transformation is trivial to implemented classically. It only remains to show the 2-bit transformation U_f can be implemented classically on two photons. Equation 3.5 shows that for the quantum black-box with $n = 2$, U_f can be written as a product of two 1-bit gates:[2]

$$U_f^{(1)} |+\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^{f(00) \oplus f(10)} |1\rangle \right),$$
$$U_f^{(2)} |+\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^{f(11) \oplus f(10)} |1\rangle \right).$$

[2] So far we have been considering the case where U_f operates on n input qubits and one auxiliary qubit, $|-\rangle$. It has been shown (see [CKH98]) that the auxiliary qubit is not necessary if we restrict ourselves to the subspace spanned by $|-\rangle$. We have presented the algorithm with the auxiliary qubit present because it is more intuitive to think of the input-dependent phase factor being an eigenvalue of the auxiliary qubit which is kicked back. The de-quantised solutions, however, bear more resemblance to this reduced version of U_f operating only on n qubits.

Each of these are valid unitary operators, and the transformation describing the black-box may be written $U_f = U_f^{(1)} \otimes U_f^{(2)}$. This means that the operation of f can be computed by applying a 1-bit operation (implemented as wave-plates) to each photon independently, and thus a classical solution for $n = 2$ is easily found. The photons need not interact with each other at any point during the algorithm, not even inside the black-box implementation.

This classical, optical method is equivalent to both the quantum solution and the previously described classical solution. The difference is in how it is represented, bringing emphasis on the fact that for $n = 2$ the quantum solution does not take advantage of uniquely quantum behaviour and is thus classical in nature. Further, it shows that the solution can be obtained without any interaction or sharing of information between qubits.

3.3 Separability in the Deutsch-Jozsa problem

Before we consider de-quantisation of the Deutsch-Jozsa problem for $n \geq 3$, we will first examine the separability of the states used in the quantum solution more carefully, as determining if a state is separable is a key step in determining if an easy de-quantisation is possible. Conditions to determine if a n-qubit state is separable are presented in [JM03]. We will review these results, before reformulating them in a recursive manner which will allow us to apply these results much more easily to the Deutsch-Jozsa problem.

In order to determine if an arbitrary n-qubit state is separable, we must first introduce the concept of pair product invariance [JM03].

Definition 9. A state $|\psi_n\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle$ with $N = 2^n$ is *pair product invariant* if and only if $\forall k \in \{1, \dots, n\}, \forall i \in \{0, \dots, K-1\} : \alpha_i \alpha_{K-i-1} = c_k$, where each c_k is a constant and $K = 2^k$.

Pair product invariance can also be reformulated recursively as follows: let P_n be the set of all pairs (i, k) such that for any two elements $(a, k), (b, k) \in P_n$ we have $\alpha_a \alpha_{K-a-1} = \alpha_b \alpha_{K-b-1} = c_k$ if and only if $|\psi\rangle$ is pair product invariant. We can recursively write this by breaking up the iteration over all $k \in \{1, \dots, n\}$.

Definition 10 (Recursive version of Definition 9)**.** We define P_n as:

$$P_n = \left\{(i, k) \mid k \in \{2, \dots, n\}, i \in \left\{0, \dots, 2^{k-1} - 1\right\}\right\}.$$

Recursively, this becomes:

$$P_n = P_{n-1} \cup \left\{(i, n) \mid i \in \left\{0, \dots, 2^{n-1} - 1\right\}\right\}, \text{ with the base case}$$
$$P_2 = \{(0, 2), (1, 2)\}.$$

Given an n-qubit state $|\psi_n\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle$, we say P_l with $l \leq n$ satisfies the pair product invariance condition if $\forall(a,k),(b,k) \in P_l : \alpha_a \alpha_{K-a-1} = \alpha_b \alpha_{K-b-1} = c_k$ where $K = 2^k$ and c_k is constant. Then $|\psi\rangle$ is *pair product invariant* if and only if P_n satisfies the pair product invariance condition. Note that the base case, P_2, is simply the well-known condition that $\alpha_{00}\alpha_{11} = \alpha_{01}\alpha_{10}$.

To see that these two definitions are equivalent, note that we can modify Definition 9 by changing i to range from 0 to $2^{k-1}-1$ instead of $K-1$ since $\alpha_i \alpha_{K-i-1} = \alpha_{K-i-1}\alpha_i$, thus avoiding double counting. The $k = 1$ case can also be removed since it now reduces to a single term, and is hence unnecessary. The definition of P_n ensures that the quantifier $\forall(a,k),(b,k) \in P_n$ runs for $k \in \{2,\ldots,n\}$, $a,b \in \left\{0,\ldots,2^{k-1}-1\right\}$. Definitions 9 and 10 can thus clearly be seen to be equivalent.

The main theorem of relevance to us regarding pair product invariance is Theorem 11 below.

Theorem 11 (Theorem 1 in [JM03]). *If $\forall i \in \{0,\ldots,N-1\} : \alpha_i \neq 0$, a state $|\psi_n\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle$ is fully separable if and only if it is pair product invariant.*

Hence, in order to determine if a state $|\psi_n\rangle$ is separable we only need to check if it is pair product invariant. The second, recursive, definition of pair product invariance will prove more useful to formulate general results on separability.

Note that any constant function is pair product invariant as all α_i are equal and the conditions are trivially satisfied. The output of the quantum black-box for constant f can be separated as:

$$\begin{aligned}\frac{1}{2^{n/2}} U_f \sum_{x\in\mathbb{B}^n} |x\rangle |-\rangle &= \frac{1}{2^{n/2}} \sum_{x\in\mathbb{B}^n} (-1)^{f(x)} |x\rangle |-\rangle \\ &= \frac{\pm 1}{2^{n/2}} \sum_{x\in\mathbb{B}^n} |x\rangle |-\rangle \\ &= \pm |+\rangle^{\otimes n} |-\rangle .\end{aligned}$$

3.3.1 The case of $n \geq 3$

For $n = 3$ we are able to find balanced functions that are not pair product invariant and thus, by Theorem 11, entangled.

If we choose f such that

$$(f(0), f(1), f(2), f(3), f(4), f(5), f(6), f(7)) = (0,0,0,1,1,1,1,0),$$

then this f is obviously balanced. For this choice of f, $f(000) \oplus f(011) \neq f(001) \oplus f(010)$. This implies that $\alpha_{000}\alpha_{011} \neq \alpha_{001}\alpha_{010}$ and hence the output of U_f in the standard quantum algorithm is not pair product invariant and is thus entangled.

The recursive definition of pair product invariance allows us to determine exactly how many separable states exist for any given n. For a function f_{n+1} to be separable, we have the necessary condition that P_n satisfies the pair product invariance condition because $P_n \subset P_{n+1}$. Hence we see that all separable functions f_{n+1} are based on a separable function f_n, and hence satisfy $f_{n+1}(0 \circ x_n) = f_n(x_n), \forall x_n \in \mathbb{B}^n$, where $0 \circ x_n$ denotes the concatenation of x_n onto 0. This determines the action of f_{n+1} on half of the possible input strings, and we must determine the possible actions on the other half. Figure 3.2 shows this nature of pair product invariance.

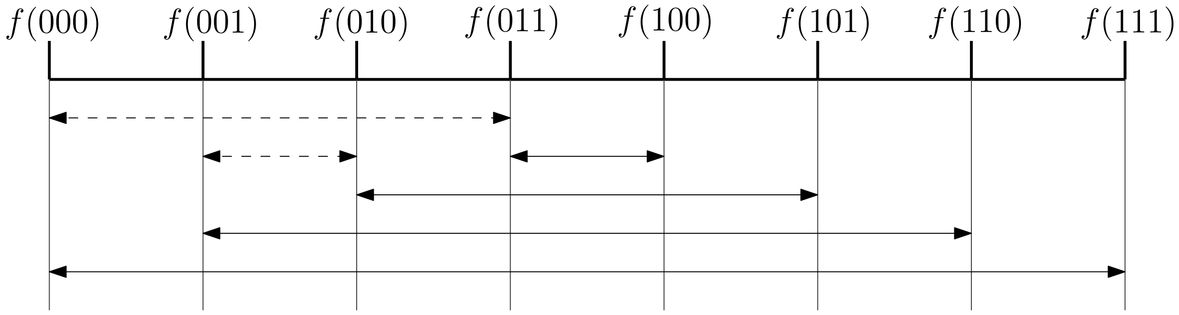


Figure 3.2: A diagram showing the recursive nature of pair product invariance. All pairs of states linked by the same type of arrow must have the same parity under addition modulo 2.

The action of f_{n+1} on the remaining $N = 2^n$ input strings is determined by the pair product invariance condition. This requires that $\alpha_{N-1}\alpha_N = \alpha_{N-2}\alpha_{N+1} = \cdots = \alpha_0\alpha_{2N-1}$. Since α_{N-1} is already determined, α_N can take on two possible values. However, once this value is chosen, all α_i for $i > N$ are uniquely determined by the pair product invariance condition. If we let a_n be the number of separable functions f_n, then f_{n+1} can have a_n possible configurations for acting on $f_{n+1}(0 \circ x_n)$, and for each of these configurations, there are two configurations for the remaining N input strings. Hence,

$$a_{n+1} = 2a_n.$$

This is a simple linear recursion with the known initial condition $a_1 = 4$. This gives us an explicit result for the number of Boolean functions f_n such that $U_{f_n} |x\rangle |-\rangle$ is separable:

$$a_n = 2^{n+1}.$$

We see that the number of separable states increases exponentially with the number of qubits being used. We also know that the number of possible functions f_n which are either balanced or constant is

$$b_n = \binom{N}{N/2} + 2 = \binom{2^n}{2^{n-1}} + 2.$$

The fraction of possible Boolean functions which can be separated is

$$\frac{a_n}{b_n} = 2^{n+1} / \left(\binom{2^n}{2^{n-1}} + 2 \right).$$

This tends towards zero extremely quickly even for small n.

The result of this observation is that even if we are promised f is valid, we can no longer be sure the output of the black-box is separable, and for $n \geq 3$ the probability that it is separable tends to zero very quickly. This means that the method of de-quantisation used for $n = 1, 2$ will not scale directly to higher n and in general yields very little information about the nature of a function. Looking at this from the view of computation with classical photons, there is no classical physical equivalent of entangled photons, as this is a purely quantum mechanical effect. However, in general it is very hard to show that no de-quantisation exists for a quantum algorithm which does not introduce exponential increase in space or time. In most cases, as earlier mentioned, a trivial method of de-quantisation is possible, but to show no better de-quantisation exists is very hard. As an example, Jozsa and Linden showed [JL03] that using the stabiliser description of quantum computation it is possible to find an efficient classical simulation of a quantum algorithm containing unbounded entanglement.

3.4 Classical and quantum black-boxes

We wish to bring attention to a slightly more subtle point in both the quantum and de-quantised oracle computational model, by illustrating an issue with the Deutsch-Jozsa algorithm. The Deutsch-Jozsa problem is the extension of a classical problem in which we are given a black-box computing the function $f : \mathbb{B}^n \to \mathbb{B}$. With such a classical oracle, it takes $O(n)$ time to prepare a query, but the classical algorithm takes $O(2^n)$ queries to determine the nature of f. Clearly in this case the running time is dominated by the black-box queries as opposed to input preparation. In the quantum algorithm we are now given a black-box U_f computing the function $g : \mathcal{H}_{2^{n+1}} \to \mathcal{H}_{2^{n+1}}$, which is the unitary quantum analogue of f. We solve this with one query to U_f, although the state preparation and measurement requires $O(n)$ operations. In this case, the state preparation dominates the running time and should be taken into account. Nonetheless, the quantum algorithm is still exponentially faster than the classical one.

Another issue however arises in the fact that the quantum algorithm takes as input a black-box operating in exponentially higher dimensions than the classical one computing f. It is thus not entirely evident that these black-boxes are equivalent, or at least directly comparable when one computes a function with more information contained within it. In the de-quantised algorithm we are similarly extending our black-box C_f to compute the function $h : \mathbb{C}^n \to \mathbb{C}^n$. It is this fact—that our classical de-quantised black-box is computing f in a higher dimension than the original classical black-box—that allows the de-quantised algorithm the ability to match the efficiency of the quantum algorithm in some cases.

This issue has barely been touched and is in need of further research, especially if we wish to work with de-quantised algorithms which operate in a different space than the trivial classical algorithms. It seems that at the heart of such efficient black-box algorithms—and perhaps quantum computation more generally—is the ability to construct a black-box which operates in dimensions which scale faster than the physical resources required as input.

3.5 General de-quantisation

While de-quantisation appears to be very hard in the Deutsch-Jozsa problem for $n \geq 3$, we can investigate the general ability to de-quantise black-box algorithms. The main task in trying to de-quantise such an algorithm is to de-quantise the black-box. Indeed, efficiently simulating a black-box algorithm requires finding a classical black-box which solves the problem without requiring exponential time or space. Using the methods previously looked at in this chapter, this step would initially require showing that both the input and output of the black-box are separable.

Firstly, we will summarise some important results due to Jozsa and Linden [JL03] about the ability to efficiently simulate standard quantum algorithms. Jozsa and Linden worked with the idea of simulating the matrix mechanical formulation of quantum algorithms, as was discussed in Section 2.3. This formalism is a little more abstract than the de-quantisation method used in this thesis, but is an effective method of showing general results. We call a qubit register $|\psi\rangle$ p-blocked if at every step of the computation no subset of more than p qubits are entangled. Their main result is Theorem 12.

Theorem 12. *Let $\mathcal{A} = (G_0, G_1, G_2, \dots)$ be a quantum algorithm computing a classical function with the property that after every unitary gate application the state $|\psi\rangle$ is p-blocked, with p independent of input size n. Then the algorithm $\mathcal{A}$ can be de-quantised using the matrix formulation.*

The proof relies on breaking up the $2^n \times 2^n$ matrices describing the gates with which the circuits in $\mathcal{A}$ are composed of into matrices operating on no more than $2p$ qubits at once.[3] Each matrix corresponding to a gate is replaced by a single matrix no larger than

[3] The $2p$ comes about rather than p in the case where a 2-qubit gate may operate on qubits from two separate entangled blocks. This approach assumes the circuit is decomposed into 1-qubit and 2-qubit gates.

$2^{2p} \times 2^{2p}$. Since p is constant, the cost of directly simulating the quantum algorithm by matrix multiplication can be reduced to a linear overhead, exponential in p rather than n.

This fixed-parameter tractability approach is mathematically equivalent to our approach, although our approach has the advantage that it retains more similarity to the quantum algorithms. It can also be easier to apply and produce simpler results than blind simulation of the matrix mechanics. Our method also more easily extends to the setting of black-box computation. The approach used by Jozsa and Linden does not directly apply to black-box computation where the nature of the black-box is unknown. It could readily be adapted by having the black-box perform a set of matrix multiplications, but such an approach is less natural. We will explore applying our de-quantisation method to arbitrary separable black-box computations since it is easier to examine and work with. Note that it is also not surprising that de-quantisation does not easily extend in the Deutsch-Jozsa problem since we have shown the entanglement grows exponentially.

The simplest case to tackle for de-quantising an arbitrary black-box algorithm is the case that both the input and output of the black-box U_f can be expressed in a separable form. In this case we can show how a simple de-quantisation in the spirit presented previously can easily be obtained. A nice feature which makes de-quantisation of a black-box simple is that we do not have to worry about considering the separability of the decomposition of the gate, as it is supplied as an arbitrarily complex unitary gate, not necessarily decomposed into gates from a universal basis (indeed, we have no way of knowing how the black-box is devised). If we knew that the input and output of U_f were separable, but had to decompose it into smaller unitary gates, we could not guarantee separability throughout the decomposed circuit representing U_f. However, the unitarity of U_f, regardless of its dimension, will allow a simple de-quantisation.

Theorem 13. *Let U_f be the unitary operator the black-box represents, and assume that for the set of input states used by the quantum algorithm $\mathcal{A}$ we wish to de-quantise, both the input and output U_f are separable. Then the black-box can be de-quantised into an efficient classical solution.*

Proof. We can write the action of U_f under these assumptions as

$$U_f \left|\psi_1\right\rangle \left|\psi_2\right\rangle \cdots \left|\psi_n\right\rangle = \left|\phi_1\right\rangle \left|\phi_2\right\rangle \cdots \left|\phi_n\right\rangle . \tag{3.7}$$

Since U_f is unitary, there must exist a unitary inverse $U_f^\dagger$, which has the effect

$$U_f^\dagger \left|\phi_1\right\rangle \left|\phi_2\right\rangle \cdots \left|\phi_n\right\rangle = \left|\psi_1\right\rangle \left|\psi_2\right\rangle \cdots \left|\psi_n\right\rangle . \tag{3.8}$$

From (3.7) we see that we each qubit undergoes the transformation $\left|\psi_i\right\rangle \rightarrow \left|\phi_i\right\rangle$, but we must show this transformation is unitary. Let us write $U_f = U_f^{(1)} \otimes U_f^{(2)} \otimes \cdots \otimes U_f^{(n)}$, where each $U_f^{(i)}$ acts on the ith qubit to implement this single-qubit transformation. Since the tensor product is distributive with respect to adjoints, we can write $U_f^\dagger = (U_f^{(1)})^\dagger \otimes (U_f^{(2)})^\dagger \otimes \cdots \otimes (U_f^{(n)})^\dagger$. Hence, we see that $U_f^{(i)} |\psi_i\rangle = |\phi_i\rangle$ and $(U_f^{(i)})^\dagger |\phi_i\rangle = |\psi_i\rangle$. Thus the single qubit transformation is indeed unitary and we can write U_f as the product of single-qubit unitary transformation above. It is then clear to see that de-quantisation of the black box is possible. Evidently, the optical method used in Section 3.2.3 can trivially be used since each $U_f^{(i)} \in \mathrm{SU}(2)$ and the black-box is then constructed out of wave-plates. We only need to prepare the n photons in the correct state, operate on them with the black-box and continue the rest of the algorithm with the transformed photons. Equivalently, a method using two-dimensional classical bits such as complex numbers could easily be used. □

This shows that, as expected, a quantum black-box algorithm can be de-quantised easily into an equivalent classical algorithm if no entanglement is present. This black-box de-quantisation could be extended to allow bounded entanglement as in Theorem 12, although the photon method would no longer be sufficient since we would have unitary transformations not in SU(2). However, higher-dimensional classical bits could easily be used. We will not explore this explicitly further, but this result on black-box simulation along with the result on standard algorithm de-quantisation allow for a powerful tool to asses quantum computations. In Chapter 4 we will apply these de-quantisation principles to the quantum Fourier transform. These de-quantisation tools provide a useful technique of developing new classical algorithms from ones which are more naturally expressed in the quantum world. If entanglement is present, a successful de-quantisation will need to handle the problem using a different representation of the process.

3.6 Summary

We have examined the ability to de-quantise the Deutsch-Jozsa problem for various values of n in order to gain a better understanding of quantum algorithms and the ability for them to give exponential improvements over classical algorithms. We have extended the method of de-quantisation presented in [Cal07] to the $n = 2$ case, and by showing separability of the quantum algorithm for $n = 2$ have obtained a similar de-quantised solution. We have shown that for $n > 2$ there exist many balanced Boolean functions f for which the output of U_f is entangled. The fraction of balanced functions for which the output is separable has been shown to approach zero very rapidly. This tells us that if we were to pick a random Boolean function which is constant or balanced, the probability of being able to learn information about the nature of the function through classical means in one black box call tends to zero.

Chapter 4

The quantum Fourier transform

The *quantum Fourier transform (QFT)* plays an important role in a large number of known algorithms for quantum computers [Gru99]. It plays a central role in Shor's algorithm for prime factorisation [Sho94] and is often thought to be at the heart of many quantum algorithms which are faster than any known classical counterpart. However, following on from recent results on classical features of the QFT algorithm [ALM07, Bro07, GN96, YS07] we will argue that the QFT algorithm itself is classical in nature. The usefulness of the QFT algorithm is a result of the linearity of the unitary transformations fundamental to quantum mechanics and computation. It is this linearity which makes the QFT such a useful tool, rather than the nature of the QFT itself.

To illustrate this, we will show how the QFT can be de-quantised into a simpler classical algorithm in many situations. In Section 4.1 we overview the QFT and present it as a compact algorithm in order to move away from the restrictions imposed by the circuit model. In Section 4.2 we de-quantise the QFT acting on a basis-state into a classical algorithm more efficient than the quantum one. We then discuss the possibility of extending this de-quantisation to more general, separable inputs. In Section 4.3 we discuss why de-quantisation of the QFT is possible, and note some common misunderstandings about the QFT which contribute to this.

4.1 Background

The *discrete Fourier transform (DFT)* on which the QFT is based is a transformation on a q-dimensional complex vector $\chi = (f(0), f(1), \ldots, f(q-1))$ into its Fourier representation $\hat{\chi} = (\hat{f}(0), \hat{f}(1), \ldots, \hat{f}(q-1))$ [Gru99]:

$$\hat{f}(c) = \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} e^{2\pi i ac/q} f(a), \tag{4.1}$$

for $c \in \{0, 1, \dots, q-1\}$. The QFT is similarly defined so that the transformation acts on a state vector in q-dimensional Hilbert space, $\mathcal{H}_q$. In quantum computation we work with a state vector defining a register comprising of n 2-state qubits, so we will only consider the case that $q = 2^n$ from this point onwards. This means that the QFT, denoted F_q, acts on the 2^n amplitudes of a particular n-qubit state, i.e.

$$\sum_{a=0}^{2^n-1} f(a) |a\rangle \xrightarrow{F_{2^n}} \sum_{c=0}^{2^n-1} \hat{f}(c) |c\rangle . \tag{4.2}$$

As a result of the linearity of quantum mechanics, in order to compute the QFT we only need to design an algorithm to transform a single component of the state vector. This is because an arbitrary state $|\psi\rangle = \sum_{a=0}^{2^n-1} f(a) |a\rangle$ transforms as:

$$F_{2^n} |\psi\rangle = \sum_{a=0}^{2^n-1} f(a) F_{2^n} |a\rangle = \frac{1}{\sqrt{2^n}} \sum_{a=0}^{2^n-1} \sum_{c=0}^{2^n-1} e^{2\pi i ac/2^n} f(a) |c\rangle = \sum_{c=0}^{2^n-1} \hat{f}(c) |c\rangle .$$

Hence we arrive at the standard definition of the QFT as the mapping [CEMM97]

$$|a\rangle \xrightarrow{F_{2^n}} \frac{1}{\sqrt{2^n}} \sum_{c=0}^{2^n-1} e^{2\pi i ac/2^n} |c\rangle , \tag{4.3}$$

with $a \in \{0, 1, \dots, 2^n - 1\}$. Keeping in mind that we are dealing with registers composing of qubits, we can decompose a (and similarly c) into its binary representation so that $a = 2^{n-1}a_1 + 2^{n-2}a_2 + \dots + 2^1 a_{n-1} + 2^0 a_n$ and $|a\rangle = |a_1 a_2 \dots a_n\rangle$. By denoting $a = a_1 a_2 \dots a_n$ and $a/2^n = 0.a_1 a_2 \dots a_n$ we observe that

$$\begin{aligned} e^{2\pi i ac/2^n} &= e^{2\pi i a(2^{n-1}c_1 + 2^{n-2}c_2 + \dots + 2^0 c_n)2^{-n}} \\ &= e^{2\pi i (a_1 a_2 \dots a_n)2^{-1}c_1} e^{2\pi i (a_1 a_2 \dots a_n)2^{-2}c_2} \dots e^{2\pi i (a_1 a_2 \dots a_n)2^{-n}c_n} \\ &= e^{2\pi i (a_1 \dots a_{n-1}.a_n)c_1} e^{2\pi i (a_1 \dots a_{n-2}.a_{n-1}a_n)c_2} \dots e^{2\pi i (0.a_1 a_2 \dots a_n)c_n} . \end{aligned} \tag{4.4}$$

Noting that for any decimal $x.y$ we have $e^{2\pi i (x.y)} = (e^{2\pi i})^x e^{2\pi i (0.y)} = e^{2\pi i (0.y)}$, we see that only the fractional part of $(a_1 \dots a_{n-j}.a_{n-j+1} \dots a_n)c_j$ is of any significance in the exponent of (4.4). Hence, we find

$$e^{2\pi i ac/2^n} |c_1 \cdots c_n\rangle = e^{2\pi i (0.a_n)c_1} |c_1\rangle\, e^{2\pi i (0.a_{n-1}a_n)c_2} |c_2\rangle \cdots e^{2\pi i (0.a_1 a_2 \dots a_n)c_n} |c_n\rangle .$$

Using this decomposition we can write (4.3) as a product state of individual qubits,

$$\frac{1}{\sqrt{2^n}} \sum_{c=0}^{2^n-1} e^{2\pi i ac/2^n} |c\rangle = \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i (0.a_n)} |1\rangle) \cdots (|0\rangle + e^{2\pi i (0.a_1 a_2 \dots a_n)} |1\rangle). \tag{4.5}$$

The quantum algorithm to implement the QFT follows directly from this factorisation. The circuit for the algorithm is shown in Figure 4.1. The algorithm can be written explicitly as follows [CEMM97]:

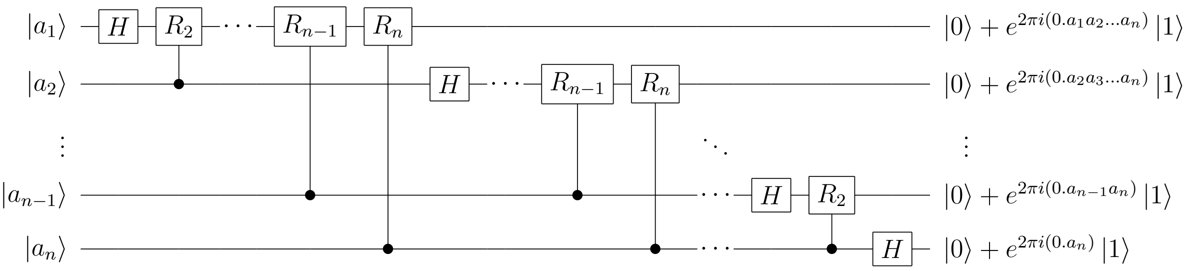

$|a_1\rangle$
$|a_2\rangle$
$\vdots$
$|a_{n-1}\rangle$
$|a_n\rangle$
H
R_2
R_{n-1}
R_n
H
R_{n-1}
R_n
H
R_2
H
$|0\rangle + e^{2\pi i(0.a_1a_2\ldots a_n)}|1\rangle$
$|0\rangle + e^{2\pi i(0.a_2a_3\ldots a_n)}|1\rangle$
$\vdots$
$|0\rangle + e^{2\pi i(0.a_{n-1}a_n)}|1\rangle$
$|0\rangle + e^{2\pi i(0.a_n)}|1\rangle$

Figure 4.1: The standard quantum circuit for the QFT. The output normalisation factors of $1/\sqrt{2}$ and swap gates to reverse qubit order are omitted.

Quantum Fourier transform

Input: The state $|a\rangle = |a_1\rangle |a_2\rangle \cdots |a_n\rangle$.

Output: The transformed state $\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i(0.a_n)}|1\rangle)\cdots(|0\rangle + e^{2\pi i(0.a_1a_2\ldots a_n)}|1\rangle)$.

```
1. For $j = 1$ to $n$, transform qubit $|a_j\rangle$ as follows:
2.     $|a_j\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.a_j)}|1\rangle)$.
3.     For $k = j + 1$ to $n$:
4.         $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.a_j\ldots a_{k-1})}|1\rangle) \xrightarrow{R_k} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.a_j\ldots a_{k-1}a_k)}|1\rangle)$ where $R_k$ is the
           unitary $k$-controlled phase shift:
```

$$R_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i/2^k} \end{pmatrix}.$$

```
5.     End For.
6.     Reverse the order of the qubits.
7. End For.
```

Clearly this produces the state (4.5) and requires $O(n^2)$ unitary R_k or H gates to run.

There are a few important things about the QFT which should be noted. While both the DFT and the QFT act on vectors in a complex vector space, the DFT acts on a classical, mathematical vector whereas the QFT acts on a physical state which we mathematically represent by a vector in $\mathcal{H}_{2^n}$. The subtle difference here is that with the classical DFT, we can read the values of all 2^n Fourier coefficients $\hat{f}(c)$ by simple inspection of the transformed vector. With the QFT, the resulting state (4.2) embeds all 2^n coefficients as amplitudes for the 2^n states of n qubits. However, the collapse of the superposition upon measurement means that it is impossible to measure the amplitudes of a quantum state without an ensemble of such states to make a statistical approximation of the amplitudes from, and detecting phase differences between states is even more difficult. Hence, the quantum state (4.2) contains all the information of the classically transformed vector but it is inaccessible to measurement. The main use of the QFT is then as a tool to extract information embedded in the relative amplitudes of states as opposed to determining the coefficients themselves.

Another result of this is that the efficiency of the QFT ($O(n^2)$ as opposed to the DFT which is $O(n2^n)$) is in some sense due to the ability to perform the transformation and utilise the information in the phases without measuring the state. Evidently, any algorithm requiring measurement needs exponential time (there are 2^n coefficients to measure), so even if quantum mechanics would allow us to measure the Fourier coefficients in state (4.2), doing so would take $O(n2^n)$ time: 2^n coefficients, n qubits each. Making use of this embedded information while avoiding measurement is certainly an important part of the fine art of developing algorithms in quantum computing.

4.2 De-quantisation investigation

Having presented the QFT, there are some issues to be brought to light. The decomposition of the transformed state (4.3) shown in Equation 4.5 is evidently not entangled. Just as for the Deutsch-Jozsa problem, the separability of this state would lead us to believe that the QFT producing it could be simulated efficiently in a classical manner, and there are certainly a few results on this observation.

It was realised shortly after the discovery of Shor's algorithm that the QFT could be computed in a semiclassical manner using classical signals resulting from quantum measurements to perform the QFT on a state using classical logic and 1-qubit gates (instead of the usual 2-qubit controlled-phase-shifts) [GN96]. This method gives the same resulting probability distribution as the quantum algorithm, but destroys the state's superposition in the process. As a result, this is only useful in an algorithm in which the QFT is the final operation before measurement. Shor's algorithm happens to be of exactly this nature, but this is only an initial step towards true classical simulation.

Much more recently, classical simulations of the QFT have been studied from the viewpoint of simulating the circuit in Figure 4.1 by using the bubble-width of the quantum circuit [ALM07] and the tensor contraction model [YS07]. The use of the bubble-width produces a classical circuit with the same effect as the quantum circuit. The inability of classical circuits to operate on anything other than trivial non-superposition inputs takes away a lot of the usefulness of this other than its illustrative purpose. The tensor-contraction model still focuses on circuits, but allows simulating the QFT on a separable input. The use of circuits, however, while illustrative, seems to overcomplicate matters significantly when it comes to classical simulation. We will explore simulations of the QFT in the form of de-quantisations similar to those seen in Chapter 3.

A minor change to our de-quantised bits is needed because the amplitudes which we need to represent in the QFT algorithm are complex-valued. We cannot use complex numbers as our two-dimensional bits, but there is no problem with simply using a two-valued vectors as our classical bits, so we will employ this procedure.

4.2.1 Basis-state de-quantisation

The de-quantisation for a basis-state needs only to simulate the transformation defined in Equation 4.3. As a result of the decomposition in Equation 4.5, the effect of the QFT on the jth qubit is easily seen to be

$$|a_j\rangle \xrightarrow{F_{2^n}} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.a_{n-j+1}\ldots a_n)}|1\rangle). \tag{4.6}$$

The difficulty in implementing this in a quantum computer is that the phase of a qubit needs to be altered depending on the values of the other qubits without altering them—that is why it is not helpful to express the quantum algorithm as we have done in Equation 4.6—and the circuit of controlled-phase-shifts is required to implement this. The information is spread over the input qubits and must be obtained without measurement. In the classical case there are no such restrictions on measurement, so de-quantisation should only require directly implementing Equation 4.6. However, evaluating the complex phase for each of the n qubits takes $O(n)$ time, leading to a $O(n^2)$ procedure. This can be reduced to $O(n)$ by calculating each phase dependent on the previous one. To do so, let ω_j be the jth phase factor and note the following:

$$\begin{aligned}\omega_j &= e^{2\pi i(0.a_{n-j+1}\ldots a_n)}\\ &= e^{2\pi i(0.a_{n-j+1})}e^{2\pi i(0.a_{n-j+2}\ldots a_n)/2}\\ &= (-1)^{a_{n-j+1}}\sqrt{\omega_{j-1}},\end{aligned}$$

and

$$\omega_1 = e^{2\pi i(0.a_n)} = (-1)^{a_n}.$$

Note that by the square-root we mean the principal root. The square-root of a complex number such as ω_j can be calculated independently of n. Specifically, if we have $s+ti = \sqrt{b+di}$ with the further requirement that for a root of unity $\sqrt{b^2+d^2} = 1$, then [BC36]:

$$\begin{aligned}s &= \frac{1}{\sqrt{2}}\sqrt{1+b},\\ t &= \frac{\operatorname{sgn}(d)}{\sqrt{2}}\sqrt{1-b},\end{aligned}$$

where $\operatorname{sgn}(d) = d/|d|$ is the sign of d. The efficient de-quantised algorithm is then the following:

Basis-state de-quantised QFT

Input: The binary string $a = a_1a_2 \dots a_n$.

Output: The n transformed two-component complex vectors $\mathbf{b_1 b_2} \dots \mathbf{b_n}$.

```
1. Let $\omega = 1$
2. For $j = 1$ to $n$:
3.     Set $\omega = (-1)^{a_{n-j+1}}\sqrt{\omega}$
4.     Set $\mathbf{b_j} = \frac{1}{\sqrt{2}} \times \begin{pmatrix} 1 \\ \omega \end{pmatrix}$
5. End For.
```

This is mathematically identical to the definition of the QFT given in Equations 4.3 and 4.5, but is computed classically in time $O(n)$. This achieves exactly the same result as finding a classical circuit simulating the quantum circuit, such as that given by Aharanov et al. [ALM07], but is much simpler. This is primarily because the quantum circuit is constructed subject to the requirement of computing the QFT without any intermediate measurements. As a result, the quantum algorithm corresponding to the circuit must conform to this too, making it more complex than an equivalent classical algorithm need be.

A classical algorithm has the further advantage over the quantum algorithm acting on a basis-state that measurement of the resulting state can be performed at will, and any required information is easily accessible. In the quantum algorithm only a single state can be measured, and no information about the amplitudes (and thus the Fourier coefficients) can be determined from a single QFT application. While this classical algorithm is no faster than the well known FFT for calculating all the coefficients, it may be beneficial if only some coefficients are required.

The ability to de-quantise the QFT acting on a basis-state is not particularly surprising. This is equivalent to the classical DFT acting on a vector with only one non-zero component, producing a fairly trivial and easily computed output. However, this highlights a little more deeply some common misconceptions about the QFT. Because of the unitary evolution of quantum mechanics, the action of the QFT on a basis-state shown in (4.3) is often taken as the definition of the QFT. While this suffices as the definition of the quantum algorithm, it is important not to forget that the actual definition of the QFT is that given in Equation 4.2. When considering classical simulations of the QFT this is even more important, as the action of the QFT on a basis-state and the corresponding circuit no longer immediately allow us to compute the complete QFT. Indeed it would take 2^n iterations of a classical algorithm simulating the basis-state behaviour to compute the complete QFT, a method which is slower than the fast Fourier transform (FFT).

4.2.2 Extension to general input states

Here we briefly consider the possibility of extending the de-quantisation to work on a wider range of input states, resulting in a less trivial de-quantisation. If the input state is entangled then it is clear that the de-quantisation is not easily extended, as the method used for the basis-state algorithm relied on the separability of the input. In such a situation, any de-quantisation attempt would need to involve a different method and work directly from the QFT definition, (4.2).

However, it is not immediately clear that the basis-state de-quantisation, which is based on Equation 4.3, could not be extended to work on arbitrary separable input states. This idea is strengthened by the fact that we used the single-qubit formula (4.6) to perform the basis-state de-quantisation. However, this implicitly relies on the other qubits in the input state having a definite value, but in the separable input case this is not necessarily the case. In Example 14 we give an example to show that the QFT can entangle separated input states, and hence a general separable input state de-quantisation is not possible in the same way as the basis-state de-quantisation was.

Example 14. Consider the input state:

$$\begin{aligned} |\phi\rangle &= |0\rangle \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \\ &= \frac{1}{\sqrt{2}} \left(|00\rangle + |01\rangle \right). \end{aligned} \tag{4.7}$$

Using the basis-state QFT definition, (4.3), we see that the two basis-states in this superposition transform as:

$$\begin{aligned} |00\rangle &\xrightarrow{F_4} \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle, \\ |01\rangle &\xrightarrow{F_4} \frac{1}{2}|00\rangle + \frac{i}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{i}{2}|11\rangle. \end{aligned}$$

Hence, we see that (4.7) transforms as:

$$|\phi\rangle \xrightarrow{F_4} \frac{1}{\sqrt{2}} \left(|00\rangle + \frac{1+i}{2}|01\rangle + \frac{1-i}{2}|11\rangle \right).$$

Using an extension of Theorem 11 for the case where the amplitudes of some states are zero [JM03], this is seen to be entangled.

There do also exist inputs which the QFT does not entangle. An example of this is the input state $|\phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$. In this case the QFT merely computes the identity function. For such input states which are not entangled by the QFT, it might be possible to develop a de-quantised algorithm. The major step in developing such a de-quantisation would be to determine conditions for when the input state will not be entangled, similar to our separability analysis for the Deutsch-Jozsa problem in Section 3.3. Indeed, the notion of pair product invariance is also extended by [JM03] to states in which some amplitudes are zero (recall in the Deutsch-Jozsa problem all amplitudes were ± 1). However, the conditions are more complicated, and such analysis is left to future research, but is certainly worth exploring as a more general de-quantisation is of interest both theoretically and practically.

4.3 Discussion

The basis-state de-quantisation computes a Fourier transform on a single state, while the QFT algorithm computes the Fourier transform on arbitrary separable or entangled input states. It is the fact that the separability of the QFT algorithm may not hold even for separable input states which restricts the de-quantisation of the QFT. In the most general case a classical simulation would need to work directly from Equation 4.2, although there may be cases in which separability could be guaranteed. This highlights the important difference between the quantum Fourier transform and the quantum algorithm computing it. The linearity of quantum mechanics ensures that the algorithm designed only for a basis-state suffices to compute the complete transform. When we depart from quantum mechanics this is no longer the case, and the de-quantised algorithm does not suffice to compute the complete QFT.

Further research should be conducted into finding if there exist conditions for which the QFT could be de-quantised to work on separable input states. Such conditions could make the de-quantisation less trivial and much more useful. Such a de-quantisation would have potential as a subroutine in de-quantisations of other quantum algorithms where the QFT is used, and such separability conditions can be guaranteed. Unfortunately Shor's algorithm has unbounded entanglement as the input to the QFT [JL03], so any de-quantisation of the QFT algorithm will likely be of little use here.

Another issue worth noting is that we must be careful to consider the complexity involved in manipulating the complex amplitudes in a state-vector when performing de-quantisation. While it did not contribute to the complexity of the de-quantised algorithms presented in this chapter, attention had to be paid to make sure this was the case, as this would not have been so if we directly implemented the obvious algorithm. In quantum computation, however, the amplitudes are just our representation of a property of physical states. It is these physical states, rather than the amplitudes, which are altered by unitary transformations, and as a result we observe the amplitudes changing. This reiterates the need for care when de-quantising, as the amplitudes have no *a priori* reason to be easily calculated, or computable at all for that matter.

4.4 Summary

We have shown that the quantum algorithm computing the QFT can be de-quantised into a classical algorithm which is just as efficient and in many senses simpler than the quantum algorithm, primarily because the need to avoid measurement of the system is no longer present. However, the de-quantised algorithm only acts on a basis-state, and as a result it fails to compute the QFT on arbitrary inputs because the separable representation of the QFT is not valid in these cases. This difference between quantum and classical algorithms highlights the common treatment of the quantum Fourier transform and the corresponding algorithm as the same thing. Instead, it is only the linearity of quantum mechanics which ensures they have the same effect.

Chapter 5

Conclusion

5.1 Future Work

The work in this thesis provides a grounding for the de-quantisation of quantum algorithms, and many more questions are raised in the process. There are certainly many algorithms that this de-quantisation procedure can be applied to, and many other de-quantisation procedures waiting to be developed. Quantum random walk (QRW) algorithms [SKW03], although a slightly different paradigm, look a promising target of possible de-quantisation. QRW algorithms appear to have bounded entanglement and are capable of matching algorithms such as Grover's search algorithm [Gro97] which in the standard formulation contains entanglement, making direct de-quantisation difficult. Indeed, finding other algorithms which can be de-quantised in the way done in this thesis would deepen our understanding of quantum computation, and potentially provide important new classical algorithms.

Exploring other possible de-quantisation techniques is also an important step. As noted in Section 3.3, developing a set of conditions which allows de-quantisation helps us to narrow down the source of benefit in quantum computation, and hence allows us to develop better quantum algorithms. While developing complete conditions would be an extremely difficult task, any further de-quantisation conditions would be of much benefit to both classical and quantum computation.

The issue of black-box complexity, as briefly discussed in Section 3.4, is a topic which should be explored further. To properly compare and develop black-box algorithms in quantum computation we need a better understanding of the complexity of unitary black-boxes in comparison to classical ones. This is also important if we are to fully understand the de-quantisation of such algorithms.

5.2 Summary

In this thesis we developed the idea of de-quantisation of quantum algorithms. We de-quantised the Deutsch-Jozsa problem up to the $n = 2$ case, and analysed the ability to de-quantise beyond this. We showed that the entanglement grows exponentially in the Deutsch-Jozsa problem, and this prevents the technique of de-quantising with classical complex numbers from extending. However, this does not mean de-quantisation is not possible, but that other methods need to be developed and the black-box complexity of quantum algorithms needs to be better understood.

We applied the same de-quantisation technique to the quantum Fourier transform in its standard form as a basis-state quantum algorithm, producing a de-quantised algorithm more efficient than the quantum algorithm. We used this to highlight key misconceptions about the nature of the QFT, and showed that it is the linearity of quantum mechanics which allows the QFT algorithm to compute an arbitrary Fourier transform, but that the algorithm itself is classical. The de-quantised version is simpler than the quantum algorithm and if it could be extended to act on separable input states under certain restrictions, it has potentially strong implications.

The procedure of de-quantisation has been shown to be applicable in many situations, and is an important technique which needs to be explored further if we wish to truly understand the benefit of quantum computation. Such understanding could lead to development of better quantum and classical algorithms.

Bibliography

[ALM07] D. Aharonov, Z. Landau, and J. Makowsky. The quantum FFT can be classically simulated. *arXiv:quant-ph/0611156v2*, Jan 2007. [29, 32, 34]

[Arv01] Arvind. Quantum entanglement and quantum computational algorithms. *Pramana - Journal of Physics*, 56(2 & 3):357–365, Jan 2001. [16]

[BB94] A. Berthiaume and G. Brassard. Oracle quantum computing. *Journal of Modern Optics*, 41(12):2521–2535, Dec. 1994. [11]

[BBC$^+$95] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995. [8]

[BC36] S. Barnard and J. M. Child. *Higher Algebra.* Macmillan, London, 1936. [33]

[Bor26] M. Born. Zur Quantenmechanik der Strossvorgänge. *Zeitschrift für Physik*, 37:863–867, 1926. English translation by J. A. Wheeler and W. H. Zurek, in *Quantum Theory and Measurement*, chapter I.2. Princeton University Press, 1983. [5]

[Bro07] D. E. Browne. Efficient classical simulation of the quantum Fourier transform. *New Journal of Physics*, 9(5):146, May 2007. [29]

[Cal07] C. S. Calude. De-quantizing the solution of Deutsch's problem. *International Journal of Quantum Information*, 5(3):409–415, Jun 2007. [11, 15, 16, 28]

[CEMM97] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A*, 1998(454):339–354, Jan 1997. [13, 17, 30]

[CKH98] D. Collins, K. W. Kim, and W. C. Holton. Deutsch-Jozsa algorithm as a test of quantum computation. *Physical Review A*, 58(3):R1633–R1636, Sep 1998. [21]

[Deu85] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. Series A*, 400:97–117, Jan 1985. [3, 13]

[DJ92] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A*, 439:553–558, Jan 1992. [16]

[EJ98] A. Ekert and R. Jozsa. Quantum algorithms: Entanglement enhanced information processing. *Philosophical Transactions of the Royal Society A*, 356(1743):1769–1782, 1998. [11]

[Fey82] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1982. [3]

[GN96] R. Griffiths and C. Niu. Semiclassical Fourier transform for quantum computation. *Physical Review Letters*, 76(17):3228–3231, Jan 1996. [29, 32]

[Gro97] L. K. Grover. Quantum computers can search arbitrarily large databases by a single query. *Physical Review Letters*, 79(23):4709–4712, 1997. [39]

[Gru99] J. Gruska. *Quantum Computing.* McGraw-Hill International Limited, 1999. [3, 5, 6, 29]

[JL03] R. Jozsa and N. Linden. On the role of entanglement in quantum-computational speed-up. *Proceedings: Mathematical, Physical and Engineering Sciences*, 459(2036):2011–2032, Jan 2003. [25, 26, 36]

[JM03] P. Jorrand and M. Mhalla. Separability of pure n-qubit states: two characterizations. *International Journal of Foundations of Computer Science*, 14(5):797–814, 2003. [22, 23, 35, 36]

[KSV02] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation.* American Mathematical Society, 2002. [7, 8, 10, 11]

[Mer07] N. D. Mermin. *Quantum Computer Science.* Cambridge University Press, 2007. [6, 15, 18]

[NC00] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000. [6, 7, 8, 9]

[Sak94] J. J. Sakurai. *Modern Quantum Mechanics.* Addison-Wesley Publishing Company, revised edition, 1994. [6, 7]

[Sho94] P. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In Shafi Goldwasser, editor, *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE Computer Society Press, 1994. [4, 29]

[Sip06] M. Sipser. *Introduction to the Theory of Computation.* Thomson Course Technology, second edition, 2006. [10]

[SKW03] N. Shenvi, J. Kemp, and K. B. Whaley. Quantum random-walk search algorithm. *Physical Review A*, 67(5):052307, 2003. [39]

[YS07] N. Yoran and A. J. Short. Efficient classical simulation of the approximate quantum Fourier transform. *Physical Review A*, 76(4):042321, Oct 2007. [29, 32]