Using Generalized Quantum Fourier Transforms in Quantum Phase Estimation Algorithms

by

Donny Cheung

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2003

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

The quantum Fourier transform (QFT) is the cornerstone of most of the important and significant algorithms for quantum computers currently known. However, the physical implementation of a QFT requires the use of quantum gates with a precision that may be difficult or even impossible to achieve due to both physical imprecision and the effects of decoherence. Fortunately, the gates requiring the most precision are also the gates that have the least effect on the final output, so this limitation in the QFT may be overcome by using the approximate quantum Fourier transform (AQFT), which ignores the smaller gates in the hope that less accurate results are compensated for by being better-suited to actual physical implementation.

This thesis presents a unified framework for generalized AQFT-based phase estimation algorithms, gives a thorough analysis of these algorithms, and explores issues surrounding the applicability of these algorithms as components of larger quantum algorithms.

Acknowledgements

I am indebted to the many people who have helped me during the process of writing this thesis, first and foremost to my supervisor, Michele Mosca, who first introduced me to the field to quantum computing. My thesis readers, Raymond Laflamme and Paul Schellenberg have provided me with indispensible feedback on all sections of thesis. The content of this thesis has also benefitted from productive discussions with Phillip Kaye, Lawrence Ioannou and Christof Zalka. Finally, I would like to thank Jared Penner for applying his technical proofreading skills to this thesis.

Contents

List of Figures

Preface

This thesis attempts to give a mostly self-contained development of the quantum Fourier transform, and the generalizations which are its main focus, from the beginning principles of quantum computation. Some rudimentary familiarity with algebra and computing is assumed.

We begin in Chapter 2 with a summary of basic notation and the quantum circuit model, which is the model of quantum computation used throughout this thesis. Chapter 3 follows with an overview of the QFT and its role in the phase estimation algorithm. From there, we develop the generalized AQFT by first looking at the AQFT in Chapter 4, and then by exploring and analyzing generalizations in Chapter 5. Finally, Chapter 6 contains a survey and discussion of significant applications of the phase estimation algorithm, then explores how using the generalized AQFT circuit impacts these applications.

This thesis contains several results which, to the best of my knowledge, have not appeared before. First, there is the development and analysis of the generalized AQFT circuit in Chapter 5 which is the title topic of my thesis. In Chapter 4, I give a classical post-processing algorithm for finding the maximum likelihood estimate in a reduced version of Kitaev's phase estimation algorithm, and a heuristic argument as to why this proposed algorithm may be efficient. Also in this chapter, I give

a new and more detailed analysis of Coppersmith's approximate quantum Fourier transform, with a new lower bound on the probability of success which makes the AQFT a much more viable and efficient option as a replacement of the QFT. Finally, the generalizations to the phase estimation algorithms developed in this thesis are analyzed with an emphasis on obtaining maximum likelihood estimates.

In addition to these results, Chapter 3 also contains a complete proof of the $\frac{8}{\pi^2}$ lower bound for the probability of having the QFT-based phase estimation algorithm return one of the two nearest estimates of the correct phase. Such a proof, to the best of my knowledge, has not yet appeared in the literature.

Chapter 1

Introduction

Quantum information processing is a relatively young scientific field which investigates the possibility of exploiting the quantum-mechanical properties of nature in performing computations. One of the most interesting aspects of quantum information processing is that this model of computation allows us to create algorithms which are far superior in asymptotic running time to any "classical" counterparts known to date which solve the same problems. The quantum algorithm known as the *quantum Fourier transform* (QFT) is the cornerstone of many of these algorithms, including the surprising quantum polynomial-time integer factorization algorithm discovered by Peter Shor.

1.1 The Quantum Fourier Transform

The QFT is a quantum algorithm which solves the classical problem of calculating the discrete Fourier transform (DFT) of a complex-valued vector of length $N = 2^n$, where n is a non-negative integer. The main difference between the QFT and the

classical DFT algorithms is that while classical algorithms take an entire complex-valued vector of length N and give us the entire DFT in the form of another vector of length N, the QFT takes a quantum state in which the initial data have been encoded into probability amplitudes and alters the amplitudes corresponding to the DFT. Unfortunately, we are unable to directly access the individual values in the final quantum state, making the QFT unsuitable for applications such as digital signal processing. However, the strength of the QFT comes from the fact that this operation is achieved in polynomial-time with respect to n giving an exponential improvement on the fastest known classical DFT algorithms. Furthermore, the quantum state prepared by the QFT is still very useful in a wide range of computational tasks.

Most of the applications of the QFT in quantum algorithms are instances of the *Phase Estimation* problem, which can be solved efficiently using an algorithm that incorporates the QFT. Given a unitary operator U and a quantum register that contains an eigenvector of U, the Phase Estimation Problem asks for the eigenvalue that corresponds to the given eigenvector. By using various different constructions for the unitary operator U, an efficient solution to the Phase Estimation problem leads to efficient solutions for many problems of interest, including integer factorization and the discrete logarithm problem, for which there are no known efficient solutions using classical computation.

1.2 Variations of the QFT

The main concern with the efficiency of the QFT algorithm is its use of a particular class of quantum gates called controlled phase-shift gates. As the number of qubits in the input, n, increases, the QFT algorithm requires exponentially smaller phase

shifts, which may be increasingly difficult or even infeasible to implement physically. Fortunately, the phase shift gates requiring the most precision are also the phase shift gates which have the least effect on the output. As a result, the idea of an *approximate quantum Fourier transform* (AQFT) was suggested, in which the smallest phase shifts would simply be ignored. Since the overall effect on the output quantum state is small, the AQFT is still suitable in solving the phase estimation algorithm, but there is a clear trade-off in accuracy. The idea behind the AQFT may be extended further by exploiting a property of the phase estimation algorithm. In the phase estimation algorithm, different parts of the computation may be separated into individual modules and recombined in different ways with different parameters, then analyzed to obtain the optimal combination of parameters for a given application or for a particular physical realization of a quantum computer.

Chapter 2

Quantum Circuits

As with classical computers, it is fortunate that an intimate knowledge of the physical details within a quantum computer is not necessary to discuss quantum algorithms, especially since a reliable, functional quantum computer has yet to be created. Instead, it is possible to describe the action of a quantum computer as a quantum circuit, consisting of a network of logical quantum gates operating on a collection of logical quantum bits, or *qubits*. This chapter introduces some of the theoretical background behind the quantum circuit model of computation, which is the model used in the discussion of quantum algorithms such as the QFT and AQFT in subsequent chapters.

2.1 Qubits and Quantum Gates

The qubit is the basic unit of information in the quantum circuit model. In classical binary logic, information is organized into bits, each of which is in one of two discrete states, 0 or 1. Likewise, a qubit may be in the two corresponding quantum states

$|0\rangle$ or $|1\rangle$, where Dirac's *ket* ($|\cdot\rangle$) notation indicates a quantum state. However, a qubit is also allowed to be in a *superposition* of the two logical states $|0\rangle$ and $|1\rangle$, which is simply a linear combination of these two states:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,,$$

where $|\psi\rangle$ denotes the state of the qubit, and $\alpha, \beta \in \mathbb{C}$ are *amplitudes*, subject to the identity $|\alpha|^2 + |\beta|^2 = 1$. Although a qubit is permitted to be in any superposition of the two *basis* states, $|0\rangle$ and $|1\rangle$, if the qubit is observed by measurement, the result of this measurement will be one of the two basis states: $|0\rangle$ with probability $|\alpha|^2$, or $|1\rangle$ with probability $|\beta|^2$. Furthermore, the act of measurement destroys the superposition so that the quantum state of the qubit becomes one of the two basis states. Although there is no way to retrieve the coefficients α and β directly from a qubit, measuring a number of qubits which are previously known to be in the same quantum state will give progressively better estimates of the values of α and β for the original state prior to measurement.

Qubit states may also be considered as vectors in a finite-dimensional inner product space over $\mathbb{C}$, otherwise known as a finite-dimensional Hilbert space. The dual of a vector $|\psi\rangle$ is defined as the complex conjugate transpose of that vector, and notated with Dirac's *bra* ($\langle\cdot|$) notation: $\langle\psi| \equiv (|\psi\rangle)^\dagger$, where $A^\dagger$ denotes the complex conjugate transpose of A.[1] The inner product of two vectors $|\psi_1\rangle$ and $|\psi_2\rangle$ is simply $\langle\psi_1|\,(|\psi_2\rangle)$, which may also be notated $\langle\psi_1|\psi_2\rangle$, which is Dirac's *bra-ket* notation. Using the Euclidean norm,

$$\||\psi\rangle\| \equiv \sqrt{\langle\psi|\psi\rangle},$$

[1] The notation for the complex conjugate transpose of a matrix is inconsistent between mathematics, computer science and physics. Since the bra-ket notation used in this thesis for quantum states is the notation used in physics, we will adopt the dagger (†) notation used in physics.

we note that the set of all possible qubit states is equivalent to the set of vectors in $\mathbb{C}^2$ with norm 1.

Unlike classical physics, quantum mechanics allows collections of multiple qubits to be *entangled* with each other, so that the quantum states of the entire collection of qubits may be in superposition in such a way that the individual qubits may not be considered as having individual qubit states. For example, a two-qubit system, or *register*, $Q_1Q_2 = |\psi_1\rangle\,|\psi_2\rangle$, may be found in one of four basis states: $|0\rangle|0\rangle$, $|0\rangle|1\rangle$, $|1\rangle|0\rangle$, or $|1\rangle|1\rangle$, which correspond to the four combinations of basis states obtained when each qubit is considered separately. However, it may also be found in a superposition of these states, such as

$$\frac{1}{\sqrt{2}}\,|0\rangle|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle|1\rangle\,,$$

which cannot be expressed as a combination of two separate qubit states, $Q_1 = |\psi_1\rangle$ and $Q_2 = |\psi_2\rangle$. This particular two-qubit state is known as the *Bell state*.

Mathematically, a system of n qubits may be considered as a tensor product of the n individual qubits. Since each individual qubit state is an element of a Hilbert space, $\mathbb{C}^2$, the state associated with any system of n qubits is an element of the Hilbert space,

$$\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 = \left(\mathbb{C}^2\right)^{\otimes n} \cong \mathbb{C}^{2^n}.$$

This is consistent with the 2^n basis states of an n-qubit system each having an amplitude independent of the other amplitudes. However, we also restrict the set of vectors corresponding to valid n-qubit systems to those with a norm of 1. The ability of a system of n qubits to store 2^n complex numbers in the form of amplitudes of its basis states provides some insight into the potential power of quantum computing over classical computing, even though the individual amplitudes are not directly accessible.

Although an n-qubit system may be in any superposition of its 2^n basis states, two quantum systems which differ only by *global phase*, i.e., a constant complex factor (of norm 1) applied to the entire system, are physically indistinguishable and considered identical. However, the *relative* phase between two components of a quantum system in superposition does affect the state of the system. In other words, given a quantum state $|\psi\rangle = \alpha |\psi_1\rangle + \beta |\psi_2\rangle$, two systems in the states $|\psi\rangle$ and $e^{i\phi} |\psi\rangle = e^{i\phi}\alpha |\psi_1\rangle + e^{i\phi}\beta |\psi_2\rangle$ are considered to be identical, but two systems in the states $\alpha |\psi_1\rangle + \beta |\psi_2\rangle$ and $\alpha |\psi_1\rangle + e^{i\phi}\beta |\psi_2\rangle$ are not.

Notationally, a quantum system $|\psi\rangle$ written as a tensor product, such as

$$|\psi\rangle \equiv |x_1\rangle \otimes |x_2\rangle \otimes \ldots \otimes |x_n\rangle ,$$

may also be abbreviated as $|x_1\rangle|x_2\rangle\ldots|x_n\rangle$ or $|x_1 x_2 \ldots x_n\rangle$, where each x_p is either 0 or 1. It is also sometimes convenient to consider $x_1 x_2 \ldots x_n$ as the binary representation of an integer,

$$x = \sum_{p=1}^{n} x_p 2^{n-p},$$

and to define $|x\rangle \equiv |x_1 x_2 \ldots x_n\rangle$, where $x \in \mathbb{Z}$.

For reasons of clarity, the probability amplitudes of a given state may remain unnormalized. The normalized vector can be obtained by dividing each amplitude by the norm of the entire vector. For example, the qubit $\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$ is frequently written as $|0\rangle + |1\rangle$.

2.2 Quantum Gates and Reversible Logic

The other essential building block of a quantum computer is the quantum logic gate, which has the ability to alter the state of a system of qubits in the same way

that classical logic gates operate on systems of bits. The most important property of quantum gates is that given a gate, G, which acts on an n-qubit register and maps $|x\rangle \rightarrow G|x\rangle$ for each $x \in \{0, 1, \ldots, 2^n - 1\}$, G will map any superposition of states $|x\rangle$ linearly, i.e., to the same superposition of corresponding outputs, $G|x\rangle$. In other words,

$$G\left(\sum_{x=0}^{2^n-1} \alpha_x |x\rangle\right) = \sum_{x=0}^{2^n-1} \alpha_x G|x\rangle .$$

This means that performing one operation on a system of qubits in a superposition of basis states effectively performs that operation on each of the basis states in that superposition. This ability gives quantum computing another advantage over classical computing.

The operations performed by quantum gates on qubits may be thought of as linear operators on vectors. However, since any operation performed on a qubit must preserve the norm of the qubit, the laws of quantum mechanics dictate that operations on closed quantum systems must be unitary, i.e., that $G^\dagger = G^{-1}$ for any quantum gate G. The requirement that these operations be unitary also implies that each operation must be invertible, which is a restriction not imposed on classical logic. Fortunately, there is a simple way to convert any function from classical logic into a reversible operator corresponding to an invertible, and thus unitary, operator. Given an operator which takes n bits, $x = x_1 x_2 \ldots x_n$ as input, and gives m bits of output, $f_1(x), f_2(x), \ldots, f_m(x)$, we construct a new operator, g, which takes $n + m$ bits of input and gives $n + m$ bits of output, defined by

$$g_p(x_1, \ldots, x_n, y_1, \ldots, y_m) = \begin{cases} x_p & \text{if } p \leq n \\ y_{p-n} \oplus f_{p-n}(x_1, \ldots, x_n) & \text{if } n < p, \end{cases}$$

where $\oplus$ denotes the XOR bit function. The operator g is clearly invertible, and by setting $y_q = 0$ for $1 \leq q \leq m$, we obtain $f_q \equiv g_{n+q}$ whenever $1 \leq q \leq m$. Given

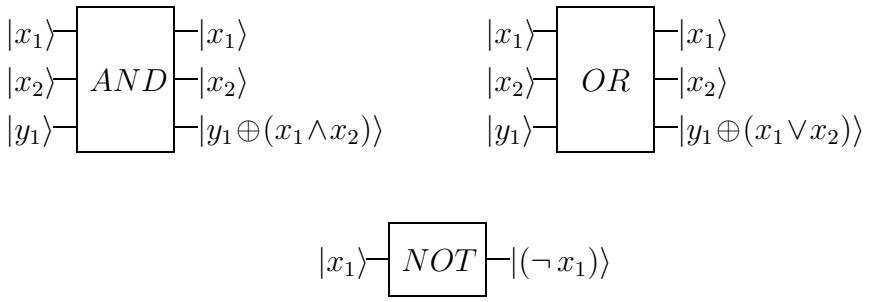

Figure 2.1: Reversible versions of classical AND, OR and NOT gates

reversible versions of the AND, OR and NOT gates (see Figure 2.1) and the fact that the composition of unitary operators is a unitary operator itself, it is easy to see that any classical circuit may be converted into a quantum circuit corresponding to a unitary operator. Note that the above construction is not needed for the NOT gate, as it is already reversible.

However, since an n-qubit quantum state is equivalent to a complex-valued vector of size 2^n, the set of possible unitary operators is much larger than the set of possible classical logical functions on n bits, as each classical logical function corresponds to an operator that can be implemented using only the reversible versions of classical gates AND, OR and NOT given above. Ideally, there would be a finite set of physically realizable quantum gates such that for any given unitary operator, it could be implemented as a quantum network using only these gates. In practice, although it is not possible to implement an arbitrary unitary operator exactly, there exist finite sets of quantum gates that are *universal* in that it is possible to approximate any unitary operator within any given degree of error using a quantum network of these gates.

Although we have described quantum gates basically as unitary operators, we make a distinction between a quantum gate, which simply describes a unitary oper-

Figure 2.2: Hadamard Gate

ation on some collection of input qubits, and the unitary operator, which operates on n-qubit quantum states. Depending on the combination and the order of qubits from an n-qubit quantum state which are used as input qubits for the quantum gate, the quantum gate may be represented by different unitary operators. In fact, for each n, the quantum gate is represented by a different set of unitary operators, as the dimension of the operator depends on n.

2.3 Commonly Used Quantum Gates

Although universal sets of gates are not difficult to construct, in practice the most useful gates either have simple physical implementations or are convenient to work with from a theoretical point of view. The universal set $\{H, T, CNOT\}$, is both easy to work with abstractly and contains gates which are relatively easy to implement in most physical realizations of quantum computers.

The Hadamard gate, H, acts on one qubit (see Figure 2.2), and corresponds to the operator

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

giving us

$$H(\alpha\,|0\rangle + \beta\,|1\rangle) = \frac{\alpha+\beta}{\sqrt{2}}\,|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}\,|1\rangle\,.$$

The $\pi/8$ gate, denoted T, is also a one-qubit quantum gate, and shifts the relative phase of the $|1\rangle$ component of a qubit by an angle of $\phi = \pi/4$, which

Figure 2.3: $\pi/8$ Phase Gate

corresponds to a factor of $e^{i\phi} = e^{i\pi/4}$. This is effected by the operator

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

Since global constant factors are ignored by qubits, we may rewrite T as

$$T = e^{i\pi/8} \begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix},$$

which explains the origin of the name "$\pi/8$ gate".

We have

$$T(\alpha\,|0\rangle + \beta\,|1\rangle) = \alpha\,|0\rangle + e^{i\pi/4}\beta\,|1\rangle$$

(see Figure 2.3). In general, phase shift gates may shift the relative phase by other amounts as well.

The controlled-NOT, or $CNOT$, gate is a two-qubit quantum gate where the value of the control qubit determines whether a NOT gate (denoted by $\oplus$ in Figure 2.4) is applied to the second qubit. In matrix form, this operator can be written as

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

giving us

$$CNOT(\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle) = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{11}|10\rangle + \alpha_{10}|11\rangle\,.$$

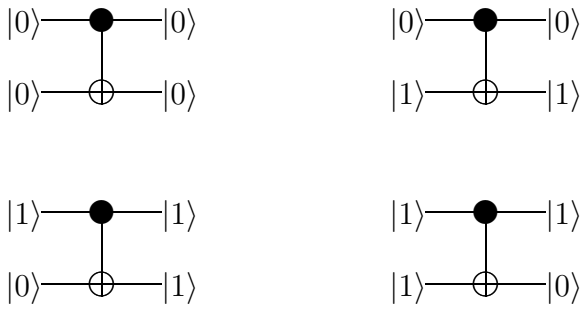

Figure 2.4: Controlled-NOT ($CNOT$) Gate (with control qubit on top)

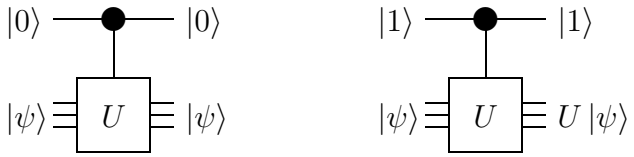

Figure 2.5: Controlled-U Gate (with control qubit on top)

Note that the control qubit is left unchanged.

In general, any operator U may be controlled by a qubit in this manner, and is represented on a quantum network diagram as in Figure 2.5. Given a control qubit, $\alpha|0\rangle + \beta|1\rangle$, and $|\psi\rangle$ as the input state for U, the controlled-U gate outputs

$$\text{c-}U((\alpha|0\rangle + \beta|1\rangle)|\psi\rangle) = \alpha|0\rangle|\psi\rangle + \beta|1\rangle(U|\psi\rangle).$$

The proof of the universality of the set $\{H, T, CNOT\}$ involves constructing a decomposition of arbitrary unitary operators into a combination of $CNOT$ gates and single-qubit operators. It can then be shown that any single-qubit operator can be approximated to arbitrary precision using a combination of Hadamard (H) and $\pi/8$ (T) gates. Both of these constructions may be found in [10].

2.4 Universal Sets of Quantum Gates

To formally define universality for a set of quantum gates, we must first define a distance function for unitary operators. First, for arbitrary operators on n-qubit systems:

$$A : \left(\mathbb{C}^2\right)^{\otimes n} \to \left(\mathbb{C}^2\right)^{\otimes n},$$

we will use the norm function induced by the Euclidean norm for vectors,

$$\|A\| = \max_{\||\psi\rangle\|=1} \|A|\psi\rangle\| .$$

Now, the distance between two unitary operators U and U' is simply defined as

$$d(U, U') = \|U - U'\| .$$

We may now define a universal set of quantum gates as a finite set of quantum gates that has the property that, given a positive integer n, a unitary operator $U : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$ which operates on n-qubit systems, and an error tolerance $\epsilon > 0$, there exists a sequence of gates (and input qubits for each gate) that implements a unitary operator U' so that $d(U, U') < \epsilon$.

The distance function d also has two useful properties. First, given two unitary operators U_1 and U_2 which correspond to the same quantum gate G being applied to different combinations of qubits within the same quantum register, let G' be an approximation of G, and U_1' and U_2' be the corresponding unitary operators which correspond to G' and approximate U_1 and U_2, respectively. Then we have $d(U_1, U_1') = d(U_2, U_2')$. This allows us to speak of the distance between a quantum gate and its approximation independently of which qubits the gate would be applied to in a given register. Secondly, given two different unitary operators U_1 and U_2 which take the same number of input qubits, and two respective approximations

U_1' and U_2', we have

$$\begin{aligned} d(U_2U_1, U_2'U_1') &= \|U_2U_1 - U_2'U_1'\| \\ &= \|U_2U_1 - U_2U_1' + U_2U_1' - U_2'U_1'\| \\ &\leq \|U_2(U_1 - U_1')\| + \|(U_2 - U_2')U_1'\| \\ &\leq \|U_2\|\,\|U_1 - U_1'\| + \|U_2 - U_2'\|\,\|U_1'\| \\ &\leq d(U_1, U_1') + d(U_2, U_2') \end{aligned}$$

using both the subadditive and submultiplicative property of the norm and the fact that $\|U\| = 1$ for any unitary operator. In other words, since a quantum network is simply a sequence of unitary operators applied to a quantum register, the total error incurred by replacing the gates in the network with an approximate version is at most the sum of the individual errors incurred by each gate being replaced. If we used a universal set of quantum gates to approximate each gate in a quantum network to within a distance of ϵ, the cumulative error would be bounded from above by a linear function of the number of gates in the network. Also, since universal sets of quantum gates are finite, there is only a constant factor overhead in using a different universal set of quantum gates to approximate each of the gates in the first set to within a fixed distance, ϵ. This gives us a measure of the efficiency of a quantum algorithm, independent of the choice of universal set quantum gates used, which is based on the number of gates it would take to implement a given function as a uniform family of circuits $\{C_n\}_{n=1}^{\infty}$, where a particular C_n handles the function for n-qubit input. The only practical restriction on uniform families is that there must exist a Turing machine which takes an integer n as input and outputs the sequence of quantum gates used to construct C_n in $O(n^c)$ steps, for some fixed integer, c.

However, the question remains over how efficiently we may approximate a given

quantum gate using gates from a universal set, with respect to the desired tolerance for error ϵ. In other words, given a specific quantum gate, we would like to know how many gates from a universal set, as a function of ϵ, are needed to construct a quantum network which approximates the given gate to within a distance of ϵ. If gates cannot be approximated efficiently, then great increases in complexity could occur when implementing quantum algorithms using a restricted set of gates.

Fortunately, the Solovay-Kitaev theorem gives us an efficient construction which allows any quantum gate to be approximated to within a distance of ϵ using a quantum network with $O(\log^c(1/\epsilon))$ gates from a universal set,[2] where c is any fixed constant whose value is greater than 2. Suppose we have a uniform family of quantum circuits $\{C_n\}_{n=1}^{\infty}$ which implements a quantum algorithm, and for each integer n, we define the function $f(n)$ to be the number of quantum gates in the corresponding circuit C_n. If we wish to approximate C_n using a universal set of gates to within a distance of ϵ, then, by our observation above, we should approximate each individual gate in C_n to within a distance of $\epsilon/f(n)$ in order to have a total error within ϵ. By the Solovay-Kitaev theorem, for each gate in C_n, the approximate version will require $O(\log^c(f(n)/\epsilon))$ gates from the universal set. This gives us a total of $O(f(n)\log^c(f(n)/\epsilon))$ for the entire circuit, which is a very small overhead in the number of gates for most purposes.

[2]Technically, the Solovay-Kitaev construction also requires that, for each gate G in the universal set, we also have a gate which performs the inverse operation G^{-1}. However, this requirement is not important for our purposes.

Also, the original construction concerns only the simulation of single-qubit gates, but can be easily expanded to accomodate arbitrary unitary operators.

Chapter 3

The Quantum Fourier Transform and Phase Estimation

In classical computing, the applications of Fourier transforms are mostly limited to digital signal processing or multi-precision integer multiplication routines. Using the ability of quantum gates to perform computations simultaneously on up to 2^n quantum basis states using only an n-qubit register, the quantum computing analogue of the Fourier transform can be used to perform computational tasks that are intractable with today's classical algorithms.

This chapter develops the quantum Fourier transform algorithm and explores its use within algorithms for quantum phase estimation.

3.1 The Discrete Fourier Transform

In simplest terms, the discrete Fourier transform is a function on a complex vector of fixed size, N, which returns another complex vector. Specifically, given the vector

$(z_0, z_1, \ldots, z_{N-1}) \in \mathbb{C}^N$, the discrete Fourier transform, DFT, will return the vector $(Z_0, Z_1, \ldots, Z_{N-1}) \in \mathbb{C}^N$, where

$$Z_k = \sum_{j=0}^{N-1} z_j e^{2\pi ijk/N}.$$

However, for the purposes of quantum information processing, we use the normalized version of the transform, which is scaled by a factor of $\frac{1}{\sqrt{N}}$ in order to make the transform a unitary operator, giving us the transform

$$Z_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} z_j e^{2\pi ijk/N}. \tag{3.1}$$

This constant factor does not affect any of the mathematical properties of the discrete Fourier transform. Instead, it allows us to construct a quantum algorithm with which to implement it.

Currently, the best classical algorithm known that computes a discrete Fourier transform is the Fast Fourier Transform (FFT), which computes the transform of a vector whose size $N = 2^n$ is an integer power of 2. By dividing the data into two halves, recursively performing a smaller DFT on each half, and combining the two results into the whole transform, the DFT can be computed in $O(N \log N)$ time. However, this is still an exponential function of n.

3.2 The Quantum Fourier Transform

The quantum Fourier transform is a quantum circuit which performs a discrete Fourier transform on the complex-valued vector of 2^n probability amplitudes associated with an n-qubit quantum system. Specifically, given an n-qubit state as a superposition of basis states $|0\rangle$, $|1\rangle$, ..., $|2^n - 1\rangle$, the QFT maps each basis state

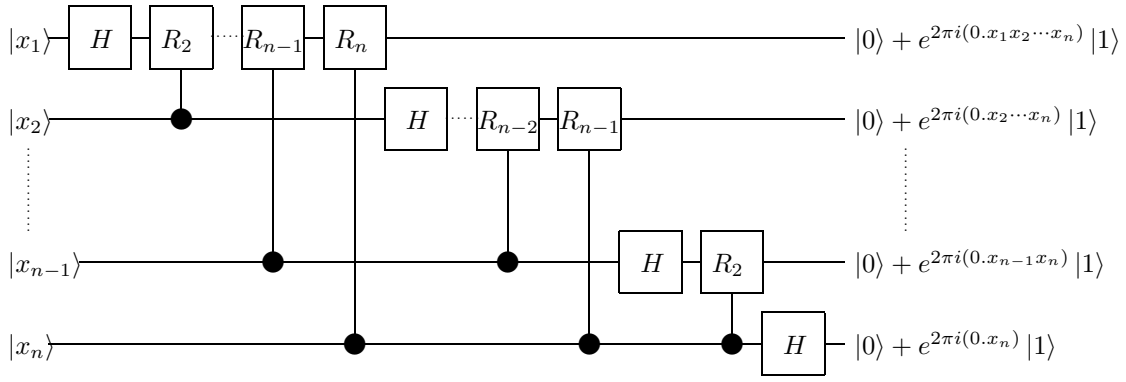

Figure 3.1: A quantum circuit that performs a quantum Fourier transform

$|j\rangle$ to

$$QFT(|j\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi ijk/2^n} |k\rangle . \tag{3.2}$$

Since quantum circuits are linear operators, this means that given the n-qubit state $|\psi\rangle$ as input, where

$$|\psi\rangle = \sum_{j=0}^{2^n-1} z_j |j\rangle ,$$

the QFT will output the n-qubit state

$$QFT(|\psi\rangle) = \sum_{j=0}^{2^n-1} QFT(z_j |j\rangle) = \sum_{j=0}^{2^n-1} \sum_{k=0}^{2^n-1} \frac{z_j e^{2\pi ijk/2^n}}{\sqrt{2^n}} |k\rangle = \sum_{k=0}^{2^n-1} Z_k |k\rangle ,$$

with Z_k as defined in Equation (3.1) above. The algorithm effectively takes the 2^n probability amplitudes of an n-qubit state as a vector of size 2^n and performs a discrete Fourier transform on that vector so that the result is encoded in the probability amplitudes of the output state.

The simplest way to show that the normalized Fourier transform is a unitary operation is to demonstrate the quantum circuit that performs the QFT. In the diagram of the QFT circuit given in Figure 3.1, the input register contains an

n-qubit basis state $|x\rangle$ which is rewritten as the tensor product of the individual qubits in its binary expansion:

$$|x\rangle \equiv |x_1 x_2 \cdots x_n\rangle \equiv |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle \,.$$

The gates labelled H are Hadamard gates, as described in Section 2.3, and the gates labelled R_m represent a series of one-qubit phase rotation gates. For each integer $m \geq 2$, the gate R_m shifts the phase of the $|1\rangle$ component of the input qubit by a factor of $e^{2\pi i/2^m}$, representing the unitary transformation

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^m} \end{bmatrix}.$$

However, in the QFT circuit, each R_m gate is controlled by another qubit, which is indicated by a large dot connected to the gate by a vertical line. This means that given a two-qubit state, $|\psi_1\rangle|\psi_2\rangle$, composed of the controlling qubit, $|\psi_1\rangle$, and the input qubit, $|\psi_2\rangle$, the controlled-R_m gate represents the unitary transformation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i/2^m} \end{bmatrix}.$$

If the controlled-R_m gate is being applied to a basis state, $|x_\ell\rangle$, where x_ℓ is either 0 or 1, then depending on the value of x_ℓ, we can either say that the controlled-R_m gate performs the identity transformation, or the R_m transformation. However, we may combine the two and equivalently say that the controlled-R_m gate performs the transformation

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i x_\ell/2^m} \end{bmatrix}$$

on the qubit $|\psi_2\rangle$, effectively performing a data-dependent phase rotation.

For the output qubits in Figure 3.1, we use the notation $0.x_\ell x_{\ell+1} x_{\ell+2} \cdots x_{\ell+m-1}$ to represent the binary fraction $x_\ell/2 + x_{\ell+1}/4 + x_{\ell+2}/8 + \cdots + x_{\ell_{m-1}}/2^m$. In the QFT circuit, each qubit is initially in the input state $|x_\ell\rangle$ before being used as the controlling qubit of a series of controlled-R_m gates, which do not change the state of the qubit. A Hadamard gate, H, is then applied to the qubit, producing the state $\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_\ell}|1\rangle)$, which can be rewritten as $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_\ell)}|1\rangle)$. From here, each controlled-R_m gate from $m = 2$ to $m = n + 1 - \ell$ is applied in sequence, with $|x_{\ell+m-1}\rangle$ as the controlling qubit. Each successive controlled-R_m gate shifts the phase of the $|1\rangle$ component of a qubit by a factor of $e^{2\pi i x_{(\ell+m-1)}/2^m}$. The final state of the qubit is thus $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_\ell x_{\ell+1}\cdots x_n)})$ at the end of the QFT circuit. Finally, in order to be a true QFT, the final order of the qubits must be reversed, giving us

$$\frac{\left(|0\rangle + e^{2\pi i(0.x_n)}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i(0.x_{n-1}x_n)}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i(0.x_1 x_2 \cdots x_n)}|1\rangle\right)}{\sqrt{2^n}}$$

as the final state of the quantum register. This reversal is not indicated in Figure 3.1.

Now, in order to demonstrate the correctness of the QFT circuit given in Figure 3.1, we need to show that the final output state given by the circuit is equivalent to $QFT(|x\rangle)$, as defined by equation (3.2). Taking the final output state,

$$\frac{\left(|0\rangle + e^{2\pi i(0.x_n)}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i(0.x_{n-1}x_n)}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i(0.x_1 x_2 \cdots x_n)}|1\rangle\right)}{\sqrt{2^n}},$$

we first note that since $e^{2\pi i} = 1$, we have

$$e^{2\pi i(0.x_\ell x_{\ell+1}\cdots x_n)} = e^{2\pi i(x_1 x_2 \cdots x_{\ell-1}.x_\ell \cdots x_n)} = e^{2\pi i(2^{\ell-1}x/2^n)},$$

allowing us to rewrite the output state as

$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i(2^{n-1}x/2^n)}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i(2^1 x/2^n)}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i(2^0 x/2^n)}|1\rangle\right).$$

Given an n-qubit basis state $|k\rangle = |k_1 k_2 \cdots k_n\rangle$, the $|k\rangle$ component of the final state is

$$\frac{1}{\sqrt{2^n}} \left(e^{2\pi i k_1 (2^{n-1}x/2^n)} |k_1\rangle\right) \otimes \cdots \otimes \left(e^{2\pi i k_{n-1} (2^1 x/2^n)} |k_{n-1}\rangle\right) \otimes \left(e^{2\pi i k_n (2^0 x/2^n)} |k_n\rangle\right),$$

using the fact that $e^{2\pi i(0)} = 1$. Expanding this tensor product yields

$$\begin{aligned} & \frac{1}{\sqrt{2^n}} \left(e^{2\pi i k_1 (2^{n-1}x/2^n)}\right) \cdots \left(e^{2\pi i k_{n-1} (2^1 x/2^n)}\right) \left(e^{2\pi i k_n (2^0 x/2^n)}\right) |k\rangle \\ = & \frac{1}{\sqrt{2^n}} e^{2\pi i (2^{n-1}k_1 + \ldots + 2^1 k_{n-1} + 2^0 k_n)(x/2^n)} |k\rangle \\ = & \frac{1}{\sqrt{2^n}} e^{2\pi i k x/2^n} |k\rangle . \end{aligned}$$

Summing over all 2^n components gives us

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k x/2^n} |k\rangle = QFT(|x\rangle)$$

as the final output state given by the circuit.

3.3 Quantum Phase Estimation and QFT

Given a unitary operator U and an *eigenstate*, which is a quantum register that contains an eigenvector of U, the Phase Estimation Problem asks for the eigenvalue which corresponds to the given eigenstate. Specifically, given U with an eigenstate $|u\rangle$ such that $U|u\rangle = \lambda|u\rangle$, where λ is the corresponding eigenvalue, we are to find ϕ so that $\lambda = e^{2\pi i \phi}$.

The QFT-based phase estimation algorithm uses the fact that if the phase amount ϕ is an exact integer multiple of $1/2^n$ and is encoded as the binary fraction $\phi = (0.x_1 x_2 \ldots x_n)$, then, given the state $|x\rangle = |x_1 x_2 \ldots x_n\rangle$ as input, the QFT

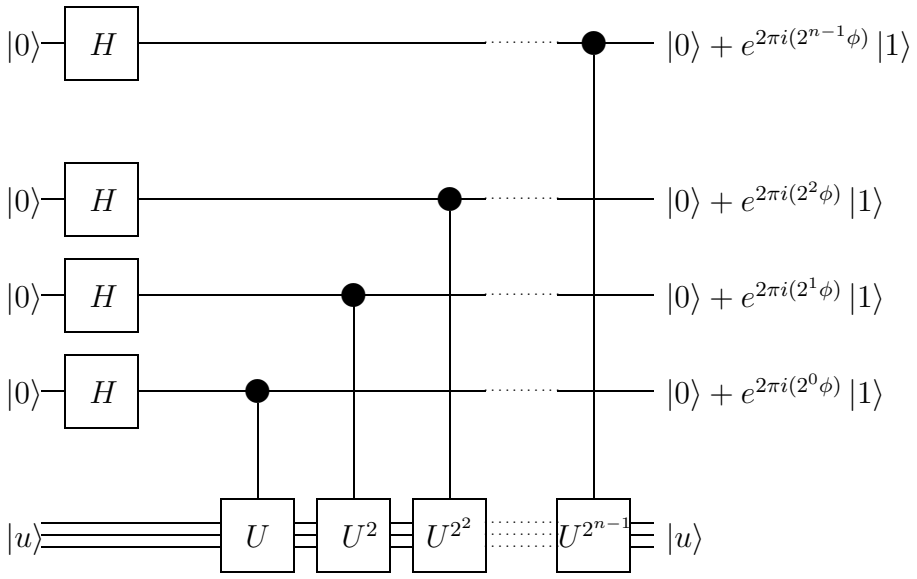

Figure 3.2: The first half of the phase estimation algorithm

returns the state

$$QFT(|x\rangle) = \frac{(|0\rangle + e^{2\pi i(2^0\phi)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(2^1\phi)}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i(2^{n-1}\phi)}|1\rangle)}{\sqrt{2^n}}.$$

This means that if the state $QFT(|x\rangle)$ could be constructed some other way, the inverse of the QFT operation, QFT^{-1}, would output the state $|x_1 x_2 \ldots x_n\rangle$, giving us the individual bits in the binary fractional representation of ϕ. Even if ϕ were not an exact integer multiple of $1/2^n$, applying QFT^{-1} to the constructed state $|\psi\rangle$ should still yield a close approximation to ϕ with high probability.

Given a quantum register in the state $|\phi\rangle$, the first half of the algorithm prepares the state

$$|\psi\rangle = \frac{(|0\rangle + e^{2\pi i(2^{n-1}\phi)}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i(2^1\phi)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(2^0\phi)}|1\rangle)}{\sqrt{2^n}}$$

using the quantum circuit given in Figure 3.2. Note that the qubits in the register $|\psi\rangle$ are in the reverse order of the qubits in $QFT(|x\rangle)$. However, the inverse QFT can easily be applied to its input qubits in reverse order as well.

To see how the circuit in Figure 3.2 produces the desired state, consider any one of the qubits initially in the state $|0\rangle$ in the input register. First, a Hadamard gate, H, is applied, leaving the qubit in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Then, a controlled-$U^{2^j}$ gate is applied, which does nothing when the control qubit is in the state $|0\rangle$, and applies U^{2^j} to the eigenstate when the control qubit is in the state $|1\rangle$.

Given the input $|0\rangle|u\rangle$, the controlled-U gate has no effect, and outputs $|0\rangle|u\rangle$. Given the input $|1\rangle|u\rangle$, the controlled-U gate outputs

$$\text{c-}U(|1\rangle|u\rangle) = |1\rangle(U|u\rangle) = |1\rangle(e^{2\pi i\phi}|u\rangle) = e^{2\pi i\phi}|1\rangle|u\rangle = (e^{2\pi i\phi}|1\rangle)|u\rangle .$$

So, given the actual input state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|u\rangle$, the controlled-$U$ gate outputs the state

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\phi}|1\rangle)|u\rangle ,$$

and, in general, the controlled-U^{2^j} gate outputs the state

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(2^j\phi)}|1\rangle)|u\rangle ,$$

giving the desired qubit states in the output register of the quantum circuit in Figure 3.2.

The second half of the QFT-based quantum phase estimation algorithm involves applying an inverse QFT to the state $|\psi\rangle$ obtained from the circuit in Figure 3.2. Since the inverse of a product of unitary operators is the product of the inverses of the same unitary operators in reverse order:

$$(U_1U_2\ldots U_n)^{-1} = U_n^{-1}\ldots U_2^{-1}U_1^{-1},$$

the inverse of a quantum circuit can be constructed by reversing the quantum network and replacing each quantum gate with the corresponding inverse gate. The inverse of a controlled phase rotation gate R_m, which rotates the phase of a qubit by

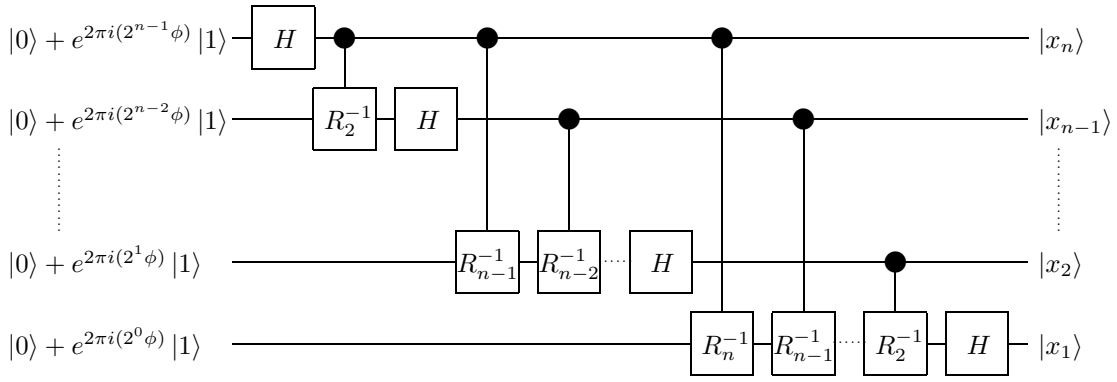

Figure 3.3: The second half of the phase estimation algorithm: The Inverse QFT

an angle of $2\pi i/2^m$ when the control qubit is in the state $|1\rangle$, is simply a controlled phase rotation gate that rotates the phase by the opposite angle, $-2\pi i/2^m$, when the control qubit is in the state $|1\rangle$. The Hadamard gate, H, is its own inverse: $H^{-1} = H$. The inverse QFT circuit applied to a quantum register containing the state $|\psi\rangle$ in reverse qubit order is shown in Figure 3.3. When the phase estimation algorithm yields $\hat{\phi}$, the estimate of the phase ϕ, the individual bits of $\hat{\phi}$ are obtained in reverse order, so we classically reverse the order of the qubits in the output $QFT^{-1}(|\psi\rangle)$. This classical operation is not shown in Figure 3.3.

3.4 Analysis of QFT-based Phase Estimation

We would not generally expect ϕ to be an exact multiple of $\frac{1}{2^n}$. Fortunately, the estimated phase $\hat{\phi}$, which is returned by the inverse QFT, is the nearest exact multiple of $\frac{1}{2^n}$ to ϕ with high probability.

To establish this fact, first note that the input state for the circuit in Figure

3.3,

$$|\psi\rangle = \frac{(|0\rangle + e^{2\pi i(2^{n-1}\phi)}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i(2^1\phi)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(2^0\phi)}|1\rangle)}{\sqrt{2^n}},$$

can be rewritten using the n-qubit basis states $|0\rangle$, $|1\rangle$, $|2\rangle$, ..., $|2^n - 1\rangle$ as

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i(k\phi)}|k\rangle$$

by expanding the tensor product. Applying the inverse QFT, which takes each basis state $|k\rangle$ and outputs

$$QFT^{-1}(|k\rangle) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{-2\pi ijk/2^n}|j\rangle,$$

we obtain

$$QFT^{-1}(|\psi\rangle) = \frac{1}{2^n} \sum_{j=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{-2\pi ijk/2^n} e^{2\pi i(k\phi)}|j\rangle.$$

The amplitude of any particular basis state $|j\rangle$ is

$$\alpha_j = \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{-2\pi ijk/2^n} e^{2\pi i(k\phi)} = \frac{1}{2^n} \sum_{k=0}^{2^n-1} e^{2\pi i(\phi - j/2^n)k}, \tag{3.3}$$

which is a geometric series. If $\phi = j/2^n$, which is an integer multiple of $1/2^n$, then

$$\alpha_j = \frac{1}{2^n} \sum_{k=0}^{2^n-1} 1 = 1.^1$$

Otherwise, we can use the geometric series formula to simplify α_j, obtaining

$$\alpha_j = \frac{1}{2^n} \frac{1 - e^{2\pi i(\phi - j/2^n)(2^n)}}{1 - e^{2\pi i(\phi - j/2^n)}} = \frac{1}{2^n} \frac{1 - e^{2\pi i(2^n\phi - j)}}{1 - e^{2\pi i(\phi - j/2^n)}}.$$

[1] Since ϕ represents the phase difference between two quantum states as a fraction of 2π, two phase amounts (such as ϕ and $j/2^n$) are understood to be equal when their fractional parts are equal. Also, j is understood to be a quantity modulo 2^n.

Therefore, the probability of obtaining a particular state $|j\rangle$ upon measurement at the end of the phase estimation algorithm is

$$P(\hat{\phi} = j/2^n) = \left| \frac{1}{2^n} \frac{1 - e^{2\pi i(2^n\phi - j)}}{1 - e^{2\pi i(\phi - j/2^n)}} \right|^2 = \frac{1}{2^{2n}} \frac{|1 - e^{2\pi i(2^n\phi - j)}|^2}{|1 - e^{2\pi i(\phi - j/2^n)}|^2}.$$

At this point, it is useful to establish the following useful lemma:

Lemma 3.4.1 *For any real number θ,*

$$|1 - e^{2\pi i\theta}|^2 = 4\sin^2 \pi\theta$$

and

$$|1 + e^{2\pi i\theta}|^2 = 4\cos^2 \pi\theta.$$

Proof We have

$$\begin{aligned} |1 - e^{2\pi i\theta}|^2 &= |1 - \cos 2\pi\theta - i \sin 2\pi\theta|^2 \\ &= (1 - \cos 2\pi\theta)^2 + \sin^2 2\pi\theta \\ &= 1 - 2\cos 2\pi\theta + \cos^2 2\pi\theta + \sin^2 2\pi\theta \\ &= 2 - 2\cos 2\pi\theta \\ &= 4\sin^2 \pi\theta, \end{aligned}$$

using the half-angle formula. The second result can be proven using a similar derivation. □

By using Lemma 3.4.1, we can simplify our expression for $P(\hat{\phi} = j/2^n)$ to obtain

$$P(\hat{\phi} = j/2^n) = \frac{1}{2^{2n}} \frac{\sin^2 \pi(2^n\phi - j)}{\sin^2 \pi(\phi - j/2^n)}.$$

Letting $\delta = \phi - j/2^n$, we can rewrite the expression as

$$P(\hat{\phi} = j/2^n) = \left(\frac{\sin \pi(2^n\delta)}{2^n \sin \pi\delta} \right)^2.$$

To determine a lower bound on the probability of the phase estimation algorithm obtaining the estimate $\hat{\phi}$ nearest to the actual value ϕ, we first establish another lemma:

Lemma 3.4.2 *For any real number $\theta \neq 0$, the sequence*

$$S_n = \left(\frac{\sin\theta}{2^n \sin\frac{\theta}{2^n}} \right)^2, \quad n \in \mathbb{Z}, n \geq 0$$

decreases monotonically, and

$$\lim_{n\to\infty} S_n = \frac{\sin^2\theta}{\theta^2}.$$

Proof Since

$$\frac{S_{n+1}}{S_n} = \frac{\left(\frac{\sin\theta}{2^{n+1}\sin\frac{\theta}{2^{n+1}}} \right)^2}{\left(\frac{\sin\theta}{2^n \sin\frac{\theta}{2^n}} \right)^2} = \frac{\left(2^n \cdot 2\sin\frac{\theta}{2^{n+1}}\cos\frac{\theta}{2^{n+1}}\right)^2}{\left(2^{n+1}\sin\frac{\theta}{2^{n+1}}\right)^2} = \cos\frac{\theta}{2^{n+1}} \leq 1,$$

S_n is a monotone decreasing sequence. To establish the limit, notice that

$$\begin{aligned} \lim_{n\to\infty} 2^n \sin\frac{\theta}{2^n} &= \lim_{t\to 0} \frac{\sin\theta t}{t} \qquad (\text{letting } t = \tfrac{1}{2^n}) \\ &= \lim_{t\to 0} \frac{\theta\sin\theta t}{\theta t} \\ &= \theta. \end{aligned}$$

Now,

$$\begin{aligned} \lim_{n\to\infty} 2^n \sin\frac{\theta}{2^n} &= \theta \\ \Rightarrow \quad \lim_{n\to\infty} \left(\frac{1}{2^n \sin\frac{\theta}{2^n}} \right)^2 &= \frac{1}{\theta^2} \\ \Rightarrow \quad \lim_{n\to\infty} \left(\frac{\sin\theta}{2^n \sin\frac{\theta}{2^n}} \right)^2 &= \left(\frac{\sin\theta}{\theta} \right)^2, \end{aligned}$$

as desired. □

Corollary 3.4.3 *For any integer $n \geq 0$ and any real number $\theta \neq 0$,*

$$\left(\frac{\sin\theta}{2^n \sin\frac{\theta}{2^n}}\right)^2 \geq \left(\frac{\sin\theta}{\theta}\right)^2.$$

□

Using Corollary 3.4.3, we can obtain lower bounds for the probability of having the phase estimation algorithm return ϕ rounded to the nearest integer multiple of $1/2^n$ (rounded upwards in case of a tie), $P\left(-\frac{1}{2^{n+1}} < \delta \leq \frac{1}{2^{n+1}}\right)$.

If ϕ is an integer multiple of $1/2^n$, say $\phi = j/2^n$, then we have $\alpha_j = 1$ from Equation (3.3). This means that $P(\hat{\phi} = j/2^n) = 1$, and so $P(\hat{\phi} = \phi) = 1$. Now, we consider the case when ϕ is not an integer multiple of $1/2^n$.

Let j be the integer so that

$$-\frac{1}{2^{n+1}} < \delta = \phi - j/2^n \leq \frac{1}{2^{n+1}}.$$

Then

$$|\delta| = |\phi - j/2^n| \leq \frac{1}{2^{n+1}},$$

which means that the probability that $\hat{\phi} = j/2^n$ is

$$\begin{aligned}
P(\hat{\phi} = j/2^n) &= \left(\frac{\sin\pi(2^n\delta)}{2^n \sin\pi\delta}\right)^2 \\
&\geq \left(\frac{\sin\pi(2^n\delta)}{\pi(2^n\delta)}\right)^2 \quad \text{(by Corollary 3.4.3)} \\
&= \left(\frac{\sin\pi|2^n\delta|}{\pi|2^n\delta|}\right)^2 \\
&\geq \left(\frac{\sin\frac{\pi}{2}}{\frac{\pi}{2}}\right)^2 = \frac{4}{\pi^2}
\end{aligned}$$

since $\frac{\sin^2\theta}{\theta^2}$ is a monotone decreasing function on the interval $\left(0, \frac{\pi}{2}\right]$ and

$$|2^n\delta| \leq \frac{2^n}{2^{n+1}} = \frac{1}{2}.$$

We may also establish a lower bound for the probability of the value of $\hat{\phi}$ being one of the two nearest integer multiples of $1/2^n$ since in practice, these will be the two most likely results of the phase estimation algorithm. We can write this probability as $P(|\delta| < \frac{1}{2^n})$.

Assuming that the value of ϕ is not an exact integer multiple of $1/2^n$, in which case the probability of obtaining the correct answer is 1, let j be the integer so that $j/2^n$ is the nearest integer multiple of $1/2^n$ which is less than ϕ. Clearly, $(j+1)/2^n$ is the nearest integer multiple of $1/2^n$ which is greater than ϕ. Again, we let $\delta_0 = \phi - j/2^n$. Note that for the other nearest estimate $(j+1)/2^n$, we have $\delta_1 = \phi - (j+1)/2^n = \delta_0 - 1/2^n$.

The probability of the phase estimation algorithm returning either $\hat{\phi} = j/2^n$ or $\hat{\phi} = (j+1)/2^n$ is

$$\begin{aligned}
P(|\delta| < \tfrac{1}{2^n}) &= P(\hat{\phi} = j/2^n) + P(\hat{\phi} = (j+1)/2^n) \\
&= \left(\frac{\sin\pi(2^n\delta_0)}{2^n \sin\frac{\pi(2^n\delta_0)}{2^n}}\right)^2 + \left(\frac{\sin\pi(2^n\delta_1)}{2^n \sin\frac{\pi(2^n\delta_1)}{2^n}}\right)^2 \\
&\geq \left(\frac{\sin\pi(2^n\delta_0)}{\pi(2^n\delta_0)}\right)^2 + \left(\frac{\sin\pi(2^n\delta_1)}{\pi(2^n\delta_1)}\right)^2 \quad \text{(by Corollary 3.4.3)} \\
&= \left(\frac{\sin\pi(2^n\delta_0)}{\pi(2^n\delta_0)}\right)^2 + \left(\frac{\sin\pi(2^n\delta_0 - 1)}{\pi(2^n\delta_0 - 1)}\right)^2 \\
&= \left(\frac{\sin\pi(2^n\delta_0)}{\pi(2^n\delta_0)}\right)^2 + \left(\frac{\sin\pi(2^n\delta_0)}{\pi(2^n\delta_0 - 1)}\right)^2. \quad (\sin^2\theta \text{ has period } \pi)
\end{aligned}$$

Since we know that $0 \leq \delta_0 \leq 1/2^n$, we can establish a lower bound on the

probability $P(|\delta| < \frac{1}{2^n})$ by establishing a lower bound for the function

$$\left(\frac{\sin \pi x}{\pi x}\right)^2 + \left(\frac{\sin \pi x}{\pi (x-1)}\right)^2 = \frac{\sin^2 \pi x}{\pi^2}\left(\frac{1}{x^2} + \frac{1}{(x-1)^2}\right)$$

for $x \in (0,1)$. Shifting x by $\frac{1}{2}$, we see that this is equivalent to finding the minimum of

$$\begin{aligned} f(x) &= \frac{\cos^2 \pi x}{\pi^2}\left(\frac{1}{(x+\frac{1}{2})^2} + \frac{1}{(x-\frac{1}{2})^2}\right) \\ &= \frac{\cos^2 \pi x}{\pi^2}\left(\frac{(x+\frac{1}{2})^2 + (x-\frac{1}{2})^2}{(x-\frac{1}{2})^2(x+\frac{1}{2})^2}\right) \\ &= \frac{8\cos^2 \pi x}{\pi^2}\left(\frac{4x^2+1}{(4x^2-1)^2}\right) \end{aligned}$$

in the interval $x \in (-\frac{1}{2}, \frac{1}{2})$.

At $x = 0$, we have $f(x) = \frac{8}{\pi^2}$. We will show that this is the minimum value, and therefore, a lower bound for $P(|\delta| < \frac{1}{2^n})$. First, note that since $f(x) = f(-x)$, we may restrict our attention to the interval $x \in [0, \frac{1}{2})$.

Since $\cos^2 \pi x = 1 - \sin^2 \pi x \geq 1 - (\pi x)^2 \geq 1 - 10x^2$, we have

$$\begin{aligned} f(x) &\geq \frac{8(1-10x^2)}{\pi^2}\left(\frac{4x^2+1}{(4x^2-1)^2}\right) \\ &= \frac{8}{\pi^2} + \frac{16x^2(1-28x^2)}{\pi^2(2x+1)^2(2x-1)^2}, \end{aligned}$$

so that $f(x) - \frac{8}{\pi^2}$ has zeroes at $\pm\frac{1}{2\sqrt{7}}$ and 0. We can quickly see that we have $f(x) \geq \frac{8}{\pi^2}$ on the interval $x \in [0, \frac{1}{2\sqrt{7}}]$.

Now, since $\cos^2 \pi x = \sin^2 \pi(x - \frac{1}{2}) \geq (\pi(x-\frac{1}{2}))^2 - \frac{1}{3}(\pi(x-\frac{1}{2}))^4$, we have

$$\begin{aligned} f(x) &\geq \frac{8}{\pi^2}\left(\pi^2\left(x-\frac{1}{2}\right)^2 - \frac{\pi^4}{3}\left(x-\frac{1}{2}\right)^4\right)\left(\frac{4x^2+1}{(4x^2-1)^2}\right) \\ &= \left(2(2x-1)^2 - \frac{\pi^2}{6}(2x-1)^4\right)\left(\frac{4x^2+1}{(4x^2-1)^2}\right) \end{aligned}$$

$$\begin{aligned} &\geq \left(2(2x-1)^2 - \frac{10}{6}(2x-1)^4\right)\left(\frac{4x^2+1}{(4x^2-1)^2}\right) \\ &= \left(2 - \frac{10}{6}(2x-1)^2\right)\left(\frac{4x^2+1}{(2x+1)^2}\right) \\ &= \frac{(4x^2+1)(1+20x-20x^2)}{3(2x+1)^2}. \end{aligned}$$

Taking the derivative of

$$g(x) = \frac{(4x^2+1)(1+20x-20x^2)}{3(2x+1)^2},$$

we obtain

$$g'(x) = \frac{8(2-9x+30x^2-20x^3-40x^4)}{3(2x+1)^3}.$$

It is easy to verify that this derivative has no zeroes in the range $x \in [0, \frac{1}{2})$, so that $g(x)$ is an increasing function on that range. It is also easy to verify that

$$g\left(\frac{1}{2\sqrt{7}}\right) = \frac{16}{3}\frac{1+5\sqrt{7}}{\left(7+\sqrt{7}\right)^2} \geq \frac{8}{\pi^2},$$

so that $f(x) \geq g(x) \geq \frac{8}{\pi^2}$ on the interval $x \in [\frac{1}{2\sqrt{7}}, \frac{1}{2})$. We have finally established that

$$P\left(|\delta| < \frac{1}{2^n}\right) \geq \frac{8}{\pi^2}.$$

3.5 Using Individual Trials in Phase Estimation

Griffiths and Niu [7] observed that the final state of each of the n qubits, $|x_p\rangle$, $1 \leq p \leq n$, output from the inverse QFT (Figure 3.3) before measurement, depends only on those qubits with higher indices, as they are the only qubits used as control bits in the controlled phase rotation gates. Indeed, the last qubit, $|x_n\rangle$, is obtained independently of all the other qubits.

Given this observation, Griffiths and Niu suggested a "semiclassical" version of the quantum Fourier transform circuit in which the output of the inverse QFT is obtained one qubit at a time: starting with $|x_n\rangle$, which can be computed without using any of the other qubits. Proceeding one qubit at a time, we can then use the results from measuring the previous qubits to "classically" determine whether a particular phase rotation gate will be applied to the current qubit. We need to apply only those phase rotation gates, rather than applying a collection of controlled phase rotation gates and letting the control qubits determine whether a phase rotation will be applied using quantum-mechanical effects. Surprisingly, replacing the controlled phase rotation gates with a classical process does not affect the output of the inverse QFT, in that the probability of any particular answer being output by the inverse QFT is the same as the probability of that answer being output by the individual trials.

To see why this particular property of quantum entanglement is true, consider a controlled quantum gate $\mathrm{c}-U$ which applies the operator U to a register $|\psi_2\rangle$, depending on the state of the control qubit $|\psi_1\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$. If we apply the controlled-U gate, we obtain

$$\mathrm{c}-U(|\psi_1\rangle |\psi_2\rangle) = \alpha_0 |0\rangle |\psi_2\rangle + \alpha_1 |1\rangle U |\psi_2\rangle .$$

If we are given that the measured value of the control qubit is 0, then the conditional probabilities for the measured values of the second register are the same as the probabilities for the measured values of $|\psi_2\rangle$. Likewise, if we are given that the measured value of $|\psi_1\rangle$ is 1, then the conditional probabilities for the measured values of the second register are the same as the probabilities for the measured values of $U |\psi_2\rangle$. In other words, the conditional probabilities are identical to the probabilities we would obtain if we simply measured $|\psi_1\rangle$ first and decided whether or not to apply U based on the result of that measurement.

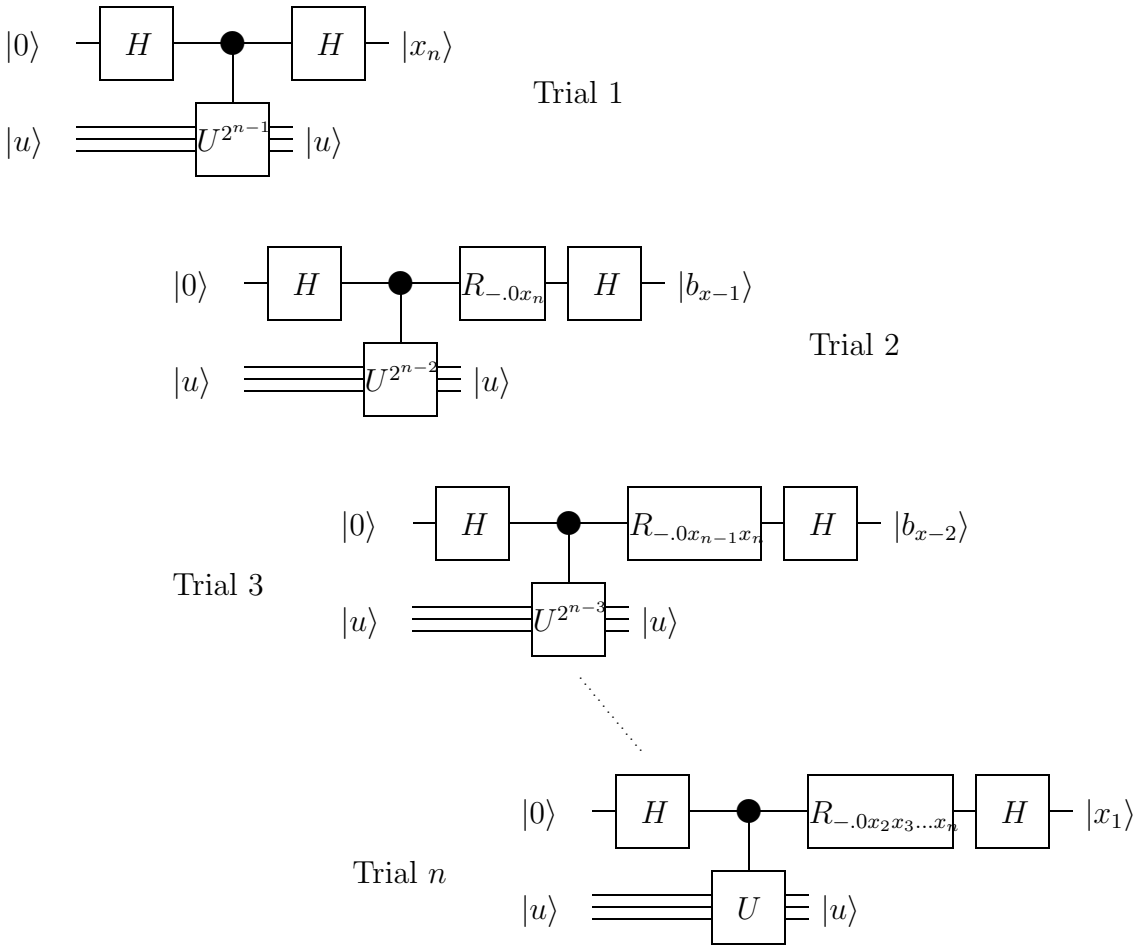

Figure 3.4: QFT-based phase estimation divided into individual trials

Now, in the first half of the quantum phase estimation circuit (Figure 3.2), each of the n qubits in the input register is independent from any of the other qubits in the input register. Since the inverse QFT can be separated into a sequence of individual semiclassical circuits for each qubit, and since each qubit input into the inverse QFT can be generated independently of the state of any other input qubits, the entire QFT-based phase estimation algorithm can be separated into a sequence of individual semiclassical *trials* which share only the eigenstate $|u\rangle$. Also, since the eigenstate $|u\rangle$ is not altered by the controlled-U operation, it may be reused for each individual trial. An example of such a sequence of individual trials is shown in Figure 3.4. For any binary fraction b, R_b represents a phase rotation gate which shifts the phase by an angle of $2\pi b$.

Separating the quantum phase estimation algorithm into a sequence of individual trials has many advantages. First of all, the individual trials are much easier to implement physically than implementing the entire phase estimation circuit all at once: fewer quantum gates need to be applied to each qubit, reducing the probability of errors. Instead of each qubit undergoing a series of controlled phase rotation gates, only the required phase rotation gates are now applied. Also, since the controlled phase rotation gates are replaced by single-qubit phase rotation gates, the need for quantum gates that act on more than one qubit is now limited to the controlled-U gates. This further simplifies the circuit, making it easier to implement and less error-prone. Finally, as the phase estimation procedure is now divided into a sequence of n independent computations, there is greater flexibility in using the algorithm multiple times to obtain more accurate results, as we are now able to repeat trials individually instead of repeating the entire algorithm.

It should be noted that we have increased the number of measurements required by a factor of n. However, each measurement is done on a single qubit rather than

on a quantum register of n qubits. Although it is expected that dividing the task of measuring an n-qubit register into one-qubit states does not require much more effort, this may not be the case for certain physical implementations of quantum computers. In these cases, however, the number of measurements still increases by only a polynomial factor.

3.6 Analysis of Individual Trials

Since we intend to take advantage of the added flexibility that comes with separating the quantum phase estimation algorithm into individual trials, it is useful to establish the probability of error for each trial individually, even though we have already established the probability of obtaining a global error of $\delta = \phi - j/2^n$ as

$$P(\hat{\phi} = j/2^n) = \left(\frac{\sin \pi(2^n \delta)}{2^n \sin \pi\delta} \right)^2$$

in Section 3.4.

Let $x = x_1 x_2 \ldots x_n$ be the binary value so that $x/2^n = (0.x_1 x_2 \ldots x_n)$ is the binary representation of the nearest n-bit estimate of ϕ, or either nearest estimate in case of a tie. Let $\gamma = \phi - x/2^n$, and let $j = j_1 j_2 \ldots j_n$ be the binary value output by the quantum phase estimation algorithm, so that $j/2^n = (0.j_1 j_2 \ldots j_n)$ is the given n-bit value which the quantum phase estimation actually returns. Let $d = d_1 d_2 \ldots d_n$ be the binary representation of $x - j$, where the subtraction is considered modulo 2^n and reduced so that $0 \leq d < 2^n$. As before, let $\delta = \phi - j/2^n = \gamma + d/2^n$ represent the total error between the actual phase and the phase estimate. Note that since $e^{2\pi i} = 1$, we have

$$e^{2\pi i(2^p x)} = e^{2\pi i(x_1 x_2 \ldots x_p . x_{p+1} \ldots x_n)} = e^{2\pi i(0.x_{p+1} \ldots x_n)}.$$

Finally, let $\hat{\phi}_p$ represent the estimate of the p-th bit of ϕ obtained from the corresponding individual trial.

Consider the trial for the least significant qubit, $|x_n\rangle$, which is computed first. We would like to compute the probability that $\hat{\phi}_n = j_n$, recalling that the correct value, if $\hat{\phi}$ is to be the nearest estimate, should be x_n. The first half of the quantum phase estimation circuit for this individual trial prepares a qubit in the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(2^{n-1}\phi)}\,|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i((0.j_n)+2^{n-1}\delta)}\,|1\rangle\right).$$

The inverse QFT applies a Hadamard gate to this state to yield

$$\frac{1 + e^{2\pi i((0.j_n)+2^{n-1}\delta)}}{2}\,|0\rangle + \frac{1 - e^{2\pi i((0.j_n)+2^{n-1}\delta)}}{2}\,|1\rangle,$$

which, since $-1 = e^{2\pi i(0.1)}$, can be rewritten as

$$\frac{1 + e^{2\pi i((0.j_n)+2^{n-1}\delta)}}{2}\,|0\rangle + \frac{1 + e^{2\pi i((0.1)+(0.j_n)+2^{n-1}\delta)}}{2}\,|1\rangle.$$

So, independent of whether j_n is 0 or 1, the final measurement of the n-th qubit yields j_n with probability

$$\begin{aligned} P(\hat{\phi}_n = j_n) &= \left|\frac{1 + e^{2\pi i((0.j_n)+(0.j_n)+2^{n-1}\delta)}}{2}\right|^2 \\ &= \left|\frac{1 + e^{2\pi i(2^{n-1}\delta)}}{2}\right|^2 && \text{(since } (0.j_n)+(0.j_n) = 1\text{)} \\ &= \cos^2 \tfrac{\pi}{2} 2^n \delta. && \text{(by Lemma 3.4.1)} \end{aligned}$$

For the next least significant qubit, $|x_{n-1}\rangle$, we would like to compute the conditional probability $P(\hat{\phi}_{n-1} = j_{n-1} | \hat{\phi}_n = j_n)$. The first half of this individual trial prepares a qubit in the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(2^{n-2}\phi)}\,|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i((0.j_{n-1}j_n)+2^{n-2}\delta)}\,|1\rangle\right).$$

The inverse phase rotation gate R_2^{-1}, which is applied only if $j_n = 1$, shifts the phase of $|1\rangle$ by $-(0.0j_n)$, giving us the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i((0.j_{n-1}j_n)-(0.0j_n)+2^{n-2}\delta)}\,|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i((0.j_{n-1})+2^{n-2}\delta)}\,|1\rangle\right)$$

so that, in a fashion similar to the first individual trial for $|x_n\rangle$, we find that

$$P(\hat{\phi}_{n-1} = j_{n-1}|\hat{\phi}_n = j_n) = \cos^2(\pi 2^{n-2}\delta) = \cos^2\left(\frac{\pi}{2}2^{n-1}\delta\right).$$

In general, for the individual qubit trial for $|x_p\rangle$, we prepare a qubit in the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(2^{p-1}\phi)}\,|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i((0.j_p\ldots j_{n-1}j_n)+2^{p-1}\delta)}\,|1\rangle\right),$$

which is phase-shifted by $-(0.0j_{p+1}j_{p+2}\ldots j_n)$ by the selective application of some of the inverse phase rotation gates R_2^{-1}, R_3^{-1}, ..., R_{n-p+1}^{-1} to yield the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i((0.j_p)+2^{p-1}\delta)}\,|1\rangle\right).$$

Given this state, the probability, P_p, of the quantum phase estimation algorithm getting the correct value for the individual trial for $|x_p\rangle$, given that the algorithm has output only the values $j_n, j_{n-1}, \ldots, j_{p+1}$ so far, is

$$\begin{aligned} P_p &= P(\hat{\phi}_p = j_p|\hat{\phi}_n = j_n, \hat{\phi}_{n-1} = j_{n-1}, \ldots, \hat{\phi}_{p+1} = j_{p+1}) \\ &= \cos^2\left(\frac{\pi}{2}2^p\delta\right). \end{aligned}$$

From this analysis, we can also see that in general, if we start with the state

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

for any t, $0 \leq t < 1$, apply a Hadamard gate, and perform a measurement, we will obtain $|j_p\rangle$ with probability $\cos^2 \pi t$.

We can now express the probability of getting $j/2^n = (0.j_1 j_2 \ldots j_n)$ as the output of the phase estimation algorithm given that the actual phase is ϕ, as

$$P(\hat{\phi} = j/2^n) = P_n P_{n-1} \ldots P_1 = \prod_{p=1}^{n} \cos^2 \frac{\pi}{2}(2^p \delta).$$

This expression can be further reduced by using the following lemma:

Lemma 3.6.1 *For any integer $n \geq 1$,*

$$\prod_{p=1}^{n} \cos^2 \frac{\theta}{2^p} = \left(\frac{\sin \theta}{2^n \sin \frac{\theta}{2^n}} \right)^2$$

Proof We proceed by induction.

For $n = 1$, we have

$$\cos^2 \frac{\theta}{2} = \left(\frac{\sin \theta}{2 \sin \frac{\theta}{2}} \right)^2,$$

which follows immediately from the double angle formula. Now, suppose that for a fixed value of n, we have

$$\prod_{p=1}^{n} \cos^2 \frac{\theta}{2^p} = \left(\frac{\sin \theta}{2^n \sin \frac{\theta}{2^n}} \right)^2.$$

Then

$$\begin{aligned}
\prod_{p=1}^{n+1} \cos^2 \frac{\theta}{2^p} &= \left(\frac{\sin \theta}{2^n \sin \frac{\theta}{2^n}} \right)^2 \cos^2 \frac{\theta}{2^{n+1}} \\
&= \left(\frac{\sin \theta \cos \frac{\theta}{2^{n+1}}}{2^n \sin \frac{\theta}{2^n}} \right)^2 \\
&= \left(\frac{\sin \theta \cos \frac{\theta}{2^{n+1}}}{2^n \cdot 2 \sin \frac{\theta}{2^{n+1}} \cos \frac{\theta}{2^{n+1}}} \right)^2 \\
&= \left(\frac{\sin \theta}{2^{n+1} \sin \frac{\theta}{2^{n+1}}} \right)^2,
\end{aligned}$$

and we are done. □

Finally, we may verify that the phase estimation algorithm consisting of individual semiclassical trials is indeed equivalent to the inverse QFT-based phase estimation algorithm by demonstrating that the probability of obtaining any particular output $j/2^n = (0.j_1 j_2 \ldots j_n)$ given an actual phase of ϕ is the same for both algorithms.

Letting $\theta = 2^n \delta\pi$ in Lemma 3.6.1, we have

$$P(\hat{\phi} = j/2^n) = \prod_{p=1}^{n} \cos^2 \frac{\pi}{2}(2^p \delta).$$

Reversing the indexing of the product, we get

$$\begin{aligned} P(\hat{\phi} = j/2^n) &= \prod_{p=1}^{n} \cos^2 \frac{\pi}{2}(2^{n+1-p}\delta) \\ &= \prod_{p=1}^{n} \cos^2 \frac{\pi(2^n\delta)}{2^p} \\ &= \left(\frac{\sin 2^n \delta\pi}{2^n \sin \delta\pi}\right)^2, \end{aligned}$$

when $\delta \neq 0$. In the other case, when $\delta = 0$, we have

$$P(\hat{\phi} = j/2^n) = \prod_{p=1}^{n} \cos^2 \frac{\pi(2^n\delta)}{2^p} = \prod_{p=1}^{n} \cos^2 0 = 1.$$

This is consistent with the analysis of the QFT-based phase estimation algorithm in Section 3.4, where the phase estimation circuit is not divided into individual trials.

Chapter 4

The Approximate Quantum Fourier Transform

The idea of an approximate quantum Fourier transform was first introduced by Don Coppersmith [5], who proposed that if the QFT could be sped up at the expense of the accuracy of the algorithm, the trade-off may yield an improvement in the phase estimation algorithm. This trade-off would increase the number of times the QFT was used to achieve a certain level of accuracy, but also reduce the complexity of each application of the QFT.

The AQFT addresses a number of issues with the QFT as it is used in the phase estimation algorithm. The most important issue is that using the QFT requires the efficient physical implementation of the series of phase rotation gates, R_m.[1] The AQFT is also a natural generalization of both the QFT-based phase estimation

[1]Since this set of gates is infinite, the Solovay-Kitaev theorem does not apply, i.e., there is no upper bound on the number of gates required to implement each successive phase rotation gate, R_m.

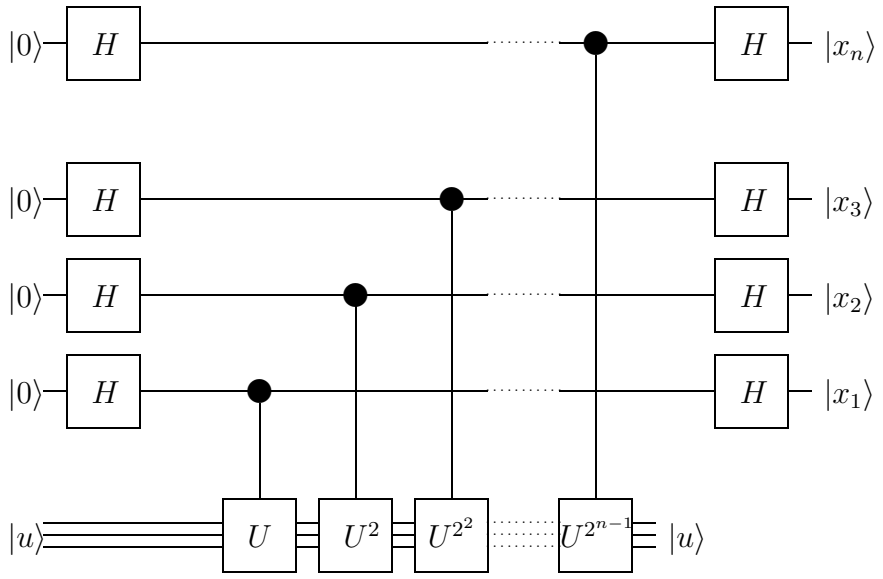

Figure 4.1: $AQFT_1$: Kitaev's circuit for solving the phase estimation problem

algorithm discussed in Chapter 3 and Kitaev's phase estimation algorithm, which will be introduced in the following section.

4.1 Kitaev's Algorithm and Variations

Kitaev originally developed a phase estimation algorithm as a component of an algorithm to solve the *Abelian Stabilizer* problem[9], which will be discussed in Chapter 6. Kitaev's phase estimation algorithm is similar in structure to the QFT-based phase estimation algorithm, but does not perform any phase rotation operations. As in the first half of the QFT-based phase estimation algorithm, a register containing the state

$$|\psi\rangle = \frac{(|0\rangle + e^{2\pi i(2^{n-1}\phi)}|1\rangle) \otimes \cdots \otimes (|0\rangle + e^{2\pi i(2^1\phi)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(2^0\phi)}|1\rangle)}{\sqrt{2^n}}$$

is prepared. However, instead of applying the inverse QFT circuit, Kitaev's algorithm applies the n-qubit Hadamard gate to this register, which is essentially a 1-qubit Hadamard gate applied to each qubit in the register. In general, applying the Hadamard gate to a qubit in the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\theta}|1\rangle)$, for $\theta \in [0, 1)$, yields the state

$$\frac{1 + e^{2\pi i\theta}}{2}|0\rangle + \frac{1 - e^{2\pi i\theta}}{2}|1\rangle$$

which, if measured, yields 0 with probability $\cos^2 \pi\theta$, and 1 with probability $\sin^2 \pi\theta$, by Lemma 3.4.1. In other words, a measurement of 0 indicates that θ is more likely to be closer to 0 (modulo 1), while a measurement of 1 indicates that θ is more likely to be closer to $\frac{1}{2}$ (modulo 1).[2]

Since for each integer p, $1 \leq p \leq n$, the first half of Kitaev's algorithm prepares a qubit in the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(2^{p-1}\phi)}|1\rangle)$, the measurements obtained from the entire algorithm do not correspond directly to the values of $2^{p-i}\phi$ as they did for the QFT-based phase estimation algorithm. In Kitaev's original algorithm, as given in [9], each trial is repeated a number of times to obtain the ratio of measurements resulting in $|0\rangle$ to measurements resulting in $|1\rangle$. This ratio is used as an estimate for the fractional part of $2^{p-i}\phi$. We require $O(n)$ measurements of each trial to achieve a given fixed error tolerance.

However, we may also consider a reduced version of Kitaev's algorithm, where we do not require that the trials be repeated in this manner. Instead, we take $\hat{\phi}$ to be the *maximum likelihood estimate* (MLE) of ϕ. The remainder of the analysis in this chapter will refer to the reduced version given here.

Given a particular set of measurements, $x_1, x_2, \ldots x_n$, obtained from the quan-

[2]In this context, "modulo 1" simply refers to the fractional part of a quantity, i.e., $\theta - \lfloor\theta\rfloor$. Distances modulo 1 are computed circularly, so that 9/10 is closer to 0 than it is to 1/2, for example.

tum circuit used in Kitaev's algorithm, which we will call $AQFT_1$, the *likelihood function* of ϕ, $L(\phi)$, is defined as the probability of the $AQFT_1$ circuit obtaining that particular set of measurements given that the initial phase associated with the unitary operator U and the eigenstate $|u\rangle$ is ϕ. The MLE is simply the value $\hat{\phi}$ which maximizes $L(\phi)$. The advantage of using the MLE is that it is the estimator of minimum variance which, in effect, means that it minimizes the expected error. Note that the output of the QFT-based phase estimation algorithm is effectively equivalent to the MLE, as the output of the inverse QFT coincides with the input phase angle of highest likelihood.

To compute the MLE, we begin by defining $P(\phi, X)$ to be the probability of obtaining the result $X = (x_1, x_2, \ldots, x_n)$ with the $AQFT_1$ circuit given the initial phase ϕ. Like the QFT-based phase estimation algorithm, the $AQFT_1$ circuit can be considered as a collection of individual trials, corresponding to each output bit, x_p, $1 \leq p \leq n$. The trial corresponding to x_p prepares the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(2^{p-1}\phi)}|1\rangle)$, applies a Hadamard gate and then a measurement, giving us the answer 0 with probability $\cos^2 2^{p-1}\phi\pi$, and the answer 1 with probability $\sin^2 2^{p-1}\phi\pi$. Equivalently, we may also say that the probability of obtaining a given output bit x_p is $\cos^2 \frac{\pi}{2}(2^p\phi + x_p)$.

Since the n individual trials in $AQFT_1$ are independent of each other, the probability of obtaining a particular result $X = (x_1, x_2, \ldots, x_n)$ given an initial phase of ϕ is

$$P(\phi, X) = \prod_{p=1}^{n} \cos^2 \frac{\pi}{2}(2^p\phi + x_p).$$

Now, we would like to find the value of ϕ which maximizes $P(\phi, X)$ for a given value of $X = (x_1, x_2, \ldots, x_n)$.

In general, it would be difficult to find the global maxima of such functions

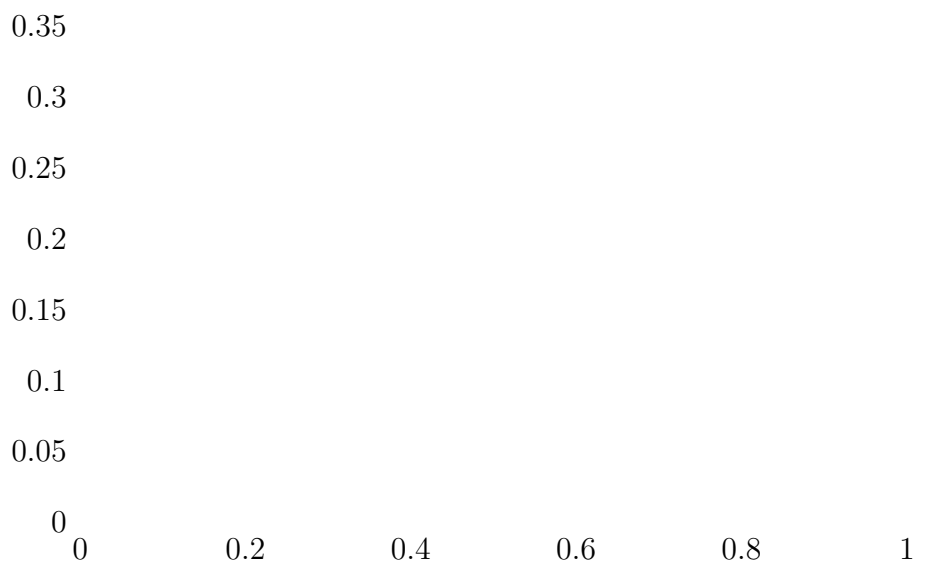

Figure 4.2: $P(\phi, (0, 1, 1, 1, 0))$

analytically, since given the results of n individual trials, $P(\phi, X)$ is a non-negative function with at least 2^{n-1} zeroes, and since $\cos^2 \frac{\pi}{2}(2^n\phi + x_n)$ is a factor of $P(\phi, X)$, thus having at least 2^{n-1} local maxima. However, it is possible to obtain the MLE numerically by taking advantage of some of the properties of $P(\phi, X)$. If we assume that the results of the individual trials, X, actually contain information about ϕ, then we should expect $P(\phi, X)$ to have a few large maxima and many smaller maxima, such as the example in Figure 4.2.

Given a function of the form

$$P(\phi, X) = \prod_{p=1}^{n} \cos^2 \frac{\pi}{2}(2^p\phi + x_p),$$

suppose that we can find a small function,

$$P_2(\phi, X) = \epsilon \left| \cos \frac{\pi}{2}(2^n\phi + x_n) \right|$$

so that $P(\phi, X) \leq P_2(\phi, X)$ whenever $\phi \in ([0, 1] \setminus I)$, where I is a small collection

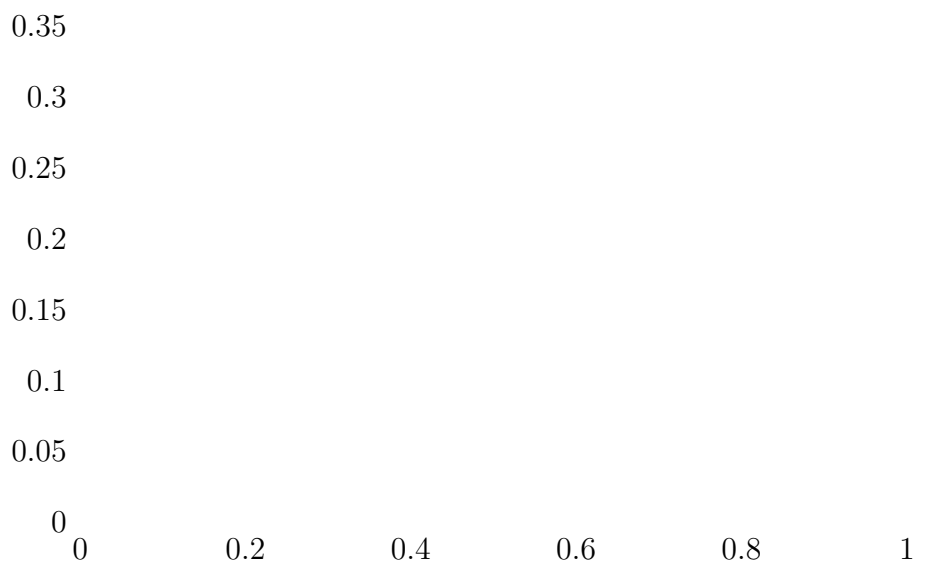

Figure 4.3: $P(\phi, (0,1,1,1,0))$, with $P_2 = \frac{1}{10}\left|\cos\frac{\pi}{2}(2^5\phi)\right|$

of intervals of the form $[\frac{r}{2^{n-1}}, \frac{r+1}{2^{n-1}})$, where r is an integer such that $0 \leq r < 2^{n-1}$. Then we may restrict our search for global maxima to those intervals in I, so long as these global maxima are larger than ϵ.

Finding such a representation may be done recursively, by first finding a representation of the function

$$P(\phi, X_{n-1}) = \prod_{p=1}^{n-1} \cos^2 \frac{\pi}{2}(2^p\phi + x_p),$$

where X_p represents the results of the first p individual trials, $X_p = (x_1, x_2, \ldots, x_p)$, obtaining

$$P_2(\phi, X_{n-1}) = \epsilon_{n-1}\left|\cos\frac{\pi}{2}(2^{n-1}\phi + x_{n-1})\right|,$$

and a set of intervals, I_{n-1}. With such a representation, to find a suitable $P_2(\phi, X)$, it is sufficient to find an ϵ such that

$$P_2(\phi, X) \;=\; \epsilon\left|\cos\frac{\pi}{2}(2^n\phi + x_n)\right|$$

$$\begin{aligned} &\geq P_2(\phi, X_{n-1}) \cos^2 \frac{\pi}{2}(2^n\phi + x_n) \\ &= \epsilon_{n-1} \left|\cos \frac{\pi}{2}(2^{n-1}\phi + x_{n-1})\right| \cos^2 \frac{\pi}{2}(2^n\phi + x_n) \end{aligned}$$

so that we have

$$\begin{aligned} P(\phi, X) &= P(\phi, X_{n-1}) \cos^2 \frac{\pi}{2}(2^n\phi + x_n) \\ &\leq P_2(\phi, X_{n-1}) \cos^2 \frac{\pi}{2}(2^n\phi + x_n) \\ &\leq P_2(\phi, X) \end{aligned}$$

for all $\phi \in ([0,1] \setminus I_{n-1})$.

However, substituting $\psi = 2^{n-1}\phi$, we see that this task is equivalent to finding ϵ so that

$$\frac{\epsilon}{\epsilon_{n-1}} \left|\cos \frac{\pi}{2}(2\psi + x_n)\right| \geq \left|\cos \frac{\pi}{2}(\psi + x_n)\right| \cos^2 \frac{\pi}{2}(2\psi + x_n).$$

The ratios $\frac{\epsilon}{\epsilon_{n-1}}$ may be precomputed. We can extend this idea of building ϵ recursively using precomputed constants by finding constants $\frac{\epsilon}{\epsilon_{n-m}}$ which relate $P(\phi, X_n)$ to $P(\phi, X_{n-m})$ for any integer m up to a preset upper bound. If we set the upper bound of m to be $O(\log n)$, then precomputing the ratios $\frac{\epsilon}{\epsilon_{n-m}}$ for all possible combinations of the bits $x_{n-m}, x_{n-m+1}, \ldots, x_n$ will be a polynomial-time computation with respect to n.

Once we obtain a new ϵ, we must also update the set of intervals, I. Note that each interval in the collection I_{n-1} gives us two intervals in I, since each interval in I is half the size of an interval in I_{n-1}. However, we may check each interval in I_{n-1} and discard those which are no longer necessary. In other words, for each interval in I, we can check to see if $P(\phi, X) \leq P_2(\phi, X)$, and discard it if it is. If the amount of information about ϕ embedded in results of the $AQFT_1$ circuit are sufficient to distinguish ϕ to within a small number of values, then we would expect the number of intervals in the collection I to remain small.

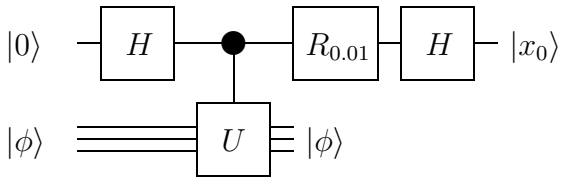

Figure 4.4: Additional trial for distinguishing ϕ and $1-\phi$

To start off this recursive procedure, we pick a threshold $T = O(\log n)$ and compute the first ϵ_T using numerical methods. For each interval $[\frac{r}{2^T}, \frac{r+1}{2^T})$, we get a lower bound on ϵ_T to guarantee that we have $P(\phi, X_T) \leq P_2(\phi, X_T)$ on that interval. We ignore the highest $O(\log n)$ of these lower bounds, adding the corresponding intervals to the collection I, and take the next highest lower bound as ϵ_T. If we select a value of T that is too small, then at some point during the algorithm, the set I will become empty. This means that we no longer have sufficient information about $P(\phi, X)$ to find the maximum. To fix this problem, we need to restart the algorithm with a larger value of T.

Finally, note that since $\cos^2 \frac{\pi}{2}(2^p\phi + x_p)$ is always an even function, the individual trials used in the $AQFT_1$ circuit cannot distinguish between the values ϕ and $1-\phi$. As a result, the function $P(\phi, X)$ will have two global maxima, resulting in two MLEs. In order to distinguish which of the two possible values is the correct value of ϕ, we perform one additional individual trial, in which a phase rotation of $\pi/2$ is applied before the final Hadamard gate and measurement, as in Figure 4.4. With this phase rotation, the intermediate state

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\theta}\,|1\rangle)$$

is transformed into the state

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(\theta+\frac{1}{4})}\,|1\rangle),$$

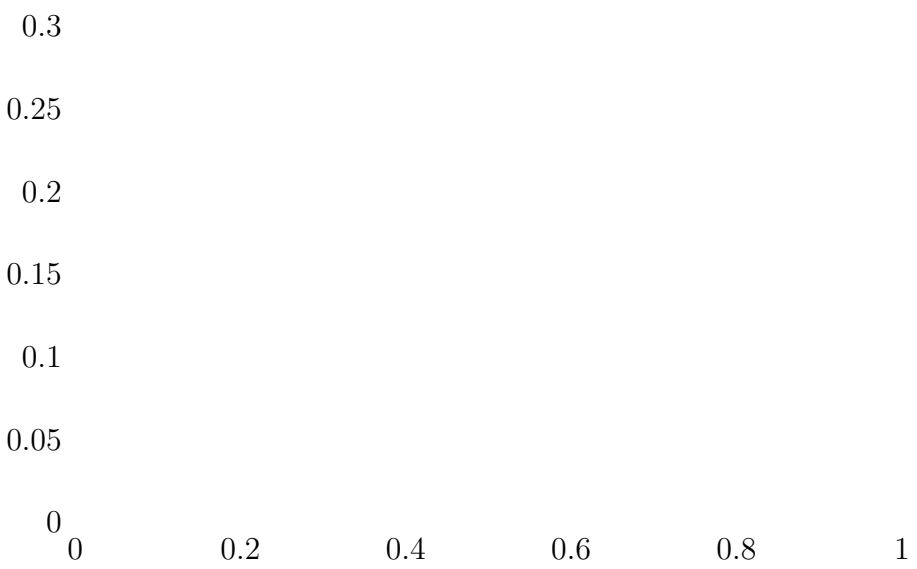

Figure 4.5: $\cos^2 \frac{\pi}{2}(2\phi + \frac{1}{2}) \cdot P(\phi, (0, 1, 1, 1, 0))$

and becomes

$$\frac{1 + e^{2\pi i(\theta + \frac{1}{4})}}{2} |0\rangle + \frac{1 - e^{2\pi i(\theta + \frac{1}{4})}}{2} |1\rangle$$

when the last Hadamard gate is applied. Upon measurement, we obtain 0 with probability $\cos^2 \pi(\theta + \frac{1}{4})$, and 1 with probability $\sin^2 \pi(\theta + \frac{1}{4})$. Equivalently, we may say that we obtain the output bit x_0 with probability $\cos^2 \frac{\pi}{2}(2\theta + \frac{1}{2} + x_0)$. Note that the probabilities we obtain when we set $\theta = \phi$ and when we set $\theta = 1 - \phi$ are necessarily different, unless $\phi = 0$, in which case ϕ and $1 - \phi$ are equivalent (modulo 1). This allows us to distinguish between these two possibilities.

We can also easily incorporate this additional individual trial into the calculation of the MLE by including a factor of $\cos^2 \frac{\pi}{2}(2\phi + \frac{1}{2} + x_0)$ when computing the probability function for obtaining the observed result given a particular set of measurements. For example, in Figure 4.5, the probability function is given for the same set of measurements as in Figure 4.2, but with an additional measurement

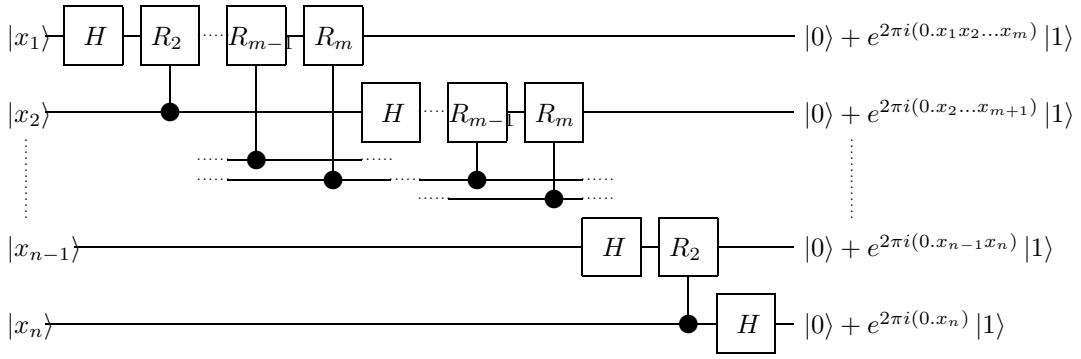

Figure 4.6: A quantum circuit implementing the AQFT

for the extra trial, which gives us a clear global maximum for the MLE.

4.2 The Approximate QFT

From the point of view of building quantum circuits, the approximate quantum Fourier transform is a simple and elegant solution to the problem of efficiently and accurately implementing the smallest phase rotation gates required by the QFT. Instead of performing all phase rotations R_p that are used in the QFT algorithm, we set a lower limit on the amount of phase shifted by any particular gate and ignore any phase rotation gates that do not shift by at least that amount. In other words, given a positive integer threshold, m, we will define the approximate QFT circuit $AQFT_m$ to be the circuit formed from the QFT, except that phase rotation gates R_p are ignored whenever we have $m > p$.

Although the approximate quantum Fourier transform is simple to describe as a quantum circuit, it is also useful to have a closed-form classical expression for the operation. We can start by observing that while the QFT outputs the state

$$\frac{\left(|0\rangle + e^{2\pi i(0.x_n)}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i(0.x_{n-1}x_n)}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i(0.x_1x_2\ldots x_n)}|1\rangle\right)}{\sqrt{2^n}}$$

when given the n-qubit basis state $|x\rangle = |x_1 x_2 \dots x_n\rangle$ as input, the approximate QFT outputs the state

$$\frac{\left(|0\rangle + e^{2\pi i(0.x_n)}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i(0.x_2 x_3 \dots x_{m+1})}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i(0.x_1 x_2 \dots x_m)}|1\rangle\right)}{\sqrt{2^n}},$$

where any phase rotation smaller than $2\pi i(2^{-m})$ is ignored altogether so that each of the binary fractions is truncated at the m-th fractional bit.

Now, consider the state $|k\rangle$ where $0 \leq k \leq 2^n - 1$, and let $k_{n-1} \dots k_1 k_0$ be the binary representation of k. Then, if we define $x_p = 0$ for convenience whenever $p > n$, we can express the probability amplitude of $|k\rangle$ in the output state $AQFT_m(|x_1 x_2 \dots x_n\rangle)$ as

$$\begin{aligned}
\prod_{p=0}^{n-1} e^{2\pi i k_p (0.x_{p+1} x_{p+2} \dots x_{p+m})} &= \exp\left(\sum_{p=0}^{n-1} 2\pi i k_p (0.x_{p+1} x_{p+2} \dots x_{p+m})\right) \\
&= \exp\left(\sum_{p=0}^{n-1} 2\pi i k_p \left(\sum_{r=1}^{m} x_{p+r} 2^{-r}\right)\right) \\
&= \exp\left(2\pi i \sum_{r=1}^{m} \sum_{p=0}^{n-1} k_p x_{p+r} 2^{-r}\right) \\
&= \exp\left(2\pi i \sum_{r=1}^{m} \sum_{q=r}^{n+r-1} k_{q-r} x_q 2^{-r}\right).
\end{aligned}$$

But since $x_q = 0$ whenever $q > n$, this can be rewritten as

$$\exp\left(2\pi i \sum_{r=1}^{m} \sum_{q=r}^{n} k_{q-r} x_q 2^{-r}\right),$$

so that we have

$$AQFT_m(|x\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(2\pi i \sum_{r=1}^{m} \sum_{q=r}^{n} k_{q-r} x_q 2^{-r}\right) |k\rangle.$$

Now, observe that $AQFT_n$ should be equivalent to the circuit QFT, as no phase rotation gates are removed to make this approximate QFT circuit. So, we would

expect that if we substitute $m = n$ into the expression we derived for $AQFT_m(|x\rangle)$ we should obtain the value of $QFT(|x\rangle)$ given in Section 3.2. To see this, we will express the Fourier transform in terms of the binary representations of j and k. First take the discrete Fourier transform for $N = 2^n$,

$$QFT(|x\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(\frac{2\pi i k x}{2^n}\right) |k\rangle .$$

Letting $k_{n-1}k_{n-2}\ldots k_1k_0$ and $0.x_1x_2\ldots x_n$ be the binary representations of k and $x/2^n$, respectively, we have the product

$$\frac{kx}{2^n} = k\frac{x}{2^n} = \sum_{p=0}^{n-1}\sum_{q=1}^{n} k_p x_q 2^{p-q},$$

obtaining

$$QFT(|x\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(2\pi i \sum_{p=0}^{n-1}\sum_{q=1}^{n} k_p x_q 2^{p-q}\right) |k\rangle ,$$

and since integer multiples of $2\pi i$ do not affect the exp function, we can ignore any terms where $p \geq q$ to get

$$\begin{aligned} QFT(|x\rangle) &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(2\pi i \sum_{p=0}^{n-1}\sum_{q=p+1}^{n} k_p x_q 2^{p-q}\right) |k\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \exp\left(2\pi i \sum_{r=1}^{n}\sum_{q=r}^{n} k_{q-r} x_q 2^{-r}\right) |k\rangle \\ &= AQFT_n(|x\rangle). \end{aligned}$$

On the other end, the approximate QFT circuit $AQFT_1$ is simply the n-qubit Hadamard transform, as it is the result of stripping all phase rotation gates from a QFT circuit. This is the circuit used in algorithms in Section 4.1 above, without the extra trial used to distinguish between eigenvalues ϕ and $1 - \phi$ is not present in $AQFT_1$.

For approximate QFT circuits between these two extremes, the arguments for using the method of finding maximum likelihood estimates given in Section 4.1 still hold. However, this is not inconsistent with the analysis of the QFT circuit given in Chapter 3, since the QFT circuit is simply the $m = n$ case of the $AQFT_m$ circuit. The MLE in the $AQFT_n$ case always coincides with the binary string obtained upon measurement, so long as we assume that the phases that are input into the algorithm, ϕ, are uniformly distributed between 0 and 2π. This is a consequence of Bayes' Theorem, which states that

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}.$$

We can take A to be the probability of measuring a particular value, $j/2^n$, and B to be the probability that $\hat{\phi}$ is the multiple of 2^{-n} nearest to the actual value of ϕ. The assumption that answers to phase estimation problems are uniformly distributed implies that $P(A)$ and $P(B)$ are constants with respect to n, j and ϕ, so that $P(B|A)$ is proportional to $P(A|B)$. This means that the value of $\hat{\phi}$ that maximizes $P(A|B)$ is also the one that maximizes $P(B|A)$, which is the probability that $j/2^n$ is measured given a fixed value for ϕ. Thus, we may find maximum likelihood estimates $\hat{\phi}$ by finding values of ϕ for which $j/2^n$ is the most probable measured output. In Chapter 3, we showed that the probability of measuring a particular value $j/2^n$ at the end of the QFT circuit is

$$P(\delta) = \left(\frac{\sin \pi(2^n\delta)}{2^n \sin \pi\delta}\right)^2 = \prod_{p=1}^{n} \cos^2 \frac{\pi}{2}(2^p\delta),$$

where $\delta = \phi - j/2^n$ is the difference between the correct value and the measured value. When $\delta = 0$, the probability given by the right-hand expression is 1, which is also the limit of the left-hand expression as δ approaches 0, since

$$\lim_{\delta\to 0}\left(\frac{\sin \pi(2^n\delta)}{2^n \sin \pi\delta}\right)^2 = \lim_{\delta\to 0}\left(\frac{\sin \pi(2^n\delta)}{\pi(2^n\delta)}\right)^2 \left(\frac{\pi\delta}{\sin \pi\delta}\right)^2 = 1 \cdot 1 = 1.$$

Since $P(\delta)$ cannot exceed 1, $\delta = 0$ gives us a maximum likelihood estimate for ϕ. To show that the maximum likelihood estimate is unique in this case, consider all values of $\delta \in (-\frac{1}{2}, \frac{1}{2}]$ such that

$$\prod_{p=1}^{n} \cos^2 \frac{\pi}{2}(2^p \delta) = 1.$$

In order for this to be true, we must have $\cos^2 \frac{\pi}{2}(2^p\delta) = 1$ for each integer p from 1 to n. Specifically, for $p = 1$, the only value of δ for which $\cos^2 \pi\delta = 1$ is $\delta = 0$. This means that our likelihood function achieves its maximum value only when $\delta = 0$, and that the QFT-based phase estimation algorithm does indeed return the only maximum likelihood estimate.

4.3 Analysis of the Approximate QFT

As with the regular QFT circuit, we may perform an approximate QFT as a series of individual trials. In the $AQFT_m$ circuit, given a phase ϕ for which we are to find an approximation $\hat{\phi} = 0.x_1x_2 \ldots x_n$, the individual trial for bit x_p starts by preparing the state $|0\rangle + e^{2\pi i(2^{p-1}\phi)} |1\rangle$. It then performs a phase rotation which multiplies the probability amplitude of the state $|1\rangle$ by a factor of $e^{-2\pi i\chi_p}$, where χ_p is the phase rotation which corresponds to the amount performed by the QFT circuit, $(0.0x_{i+1}x_{i+2} \ldots x_n)$, but without any phase rotations which are smaller than the given threshold m. So, we have $\chi_p = (0.0x_{p+1} \ldots x_{p+m-1})$ for the x_p where $p \leq n - m$, and $\chi_p = (0.0x_{p+1} \ldots x_n)$ otherwise.

We would now like to determine the probability that the output of the $AQFT_m$ circuit, $\hat{\phi} = (0.x_1x_2 \ldots x_n)$, is the nearest integer multiple of $1/2^n$ when interpreted directly as a binary fraction, as we did for the $QFT = AQFT_n$ circuit. In Section 3.6, we established that given a state of the form $|0\rangle + e^{2\pi i((0.x_p)+\delta_p)} |1\rangle$, applying

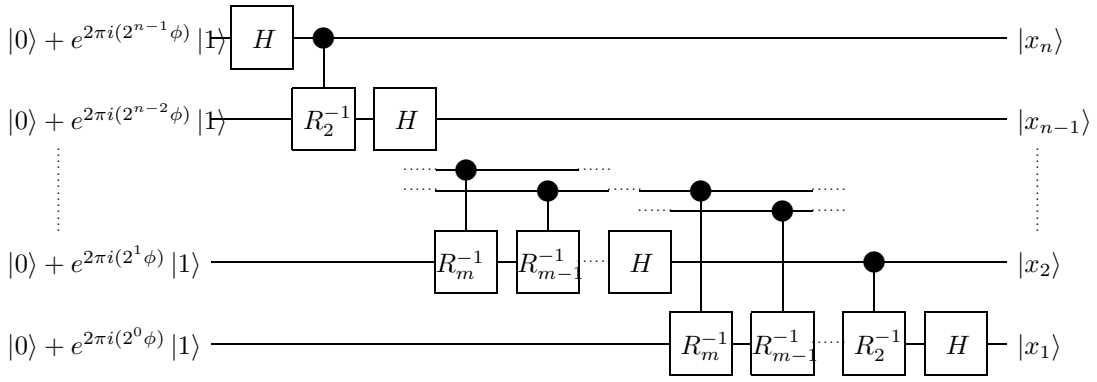

Figure 4.7: The second half of the AQFT-based quantum phase estimation circuit

the Hadamard gate, and measuring, yields $|x_p\rangle$ with probability $\cos^2(\pi\delta_p)$. For the $AQFT_m$ circuit, determining bounds for error amount δ_p requires looking at the two cases separately.

In the first case, where we have $p + m - 1 < n$ and $\chi_p = (0.0x_{p+1} \ldots x_{p+m-1})$ for the individual trial that computes bit x_p, δ_p will include the phase rotation amount that was not applied by the $AQFT_m$ trial that would have been applied by the corresponding QFT trial, and the difference between ϕ and its nearest integer estimate, amplified by a factor of 2^p by the $AQFT_m$ circuit. In other words, we have

$$\delta_p = 2^{-m}(0.x_{p+m}x_{p+m+1} \ldots x_n) + 2^{p-1}\delta,$$

where $\delta = \phi - \hat{\phi}$. Since we are assuming that $\hat{\phi}$ is the nearest estimate of ϕ, we also have $-2^{p-n-1} \leq 2^{p-1}\delta \leq 2^{p-n-1}$. Now, since $(0.x_{p+m} \ldots x_n)$ reaches its minimum value when all the bits are 0, and its maximum value when all its bits are 1, we have

$$0 \leq 2^{-m}(0.x_{p+m} \ldots x_n) \leq 2^{-m}(1 - 2^{p+m-n-1}) = 2^{-m} - 2^{p-n-1}.$$

Adding the two inequalities together, we obtain $-2^{-m} < -2^{-n-1} \leq \delta_p \leq 2^{-m}$. This means that in the individual trial for bit x_p, the final measurement would yield $|x_p\rangle$

with a probability of at least $\cos^2(\pi 2^{-m})$ in this case.

In the second case, where $p \geq n-m+1$, the individual bit trials are identical to their counterparts in the QFT circuit, and so the same lower bound for the error probability of $\cos^2(\frac{\pi}{2}2^p\delta)$ applies. Since $|\delta| \leq 2^{-n-1}$, we have an error probability lower bound of $\cos^2(\frac{\pi}{2}2^{p-n-1})$. As with the regular QFT circuit, the probability P of getting the correct output with the $AQFT_m$ circuit is simply the product of each individual bit trial being correct. Putting these lower bounds together, we can give a lower bound for P:

$$\begin{aligned}
P &= P_n P_{n-1} \ldots P_1 \\
&\geq \prod_{p=n-m+1}^{n} \cos^2\left(\frac{\pi}{2}2^{p-n-1}\right) \prod_{p=1}^{n-m} \cos^2(\pi 2^{-m}) \\
&= \left(\frac{\sin\frac{\pi}{2}}{2^m \sin\frac{\pi/2}{2^m}}\right)^2 \left(\cos^2(\pi 2^{-m})\right)^{n-m} \\
&= \left(\frac{1}{2^m \sin\frac{\pi/2}{2^m}}\right)^2 \left(\cos^2(\pi 2^{-m})\right)^{n-m} \\
&\geq \left(\frac{1}{\pi/2}\right)^2 \left(\cos^2(\pi 2^{-m})\right)^{n-m} \\
&= \frac{4}{\pi^2}\left(\cos^2(\pi 2^{-m})\right)^{n-m}.
\end{aligned}$$

It is clear that if m grows too slowly in comparison to n, then the expression will approach 0 asymptotically as n approaches ∞, making this bound useless. Therefore, we would like to have a lower bound in terms of n for the values of m which would make this bound non-trivial. Suppose that $m \geq \log_2 n + 2$, so that $2^m \geq 4n$. Then, we would have

$$\begin{aligned}
\frac{4}{\pi^2}\left(\cos^2(\pi 2^{-m})\right)^{n-m} &\geq \frac{4}{\pi^2}\left(\cos^2\left(\frac{\pi}{4n}\right)\right)^{n-m} \\
&\geq \frac{4}{\pi^2}\left(\cos^2\left(\frac{\pi}{4n}\right)\right)^{n}.
\end{aligned}$$

Using l'Hôpital's rule, we may evaluate the limit of this expression as n tends to infinity. We have

$$\begin{aligned}
\lim_{n\to\infty} \frac{4}{\pi^2}\left(\cos^2\left(\frac{\pi}{4n}\right)\right)^n &= \lim_{n\to\infty} \frac{4}{\pi^2}\exp\left(2n\ln\cos\left(\frac{\pi}{4n}\right)\right) \\
&= \lim_{n\to\infty} \frac{4}{\pi^2}\exp\left(\frac{1}{1/2n}\ln\cos\left(\frac{\pi}{4n}\right)\right) \\
&= \lim_{n\to\infty} \frac{4}{\pi^2}\exp\left(\frac{1}{-1/2n^2}\left(\frac{\pi}{4n^2}\tan\left(\frac{\pi}{4n}\right)\right)\right) \\
&= \lim_{n\to\infty} \frac{4}{\pi^2}\exp\left(-\frac{\pi}{2}\tan\left(\frac{\pi}{4n}\right)\right) \\
&= \frac{4}{\pi^2}\exp\left(-\frac{\pi}{2}\tan(0)\right) \\
&= \frac{4}{\pi^2}.
\end{aligned}$$

This means that asymptotically, for $m \geq \log_2 n+2$, the lower bound for the accuracy of the $AQFT_m$ circuit approaches the lower bound for the accuracy of the full QFT. This is significant, since the full QFT on an n-qubit register requires implementing phase rotation gates which are able to accurately adjust the relative phase of a state by a factor of at least $e^{2\pi i/2^n}$. By the Solovay-Kitaev theorem, given a fixed universal set of quantum gates, implementing such a gate with accuracy on that order requires at least $O(n^2)$ gates. Since the QFT circuit itself contains $O(n^2)$ such gates, the total number of gates required becomes $O(n^4)$, which is unacceptable for applications such as integer factorization, where values of n are typically above 1000. In the $AQFT_m$ circuit, the smallest phase rotation gate only needs to adjust the relative phase of a state by a factor of $e^{2\pi i/2^m}$, meaning that we would require $O(m^2)$ gates to implement it. When $m \geq \log_2 n + 2$, we have $O(m^2) = O(\log^2 n)$. In addition, since the $AQFT_m$ circuit uses only $O(nm)$ gates, we have a total of $O(n \log n \log^2 n)$, which is much more feasible for practical applications. The fact that this improvement is achieved with no significant loss of accuracy indicates that much larger applications of quantum phase estimation may be feasible.

We can also establish a lower bound for P when we have $m \geq \log_2 n + 2$ for some fixed value of n, as an asymptotic bound gives no information about the value of P for practical values of n. We have

$$\begin{aligned} P &\geq \frac{4}{\pi^2}\left(\cos^2\left(\frac{\pi}{4n}\right)\right)^{n-m} \\ &= \frac{4}{\pi^2}\left(1-\sin^2\left(\frac{\pi}{4n}\right)\right)^{n-m} \\ &\geq \frac{4}{\pi^2}\left(1-(n-m)\sin^2\left(\frac{\pi}{4n}\right)\right), \end{aligned}$$

using the Bernoulli inequality, which states that $(1+t)^c \geq 1+ct$ whenever $t > -1$ and $c \geq 0$. Since $x \geq \sin x$ when $x \geq 0$, we have

$$\begin{aligned} P &\geq \frac{4}{\pi^2}\left(1-(n-m)\sin^2\left(\frac{\pi}{4n}\right)\right) \\ &\geq \frac{4}{\pi^2}\left(1-(n-m)\left(\frac{\pi}{4n}\right)^2\right) \\ &= \frac{4}{\pi^2}-\frac{1}{4n}+\frac{m}{4n^2} \\ &\geq \frac{4}{\pi^2}-\frac{1}{4n}. \end{aligned}$$

This indicates that even for smaller values of n, the probability of the $AQFT_m$ circuit returning the best estimate of ϕ is high enough to make it a practical alternative to using the full QFT circuit.

Note that in the case of the regular QFT circuit, when we have $m = n$, we obtain the same bound for P:

$$P \geq \frac{4}{\pi^2}-\frac{1}{4n}+\frac{m}{4n^2} = \frac{4}{\pi^2},$$

as the one established in Chapter 3. Also note that since we stipulated that $m \geq \log_2 n + 2$, this result is only meaningful if $n \geq 4$. This allows us to establish the fixed bound

$$P \geq \frac{4}{\pi^2}-\frac{1}{16}$$

for $AQFT_m$ circuits where $m \geq \log_2 n + 2$.

Finally, we note that this result is much stronger than the bound of

$$P \geq \frac{8}{\pi^2} \sin^2 \left(\frac{\pi m}{4n} \right)$$

previously established by Barenco, *et al.* in [2]. We can establish this from the bound

$$P \geq \frac{4}{\pi^2} \left(1 - (n-m) \left(\frac{\pi}{4n} \right)^2 \right).$$

Since $\frac{4}{\pi^2} \geq \frac{1}{4n}$ for all $n \geq 4$, we have

$$\begin{aligned} P &\geq \frac{4}{\pi^2} \left(1 - (n-m) \left(\frac{\pi}{4n} \right)^2 \right) \\ &= \frac{4}{\pi^2} - (n-m) \frac{1}{4n^2} \\ &\geq \frac{4}{\pi^2} - (n-m) \frac{4}{\pi^2 n} \\ &= \frac{4m}{\pi^2 n}. \end{aligned}$$

Consider $\frac{4m}{\pi^2 n}$ as a real function of m, with n as a fixed constant. We see that on the interval $m \in [0, n]$, it represents a linear function from $(m, f(m)) = (0, 0)$ to $(m, f(m)) = (n, \frac{4}{\pi^2})$. However, on the interval $m \in [0, n]$, the function

$$\frac{8}{\pi^2} \sin^2 \left(\frac{\pi m}{4n} \right)$$

is a convex function which shares the same two endpoints. Thus, we can conclude that

$$\frac{4m}{\pi^2 n} \geq \frac{8}{\pi^2} \sin^2 \left(\frac{\pi m}{4n} \right)$$

on the interval $m \in [0, n]$, as desired. Furthermore, since we have

$$\lim_{n \to \infty} \frac{8}{\pi^2} \sin^2 \left(\frac{\pi m}{4n} \right) = 0$$

when m is fixed at $\log_2 n + 2$, it was not previously known that the $AQFT_m$ circuit would perform so well for large n.

4.4 An MLE for the Approximate QFT

Having a lower bound for the probability that the answer output by the $AQFT_m$ circuit is correct does not necessarily give any indication as to whether the output is a good estimator for the value of ϕ. We also need to find the MLE.

The analysis in the previous section suggests that when we have $m \geq \log_2 n + 2$, the MLE is very close to the value constructed directly from the output of the inverse AQFT circuit. This section establishes this fact. In the other case, when $m < \log_2 n + 2$, finding the MLE would probably require analysis similar to the post-processing algorithm used with the $AQFT_1$ circuit, as described in Section 4.1.

As before, let $\delta = \phi - \hat{\phi}$ be the difference between the actual value ϕ, and the value output by the $AQFT_m$ circuit. Let $P(\delta)$ be the likelihood of δ being the error given that $\hat{\phi}$ is the output value. Then, the MLE of ϕ is $\hat{\phi} + \hat{\delta}$, where $\hat{\delta}$ is the value of δ which maximizes $P(\delta)$. Again, let δ_p be the error amount between the phase amount in trial p, and the correct amount, $(0.x_p)$.

Since we are considering only the case when $m \geq \log_2 n + 2$, we may assume that $2^m \geq 4n$, and that $n \geq 4$. The likelihood of obtaining an error of δ is

$$\begin{aligned} P(\delta) &= \prod_{p=1}^{n} P_p(\delta_p) \\ &= \prod_{p=1}^{n-m} \cos^2(2^{-m}(0.x_{p+m} \ldots x_n) + 2^{p-1}\delta) \prod_{p=n-m+1}^{n} \cos^2(2^{p-1}\delta), \end{aligned}$$

where each P_p is the probability of the p-th individual trial returning the correct answer given an error of δ in the phase rotation. From this, we have

$$\begin{aligned} P(0) &= \prod_{p=1}^{n-m} \cos^2(2^{-m}(0.x_{p+m} \ldots x_n)) \\ &\geq (\cos^2(\pi 2^{-m}))^{n-m} \end{aligned}$$

$$\begin{aligned} &\geq \left(\cos^2\left(\frac{\pi}{4n}\right)\right)^{n-m} \\ &\geq 1 - \frac{\pi^2}{16n} \qquad \text{(as in Section 4.3)} \\ &\geq 1 - \frac{\pi^2}{64} > \frac{3}{4}, \end{aligned}$$

since $n \geq 4$.

Now, we will show that $P(\delta) \leq P(0)$ whenever $|\delta| \geq \frac{1}{2^n}$. In general, we have

$$P(\delta) = P_n(\delta)P_{n-1}(\delta)P_{n-2}(\delta)\cdots P_1(\delta).$$

Specifically, we have

$$P_p(\delta) = \cos^2(\pi 2^{-m}(0.x_{m+p}\ldots x_n) + \pi 2^{p-1}\delta),$$

when $1 \leq p \leq n-m$, and

$$P_p(\delta) = \cos^2(\pi 2^{p-1}\delta),$$

when $p > n-m$.

Consider the interval $\delta \in [\frac{1}{2^p} - \frac{1}{2^{p+1}}, \frac{1}{2^p} + \frac{1}{2^{p+1}}]$ for $1 \leq p \leq n$. When $1 \leq p \leq n-m$, we have

$$\begin{aligned} P(\delta) &\leq P_p(\delta) \\ &= \cos^2(2^{p-1}\pi\delta + \pi 2^{-m}(0.x_{m+p}\ldots x_n)). \end{aligned}$$

In particular, since in the given interval, we have

$$\left|\delta - \frac{1}{2^p}\right| \leq \frac{1}{2^{p+1}},$$

we also have

$$\left|\delta + \frac{2^{-m}(0.x_{m+p}\ldots x_n)}{2^{p-1}} - \frac{1}{2^p}\right| \leq \frac{1}{2^{p+1}} + \frac{2^{-m}}{2^{p-1}},$$

and so

$$\begin{aligned}
P(\delta) &\leq \cos^2(2^{p-1}\pi\delta + \pi 2^{-m}(0.x_{m+p}\ldots x_n)) \\
&= \sin^2\left(2^{p-1}\pi\delta + \pi 2^{-m}(0.x_{m+p}\ldots x_n) + \frac{\pi}{2}\right) \\
&= \sin^2\left(2^{p-1}\pi\left|\delta + \frac{2^{-m}(0.x_{m+p}\ldots x_n))}{2^{p-1}} - \frac{1}{2^p}\right|\right) \\
&\leq \sin^2\left(2^{p-1}\pi\left(\frac{1}{2^{p+1}} + \frac{2^{-m}}{2^{p-1}}\right)\right) \\
&= \sin^2\left(\frac{\pi}{4} + \pi 2^{-m}\right) \\
&\leq \sin^2\left(\frac{\pi}{4} + \pi 2^{-4}\right) \\
&\leq \sin^2\left(\frac{\pi}{3}\right) \\
&= \frac{3}{4}.
\end{aligned}$$

When $n - m < p \leq n$, we have

$$\begin{aligned}
P(\delta) &\leq \cos^2(2^{p-1}\pi\delta) \\
&= \sin^2\left(2^{p-1}\pi\left|\delta - \frac{1}{2^p}\right|\right) \\
&= \sin^2\left(\frac{\pi}{4}\right) \\
&= \frac{1}{2} \\
&\leq \frac{3}{4}.
\end{aligned}$$

We can derive a similar result for the intervals $\delta \in [-\frac{1}{2^p} - \frac{1}{2^{p+1}}, -\frac{1}{2^p} + \frac{1}{2^{p+1}}]$.

As p ranges from 1 to n, the intervals $\delta \in [\pm\frac{1}{2^p} - \frac{1}{2^{p+1}}, \pm\frac{1}{2^p} + \frac{1}{2^{p+1}}]$ cover all of $[-\frac{1}{2}, -\frac{1}{2^{n+1}}] \cup [\frac{1}{2^{n+1}}, \frac{1}{2}]$, giving us the desired result.

We have now established that when $m \geq \log_2 n+2$ (and as a consequence $n \geq 4$), we have $P(\delta) \leq \frac{3}{4} < P(0)$ for all $\delta \notin (-\frac{1}{2^n}, \frac{1}{2^n})$. This means that the value $\hat{\delta}$ which maximizes $P(\delta)$ must be within the range $\hat{\delta} \in (-\frac{1}{2^n}, \frac{1}{2^n})$. In a sense, this means that

the output value returned by the $AQFT_m$ circuit is "close enough" to the actual MLE when $m \geq \log_2 n + 2$. In fact, the true MLE can be computed efficiently using numerical methods, given that it is within such a narrow range. However, in practice, since the likelihood interval corresponding to the MLE occupies a good part of the interval $(-\frac{1}{2^n}, \frac{1}{2^n})$, the exact value of the MLE is unimportant.

Chapter 5

Generalized Phase Estimation Algorithms

In the last chapter, we introduced the AQFT-based phase estimation algorithm and established that the AQFT can still be effective without precise phase rotation gates which the full QFT-based phase estimation requires. This indicates that it should be possible to construct generic phase estimation algorithms with even more flexibility in using phase rotation gates. This chapter explores the class of phase estimation algorithms that stems from this generalization of the AQFT.

5.1 A General Model for AQFT Circuits

The $AQFT_m$ circuit from Chapter 4 can be characterized in a way that suggests a natural generalization. We consider the $AQFT_m$ circuit on an n-qubit register as a sequence of n trials, $T_n, T_{n-1}, \ldots, T_1$, where trial T_p constructs the state $|0\rangle + e^{2\pi i(2^{p-1}\phi)}|1\rangle$ and applies a phase rotation of $e^{-2\pi i\chi_p}$, where χ_p is a real number in

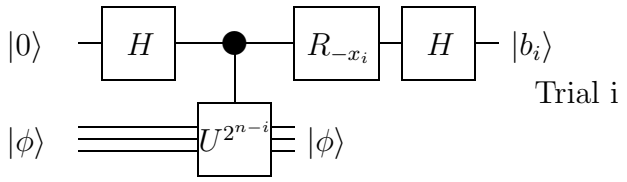

Figure 5.1: A generalized individual trial T_p

the range $[0, 1)$. In practice, χ_p will be a function of the results of the previous trials $T_n, \ldots, T_{p+1}$ and will attempt to adjust the prepared state so that it is as close to $|0\rangle + e^{2\pi i(0.x_p)} |1\rangle$ as possible.

Specifically, in the regular QFT circuit with individual trials, for trial T_n, we have $\chi_n = 0$, and for trial T_p where $p < n$, we have $\chi_p = 0.0x_{p+1}x_{p+2}\ldots x_n$. With $AQFT_m$ where $0 < m < n$, we have $\chi_n = 0$ for trial T_n, $\chi_p = 0.0x_{p+1}\ldots x_n$ for trials T_p where $n - m < p < n$, and $\chi_p = 0.0x_{p+1}\ldots x_{p+m-1}$ for trials T_p where $p \leq n - m$. For $AQFT_1$, we have $\chi_p = 0$ for each trial T_p.

In general, we may consider arbitrary values for χ_p, and we may adapt the post-processing algorithm used with the $AQFT_1$ circuit in order to obtain the maximum likelihood estimate, $\hat{\phi}$, as the output of the generalized phase estimation algorithm. However, in Section 4.4, we showed that for $m \geq \log_2 n + 2$ the output of the $AQFT_m$ circuit is essentially equivalent to the MLE. This chapter focuses primarily on generalized AQFT circuits for which this property still holds.

In the $AQFT_m$ circuit, each of the phase rotation amounts required by the full QFT, $\chi_p = 0.0x_{p+1}x_{p+2}\ldots x_n$, is approximated to within a certain tolerance, τ_p. In the case of the $AQFT_m$, we have $\tau_p = 0$ when $n - m < p \leq n$ and $\tau_p = \frac{1}{2^m} - \frac{1}{2^{n-p+1}}$ when $p \leq n-m$. However, we may also consider the general question of implementing a quantum gate affecting a certain phase rotation within a tolerance of τ_p. Recall that the Solovay-Kitaev theorem, previously mentioned in Chapter

2, gives a construction that requires $O(\log^{2+o(1)}(1/\tau_p))$ gates drawn from a fixed universal set of quantum gates.

In the analysis of the AQFT in Section 4.3, we did not consider the effects of errors originating from the individual rotation gates themselves. These errors can be naturally incorporated into our generalized model as the constant tolerances τ_p. However, it is not acceptable in this model to have $\tau_p = 0$ as we do in the $AQFT_m$ circuit when $n - m < p \leq n$, since implementing a quantum rotation gate with an error tolerance of 0 is not possible. To address this problem, we can relax each tolerance in the $AQFT_m$ circuit to $\tau_p = \frac{1}{2^m}$. This relaxation will give us a lower bound for the probability of success for any generalized AQFT algorithm such that $\max_p \tau_p = \frac{1}{2^m}$. In the next section, we show that this relaxation affects only the probability of obtaining the correct answer as output by a constant factor relative to the $AQFT_m$, which can easily be compensated for. The proof of this fact is very similar to the proof of the corresponding result for the $AQFT_m$ circuit given in Section 4.4.

5.2 Analysis of the Generalized AQFT

We proceed in a similar manner to the analysis of the AQFT in Chapter 4. As in Section 4.3, we take $|0\rangle + e^{2\pi i((0.x_p)+\delta_p)}|1\rangle$ to be the qubit state constructed by the quantum circuit for individual trial p. If δ is the offset between the correct phase, ϕ, and the measured phase, $\hat{\phi}$, then we have $\delta_p = \chi_p + 2^{p-1}\delta$. The likelihood of obtaining an error of δ is

$$P(\delta) \;=\; \prod_{p=1}^{n} \cos^2(\pi\delta_p)$$

$$= \prod_{p=1}^{n} \cos^2(2^{p-1}\pi\delta + \pi\chi_p).$$

We will start, as in Section 4.4, by giving a lower bound for $P(0)$. We have

$$\begin{aligned} P(0) &= \prod_{p=1}^{n} \cos^2(\pi\chi_p) \\ &\geq \left(\cos^2(\pi 2^{-m})\right)^n \\ &\geq 1 - \frac{\pi^2}{16n} \\ &\geq \frac{3}{4}. \end{aligned}$$

Now, we show that when $\delta \notin [-\frac{1}{2^{n+1}}, \frac{1}{2^{n+1}}]$, we have $P(\delta) \leq \frac{3}{4}$, which means that the answer output by the generalized AQFT circuit is the integer multiple of $\frac{1}{2^n}$ nearest to the maximum likelihood estimate of the phase amount $\hat{\phi}$.

Consider the interval $\delta \in [\frac{1}{2^p} - \frac{1}{2^{p+1}}, \frac{1}{2^p} + \frac{1}{2^{p+1}}]$ for $1 \leq p \leq n$. We have

$$\begin{aligned} P(\delta) &\leq P_p(\delta) \\ &= \cos^2(2^{p-1}\pi\delta + \pi\chi_p). \end{aligned}$$

In particular, since

$$\left|\delta - \frac{1}{2^p}\right| \leq \frac{1}{2^{p+1}},$$

we have

$$\left|\delta + \frac{\chi_p}{2^{p-1}} - \frac{1}{2^p}\right| \leq \frac{1}{2^{p+1}} + \frac{2^{-m}}{2^{p-1}},$$

and so

$$\begin{aligned} P(\delta) &\leq \cos^2(2^{p-1}\pi\delta + \pi\chi_p) \\ &= \sin^2\left(2^{p-1}\pi\delta + \pi\chi_p + \frac{\pi}{2}\right) \\ &= \sin^2\left(2^{p-1}\pi\left|\delta + \frac{\chi_p}{2^{p-1}} - \frac{1}{2^p}\right|\right) \end{aligned}$$

$$\begin{aligned}
&\leq \sin^2\left(2^{p-1}\pi\left(\frac{1}{2^{p+1}}+\frac{2^{-m}}{2^{p-1}}\right)\right)\\
&= \sin^2\left(\frac{\pi}{4}+\pi 2^{-m}\right)\\
&\leq \sin^2\left(\frac{\pi}{4}+\pi 2^{-4}\right)\\
&\leq \sin^2\left(\frac{\pi}{3}\right)\\
&= \frac{3}{4}.
\end{aligned}$$

We can derive a similar result for the interval $\delta \in [-\frac{1}{2^p}-\frac{1}{2^{p+1}}, -\frac{1}{2^p}+\frac{1}{2^{p+1}}]$.

As p ranges from 1 to n, the intervals $\delta \in [\pm\frac{1}{2^p}-\frac{1}{2^{p+1}}, \pm\frac{1}{2^p}+\frac{1}{2^{p+1}}]$ cover all of $[-\frac{1}{2}, -\frac{1}{2^{n+1}}] \cup [\frac{1}{2^{n+1}}, \frac{1}{2}]$, giving us the desired result.

Now, we need a lower bound on the probability of the answer output by the generalized AQFT circuit being correct. We proceed in a similar manner to the analysis in Section 4.3. This means finding a lower bound for $P(\delta)$ when $\delta \in [-\frac{1}{2^{n+1}}, \frac{1}{2^{n+1}}]$.

We have

$$\begin{aligned}
P(\delta) &= \prod_{p=1}^{n} \cos^2(2^{p-1}\pi\delta + \pi\chi_p)\\
&\geq \prod_{p=1}^{n} \cos^2\left(2^{p-1}\pi\frac{1}{2^{n+1}} + \pi\frac{1}{2^m}\right)\\
&= \prod_{p=1}^{n} \cos^2(\pi 2^{-m} + \pi 2^{p-n-2})\\
&\geq \prod_{p=1}^{n-m+2} \cos^2(2\pi 2^{-m}) \prod_{p=n-m+3}^{n} \cos^2(\pi 2^{p-n-2} + \pi 2^{-m})\\
&= \left(\cos^2(\pi 2^{-m+1})\right)^{n-m+2} \prod_{q=1}^{m-2} \cos^2(\pi 2^{-q-1} + \pi 2^{-m})\\
&\geq \left(\cos^2\frac{\pi}{2n}\right)^{n} \prod_{q=1}^{m-2} \cos^2(\pi 2^{-q-1} + \pi 2^{-m})
\end{aligned}$$

$$\begin{aligned} &\geq \left(1 - n\left(\frac{\pi}{2n}\right)^2\right) \prod_{q=1}^{m-2} \cos^2(\pi 2^{-q-1} + \pi 2^{-m}) \\ &\geq \left(1 - \frac{\pi^2}{4n}\right) \prod_{q=1}^{m-2} \cos^2(\pi 2^{-q-1} + \pi 2^{-m}), \end{aligned}$$

by using the Bernoulli inequality as in Section 4.3.

We can analyze the product

$$\prod_{q=1}^{m-2} \cos^2(\pi 2^{-q-1} + \pi 2^{-m})$$

using a method similar to the proof of Lemma 3.6.1. We show, by induction, that for any integer k, $1 \leq k \leq m-2$, we have

$$\prod_{q=1}^{k} \cos^2(\pi 2^{-q-1} + \pi 2^{-m}) \geq \frac{\sin^2\left(\frac{\pi}{2} + \pi 2^{-m+1}\right)}{\left(2^k \sin\left(\frac{\pi/2}{2^k} + \pi 2^{-m}\right)\right)^2}.$$

For $k = 1$, we have

$$\cos^2\left(\frac{\pi}{4} + \pi 2^{-m}\right) = \frac{\sin^2\left(\frac{\pi}{2} + \pi 2^{-m+1}\right)}{\left(2\sin\left(\frac{\pi/2}{2} + \pi 2^{-m}\right)\right)^2},$$

which follows immediately from the double angle formula. Now, suppose that for some fixed k such that $k < m-2$, we have

$$\prod_{q=1}^{k} \cos^2(\pi 2^{-q-1} + \pi 2^{-m}) \geq \frac{\sin^2\left(\frac{\pi}{2} + \pi 2^{-m+1}\right)}{\left(2^k \sin\left(\frac{\pi/2}{2^k} + \pi 2^{-m}\right)\right)^2}.$$

Then,

$$\begin{aligned} \prod_{q=1}^{k+1} \cos^2(\pi 2^{-q-1} + \pi 2^{-m}) &= \cos^2(\pi 2^{-k-2} + \pi 2^{-m}) \prod_{q=1}^{k} \cos^2(\pi 2^{-q-1} + \pi 2^{-m}) \\ &\geq \cos^2(\pi 2^{-k-2} + \pi 2^{-m}) \frac{\sin^2\left(\frac{\pi}{2} + \pi 2^{-m+1}\right)}{\left(2^k \sin\left(\frac{\pi/2}{2^k} + \pi 2^{-m}\right)\right)^2} \end{aligned}$$

$$
\begin{aligned}
&= \left(\frac{\sin(\pi 2^{-k-1}+\pi 2^{-m+1})}{2\sin(\pi 2^{-k-2}+\pi 2^{-m})}\right)^2 \frac{\sin^2\left(\frac{\pi}{2}+\pi 2^{-m+1}\right)}{\left(2^k \sin\left(\frac{\pi/2}{2^k}+\pi 2^{-m}\right)\right)^2} \\
&\geq \left(\frac{\sin(\pi 2^{-k-1}+\pi 2^{-m})}{2\sin(\pi 2^{-k-2}+\pi 2^{-m})}\right)^2 \frac{\sin^2\left(\frac{\pi}{2}+\pi 2^{-m+1}\right)}{\left(2^k \sin\left(\frac{\pi/2}{2^k}+\pi 2^{-m}\right)\right)^2} \\
&= \frac{\sin^2\left(\frac{\pi}{2}+\pi 2^{-m+1}\right)}{\left(2^{k+1} \sin\left(\frac{\pi/2}{2^{k+1}}+\pi 2^{-m}\right)\right)^2},
\end{aligned}
$$

as desired. Our result follows by induction.

So, using this result, we have

$$
\begin{aligned}
P(\delta) &\geq \left(1-\frac{\pi^2}{4n}\right)\prod_{q=1}^{m-2}\cos^2(\pi 2^{-q-1}+\pi 2^{-m}) \\
&\geq \left(1-\frac{\pi^2}{4n}\right)\frac{\sin^2\left(\frac{\pi}{2}+\pi 2^{-m+1}\right)}{\left(2^{m-2}\sin\left(\frac{\pi/2}{2^{m-2}}+\pi 2^{-m}\right)\right)^2} \\
&= \left(1-\frac{\pi^2}{4n}\right)\frac{\cos^2(\pi 2^{-m+1})}{\left(2^{m-2}\sin\left(3\pi 2^{-m}\right)\right)^2} \\
&\geq \left(1-\frac{\pi^2}{4n}\right)\frac{1-(\pi 2^{-m+1})^2}{\left(2^{m-2}(3\pi 2^{-m})\right)^2} \\
&\geq \left(1-\frac{\pi^2}{4n}\right)\frac{1-(\frac{2\pi}{4n})^2}{(3\pi 2^{-2})^2} \\
&\geq \left(1-\frac{\pi^2}{4n}\right)\left(\frac{16}{9\pi^2}-\frac{4}{9n^2}\right),
\end{aligned}
$$

which approaches a constant asymptotic bound of $\frac{16}{9\pi^2}$ as n approaches infinity.

5.3 Repeated Individual Trials

As with any algorithm, the natural way to increase the accuracy of a phase estimation algorithm is to repeat the entire algorithm a number of times. However, since we can divide the generalized AQFT circuit into individual trials, it is possible to

repeat each *trial* a different number of times. For each p, $1 \leq p \leq n$, we let λ_p be the number of times we repeat the individual trial for bit x_p. This raises the question of how many times to perform each trial given some limitation on computation resources. If we let μ_p be the amount of resources required to perform a trial for the bit x_p (including the controlled-U^{2^p}) and the rotation, then we would like to optimize the parameters λ_p given an upper bound for $\sum_p \lambda_p \mu_p$. However, we are left with the question of how to quantify what it is we would like to optimize. Different applications of phase estimation have different needs that may be best served with different values for λ_p. As an example, an application that does not require very much precision in the answer would not benefit from more measurements of the last bit x_n as much as an application that does require that precision.

One way to quantify this notion is through a cost function. Define the function $C : [-\frac{1}{2}, \frac{1}{2}] \to \mathbb{R}$ so that $C(\delta)$ will indicate the cost of obtaining an error of $\delta = \phi - \hat{\phi}$ in the phase amount. Then, given an upper bound on $\sum_p \lambda_p \mu_p$, R, we need to find parameters that minimize the expected cost of the result of the phase estimation algorithm.

As for the set of valid cost functions, we may make the practical assumption that getting a more accurate answer is always better, so that $C(x)$ is a monotone increasing function on $[0, \frac{1}{2})$ and a monotone decreasing function on $(-\frac{1}{2}, 0]$. The optimal values for λ_p can be approximated efficiently as a precomputation using numerical techniques, given fixed values for the μ_p and a fixed upper bound for $\sum_p \lambda_p \mu_p$. Given any specific application, it is not unreasonable to have fixed values for these parameters. In addition, if we consider the parameters λ_p as a function of R, the total amount of resources available, then, given optimal parameters $\lambda_p(R)$ for a certain amount of resources R and $2R$ for twice as many resources, the optimal parameters should be fairly close to simply repeating the optimal configuration

twice. Simply put, we should expect $\lambda_p(2R) \approx 2\lambda_p(R)$ for sufficiently large R. Because of this, we conjecture that the ratios between the various parameters $\lambda_p(R)$ approach optimal ratios asymptotically, and that given a fixed R, the optimal solution is simply the one which best approximates these optimal ratios. These optimal ratios should depend only on the value of the cost function, C.

Chapter 6

Applications and Future Work

This chapter surveys a few of the applications of quantum phase estimation and some avenues for further research on generalized AQFT circuits, including applications other than phase estimation.

6.1 Order Finding

Order finding is an example of a quantum algorithm that solves a problem in a polynomial number of gates for which there is no known classical polynomial time algorithm. The problem is to find the order of a given x modulo N, which is the smallest positive integer r so that $x^r \equiv 1 \bmod N$. x is a fixed number which is chosen as an input, and is presumed to be coprime with N, since it is easily verifiable. This restriction ensures that the order r always exists.

This is a simple application of the phase estimation algorithm. In this case, the unitary operator U will operate on n bits, where $N \leq 2^n$. For any pure state $|k\rangle$, U will map it to the state $|kx \bmod N\rangle$, unless $k \geq N$, in which case U will map

it to back to $|k\rangle$. Since x is invertible modulo N, the map U is invertible as well. Since modular multiplication is easily implemented with classical circuits, it is also easy to implement using quantum gates.

The more difficult part is to prepare the eigenstate needed by the phase estimation algorithm. We can verify that the eigenstates of U are

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} \left|x^k \bmod N\right\rangle,$$

for $s = 0, 1, \ldots, r-1$, and the corresponding eigenvalues are

$$u_s = e^{2\pi i s/r}$$

by computing

$$\begin{aligned} U|u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} \left|x^{k+1} \bmod N\right\rangle \\ &= e^{2\pi i s/r} |u_s\rangle . \end{aligned}$$

However, since we have no foreknowledge about the value of r (the order of x), we cannot construct these eigenstates beforehand.

Fortunately, we can make the observation that

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle ,$$

as everything conveniently cancels out. This means that the state $|1\rangle$, which is easy to prepare, is an equal superposition of all of the eigenstates of U. Inputting this state into the phase estimation algorithm will output a superposition of states corresponding to the individual eigenvalues of U. This means that when we make a measurement, we will get an estimate of s/r for some random s. We may then use continued fraction expansions to find r on a classical computer, as long as the estimate of s/r is sufficiently accurate.

In the phase estimation algorithm, we need to perform the controlled-U^{2^p} gate for each p, $1 \leq p \leq n$. Since U^{2^p} is simply modular exponentiation, which can be implemented classically in polynomial time, they can be efficiently implemented by a quantum computer as well. This completes the order finding algorithm.

6.2 Shor's Polynomial-time Factoring Algorithm

Shor's quantum factoring algorithm is simply an application of the order-finding algorithm. In this algorithm we can assume that N, the integer to be factored, is not an exact power of another integer of the form a^b where $b \geq 2$, as there are efficient classical algorithms to find a and b in this case. We would then proceed to use the factorization algorithm to factor a. It is also assumed that N is odd and composite, as these are also easily checked classically.

We first choose a random value x between 2 and $N-1$ so that it is coprime with N. If we happen to select an x which is not coprime, it can be used to find a factor of N classically. We then use the order-finding algorithm to find the smallest positive integer r so that $x^r \equiv 1 \bmod N$. We would like to have r so that it is even, and so that $x^{r/2} \not\equiv -1 \bmod N$. It can be shown that this will happen with probability at least 3/4. If the order r is not even, or does not satisfy $x^{r/2} \not\equiv -1 \bmod N$, then we choose a different value of x at random, and try again.

Upon finding such an r, we will have x such that $\left(x^{r/2}\right)^2 \equiv 1 \bmod N$. Since r is the order of x, we know that $x^{r/2} \not\equiv 1 \bmod N$. From our choice of r, we also know that $x^{r/2} \not\equiv -1 \bmod N$. Thus, we have a non-trivial solution to $(x^{r/2}-1)(x^{r/2}+1) \equiv 0 \bmod N$, which means that one of these two factors must reveal a non-trivial factor of N.

6.3 The Abelian Hidden Subgroup Problem

The Abelian hidden subgroup problem (AHSP) is a generalization of many of the applications of phase estimation, such as order finding and discrete logarithms. In the AHSP, we are given a known finite Abelian group, G, and an unknown subgroup, $K \leq G$. Suppose also that there is a function, f, which can distinguish between cosets of K. Specifically, f is a function, $f : G \to \mathbb{Z}$ such that for $g_1, g_2 \in G$, $f(g_1) = f(g_2)$ whenever g_1 and g_2 belong to the same coset of K, and $f(g_1) \neq f(g_2)$ whenever they belong to different cosets. Suppose that we are given a black-box[1] quantum gate, U_f, which performs the operation $U_f : |g\rangle |j\rangle \mapsto |g\rangle |f(g) \oplus j\rangle$, where $\oplus$ represents the binary bit-wise "XOR" operation. Then the Abelian hidden subgroup problem asks for a set of elements from G that form a generator for the subgroup K.

To solve the AHSP, we require an extension of the quantum Fourier transform that acts on n-qubit registers as representations of Abelian groups rather than simply as a representation of the integers modulo 2^n, which is itself an Abelian group. Since finite Abelian groups are direct products of cyclic groups of prime power order, we simply represent each cyclic group with a quantum register. We then define the Fourier transform over the entire group as the combination of Fourier transforms over each individual cycle. For each cyclic group in the direct product, $\mathbb{Z}_N$, the QFT maps each basis state $|j\rangle$ to

$$QFT(|j\rangle) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle$$

for $0 \leq j \leq N-1$, and simply maps $|j\rangle$ to itself for $j \geq N$. Given a basis state corresponding to an element of G, $|j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_q\rangle$, where each term corresponds

[1] This gate is a "black box" in the sense that the algorithm which solves the AHSP does not rely on any knowledge of the implementation of such a gate.

to one of the cyclic groups that make up G, the QFT circuit will map this state to

$$QFT(|j_1\rangle) \otimes \cdots \otimes QFT(|j_q\rangle).$$

A description of the actual quantum circuit that performs the QFT over an arbitrary Abelian group can be found in [9].

The first step in solving the AHSP is to construct the state

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle \, |f(g)\rangle$$

by first creating a superposition of the basis states, $|g\rangle$, adjoining a quantum register of all $|0\rangle$ qubits and performing the operation U_f. If we perform a measurement on the second register, we will obtain a fixed state, $|f(g')\rangle$, for some specific $g' \in G$. However, this partial measurement in the second register causes only those states $|g\rangle$ which corresponded to $|f(g')\rangle$ to remain in the first register. This is because when the first register is measured, the outcome must have the property that the combination of the two measurements is a possible outcome of measuring both registers at the same time. This means that the first register contains only states such that $f(g) = f(g')$, which includes all the states that correspond to the coset of K containing g', or $K + g'$.

Our first register is now in the state

$$\frac{1}{\sqrt{|K|}} \sum_{g \in K} |g' + g\rangle$$

and we apply the QFT corresponding to the Abelian group G. If G were a cyclic group of prime power order, $\mathbb{Z}_{p^n}$, then the subgroup $K \leq G$ would be isomorphic to $\mathbb{Z}_{p^m}$ for some m such that $0 \leq m \leq n$. As a subgroup of G, K would be generated by the element $p^{n-m} \in G$, and so, letting $|j'\rangle$ be the state corresponding $g' \in G$,

we have

$$QFT\left(\frac{1}{\sqrt{|K|}}\sum_{g\in K}|g'+g\rangle\right) = \frac{1}{\sqrt{p^m}}\sum_{j=0}^{p^m-1}\frac{1}{\sqrt{p^n}}\sum_{k=0}^{p^n-1}e^{2\pi i(j'+p^{n-m}j)k/p^n}|k\rangle$$
$$= \frac{1}{\sqrt{p^n}}\sum_{k=0}^{p^n-1}e^{2\pi ij'k/p^n}\frac{1}{\sqrt{p^m}}\left(\sum_{j=0}^{p^m-1}e^{2\pi ijk/p^m}\right)|k\rangle\,.$$

If k is a multiple of p^m, then $e^{2\pi ijk/p^m}=1$. Otherwise, $e^{2\pi ijk/p^m}=0$. So, we have

$$QFT\left(\frac{1}{\sqrt{|K|}}\sum_{g\in K}|g'+g\rangle\right)=\frac{1}{\sqrt{p^{n-m}}}\sum_{k'=0}^{p^{n-m}-1}e^{2\pi ij'k'/p^{n-m}}|p^m k\rangle\,.$$

If we now perform a measurement on the first register, we are certain to obtain an element of the subgroup $K^\perp$, which is isomorphic to $\mathbb{Z}_{p^{n-m}}$. With $O(n)$ such elements, n independent elements can be found with high probability, so that generators for both $K^\perp$ and K can be determined classically.

For a direct product of such cyclic groups, the result will just be the combination of the individual results, and we again obtain an element of $K^\perp$ after applying the QFT and measuring. With $O(\log|G|)$ such elements, a generating set for K can be found.

6.4 Future Work

Generalized AQFT circuits over Finite Abelian Groups

Although the focus of this thesis is the quantum Fourier transform over the group $\mathbb{Z}_{2^n}$, the question remains as to whether the generalizations and extensions of the QFT developed in previous chapters apply to the quantum Fourier transform over a general finite Abelian group.

The techniques used to relax the requirements of the full QFT to obtain the AQFT and the generalized AQFT should be applicable to the QFT algorithm for arbitrary Abelian groups as well.

Classical Post-Processing Algorithms

In the original version of Kitaev's algorithm, as given in [9], the need for a classical post-processing algorithm is avoided by repeating each individual trial for any particular bit, x_p, until it is correct with high probability, before moving onto the next bit. This procedure repeats each trial $O(n)$ times in order to reach a fixed accuracy rate.

However, in an information-theoretic sense, the output of even one set of individual trials should contain enough information about the phase in order to make an estimate correct to the nearest multiple of $1/2^n$ with a fixed constant probability. The classical post-processing algorithm given in Section 4.1 attempts to extract phase information efficiently. However, the analysis given for this algorithm is a heuristic analysis, and whether the algorithm is indeed polynomial-time or not hinges on whether the value of the required threshold parameter T is indeed in $O(\log n)$. In addition, many small improvements might also be made to the algorithm in order to make it faster. For practical purposes, it may be sufficient that T grow only slightly faster than $O(\log n)$, making the algorithm not polynomial-time, although perhaps still of practical value. Additionally, the algorithm has a higher chance of success when it has more unambiguous information about the answer encoded in the likelihood function, so it may be that the algorithm becomes polynomial-time after a certain number of good measurements are performed.

Finally, this post-processing algorithm is potentially applicable to a number

of other quantum algorithms for which finding the MLE directly appears to be a required step in order for the algorithm to work at all.

The Dihedral Hidden Subgroup Problem

One such problem is the *Dihedral Hidden Subgroup Problem*, described by Ettinger and Høyer in [6]. Although there is an algorithm to perform quantum Fourier transforms over finite Abelian groups, there is no such algorithm for non-Abelian groups. Given a dihedral group D_n, which is the group of symmetries of a n-sided regular polygon, the Dihedral HSP seeks the generators of a subgroup $K \leq D_n$ given a function $f : D_n \to \mathbb{Z}$ that distinguishes cosets of K in the same way the corresponding function does for the Abelian HSP.

However, the quantum algorithm given by Ettinger and Høyer requires classical post-processing equivalent to finding an approximation of the MLE, which they leave as an open problem. If the heuristic algorithm used with the $AQFT_1$ circuit in Section 4.1 can be modified for this purpose efficiently, then when combined with the quantum algorithm, it will become an efficient algorithm to solve an instance of the non-Abelian HSP.

Optimal Configurations for Generalized AQFT Circuits

In the analysis of the generalized AQFT in Chapter 5, the individual tolerance parameters τ_p were set to the value of the largest tolerance in order to obtain a lower bound for the probability of success. However, these parameters can also be adjusted according to a fixed amount of computational resources. As an example, in the regular AQFT circuit, having the first $O(\log_2 n)$ trials computed with a

significantly smaller tolerance for error gives much better results. However, parameterizing the entire phase estimation algorithm in terms of individual tolerances τ_p and repetition amounts λ_p does not lend itself well to analysis.

Additionally, it was conjectured in Section 5.3 that there exists a set of optimal ratios λ_p for a generalized phase estimation algorithm that are approached as the amount of resources goes to infinity.

6.5 Conclusion

The analysis of the approximate QFT and its generalization indicates that, as a quantum algorithm, the quantum Fourier transform is quite tolerant of errors. These improvements to the QFT do not allow us to construct entirely new quantum algorithms, but only to improve the performance of current algorithms which use the QFT. However, such improvements are still useful as they allow us to extend the range of feasible inputs to current quantum algorithms.

As one would expect, the greatest improvement in performance is evident at the first step of simplification, from the QFT to the $AQFT_m$ circuit. However, as we saw in Chapter 5, it is possible to be much more flexible with the rotation gates than the description of the AQFT circuit allows for. This is especially significant considering that any actual implementation of a generalized AQFT circuit will need to take into account the Solovay-Kitaev construction for approximating the rotation gates. Taking the bounds given by Solovay-Kitaev directly into account in our generalized AQFT model makes the analysis more relevant to any future attempts at implementing quantum algorithms. The improvements in performance presented in this thesis are of particular significance to any potential physical implementation of QFT-based algorithms, as the generalized phase estimation algorithms tend to

use less quantum gates, which reduces the overall potential for errors. The analysis of these algorithms also suggests that these quantum circuits are robust with respect to tolerance to errors.

Bibliography

[1] Aharonov, D., *Quantum Computation*, in *Annual Reviews of Computational Physics* VI, Stauffer, D., ed., World Scientific (1998).
http://www.arxiv.org e-print quant-ph/9812037

[2] Barenco, A., Ekert, A., Suominen, K.-A., Törmä, P., *Approximate Quantum Fourier Transform and Decoherence*, Physics Review A, **54**(1), pp. 139–146 (1996).
http://www.arxiv.org e-print quant-ph/9601018.

[3] Bhimji, W., *Approximate Quantum Fourier Transforms and Phase Estimations*. M.Phys. Project Paper, University of Oxford (1998).

[4] Cleve, R., Watrous, J., *Fast Parallel Circuits for the Quantum Fourier Transform*, IEEE Symposium on Foundations of Computer Science, pp. 526–536 (2000).
http://www.arxiv.org e-print quant-ph/0006004.

[5] Coppersmith, D., *An Approximate Fourier Transform Useful in Quantum Factoring*, IBM Research Report RC19642 (1994).
http://www.arxiv.org e-print quant-ph/0201067.

[6] Ettinger, M., Høyer, P., *On Quantum Algorithms for Noncommutative Hidden Subgroups*, Lecture Notes in Computer Science, **1563**, pp. 478–487 (1999). http://www.arxiv.org e-print quant-ph/9807029.

[7] Griffiths, R. B., Niu, C.-S., *Semiclassical Fourier Transform for Quantum Computation*, Physical Review Letters **76**, pp. 3228–3231 (1996). http://www.arxiv.org e-print quant-ph/9511007.

[8] Jozsa, R., *Quantum Algorithms and the Fourier Transform*, Proc. Roy. Soc. Lond. A, **454**, pp. 323–337 (1998). http://www.arxiv.org e-print quant-ph/9707033.

[9] Kitaev, A. Yu., *Quantum Measurements and the Abelian Stabilizer Problem*, Electronic Colloquium on Computational Complexity **3**(3) (1996). http://www.arxiv.org e-print quant-ph/9511026.

[10] Nielsen, M., and Chuang, I., *Quantum Computation and Quantum Information*, Cambridge University Press (2000).

[11] Press, W., et. al., *Numerical Recipes, 2nd ed.*, Cambridge University Press (1992).