The Quantum Fourier Transform and Extensions of the Abelian Hidden Subgroup Problem

by

Lisa Ruth Hales

B.A. (University of California at Berkeley) 1991

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Logic and the Methodology of Science

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Spring 2002

Abstract

The Quantum Fourier Transform and Extensions of the Abelian Hidden Subgroup Problem

by

Lisa Ruth Hales

Doctor of Philosophy in Logic and the Methodology of Science

University of California at Berkeley

Professor Umesh V. Vazirani, Chair

The quantum Fourier transform (QFT) has emerged as the primary tool in quantum algorithms which achieve exponential advantage over classical computation and lies at the heart of the solution to the abelian hidden subgroup problem, of which Shor's celebrated factoring and discrete log algorithms are a special case. We begin by addressing various computational issues surrounding the QFT and give improved parallel circuits for both the QFT over a power of 2 and the QFT over an arbitrary cyclic group. These circuits are based on new insight into the relationship between the discrete Fourier transform over different cyclic groups. We then exploit this insight to extend the class of hidden subgroup problems with efficient quantum solutions. First we relax the condition that the underlying hidden subgroup function be distinct on distinct cosets of the subgroup in question and show that this relaxation can be solved whenever G is a finitely-generated abelian group. We then extend this reasoning to the hidden cyclic subgroup problem over the reals, showing how to efficiently generate the bits of the period of any sufficiently piecewise-continuous function on $\Re$. Finally, we show that this problem of period-finding over $\Re$, viewed as an oracle promise problem, is strictly harder than its integral counterpart. In particular, period-finding over $\Re$ lies outside the complexity class MA, a class which contains period-finding over the integers.

Professor Umesh V. Vazirani
Dissertation Committee Chair

To Samantha,

a faithful friend.

Contents

List of Figures

Acknowledgements

I want to thank my advisor for his patience, his insight and his sense of humor without which this thesis would never have been completed.

I am grateful to Sean Hallgren, the co-author of many of the results in this thesis. I really enjoyed the countless hours spent together in cafes over the years and hope to collaborate again.

I want to thank the many members of the Logic group who have given me support and encouragement during my years at Berkeley. Professor John Addison deserves particular thanks for introducing me to the complexities of Complexity theory and for his ever-present sense of humor. I would certainly never have finished without the support and friendship of the group secretary, Catalina Cordoba. Good luck in your retirement! And thanks to Richard Zach for making the "middle years" of my graduate career more fun.

I also want to thank the faculty, staff, and students of my adopted department, Computer Science, for making me feel welcome.

Finally, I must thank my family. First, my Mom and Dad for their constant love and support without which none of this would have been possible. Second, my husband for putting up with my seemingly infinite thesis-writing and for his almost hourly help with my computer. My sister Kathy for doing the dishes, taking out the dog, and mowing the lawn even though she had her own thesis to write. Charlotte for her smiling face and hugs. Eve for her kicks and prods over the past few months which have served as a continuous reminder of the urgency of my task. And finally Sal for helping to heal a broken heart.

Chapter 1

Introduction

The primary tool underlying all quantum algorithms which achieve exponential advantage over classical computation is the quantum Fourier transform (QFT). The fact that the QFT over exponentially large groups can be computed efficiently is at the heart of the solution to the Abelian hidden subgroup problem, of which Shor's celebrated factoring and discrete logarithm algorithms [?] are a special case. The aim of this dissertation is twofold. First, we give improved quantum circuits for computing the QFT. Second, we use the resulting insight into the structure of the QFT to extend the class of hidden subgroup problems with efficient quantum solutions.

In particular, after surveying existing techniques computing the QFT over finite Abelian groups, we give explicit parallel circuits for approximating the QFT over a power of 2, tightening the results of [?]. We then give improved parallel circuits for approximating the QFT over an arbitrary cyclic group, based on new insight into the relationship between the discrete Fourier transforms over different cyclic groups. This insight also leads to a particularly elegant method of "Fourier sampling" ([?],[?],[?]) and simplifies the presentation of the standard Abelian hidden subgroup algorithm.

Second, we extend the class of Abelian hidden subgroup promise problems which have efficient quantum algorithms. Given oracle access to a function f defined on a group G and constant on cosets of some unknown subgroup $H \leq G$, a solution to the hidden subgroup problem is a list of generators for the subgroup H. This problem can be solved efficiently on a quantum computer whenever G is a finitely-generated Abelian group and f is distinct on distinct cosets([?]).

We first use our Fourier sampling procedure to relax this distinctness requirement, requiring only that the encoding of H by f be probabilistically unambiguous. This extends the results of [?] and [?] who relax the distinctness condition only slightly. Moreover, our result is tight – we give a corresponding lower bound which shows that, in the absence of such an unambiguous encoding, no polynomial-time algorithm, classical or quantum, can recover the desired hidden subgroup.

Finally, we give an efficient quantum algorithm for the hidden cyclic subgroup problem over the reals $\Re$. More specifically, given a sufficiently piecewise-continuous periodic function defined on $\Re$, we show how to efficiently generate the bits of its period. Again we must require that the encoding of the period be probabilistically unambiguous. This generalizes a result of [?] which gives a quantum algorithm finding the period of a subclass of such functions and an important application, namely an efficient quantum solution to Pell's equation. Furthermore, we show that the hidden cyclic subgroup problem over $\Re$ is harder than the analogous problem over Z. In particular we show that a decision version of the problem over $\Re$ is outside of the complexity class MA, whereas any decision problem which reduces to the problem over Z lies inside of this class.

1.1 Outline

The remainder of this Chapter is devoted to setting up our quantum circuit model. Chapter 2 introduces the QFT and its relation to the hidden subgroup problem while Chapter 3 surveys earlier techniques for implementing the QFT. In Chapter 4 we give new parallel circuits for computing the QFT over a power of 2. Chapter 5 contains both our algorithm for computing the QFT over an arbitrary cyclic group and also the associated Fourier sampling procedures. The technical results leading to these algorithms can be found in Chapter 9.

We then turn to extensions of the hidden subgroup algorithm. Chapter 6 extends the hidden subgroup algorithm over finitely generated Abelian groups to the case where the given function is not distinct on distinct cosets. The associated lower bound is found in Section 5. Chapter 7 is devoted to period-finding over the reals and and Chapter 8 to the proof that this problem is outside of MA.

1.2 Notation

We shall be primarily concerned with the vector space V_n over the field of complex numbers C consisting of formal linear combinations of bit-strings $k \in \{0,1\}^n$. We use the Dirac "ket" notation, $|\cdot\rangle$, for vectors in this space and reserve $|i\rangle$, $|j\rangle$ and $|k\rangle$ for the basis vectors corresponding to the bit-strings $i, j, k \in \{0,1\}^n$. $|v\rangle$, $|w\rangle$, and $|u\rangle$ denote arbitrary vectors in this space with

$$|v\rangle = \sum_{j \in \{0,1\}^n} v_j |j\rangle.$$

V_n is a Hilbert Space with **inner product**

$$\big\langle |v\rangle, |w\rangle \big\rangle = \sum_{i \in \{0,1\}^n} v_i \overline{w_i},$$

where $\overline{w_i}$ is the complex conjugate of w_i. By using the standard "bra" notation, $\langle \cdot |$, to denote the dual vector, i.e. $\langle v|$ is the linear operator from V_n to C defined by

$$\langle v||w\rangle = \big\langle |v\rangle, |w\rangle \big\rangle,$$

the vertical bars of the adjacent "bra" and "ket" in the inner product of $|v\rangle$ and $|w\rangle$ can be conveniently merged and written as $\langle v|w\rangle$. We let $\| \cdot \|$ denote the **norm** associated with this inner product,

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

Note that the vectors $|k\rangle$ form an orthonormal basis for V_n under this inner product. We will also use $\| \cdot \|$ to denote the associated **operator norm**. In particular if U is a (linear) operator on V_n then

$$\|U\| = max_{|v\rangle \in V} \frac{\|U|v\rangle\|}{\||v\rangle\|}.$$

For any linear operator U on V_n there is a unique linear operator $U^\dagger$ on V_n satisfying

$$\langle Uv|w\rangle = \langle v|U^\dagger w\rangle$$

for all vectors $|v\rangle$ and $|w\rangle$. $U^\dagger$ is called the **Hermitian adjoint** of U. Its matrix representation is the conjugate transpose of the matrix representing U, in other words if U has matrix representation (u_{ij}) then the matrix representation of $U^\dagger$ has ijth entry $\overline{u_{ji}}$. An operator U is called **unitary** if its Hermitian adjoint is also its inverse, that is, if $UU^\dagger = I$. This is equivalent to the condition that the vectors $U|k\rangle$ form an orthonormal basis for V.

An important construction underlying the quantum mechanics of multiparticle systems is the **tensor product** of vector spaces. Given any pair of bases for the vector spaces V and W, their Cartesian product forms a basis for the tensor product of V and W, denoted $V \otimes W$. That is, if $|j\rangle$ and $|k\rangle$ are elements of V and W's respective bases then $|j\rangle \otimes |k\rangle$ is an element of the resulting basis for $V \otimes W$. $V \otimes W$ then consists of all linear combinations of these basis elements modulo the following equivalences:

1. For any scalar $c \in C$ and elements $|v\rangle \in V$ and $|w\rangle \in W$,
$$c\left(|v\rangle \otimes |w\rangle\right) = \left(c|v\rangle\right) \otimes |w\rangle = |v\rangle \otimes \left(c|w\rangle\right).$$
2. For any $|v\rangle$ and $|v'\rangle$ in V and $|w\rangle$ and $|w'\rangle$ in W the following hold
$$\big(|v\rangle + |v'\rangle\big) \otimes |w\rangle = |v\rangle \otimes |w\rangle + |v'\rangle \otimes |w\rangle,$$
and
$$|v\rangle \otimes \big(|w\rangle + |w'\rangle\big) = |v\rangle \otimes |w\rangle + |v\rangle \otimes |w'\rangle.$$

These relations can also be used to give a basis independent construction of $V \otimes W$ – it is the free product of V and W modulo these equivalences. It is not hard to see that these equivalence relations do not collapse any elements of the basis for $V \otimes W$ described above and thus $\dim(V \otimes W) = \dim(V)\dim(W)$. It is worth noting that, while for any pair of vectors $|v\rangle \in V$ and $|w\rangle \in W$ the vector $|v\rangle \otimes |w\rangle$ is in $V \otimes W$, vectors of this form comprise only a tiny fraction of the tensor space. In particular, their description only has dimension $\dim(V) + \dim(W)$.

The tensor product $V \otimes W$ inherits a natural inner product structure from V and W by taking as an orthonormal basis any Cartesian product of orthonormal bases of V and W. Also, for any two linear operators A and B on V and W respectively, we can define their tensor product $A \otimes B$ which is the linear operator on $V \otimes W$ satisfying

$$A \otimes B\,(|j\rangle \otimes |k\rangle) = A|j\rangle \otimes B|k\rangle,$$

for basis elements $|j\rangle \in V$ and $|k\rangle \in W$. More generally, any bilinear map on the cartesian product $V \times W$ induces a linear transformation of the tensor product $V \otimes W$ – a category theoretic definition of the tensor product can be formulated in these terms.

Example: As a concrete example of the tensor product, recall the vector spaces V_n defined previously. The tensor product of any two spaces V_n and V_m has an orthonormal basis consisting of elements of the form $|j\rangle \otimes |k\rangle$ where j and k are bitstrings of length n and m respectively. The resulting space $V_n \otimes V_m$ is clearly isomorphic to V_{n+m} by extending the obvious map of basis elements

$$|j\rangle \otimes |k\rangle \longrightarrow |jk\rangle$$

Notice that this map is also preserves the corresponding inner product. ■

1.3 Qubits

A **qubit** is the abstraction of a two-level quantum particle, in the same sense that a bit is the abstraction of a classical storage device which can be in one of two positions, 0 or 1. While such a classical storage device is always either in position 0 or in position 1, quantum particles can exist in a complex combination or "superposition" of levels and are described by a unit vector in V_1, that is, a vector

$$\alpha_0|0\rangle + \alpha_1|1\rangle \tag{1.1}$$

where the α_i are complex numbers satisfying $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

A **measurement** is the abstraction of a physical procedure which obtains classical information about the state of the quantum particle. A measurement is represented mathematically as the projection of the state vector onto a pair of orthogonal subspaces. For instance, a measurement of the state (1.1) in the standard basis projects the state vector onto the subspaces generated by $|0\rangle$ and $|1\rangle$ respectively, yielding the state $|0\rangle$ with probability $|\alpha_0|^2$ and $|1\rangle$ with probability $|\alpha_1|^2$.

Quantum computation entails the manipulation multi-particle quantum systems. A system of n qubits is described by a unit vector in V_n, the tensor product of the individual vector spaces V_1 inhabited by each qubit. The state vector itself, however, need not be a product of vectors in these component spaces – recall that such product vectors form but a tiny fraction of the entire tensored space. A state vector which cannot be decomposed as such a product is **entangled**. Entangled states play a critical role both in quantum computation and quantum information theory.

1.4 Circuits: Classical vs. Quantum

Various models for quantum computation have been proposed. For our purposes it will be most convenient to work in the quantum circuit model. Before we specify the particulars of our model we review some of the features peculiar to quantum computation by contrasting a particular classical probabilistic circuit model with its quantum counterpart.

A classical probabilistic circuit takes as input a string of bits, runs them through a sequence of one and two-bit probabilistic gates, and outputs a string according to the probability distribution induced by the array of gates. For our purposes it will be convenient to assume these gates have equal length input and output. In particular, we take the deterministic NOT and $FAND$ (fan-out and) together with a probabilistic $NOT_{1/2}$ gate as our basis. The $NOT_{1/2}$ gate acts like the deterministic NOT gate with probability 1/2 and with probability 1/2 allow the bits to pass through unaffected. We can represent these gates by their transition matrices,

$$NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad FAND = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad NOT_{1/2} = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix},$$

where the rows and columns are indexed by bit strings in lexicographic order and the matrix entries represent the transition probabilities induced by the gate. The state of the circuit at any stage s can be described by a probability distribution on n-bit strings, or equivalently as a vector in V_n

$$\sum_{i \in \{0,1\}^n} p_{i,s} |i\rangle,$$

where $p_{i,s}$ denotes the probability that the bits are in the configuration $|i\rangle$ at stage s. Thus the $p_{i,s}$ are nonnegative reals satisfying

$$\sum_{i \in \{0,1\}^n} p_{i,s} = 1.$$

The probabilities $p_{i,s}$ can be gotten from the $p_{i,s-1}$ by the formula

$$p_{i,s} = \sum_{j\in\{0,1\}^n} p_{j,s-1} t_{j,i,s-1},$$

where $t_{j,i,s-1}$ is the probability that the string $|j\rangle$ transitions to the string $|i\rangle$ when the gate of stage $s-1$ is applied. Again, the $t_{j,i,s}$'s are nonnegative reals satisfying

$$\sum_{i\in\{0,1\}^n} t_{j,i,s} = 1$$

for all j, s. The matrix of values $(t_{j,i,s})$ for a fixed s is a tensor product of the identity matrix and the transition matrix of the gate (see above) applied at stage s. A classical probabilistic circuit with final state

$$\sum_{i\in\{0,1\}^n} p_{i,s_f} |i\rangle$$

outputs the string i with probability p_{i,s_f} at the conclusion of the algorithm.

In analogy to the classical case, a quantum circuit takes as input a string of qubits and runs them through a sequence of one and two-bit quantum gates. In this case the state of the machine at any stage s is unit vector in V_n,

$$|\alpha_s\rangle = \sum_{i\in\{0,1\}^n} \alpha_{i,s} |i\rangle$$

where the α_i are complex numbers satisfying

$$\sum_{i\in\{0,1\}^n} |\alpha_i|^2 = 1.$$

As in the classical case we assume that the input is a determinate bitstring, i.e. a quantum state of the form $|i\rangle$ for some $i \in \{0,1\}^n$. As before, the amplitude $\alpha_{i,s}$ of a state $|i\rangle$ at stage s can be gotten from the $\alpha_{i,s-1}$ by the formula

$$\alpha_{i,s} = \sum_{j\in\{0,1\}^n} \alpha_{j,s-1} \tau_{j,i,s-1},$$

where $\tau_{j,i,s-1}$ is the amplitude with which the state $|j\rangle$ transitions to the state $|i\rangle$ when the gate of stage $s-1$ is applied. Again the the matrix of values $(\tau_{j,i,s})$ is a tensor product of the identity matrix with the transition matrix of the gate applied at stage s, but in the quantum model the $\tau_{j,i,s}$'s are not positive real probabilities, but complex numbers whose amplitudes squared satisfy

$$\sum_{i \in \{0,1\}^n} |\tau_{j,i,s}|^2 = 1$$

for all j, s.

In order to obtain classical information from the final quantum state $|\alpha_{s_f}\rangle$ output by the array, a measurement in the standard basis is performed at the conclusion of the algorithm. The probability of measuring a particular string $|i\rangle$ is given by

$$\left|\alpha_{i,s_f}\right|^2.$$

We now isolate the aspects of the quantum model which distinguish it from its classical counterpart. First, we focus on the class of allowable gates. The principles of quantum mechanics require that the evolution of a quantum state be reversible – no information can be gained or lost. The classical $FAND$ gate, for example, violates this principle since it maps both the strings 00 and 01 to the string 00. This reversibility requirement restricts the class of allowable gates to those whose transition matrices are unitary and raises the question of whether a quantum device is capable of even simulating a classical probabilistic circuit, much less moving beyond it. Such a simulation is possible if we allow the circuit to maintain a copy of its input throughout the computation. That is, if there is a classical probabilistic circuit mapping

$$|i\rangle \longrightarrow |j\rangle \text{ with probability } p_{ij}$$

then there is a quantum circuit of polynomially-related size mapping

$$|i\rangle|0\rangle \longrightarrow \alpha_{ij}|i\rangle|j\rangle \text{ where } |\alpha_{ij}|^2 = p_{ij}.$$

The following three gates, known as the Hadamard, the controlled-not, and the 1/8-rotation respectively, together suffice for such a simulation.

$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \quad CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad R_3 = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/8} \end{pmatrix}.$$

Moreover, they are universal for quantum computation, that is, they can be used to approximate any unitary transformation on n qubits with arbitrary precision.

Theorem 1. *Any unitary transformation on n qubits can be approximated to within ϵ by a quantum circuit of size $O\left(n^2 4^n \log^c\left(n^2 4^n/\epsilon\right)\right)$ over the gates $\{H, CNOT, R_3\}$.*

Theorem 1 was proved independently by Solovay and Kitaev. See [**?**] for a nice proof and history.

While quantum circuits can efficiently simulate their classical probabilistic counterpart, the converse appears to be false. What are some of the difficulties inherent in such a simulation? One fundamental difference between the quantum and classical models is that in the quantum setting the transition function τ is complex-valued. This expresses the phenomenon of Quantum Interference – nontrivial computational paths can cancel each other out and disappear – a property which lies at the heart of the apparent exponential power of the quantum model over its classical counterpart. Another difference is that the norms of the amplitudes of the state vector and the transition function are only quadratically related to their associated probabilities. This property is exploited by the data-base search algorithm of Grover [?] and its extensions which achieve a corresponding quadratic speedup over probabilistic classical computation.

So far the only classical simulations of quantum circuits involve keeping an explicit record of the exponentially many amplitudes associated with each step of the computation. Such a brute-force simulation can be accomplished using polynomial-space (but exponential-time) on a classical Turing machine and, more specifically, inhabits the complexity class $P^{\#} \subseteq PSPACE$ [?]. This indicates that proving outright that quantum circuits cannot be efficiently simulated by classical computation is very difficult – such a proof would imply that $P \neq PSPACE$, a long-standing open question in complexity theory. On the other hand, Shor's quantum algorithms for factoring and discrete logarithms, well-studied problems for which there is no efficient classical solution, together with various oracle results, provide indirect evidence that no such simulation exists.

Finally, we note that classical devices have been proposed that purportedly solve NP-complete problems in polynomial-time [?]. In each case it was shown that the device in question required either exponential precision or energy and that its apparent power was hidden in one of these untenable physical assumptions. It is critical to point out that this is not the case in our quantum circuit model. In particular, we need not implement our basic gates exactly, nor even with exponential precision, to achieve the apparent power over classical circuits. It suffices to be able to approximate these gates to within an arbitrary inverse polynomial, that is, to implement a unitary transformation U satisfying

$$\|U - G\| < 1/p(n),$$

where G is the unitary transformation induced by the desired gate. This follows from fundamental work of [?] showing that the errors incurred by such gate approximations are additive. Thus for any given polynomial-size circuit each of these errors need only be less than a (larger) inverse polynomial in order for the distribution output by the approximation to be close to that of the exact circuit.

1.5 A Quantum Circuit Model

We now turn to the particular quantum circuit model used in this thesis. Our circuits will use the following slightly redundant set of gates, in keeping with [?]. The Hadamard,

$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}. \tag{1.2}$$

The single qubit rotation gates,

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}. \tag{1.3}$$

And, finally, the 2-qubit controlled rotation gates which perform the rotation R_k if and only if the control bit is a 1. These three types of gates are shown in Figure (1.1). For simplicity we assume that we are also able to run these gates in reverse. Multiple gates may be performed in parallel (i.e. to distinct sets of bits) at any given stage, allowing for

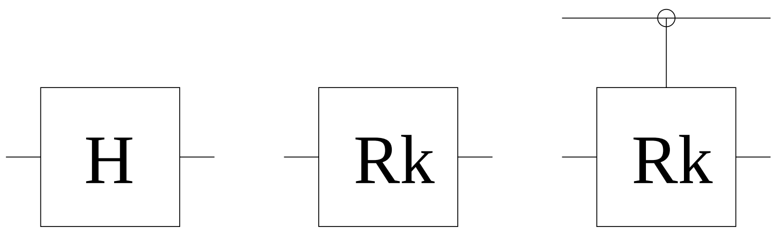


Figure 1.1: Quantum Gates: Hadamard, Rotation, Controlled Rotation.

both size (total number of gates) and depth (number of stages) analyses of our algorithms. In our discussion of the hidden subgroup Problem we will require quantum circuits which have oracle access to some function f. For our purposes we can assume without loss of generality that our input includes a special string of clean (all zero) qubits which are left untouched throughout the algorithm except for a single function call. At this point the oracle is invoked and the result is copied into the string of clean bits. A more general model would have to allow for multiple calls to the oracle and for manipulations of the resulting strings, but this restricted version suffices for our purposes.

1.5.1 Arithmetic Quantum Circuits

It will be useful for us in presenting our results to build a small repetoire of important subcircuits, in particular, quantum circuits for basic arithmetic operations. The following two lemmas allow us to translate classical results about arithmetic circuits to the quantum setting.

Lemma 1. *Suppose the map*

$$|x\rangle \longrightarrow |f(x)\rangle$$

is computable by classical deterministic circuits of size $s(n)$ and depth $d(n)$. Then the map

$$|x\rangle|0\rangle \longrightarrow |x\rangle|f(x)\rangle$$

is computable by quantum circuits of size and depth $O\left(s(n)\right)$ and $O\left(d(n)\right)$ respectively.

Lemma 2. *Suppose that f is 1-1 and the maps*

$$|x\rangle \longrightarrow |f(x)\rangle \quad \textit{and} \quad |x\rangle \longrightarrow |f^{-1}(x)\rangle$$

are computable by classical deterministic circuits of size $s_1(n)$ and $s_2(n)$ and depth $d_1(n)$ and $d_2(n)$ respectively. Then the map

$$|x\rangle \longrightarrow |f(x)\rangle$$

is computable by quantum circuits of size $O\left(s_1(n) + s_2(n)\right)$ and depth $O(d_1(n) + d_2(n))$.

These lemmas were originally proved in the context of classical reversible computation [**?**] – there is nothing inherently "quantum" about their proofs, which we proceed to sketch.

The first step in constructing a quantum (or classical reversible) circuit from a deterministic one is developing subcircuits which can simulate a universal set of classical boolean gates, such as NOT and AND. Simulating the classical NOT gate is easy since it is already reversible (Figure 1.2). Simulating the classical AND gate proves trickier. The three-qubit transformation pictured with its truth table in Figure 1.3 can be accomplished in constant size and depth by a surprisingly complicated configuration of our basic gates (See, for example, [**?**], page 182).

This gate is sometimes referred to as the controlled-controlled not, since it performs a controlled-not on the last two qubits if and only if the first is a 1, but more often it is

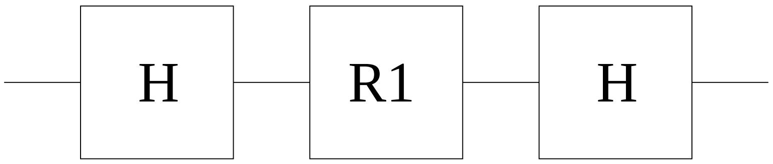


Figure 1.2: Quantum Not.

IN	OUT
111	110
110	111
101	101
100	100
011	011
010	010
001	001
000	000

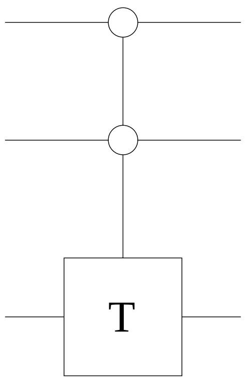


Figure 1.3: Toffoli Gate.

called the Toffolli gate in reference to [**?**]. It is easy to see from its truth table representation that, if the third qubit is set to $|0\rangle$, the AND of the first two input qubits is recorded in the third output.

With these two subcircuits now in hand, suppose we are given a classical circuit computing our function f. We replace each NOT gate by subcircuit (1.2) and each AND by subcircuit (1.3) supplemented with a clean qubit in its third register. This necessitates a supply of at most $s(n)$ clean qubits and yields a quantum circuit mapping

$$|x\rangle|0\rangle \longrightarrow |x\rangle|junk_x\rangle|f(x)\rangle,$$

where $junk_x$ are the junk-bits output by the first two registers of each Toffolli. The size and depth of this portion of the circuit are related to the classical circuit by constants deriving from the size and depth of the subcircuits (1.2) and (1.3).

$f(x)$ is then copied into a remaining set of clean bits,

$$|x\rangle|junk_x\rangle|f(x)\rangle|0\rangle \longrightarrow |x\rangle|junk_x\rangle|f(x)\rangle|f(x)\rangle.$$

A single bit can be copied into a clean qubit via the controlled-not subcircuit of Figure 1.4, and we let COPY denote the important subcircuit of size $3n$ and depth 3 consisting of n of

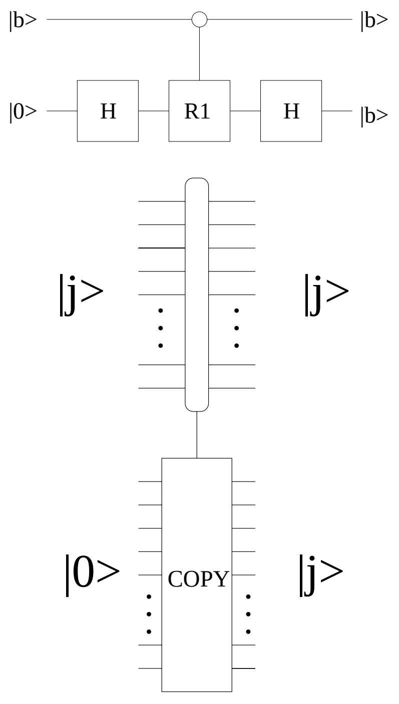


Figure 1.4: Controlled Not and Quantum Copy.

these controlled-nots in parallel.

Finally, the initial computation is run in reverse, yielding the desired map

$$|x\rangle|0\rangle \longrightarrow |x\rangle|0\rangle|f(x)\rangle.$$

The size and depth bounds follow easily.

We now turn to the second lemma. We have already shown how to construct circuits performing both

$$|x\rangle|0\rangle \longrightarrow |x\rangle|f(x)\rangle$$

and

$$|f(x)\rangle|0\rangle \longrightarrow |f(x)\rangle|x\rangle.$$

These circuits need merely be composed – the second in reverse – to obtain a circuit computing the desired

$$|x\rangle|0\rangle \longrightarrow |0\rangle|f(x)\rangle.$$

One drawback of this simple construction is the inflation of the number of qubits needed to accomplish the computation in question. In particular, since each Toffolli gate requires at least one clean qubit, the number of qubits required is proportional to the size of the classical circuits involved. It is possible to improve this space bound at the expense of the other parameters [**?**], but we shall be primarily concerned with minimizing the overall size and depth of our circuits. See also [**?**] for quantum arithmetic circuits constructed with an emphasis on minimizing this space requirement.

Addition and subtraction can both be accomplished by classical circuits of size and depth $O(n)$ and $O(\log n)$ respectively. By Lemma 2, then, the quantum addition circuit

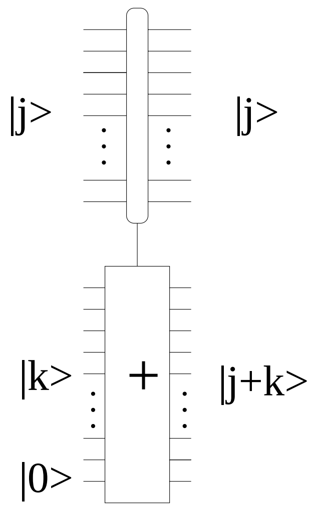


Figure 1.5: Quantum Addition

pictured in 1.5 also has size $O(n)$ and depth $O(\log n)$. We shall also use a modular addition operation, denoted $+_N$, which maps $|j\rangle|k\rangle$ to $|j\rangle|j+k \bmod N\rangle$ for $j, k < N$. It is easy to see that this bijection can be accomplished with asymptotic size and depth identical to regular addition. Finally, we will have occasion to run each of these circuits backwards, performing $-$ and $-_N$ respectively.

We shall also require two types of multiplication circuits. The first, pictured in Figure 1.6, takes as input an n-bit decimal $d > 1$, an n-bit integer j, and an integer k with $0 \leq k < d$. It outputs the nearest to $dj + k$ with ties broken by some consistent convention. The requirement $d > 1$ ensures that this map is a bijection and thus Lemma 2 can be invoked. We let $\div$ denote this multiplication circuit run in reverse. Analyzing the circuit's size and depth is more complicated than in the case of addition. Currently the best classical circuits for multiplication have size $O(n \log n \log\log n)$ and depth $O(\log n)$. But in order to apply Lemma 2 we must also perform division, a notoriously stubborn operation to

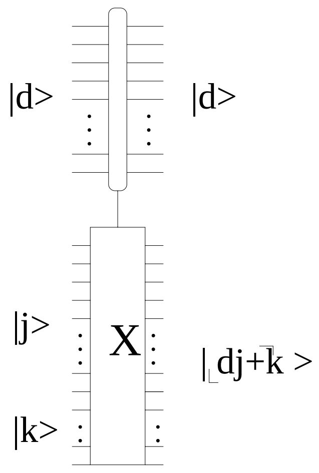


Figure 1.6: Quantum Multiplication with Remainder

parallelize. There are classical division circuits of size $O(n \log n \log \log n)$ which have depth $O(\log n \log \log n)$[**?**]. However, the smallest $O(\log n)$-depth classical division circuits have size $O(n^{1+\epsilon})$[**?**]. Thus the quantum multiplication circuit (1.6) can *either* be performed in simultaneous size and depth $O(n \log n \log \log n)$ and $O(\log n \log \log n)$ *or* $O(n^{1+\epsilon})$ and $O(\log n)$.

If an n-bit approximation to $1/d$ is available – in Algorithm 3 we can prepare this inverse classically – multiplication by $1/d$ can be substituted for division. The multiplication circuit pictured in Figure 1.7 can thus be performed in simultaneous size and depth $O(n \log n \log \log n)$ and $O(\log n)$ respectively.

The classical multiplication and division techniques which achieve these various sub-quadratic circuit sizes all make use of the discrete Fourier transform. This raises the interesting question of whether there is some inherently quantum method which improves upon these techniques, perhaps by using the QFT. As noted in [**?**] it could allow quantum

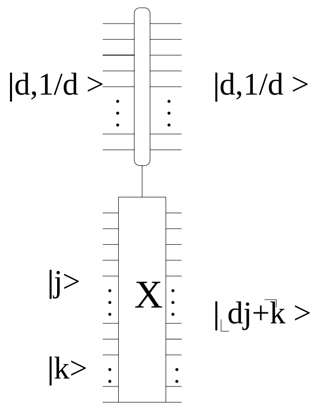


Figure 1.7: Quantum Multiplication with Inverse

*de*coding of RSA encryption to run asymptotically faster than the corresponding classical *en*coding.

Chapter 2

Quantum Fourier Transforms and The Hidden Subgroup Problem

2.1 The Discrete Fourier Transform

Let G be a finite abelian group and let V be the vector space over the complex numbers consisting of formal linear combinations of elements of G,

$$|v\rangle = \sum_{g \in G} v_g |g\rangle.$$

We use $*$ to denote the group convolution operation induced by G on V, that is, the operation

$$|v\rangle * |w\rangle = \sum_{g \in G} \left(\sum_{hk=g} v_h w_k \right) |g\rangle.$$

Notice that the $|g\rangle \in G$ form a group under this operation which is trivially isomorphic to G itself.

The discrete Fourier transform, or DFT, is a symmetric unitary transformation F

of V satisfying

$$cF\left(|g\rangle * |h\rangle\right) = F|g\rangle \cdot F|h\rangle \tag{2.1}$$

for all $g, h \in G$, where $\cdot$ denotes pointwise vector multiplication and c is the normalization factor $1/\sqrt{|G|}$. The DFT thus exhibits a group isomorphism between the $|g\rangle \in G$ under group convolution and the $|F(g)\rangle \in G$ under pointwise multiplication. This characterization of the DFT is sufficient for the applications discussed in this thesis. For the definition of the DFT in terms of group representations, still in the setting of quantum computation, see [**?**].

Cyclic G

If $G = Z_N$, the cyclic group on N elements, then the transformation

$$|j\rangle \longrightarrow |k\rangle \text{ with amplitude } \frac{1}{\sqrt{N}}\omega_N^{jk}.$$

where $\omega_N = e^{\frac{2\pi i}{N}}$ satisfies (2.1). In fact, ω_N could be replaced with any primitive Nth root of unity and the resulting map would still satisfy this condition. It is not hard to show that these are the only such transformations and thus our characterization yields a unique DFT up to isomorphism of the underlying cyclic group.

Example: A simple example is the DFT over Z_2, which is the map sending

$$|0\rangle \longrightarrow \frac{1}{\sqrt{2}}\omega_2^{0\cdot 0}|0\rangle + \frac{1}{\sqrt{2}}\omega_2^{0\cdot 1}|1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle.$$

and

$$|0\rangle \longrightarrow \frac{1}{\sqrt{2}}\omega_2^{1\cdot 0}|0\rangle + \frac{1}{\sqrt{2}}\omega_2^{1\cdot 1}|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

The matrix representation of this map is thus

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}. \tag{2.2}$$

■

Finite Abelian G

By the Fundamental Theorem on finite Abelian groups, any such G can be decomposed as a direct product of cyclic subgroups. Its DFT is the tensor product of the DFT's corresponding to each cyclic subgroup in this decomposition. Again it is possible to show that there is a unique map satisfying (2.1) up to isomorphism of the underlying group. Before giving a description of these maps we present a simple example.

Example: The simplest example of this tensor product construction is the DFT over $\bigoplus_n Z_2 = (Z_2)^n$, sending

$$|j\rangle \longrightarrow |k\rangle$$

with amplitude

$$\prod_{i<n} \frac{1}{\sqrt{2}} \omega_2^{j_i k_i} = \frac{1}{\sqrt{2^n}} \omega^{j \cdot_2 k},$$

where $\cdot_2$ denotes the mod 2 dot product. ■

More generally if $G = \bigoplus_{i<n} Z_{p_i}$ is an arbitrary finite abelian group given by its decomposition as a direct product of cyclic groups we can describe the DFT over G in a uniform manner. We first define the mod G dot product, $\cdot_G$ as follows.

Definition 1. *Suppose $G = \bigoplus_{i<n} Z_{p_i}$. Let $P = |G| = p_1 p_2 \cdots p_n$ and $P_i = P/p_i$ Then $\cdot_G$*

is the binary operation on G given by

$$g \cdot_G h = \frac{1}{P}\left[\left(\sum_{0<j\leq n} P_j g_j h_j\right) \bmod P\right],$$

where g and h equal $(g_1, g_2, \ldots, g_n)$ and $(h_1, h_2, \ldots, h_n)$ respectively.

The value of this definition is that the DFT over G can now be simply described as sending

$$|g\rangle \longrightarrow |h\rangle \text{ with amplitude } \frac{1}{\sqrt{|G|}}\omega^{g\cdot_G h}.$$

DFT vs. QFT

The classical task of computing the DFT over Z_N of an explicitly given vector of complex numbers $v = (v_1, v_2, \ldots, v_N)$, a task which naively appears to require $O(N^2)$ arithmetic operations, can actually be accomplished in $O(N \log N)$ arithmetic operations, by techniques referred to as the fast Fourier transform, or FFT (See Sections 3.2.1 and 3.3). This nontrivial algorithm, together with the fact that the DFT maps $*$, i.e. group convolution, to $\cdot$, pointwise multiplication, is exploited by the many classical applications of the DFT, such as Fast Polynomial and Integer Multiplication.

In contrast to this classical computational task, the quantum Fourier transform or QFT refers to the implementation of the discrete Fourier transform on the underlying quantum state space. In other words, the input is not an explicit vector of complex values, but a quantum state

$$|\alpha\rangle = \sum_{i<N} \alpha_i |i\rangle$$

whose *amplitudes* represent the vector to be transformed. The output of the QFT over Z_N

is the quantum state

$$|\hat{\alpha}\rangle = \sum_{j<N} \hat{\alpha}_j |j\rangle,$$

where

$$\hat{\alpha}_j = \frac{1}{\sqrt{N}} \sum_{i<N} \alpha_i \omega_N^{ij}.$$

Example: It is not hard to see that one of our basic quantum gates, namely the Hadamard (See Equation 1.2 and Example 2.1) is precisely the QFT over Z_2. Moreover, applying n Hadamards independently to each qubit as pictured in Figure 2.1 accomplishes

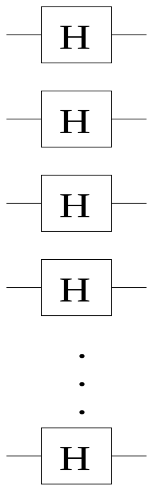


Figure 2.1: QFT over $(Z_2)^n$.

the QFT over $(Z_2)^n$, which we will denote $(F_2)^n$. This is the first and simplest example of a polynomial sized quantum circuit implementing the QFT over an exponential-sized group.

■

The fact that the QFT over exponentially large groups can be efficiently implemented is the basis for for all quantum algorithms achieving exponential advantage over classical computation. However, it is important to notice that, in and of itself, the ability to perform the QFT over an exponentially large group does not represent an exponential speedup of any classical task, such as DFT computation. This contrast between the classical DFT and the QFT has been likened to, on the one hand, producing a list of all the probabilities of points in the sample space of some distribution (the classical DFT) and, on the other hand, producing a method for sampling from that distribution (the QFT). We now turn to a situation where the ability to compute the quantum Fourier transform over an exponentially large group *does* give quantum computation advantage over classical, an astounding exponential advantage to be precise.

2.2 Simon's Algorithm

Simon [?] gave a polynomial-time quantum algorithm for the following promise problem.

GIVEN: A function f defined on $(Z_2)^n$ which is 2-1 and satisfies $f(x) = f(x \oplus b)$ for all x and some fixed b.

FIND: b.

This is our first example of a hidden subgroup problem. The given function f is defined on the group $(Z_2)^n$ and constant on cosets of the unknown subgroup $\{0, b\}$. The goal is to reconstruct this subgroup. The following quantum procedure is commonly referred to as Simon's algorithm. It exploits a certain coset invariance property of the QFT – regardless of which coset x of $\{0, b\}$ is input to the QFT at Step 2, the output distributions are identical The information about the particular coset is concentrated in the complex phases of the final superposition, while its distribution encodes just the underlying subgroup $\{0, b\}$.

Algorithm 1. *Simon's Algorithm*[1]

1. *We prepare the input to the Fourier transform as follows:*

$$|0\rangle|0\rangle \longrightarrow \sum_{x<2^n} |x\rangle|0\rangle \longrightarrow \sum_{x<2^n} |x\rangle|f(x)\rangle = \sum_{a\in R_f} (|x\rangle + |x\oplus b\rangle)\,|a\rangle$$

where R_f denotes the range of f.

2. *Quantum Fourier Transform over $\bigoplus_n Z_2$:*

$$\begin{aligned}(|x\rangle + |x\oplus b\rangle)\,|a\rangle \quad &\longrightarrow \quad \sum_{y<2^n} \left((-1)^{x\cdot y} + (-1)^{(x\oplus b)\cdot y}\right)|y\rangle|a\rangle \\ &= \quad \sum_{y,y\cdot_2 b=0} (-1)^{x\cdot y}|y\rangle|a\rangle,\end{aligned}$$

where $\cdot_2$ denotes the mod 2 dot product.

3. *Measure the first register.*

Repeat this quantum subroutine $O(n)$ times and obtaining $\{y_i\}$. Solve(classically) the system of equations $y_i \cdot_2 z = 0$. Output this solution

Our quantum subroutine outputs a y_i uniformly at random from the set of y such that $y \cdot_2 b = 0$. It is not hard to show that after $O(n)$ repetitions of the subroutine the resulting system of linear equations will have a unique solution with high probability and the correctness of the algorithm follows.

It is possible to show that any classical probabilistic algorithm for this problem has query complexity $\Omega(2^{\frac{n}{2}})$ [?], thus this is an example of a (promise) problem where quantum computation achieves exponential advantage over classical computation. In fact,

[1]In this and all later quantum procedures we shall feel free to suppress global normalization factors in order to preserve readability.

we show in Chapter 8, Section 8.2 that, even in the presence of non-determinism, any classical probabilistic method of distinguishing functions that are $1-1$ on $(Z_2)^n$ from the $2-1$ functions described above requires a similar exponential number of queries. In contrast, this can be accomplished in polynomial-time on a quantum computer by a slight modification of Algorithm 1.

2.3 Generalizing Simon's Algorithm: The Abelian Hidden Subgroup Problem

Algorithm 1 is the prototype for all later hidden subgroup algorithms, including Shor's celebrated algorithms factoring and discrete logarithm[**?**]. We reinterpret Algorithm 1 in terms of "Fourier sampling" over $G = \bigoplus_n Z_2$, then show how this procedure generalizes to an arbitrary finite Abelian group. Our approach is similar to that of [**?**].

Algorithm 2. *1. Prepare*

$$|\alpha\rangle = \sum_{x \in G} |x\rangle |f_H(x)\rangle$$

where f_H is constant and distinct on the cosets of $H \le G$

2. Sample from $\mathcal{D}_{F_G|\alpha\rangle}$, the distribution gotten by measuring the first register of

$$F_G|\alpha\rangle = \sum_{x \in G} F_G|x\rangle|f_H(x)\rangle,$$

where F_G denotes the QFT over G.

Repeat this quantum subroutine $O(n^2)$ times where $n = \log|G|$ obtaining samples $\{y_i\}$. Solve(classically) the system of equations $y_i \cdot_G z = 0$. Output this solution

We can describe the distribution sampled by the quantum procedure using the following definition:

Definition 2. *Let G be a finite abelian group. For any subgroup $H \leq G$ let $H^{\perp} \leq G$ be the subgroup consisting of the elements $g \in G$ satisfying*

$$g \cdot_G h = 0$$

for all $h \in H$.

In the particular case $G = \bigoplus_n Z_2$ and $H = \{0, b\}$, i.e. Simon's algorithm, we have already seen that the distribution $\mathcal{D}_{F_G|\alpha\rangle}$ is supported uniformly on the subgroup $H^{\perp}$. We now show that for *any* finite abelian G and $H \leq G$, $\mathcal{D}_{F_G|\alpha\rangle}$ is uniformly supported on $H^{\perp}$.

Recall that

$$|\alpha\rangle = \sum_{x \in G} |x\rangle |f_H(x)\rangle.$$

For any $g \in G$ let

$$|g * \alpha\rangle = |g\rangle * |\alpha\rangle = \frac{1}{\sqrt{|G|}} \sum_{x \in G} |g + x\rangle |f_H(x)\rangle,$$

be the convolution of $|g\rangle$ and the first register of $|\alpha\rangle$. Then for any $h \in H$,

$$cF_G|\alpha\rangle = cF_G|h * \alpha\rangle = F_G|h\rangle \cdot F_G|\alpha\rangle, \tag{2.3}$$

for $c = \frac{1}{\sqrt{|G|}}$. The first equality follows from the fact that, since f_H is constant on cosets of H, $|\alpha\rangle = |h * \alpha\rangle$, and the second from our definition of the Fourier transform. Since the amplitude of $F_G|h\rangle$ at $|x\rangle$ is $\frac{1}{\sqrt{|G|}}\omega^{x \cdot_G h}$ this equality can hold only if $F_G|\alpha\rangle$ is supported on $|x\rangle$ satisfying $\omega^{x \cdot_G h} = 1$ for all $h \in H$. This is precisely $H^{\perp}$ as claimed.

Showing that the distribution is uniform on $H^{\perp}$ requires more detail. Fix any state $|y\rangle|a\rangle$ with $y \in H^{\perp}$ and a in the range of f_H. The amplitude at this point is determined by the Fourier transform of the superposition

$$\frac{1}{\sqrt{|G|}} \sum_{f(x)=a} |x\rangle|a\rangle = \frac{1}{\sqrt{|G|}} \sum_{h \in H} |k+h\rangle|a\rangle$$

for some fixed k, since f_H is both constant and distinct on the cosets of H. The resulting amplitude at $|y\rangle|a\rangle$ is thus

$$\frac{1}{|G|} \sum_{h \in H} \omega^{(k+h)\cdot_G y} = \frac{1}{|G|} \omega^{k \cdot_G y} \left(\sum_{h \in H} \omega^{h \cdot_G y} \right) = \omega^{k \cdot_G y} \frac{|H|}{|G|}$$

where the last equality follows from the assumption that $y \in H^{\perp}$. Clearly the norm of this amplitude is independent of y (and a), whose influence is only seen in the complex phase $\omega^{k \cdot_G y}$, and the probabilities arising from the squares of these norms are thus uniformly distributed over $H^{\perp}$.

How many samples are required in order to generate $H^{\perp}$ and thus solve for the generators of H? In the special case of $G = \bigoplus_n Z_2$ and $H = \{0, b\}$ a simple argument shows that $O(n)$ samples suffice: For any n-bit $y \neq b$ the probability that a random element of $x \in \{0, b\}^{\perp}$ satisfies $y \cdot_2 x = 0$ is at most $1/2$. Since there are $2^n - 1$ such y the probability that b is not uniquely determined falls off as $2^n (1/2)^t$ where t is the number of samples.

In general we need to ensure that our samples samples are not contained in any proper subgroup of $H^{\perp}$. This is achieved after $O(n^2)$ samples where $n = \log |G|$. In particular, there are at most $|G|^n < 2^{n^2}$ such subgroups – each is determined by a set of at most n generators chosen from G. Moreover, each has size at most half of $H^{\perp}$. Thus the probability that there exists one such subgroup containing all t samples decreases as

$$2^{n^2} \left(\frac{1}{2} \right)^t .$$

In theory, then, Algorithm 2 solves the hidden subgroup problem for any finite abelian group G. But so far we have only seen circuits implementing this procedure in the special case $G = \bigoplus_n Z_2$ (Figure 2.1). Step 1 requires the preparation of an equal superposition over the group G and an evaluation of the function f_H. This is easily accomplished. Generalizing Step 2 hinges upon extending the class of groups with efficient QFT's. More specifically, the class of cyclic groups with efficient QFT's must be extended since these can be tensored together to produce the QFT over an arbitrary finite abelian group. We turn to this topic in the next Chapter.

Chapter 3

Computing the Quantum Fourier Transform

3.1 The QFT over Z_N, N Smooth

Two separate methods emerged for extending the class of cyclic groups with efficient QFT's. The first, developed by Shor [?] and subsequently Cleve [?], was based on the recognition that the component QFT's over Z_2 in the circuit for the QFT over $\bigoplus_n Z_2$ could be replaced by QFT's over any sufficiently small cyclic group. In particular, since any n-bit unitary operation can be approximated by exponential-size quantum circuits via Theorem 1, the QFT over Z_m can always be approximated by a circuit of size $O(m^2)$. It follows that the QFT over any group of the form $\bigoplus_{i<n} Z_{m_i}$ where the $m_i = O(n^k)$ can be efficiently computed. More importantly, this insight allows us to efficiently compute the QFT over a special class of exponentially large cyclic groups. In particular, suppose that

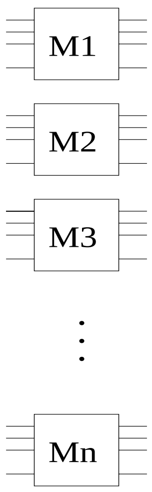


Figure 3.1: QFT over $\bigoplus_{i<n} Z_{m_i}$.

$N = m_1 m_2 ... m_k$ where the m_i are pairwise relatively prime. By the Chinese remainder theorem we have $Z_N \cong \bigoplus_{i<k} Z_{m_i}$ via the isomorphism

$$a \bmod N \longrightarrow (a \bmod m_1, a \bmod m_2, \ldots, a \bmod m_k). \tag{3.1}$$

This isomorphism is easy to compute, and can be inverted as well using the formula

$$a \bmod N = \sum_{i<k} (a \bmod m_i) N_i \left(N_i^{-1} \bmod m_i \right), \tag{3.2}$$

where $N_i = N/m_i$.

Thus if we are given an N which factors into pairwise relatively prime m_i satisfying $m_i = O(\log^k N)$, we can compute the QFT mod Z_N by first computing 3.1, then performing the QFT over $\bigoplus_{i<k} Z_{m_i}$, and then inverting by 3.2. Obviously we need to know the factorization $m_1 m_2 \cdots m_k$ but these can easily be computed classically since they are so small. This reasoning thus extended the class of groups whose QFT could be efficiently

implemented to include cyclic groups Z_N for N smooth, i.e. with prime power factors all of size $O(n = \log N)$[?], and, more generally, for any N with prime power factors equal to $O(n^k)$[?].

3.2 The QFT over Z_{2^n}

The second method was developed independently by Coppersmith [?] and Deutsch. By exploiting the same recursive structure of the DFT over Z_{2^n} which leads to the classical FFT, the QFT over Z_{2^n} can be computed exactly by a quantum circuit of size and depth n^2. As in the classical setting the method can be generalized with only minor changes to any $N = c^m$ where c is a constant.

The recursive structure of the DFT over Z_{2^n} is encapsulated by the following product representation. Let $j = j_1 j_2 \cdots j_n$ is be the bit representation of j where j_1 is the most significant. Then the DFT over Z_{2^n} can be written as

$$|j\rangle \longrightarrow \frac{\big(|0\rangle + \omega^{.j_n}|1\rangle\big)\big(|0\rangle + \omega^{.j_{n-1}j_n}|1\rangle\big)\cdots\big(|0\rangle + \omega^{.j_1 j_2 \cdots j_n}|1\rangle\big)}{2^{n/2}}. \tag{3.3}$$

We digress briefly to show how to derive the classical FFT from this expression.

3.2.1 The Classical FFT

The classical FFT algorithm was first proposed in [?] but its motivation goes back to Gauss. The product representation of Equation 3.3 lends itself most nicely to the decimation-in-frequency, as opposed to decimation-in-time, version of the classical FFT. Suppose the input to the DFT is the vector $|v\rangle$. Let $|w\rangle$ and $|z\rangle$ be the length 2^{n-1} vectors with amplitudes

$$w_j = v_{0j} + v_{1j}$$

and

$$z_j = \omega^{\cdot 0j} v_{0j} + \omega^{\cdot 1j} v_{1j}.$$

Then the amplitude of the product expression (3.3) at an even integer $|k\rangle = |k_1 k_2 \cdots k_n\rangle$ is just the amplitude at $|k_1 k_2 \cdots k_{n-1}\rangle$ of the DFT over 2^{n-1} of $|w\rangle$. Furthermore, the amplitude at an odd $|k\rangle$ is just the amplitude at $|k_1 k_2 \cdots k_{n-1}\rangle$ of the DFT over 2^{n-1} of $|z\rangle$.

The DFT over 2^n can then be gotten by performing these 2 related DFT's over 2^{n-1}. Computing the vectors $|w\rangle$ and $|z\rangle$ from $|v\rangle$ requires $O(N)$ arithmetic operations. Thus we get the recurrence relation

$$T(N) = 2T(N/2) + O(N)$$

for the arithmetic complexity of the classical DFT over $N = 2^n$, leading to the well-known bound of $O(N \log N)$.

Recall our definition of the Fourier transform as a map taking convolution to pointwise multiplication and back again. Computing the convolution of two vectors $|v\rangle$ and $|w\rangle$ in a brute force manner requires $2N$ arithmetic operations for each amplitude

$$\sum_{i<N} v_i w_{l-i}$$

and thus $2N^2$ for the vector as a whole. If we instead perform an FFT, pointwise multiply, then invert the FFT, only $O(N \log N)$ arithmetic operations are involved. This rather startling fact is the basis for the well known fast polynomial and integer multiplication algorithms [?].

3.2.2 The QFT over Z_{2^n}

The product representation (3.3) leads even more directly to the following quantum gate array which computes the QFT mod Z_{2^n}. exactly. This gate array has size and depth

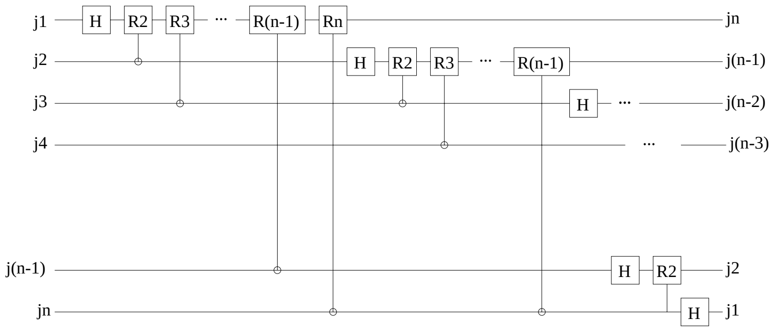


Figure 3.2: QFT over Z_{2^n}.

$\frac{n(n+1)}{2} = O(n^2)$. The gates can easily be rearranged so that the circuit has depth $O(n)$[?]. In particular, let G_{ij} for $i < j$ denote the controlled-rotation gates R_{j-i+1} whose inputs are the *ith* and *jth* wires in Circuit 3.2. Also let G_{ii} be the Hadamard gate which is applied to the *ith* bit. It is not hard to see the only requirement imposed by the above circuit is that whenever $i + j < i' + j'$, G_{ij} must precede $G_{i'j'}$ in the computation. By arranging the gates in $2n - 1$ stages, where at the kth stage the all the gates G_{ij} with $i + j = k$ are performed in parallel, the exact QFT over 2^n can be performed in size $\frac{n(n+1)}{2} = O(n^2)$ and depth $2n + 1 = O(n)$.

In practice, we are interested in merely approximating the QFT to within an arbitrary inverse polynomial. Since most of the rotation gates in the Circuit 3.2are very small, by just omitting the rotations in $O(\epsilon/n^2)$ a circuit of size and depth $O(n \log \frac{n}{\epsilon})$ which approximates the QFT over Z_{2^n} to within ϵ can be achieved [**?**]. In particular, when ϵ is an inverse polynomial this gives a gate array of size $O(n \log n)$. Since such approximations suffice for any polynomial-time computation, this is a clear benefit of this recursive technique over the technique of Section 3.1 for which there is no similar approximation technique. Unfortunately, this benefit applies only to the size of the circuits – the depth of the parallel version of Circuit 3.2 outlined in the previous paragraph is not further reduced by this omission of gates.

Shor's algorithms for factoring and discrete log can be based on either the QFT over Z_N for smooth N or the QFT over Z_{2^n}, but the inability to transform over an arbitrary cyclic group complicates their proof. While there is no direct way to extend either of these methods to encompass a larger class of cyclic groups, the reliance of the QFT over Z_{2^n} on the insight which led to the FFT over the same domain raises a natural question. We have an generalized classical FFT algorithm, i.e. an algorithm for computing the classical DFT over Z_N for arbitrary N which has arithmetic complexity $O(N \log N)$, identical to the standard FFT over a power of two. Why not try to base a QFT on these classical methods?

3.3 Quantum Chirp-Z.

Since the circuit performing the QFT over a power of 2 is derived from the classical FFT circuit, it is natural to try to derive a circuit for the QFT over a general modulus from the corresponding general modulus classical method. We first review this method, known as the chirp-z transform and attributed to Rabiner et al [**?**]. We then translate this approach to the quantum setting and show that with a slight modification we do obtain an efficient ϵ-approximate QFT which succeeds with probability ϵ^2. On the one hand, this is not strong enough to be useful in a general setting – in particular, if an algorithm involves more than a constant number of QFT's replacing them all with these approximations would reduce the success probability to below an inverse polynomial. On the other hand, all the hidden subgroup algorithms to date use just a constant number of QFT's in each quantum subroutine and thus this method could be used. It will not be as efficient as the Eigenvalue Estimation procedure of Section 3.4 and our Algorithm 3 but may be of independent interest.

The classical chirp-z transform is essentially a method of reducing the transform over an arbitrary modulus to a combination of multiplication and convolution. The net result is that the transform over an arbitrary modulus N with $n = \lfloor \log N \rfloor$ can be accomplished via 3 FFT's over 2^{n+2} together with $O(N)$ extra arithmetic operations. Thus the asymptotic arithmetic complexity of the general modulus DFT is the same as that of a power of two, namely $O(N \log N)$.

We now describe this in some detail. Given $|a\rangle = \sum_{i<N} a_i |i\rangle$ we wish to compute the vector $|\hat{a}\rangle$ where

$$\hat{a}_j = \sum_{i<N} a_i \omega_N^{ij}.$$

We let

$$|b\rangle = \sum_{i<N} a_i \omega_N^{-i^2/2} |i\rangle \text{ and } |c\rangle = \sum_{i<2^{n+1}} \omega_N^{i^2/2} |i\rangle.$$

Clearly $|b\rangle$ can be generated from $|a\rangle$, and $|c\rangle$ created, using $O(N)$ arithmetic operations.

The crucial insight is that kth convolution coefficient of $|b\rangle$ and $|c\rangle$,

$$d_k = \sum_{i<2^{n+1}} b_i c_{k-i},$$

satisfies

$$d_k \omega_N^{-k^2/2} = \sum_{i<N} a_i \omega_N^{ik} = \hat{a}_{k-N}$$

whenever $k \geq N$. Thus the convolution vector $|d\rangle$ can be used to produce the desired vector $|\hat{a}\rangle$ via $O(N)$ arithmetic operations. As discussed in Section 3.2.1, the convolution vector $|d\rangle$ is obtained by computing the FFT mod 2^{n+1} of the vectors $|b\rangle$ and $|c\rangle$, pointwise multiplying the two resulting vectors, and computing the inverse FFT of this product.

This method uses $O(N)$ arithmetic operations to create the vectors $|b\rangle$ and $|c\rangle$, perform the pointwise multiplications which are sandwiched between the FFT's, and recover the Fourier coefficients from the convolution coefficients. Moreover it involves a total of three FFT's over 2^{n+1}, leading to an overall arithmetic complexity of $O(N \log N)$.

Is it possible to implement this type of convolution reduction in the quantum setting? Recall that in this case we are given as input the superposition $|\alpha\rangle = \sum_{i<N} \alpha_i |i\rangle$ and we wish to output the superposition $|\hat{\alpha}\rangle$ where

$$\hat{\alpha}_i = \sum_{i<N} \alpha_i \omega_N^{ij}.$$

The superpositions analogous to $|b\rangle$ and $|c\rangle$ above, namely

$$|\beta\rangle = \sum_{i<N} \alpha_i \omega_N^{-i^2/2} |i\rangle \text{ and } |\gamma\rangle = \sum_{i<2^{n+1}} \omega_N^{i^2/2} |i\rangle,$$

can be created easily. For example, the map

$$|\alpha\rangle \longrightarrow |\beta\rangle$$

is achieved by computing $\frac{-i^2}{2N}$, putting this value into the phase, and then erasing it.

Convolution of these two superpositions poses a problem. We can perform the required QFT's over 2^{n+2} yielding the superposition

$$|\hat{\beta}\rangle|\hat{\gamma}\rangle = \sum_{i,j<2^{n+2}} \hat{\beta}_i\hat{\gamma}_j|i\rangle|j\rangle. \tag{3.4}$$

We desire the superposition $\sum_{i<2^{n+2}} \hat{\beta}_i\hat{\gamma}_i|i\rangle$ corresponding to the pointwise multiplication of $|\hat{\beta}\rangle$ and $|\hat{\gamma}\rangle$, but the best we can do is to subtract the first register in 3.4 from the second and measure this difference, yielding

$$\sum_{i<2^{n+2}} \hat{\beta}_i\hat{\gamma}_j|i\rangle|j-i\rangle \tag{3.5}$$

for each possible $(j-i)$ with equal probability. If $(j-i)=0$ then this is the desired superposition and taking the inverse QFT over 2^{n+2} completes the convolution. We then finish the algorithm by collapsing the superposition to the interval $\{N,\ldots,2N-1\}$ and shifting the phase at $|l\rangle$ by $\omega_N^{-l^2/2}$.

But the probability that $(j-i)=0$ is exponentially small. More likely the value in the second register will be some non-zero $h=j-i$. Then the output of the algorithm corresponds to having convolved, instead of the desired superpositions $|\beta\rangle$ and $|\gamma\rangle$, the superpositions $|\beta'\rangle$ and $|\gamma\rangle$ where

$$\beta'_i = \alpha_i\omega_{2^{n+1}}^{ih}\omega_N^{-i^2}.$$

Thus if we collapse to the appropriate interval $\{N,\ldots,2N-1\}$ and shift phases as described above we will have computed the transform over N of the superposition $|\alpha'\rangle$ where

$$\alpha'_i = \alpha_i\omega_{2^{n+1}}^{ih}$$

instead of the desired $|\hat{\alpha}\rangle$.

Now, if

$$\left|\omega_{2^{n+1}}^{h} - \omega_N^k\right| < \frac{\epsilon^2}{N} \tag{3.6}$$

for some integer k then the superposition $|\alpha'\rangle$ defined above and the superposition $|\alpha^k\rangle$ with amplitudes

$$\alpha_i' = \alpha_i \omega_N^{ik}$$

have distance at most $O(\epsilon)$. This is most easily be seen by observing that their inner product is large. The transform of $|\alpha^k\rangle$ over N is just the shift $\pmod N$ by k of $|\hat{\alpha}\rangle$, and thus the transform of $|\alpha'\rangle$ over N shifted by k is $O(\epsilon)$-close to the desired $|\hat{\alpha}\rangle$ whenever Equation 3.6 holds.

Since the k which minimizes the difference in Equation 3.6 can be ascertained from h, whenever this difference is suitably small we can perform the required shift and achieve an ϵ-approximation to $|\hat{\alpha}\rangle$. Since the condition of Equation 3.6 holds for a ϵ^2 fraction of the h the success probability is as claimed.

3.4 Eigenvalue Estimation

Kitaev [?] gave the first algorithm approximating the QFT over an arbitrary cyclic group based on his method of Eigenvalue Estimation. These techniques were further refined in ([?],[?],[?]). Our presentation of this QFT algorithm merges some of these later refinements with Kitaev's original approach.

We first note that we can perform the map

$$|i\rangle|0\rangle \longrightarrow |i\rangle \sum_{j<N} \omega_N^{ij}|j\rangle = |i\rangle|\hat{i}\rangle. \tag{3.7}$$

Specifically, we begin by putting the second register into an equal superposition over an appropriately large interval and computing $\frac{ij}{N}$. This value is then placed into the phase and the computation of $\frac{ij}{N}$ is erased.

More interestingly, it is also possible to approximate the map

$$|\hat{i}\rangle|0\rangle \longrightarrow |\hat{i}\rangle|i\rangle. \tag{3.8}$$

By combining the map 3.7 with 3.8 in reverse we achieve an approximation to the desired transform.

Map 3.8 is based upon a procedure for estimating the eigenvalues of a unitary operator. More specifically, suppose that we are able to perform the operations controlled-U, controlled-U^2, ... , controlled-U^{2^k} for some unitary operator U. Assume further that we are given an eigenvector $|\phi\rangle$ of U with eigenvalue ω^λ. Circuit 3.3 allows us to determine the most significant bits of λ with high probability. In particular, if λ is exactly k-bits then the input to the QFT in Circuit 3.3 is exactly $F_{2^n}^{-1}|\lambda\rangle$ and the procedure produces λ with probability 1. More generally, to achieve the first m bits of λ with probability at least $1-\epsilon$ it suffices to choose $k = m + O(\log(1/\epsilon))$. How does this enable us to approximate 3.8? It is easy to see that the the Fourier basis state $|\hat{i}\rangle$ of the QFT over Z_N is an eigenvector with eigenvalue $\omega^{i/N}$ of the unitary operator $U = (+1 \bmod N)$. Thus we can use the above circuit to recover i/N from $|\hat{i}\rangle$ with high probability. Multiplying this result by N allows

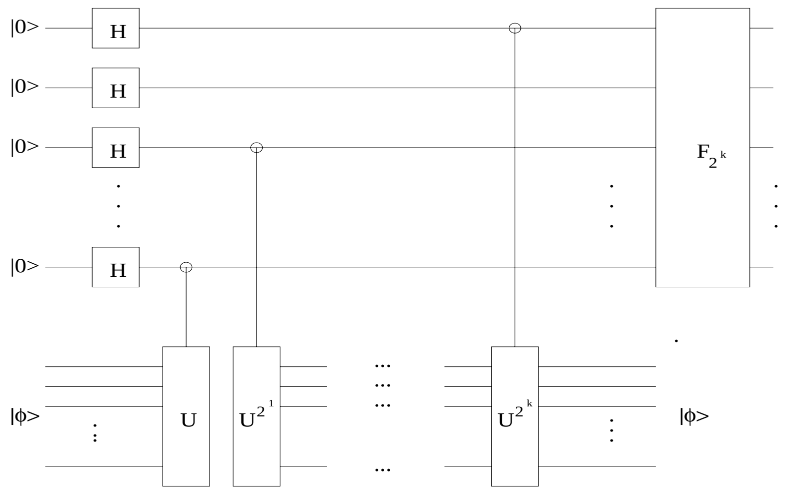


Figure 3.3: Eigenvalue Estimation.

us to approximate the map

$$|\hat{i}\rangle|0\rangle|0\rangle \longrightarrow |\hat{i}\rangle|i\rangle|junk_i\rangle \tag{3.9}$$

where the last bits are junk deriving from the rounding off of i/N to its most significant bits.

The junk produced by a map such as 3.9 can always be cleaned up using the methods outlined in the proof of Lemma 1, yielding an approximation to 3.8. In general, if the original map is accurate to within ϵ, the junkless version produced by this method will be accurate to within $\sqrt{N}\epsilon$. In this particular case, however, since a copy of the eigenvector $|\hat{i}\rangle$ is maintained throughout the computation, the errors produced will be orthogonal and maps 3.9 and 3.8 will have the same error bound. This seems to have been overlooked in [**?**] which mentions only the more general accuracy result.

This version of Kitaev's algorithm ostensibly has size and depth $O(n^2)$ and $O(n)$ respectively matching the running time of $O(n^2)$ claimed in [**?**].

Chapter 4

Parallel Circuits for the Quantum Fourier Transform over Z_{2^n}

The question of which quantum procedures can be performed in parallel, i.e. by circuits of polylogarithmic depth, is of both theoretical and practical interest. There are simple, natural problems, such as computing the greatest common divisor of two integers, which have no known classical parallelizations. Finding parallel quantum circuits for such a problem would further support and elucidate the apparent power of quantum over classical computation. On the practical side, parallel computations can significantly reduce the computational cost of fault-tolerant implementations of quantum algorithms. In particular, a robust model of computational noise must assume that an error can occur in a qubit at a given stage in time whether or not the qubit is undergoing a gate transformation at that particular stage. Under this assumption the size of the fault-tolerant implementation of a parallel circuit – see for example [**?**], Chapter 10 – will be smaller than the fault-tolerant implementation of the non-parallel version by as much as a factor of $O(n)$, where n is the number of qubits in the original non-parallel circuit.

We give explicit parallel circuits for approximating the QFT over a power of 2 to within an arbitrary inverse polynomial. The existence of such circuits with simultaneous size and depth $O(n \log n)$ and $O(\log n)$ respectively was proved in [**?**]. Our construction simplifes their approach and reduces the number of qubits required from $O(n \log n)$ to $O(n)$. In some sense this shows that the approximate QFT is inherently parallel, since there is no price to be paid for parallelization – asymptotically the size *and* width of the parallel circuits are the same as the apparently optimal nonparallel construction.

Our construction uses three basic maps, each of which can be approximated by shallow depth circuits. The first is the map

$$|j\rangle|0\rangle \longrightarrow |j\rangle|\hat{j}\rangle, \tag{4.1}$$

which we shall refer to as the quantum Fourier state computation, QFS for short, in keeping with [**?**]. The second is the map

$$|\hat{j}\rangle|0\rangle \longrightarrow |\hat{j}\rangle|\hat{j}\rangle, \tag{4.2}$$

which copies a Fourier basis state. Last and most interesting is the map

$$|j\rangle|\hat{j}\rangle|\hat{j}\rangle|\hat{j}\rangle = |j\rangle|\hat{j}\rangle^3 \longrightarrow |0\rangle|\hat{j}\rangle^3 \tag{4.3}$$

which erases the identity of a Fourier basis state from just three copies of that state. We refer to this as Fourier phase estimation or FPE again in keeping with [**?**]. It is easy to compose these maps to produce a QFT in the following manner

$$|j\rangle|0\rangle \xrightarrow{\text{QFS}} |j\rangle|\hat{j}\rangle \overset{\text{QCOPYx2}}{\longrightarrow} |j\rangle|\hat{j}\rangle^3 \xrightarrow{\text{FPE}} |0\rangle|\hat{j}\rangle^3 \overset{\text{reverse QCOPYx2}}{\longrightarrow} |\hat{j}\rangle|0\rangle$$

Shallow circuits for map 4.2 and an approximation to map 4.1 and were exhibited in [**?**]. Their method of Fourier phase estimation, however, uses $O(\log n)$ copies of the Fourier basis state $|\hat{j}\rangle$ to erase its identity $|j\rangle$. This required an ancilla of $O(n \log n)$ qubits and also complicated the task of copying $|\hat{j}\rangle$ – in order to make the required $O(\log n)$ copies in parallel classical results about prefix addition were required. By requiring only three copies of the Fourier basis state in our Fourier phase estimation we are able not only to reduce the qubit requirement but also to simplify the circuits to the point of making them explicit, modulo our basic repetoire of arithmetic operations (see Section 1.5.1). We first turn to this new Fourier phase estimation procedure 4.3, then give the circuits for maps 4.1 and 4.2, and finally show how to combine these with a simple preprocessing step to achieve an adequate approximation.

4.1 Fourier Phase Estimation

We now describe the circuit, pictured in Figure 4.1, which approximates the map

$$|j\rangle|\hat{j}\rangle^3 \longrightarrow |0\rangle|\hat{j}\rangle^3. \qquad (4.4)$$

A collection of exact QFT's modulo 2^{2k} for $k = O(\log n)$ are performed in parallel on the bits of the first and second copies of the Fourier basis state $|\hat{j}\rangle$. We assume for simplicity that $2k$ divides n. The first copy of the Fourier basis state undergoes $n/2k$ QFT's modulo 2^{2k}, applied in parallel to each consecutive sequence of $2k$ bits. The most significant k bits output by each QFT are used as an estimate for the corresponding bits of j and are thus xored into these bits to erase them. The second copy of the Fourier basis state undergoes $n/2k$ -1 QFT's modulo 2^{2k}, applied in parallel to each consecutive sequence of $2k$ bits beginning with the $k+1$st bit. As before, the leading k bits of each QFT are xored into the corresponding bits of j. The QFT computations are then reversed. The third copy of the Fourier basis state is left alone – its sole purpose is to ensure the orthogonality of errors from distinct basis states.

Recall we can compute the exact QFT modulo 2^l in size $l(l+1)/2$ and depth $2l-1$ as discussed in Section 3.2.2. Thus the above computation has depth $8k$ and size $O(kn)$. To analyze its error we must examine the input and output of each QFT modulo 2^{2k}. Without loss of generality we look at the topmost QFT which is applied to the first $2k$ bits of $|\hat{j}\rangle$, i.e the input is

$$\frac{1}{2^k}\left(|0\rangle + \omega^{.j_1 j_2 \cdots j_{2k} j_{2k+1} \cdots j_n}|1\rangle\right)\left(|0\rangle + \omega^{.j_2 \cdots j_{2k} j_{2k+1} \cdots j_n}|1\rangle\right)\cdots\left(|0\rangle + \omega^{.j_{2k} j_{2k+1} \cdots j_n}|1\rangle\right).$$

The output of the QFT modulo 2^{2k} on this input is a smeared pointmass concentrated at integers near the decimal $j_1 j_2 \cdots j_{2k}.j_{2k+1} \cdots j_n$. In particular, its amplitude at $|x\rangle$ is

$$\frac{1}{2^{2k}} \sum_{l<2^{2k}} \omega_{2^{2k}}^{l(x - j_1 j_2 \cdots j_{2k}.j_{2k+1} \cdots j_n)}.$$

This is the sum of 2^{2k} equally spaced vectors which wrap around the unit circle

$$|x - j_1 j_2 \cdots j_{2k}.j_{2k+1} \cdots j_n|_{2^{2k}}$$

times where $|\cdot|_{2^{2k}}$ is distance mod 2^{2k}. The complete revolutions effectively cancel out and the only contributions to the final amplitude come from the last fractional revolution.

There are $2^{2k}/|x - j_1 j_2 \cdots j_{2k}.j_{2k+1} \cdots j_n|_{2^{2k}}$ vectors in this fractional revolution, and each has length $\frac{1}{2^{2k}}$ leading to an amplitude which is at most a small constant times

$$\frac{1}{|x - j_1 j_2 \cdots j_{2k}.j_{2k+1} \cdots j_n|_{2^{2k}}}.$$

See the proof of Claim 3, Section 9.1.2 for a formal argument via geometric series of a similar bound.

It follows that the probability, i.e. total amplitude squared, of being more than t units away from $j_1 j_2 \cdots j_{2k}$ is $O(1/t)$. Since we are using the output of the QFT modulo 2^{2k} to estimate just the leading bits $j_1 j_2 \cdots j_k$, we need merely ensure that with high probability no carry into these first k-bits has occurred. In other words we need to bound the probability that the offset, t, combines with the bits $j_{k+1} j_{k+2} \cdots j_{2k}$ to induce such a carry. This probability is proportional to

$$\frac{1}{|j_{k+1} j_{k+2} \cdots j_{2k}|_{2^k}}. \tag{4.5}$$

Unfortunately, this expression is not always small. In particular if $j_{k+1} j_{k+2} \cdots j_{2k}$ is very close to zero mod 2^k then much of the smeared pointmass will be at points whose leading k bits differ from $j_1 j_2 \cdots j_k$. Fortunately, this will be a problem for only a small fraction of j and we will give a simple processing procedure to reduce the error arising from these bad basis states.

First we derive an expression for the total error arising from this circuit. Let $\mathbf{j_i}$ denote the ith sequence of k bits of j, that is, $j = \mathbf{j_1 j_2} \cdots \mathbf{j_{n/k}}$ and $\mathbf{j_i} = j_{ki+1} j_{ki+2} \cdots j_{k(i+1)}$. Then we can generalize the above reasoning to bound the squared error of our circuit on a fixed input $|j\rangle|\hat{j}\rangle^3$ by

$$\sum_{1<i<n/k} \frac{1}{|\mathbf{j_i}|_{2^k}}. \tag{4.6}$$

Since we have maintained a third copy of $|\hat{j}\rangle$ throughout the computation errors arising from different j are orthogonal. Thus the total squared error of the circuit on input

$$\sum_j \alpha_j |j\rangle|\hat{j}\rangle^3$$

is bounded by

$$\sum_j |\alpha_j|^2 \max\left(1, \left|\sum_{1<i<n/k} \frac{1}{|\mathbf{j_i}|_{2^k}}\right|^2\right).$$

We define a set of bad values j, denoted B, by letting $j \in B$ if there exists an i such that $|\mathbf{j_i}|_{2^k} < 2^{k/2}$. Then the above expression is less than

$$\begin{aligned} \sum_{j \notin B} |\alpha_j|^2 \left|\sum_{1<i<n/k} \frac{1}{|\mathbf{j_i}|_{2^k}}\right|^2 + \sum_{j \in B} |\alpha_j|^2 &\leq \sum_{j \notin B} |\alpha_j|^2 \frac{n^2}{2^k} + \sum_{j \in B} |\alpha_j|^2 & (4.7) \\ &\leq \frac{n^2}{2^k} + \sum_{j \in B} |\alpha_j|^2 . & (4.8) \end{aligned}$$

By choosing $k \in O(\log n)$ we can make the first of these two terms an arbitrary inverse polynomial. The second term is a problem. If the input superposition is supported on the set B then this term is one. On the other hand, the $j \in B$ form a small fraction of the whole – at most an $\frac{n}{2^{k/2}}$ fraction to be precise. Thus if the input α is fairly evenly distributed this second term will also be an arbitrary inverse polynomial for $k \in O(\log n)$. We will use a simple procedure – taking a random shift of our original superposition, computing the approximate QFT, and then undoing the effect of the shift – to mimic a uniformly distributed input and thus ensure that the overall error is polynomially small. We note that for many important applications, such as Shor's Factoring and Discrete Log algorithms, the input superposition is uniformly distributed to begin with and this procedure is not required. This will also be true when our parallel circuits for the QFT over an arbitrary modulus invoke the parallel circuits for the QFT over a power of 2 as a subroutine.

Finally we note that by overlapping the bit estimates from the 2 copies of $|\hat{j}\rangle$ and performing a $O(\log n)$- depth carrying procedure similar to that outlined in [?], one could get rid of this problematic second term entirely. However, the pre- and postprocessing procedures we have chosen are easier to express using our set of basic arithmetic circuits and also easy to omit when, as in the algorithms mentioned, it is unnecessary.

4.2 Quantum Fourier State Computation

We now turn to the task of approximating the map

$$|j\rangle|0\rangle \longrightarrow |j\rangle|\hat{j}\rangle,$$

using parallel circuits. The circuit pictured in Figure 4.2 computes this map exactly in depth n and size $O(n^2)$. By simply omitting the small rotations, i.e. the R_k for $k \in \Omega(\log n)$, in this circuit a la Copppersmith we can approximate this map to within an arbitrary inverse polynomial. The resulting circuit, which we denote AQFS, has size $O(n \log n)$ and depth $O(\log n)$.

4.3 Copying a Fourier Basis State

As was pointed out in [?], the map

$$|\hat{j}\rangle|0\rangle \longrightarrow |\hat{j}\rangle|\hat{j}\rangle, \tag{4.9}$$

can easily be accomplished exactly in size and depth $O(n)$ and $O(\log n)$. First note that applying n Hadamard gates in parallel to the second register accomplishes the transformation

$$|\hat{j}\rangle|0\rangle \longrightarrow |\hat{j}\rangle|\hat{0}\rangle. \tag{4.10}$$

Simply subtracting the second register from the first $\pmod{2^n}$ accomplishes the map

$$|\hat{j}\rangle|\hat{k}\rangle \longrightarrow |\hat{j}\rangle|\widehat{j+k}\rangle \tag{4.11}$$

since

$$\begin{aligned}
|\hat{j}\rangle|\hat{k}\rangle &= \left(\sum_{i<2^n} \omega_{2^n}^{ij}|i\rangle\right)\left(\sum_{i'<2^n} \omega_{2^n}^{i'k}|i'\rangle\right) && (4.12)\\
&= \sum_{i,i'<2^n} \omega_{2^n}^{ij+i'k}|i\rangle|i'\rangle && (4.13)\\
&\longrightarrow \sum_{i,i'<2^n} \omega_{2^n}^{ij+i'k}|i-i'\rangle|i'\rangle && (4.14)\\
&= \sum_{i,i'<2^n} \omega_{2^n}^{(i-i')j+i'(j+k)}|i-i'\rangle|i'\rangle && (4.15)\\
&= \left(\sum_{i<2^n} \omega_{2^n}^{ij}|i\rangle\right)\left(\sum_{i'<2^n} \omega_{2^n}^{i'(j+k)}|i'\rangle\right) && (4.16)\\
&= |\hat{j}\rangle|\widehat{j+k}\rangle. && (4.17)
\end{aligned}$$

This subtraction can be performed in size and depth $O(n)$ and $O(\log n)$ respectively as discussed in Section 1.5.1.

4.4 Putting it all Together

We now present two circuits. Circuit 4.3 is an approximate parallel QFT modulo 2^n which works with high accuracy whenever the input superposition is sufficiently uniform (in the norms of the amplitudes) over 2^n. Circuit 4.4, which calls Circuit 4.3 as a subroutine, is an approximate parallel QFT modulo 2^n which achieves arbitrary inverse polynomial precision for all input superpositions.

Assume temporarily that all QFS maps are performed exactly. If the FPE circuit called as a subroutine in Circuit 4.3 uses QFT's of size $2k$, and thus has depth $O(k)$, then the size of the squared error of Circuit 4.3 on input $|\alpha\rangle$ is bounded by

$$\frac{n^2}{2^k} + \sum_{j \in B} |\alpha_j|^2$$

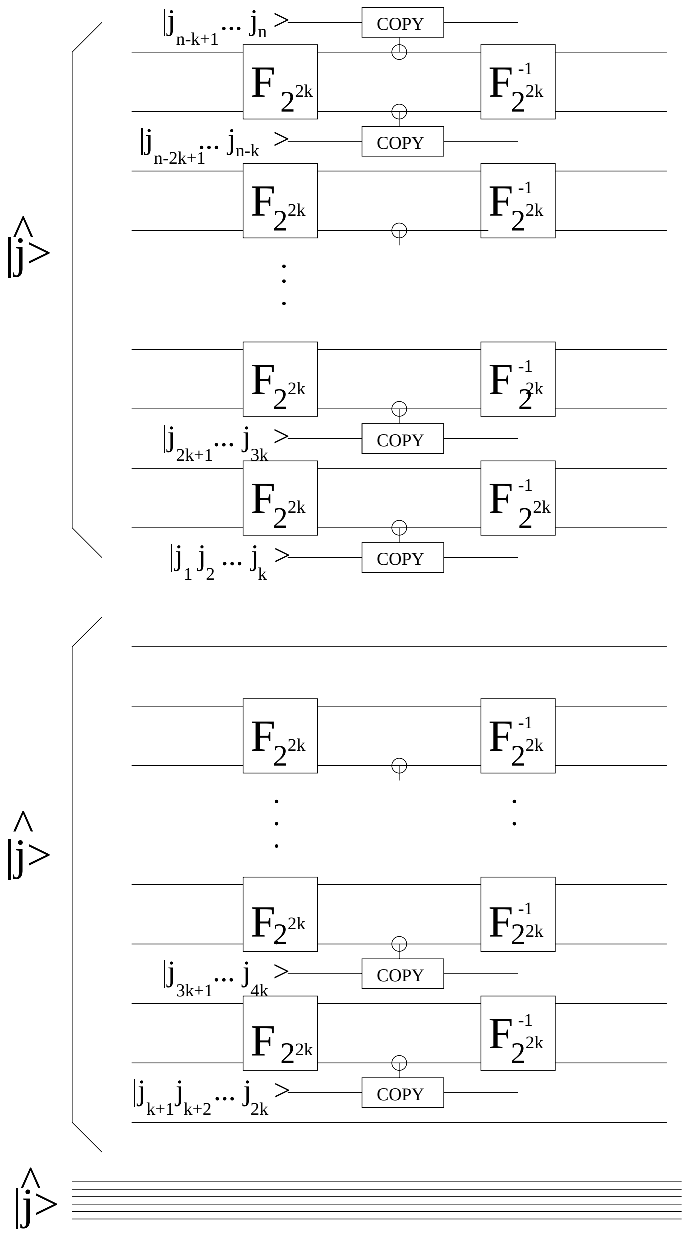
where B is the subset of indices of size $\frac{n}{2^{k/2}}$ defined in Section 4.1.

The size of the squared error of Circuit 4.4 is then bounded by

$$\frac{1}{2^n} \sum_{k<2^n} \left(\frac{n^2}{2^k} + \sum_{j+k \in B} |\alpha_j|^2 \right) = \frac{n^2}{2^k} + \frac{|B|}{2^n} = \frac{n^2}{2^k} + \frac{n}{2^{k/2}}$$

and it suffices to choose $k \in O(\log n)$ to obtain inverse polynomial accuracy.

Finally, if we perform the QFS maps in depth $O(\log n)$ the resulting inverse polynomial error simply adds to the error already analyzed and we get an overall circuit of size and depth $O(n \log n)$ and $O(\log n)$ respectively which approximates the QFT to within an arbitrary inverse polynomial.



$|j_{n-k+1} \dots j_n\rangle$
COPY
$F_{2^{2k}}$
$F^{-1}_{2^{2k}}$
$|j_{n-2k+1} \dots j_{n-k}\rangle$
$|\hat{j}\rangle$
$|j_{2k+1} \dots j_{3k}\rangle$
$|j_1 j_2 \dots j_k\rangle$
$|j_{3k+1} \dots j_{4k}\rangle$
$|j_{k+1} j_{k+2} \dots j_{2k}\rangle$

Figure 4.1: Quantum Fourier Phase Estimation (FPE): $|j\rangle|\hat{j}\rangle^3 \longrightarrow |0\rangle|\hat{j}\rangle^3$.

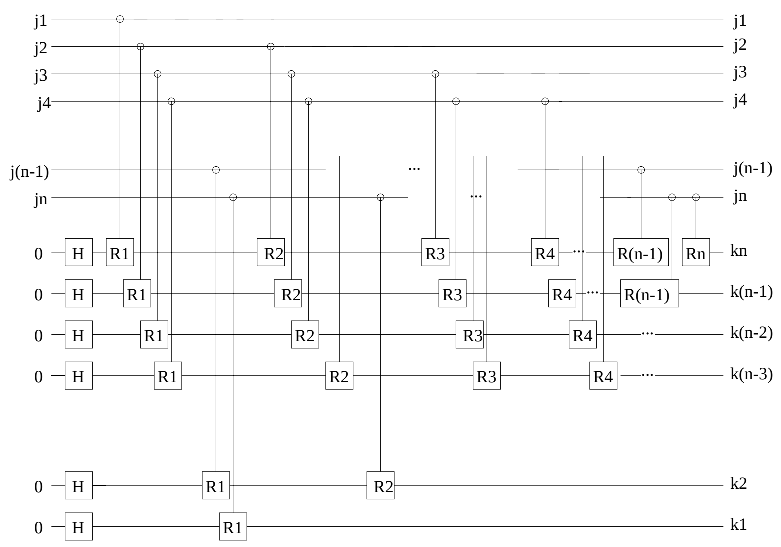


Figure 4.2: Exact Quantum Fourier State Computation (QFS): $|j\rangle|0\rangle \longrightarrow |j\rangle|\hat{j}\rangle$. The approximate version (AQFS) just omits the R_k for $k \in \Omega(\log n)$.

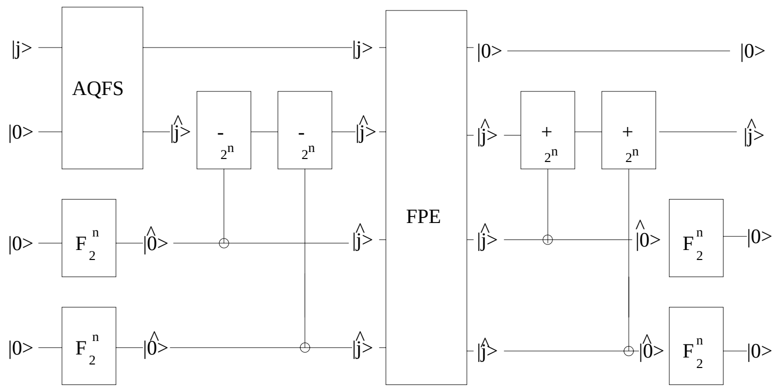


Figure 4.3: Approximate Parallel QFT for Uniform Inputs (UQFT)

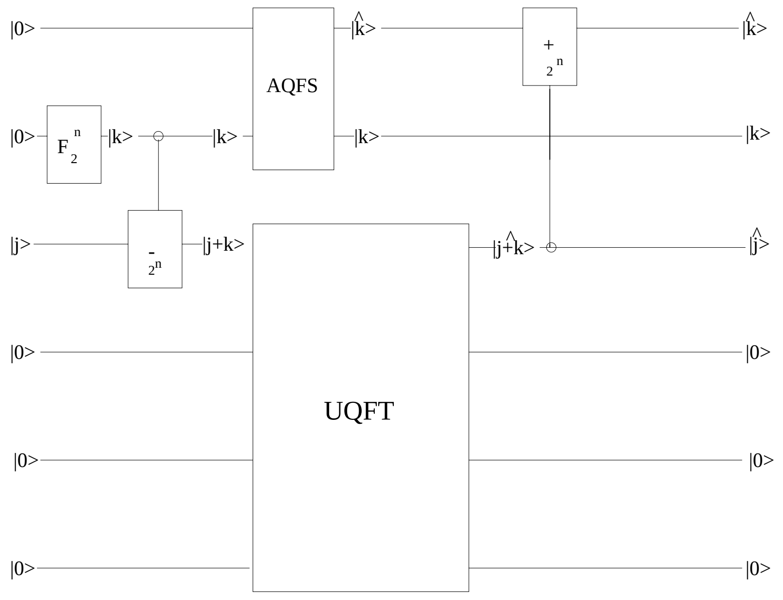


Figure 4.4: Approximate Parallel QFT

Chapter 5

An Approximate Quantum Fourier Transform over an Arbitrary Z_N

Let $|\alpha\rangle = \sum_{i<N} \alpha_i |i\rangle$ be an arbitrary quantum superposition and let $|\hat{\alpha}\rangle$ denote the quantum Fourier transform of $|\alpha\rangle$ over Z_N. We give quantum circuits which approximate this QFT to within an arbitrary ϵ. When ϵ is an inverse polynomial in $n = \log N$ the circuits achieve a substantial speedup over the $O(n^2)$ method of [?]. Our method continues to work for smaller ϵ but with asymptotic size identical to earlier methods. A preliminary version of these results can be found in [?].

We focus on the relevant situation of ϵ an inverse polynomial. In particular, we show that in this case circuits of size $O(n \log n \log\log n)$ and depth $O(\log n)$ can be achieved.

Theorem 2. *There are quantum circuits of size $O(n \log n \log\log n)$ and depth $O(\log n)$ which approximate the QFT over Z_N to within an arbitrary inverse polynomial.*

More specifically, the bottleneck in our algorithm is multiplication by our modulus

N and an n-bit approximation to its inverse $1/N$, denoted $\widetilde{1/N}$. Let $s(M)$ and $d(M)$ denote the simultaneous size and depth of quantum circuits which multiply an arbitrary n-bit integer by M, that is, map

$$|M\rangle|j\rangle \longrightarrow |M\rangle|j\rangle|Mj\rangle.^{1}$$

Then our Algorithm 3 yields the following:

Theorem 3. *There are quantum circuits of simultaneous size and depth*

$$O\left(s(N) + s(\widetilde{1/N}) + n\log n\right) \text{ and } O\left(d(N) + d(\widetilde{1/N}) + \log n\right)$$

respectively which approximate the QFT over Z_N to within an arbitrary inverse polynomial.

5.1 The Algorithm

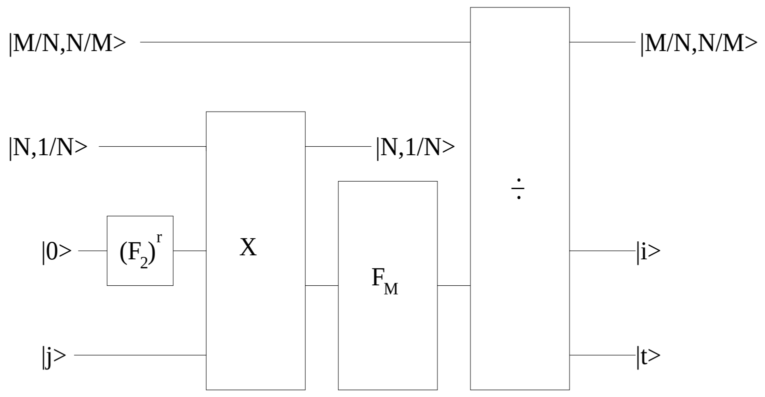


Figure 5.1: Approximate QFT over Z_N.

We now describe the action of the circuit pictured in Figure 5.1 on input $|\alpha\rangle = \sum_{i<N} \alpha_i |i\rangle$ and parameters $R = 2^r$ and $M \geq RN$. We will require a supply of $\lfloor \log(M/N) \rfloor +$

[1]Notice that both the inputs M and j are preserved and thus quantum circuits for this map can be generated from classical multiplication circuits using Lemma 1 – no division circuits are necessary.

1 clean bits in an auxiliary register and $O(n)$-bit approximations to the decimals $1/N$, N/M, and M/N. In particular, our input will be of the form

$$|C\rangle|0\rangle|\alpha\rangle$$

where $|C\rangle$ is a control register containing these approximation and a copy of our modulus N. Our output will be a superposition which is close to

$$|C\rangle|\hat{\alpha}\rangle|\eta\rangle$$

for some $|\eta\rangle$, where $|\hat{\alpha}\rangle$ denotes the QFT over Z_N of $|\alpha\rangle$.

Algorithm 3. *Input:* $|C\rangle|0\rangle|\alpha\rangle$

1. *QFT over* $(Z_2)^r$*:*

$$|0\rangle \longrightarrow \sum_{i<R} \frac{1}{\sqrt{R}}|i\rangle.$$

2. *Repeat* $|\alpha\rangle$ *R-times:*

$$\begin{aligned} |N,1/N\rangle \sum_{i<R} \frac{1}{\sqrt{R}}|i\rangle|\alpha\rangle &= |N,1/N\rangle \sum_{i<R} \frac{1}{\sqrt{R}}|i\rangle \sum_{j<N} \alpha_j|j\rangle \\ &\longrightarrow |N,1/N\rangle \sum_{j<N,i<R} \frac{1}{\sqrt{R}}\alpha_j|j+iN\rangle \\ &\overset{def}{=} |N,1/N\rangle|\beta\rangle. \end{aligned}$$

3. *QFT over* Z_M*,*

$$|\beta\rangle \longrightarrow |\hat{\beta}^M\rangle$$

4. *Division by* M/N*:*

$$|j\rangle \longrightarrow |i\rangle|t\rangle$$

where $j = \lfloor \frac{M}{N} i \rceil + t$ *and* $-\frac{M}{2N} \leq t < \frac{M}{2N}$.

The correctness of the algorithm is a consequence of the fact that for a typical remainder t, the subvector indexed by integers of the form $\lfloor \frac{M}{N} i \rceil + t$ (and renormalized to unit length) is close to the desired $|\hat{\alpha}\rangle$. It is worth noting that if $M = RN$ this is exactly true – the only remainder with any amplitude is zero and the subvector at integers $\frac{M}{N} i$ is exactly $|\hat{\alpha}\rangle$. More generally the approximation follows from Theorem 10, which yields the following Corollary.

Corollary 1. *Let* $|C\rangle|\gamma\rangle$ *be the output of the above algorithm. Then there is a superposition* $|\eta\rangle$ *so that*

$$\| |\gamma\rangle - |\hat{\alpha}\rangle|\eta\rangle \| < \frac{4RN}{M} + \frac{8 \log N}{\sqrt{R}}.$$

In order to achieve a QFT which is accurate to within ϵ, then, it suffices to take $R = \Omega\left(\frac{\log^2 N}{\epsilon^2}\right)$ and $M = \Omega\left(\frac{RN}{\epsilon}\right)$.

5.1.1 Size and Depth Analysis

We now return to Circuit 5.1 and analyse its size and depth requirements, restricting our analysis to the situation where ϵ is an inverse polynomial. First, since we are free to choose M to be a power of 2, the QFT at step 3 can be implemented by the parallel circuits of Chapter 4. Also note that the bounds from Corollary 1 show that we can take M to have only $n + O(\log n)$ bits. The subcircuit for this transform thus has size and depth $O(n \log n)$ and $O(\log n)$ respectively with constants approaching those of the parallel circuits over 2^n itself.

We now turn to the two multiplication procedures sandwiching the transform. Step 2 involves the multiplication of integers less than R by our n-bit modulus N and its inverse $1/N$. But R can be chosen to have only $\log n$-bits. In this case classical "grade school" multiplication techniques combined with carry-save adders are more efficient than FFT related techniques and their translation to the quantum setting yields circuits of size and depth $O(n \log n)$ and $O(\log n)$ respectively. The fact that R can be taken to have only $\log n$ bits is a consequence of the circulant analysis of Section 9.2.4.

The bottleneck in Algorithm 3 is the final step where we divide by the n-bit approximation by M/N. In other words we run Circuit 1.7 on inputs $M/N, N/M$ in reverse. Since we choose M to be a power of 2 this is equivalent in complexity to multiplication by N and $1/N$. We note that if there is a special technique for quickly multiplying and dividing by our modulus N the circuit size and depth can be improved. For instance, if $N = c^m$ is a constant power then we can perform reversible multiplication with a circuit of size $O(n \log n)$ and depth $O(\log n)$. This gives us an algorithm which matches the asymptotic size and depth of the QFT over a power of 2. Of course, a circuit for the QFT over a modulus of this form could also be constructed directly in analogy with the power of 2 case and would achieve similar asymptotic size and depth. We conjecture that by just changing the multiplication technique used in Step 4 to suit the particular modulus N our circuits can always be made asymptotically optimal.

We emphasize that the only reversible multiplication is by the modulus and its inverse, not between arbitrary n-bit integers. The inverse can thus be prepared classically and we can make use of Circuit 1.7. This allows us to avoid the problem of optimizing the simultaneous size and depth of division circuits – see the discussion at the end of Section 1.5.1. This is another clear benefit of our technique over earlier approaches. We note that there appears to be a close relationship between the complexity of approximating the QFT over Z_N and reversible multiplication by N. Our algorithm shows that, with low ($O(n \log n)$) overhead, approximate circuits for reversible multiplication by N lead to circuits for approximating the QFT over Z_N . On the other hand one can show that, with similar overhead, circuits approximating the QFT over Z_{N2^n} can be converted to approximate circuits for reversible multiplication by N. Unfortunately this relationship does not lead to a faster-than-classical quantum multiplication algorithm, the "tantalizing" question posed by Shor[?].

5.2 Fourier Sampling

In many quantum algorithms (see [?, ?, ?, ?]), including the Hidden Subgroup Algorithm 2, the QFT occurs as the final quantum step and a measurement of the superposition immediately follows. We refer to this procedure as Fourier sampling [?]. Suppose that we wish to sample from $\mathcal{D}_{|\hat{\alpha}\rangle}$, the distribution induced by measuring $|\hat{\alpha}\rangle = F_N|\alpha\rangle$, for some given N and $|\alpha\rangle$. In this situation since we need only insure that the distribution we sample from is ϵ-close to $\mathcal{D}_{|\hat{\alpha}\rangle}$ – we need not worry about the phases of the amplitudes in the final superposition. This simplifies the computation of the previous Section in two ways.

First, we can reduce the size of the QFT, F_M, which appears as a subroutine in our circuit. In particular we can choose M to be any integer at least RN as opposed to requiring $M = \Omega\left(\frac{RN}{\epsilon}\right)$ as in the previous algorithm. This is because we are now lumping together the probabilities of all outputs of the form $j = \lfloor \frac{M}{N} i \rceil + t$ for $-\frac{M}{2N} \le t < \frac{M}{2N}$ – we are no longer concerned with the individual superpositions corresponding to a fixed remainder t or with phases of our amplitudes.

Second, and more significantly, we can reduce the asymptotic size and depth of the quantum circuits by measuring immediately after F_M and performing the final division classically. This reduces the quantum circuit size and depth to $O(n \log n)$ and $O(\log n)$ respectively.

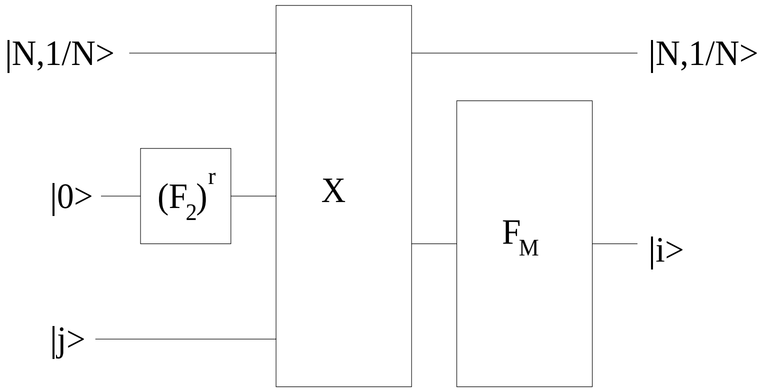


Figure 5.2: Fourier Sampling over Z_N.

Algorithm 4. *Input:* $|N, 1/N\rangle|0\rangle|\alpha\rangle$

1. *QFT over* $(Z_2)^r$*:*

$$|0\rangle|\alpha\rangle \longrightarrow \sum_{i<R} \frac{1}{\sqrt{R}}|i\rangle|\alpha\rangle.$$

2. *Repeat* $|\alpha\rangle$ *R-times:*

$$\begin{aligned} |N, 1/N\rangle \sum_{i<R} \frac{1}{\sqrt{R}} |i\rangle|\alpha\rangle &= |N, 1/N\rangle \sum_{i<R} \frac{1}{\sqrt{R}} |i\rangle \sum_{j<N} \alpha_j |j\rangle \\ &\longrightarrow |N, 1/N\rangle \sum_{j<N, i<R} \frac{1}{\sqrt{R}} \alpha_j |j + iN\rangle \\ &\overset{def}{=} |N, 1/N\rangle|\beta\rangle. \end{aligned}$$

3. *QFT over Z_M,*

$$|\beta\rangle \longrightarrow |\hat{\beta}^M\rangle$$

4. *Measure*

5. *(Classical) Divide $|j\rangle$ by M/N to output i such that $j = \lfloor \frac{M}{N} i \rceil + t$ for $-\frac{M}{2N} \leq t < \frac{M}{2N}$.*

Let $\mathcal{D}_{|\hat{\alpha}\rangle}$ be the distribution on $\{0, ..., (N-1)\}$ induced by measuring $|\hat{\alpha}\rangle$ and let $\mathcal{D}$ be the distribution induced by Algorithm 4. Corollary 1 from the previous section could be used to prove that these distributions are close for sufficiently large R and $M >> RN$. However, we use Theorem 11 to show that this is still true for any $M \geq RN$. In particular, the following Corollary is a direct application of this Theorem.

Corollary 2.

$$\|\mathcal{D}_{|\hat{\alpha}\rangle} - \mathcal{D}\|_1 < \frac{8 \log N}{\sqrt{R}}.$$

5.3 Fourier Sampling and The Hidden Subgroup Problem over Z

In the previous section we gave a procedure which, given N and $|\alpha\rangle = \sum_{i<N} \alpha_i |i\rangle$ as input, sampled from the $\mathcal{D}_{|\hat{\alpha}\rangle}$, the distribution induced by measuring $|\hat{\alpha}\rangle = F_N|\alpha\rangle$. This

procedure is the basis for the finite Abelian hidden subgroup algorithm, of which Shor's Discrete Log algorithm is a special case. We now give a procedure for the hidden subgroup problem over Z. The procedure itself is essentially identical to the quantum portion of Shor's algorithm, but we give a more general analysis.

Suppose we have the ability to generate arbitrarily long repetitions of some fixed superposition $|\alpha\rangle = \sum_{i<N} \alpha_i |i\rangle$, that is, superpositions of the form $\sum_{i<M} \alpha_{(i \mod N)} |i\rangle$, but that $|\alpha\rangle$ and N are themselves unknown. Let $\mathcal{D}_{|\hat{\alpha}\rangle/N}$ be the distribution on fractions with denominator N and numerators distributed according to $\mathcal{D}_{|\hat{\alpha}\rangle}$. The following algorithm allows us to sample from a distribution which is arbitrarily close to $\mathcal{D}_{|\hat{\alpha}\rangle/N}$.

Algorithm 5. *Input:* $\sum_{i<M} \alpha_{(i \mod N)} |i\rangle$

1. *QFT over* Z_M
2. *Measure*
3. *(Classical)Divide the result by* M*, and use the continued fractions method to round to the nearest fraction with denominator less than* T*, where* T *is a known upper bound on* N*.*

In particular, if $\mathcal{D}$ is the distribution on fractions with denominator less than T output by our algorithm, then we have the following Lemma, whose proof appears in Section 5.3.1.

Lemma 3.

$$\|\mathcal{D}_{|\hat{\alpha}\rangle/N} - \mathcal{D}\|_1 = O\left(\frac{T \log T}{\sqrt{M}}\right).$$

To make these distributions ϵ-close, then, it suffices to take $M = \Omega\left(\frac{T^2 \log^2 T}{\epsilon^2}\right)$. It is important to notice that even sampling **exactly** from $\mathcal{D}_{|\hat{\alpha}\rangle/N}$ does not immediately give us access to the distribution $\mathcal{D}_{|\hat{\alpha}\rangle}$ or to the value N because the fractions obtained are in reduced form.

We now use this algorithm to solve the Hidden Subgroup problem over Z. Recall that we are given a function f defined on G which is both constant and distinct on the cosets of an unknown subgroup $H \leq G$. The goal is to determine H. In the case of $G = Z$, H must be a cyclic subgroup generated by some element N of Z, i.e. $H = \langle N \rangle$. The function f can be equivalently described as a function with period N which is one-to-one within each period. Determining the subgroup H is equivalent to determining this period.

Given an upper bound T on N we can easily create the superposition $\sum_{i<M} |i\rangle|f(i)\rangle$ where $M = \Omega\left(T^2 \log^2 T \log^2 \log T\right)$ is a power of 2. Using this as input to the above algorithm we can sample from a distribution very close to $\mathcal{D}_{|\hat{\alpha}\rangle/N}$ where $|\alpha\rangle = \sum_{i<N} |i\rangle|f(i)\rangle$. Now, $\mathcal{D}_{|\hat{\alpha}\rangle}$ is uniform on $\{0, \ldots, N-1\}$ – this follows easily from the fact that f is one-to-one within its period – and thus $\mathcal{D}_{|\hat{\alpha}\rangle/N}$ is uniform on fractions with denominator N.

We test the denominator of each fraction output by our procedure to see if it is the period by evaluating the function at a pair of values. This will allow us to correctly discard all denominators less than N. We then accept the smallest value which passes our test. This procedure correctly recovers N as long as it actually appeared as a denominator, i.e. as long as we sampled a fraction with denominator N and numerator relatively prime to N. Such numerators constitute a $c/\log n$ fraction of the set $\{0, \ldots, N-1\}$, for some constant c. By our choice of M this set must constitute a similarly sized fraction of the distribution output by our algorithm and by sampling $O(n \log n)$ times such a numerator will occur with exponentially high probability..

5.3.1 Proof of Lemma 3

We can analyse the distribution output by Algorithm 5 on input $\sum_{i<M} \alpha_{(i \bmod N)} |i\rangle$ and parameter $T > N$ by comparison with the action of Algorithm 4 on a specially constructed input. In particular, let $|\tilde{\alpha}\rangle$ be the superposition $|\alpha\rangle$ repeated T-times, i.e.

$$|\tilde{\alpha}\rangle = \sum_{j<T} \sum_{i<N} \alpha_i |i + jN\rangle.$$

We look at the distribution output by Algorithm 4 on input

$$|TN\rangle|0\rangle|\tilde{\alpha}\rangle$$

and parameters $R = \lfloor \frac{M}{TN} \rfloor$ and M. By Corollary 2 this distribution is close to $\mathcal{D}_{F_{TN}|\tilde{\alpha}\rangle}$ for an appropriate choice of M. Moreover, since the amplitude of $F_{TN}|\tilde{\alpha}\rangle$ at Ti is identical to the amplitude of $|\hat{\alpha}\rangle = F_N|\alpha\rangle$ at i, $\mathcal{D}_{F_{TN}|\tilde{\alpha}\rangle}$ is just $\mathcal{D}_{|\hat{\alpha}\rangle}$ distributed over multiples Ti of T.

Now, it is not hard to see that the input to the QFT over Z_M when Algorithm 4 is run on input $|TN\rangle|0\rangle|\tilde{\alpha}\rangle$ and the above parameters is very close to the input to the QFT over Z_M in Algorithm 5. Thus the distribution of outputs from Step 4 of Algorithm 4 on

$$|TN\rangle|0\rangle|\tilde{\alpha}\rangle$$

is exponentially close to the distribution of outputs from Step 2 of Algorithm 5. By the previous paragraph we need merely insure that an output interpreted by the former algorithm as Ti is interpreted in the latter case as i/N. An output k which the former algorithm rounded to Ti must have satisfied

$$\left| k - \frac{M}{TN} Ti \right| < \frac{M}{2TN}. \tag{5.1}$$

Algorithm 5 divides this same k by M and uses the continued fraction method to round to the nearest fraction with denominator less than T. Dividing Equation 5.1 by M yields

$$\left| \frac{k}{M} - \frac{i}{N} \right| < \frac{1}{2TN}, \tag{5.2}$$

which implies that $\frac{i}{N}$ must be this nearest fraction given by the continued fractions procedure, as desired.

This distance between $\mathcal{D}$ and $\mathcal{D}_{|\hat{\alpha}\rangle/N}$ is thus given by

$$\frac{8 \log N}{\sqrt{M/TN}} + \frac{TN}{M - TN} = O\left(\frac{T \log T}{\sqrt{M}} \right)$$

where the first term in the sum comes from the error in Algorithm 4 and the second from the distance between the inputs to the QFT's in Algorithms 4 and 5 respectively.

Chapter 6

A Relaxation of the Abelian Hidden Subgroup Problem

Recall the hidden subgroup problem introduced in Section 2.2. We are given oracle access to a function f_H defined on a group G and constant on cosets of some unknown subgroup $H \leq G$. The challenge is to find a set of generators of H. The standard hidden subgroup problem assumes further that f_H is distinct on distinct cosets of H. This standard version can be solved efficiently on a quantum computer, that is in time polynomial in $n = \log |G/H|$, whenever G is a finitely generated Abelian group([**?**]).

The related problem which relaxes the requirement that f_H be distinct on distinct cosets of H was first addressed in [**?**] and later in [**?**]. We shall refer to this as the relaxed hidden subgroup problem. Both [**?**] and [**?**] give algorithms which partially solve the relaxed hidden subgroup problem, with the former addressing just the case $G = Z$ and the latter the general problem for finitely generated Abelian G. However, in both results the coset distinctness requirement is changed only slightly. In particular, the function f_H is allowed to map m cosets to one, but m must be both polynomial in n and smaller than the smallest prime divisor of $|G|$. Notice that for some groups, such as $G = \bigoplus_n Z_2$, this amounts to no relaxation at all.

As noted in [?], however, *some* restriction must be placed on the behavior of the function f_H, since, once the distinctness requirement is dropped, there are functions f_H and f_K for $K \neq H$ which differ on an exponentially small fraction of their inputs. Using existing lower bound techniques based on the unitary evolution of quantum computation we should expect that such functions require exponentially many queries, and thus exponential-time, to distinguish.

We solve the relaxed hidden subgroup problem for finitely generated Abelian groups. In particular, we define a stratification of the functions f_H into classes, then give a tight characterization of which classes have polynomial-time algorithms by exhibiting both an algorithm and a lower bound for each class. These results generalize and simplify our earlier work on many-to-one periodic functions presented in [?], and use in a crucial way the Fourier sampling procedures of Sections 5.2 and 5.3.

6.1 Definitions and Main Theorems

Throughout our discussion f_H will denote a function defined on a finitely generated Abelian group G and constant on cosets of the subgroup $H \leq G$. Notice that for any $K \leq H$ f_H induces a well-defined function on G/K which we denote $f_{H/K}$. We define the input length of the hidden subgroup problem given by the function f_H on G to be $n = \lceil \log |G/H| \rceil$.

and assume without loss of generality that the range of f_H is contained in the set $\{0,1\}^n$.

Let $\mathbf{D}$ be the generalized Hamming metric on the functions f_H:

Definition 3.

$$\mathbf{D}\left(f_H, f_K\right)$$

is the fraction of elements in the group G for which $f_H(x) \neq f_K(x)$.

In other words, f_H and f_K are ϵ-close under $\mathbf{D}$ if they disagree on at most an ϵ fraction of the elements of G[1]

Using this metric we stratify our functions into classes in the following manner:

Definition 4. *For any function $d(n)$ let*

$$C_{1/d(n)} = \{f_H | \forall f_K \text{ with } K \not\leq H,\ \mathbf{D}(f_H, f_K) > 1/d(n)\}.$$

We think of the function f_H as a codeword for the subgroup H. The class $C_{1/d(n)}$ is then a code with minimum distance $1/d(n)$. Our results show that there is an efficient quantum decoding procedure for $C_{1/d(n)}$ if and only if $d(n)$ is a polynomial.

More formally we will prove the following two theorems:

Theorem 4. *Given any polynomial $d(n)$ there is an efficient quantum algorithm $\boldsymbol{A}$[2] which, given any finitely generated Abelian G and $f_H \in C_{1/d(n)}$, outputs the generators of H with exponentially high probability.*

[1]A careful reader should object at this point that this definition only really makes sense when G is a finite group. We take the distance between f_H and f_K defined on an infinite G to be the distance between the induced functions $f_{H/(H\cap K)}$ and $f_{K/(H\cap K)}$ which are defined on the finite group $G/(H\cap K)$.

[2]Throughout the paper we will assume that $\boldsymbol{A}$ has a blackbox subroutine for computing values of f

Theorem 5. *Let $d(n) = o(2^n)$ be given. Suppose that $\boldsymbol{A}$ is a quantum algorithm which correctly computes generators of H from any $f_H \in C_{1/d(n)}$ with probability at least $3/4$. Then $\boldsymbol{A}$ has worst-case run-time $\Omega(\sqrt[4]{d(n)})$.*

Our algorithm uses the same quantum subroutine as the standard hidden subgroup problem – namely Fourier Sampling. The relaxed problem requires more repetitions of this quantum subroutine and, in the case of $G = Z$, a more elaborate classical post-processing.

The lower bound is proved in the special case where $G = Z$ and thus the hidden subgroup function is periodic on Z and potentially many-to-one within each period. The proof is a slight variation – allowing for the periodic structure of f – on the standard lower bound technique of [**?**]. This is sufficient to establish the polynomial vs. superpolynomial gap which is our primary concern. It is likely that the more sophisticated techniques of [**?**] could be used to improve this lower bound.

6.2 Finite Abelian G

We first solve the special case of the relaxed hidden subgroup problem where the underlying group G is finite Abelian. Section 6.4 addresses the case $G = Z$ and shows how to combine these to give an algorithm which works for any finitely generated Abelian group.

As in our discussion of the standard hidden subgroup problem we assume that the group G is given to us as a direct product $\bigoplus_{i<n} Z_{p_i}$. More concretely, the input to our quantum algorithm is the list $(p_1, \ldots, p_k)$ and our function f_H is defined on the **set** $\bigoplus_{i<n} Z_{p_i}$. As in the standard hidden subgroup problem there is a quantum procedure which produces such a description under very general conditions [**?**]

Algorithm 6. *1. Prepare*

$$|\alpha\rangle = \sum_{x \in G} |x\rangle |f_H(x)\rangle.$$

2. Sample from $\mathcal{D}_{F_G|\alpha\rangle}$ where

$$F_G|\alpha\rangle = \sum_{x \in G} F_G|x\rangle |f_H(x)\rangle.$$

Repeat this procedure $\Omega\left(n^2 d^2(n)\right)$ times obtaining outputs y_i. Solve the corresponding system of equations $y_i \cdot_G x$ and output this solution.

As mentioned previously, our quantum subroutine is identical to that of the standard case but we must increase the number of samples by a factor of $d^2(n)$. As before, the correctness of this algorithm is equivalent to the condition that the samples y_i generate the subgroup $H^\perp$. We first note that, as in the standard case, the distribution $\mathcal{D}_{F_G|\alpha\rangle}$ is supported on this subgroup. The argument given in the standard case (see Section 2.3) hinges on the fact that for any $h \in H$

$$|\alpha\rangle = \sum_{x \in G} |x\rangle |f_H(x)\rangle$$

and

$$|h * \alpha\rangle = |h\rangle * |\alpha\rangle = \frac{1}{\sqrt{|G|}} \sum_{x \in G} |h + x\rangle |f_H(x)\rangle$$

are identical, which remains true in the relaxed problem as well.

In order to establish the correctness of the algorithm we need further that the outputs generate $H^\perp$ with high probability. Recall that in the standard case we used the fact that the distribution was uniform on $H^\perp$ to argue that $O(n^2)$ samples must generate this subgroup with high probability. Uniformity no longer holds in the relaxed case. Instead we substitute the following property which limits the probability that our samples remain trapped in some proper subgroup of $H^\perp$:

Lemma 4. *Suppose that $f_H \in C_{1/d(n)}$. Then for every proper subgroup $K < H^\perp$, if y is chosen according to $\mathcal{D}_{F_G|\alpha\rangle}$*

$$Pr\left(y \in K\right) < 1 - 1/4d^2(n).$$

This lemma, proved in Section 6.2.1, is the main technical result of this Chapter. The correctness of the algorithm follows from the Lemma by an argument similar to that of the standard case. In particular, in order for our outputs to generate $H^\perp$ they must lie outside of any proper subgroup K of $H^\perp$. There are at most 2^{n^2} such subgroups, since each is determined by a set of at most n generators and

$$\left|H^\perp\right| = |G/H| < 2^n.$$

The probability that there exists a proper subgroup of $H^\perp$ containing all our outputs is therefore upper bounded by the quantity

$$2^{n^2}\left(1 - 1/4d^2(n)\right)^t$$

where t is the number of repetitions of the quantum subroutine. Thus if we choose $t = \Omega\left(n^2 d^2(n)\right)$ the outputs will generate $H^\perp$ with high probability.

6.2.1 Proof of the Reconstruction Lemma

We prove a reformulation of Lemma 4 which replaces the quantification over subgroups of $H^\perp$ with the quantification over subgroups of $H^\perp$ which are themselves "perps".

Since $H = \left(H^{\perp}\right)^{\perp}$ all subgroups are of this form and the content of the lemma is unchanged. We first sketch how to establish that $H = \left(H^{\perp}\right)^{\perp}$, then proceed to the proof of the reformulated lemma. Notice that it follows trivially form the definition of $H^{\perp}$ that $H \subseteq \left(H^{\perp}\right)^{\perp}$. Since H is finite it then suffices to show that $|H| = \left|\left(H^{\perp}\right)^{\perp}\right|$ which follows from $|G/K| = \left|K^{\perp}\right|$ for all subgroups K. This last equality can be proved by showing that the QFT over G maps a subspace of dimension $|G/K|$ to one of dimension $\left|K^{\perp}\right|$ and using the fact that the QFT is unitary.

Lemma 4. *Suppose that $f_H \in C_{1/d(n)}$. Then for every proper subgroup $K^{\perp}$ of $H^{\perp}$, if y is chosen according to $\mathcal{D}_{F_G|\alpha\rangle}$ then*

$$Pr\left(y \in K^{\perp}\right) < 1 - 1/4d^2(n).$$

Proof. We give a proof by contradiction. Suppose there exists a $K^{\perp}$ which violates the lemma. We reconstruct a function f_K with

$$\mathbf{D}\left(f_H, f_K\right) < 1/d(n),$$

This contradicts the assumption $f_H \in C_{1/d(n)}$, since if $K^{\perp}$ is a proper subgroup of $H^{\perp}$ then $K \nleq H$.

We first note that for any $g \in G$ the amplitudes of $F_G|\alpha\rangle$ $F_G|g * \alpha\rangle$ at x are related by the phase $\omega^{g \cdot_G x}$. Thus for any $k \in K$ the amplitudes of $F_G|\alpha\rangle$ and $F_G|k * \alpha\rangle$ are identical at elements of $K^{\perp}$. Moreover by our assumption, when y is chosen according to $\mathcal{D}_{F_G|\alpha\rangle}$,

$$Pr\left(y \in K^{\perp}\right) > 1 - 1/4d^2(n),$$

and thus the superpositions $F_G|\alpha\rangle$ and $F_G|k * \alpha\rangle$ are heavily supported on this subgroup

$K^{\perp}$. The superpositions must therefore be close. In particular,

$$\big\langle F_G|\alpha\rangle, F_G|k * \alpha\rangle\big\rangle > \sqrt{1 - 1/4d^2(n)}.$$

This implies the same lower bound for the inner product

$$\big\langle|\alpha\rangle, |k * \alpha\rangle\big\rangle,$$

indicating that the vectors $|\alpha\rangle$and $|k * \alpha\rangle$ also have almost the same direction. This can only be the case if they agree on most of their coordinates. In particular, if c is the fraction of x for which $f_H(x) = f_H(x +_G k)$, then

$$\big\langle|\alpha\rangle, |k * \alpha\rangle\big\rangle = \sqrt{c} > \sqrt{1 - 1/4d^2(n)}.$$

In other words, for every $k \in K$ at least a $1 - 1/4d^2(n)$ fraction of the $x \in G$ satisfy

$$f_H(x) = f_H(x + k). \tag{6.1}$$

We now define our new function f_K which is constant on cosets of K but still close to f_H. For each coset xK we define f_K to be uniformly equal to the majority value of f_H on xK, if one exists, and uniformly equal to 0 otherwise. Clearly f_K is constant on cosets of K but it remains to show that

$$\mathbf{D}(f_H, f_K) < 1/d(n)$$

in order to obtain a contradiction. But by (6.1) together with a standard averaging argument we have that for at least a $1 - 1/2d(n)$ fraction of the cosets xK, f_H is constant on a $1 - 1/2d(n)$ fraction of the coset. This implies that

$$\mathbf{D}(f_H, f_K) < 1/2d(n) + 1/2d(n) = 1/d(n),$$

as desired. □

6.3 The Relaxed Hidden Subgroup Problem over Z

We now give an algorithm for the relaxed hidden subgroup problem over $G = Z$. Let f_H be defined on Z and constant on cosets of $H \leq Z$. In this case H must be generated by some $N \in Z$ and we refer to f_H as $f_{\langle N \rangle}$. $f_{\langle N \rangle}$ is equivalently a periodic function with period N. In this relaxed problem $f_{\langle N \rangle}$ may not be distinct on distinct cosets, in other words the function is potentially many-to-one within each period. The distinctness requirement is replaced with the assumption that $f_{\langle N \rangle} \in C_{1/d(n)}$ for some polynomial $d(n)$.

Let

$$|\alpha\rangle = \sum_{i<N} |x\rangle |f_{\langle N \rangle}(x)\rangle$$

and $|\hat{\alpha}\rangle = F_N|\alpha\rangle$ be the superposition obtained by performing the QFT over Z_N of the first register of $|\alpha\rangle$. We first note that the restriction of $f_{\langle N \rangle}$ to the set $\{0, 1, \ldots, N-1\}$ is the function induced by $f_{\langle N \rangle}$ on $Z/\langle N \rangle$. It is easy to see that this induced function is still in $C_{1/d(n)}$ but now encodes the trivial subgroup $\langle 0 \rangle$. By the results of Section 6.2 if we sample from $\mathcal{D}_{|\hat{\alpha}\rangle}$ $\Omega(n^2 d^2(n))$ times we will almost surely obtain a set $\{y_i\}$ generating $\langle O \rangle^{\perp} = Z_N$, that is, a set $\{y_i\}$ satisfying $gcd(y_1, \ldots, y_k, N) = 1$

While we cannot create the superposition $|\alpha\rangle$, we can create arbitrarily long repetitions of $|\alpha\rangle$ by evaluating f_H on some interval. The Fourier sampling procedure of Section 5.3. then allows us to sample from a distribution exponentially close to $\mathcal{D}_{|\hat{\alpha}\rangle/N}$, the distribution on fractions with denominator N and numerators distributed according to $\mathcal{D}_{|\hat{\alpha}\rangle}$. We can thus assume we are sampling exactly from the distribution $\mathcal{D}_{|\hat{\alpha}\rangle/N}$, and by the above paragraph after $\Omega(n^2 d^2(n))$ samples the numerators of the fractions satisfy $gcd(y_1, \ldots, y_k, N) = 1$ with exponentially high probability. By taking the least common multiple of all denominators we recover the desired N.

6.4 Finitely Generated Abelian G

The finitely generated case can be reduced to the finite case by restricting f_H to each of the infinite cyclic components of G and using the algorithm of the previous section to find the periods of these restricted functions. More formally, given a description

$$(p_1, \ldots, p_k, m)$$

of the group

$$G = \left(\bigoplus_{i<n} Z_{p_i} \right) + \left(\bigoplus_{i<m} Z \right),$$

by finding the periods N_i of the restriction of f_H to each of the m copies of Z we obtain a finite Abelian $G' = \left(\bigoplus_{i<n} Z_{p_i}\right) + \left(\bigoplus_{i<m} Z_{N_i}\right)$ so that the restriction of our function f_H to G' now encodes a subgroup $H' \leq G'$ and is still in $C_{1/d(n)}$. Moreover the generators of H are precisely the generators of H' together with the periods N_i. This accomplishes the desired reduction.

6.5 Proof of Lower Bound, Theorem 5

We need the following definition and theorem from [**?**]. Theorem 6 expresses the fact that if a quantum algorithm makes few queries to an oracle function there must be values of that function which have been hardly examined and thus can be changed without significantly changing the algorithm's behavior. Its proof combines the unitary evolution of quantum computation with a hybrid argument.

Definition 5. *[?] Let $|\phi_i\rangle$ be the superposition of A^f on input x at time i. We denote by $q_y(|\phi_i\rangle)$ the sum of squared magnitudes in $|\phi_i\rangle$ of configurations of M which are querying the oracle on string y.*

Theorem 6. *[?] Let $|\phi_i\rangle$ be the superposition of A^f on input x at time i. Let $\epsilon > 0$. Let $S \subseteq [0, T-1] \times \Sigma_*$ be a set of time-strings pairs such that $\sum_{(i,y)\in S} q_y(|\phi_i\rangle) \leq \frac{\epsilon^2}{T}$. Now suppose the answer to each query $(i, y) \in S$ is modified to some arbitrary fixed $a_{i,y}$ (these answers need not be consistent with an oracle). Let $|\phi'_i\rangle$ be the time i superposition of A on input x with oracle answers modified as stated above. Then $\| |\phi_i\rangle - |\phi'_i\rangle \| \leq \epsilon$.*

In our case we wish to use Theorem 6 to show that if a quantum algorithm computes with constant probability the period N of any $f \in C_{1/d(n)}$ defined on $G = Z$ then it must make at least $\Omega(\sqrt[4]{d(n)})$ queries to the function's values. To this end we first look at the algorithm's behavior when $f(x) = 0$ for all x (Note that the all-zeroes function is in every class $C_{1/d(n)}$).

We wish to use this behavior to generate a function $g \in C_{1/d(n)}$ which has period greater than 1 and which the algorithm cannot distinguish from the all-zeroes function without making lots of queries. This is similar to earlier applications of Theorem 6 but with the added complication that g must be periodic and at least $1/d(n)$ away from any function of smaller period. We ensure periodicity by first deciding on the period N of g and then changing the value of the function simultaneously on all points of the form $x + kN$. We show that the latter complication can be resolved by choosing g to have prime period and to be sufficiently different from the all-zeroes function.

Proof. (Proof of Theorem 5) Given A^f computing the period of any function in $C_{1/d(n)}$ in time T, we initially examine A^o where o denotes the all-zero function.

Fix a prime N such that $\sqrt{d(n)} < N < 2^n$. For $0 \leq x < N$ let

$$S_x = [0, T-1] \times \{y | y = x + kN\}.$$

The average value of $\sum_{(i,y)\in S_x} q_y(|\phi_i\rangle)$ is $\frac{T}{N}$ and thus at least 1/2 of the sets S_x satisfy

$$\sum_{(i,y)\in S_x} q_y(|\phi_i\rangle) \leq 2\frac{T}{N}. \tag{6.2}$$

Let U be any set of $3N/\sqrt{d(n)}$ x which satisfy (6.2). We let our new function g satisfy $g(x + kN) = 1$ for $x \in U$ and $g(x) = o(x) = 0$ otherwise. Note that $g(x)$ has period our chosen prime N and that $D(o, g) \geq 3/\sqrt{d(n)}$.

Furthermore, let $S_U = \bigcup_{x\in U} S_x \subseteq [0, T-1] \times \Sigma_*$. Then

$$\sum_{(i,y)\in S_U} q_y(|\phi_i\rangle) \leq \frac{6T}{\sqrt{d(n)}},$$

and we can take the ϵ of Theorem 6 to be $\frac{\sqrt{6T}}{\sqrt[4]{d(n)}}$. Thus in order for our algorithm A to distinguish between the all-zeros function o and our new period-N function g with constant probability, A must have worst-case run-time $\Omega(\sqrt[4]{d(n)})$.

To prove our theorem, however, we need to verify that our function g is actually in $C_{1/d(n)}$. We need the following claim whose simple proof is in the next section.

Claim 1. *For any periodic functions f and g with periods N_f and N_g respectively, if $D(f, g) < \epsilon^2 < 1/16$ then there is a function h with period $N_h = gcd(N_f, N_g)$ and $D(h, g) < 3\epsilon$.*

Think of the g in the claim as being our g constructed above. We need to argue that there are no functions of smaller period within $1/d(n)$ of g. By our claim if such a function

f existed then there would be a function h with period $N_h = gcd(N_f, N_g) = 1$ (since the period of g is a prime) and $D(h, g) < 3/\sqrt{d(n)}$. But g and the all-zeroes function, which is the only plausible candidate for h, have distance at least $3/\sqrt{d(n)}$, a contradiction. □

6.5.1 Proof of Claim 1

Proof. Let $N_h = gcd(N_f, N_g)$. Fix k and l such that $lN_f - kN_g = N_h$. We will define a function h which is constant on flights of the form $[x + kN_h] = (x, x + N_h, x + 2N_h, ...)$ and within 3ϵ of g. Since $D(f, g) < \epsilon^2$, with probability at least $1 - \epsilon$ when we choose a random flight $[x + kN_h]$ at least a $1 - \epsilon$ fraction of points y in that flight will satisfy $f(y) = g(y)$. For such a "good" flight, choose y and z independently at random in the flight and let j satisfy $jN_h = y - z$. Then the point $w = y + jkN_g = z + jlN_f$ is uniformly distributed over the flight. Thus $f(z) = f(w) = g(w) = g(y)$ with probability at least $1 - \epsilon$. Putting these two facts together we get that when y and z are chosen at random in a "good" flight, $g(y) = g(z)$ with probability at least $1 - 2\epsilon$. Using the fact that $\epsilon < 1/4$, this implies that at least a $1 - 2\epsilon$ fraction of points in the flight share the same g value. We let the value of h on all points in the flight be this overwhelming g value, and for "bad" flights we define h to be uniformly 0. Then it follows that $D(g, h) < 2\epsilon + \epsilon = 3\epsilon$, as claimed. □

Chapter 7

Hidden Subgroups over the Reals

We now expand the ideas of the previous section to show how to find the period of certain periodic functions defined on the reals, effectively solving the hidden cyclic subgroup problem over $< \Re, + >$. This generalizes a recent result of [?] which gives a quantum algorithm finding the period of a subclass of these periodic functions sufficient to yield a polynomial-time quantum solution to Pell's equation. Solving Pell's equation has been shown to be at least as hard as factoring but no reduction in the opposite direction exists. In Chapter 8 we give evidence in a relativized setting that period-finding over the reals is in fact harder than over the integers. In particular, we show that the problem over the reals lies outside of the complexity class MA, a complexity class which contains the analogous problem over the integers.

Throughout our discussion f will denote a piecewise continuous function from $\Re \rightarrow [0, 1]$ with period p. Our quantum machine is allowed oracle access to approximate versions of f. In particular, on call $|i\rangle|j\rangle|t\rangle|0\rangle$ the oracle xors the first t-bits of $f(i/j)$, denoted $f_t(i/j)$, into the last register, returning $|i\rangle|j\rangle|t\rangle|f_t(i/j)\rangle$.

The input length of f is (n, k) if $p < 2^n$ and the n-bit approximating step function f_n has average step interval at least $1/2^k$, where the average step interval is defined to be the ratio of the period p to the number of step intervals in that period. We define a metric on these functions which is the continuous analog of Definition 3, Section 6.1.

Definition 6. *Let $i_{f,g}(x) = 1$ whenever $|f(x) - g(x)| > 2^{-n}$ and 0 otherwise. Then*

$$\mathbf{D}(f, g) = \lim_{t\to\infty} \frac{1}{t} \int_0^t i_{f,g}(x)dx.$$

Just as in the case of functions defined on Z (Definition 4, Section 6.1) we use this metric to stratify the functions into classes. Again, if we think of f as an encoding of its period p then the class $C_{1/d(n)}$ is a code with minimum distance $1/d(n)$. If $f \in C_{1/d(n)}$ then in order to reduce its encoded period one needs to change at least a $1/d(n)$ "fraction" of its values by at least $1/2^n$. In other words, f encodes its period $1/d(n)$-unambiguously and does so using just n-bits of output.

Definition 7.

$$C_{1/d(n)} = \{f | \forall g,\ \textit{if}\ p_g < p_f\ \textit{then}\ \mathbf{D}(f, g) > 1/d(n)\}$$

We can now state the main theorem of this section:

Theorem 7. *For any polynomial $d(n)$ there is a quantum algorithm A which generates the first m-bits of the period of any $f \in C_{1/d(n)}$ with exponentially high probability in time $poly(n, k, m)$.*

That the condition $f \in C_{1/d(n)}$ for $d(n)$ a polynomial is necessary for an efficient quantum algorithm to exist follows almost immediately from the lower bound result

(Theorem 5) of Chapter 6 – after interpreting functions on Z with integral period as step functions on $\Re$ with step interval 1 in some canonical way, all that remains is to check that the respective definitions of $C_{1/d(n)}$ do in fact coincide.

7.1 Overview

Before we give a summary of the procedure we note that it is sufficient to give an algorithm in the restricted case where the given function f is a step function with n-bit range, i.e. $f = f_n$, and has average step interval ≥ 1. The first m bits of the period of an arbitrary $f \in C_{1/d(n)}$ of input length (n, k) can then be found by running this algorithm to find the first m bits of the period of the function $f_n(x/2^k) \in C_{1/d(n)} \subset C_{1/d(n+k)}$ which satisfies these restrictions and has input length $(n+k, 1)$. We will thus assume without loss of generality that our function f has n-bit range and average step interval ≥ 1.

The quantum portion of the algorithm is just Fourier Sampling, in this case sampling from the distribution induced by measuring $F_{MN}\left(\sum_{i\in\pm\frac{MN}{2}} |i\rangle|f(i/N)\rangle\right)$ for some M, N. The tricky part lies in showing that M and N can be chosen simultaneously to yield the desired information about the period. Suppose we fix M and choose $N >> M$. Then it is easy to see that evaluating the functions $f(\frac{x}{N})$ and $f(\frac{px}{\lfloor Np \rceil})$ on the interval $[\pm\frac{MN}{2}]$ results in exponentially close superpositions (Lemma 6). This is useful because the latter function has integral period $\lfloor Np \rceil$ (easy to see) and is in $C_{1/2d(n)}$ when regarded as a function on the integers (Lemma 7). This allows us to use the results of the previous chapter to analyze the distribution output by the Fourier sampling procedure.

In particular, suppose by some fortuitous luck that $\lfloor Np \rceil$ actually divides MN.

Then we know that the the Fourier sampling procedure always outputs integers of the form $k_i = \frac{j_i MN}{\lfloor Np \rceil}$ and that $gcd(j_1, \ldots, j_t, \lfloor Np \rceil) = 1$[1] is satisfied with high probability after just $O(n^2 d^2(n))$ samples. This would allow us to reconstruct $\lfloor Np \rceil$ just by taking the lcm of the denominators of the fractions $\frac{k_i}{MN}$.

Now, dropping the improbable assumption that $\lfloor Np \rceil$ divides MN, we can use Corollary 2, Section 5.2 to conclude that if $MN >> Np \log^2(Np)$ then we will sample approximations k_i to the fractions $\frac{j_i MN}{\lfloor Np \rceil}$, where the approximations satisfy

$$\left| k_i - \frac{j_i MN}{\lfloor Np \rceil} \right| < \frac{MN}{2Np} \tag{7.1}$$

and the j_i are distributed as described in the previous paragraph. The requirement $MN >> Np \log^2(Np)$ is still compatible with choosing $N >> M$, so if we could just reconstruct the fractions $\frac{j_i MN}{\lfloor Np \rceil}$ from the approximations k_i we would be done.

Unfortunately, reconstructing the $\frac{j_i}{\lfloor Np \rceil}$ from the $\frac{k_i}{MN}$ using the continued fractions method requires a tighter bound than Equation 7.1 provides – the fractions would need to be within $\frac{1}{2\lfloor Np \rceil^2}$ of each other. Previous results obtain this tighter bound by evaluating the function past the square of its period – see for example Algorithm 5, Section 5.3. This is not an option for us since it would entail choosing $MN >> (Np)^2 \log^2(Np)$, incompatible with our initial assumption of $N >> M$. We bypass this problem in the following manner. First we argue (Lemma 5) that the approximations k_i output by our procedure are actually very small in absolute value. In particular, rather than ranging out to the maximal possible $\pm\frac{MN}{2}$, with exponentially high probability they are within $\pm O(2^{2n} M)$, regardless of our choice of N. This implies that the j_i are within $\pm O(2^{3n})$. We can then use continued

[1]Actually in the end we will require, and show, that in this case the stronger condition $gcd(j_1, \ldots, j_t) = 1$ is satisfied.

fractions to round the ratio k_l/k_m of any pair of outputs of the Fourier sampling procedure to the nearest fraction with denominator less than 2^{3n} and this modified continued fractions procedure terminates correctly, yielding j_l/j_m with the correct distribution, as long as $M = \Omega(2^{11n}\log^2(MN))$. We can thus reconstruct j_1 by taking the lcm of the numerators fractions j_1/j_i for sufficiently many i. Finally, as long as $M = \Omega(2^m 2^{11n}\log^2(MN))$, the leading m bits of Mj_1/k_1 coincide with p's and we can output them as our final answer.

7.2 The Algorithm

Choose $M = \Omega\left(2^m 2^{11n}\log^2 MN\right)$ and $N = \Omega\left(2^{4n}M\right)$, both powers of two.

Algorithm 7. *Fourier Sampling over $\Re$*

1. *Generate input superposition*

$$\sum_{i\in\pm\frac{MN}{2}} |i\rangle|f(i/N)\rangle.$$

2. *Fourier Sample over Z_{MN}*

Repeat this quantum subroutine $O(n^2d^2(n))$-times. Discard any sample which is less than $\frac{M}{2^m 2^{10n}}$ and let $\{k_i\}$ denote the remaining valid samples.

- *(Classical)Use the continued fractions method to round each fraction k_1/k_i to the closest fraction with denominator less than 2^{3n}.*

- *(Classical) Let j_1 be the least common multiple of the numerators of these fractions. Output the leading m bits of Mj_1/k_1.*

It suffices to show that the outputs $\{k_i\}$ from the quantum subroutine satisfy

$$\left| k_i - \frac{j_i MN}{\lfloor Np \rceil} \right| = O\left(\frac{M}{2^m 2^{10n}} \right) \tag{7.2}$$

for integers $j_i < 2^{3n}$ satisfying $gcd(j_i) = 1$. This bound implies $|k_1/k_i - j_1/j_i|$ is at most $1/2^{6n}$ and thus the continued fractions procedure correctly delivers each fraction j_1/j_i. Since the j_i are relatively prime, j_1 will be the least common multiple of their numerators. And, finally, k_1 will be sufficiently close to $j_1 M/p$, that is, within $M/2^{m+n+1}$, to correctly deliver the first m bits of p.

We proceed to show that the statement involving Equation 7.2 is true with exponentially high probability via the following three lemmas, proved in Sections 7.2.1 and 7.2.2. The first is used to establish that the j_i are small. The second and third allow us to use previous results about functions with integral period to understand the distribution of the j_i and the quality of the approximations k_i. In each of these lemmas we assume that f has minimal, as opposed to average, step size at least 1. But it is easy to show that given any f with average step size at least 1, the function $f(x2^{-n})$ is exponentially close to a function with period $2^n p$ and *minimal* step-size 1. Thus this assumption entails only a constant factor penalty in the run-time of the algorithm.

Lemma 5. *Let f be an integral-valued step function on $\Re$ with minimal step size ≥ 1. Let $\mathcal{D}_{MN}$ be the distribution on the integers in $\pm MN/2$ induced by sampling*

$$F_{MN}\left(\sum_{i \in \pm \frac{MN}{2}} |i\rangle |f(i/N)\rangle \right).$$

Then for all k dividing N

$$Pr_{x \in \mathcal{D}_{MN}}\left(|x| > k^2 M \right) = O(1/k).$$

Lemma 6. *Let f be an integral-valued step function on $\Re$ with minimal step size ≥ 1 and period p. Then for any $t \in \Re$*

$$\frac{1}{MN}\left\|\sum_{i\in\pm\frac{MN}{2}}|i\rangle|f\left(i/N\right)\rangle - \sum_{i\in\pm\frac{MN}{2}}|i\rangle|f\left(pi/(Np+t)\right)\rangle\right\|^2 = O\left(\frac{tM}{Np}\right).$$

Lemma 7. *Let $f \in C_{1/d(n)}$ be an integral-valued step function on $\Re$ with minimal step size ≥ 1 and period $p < 2^n$. Then any rescaling of f, $f(\alpha x)$ which has integral period at least $4d(n)p$ is in $C_{1/2d(n)}$ when restricted to Z (See Definition 4 Section 6.1).*

By Lemma 5, with exponentially high probability our samples k_i are at most $2^{2n}M$, and thus the j_i are at most $2^{2n}p < 2^{3n}$. By Lemma 6 and our choice of $N = \Omega\left(2^{4n}M\right)$ we can assume that we are Fourier sampling, not our given function, but instead the function $f(pi/\lfloor Np\rceil)$ which has integral period $\lfloor Np\rceil$. By Lemma 7 this function is in $C_{1/2d(n)}$ when restricted to Z. Thus by the results of Section 6.3, after just $\Omega(n^2d^2(n))$ samples with exponentially high probability we have approximations to fractions $j_iMN/\lfloor Np\rceil$ with $gcd(j_i, \lfloor Np\rceil) = 1$. In this case we need further that $gcd(j_i) = 1$. If this was not true there would be some common divisor $d < 2^{3n}$, our bound on the j_i. Choose $r < 2^{3n}$ so that d divides $Np + r$. Then by using Lemma 6 and our choice of N a second time with the function $f(pi/Np+r)$ we get that the j_i also satisfy $gcd(j_i, Np+r) = 1$ with exponentially high probability, a contradiction.

Finally, we need to ensure that the k_i satisfy

$$\left|k_i - \frac{j_iMN}{\lfloor Np\rceil}\right| = O\left(\frac{M}{2^m 2^{10n}}\right)$$

Again we assume we are Fourier Sampling the function $f(pi/\lfloor Np\rceil)$ over Z_{MN}. If $\lfloor Np\rceil$ divided MN we would be done – the k_i would exactly equal the desired fractions. In general we can apply the Fourier sampling results from Section 5.2. By measuring $F_{MN}(\sum_{i\in\pm\frac{MN}{2}}|i\rangle|f(i)\rangle)$ and rounding to the nearest multiple of $MN/k\lfloor Np\rceil$ we approximate the distribution gotten by Fourier sampling $f(pi/\lfloor Np\rceil)$ over $k\lfloor Np\rceil$, i.e. the desired distribution. These distributions are exponentially close as long as the number of repetitions $MN/k\lfloor Np\rceil$ of this initial superposition (this ratio corresponds to the R in Corollary 2, Section 5.2 with $k\lfloor Np\rceil$ corresponding to N) is $\Omega\left(2^n\log^2(k\lfloor Np\rceil)\right)$. Since the ratio $MN/k\lfloor Np\rceil$ is also the approximation error we must have $MN/k\lfloor Np\rceil=\Omega\left(2^n\log^2(k\lfloor Np\rceil)\right)=O(M/2^m2^{10n})$ which holds by our choice of $M=\Omega\left(2^m2^{11n}\log^2 MN\right)$.

We note that without the results of Chapter 9, a naive analysis – see the discussion in Chapter 9, Section 9.2.3 – would require that the number of repetitions $MN/k\lfloor Np\rceil$ be at least $k\lfloor Np\rceil$ in order for the distributions to be close. This would force $M>N$ which is incompatible with the earlier condition $N=\Omega\left(2^{4n}M\right)$.

7.2.1 Proof of Lemma 5

We now prove Lemma 5. Notice that this lemma applies to any step function on $\Re$ with minimal interval 1 – we do not require that the function be periodic. We are taking the Fourier transform of this function evaluated on the fixed interval $\pm M/2$ and allowing the spacing of the evaluations to become finer and finer. The resulting distributions/superpositions approach a fixed limit which is concentrated within small multiples of $\pm M/2$. Intuitively this is because allowing the evaluations' spacing to become finer while the step function remains fixed does not add any large Fourier coefficients – these correspond to functions which vary rapidly and our step function is appearing increasingly smooth. For our purposes it suffices to prove the following Lemma about the tails of these distributions.

Lemma 5. *Let f be an integral-valued step function on $\Re$ with minimal step size ≥ 1. Let $\mathcal{D}_{MN}$ be the distribution on the integers in $\pm MN/2$ induced by sampling*

$$F_{MN}\left(\frac{1}{\sqrt{MN}}\sum_{i\in\pm MN/2}|i\rangle|f(i/N)\rangle\right).$$

Then for all k dividing N

$$Pr_{x\in\mathcal{D}_{MN}}\left(|x|>k^2M\right)=O(1/k).$$

Proof. Fix any k dividing N. Let

$$|\alpha\rangle=\frac{1}{\sqrt{MN}}\sum_{i\in\pm MN/2}|i\rangle|f(i/N)\rangle$$

and

$$|\alpha'\rangle=\frac{1}{\sqrt{MN}}\sum_{i\in\pm Mk/2}\sum_{j\in\pm N/2k}|\frac{N}{k}i+j\rangle|f(i/k)\rangle.$$

We claim that $\||\alpha\rangle-|\alpha'\rangle\|^2=O(1/k)$. This squared distance is just twice the fraction of pairs (i,j) for which $f(i/k)\neq f(i/k+j/N)$. Since $|j|<N/2k$ this can only be true for when i/k is within $1/2k$ of either end of an interval. Since the intervals have length at least 1 this occurs for at most a $1/k$ fraction of the i and likewise for the pairs (i,j).

Now, the behavior of $F_{MN}|\alpha'\rangle$ is easy to analyze. Its amplitude at $|x\rangle|c\rangle$ is

$$\frac{1}{MN}\sum_{i\in\pm Mk/2 f(i/k)=c}\sum_{j\in\pm N/2k}\omega_{MN}^{(\frac{N}{k}i+j)x}=\left(\frac{1}{Mk}\sum_{i\in\pm Mk/2 f(i/k)=c}\omega_{Mk}^{ix}\right)\left(\frac{k}{N}\sum_{j\in\pm N/2k}\omega_{MN}^{jx}\right). \tag{7.3}$$

The RHS of Equation 7.3 is easily seen to be the product of the amplitude of

$$F_{Mk}\left(\frac{1}{\sqrt{Mk}}\sum_{i\in\pm Mk/2 f(i/k)=c}|i\rangle\right) \tag{7.4}$$

at $|x \bmod Mk\rangle$ and the amplitude of

$$\sqrt{Mk} \cdot F_{MN}\left(F_{N/k}^{-1}|0\rangle\right) \tag{7.5}$$

at $|x\rangle$. The amplitudes in Superposition 7.5 fall off away from zero like $1/x$ while the amplitudes in Superposition 7.4 just keep repeating in blocks of size Mk. This will allow us to show that their product also falls off quickly away from zero. In particular, Observation 2, Section 9.2.5 gives us that the amplitude of $F_{MN}\left(F_{N/k}^{-1}|0\rangle\right)$ at $|j\rangle$ is at most

$$\sqrt{Mk} \cdot \frac{2}{|j|}. \tag{7.6}$$

For convenience we let $|\beta\rangle$ denote the Superposition 7.4, that is

$$|\beta\rangle = \sum_{i \in \pm Mk/2} \beta_i |i\rangle = F_{Mk}\left(\frac{1}{\sqrt{Mk}} \sum_{i \in \pm Mk/2 f(i/k)=c} |i\rangle\right).$$

Then we can use (7.6) to bound the sum of the amplitudes squared of $F_{MN}|\alpha'\rangle$ in the tth block of size Mk by

$$\sum_{i \in \pm Mk/2} |\beta_i|^2 \left|\frac{2Mk}{tMk}\right|^2 = O\left(\frac{1}{t^2}\right).$$

In other words the probability falls off as $1/t^2$ with the tth block of size Mk. Thus the probability of being larger than k^2M is $O(1/k)$.

Combining the closeness of $|\alpha\rangle$ and $|\alpha'\rangle$ with this falloff of $F_{MN}|\alpha'\rangle$ gives the desired result. $\square$

7.2.2 Proofs of Lemmas 6 and 7

We now prove two easy lemmas which allow us to use results from the previous Chapters about functions with integral period.

Lemma 6. *Let f be an integral-valued step function on $\Re$ with minimal step size ≥ 1 and period $p \in \Re$. Then for any t*

$$\frac{1}{MN}\left\| \sum_{i \in \pm \frac{MN}{2}} |i\rangle f(i/N) - \sum_{i \in \pm \frac{MN}{2}} |i\rangle |f(pi/Np+t)\rangle \right\|^2 = O\left(\frac{tM}{Np}\right).$$

This squared distance is just twice the probability that $f(i/N) \neq f(pi/Np+t)$. Since for all i

$$\left|\frac{i}{N} - \frac{pi}{Np+t}\right| = \left|\frac{ti}{N(Np+t)}\right| = O\left(\frac{tM}{Np}\right),$$

in order for the function values to differ, i/N must be within $O(tM/Np)$ of the end of a step interval. Since the intervals have length at least 1 this applies to at most a $O(tM/Np)$ fraction of the i.

Lemma 7. *Let $f \in C_{1/d(n)}$ be an integral-valued step function on $\Re$ with minimal step size ≥ 1 and period $p < 2^n$. Then any rescaling of f, $f(tx)$ which has integral period at least $4d(n)p$ is in $C_{1/2d(n)}$ when regarded as a function on Z (See Definition 4, Section 6.1.)*

Let $f(tx)$ be any rescaling of f with integral period $p_t > 4d(n)p$. Suppose $f(tx) \notin C_{1/2d(n)}$ as a function over Z. Then there exists a function g with integral period $p_g < p_t$ so that $f(tx)$ and g differ on less than a $1/2d(n)$ fraction of the inputs in $[0, p_g \cdot p_t]$. We can turn g into a function on $\Re$ with period p_g by letting its value at a non-integral input correspond to its value at the nearest integer. The distance between $f(tx)$ and g when regarded as functions on $\Re$ is small. In particular, since $f(tx)$ has been rescaled to have step intervals of size at least $4d(n)$, they are identical at at least a $1 - 1/2d(n) - 2/4d(n)$ fraction of the values, leading to a distance of at most $1/d(n)$. Our original function f and the function $g(x/t)$ have the same distance, with the latter function's period equal to

$p_g/t < p_f$, a contradiction to our assumption that $f \in C_{1/d(n)}$.

Chapter 8

Hidden Subgroups over the Reals and MA

8.1 Quantum vs. Classical Complexity Classes

A primary method for delineating the power of quantum computation is by comparison with various classical complexity classes. The Arthur-Merlin hierarchy [?] of probabilistically-checkable interactive proofs provides a natural backdrop for measuring quantum complexity. First, due to the inherently probabilistic nature of quantum computation, this hierarchy is a more natural choice than PH as a basis for comparison. In addition, problems like Graph Isomorphism which have defied classification as NP-complete are considered the most plausible candidates for possessing efficient quantum algorithms achieving exponential advantage over classical computation. These problems also tend to have non-trivial characterizations in the AM hierarchy – for instance, Graph Isomorphism is known to be in $Co - AM$.

Unlike PH, the Arthur-Merlin hierarchy is known not to be strict. In particular, $MA \subset AM$ and any constant number of rounds of interaction can be reduced to $AM \subseteq \Pi_2$ [?]. However, allowing polynomially many rounds of interaction yields all of $PSPACE$ – this is the well-known result $IP = PSPACE$ [?]. While it may be possible to show directly that BQP lies inside a particular level of the Arthur-Merlin hierarchy, results showing that BQP lies outside a level of the hierarchy can only be given in the relativized or oracle setting. In particular, since it is known that $P \subseteq BQP \subseteq P^{\#} \subseteq PSPACE$ a direct result of this sort would prove $P \neq PSPACE$, one of the nasty, long-standing open problems in complexity.

There is an oracle $\mathcal{O}$ separating BQP from MA, that is, for which $BQP^{\mathcal{O}} \not\subseteq MA^{\mathcal{O}}$. This was first claimed in [?] but the first proof was given in [?] via a different oracle. This also implies a separation between BQP and $MA \cup Co - MA$ due to the fact that BQP is closed under complementation. An open and intriguing question is whether there exists an oracle separating BQP from AM. There has been speculation that BQP is actually contained in AM. This is due to the fact that AM can perform an approximate count of the number the accepting paths of an NP-machine. The proof that $BQP \subseteq P^{\#}$ relies on the fact that exact counts of this form are sufficient to solve any problem in BQP and it has been conjectured that approximate counting might also be sufficient. An oracle separation of BQP from AM would be an indication to the contrary. In addition it would show that any proof of $BQP \subseteq AM$ must use non-relativizing techniques, in contrast to the result $BQP \subseteq P^{\#}$.

We exhibit two oracle promise problems which achieve the weaker separation

$BQP^{\mathcal{O}} \not\subseteq MA^{\mathcal{O}}$. The first of these promise problems is just a decision version of Simon's problem (Section 2.2) and its virtue lies in being much simpler than the oracles of [**?**] and [**?**] – the proof that it is outside of MA is almost trivial. We also give a simple variant of this problem which is in BQP by the results of Chapter 6 but which we suspect to be outside of AM, in other words, a candidate for the stronger separation result discussed above.

The second problem which separates BQP from MA is the decision version of period-finding over $\Re$, shown to have an efficient quantum solution in Chapter 7. We observe that the analogous problem over the integers is in MA, and thus demonstrate that period-finding over the reals is more difficult than its integral counterpart. This may also support the current state of knowledge about the relationship between factoring and Pell's equation. There is a reduction from factoring, which can be reduced to period-finding over the integers, to Pell's equation, which can be reduced to period-finding over the reals, but no reduction in the opposite direction exists.

8.2 MA

We take for our definition of MA a version with one-sided error which has been shown to be equivalent (see for example [**?**]) to the standard definition given in [**?**]:

Definition 8. *A promise problem $\mathcal{P}$ is in MA if and only if for all sufficiently large polynomials q there is a polynomial r and a predicate R computable in deterministic polynomial-time with access to f such that*

$$f \in \mathcal{Y}_n \longrightarrow \exists x \in \Sigma^{q(n)} \forall y \in \Sigma^{r(n)} R^f(x, y) = 1$$

and

$$f \in \mathcal{N}_n \longrightarrow \forall x \in \Sigma^{q(n)}, \left| y \in \Sigma^{r(n)} R^f(x,y) = 1 \right| < 2^{-2q(n)} 2^{r(n)}.$$

The following Lemma is implicit in the literature and is useful in proving lower bounds related to MA. For a given predicate R and a pair of strings $x \in \Sigma^{q(n)}$ and $y \in \Sigma^{r(n)}$, we say that two oracles f and g are equivalent under $R(x,y)$, or $f \sim_{R(x,y)} g$, if the runs of $R^f(x,y)$ and $R^g(x,y)$ produce identical oracle query/answer transcripts. Then we have the following:

Lemma 8. *If a promise problem $\mathcal{P} \in MA$ via R then for all n there exists an oracle $f \in \mathcal{Y}_n$ and strings and $x \in \Sigma^{q(n)}$ and $y \in \Sigma^{r(n)}$ such that*

$$\frac{Pr_{g \in \mathcal{N}_n}\left(g \sim_{R(x,y)} f\right)}{Pr_{g \in \mathcal{Y}_n}\left(g \sim_{R(x,y)} f\right)} < 2^{-q(n)}. \tag{8.1}$$

We give a proof of Lemma 8 in Section 8.3.2. As an easy application we give a proof that the following decision version of Simon's problem (Section 2.2) is outside of MA.

Promise Problem 1. *$\mathcal{NBS}$(No Bit-string)*

$\mathcal{Y}_n$: $f : (Z_2)^n \to (Z_2)^n$ is $1-1$.

$\mathcal{N}_n$: $f : (Z_2)^n \to (Z_2)^n$ is $2-1$ and there exists some b such that for all x, $f(x) = f(x \oplus b)$.

$\mathcal{NBS} \notin MA$. It is easy to see that $|\mathcal{Y}_n| = 2^n!$ and

$$|\mathcal{N}_n| = 2^n \frac{2^n!}{(2^n - 2^{n-1})!}.$$

Take any $f \in \mathcal{Y}_n$ and strings x, y and look at the transcript of $R^f(x,y)$. We can assume without loss that all such transcripts contain exactly t oracle queries, where t is bounded by the polynomial run-time of R. The number of functions in $g \in \mathcal{Y}_n$ such that $g \sim_{R(x,y)} f$, that is, which induce an oracle transcript identical to f's, is exactly $(2^n - t)!$. The number of oracles $g \in \mathcal{N}_n$ with $g \sim_{R(x,y)} f$ is at least

$$(2^n - t^2)\frac{(2^n - t)!}{(2^n - 2^{n-1})!}$$

because at most $t(t+1)/2$ hidden bit strings have been ruled out by the t oracle queries. Thus we get that

$$Pr_{g \in \mathcal{N}_n}\left(g \sim_{R(x,y)} f\right) \geq \frac{(2^n - t^2)\frac{(2^n-t)!}{(2^n-2^{n-1})!}}{2^n \frac{2^n!}{(2^n-2^{n-1})!}} = \frac{(2^n - t^2)(2^n - t)!}{2^n(2^n!)}$$

while

$$Pr_{g \in \mathcal{Y}_n}\left(g \sim_{R(x,y)} f\right) = \frac{(2^n - t)!}{2^n!}.$$

For all oracles f and strings x, y the ratio (8.1) of these two quantities is thus at least

$$\frac{2^n - t^2}{2^n} = \Omega(1)$$

since t is bounded by a polynomial. Thus we have $\mathcal{NBS} \notin MA$. $\square$

Since $\mathcal{NBS} \in BQP$ via Simon's algorithm (modified slightly to answer the appropriate decision problem), a routine diagonalization procedure – see for example [**?**] – gives the oracle separation result $BQP^{\mathcal{O}} \not\subseteq MA^{\mathcal{O}}$.

There is an easy protocol showing that $\mathcal{NBS} \in AM$ – the verifier chooses a value $y \in (Z_2)^n$ from the possible range of f and the prover provides an $x \in (Z_2)^n$ with the verifier accepting iff $f(x) = y$. It is easy to see that the prover can convince the verifier with probability 1 if $f \in \mathcal{Y}$ and with probability at most 1/2 otherwise. This is a simple example of an approximate counting protocol – in this case the size of the range of f is being estimated. Notice that the protocol actually distinguishes between arbitrary $1-1$ and $2-1$ functions from $(Z_2)^n$ to $(Z_2)^n$ and has nothing to do with the hidden bit-string structure of the $\mathcal{N}$ functions.

8.3 Period-finding over $\Re$ is outside of MA

We now prove that the period-finding problem over R, for which an efficient quantum algorithm was given in Chapter 7, is not in MA. In particular we show this for a decision version of the period-finding problem which corresponds to learning the leading bit of the period.

Promise Problem 2. $\mathcal{P}\Re$*(Period-finding over* $\Re$*)*

$\mathcal{Y}_n$*:* $f \in C_{1/3}$ *is a step function on* $\Re$ *with average interval* ≥ 1 *and period* $p < 2^n$ *satisfying* $2^m \leq p < 2^{m+1}$.

$\mathcal{N}_n$*:* $f \in C_{1/3}$ *is a step function on* $\Re$ *with average interval* ≥ 1 *and period* $p < 2^n$ *which does not satisfy* $2^m \leq p < 2^{m+1}$.

Theorem 8.

$$\mathcal{P}\Re \notin MA$$

The fact that this problem is outside of MA supports the intuition that period-finding over the reals is more difficult than over the integers. In particular, the analogous decision problem over Z,

Promise Problem 3. $\mathcal{P}\mathcal{Z}$*(Period-finding over* Z*)*

$\mathcal{Y}_n$: *$f \in C_{1/3}$ is a function on Z with integral period $p < 2^n$ satisfying $2^m \leq p < 2^{m+1}$.*

$\mathcal{N}_n$: *$f \in C_{1/3}$ is a function on Z with integral period $p < 2^n$ which does not satisfy $2^m \leq p < 2^{m+1}$.*

is in MA. In this case a proof that $f \in \mathcal{Y}_n$ could consist of the period p, the prime factorization of p, and primality certificates for each of these prime factors. If f is one-to-one on its period we can test this proof deterministically (and thus this restricted problem is in NP). We would first verify the factorization of p and the validity of the primality certificates – see [**?**] for the proof that $PRIMES \in NP$. Then we check that $f(x) = f(x + p)$ for an arbitrary choice of x. This insures that the claimed p is a multiple of the period. Finally, for each prime p_i in the factorization of p we verify that for an arbitrarily chosen x, $f(x) \neq f(x+p/p_i)$. This test, which can be done efficiently since there are at most n such primes, rules out any p which is a proper multiple of the true period. After this verification that p is in fact the period we accept iff $2^m < p < 2^{m+1}$.

For a general $f \in C_{1/3}$ we need merely randomize the function checks in the above proof, accepting if $f(x) = f(x + p)$ for a randomly chosen x and if for each i $f(x) \neq f(x + p/p_i)$ with significant probability. This gives a probabilistic check of the above proof and establishes that $\mathcal{PZ} \in MA$.

We now show that $\mathcal{P}\Re \notin MA$. The proof is based on the fact that, while in the integral case there is a short proof to rule out any multiple of the period, such a proof does not exist when the period is allowed to be rational. In the integral case we can check the function at $\leq n$ pairs of points p/p_i apart, one for each prime p_i in the factorization of p, and ensure that none of the potentially exponentially many proper divisors of p is the period.

In the real case to ensure that the function has period p we must rule out all rationals p/k, $k < p$ as possible periods. There is no similar polynomially sized set of points which can accomplish this check, even probabilistically.

$\mathcal{P}\Re \notin MA$. We first describe the restricted distributions of $\mathcal{Y}$ and $\mathcal{N}$ oracles which we will use. Picking the correct restriction of the original promise problem $\mathcal{P}\Re$ is half the battle – one must find a restriction which is fairly structured in order to count the oracles, but too much structure invariably reduces the problem to the integral version which *does* have an MA proof system.

We first fix the parameter m so that 2^m is superpolynomial in n. Then let $\{p_i, i \in I\}$ be the set of primes satisfying $1 < p_i < 2^{\frac{m}{4}}$ and note that, by the Prime Number Theorem, $|I|$ is also superpolynomial in n. Finally, let

$$N = k \prod_{i \in I} p_i$$

for some integer k.

Definition 9 ($\mathcal{Y}$ and $\mathcal{N}$). *Our $\mathcal{Y}$ functions all have period 2^m and are specified in the following manner. We choose $2^m - 1$ values uniformly at random in the set*

$$[0, 2^m] \cap \{ \textit{ fractions with denominator } N\}.$$

These are the endpoints of the step intervals of the function. We then choose a value in $\{1, \ldots, 2^m\}$ for each of our steps in such a way that the function is $1-1$ modulo its step intervals. Finally, we discard any function which has maximal step interval at least $2^{m/3}$. The number of such functions is

$$2^m!N2^m()2^m - 1$$

minus the functions discarded for having too long an interval. The fraction of functions thus discarded is very small – the probability of having an interval of length at least $2^{m/3}$ *is less than*

$$2^{\frac{2m}{3}+1}\left(1-\frac{1}{2^{\frac{2m}{3}+1}}\right)^{2^m-1} < 2^m e^{-2^{\frac{m}{3}}}$$

and we shall be able to ignore it in our calculations.

We now turn to our $\mathcal{N}$ *functions. For each* $i \in I$ $\mathcal{N}_i$ *will be a collection of functions with period* $\frac{2^m}{p_i}$*. We define the functions on their period in an manner similar to the* $\mathcal{Y}$ *functions. We choose* $\lfloor\frac{2^m}{p_i}\rfloor - 1$ *values uniformly at random in the set*

$$\left[0, \frac{2^m}{p_i}\right] \cap \{ \textit{ fractions with denominator } N\}.$$

These are the endpoints of the step intervals of the function. We then choose a value in $\{1, \ldots, 2^m\}$ *for each of our steps in such a way that the function is* $1-1$ *on its period modulo the step intervals. Again we discard the very small fraction of functions which have maximal step interval at least* $2^{m/3}$*. The number of such functions is*

$$\frac{2^m!}{\left(2^m - \lfloor\frac{2^m}{p_i}\rfloor\right)!} N\frac{2^m}{p_i}()\lfloor\frac{2^m}{p_i}\rfloor - 1$$

minus the small fraction of functions discarded for having too long an interval. Again these form such a small fraction of the total that we can effectively ignore them. Finally, we shall be interested in the class $\mathcal{N}$ *which is a weighted union of the* $\mathcal{N}_i, i \in I$*, with each* $\mathcal{N}_i$ *reweighted to have an equal number of functions.*

We note that the $\mathcal{Y}$ and $\mathcal{N}$ oracles defined above are in fact a subclass of the original promise problem $\mathcal{P}\Re$. Clearly they have period less than 2^n and average interval at least 1. The fact that they are in $C_{1/3}$ follows from the cap on the length of the maximal step interval together with the fact that they are $1-1$ modulo their steps.

Let $R(x, y)$ be any deterministic predicate which runs in time $t(n)$ and purports to yield an MA proof system for $\mathcal{P}\mathfrak{R}$ with parameters $q(n)$ and $r(n)$. Then clearly $R(x, y)$ also yields an MA proof system for the $\mathcal{Y}$ and $\mathcal{N}$ oracles defined above with the same parameters. We can further assume that on this restricted problem all oracle queries are made on inputs in the interval $[0, 2^m]$ by interpreting the original queries $\pmod{2^m}$.

Recall the equivalence relation $\sim_{R(x,y)}$ defined in Section 8.2. The following lemma is the main technical result establishing our theorem and is proved in Section 8.3.1:

Lemma 9. *There is a constant $c > 0$ such that for all sufficiently large n, $f \in \mathcal{Y}$, $i \in I$, and strings $x \in \Sigma^{q(n)}$ and $y \in \Sigma^{r(n)}$*

$$\frac{Pr_{g \in \mathcal{N}_i}\left(g \sim_{R(x,y)} f\right)}{Pr_{g \in \mathcal{Y}}\left(g \sim_{R(x,y)} f\right)} \geq c \tag{8.2}$$

unless the transcript of $R^f(x, y)$ includes a pair of oracle inputs (u, v) satisfying

$$k\frac{2^m}{p_i} - 2^{m/3} < u - v < k\frac{2^m}{p_i} + 2^{m/3}, \tag{8.3}$$

for some integer k satisfying $|k| < p_i$.

Informally this says that R can only distinguish between the $\mathcal{Y}$ oracles which have period 2^m and the $\mathcal{N}_i$ oracles with period $2^m/p_i$ if it actually queries a pair of intervals which are a multiple of $2^m/p_i$ apart and thus rules out the possibility of a $\mathcal{N}_i$ oracle.

We now turn to the question of distinguishing $\mathcal{Y}$ oracles from the full collection of $\mathcal{N}$ oracles. The idea is that a successful MA proof would have to rule out the possibility that $f \in \mathcal{N}_i$ for almost all $i \in I$ and thus examine the function on pairs (u, v) of the above form for almost all $i \in I$, but this requires making exponentially many oracle queries in polynomial-time!

We first claim that any pair of queries (u, v) to the oracle can satisfy Equation (8.3) for at most n of the $i \in I$. Suppose (u, v) satisfies Equation (8.3) for the prime p and the integer k. Then in order for it to also satisfy the same equation for another prime p_i we must have

$$\left| k\frac{2^m}{p} - k_i\frac{2^m}{p_i} \right| < 2^{m/3+1}$$

which implies

$$\left| \frac{k}{p} - \frac{k_i}{p_i} \right| < \frac{1}{(2^{m/4})^2}.$$

By our choice of $p, p_i < 2^{m/4}$ these fractions must therefore be exactly equal, or $k = lpp_i$ for some integer l. Since $|k| < 2^n$ it can have at most n distinct prime factors and thus Equation (8.3) can be satisfied simultaneously for at most n of the $i \in I$.

We now show that for all $f \in \mathcal{Y}$ and for all x, y,

$$\frac{Pr_{g \in \mathcal{N}}\left(g \sim_{R(x,y)} f\right)}{Pr_{g \in \mathcal{Y}}\left(g \sim_{R(x,y)} f\right)} = \Omega(1)$$

This will establish the result since it is a violation of Lemma 8. Now,

$$\begin{aligned}
\frac{Pr_{g \in \mathcal{N}}\left(g \sim_{R(x,y)} f\right)}{Pr_{g \in \mathcal{Y}}\left(g \sim_{R(x,y)} f\right)} &= \frac{\left|\{g \in \mathcal{N} | f \sim_{R(x,y)} g\}\right|}{|\mathcal{N}|} \frac{|\mathcal{Y}|}{\left|\{g \in \mathcal{Y} | f \sim_{R(x,y)} g\}\right|} && (8.4)\\
&= \frac{\sum_{i \in I}\left|\{g \in \mathcal{N}_i | f \sim_{R(x,y)} g\}\right|}{\sum_{i \in I}|\mathcal{N}_i|} \frac{|\mathcal{Y}|}{\left|\{g \in \mathcal{Y} | f \sim_{R(x,y)} g\}\right|} && (8.5)\\
&= \sum_{i \in I} \frac{\left|\{g \in \mathcal{N}_i | f \sim_{R(x,y)} g\}\right|}{|I|\,|\mathcal{N}_i|} \frac{|\mathcal{Y}|}{\left|\{g \in \mathcal{Y} | f \sim_{R(x,y)} g\}\right|} && (8.6)\\
&= \sum_{i \in I} \frac{1}{|I|} \frac{Pr_{g \in \mathcal{N}_i}\left(g \sim_{R(x,y)} f\right)}{Pr_{g \in \mathcal{Y}}\left(g \sim_{R(x,y)} f\right)} && (8.7)\\
& && (8.8)
\end{aligned}$$

where the second to the last equation follows from the fact that the $\mathcal{N}_i$ have been given equal weights. By throwing out the at most nt^2 $i \in I$ for which pairs of queries satisfying Equation (8.3) have been made, and applying the bound in Lemma 9 to the rest we have that the above quantity is

$$\Omega\left(\sum_{i<|I|-nt^2} \frac{1}{|I|}\right) = \Omega\left(\frac{|I|-nt^2}{|I|}\right) = \Omega(1)$$

where the last equality follows from the fact that $|I|$ is superpolynomial in n. This completes the proof. $\square$

8.3.1 Proof of Lemma 9

Lemma 9. *There is a constant $c > 0$ such that for all sufficiently large n, $f \in \mathcal{Y}$, $i \in I$, and strings $x \in \Sigma^{q(n)}$ and $y \in \Sigma^{r(n)}$*

$$\frac{Pr_{g\in\mathcal{N}_i}\left(g \sim_{R(x,y)} f\right)}{Pr_{g\in\mathcal{Y}}\left(g \sim_{R(x,y)} f\right)} \geq c \tag{8.9}$$

unless the transcript of $R^f(x,y)$ includes a pair of oracle inputs (u,v) satisfying

$$k\frac{2^m}{p_i} - 2^{m/3} < u - v < k\frac{2^m}{p_i} + 2^{m/3}, \tag{8.10}$$

for some integer $|k| < p_i$.

We first define an equivalence relation on our functions which is a refinement of $\sim_{R(x,y)}$. We let $f{\sim}g$ if both $f \sim_{R(x,y)} g$ *and* the at most t step intervals queried on a run of $R^f(x,y)$ are identical on their endpoints. In other words, not only the values of the steps which are queried but also the steps themselves are identical. This is clearly a refinement of $\sim_{R(x,y)}$ and thus it suffices to prove the above lemma with $\sim_{R(x,y)}$ replaced by $\sim$.

Fix any $f \in \mathcal{Y}$, $i \in I$, and strings $x \in \Sigma^{q(n)}$ and $y \in \Sigma^{r(n)}$. We can assume without loss of generality that exactly t intervals are queried on any run, where $t = t(n)$ is the run-time of R. Let T denote the total length of the step intervals which are queried in $R^f(x,y)$ – note that $T < t2^{m/3}$ since no interval is longer than $2^{m/3}$. The number of $\mathcal{Y}$ functions g for which $f \sim g$ is

$$(2^m - t)!N(2^m - T)()2^m - 2t - 1.$$

minus the exponentially small fraction of these functions which have maximal interval greater than $2^{m/3}$.

Now, if none of these t intervals overlap when they are mapped back $\pmod{2^m/p_i}$ to the interval $[0, 2^m/p_i]$, and this is the case when Equation (8.10) is not satisfied by any pair of queries, then the number of $\mathcal{N}$ functions g for which $f \sim g$ is

$$\frac{(2^m - t)!}{\left(2^m - \lfloor \frac{2^m}{p_i} \rfloor\right)!} N(\frac{2^m}{p_i} - T)()\lfloor \frac{2^m}{p_i} \rceil - 2t - 1,$$

minus the small fraction of these functions which have maximal interval greater than $2^{m/3}$. Here we also use the fact that N is a multiple of p_i for each $i \in I$. This ensures that the endpoints of the original intervals interpreted $\pmod{2^m/p_i}$ are valid choices for the $\mathcal{N}_i$ oracles.

By cancelling all the factorials in the ratio in question,

$$\frac{Pr_{g\in\mathcal{N}_i}\left(g \sim_{R(x,y)} f\right)}{Pr_{g\in\mathcal{Y}}\left(g \sim_{R(x,y)} f\right)} \approx \frac{N(\frac{2^m}{p_i} - T)()\lfloor\frac{2^m}{p_i}\rceil - 2t - 1}{N\frac{2^m}{p_i}()\lfloor\frac{2^m}{p_i}\rceil - 1} \frac{N2^m()2^m - 1}{N(2^m - T)()2^m - 2t - 1}.$$

where the approximation reflects the fact that we have thrown out an exponentially small fraction from each class for having too large a maximal interval. At this point it is easy to see that this can be ignored. Since we are free to choose $N = k\prod_{i\in I} p_i$ as large as we want we use the fact that when N is sufficiently large $N()l \approx \frac{N^l}{l!}$ to conclude that Equation 8.3.1 is approximately

$$\frac{\left(\lfloor\frac{2^m}{p_i}\rceil - 1\right)!\,(2^m - 2t - 1)!}{\left(\lfloor\frac{2^m}{p_i}\rceil - 2t - 1\right)!\,(2^m - 1)!} \cdot \frac{\left(N\frac{2^m}{p_i}\right)^{\lfloor\frac{2^m}{p_i}\rceil - 2t - 1}}{\left(N\frac{2^m}{p_i}\right)^{\lfloor\frac{2^m}{p_i}\rceil - 1}} \frac{(N2^m)^{2^m - 1}}{(N2^m)^{2^m - 2t - 1}} \cdot \frac{\left(1 - \frac{T}{2^m/p_i}\right)^{\lfloor\frac{2^m}{p_i}\rceil - 2t - 1}}{\left(1 - \frac{T}{2^m}\right)^{2^m - 2t - 1}}. \tag{8.11}$$

The first ratio in Equation 8.3.1 can be seen to approach p_i^{-2t} as n (and thus m) goes to infinity by cancelling terms in the factorials. In a similar manner the second ratio can be seen to approach p_i^{2t}. We now proceed to show that the third ratio is approaches 1 and the Lemma follows. We can rewrite this ratio as

$$\left[\frac{\left(1 - \frac{1}{2^m/Tp_i}\right)^{\frac{2^m}{Tp_i}}}{\left(1 - \frac{1}{2^m/T}\right)^{\frac{2^m}{T}}}\right]^T \cdot \frac{\left(1 - \frac{1}{2^m/Tp_i}\right)^{\epsilon - 2t - 1}}{\left(1 - \frac{1}{2^m/T}\right)^{-2t - 1}}$$

where $\epsilon = \lfloor\frac{2^m}{p_i}\rceil - \frac{2^m}{p_i} \leq 1/2$. Now the second of these ratios has numerator and denominator both close to 1 since $t << 2^m/Tp_i$. Thus we can ignore this ratio and focus on the first.

We use the fact that the expression $(1 - 1/n)^n$ converges to e^{-1} with error $O(1/n)$ to conclude that the ratio

$$\frac{\left(1 - \frac{1}{2^m/Tp_i}\right)^{\frac{2^m}{Tp_i}}}{\left(1 - \frac{1}{2^m/T}\right)^{\frac{2^m}{T}}}$$

is within $O(Tp_i/2^m)$ of 1. Finally since

$$T << \frac{2^m}{Tp_i}$$

this ratio raised to the Tth power is still very close to 1 and the result follows.

8.3.2 Proof of Lemma 8

Lemma 8. *If a promise problem $\mathcal{P} \in MA$ via R then for all n there exists an oracle $f \in \mathcal{Y}_n$ and strings and $x \in \Sigma^{q(n)}$ and $y \in \Sigma^{r(n)}$ such that*

$$\frac{Pr_{g \in \mathcal{N}_n}\left(g \sim_{R(x,y)} f\right)}{Pr_{g \in \mathcal{Y}_n}\left(g \sim_{R(x,y)} f\right)} < 2^{-q(n)}.$$

Proof. Since there are $2^{q(n)}$ possible proof strings x there exists at least one such string which serves as a valid proof for at least a $2^{-q(n)}$ fraction of the oracles. Fix any such proof x. We have that

$$\forall y \in \Sigma^{r(n)}\ |\{g \in \mathcal{Y} | R^g(x,y) = 1\}| \geq 2^{-q(n)}|\mathcal{Y}|,$$

and thus

$$\sum_{y \in \Sigma^{r(n)}} |\{g \in \mathcal{Y} | R^g(x,y) = 1\}| \geq 2^{-q(n)}|\mathcal{Y}|2^{r(n)}.$$

If Equation 8.3.2 is violated for all $f, x,$ and y then by viewing each set $\{g \in \mathcal{N} | R^g(x,y) = 1\}$ as a union of $\sim_{R(x,y)}$ equivalence classes we have that for all y

$$|\{g \in \mathcal{N} | R^g(x,y) = 1\}| \geq 2^{-q(n)}\,|\{g \in \mathcal{Y} | R^g(x,y) = 1\}|\,\frac{|\mathcal{N}|}{|\mathcal{Y}|}.$$

Putting these together we get that

$$\sum_{y \in \Sigma^{r(n)}} |\{g \in \mathcal{N} | R^g(x,y) = 1\}| = \sum_{g \in \mathcal{N}} \left|\{y \in \Sigma^{r(n)} | R^g(x,y) = 1\}\right| \geq 2^{-2q(n)}|\mathcal{N}|2^{r(n)}.$$

But this implies that there exists a $g \in \mathcal{N}$ such that

$$|\{y | R^g(x,y) = 1\}| \geq 2^{-2q(n)}2^{r(n)},$$

contradicting the definition of MA. $\square$

Chapter 9

Fourier Transform Theorems

In this chapter we establish the technical results leading to the QFT Algorithm 3 and the Fourier Sampling Algorithms 4 and 5. First prove a version of the Fourier Sampling Lemma 9 ([**?**],[**?**]) and show how this leads to a simple algorithm for approximating the QFT over an arbitrary cyclic group. While this technique, like the quantum chirp-z method of Section 3.3, can only be used to replace a finite number of QFT's in a given computation, it may be of independent interest. Also, the proofs of Theorems 10 and 11 which lead directly to the highly efficient Algorithms 3 and 4 rely on an elaboration of the techniques used in this earlier lemma.

9.1 Fourier Sampling Lemma

In this section we prove a relationship between the Fourier transforms over different moduli of a fixed vector. In particular, let $|v\rangle = \sum_{i<N} v_i|i\rangle$ be a unit vector and let $|\hat{v}\rangle$ and

$|\hat{v}^M\rangle$ be its Fourier transforms mod N and M respectively, where $M > N$.[1]

We exhibit a subvector of $|\hat{v}^M\rangle$ whose direction is a good approximation to $|\hat{v}\rangle$'s whenever M is sufficiently large. In particular, let j' denote the integer nearest $\frac{M}{N}j$ with ties broken by some standard convention. Let $|\hat{v}^M\rangle'$ be the subvector of $|\hat{v}^M\rangle$ consisting of the entries indexed by integers j' renormalized by $\sqrt{\frac{M}{N}}$. That is,

$$|\hat{v}^M\rangle' = \sqrt{\frac{M}{N}} \sum_{j<N} \hat{v}^M_{j'} |j\rangle.$$

Then the L_2 distance between the vector $|\hat{v}\rangle$ and $|\hat{v}^M\rangle'$ becomes arbitrarily small as M is increased relative to N. The fact that this is true for $M = \Omega(N^{3/2})$ is almost trivial, but we show that this is already true for $M = \Omega(N \log N)$. This exponential improvement in the ratio $\frac{M}{N}$ is crucial for the quantum applications discussed in Section 9.1.1.

First, it is easily seen that for $M = \Omega\left(\frac{N^{3/2}}{\epsilon}\right)$, $|\hat{v}\rangle$ and $|\hat{v}^M\rangle'$ are ϵ-close in L_2 norm. The square of the L_2 distance between the vectors $|\hat{v}\rangle$ and $|\hat{v}^M\rangle'$ is given by

$$\begin{aligned}
\sum_{j<N} \left| v_j - \sqrt{\frac{M}{N}} \hat{v}^M_{j'} \right|^2 &= \sum_{j<N} \left| \frac{1}{\sqrt{N}} \sum_{i<N} v_i \omega_N^{ij} - \frac{1}{\sqrt{N}} \sum_{i<N} v_i \omega_M^{ij'} \right|^2 \\
&= \frac{1}{N} \sum_{j<N} \left| \sum_{i<N} v_i \omega_N^{ij} - \sum_{i<N} v_i \omega_N^{ij} \omega_M^{i\delta_j} \right|^2 \\
&= \frac{1}{N} \sum_{j<N} \left| \sum_{i<N} v_i \omega_N^{ij} (1 - \omega_M^{i\delta_j}) \right|^2 \\
&\leq \frac{1}{N} \sum_{j<N} \left(\sum_{i<N} \left| v_i \omega_N^{ij} \right| \left| 1 - \omega_M^{i\delta_j} \right| \right)^2,
\end{aligned}$$

where $\delta_j = j' - \frac{M}{N} j < 1$.

We first use the fact that since $|i\delta_j| < N$,

$$\left| 1 - \omega_M^{i\delta_j} \right| < \frac{N}{M},$$

[1]We interpret $|v\rangle$ as a unit vector of length M with entries greater than N uniformly equal to zero.

and then apply the inequality

$$\sum_{i<N} |u_i| < \sqrt{N}$$

which holds for any unit vector $|u\rangle$, to obtain

$$\begin{aligned} \frac{1}{N}\sum_{j<N}\left(\sum_{i<N}\left|v_i\omega_N^{ij}\right|\left|1-\omega_M^{i\delta_j}\right|\right)^2 &\leq \frac{1}{N}\sum_{j<N}\left(\frac{N}{M}\sum_{i<N}\left|v_i\omega_N^{ij}\right|\right)^2 \\ &\leq \frac{N^3}{M^2} \end{aligned}$$

from which the claim follows.

However, this relationship cannot be exploited easily in the quantum setting. In short, in order for $|\hat{v}^M\rangle'$ to be a good approximation to $|\hat{v}\rangle$, M must be chosen so that the ratio $\frac{M}{N}$ is exponentially large. But then the desired subvector of $|\hat{v}^M\rangle$ is an exponentially small fraction of the whole of $|\hat{v}^M\rangle$ and cannot efficiently be recovered.

But this relationship actually holds for much smaller M. In particular, we show that that it holds for $M = \Omega\left(\frac{N\log N}{\epsilon}\right)$, an exponential improvement in the ratio $\frac{M}{N}$.

Theorem 9. *Given any unit vector* $|v\rangle = \sum_{i<N} v_i|i\rangle$

$$\left\| |\hat{v}\rangle - |\hat{v}^M\rangle' \right\| = O\left(\frac{N\log N}{M}\right).$$

A version of this theorem which referred only to the distributions induced by $|\hat{v}\rangle$ and $|\hat{v}^M\rangle'$ first appeared in [?]. The proof was later simplified, and the bounds improved, in [?]. The proof given in Section 9.1.3 is based on this simplification.

9.1.1 Application: An Approximate QFT over an Arbitrary Modulus N

We give a simple algorithm for an approximate QFT over an arbitrary modulus based on Theorem 9. This algorithm suffers from the same drawbacks as the chirp-z method discussed in Section 3.3, namely it only succeeds with inverse polynomial probability and thus can only be used to replace a constant number of QFT's in a given quantum procedure. However, the number of repetitions required to achieve an ϵ approximation with high probability is now linear rather than quadratic in $O(\frac{1}{\epsilon})$. Furthermore the algorithm is extremely simple. We note that this is particularly true in the Fourier Sampling setting, that is, if the transform to be approximated occurs as the last step in a quantum algorithm with only the distribution induced by the final superposition being of interest. In this case measurement can take place immediately following Step 1 and the rounding procedure can be accomplished classically. This gives us an very short quantum subroutine, but one which must be repeated many times for the required result, a trade-off which may be very desirable when decoherence is taken into account.

Let N and ϵ be given. Choose $M = \Omega\left(\frac{N \log N}{\epsilon}\right)$:

Algorithm 8. *Input:* $|\alpha\rangle$

1. *Transform* $|\alpha\rangle$ *over* Z_M*:*

$$|\alpha\rangle \longrightarrow F_M|\alpha\rangle$$

2. *If* $x = \lfloor \frac{M}{N} i \rceil$ *map*

$$|x\rangle|0\rangle \longrightarrow |i\rangle|1\rangle.$$

3. Measure the second register.

If a 1 *is measured in the second register which occurs with probability* $\frac{\epsilon}{\log N}$*, then we output the successful approximate QFT.*

The correctness of this procedure follows directly from our theorem. If a 1 is measured in the second register then we have collapsed to a superposition in the direction of

$$\sum_{j<N} \hat{\alpha}^{M}_{j'} |j\rangle,$$

By our Theorem the vector is ϵ-close to the desired

$$\sum_{j<N} \hat{\alpha}_j |j\rangle.$$

Moreover, since

$$\sqrt{\frac{M}{N}} \sum_{j<N} \hat{\alpha}^{M}_{j'} |j\rangle,$$

is approximately a unit vector,

$$\sum_{j<N} |\hat{\alpha}^{M}_{j'}|^2 = \frac{N}{M}$$

and the success probability is also correct.

9.1.2 Two Claims

To prove Theorem 9 we first examine the special case when the initial vector $|v\rangle$ is an element of the Fourier basis mod N, in other words

$$|v\rangle = |\hat{j}\rangle = \sum_{i<N} \frac{1}{\sqrt{N}} \omega_N^{-ij} |i\rangle.$$

The Fourier transform over N of $|\hat{j}\rangle$ is just the standard basis vector $|j\rangle$, i.e. a pointmass at j. We let j^M denote the Fourier transform over M of $|\hat{j}\rangle$ and $|j^M\rangle'$ the subvector of j^M

at entries of the form $i' = \lfloor \frac{M}{N} i \rceil$ renormalized by $\sqrt{\frac{M}{N}}$ in keeping with our earlier notation. The vector j^M is a smeared pointmass concentrated near j' and the entries of $|j^M\rangle'$ satisfy the following:

Claim 2. $\left|1 - j_j^{M\prime}\right| \leq \pi \frac{N}{M}$

Claim 3. *For $k \neq j$,*

$$\left|j_k^{M\prime}\right| \leq \frac{1}{|k-j|_N} \frac{N}{M}$$

where $|x|_N = \begin{cases} x \bmod N & \textit{if } 0 \leq x \bmod N \leq \frac{N}{2} \\ -x \bmod N & \textit{otherwise} \end{cases}$

These claims are proved in Section 9.1.4. They yield a version of our main theorem in the special case that $|v\rangle$ is a Fourier basis vector:

Observation 1. *If $|v\rangle = |\hat{j}\rangle$ is an element of the Fourier basis mod N and $M = \Omega\left(\frac{N}{\epsilon}\right)$, then*

$$\left\| |\hat{v}\rangle - |\hat{v}^M\rangle' \right\| < \epsilon.$$

We leave the proof of Observation 1 to the reader. This Observation does not lead directly to our Theorem 9. In particular if we try to extend it linearly to allow for an arbitrary vector $|v\rangle$ we are forced to choose $M = \Omega(\frac{N^{3/2}}{\epsilon})$ to achieve a bound of ϵ – the argument is that of Section 9.1 now expressed in the Fourier rather than the standard basis. Fortunately, a more careful examiniation of Claim 3 gives us crucial information about the structure of the error vectors

$$|j\rangle - |j^M\rangle'$$

which will allow us to conclude our theorem.

9.1.3 Proof of Theorem 9

We wish to bound the quantity

$$\left\| |\hat{v}\rangle - |\hat{v}^M\rangle' \right\| = \left\| \sum_{j<N} \hat{v}_j \left(|j\rangle - |j^M\rangle' \right) \right\| = \sum_{i<N} \left| \sum_{j<N} \hat{v}_j \left(|j\rangle - |j^M\rangle' \right)_i \right|^2 . \tag{9.1}$$

This is the squared length of the vector which results from applying the matrix with ijth entry $\left(|j\rangle - |j^M\rangle'\right)_i$ to the unit vector $|\hat{v}\rangle$, in other words the best bound on this expression is exactly the squared operator norm of this matrix. By Claims 2 and 3 we have

$$\left| \left(|j\rangle - |j^M\rangle' \right)_j \right| < \pi \frac{N}{M}$$

and for $i \neq j$

$$\left| \left(|j\rangle - |j^M\rangle' \right)_i \right| < \frac{2}{|i-j|_N} \frac{N}{M}.$$

It suffices, then, to bound the squared operator norm of the matrix A with

$$A_{ij} = \begin{cases} \pi \frac{N}{M} & \text{if } i = j \\ \frac{1}{|i-j|_N} \frac{N}{M} & \text{otherwise} \end{cases}$$

This $N \times N$ matrix has the property that each row is the shift by one $(\bmod N)$ of the previous row, i.e. $A_{i,j} = A_{i+1 \bmod N, j+1 \bmod N}$, and all its entries are nonnegative reals. Because of this shift property – such a matrix is commonly referred to as circulant – its eigenvalues are all of the form $\sum_{i<N} \omega_N^{jk} A_{ij}$ for some integer k. Moreover since the entries A_{ij} are nonnegative reals the maximum eigenvalue is found by setting $k = 0$, corresponding to an eigenvector with all equal entries. This maximum eigenvalue is precisely the operator norm of A and can be found by taking the sum of any row of the matrix. Using the fact

that $\sum_{i<N} \frac{1}{i} = \log N$ we have the sum of a row is $O\left(\frac{N \log N}{M}\right)$ and thus

$$\left\| |\hat{v}\rangle - |\hat{v}^M\rangle' \right\| = O\left(\frac{N \log N}{M}\right) \tag{9.2}$$

which establishes our theorem.

9.1.4 Proofs of Claims 2 and 3

We now prove Claims 2 and 3.

Proof of Claim 2. To establish

$$\left|1 - j_j^{M\prime}\right| \leq \pi \frac{N}{M}$$

we note that

$$\begin{aligned} \left|1 - j_j^{M\prime}\right| &= \left|1 - \sqrt{\frac{M}{N}} j_{j'}^{M}\right| &(9.3)\\ &= \left|1 - \frac{1}{N} \sum_{i<N} \omega_N^{-ij} \omega_M^{ij'}\right| &(9.4)\\ &= \left|1 - \frac{1}{N} \sum_{i<N} \omega_M^{i\epsilon}\right| &(9.5) \end{aligned}$$

where $\epsilon = j' - \frac{M}{N} j \leq 1/2$ This quantity is easily seen to be less than the arclength $2\pi\epsilon\frac{N}{M}$ and the claim follows. □

Proof of Claim 3. We now establish that for $k \neq j$,

$$\left|j_k^{M\prime}\right| \leq \frac{1}{|k - j|_N} \frac{N}{M}$$

where $|x|_N = \begin{cases} x \bmod N & \text{if } 0 \leq x \bmod N \leq \frac{N}{2} \\ -x \bmod N & \text{otherwise.} \end{cases}$

$$\begin{aligned} \left|j_k^{M\prime}\right| &= \left|\sqrt{\frac{M}{N}} j_{k'}^M\right| && (9.6)\\ &= \left|\frac{1}{N}\sum_{i<N} \omega_N^{-ij}\omega_M^{ik'}\right| && (9.7)\\ &= \left|\frac{1}{N}\sum_{i<N} \omega_N^{i(k-j+\frac{N}{M}\epsilon)}\right| && (9.8)\\ &= \frac{1}{N}\frac{\left|\omega_N^{N(k-j+\frac{N}{M}\epsilon)}-1\right|}{\left|\omega_N^{k-j+\frac{N}{M}\epsilon}-1\right|}, && (9.9)\end{aligned}$$

where $\epsilon = k' - \frac{M}{N}k \leq 1/2$. The numerator $\left|\omega_N^{N(k-j+\frac{N}{M}\epsilon)}-1\right|$ is at most the arclength $2\pi\frac{N}{M}\epsilon < \pi\frac{N}{M}$ and the denominator $\left|\omega_N^{k-j+\frac{N}{M}\epsilon}-1\right|$ is at least $\frac{2\pi|k-j+\frac{N}{M}\epsilon|_N}{N} > \frac{\pi|k-j|_N}{N}$ leading directly to the claimed bound.

□

9.2 Fourier Transform Theorems

In this section we establish the technical results leading to Algorithms 3 and 4. In particular, we prove a relationship between the transform $|\hat{v}\rangle$ over N of a given vector $|v\rangle$ and the transform over $M > N$, not of that same vector $|v\rangle$ (as in previous Section), but of a vector consisting of many repetitions of $|v\rangle$. By repeating the vector $|v\rangle$ many times and transforming over a large M we get a vector with not just one length N subvector whose renormalization approximates $|\hat{v}\rangle$ (as in the previous Section) but a vector for which most length N subvectors have this property. Analogous to the previous section, the fact that this is true when $|v\rangle$ is repeated $\Omega(N)$ times is easy to prove but we show, via an amplification of the circulant argument of Section 9.1.3, that this holds when the number of repetitions is only $O(\log^2 N)$. This improvement is responsible for the improved efficiency of Algorithms 3 and 4 over earlier methods and is also used crucially in the proof of Theorem 7.

More formally, let $|v\rangle = \sum_{i<N} v_i|i\rangle$ be an arbitrary unit vector, and let $|w\rangle$ be the unit vector consisting of R repetitions of $|v\rangle$, that is

$$|w\rangle = \frac{1}{\sqrt{R}} \sum_{j<R} \sum_{i<N} v_i |jN+i\rangle$$

Then we can establish a strong relationship between the vectors $|\hat{v}\rangle$ and $|\hat{w}^M\rangle$ for sufficiently large R and M. Recall from the previous section that i' denotes the integer nearest i.

Theorem 10. *Let $|v\rangle$ and $|w\rangle$ be as above. Then for any $M > RN$ there is a vector $|u\rangle = \sum_{|t|<\frac{M}{2N}} u_t|t\rangle$ so that*

$$\left\| |\hat{w}^M\rangle - \sum_{i<N} \hat{v}_i |u\rangle^{i'} \right\| < \frac{4RN}{M} + \frac{8\log N}{\sqrt{R}}$$

where $|u\rangle^{i'} = \sum_{|t|<\frac{M}{2N}} u_t|i'+t\rangle$ is the vector $|u\rangle$ with indices shifted by i'.

This theorem forms the basis for the Fourier Transform algorithm of Section 5.1. By measuring the offset from the nearest i', the superposition $\sum_{i<N} \hat{v}_i|u\rangle^{i'}$ collapses exactly to the desired $|\hat{v}\rangle$. This property approximately holds for the superposition $|\hat{w}^M\rangle$ (which we can generate) by virtue of its closeness to $\sum_{i<N} \hat{v}_i|u\rangle^{i'}$.

As a byproduct of the proof of Theorem 10 we get a related theorem which is useful in the Fourier Sampling setting, that is, in the case where we are concerned with the distribution induced by the final superposition. We let $\mathcal{D}_{|\hat{v}\rangle}$ be the probability distribution on the set $\{0, ...N-1\}$ induced by measuring $|\hat{v}\rangle$ and $\mathcal{D}_{|\hat{w}^M\rangle}$ be the distribution on the same set induced by measuring $|\hat{w}^M\rangle$ and interpreting integers within $M/2N$ of i' as i.

More formally,

$$\mathcal{D}_{|\hat{v}\rangle}(i) = |\hat{v}_i|^2$$

and

$$\mathcal{D}_{|\hat{w}^M\rangle}(i) = \sum_{|t|<\frac{M}{2N}} \left|\hat{w}^M_{i'+t}\right|^2 .$$

Then we can prove the following theorem:

Theorem 11. *Let $|v\rangle$ and $|w\rangle$ be as above. Then for any $M \geq NR$*

$$\left\|\mathcal{D}_{|\hat{w}^M\rangle} - \mathcal{D}_{|\hat{v}\rangle}\right\|_1 < O\left(\frac{\log N}{\sqrt{R}}\right).$$

Notice that in order to make these distributions close we need only make sure that R is sufficiently large and then M can be taken to be any integer greater than RN.

9.2.1 Proof of Theorem 10

First note that the Fourier transform over RN of

$$|w\rangle = \frac{1}{\sqrt{R}} \sum_{j<R} \sum_{i<N} v_i |jN + i\rangle$$

is

$$|\hat{w}\rangle = \sum_{i<N} \hat{v}_i |Ri\rangle. \tag{9.10}$$

Recall that we are trying to show that there exists some vector $|u\rangle$ supported on the integers in the interval $(-\frac{M}{2N}, \frac{M}{2N})$ such that $|\hat{w}^M\rangle$ is close to a vector of the form

$$\sum_{i<N} \hat{v}_i |u\rangle^{i'}$$

where $|u\rangle^{i'}$ is the vector $|u\rangle$ with indices shifted by $i' = \lfloor \frac{M}{N} i \rceil$. In the case that $M = RN$ Equation (9.10) immediately yields our theorem with the vector $|u\rangle = |0\rangle$ and no error at all. For a general $M > RN$

$$|\hat{w}^M\rangle = \sum_{i<N} \hat{v}_i F_M F_{RN}^{-1}(|Ri\rangle),$$

We let $|i^M\rangle = F_M F_{RN}^{-1}(|Ri\rangle)$ and thus

$$|\hat{w}^M\rangle = \sum_{i<N} \hat{v}_i |i^M\rangle.$$

The $|i^M\rangle$ are neither supported on the intervals $(i' - \frac{M}{2N}, i' + \frac{M}{2N})$ nor the shifts by i' of a fixed vector, but we show that for sufficiently large R and M these conditions approximately hold.

To this end we define $|b_i\rangle$ (for "bump") to be the vector $|i^M\rangle$ restricted to the integers in the open interval $(i' - \frac{M}{2N}, i' + \frac{M}{2N})$, an interval which we denote by (i'). We let $|t_i\rangle$ (for "tail") be the rest of $|i^M\rangle$. Thus the $|t_i\rangle$ are supported on the indices outside of (i') and we have $|t_i\rangle = |i^M\rangle - |b_i\rangle$. Note also that

$$|\hat{w}^M\rangle = \sum_{i<N} \hat{v}_i |i^M\rangle = \sum_{i<N} \hat{v}_i |b_i\rangle + \sum_{i<N} \hat{v}_i |t_i\rangle.$$

Finally, let $|b_0\rangle^{i'}$ be the vector $|b_0\rangle$ shifted by i'. Our aim will be to show that $|b_0\rangle$ is our candidate for $|u\rangle$, in other words that

$$|\hat{w}^M\rangle = \sum_{i<N} \hat{v}_i |b_i\rangle + \sum_{i<N} \hat{v}_i |t_i\rangle \approx \sum_{i<N} \hat{v}_i |b_0\rangle^{i'}.$$

We first bound $\left\| \sum_{i<N} \hat{v}_i |t_i\rangle \right\|$, then show that each $|b_i\rangle$ is very close to $|b_0\rangle^{i'}$. Since the $|b_i\rangle$'s have disjoint support the closeness of the vectors follows. More formally, we will prove the following two claims:

Claim 4.

$$\left\| |\hat{w}^M\rangle - \sum_{i<N} \hat{v}_i |b_i\rangle \right\| = \left\| \sum_{i<N} \hat{v}_i |t_i\rangle \right\| \leq \frac{8 \log N}{\sqrt{R}} \tag{9.11}$$

Claim 4 states that making R large (i.e. increasing the number of repetitions of $|v\rangle$) reduces the effect of these tails.

Claim 5. *Let $|b_0\rangle^{i'}$ be the superposition $|b_0\rangle$ shifted by i'. Then*

$$\left\| |b_0\rangle^{i'} - |b_i\rangle \right\| < \frac{4RN}{M}.$$

From Claim 5 and the fact that the $|b_i\rangle$ have disjoint supports,

$$\left\| \sum_{i<N} \hat{v}_i |b_0\rangle^{i'} - \sum_{i<N} \hat{v}_i |b_i\rangle \right\| \leq \frac{4RN}{M}.$$

Combining this with Claim 4 via the triangle inequality we have,

$$\left\| |\hat{w}^M\rangle - \sum_{i<N} \hat{v}_i |b_0\rangle^{i'} \right\| \leq \frac{4RN}{M} + \frac{8 \log N}{\sqrt{R}}, \tag{9.12}$$

as desired.

9.2.2 Proof of Theorem 11

In this case we wish to show that the distribution $\mathcal{D}_{|\hat{v}\rangle}$ on $\{0, ..., N-1\}$ induced by sampling $|\hat{v}\rangle$ and the distribution $\mathcal{D}_{|\hat{w}^M\rangle}$ on $\{0, ..., N-1\}$ induced by sampling $|\hat{w}^M\rangle$ and interpreting integers within $M/2N$ units of i' as i, are close. The closeness of these distributions turns out to follow from Claim 4 alone, allowing us to drop the dependence of the error on the ratio M/RN. In particular, as long as R is sufficiently large, any $M > RN$ will do.

Let

$$d(i) = |\hat{v}_i|^2 \langle b_i | b_i \rangle.$$

Then d is the sub-distribution induced by measuring the (generally sub-unit length) superposition $\sum_{i<N} \hat{v}_i |b_i\rangle$ and interpreting integers within $M/2N$ units of i' as i. By Claim 4 we have

$$\left\| |\hat{w}^M\rangle - \sum_{i<N} \hat{v}_i |b_i\rangle \right\| \leq \frac{8 \log N}{\sqrt{R}}$$

from which it follows that

$$\left| \mathcal{D}_{|\hat{w}^M\rangle} - d \right| = O\left(\frac{\log N}{\sqrt{R}} \right).$$

Now,

$$\left| d - \mathcal{D}_{|\hat{v}\rangle} \right| \leq \sum_{i<N} |\hat{v}_i|^2 \left(\langle b_i | b_i \rangle - 1 \right)$$

and

$$1 - \langle b_i | b_i \rangle = \langle t_i | t_i \rangle = O\left(\frac{\log^2 N}{R} \right)$$

by applying Claim 4 to the vector $|\hat{v}\rangle = |i\rangle$. The result then follows from the triangle inequality.

9.2.3 Proof of Claim 4

Proof. In order to bound $\left\| \sum_{i<N} \hat{v}_i |t_i\rangle \right\|$ we will use the following observation which establishes that the amplitudes in $|i^M\rangle$ fall off quickly away from i'. Recall that $|t_i\rangle$ is identical to the superposition $|i^M\rangle$ except that it is missing all the amplitudes at $j \in (i')$ where (i') is the interval $(i' - \frac{M}{2N}, i' + \frac{M}{2N})$. Thus this falloff applies to the $|t_i\rangle$ as well. This Observation is closely related to Claim 3 of the previous Section and it proof is in Section 9.2.5.

Observation 2.

$$|i_j^M| = \left| \frac{1}{\sqrt{M}} \frac{1}{\sqrt{RN}} \sum_{k<RN} \omega_M^{k(j-\frac{M}{N}i)} \right| \leq \sqrt{\frac{M}{RN}} \frac{2}{\left|j - \frac{M}{N}i\right|_M}$$

where $|x|_M = \begin{cases} x \bmod M & \textit{if } 0 \leq x \bmod M \leq M/2 \\ -x \bmod M & \textit{otherwise} \end{cases}$

We now use this to bound $\left\|\sum_{i<N} \hat{v}_i|t_i\rangle\right\|$. We first note that Observation 2 can be used to show that $\||t_i\rangle\| = O\left(\frac{1}{\sqrt{R}}\right)$. A naive analysis of the quantity $\left\|\sum_{i<N} \hat{v}_i|t_i\rangle\right\|$ – see the discussion in Section 9.1 – would then give a bound of $O\left(\sqrt{\frac{N}{R}}\right)$. Instead we achieve an improved bound by a more complex version of the circulant argument of Section 9.1.3. The first equality below is by the definition of $|t_i\rangle$ and the second is by the above observation:

$$\begin{aligned} \left\| \sum_{i<N} \hat{v}_i |t_i\rangle \right\|^2 &= \sum_{j<M} \left| \sum_{i,j \notin (i')} \hat{v}_i |t_i\rangle_j \right|^2 \\ &\leq \sum_{j<M} \frac{4M}{RN} \left(\sum_{i,j \notin (i')} \frac{|\hat{v}_i|}{\left|j - \frac{M}{N}i\right|_M} \right)^2. \end{aligned}$$

This expression is almost maximized by taking the $\hat{v}_i = 1/\sqrt{N}$ for all i. In particular, the expression can be bounded by four times its value at this vector. The proof of this fact is in Section 9.2.4 and is the heart of the Theorem. It is proved by an extension of the circulant argument used in Theorem 9.

$$\left\| \sum_{i<N} \hat{v}_i |t_i\rangle \right\|^2 \leq \sum_{j<M} \frac{16M}{N^2 R} \left(\sum_{i,j \notin (i')} \frac{1}{\left|j - \frac{M}{N}i\right|_M} \right)^2. \tag{9.13}$$

Using the fact that the smallest denominator $\left|j - \frac{M}{N}i\right|_M$ is at least $\frac{M}{2N}$ and the rest are spaced out by $\frac{M}{N}$ we have

$$\sum_{i,j\notin(i')} \frac{1}{|j-\frac{M}{N}i|_M} \leq \frac{2N\log N}{M}.$$

Therefore

$$\left\| |\hat{w}^M\rangle - \sum_{i<N} \hat{v}_i|b_i\rangle \right\| = \left\| \sum_{i<N} \hat{v}_i|t_i\rangle \right\| \leq \frac{8\log N}{\sqrt{R}}, \tag{9.14}$$

as desired. □

9.2.4 Proof of Bound in Claim 4

Claim 6. *For any unit vector $|x\rangle$ and $M > 8N$*

$$\sum_{j<M} \left| \sum_{i<N, j\notin(i')} \frac{x_i}{|j-\frac{M}{N}i|_M} \right|^2 \leq \frac{4}{N} \sum_{j<M} \left(\sum_{i<N, j\notin(i')} \frac{1}{|j-\frac{M}{N}i|_M} \right)^2 \tag{9.15}$$

Proof. The left hand side of Equation (9.15) is at most the squared operator norm of the $M\times N$ matrix A with entries

$$A_{ji} = \begin{cases} 0 & \text{if } j\in(i') \\ \frac{1}{|j-\frac{M}{N}i|_M} & \text{otherwise} \end{cases}$$

Note that our matrix is positive and has the property that each row is comprised of samples of the same underlying function but with the samples shifted by $\frac{N}{M}$ from one row to the next. We argued in Section 9.1.3 that the operator norm of any positive $N\times N$ matrix with the property that each row is the shift by one $\bmod N$ of the previous row is found by applying the matrix to the unit vector with entries uniformly equal to $\frac{1}{\sqrt{N}}$.

Now, while our rectangular matrix is obviously not of this form, by reindexing and changing the denominators of the entries only slightly – so that a fixed set of integral t can be used – the expression $\|A(x)\|^2$ becomes

$$\|\hat{A}x\|^2 = \sum_{t\in\pm\frac{M}{2N}} \sum_{k<N} \left(\sum_{i<N, i\neq k} \frac{x_i}{|\frac{M}{N}k + t - \frac{M}{N}i|_M} \right)^2 .$$

Notice that the matrix giving rise to each of the double sums indexed by k and i in this new expression is $N \times N$ and has the properties discussed previously – the entries depend only on the quantity $(k-i) \mod N$. Thus each individual sum, and therefore the entire sum is maximized by choosing the entries of x_i to be equal.

Finally, we can relate $\|\hat{A}x\|^2$ to our original expression as follows: we added at most ± 1 to the denominators of our original matrix entries. Since these denominators were all larger than $\frac{M}{2N} > 4$ this at most doubled/halved the squared sums of the entries. Thus we have:

$$\frac{1}{2} \le \frac{\|\hat{A}x\|^2}{\|Ax\|^2} \le 2.$$

Finally, we use this to bound the squared operator norm of A. Let x_0 maximize $\|\hat{A}x\|^2$. Then for any x we have $\|Ax\|^2 \le 2\|\hat{A}x\|^2 \le 2\|\hat{A}x_0\|^2 \le 4\|Ax_0\|^2$. Thus our expression is bounded by four times it's value at the unit vector with entries uniformly equal to $\frac{1}{\sqrt{N}}$, as claimed. □

9.2.5 Proof of Observation 2

Proof. Recall that

$$|i^M\rangle = F_M F_{RN}^{-1} |Ri\rangle = \frac{1}{\sqrt{M}} \frac{1}{\sqrt{RN}} \sum_{j=0}^{M-1} \sum_{k=0}^{RN-1} \omega^{k(\frac{j}{M}-\frac{i}{N})} |j\rangle.$$

We have

$$|i_j^M| \quad = \quad \left| \frac{1}{\sqrt{M}} \frac{1}{\sqrt{RN}} \sum_{k=0}^{RN-1} \omega^{k(\frac{j}{M}-\frac{i}{N})} \right| = \frac{1}{\sqrt{M}} \frac{1}{\sqrt{RN}} \left| \frac{1-\omega^{RNj/M}}{1-\omega^{(\frac{j}{M}-\frac{i}{N})}} \right|$$

where the second equality applies the formula for geometric series. Then since

$$\left|1-\omega^{\left(\frac{j}{M}-\frac{i}{N}\right)}\right| = \left|1-\omega_M^{|j-\frac{M}{N}i|_M}\right| \geq \frac{1}{M}\left|j-\frac{M}{N}i\right|_M$$

and $\left|1-\omega^{RNj/M}\right| < 2$, we have

$$|i_j^M| \leq \sqrt{\frac{M}{RN}}\frac{2}{|j-\frac{M}{N}i|_M}$$

as claimed. □

9.2.6 Proof of Claim 5

Proof. We first note that to show that the restricted "bump" vectors are close, that is,

$$\left\| |b_0\rangle^{i'} - |b_i\rangle \right\| < \frac{4RN}{M}$$

it suffices to show that the corresponding full vectors $|0^M\rangle$ and $|i^M\rangle$ satisfy the same bound, that is

$$\left\| |0^M\rangle^{i'} - |i^M\rangle \right\| < \frac{4RN}{M}.$$

But recalling that $|i^M\rangle = F_M F_{RN}^{-1}|Ri\rangle$ and using the fact that F_M is unitary we have

$$\begin{aligned}\left\| |0^M\rangle^{i'} - |i^M\rangle \right\|^2 &= \left\| F_M^{-1}\left(|0^M\rangle^{i'} - |i^M\rangle\right)\right\|^2 \\ &= \sum_{j<RN}\left|\frac{1}{\sqrt{RN}}\omega_N^{-i'j} - \frac{1}{\sqrt{RN}}\omega_N^{-ij}\right|^2 \\ &\leq \sum_{j<RN}\left|\frac{1}{\sqrt{RN}}\omega_N^{-ij}\left(\omega_N^{\delta_i j}-1\right)\right|^2\end{aligned}$$

where $\delta_i = i' - \frac{M}{N}i$. But since j only goes up to RN,

$$\left|\omega_M^{\delta_i j} - 1\right| < \frac{4RN}{M}$$

and thus

$$\begin{aligned}
&\sum_{j<RN} \left| \frac{1}{\sqrt{RN}} \omega_N^{-ij} \left(\omega_N^{\delta_i j} - 1 \right) \right|^2 \\
&\qquad \leq \sum_{j<RN} \left| \frac{1}{\sqrt{RN}} \omega_N^{-ij} \right|^2 \left| \omega_N^{\delta_i j} - 1 \right|^2 \\
&\qquad \leq \left(\frac{4RN}{M} \right)^2,
\end{aligned}$$

as desired. □