Designing a Quantum Circuit for Modeling the Heat Equation

A thesis

submitted by

NICOLAS ANZALONE

In partial fulfillment of the requirements
for the degree of

Master of Sciences

in

Mathematics

TUFTS UNIVERSITY

April, 2005

Advisor: Bruce M. Boghosian

Abstract

Computer simulations of certain natural phenomena are computationally expensive on a classical computer. The movement of fluids is one example of a dynamical system we would like to simulate on a computer but whose intractability makes it impracticable for strongly turbulent flow. Relatively recently, mathematicians and physicists have come to believe that certain problems which may be "intractable" on a computer which obeys the macroscopic classical laws of physics may be much easier on a computer which obeys the microscopic laws of quantum physics. To this end, Quantum Computing has become an extremely fast growing area of active research. Some are trying to find practical ways to implement a Quantum Computer on a "larger" scale. Others assume that some day we will be able to build a quantum computer and instead prefer to investigate what types of classically intractable problems might be solvable on a quantum computer. The equations governing certain fluid behaviors are similar in nature (albeit more complicated) then the diffusion equation for heat. It is plausible that quantum algorithms for heat diffusion might be generalized to the more complicated case of fluid dynamics. In this paper, I present a quantum circuit which implements a discrete model for the heat equation in one dimension.

Acknowledgements

I wish to thank my advisor Bruce Boghosian for an extremely engaging year of mathematics. My meetings with him have been the highlight of every week. I have learned a tremendous amount and have had much fun in the process. Without his guidance and efforts this paper would not have been possible. Much credit also goes to Marjorie Hahn and Cristoph Borgers, who have managed to find time in their busy schedules to serve on my committee.

I would also like to thank all of my fellow graduate students. For me, mathematics is as much a social endeavor as an intellectual one. Having such a wonderful group of people as classmates has made the experience that much more fun. In particular I would like to thank Aleksei Beltukov, Gianluca Caterina, Matthew Enlow, Brian Collins, Mark O'Brien, Arthur Weiss, Wei Zhang, Erin Munro, Rob Munger, Mathew Goodman, Lucas Finn, Malena Espanol, and Alex Shaller. Thanks to all of my instructors, including Richard Weiss, Loring Tu, Boris Hasselblatt, Monica Moreno Rocha, Zbigniew Nitecki, and Fulton Gonzalez. Special thanks also goes to Peter Love, Kim Ruane, Mary Glaser, Linda Garant, Gail Kaufmann, Susan Cleary and Sarah Schnable. All of you have helped in one way or another to shape my graduate education. Also, my undergraduate mentors deserve credit for my conversion to Mathematics from Computer Science. In particular, thanks to Ethan Bolker, Joan Lukas, Carl Offner, and Patrick O'Neil for their encouragement and support.

Thanks also goes to my family and friends, who have been enormously supportive and encouraging. Special thanks goes to my mother for letting me stay in her house in Arlington

during the week for two years while in school!

Finally I would like to thank my wife Heather Alger. The past two years would not have been possible without her love, patience, and support. She believed in me before I believed in myself, and she has made significant sacrifices while I pursued my educational dreams.

Contents

List of Figures

[1] For a derivation of the output of the Hadamard gate, see Section 1.2 on page 7

Designing a Quantum Circuit for Modeling the Heat Equation

Nicolas Anzalone

April, 2005

Chapter 0

Introduction

In this paper, we will address primarily a simplified model involving the diffusion of heat in one dimension. This model has received a fair amount of attention from Quantum Computation circles recently [7], [9]. After showing how a discrete model behaves asymptotically like the heat equation in one dimension, we will discuss how to design a quantum circuit which implements this discrete model.

In [9], Cory develops a quantum circuit to solve the diffusion of heat in one dimension. This circuit uses two qubits for every possible site location on a discrete one-dimensional lattice. Inspired by [7], we develop a circuit using only $\mathcal{O}(n)$ qubits for every 2^n site locations to gain significant circuit efficiency. Hence, the calculation scales logarithmically in the number of positions which the particle can occupy in our discrete model. This represents an enormous improvement over the algorithmic complexity of current classical fluid simulation. Because we store 2^n positions using only n qubits, it is not unrealistic that with a quantum computer we could perform a simulation using 2^{100} potential particle sites (since this would only require on the order of 100 qubits). This is not something that is currently possible with a classical computer.

It is interesting to note that, when run for t time steps, the quantum circuit discussed in

this paper actually has encoded within it the exact probability distribution of the particle position after t steps through the discrete model. However the fundamental properties of a quantum system prevent us from extracting the probability distribution easily. We discuss what information it is possible to extract using one currently known method.

The basic circuit developed in this paper is scheduled to be implemented on a small scale at MIT by Tufts undergraduate student Troy Borneman for a Senior Project. This paper will also address some of the complications of implementing even a small scale version of this circuit using current Nuclear Magnetic Resonance (NMR) technology, and ways that we optimized the circuit so that we could test a meaningful implementation of the theory behind it.

The diffusion of heat is well understood. However we hope the model presented can be generalized so that we might use it to study the behavior of more complex systems of differential equations. In particular, we hope that a model similar to the one presented in this paper might enable us to simulate the behavior of certain fluid dynamical systems.

It should be noted that [6] and [8] were both invaluable resources while preparing this paper. While most of the specific citations are from other texts, these books were critical in the development of a general background in the field of Quantum Computation.

Chapter 1

Background

This chapter may be skipped if the reader already has a good understanding of basic Quantum Computation principles and notations. Quantum Computation is a large and fast growing field. Here we simply introduce concepts and notation which are relevant to this paper. The content in this chapter is taken from [6] and [8], and I direct the reader to these references if they would like further reading on the subject. Both are excellent books.

1.1 Basic Definitions and Notation

We start with some basic definitions and notation that are probably familiar to most readers.

Definition 1. *Let $\alpha = x + iy$ be a complex number. Then $\alpha^* = x - iy$ is defined as the complex conjugate. Suppose that U is a matrix with complex components $\{\alpha_{ij}\}$. Then U^* is defined as the matrix whose $(i,j)^{th}$ component β_{ij} is given by the complex conjugate of the corresponding component in U. That is, $\beta_{ij} = \alpha_{ij}^*$.*

Definition 2. *Let U be a matrix. Then U^T is the transpose matrix of U.*

Definition 3. *Let the m by n matrix U have complex entries α_{ij} such that*

$$U = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n} \\ \vdots & \ddots & & \vdots \\ \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mn} \end{pmatrix}.$$

Then the operation † is called Hermitian conjugation, and is defined by $U^\dagger = (U^T)^*$. Hence,

$$U^\dagger = \begin{pmatrix} \alpha_{11}^* & \alpha_{21}^* & \cdots & \alpha_{n1}^* \\ \alpha_{12}^* & \alpha_{22}^* & \cdots & \alpha_{n2}^* \\ \vdots & \ddots & & \vdots \\ \alpha_{1m}^* & \alpha_{2m}^* & \cdots & \alpha_{nm}^* \end{pmatrix}.$$

The resulting matrix $U^\dagger$ is also sometimes referred to as the adjoint *matrix of U.*

Definition 4. *Let U be a matrix. If $U^\dagger = U$, U is called a* Hermitian *or* self-adjoint *matrix.*

Definition 5. *A* qubit *is a pair of complex numbers (α, β) such that $|\alpha|^2 + |\beta|^2 = 1$.*

Remark 1. *The qubit is the quantum analogy of a classical computer bit. In a classical computer, a bit is a component which can assume either the value 0 or the value 1. In a quantum computer, the qubit can be in state $|0\rangle$ or state $|1\rangle$. We will define these states more rigorously later, but for now it is sufficient to know that while a classical bit is said to have a value, a qubit is said to be in a particular state. When a qubit in the state $|0\rangle$ or $|1\rangle$ is* measured*, we observe a value of 0 or 1 respectively. A qubit in the state $|0\rangle$ or $|1\rangle$ is said to be in a* classical state*.*

Definition 6. *A complex* superposition *of two classical states is a complex linear combination of the two states.*

In addition to the two classical states, a qubit can also assume a complex superposition of them. We will further examine the nature of superpositions shortly.

Definition 7. *Let U be a matrix. If $UU^\dagger = I$, U is called a unitary matrix. Quantum operations are often represented as matrices, and a quantum operation whose matrix is unitary is referred to as a* unitary operation.

Unitary matrices are a ubiquitous concept in Quantum Computation. They are extremely important because it is postulated that all quantum mechanical operations may be represented by the action of a unitary matrix on a state vector. We end with a definition of the Hadamard gate and a description of some other basic logical gates used in circuit diagrams. The Hadamard is a very basic quantum gate which is used frequently in later sections.

Definition 8. *Let H be the Hadamard gate. H is defined as*

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1.1}$$

Another basic logical gate is the `not` gate.

Definition 9. *The* `not` *gate negates the value of a given qubit. The matrix identified with the* `not` *gate is*

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{1.2}$$

Remark 2. *A* `not` *applied to $|0\rangle$ will yield $|1\rangle$, and similarly a* `not` *applied to $|1\rangle$ gives $|0\rangle$. In a circuit diagram, the* `not` *gate is indicated by an open circle with a cross inside.*

A `not` gate can be *controlled* by the conditional value of a different qubit.

Definition 10. *A* controlled `not` *gate, or* `cnot` *gate takes as input both a control qubit and a target qubit. Its behavior is defined as implementing a* `not` *gate on the target qubit only*

when the control qubit is in state $|1\rangle$.

Remark 3. *The* `cnot` *gate is represented in a circuit diagram by a filled circle on the controlling qubit, with a line to a crossed circle on the target qubit. It is also possible to apply a* `not` *conditional upon the controlling qubit being in the state* $|0\rangle$ *rather than* $|1\rangle$. *In a circuit diagram this is indicated with an open circle on the controlling qubit. For clarity, we omit the matrix representation of the remaining gates.*

Definition 11. *A* Toffoli *gate will* `not` *the value of a target qubit only if the two controlling qubits both have value* 1.

Remark 4. *In a circuit diagram, the Toffoli gate looks just like the* `cnot` *gate with an extra controlling filled circle. Pictures of all the above gates and what they do are in Figure 1.1.*

After introducing the concepts in the next two sections, we will give some examples of how the Hadamard gate is used in a quantum circuit. Specifically, the output of the Hadamard gate shown in Figure 1.1 will be explained in Section 1.3. All of the above-mentioned logical gates may be represented by unitary matrices.

1.2 State Vectors: Bra and Ket

There are two ways traditionally used to represent a quantum state. The one used through most of this paper, and perhaps the more intuitive of the two, uses the physicists' notation of *bra* (e.g. $\langle\psi|$) and *ket* (e.g. $|\psi\rangle$). (The second representation of quantum states uses density matrices, which we will not need in this paper.) From a mathematical perspective, *bra* and *ket* are best introduced as being the Hermitian conjugates of each other, where *ket* is a column vector and *bra* is a row vector. From the definitions in Section 1.1 we can write

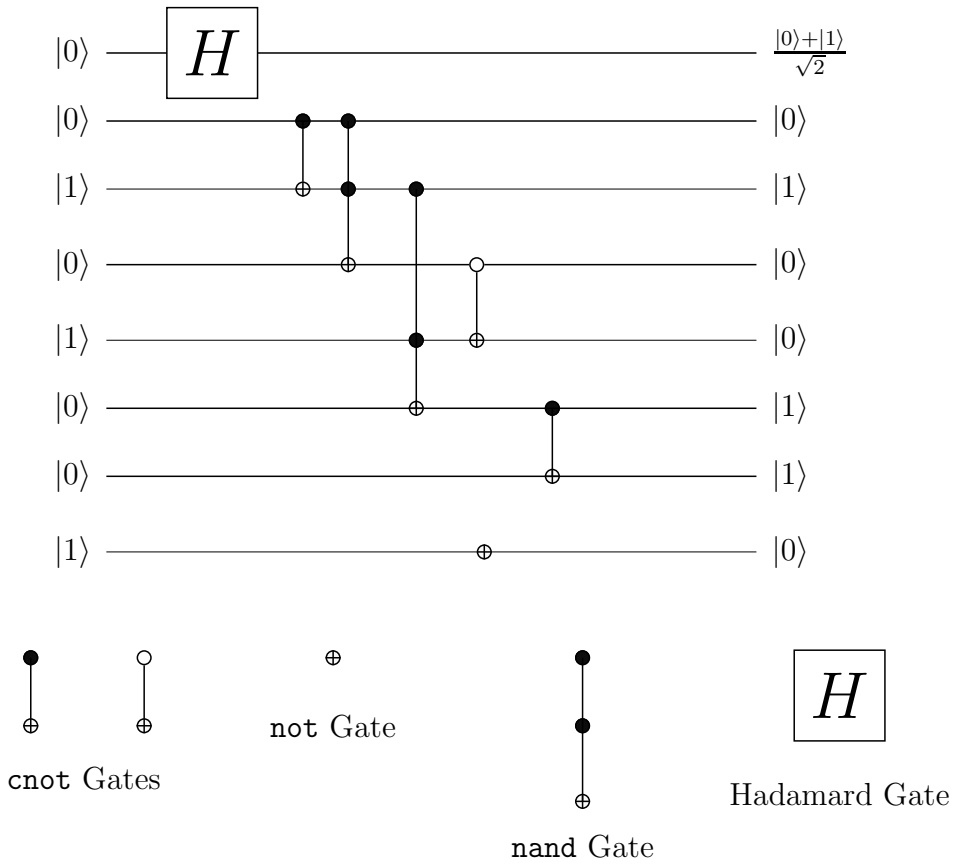

Figure 1.1: Some examples of logical gates [1]

this as $\langle\psi| = |\psi\rangle^\dagger$. For example, let $\{\alpha_i\}$ be a sequence of 2^n complex numbers. Then

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n} \end{pmatrix}, \tag{1.3}$$

$$\tag{1.4}$$

while

$$\langle\psi| = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_{2^n}^*)\,. \tag{1.5}$$

The inner product of these two is written $\langle\psi|\psi\rangle$ and is computed in the usual manner:

$$\begin{aligned} \langle\psi|\psi\rangle &= \sum_{i=1}^{2^n} \alpha_i \alpha_i^* \\ &= \sum_{i=1}^{2^n} |\alpha_i|^2. \end{aligned}$$

Just as with any complex vector, we can think of $|\psi\rangle$ as a sum of basis states multiplied by complex constants. Because it makes manipulating the vectors in a circuit diagram easier, it is typical to represent these basis states as binary numbers. This will become more clear later on, but here is an illustration of the notation:

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n} \end{pmatrix} \qquad = \alpha_1 |\underbrace{000\cdots00}_{n}\rangle + \alpha_2 |000\cdots01\rangle + \cdots + \alpha_{2^n} |111\cdots11\rangle. \tag{1.6}$$

The sum 1.6 is an example of a superposition of the basis states. Suppose we have a quantum system consisting of a superposition of basis states. If we measure the system and find that it is in a particular basis state, we have *collapsed* the superposition onto this single state. Hence we can think of these basis states as the set of all possible observable outcomes following measurement. It is very important to note that this process of measurement is nonunitary. There is a *Measurement Postulate* which states that for $0 \leq x < 2^n$, the probability of observing the basis state $|x\rangle$ is given by $|\alpha_x|^2$.

When two quantum systems interact with each other, the state of the combined system is represented by taking the tensor product of the two component systems. For example, if $|\psi\rangle$ and $|\phi\rangle$ are two quantum systems such that

$$|\psi\rangle = \alpha_1|0\rangle + \alpha_2|1\rangle, \tag{1.7}$$

$$|\phi\rangle = \beta_1|0\rangle + \beta_2|1\rangle \tag{1.8}$$

then the total state of the combined system is given by

$$|\psi\rangle \otimes |\phi\rangle = \alpha_1\beta_1|0\rangle|0\rangle + \alpha_1\beta_2|0\rangle|1\rangle + \alpha_2\beta_1|1\rangle|0\rangle + \alpha_2\beta_2|1\rangle|1\rangle \tag{1.9}$$

which can also be written as

$$= \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle + \alpha_2\beta_1|10\rangle + \alpha_2\beta_2|11\rangle. \tag{1.10}$$

Remark 5. *Occasionally, two tensored basis states such as the ones in Equation 1.9 are left in that form. This is usually done to emphasize that the two tensored basis states represent specific and distinct attributes or components of a circuit.*

1.3 Quantum Circuits and Quantum Evolution

Although it is convenient to manipulate circuit diagrams using the notation introduced in Section 1.2, quantum operations such as the Hadamard gate (Definition 8) act on a more typical definition of basis vectors. This can be a little confusing since we are using strings of 0's and 1's in both cases. Consider a qubit $|\psi\rangle$. For $\alpha, \beta \in \mathbb{C}$ (recall that a qubit is a pair of complex numbers),

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{1.11}$$

If we want to consider Equation 1.11 in terms of conventional vectors, then we can identify $|0\rangle$ with the first component of a vector and $|1\rangle$ with the second. This gives

$$|\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{1.12}$$

In fact, we more or less need to use the form introduced in equation 1.12 if we want to consider what happens to $|\psi\rangle$ when we act on it with the Hadamard gate. Acting on a qubit or quantum state with a quantum gate is sometimes also referred to as *evolving* the quantum state. Evolution is attained by simply multiplying the gate on the right with the

state vector. Let $|\psi'\rangle$ be the value of $|\psi\rangle$ after evolution with the Hadamard gate. Then

$$|\psi'\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} |\psi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} (\alpha|0\rangle + \beta|1\rangle) \tag{1.13}$$

$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \left[\alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] \tag{1.14}$$

$$= \frac{\alpha}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{\beta}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.15}$$

$$= \frac{\alpha}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{\beta}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{1.16}$$

$$= \frac{\alpha}{\sqrt{2}} (|0\rangle + |1\rangle) + \frac{\beta}{\sqrt{2}} (|0\rangle - |1\rangle) \tag{1.17}$$

$$= \frac{\alpha + \beta}{\sqrt{2}} |0\rangle + \frac{\alpha - \beta}{\sqrt{2}} |1\rangle \tag{1.18}$$

If $|\psi\rangle$ is initialized to $|0\rangle$, then $\beta = 0, \alpha = 1$ and Equation 1.18 simplifies to $\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$. Similarly, if $|\psi\rangle$ is initialized to $|1\rangle$, then $\alpha = 0, \beta = 1$ and Equation 1.18 simplifies to $\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$.

Remark 6. *These simple substitution rules are used extensively in Chapter 3. Note that in the event that $|\psi\rangle = |0\rangle$, this shows that applying a Hadamard gate has the effect of putting $|\psi\rangle$ into a superposition of $|0\rangle$ and $|1\rangle$, such that upon measurement we have an equal probability of observing 0 or 1. This makes the Hadamard gate an ideal candidate for assisting us in developing a circuit to implement a random walk where a particle moves either left or right, each with probability $\frac{1}{2}$.*

Chapter 2

Discrete Model for the Heat Equation in One Dimension

A good physical description of what is modeled by the one dimensional heat equation is given in [5] on page 629. We can think of this model as a metal rod through which heat moves only towards the ends of the rod (i.e. the rod is insulated such that no heat is lost through the cylindrical surface). Let u be the temperature of the rod at a given position x at time t. Then for some constant k,

$$\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2} \tag{2.1}$$

describes the movement of heat.

It turns out that there is a discrete model which can be proven to behave asymptotically like this equation. In order to prove it analytically, however, we must start with the discrete model and work backwards. This correspondence has been known since classical times - it was probably known to deMoivre - but the derivation used here follows that in the notes of Professor Bruce Boghosian at Tufts University [3].

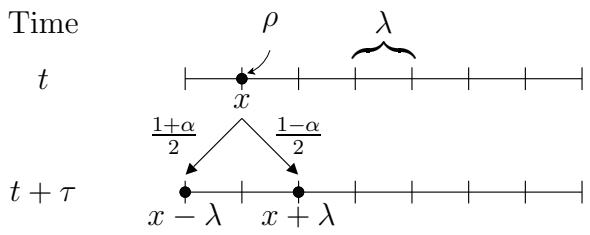

Figure 2.1: A random walk in one dimension

2.1 Random Walk

Consider the discrete random walk of a particle ρ along a line with a finite number of equally spaced potential locations for ρ at any particular time. Let $x = \lambda_j, j \in \mathbb{Z}$, be the position of ρ at time t. Suppose that after every fixed time interval τ, ρ moves left a fixed number λ of units with probability $\frac{1+\alpha}{2}$, and ρ moves right λ units with probability $\frac{1-\alpha}{2}$ (see Figure 2.1). Let $P(x, t+\tau)$ be the probability that ρ is in position x at time $t+\tau$. Then

$$P(x, t+\tau) = \left(\frac{1-\alpha}{2}\right) P(x-\lambda, t) + \left(\frac{1+\alpha}{2}\right) P(x+\lambda, t). \tag{2.2}$$

In other words, the probability that ρ is in position x at time $t+\tau$ is equal to the probability that (at time t) ρ was in position $x-\lambda$ and moved right plus the probability that ρ was in position $x+\lambda$ and moved left.

Fixing x and using Taylor's formula with remainder [1], $\exists T \in [t, t+\tau]$ such that

$$\begin{aligned} P(x, t+\tau) &= P(x,t) + \frac{\partial P}{\partial t}(x,t)(t+\tau-t) + \frac{\partial^2 P}{\partial t^2}(x,T)\frac{(t+\tau-t)^2}{2!} && (2.3) \\ &= P(x,t) + \tau\frac{\partial P}{\partial t}(x,t) + \frac{\tau^2}{2}\frac{\partial^2 P}{\partial t^2}(x,T). && (2.4) \end{aligned}$$

Similarly, if we fix t then $\exists X_- \in [x, x-\lambda]$ and $\exists X_+ \in [x, x+\lambda]$ such that

$$\left(\frac{1-\alpha}{2}\right) P(x-\lambda,t) = \left(\frac{1-\alpha}{2}\right)\left[P(x,t)+\frac{\partial P}{\partial x}(x,t)(x-\lambda-x)+\frac{\partial^2 P}{\partial x^2}(x,t)\frac{(x-\lambda-x)^2}{2!}+ \frac{\partial^3 P}{\partial x^3}(X_-,t)\frac{(x-\lambda-x)^3}{3!}\right] \quad (2.5)$$

$$= \left(\frac{1-\alpha}{2}\right)\left[P(x,t)-\lambda\frac{\partial P}{\partial x}(x,t)+\frac{\lambda^2}{2}\frac{\partial^2 P}{\partial x^2}(x,t)-\frac{\lambda^3}{6}\frac{\partial^3 P}{\partial x^3}(X_-,t)\right] \quad (2.6)$$

and

$$\left(\frac{1+\alpha}{2}\right) P(x-\lambda,t) = \left(\frac{1+\alpha}{2}\right)\left[P(x,t)+\frac{\partial P}{\partial x}(x,t)(x+\lambda-x)+\frac{\partial^2 P}{\partial x^2}(x,t)\frac{(x+\lambda-x)^2}{2!}+ \frac{\partial^3 P}{\partial x^3}(X_+,t)\frac{(x+\lambda-x)^3}{3!}\right] \quad (2.7)$$

$$= \left(\frac{1+\alpha}{2}\right)\left[P(x,t)+\lambda\frac{\partial P}{\partial x}(x,t)+\frac{\lambda^2}{2}\frac{\partial^2 P}{\partial x^2}(x,t)+\frac{\lambda^3}{6}\frac{\partial^3 P}{\partial x^3}(X_+,t)\right] \quad (2.8)$$

Substituting Equations 2.3, 2.5, and 2.7 into Equation 2.2 yields

$$P(x,t)+\tau\frac{\partial P}{\partial t}(x,t)+\frac{\tau^2}{2}\frac{\partial^2 P}{\partial t^2}(x,T) = \left(\frac{1-\alpha}{2}\right)\left[P(x,t)-\lambda\frac{\partial P}{\partial x}(x,t)+\frac{\lambda^2}{2}\frac{\partial^2 P}{\partial x^2}(x,t)- \frac{\lambda^3}{6}\frac{\partial^3 P}{\partial x^3}(X_-,t)\right]+ \left(\frac{1+\alpha}{2}\right)\left[P(x,t)+\lambda\frac{\partial P}{\partial x}(x,t)+\frac{\lambda^2}{2}\frac{\partial^2 P}{\partial x^2}(x,t)+ \frac{\lambda^3}{6}\frac{\partial^3 P}{\partial x^3}(X_+,t)\right] \quad (2.9)$$

$$= P(x,t)+\alpha\lambda\frac{\partial P}{\partial x}(x,t)+\frac{\lambda^2}{2}\frac{\partial^2 P}{\partial x^2}(x,t)+ \frac{\lambda^3}{6}\left[\left(\frac{1+\alpha}{2}\right)\frac{\partial^3 P}{\partial x^3}(X_+,t)-\left(\frac{1-\alpha}{2}\right)\frac{\partial^3 P}{\partial x^3}(X_-,t)\right], \quad (2.10)$$

or

$$\begin{aligned}\tau\frac{\partial P}{\partial t}(x,t)+\frac{\tau^2}{2}\frac{\partial^2 P}{\partial t^2}(x,T) &= \alpha\lambda\frac{\partial P}{\partial x}(x,t)+\frac{\lambda^2}{2}\frac{\partial^2 P}{\partial x^2}(x,t)+\\ &\quad \frac{\lambda^3}{6}\left[\left(\frac{1+\alpha}{2}\right)\frac{\partial^3 P}{\partial x^3}(X_+,t)-\left(\frac{1-\alpha}{2}\right)\frac{\partial^3 P}{\partial x^3}(X_-,t)\right].\end{aligned} \tag{2.11}$$

For the purposes of this paper, we will use as our primary example a random walk where the particle moves left or right with equal probability, which is the probability needed to simulate the heat equation. That is, $\alpha = 0$. This simplifies Equation 2.11 to

$$\tau\frac{\partial P}{\partial t}(x,t)+\frac{\tau^2}{2}\frac{\partial^2 P}{\partial t^2}(x,T)=\frac{\lambda^2}{2}\frac{\partial^2 P}{\partial x^2}(x,t)+\frac{\lambda^3}{12}\left[\frac{\partial^3 P}{\partial x^3}(X_+,t)-\frac{\partial^3 P}{\partial x^3}(X_-,t)\right] \tag{2.12}$$

$$\frac{\partial P}{\partial t}(x,t)+\frac{\tau}{2}\frac{\partial^2 P}{\partial t^2}(x,T)=\frac{\lambda^2}{2\tau}\frac{\partial^2 P}{\partial x^2}(x,t)+\frac{\lambda^3}{12\tau}\left[\frac{\partial^3 P}{\partial x^3}(X_+,t)-\frac{\partial^3 P}{\partial x^3}(X_-,t)\right] \tag{2.13}$$

In order to show that our discrete model behaves like the heat equation asymptotically, we need to take the limit of this equation as the value τ of our time steps and the distance λ which the particle ρ moves at every time step tend toward zero. We are free to choose how quickly they tend toward zero with respect to each other. In particular, if we let τ decrease more quickly than λ such that as they both tend toward zero, $\lim_{\tau,\lambda\to 0}\frac{\lambda^2}{2\tau} = k$, for some constant k the second terms on both sides of 2.13 will tend to zero. Hence if we let $\tau \sim \lambda^2$, the limit of Equation 2.13 is given by

$$\lim_{\tau,\lambda\to 0}\left(\frac{\partial P}{\partial t}(x,t)+\frac{\tau}{2}\frac{\partial^2 P}{\partial t^2}(x,T)\right)=\lim_{\tau,\lambda\to 0}\left(\frac{\lambda^2}{2\tau}\frac{\partial^2 P}{\partial x^2}(x,t)+\frac{\lambda^3}{12\tau}\left[\frac{\partial^3 P}{\partial x^3}(X_+,t)-\frac{\partial^3 P}{\partial x^3}(X_-,t)\right]\right), \tag{2.14}$$

or

$$\lim_{\lambda\to 0}\left(\frac{\partial P}{\partial t}(x,t)+\frac{\lambda^2}{2}\frac{\partial^2 P}{\partial t^2}(x,T)\right)=\lim_{\lambda\to 0}\left(\frac{\lambda^2}{2\frac{1}{2k}\lambda^2}\frac{\partial^2 P}{\partial x^2}(x,t)+\frac{\lambda^3}{12\frac{1}{2k}\lambda^2}\left[\frac{\partial^3 P}{\partial x^3}(X_+,t)-\frac{\partial^3 P}{\partial x^3}(X_-,t)\right]\right), \tag{2.15}$$

or

$$\frac{\partial P}{\partial t}(x,t)=k\frac{\partial^2 P}{\partial x^2}(x,t), \tag{2.16}$$

which is in the form of the heat equation.

Remark 7. *Although our primary example in this paper is with a random walk in which the particle moves left or right with equal probability, this assumption is not necessary to prove that it will behave asymptotically like an important equation. It turns out that depending on how we scale α, τ, and λ relative to each other, we can show that this random walk will behave asymptotically like a variety of differential equations. We are not limited to this type of random walk either. The quantum circuit presented in this paper can be modified to simulate a random walk where the particle moves left or right or stays stationary at each time step. In fact many different types of random walks are possible with relatively minor changes to the circuit presented in the next chapter.*

Chapter 3

Designing a Quantum Circuit

We have established a discrete model and proved that it behaves asymptotically like the heat equation. Now we can discuss ways to implement it on a quantum computer. One natural way to begin thinking about implementing a random walk on a computer is to encode the position of the particle on a line by a number in a binary register. Moving the particle λ units to the right corresponds with adding one to the register. Moving the particle λ units to the left corresponds with subtracting one from the register. To avoid having the particle walk off the edge of the our line (since a binary register must necessarily have finite length), we give the line periodic boundary conditions. That is, if the particle is on the leftmost edge of the line at time t, and hops λ units to the left at time $t + \tau$, then the particle appears on the rightmost position of the line (see Figure 3.1). Periodic boundary conditions work well with the register model, since binary addition on a register of size 2^n is generally performed modulo 2^n.

This model is also easily generalized. In particular, if we wished to create a model where the particle hopped $k\lambda$ units at each time step, then we could add or subtract k to the register and the model still performs correctly. It can also be proved using an argument similar to the one in Chapter 2 that a random walk in two dimensions behaves asymptotically like the two

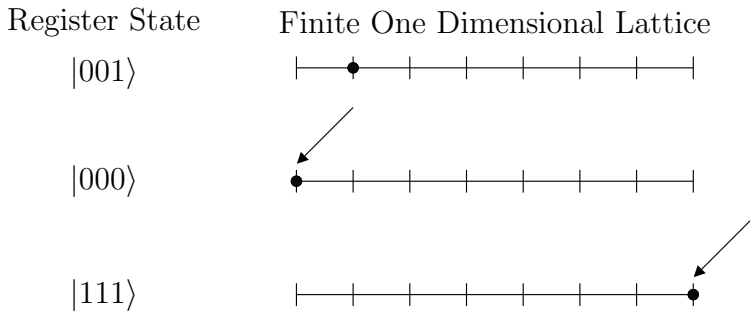

Figure 3.1: Subtracting two from a three-qubit register

dimensional heat equation (analogous to an infinitely thin sheet of material whose top and bottom are insulated so that heat can diffuse only in the plane of the material). It is then easy to modify our approach only slightly to accommodate this new model. In particular, we create two binary registers. One tracks the x-coordinate of the particle, the other the y-coordinate. Then we can hop in a particular direction on the grid by adding appropriate numbers to the x and/or y registers.

Remark 8. *We can use this general model for a random n-dimensional walk by simply using one register for each dimension. In particular, we will show later that our circuit grows only logarithmically in complexity with respect to the number of sites on our lattice. Since increasing the dimension only requires a linear increase in the circuit complexity, this means that the complexity remains logarithmic with respect to the number of site positions as our dimension increases!*

3.1 Quantum Addition

Next we need to design a quantum circuit to implement this random walk algorithm. One design of a quantum circuit which adds two three-qubit binary numbers (taken from [2]) is shown in Figure 3.2. In this picture, $|\phi\rangle$ and $|\psi\rangle$ are the two registers that we wish to add. The sum is placed in $|\psi\rangle$ and overwrites the original value (we emphasize this by

using the notation $|\psi'\rangle$ to indicate that the state may have changed). The kets $|\psi_i\rangle$ and $|\phi_i\rangle$ represent the i^{th} qubit of $|\psi\rangle$ and $|\phi\rangle$ respectively. The $|anc_i\rangle$ are ancillary qubits used as carry registers. They are initialized to $|0\rangle$ and this circuit returns them to that value after the carry information is no longer needed. Note that this design can be generalized very easily to addition registers of any fixed size n. The first and last qubits are somewhat exceptional since there is no incoming carry information for the first qubit and no outgoing carry information for the last qubit. The idea for any number of qubits in the middle can be seen by looking at the behavior of the middle triple of qubits $|\phi_2\rangle$, $|\psi_2\rangle$, and $|anc_2\rangle$ in Figure 3.2. In order to speak more generally, however, let us refer to them as the i^{th} triple, or $|\phi_i\rangle$, $|\psi_i\rangle$, and $|anc_i\rangle$, with the understanding that our model uses $i = 2$ in Figure 3.2.

To begin, we assume that the carry qubit from the previous triple of qubits is set properly. If $|\phi_i\rangle$ and $|\psi_i\rangle$ are both in state $|1\rangle$, then we set the carry qubit $|anc_i\rangle$ to state $|1\rangle$ also. Next, if $|\phi_i\rangle = |1\rangle$, we add it to $|\psi_i\rangle$ using a logical *not*. Note that we have already saved carry information in $|anc_i\rangle$ so that this operation will be correct regardless of the state of $|\psi_i\rangle$. At this point if $|\psi_i\rangle = |0\rangle$, then we will not need to carry anything additional to the next triple of qubits. If on the other hand $|\psi_i\rangle = |1\rangle$, then exactly one of $|\phi_i\rangle$ and $|\psi_i\rangle$ was initially in state $|0\rangle$. So $|anc_i\rangle$ will still be $|0\rangle$. But in this case we will need to store carry information for the following triple if the carry qubit from the previous triple (in this case $|anc_{i-1}\rangle$) in in state $|1\rangle$. Hence we logically `not` $|anc_{i+1}\rangle$ if both $|anc_{i-1}\rangle = |1\rangle$ and $|\psi_i\rangle = |1\rangle$.

At this time we have not irretrievably lost any information about $|anc_i\rangle$. It is still possible to undo everything we have done to $|anc_i\rangle$ in the previous paragraph. This will change as soon as we add the carry qubit from the previous triple (i.e., $|anc_{i-1}\rangle$) to $|\psi_i\rangle$). We have properly set the value for $|anc_{i+1}\rangle$, however, which means that we have everything we need to proceed with the next triple. Also, we would like (if possible) to reset $|anc_i\rangle$ to $|0\rangle$ to clean things up. For this reason, we postpone adding $|anc_{i-1}\rangle$ to $|\psi_i\rangle$, and instead continue to add all subsequent triples of qubits before proceeding with the current triple.

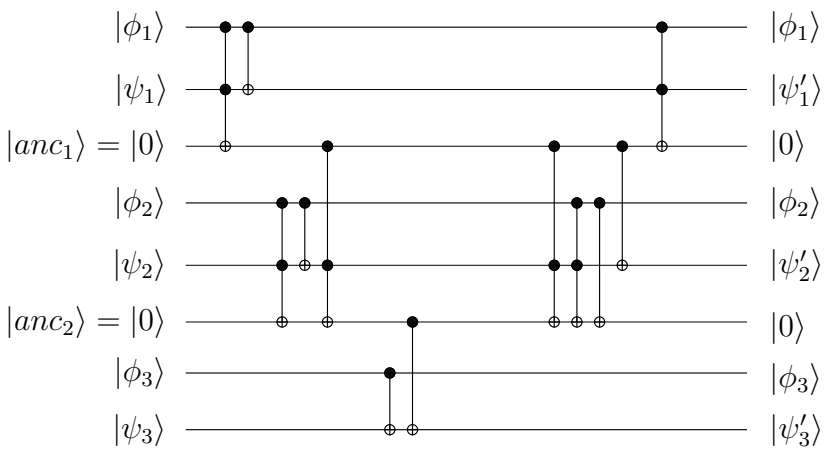

Figure 3.2: A three-qubit adder

Since each subsequent triple will be executed in a like manner, let us now assume that all subsequent triples have been correctly added together. Note that this circuit does not save carry information from the last qubits (in other words, the operation is performed modulo 2^n, where n is the number of qubits in the registers we are adding together). First we simply undo the last operation performed on the $|anc_i\rangle$ qubit by flipping it in the event that both $|anc_{i-1}\rangle = |1\rangle$ and $|\psi_i\rangle = |1\rangle$. In order to return $|anc_i\rangle$ to its original value, we need only flip it in the event that $|\phi_i\rangle = |1\rangle$ and $|\psi_i\rangle = |0\rangle$. Because we have already added the two together and carried if necessary, the only way this state could have been obtained is if both qubits were originally in state $|1\rangle$ in which case we would have had to carry. So assume that $|\psi_i\rangle = |0\rangle$, and flip $|anc_i\rangle$ in the event that $|\phi_i\rangle = |1\rangle$. If our assumption is false, then before incorrectly flipping $|anc_i\rangle$ we would have necessarily had $|\phi_i\rangle = |\psi_i\rangle = |1\rangle$. Hence if this is true, we flip $|anc_i\rangle$ once more to return it to the correct state. These last two operations are logically commutative, and in Figure 3.2 they appear in the reverse order in which I explained their functions. Finally, $|anc_i\rangle$ has been returned to its original state $|0\rangle$, and we add together $|anc_{i-1}\rangle + |\psi_i\rangle$ by applying a logical `not` to $|\psi_i\rangle$ in the event that $|anc_{i-1}\rangle = |1\rangle$ before proceeding to the $(i-1)^{th}$ triple.

3.2 A Quantum Circuit for a Classical Random Walk

Given the general adder described in Section 3.1, designing a circuit to add or subtract one is relatively easy. Note that $-1 = 0 - 1$ in binary is represented by a register filled with 1's. Hence subtracting one is achieved by adding $2^n - 1$, where n is the number of bits in the register. Using this we can implement our quantum circuit by assigning one n-bit register to keep track of the particle position (call this $|\psi\rangle$), and another to hold the value to be added to the first register (call this $|\phi\rangle$) to obtain the particle position at the next time step. We use a control bit to decide which we add. If the control bit is in state $|0\rangle$, then we add one. If the control bit is in state $|1\rangle$, then we logically `not` every bit except the least significant bit (which is already one), and add the result, which has the effect of subtracting one.

If we initialize the control bit to state $|0\rangle$ and pass it through a Hadamard gate, it will remain $|0\rangle$ or become $|1\rangle$ with equal probability (see the end of Section 1.3 to review why this is so). We can then use this control qubit's superposition to conditionally `not` every bit of the register $|\phi\rangle$ except the least significant bit. This gives us a good basic circuit for each time step, a detailed picture of which is in Figure 3.3. In general, using Figure 3.3, we can see that this requires $n + 2(n-1) + 1$ qubits plus the number of gates. We have $2(n-1)$ gates to switch from adding one to subtracting one, plus $3(n-2) + 3 = 3(n-1)$ gates descending, plus $4(n-2) + 2$ gates ascending plus one Hadamard gate. This yields a total circuit complexity on the order of $12n - 7$.

The circuit in Figure 3.4 implements the quantum adder described above on three qubits, or eight lattice positions. It adds one or subtracts one to/from a three-qubit register, each with probability $\frac{1}{2}$. The circuit in Figure 3.5 implements the same adder which adds/subtracts one (each with probability $\frac{1}{4}$) or adds zero (with probability $\frac{1}{2}$). Note that the organization of these circuits has been modified to clarify what is input to and output from the circuit.

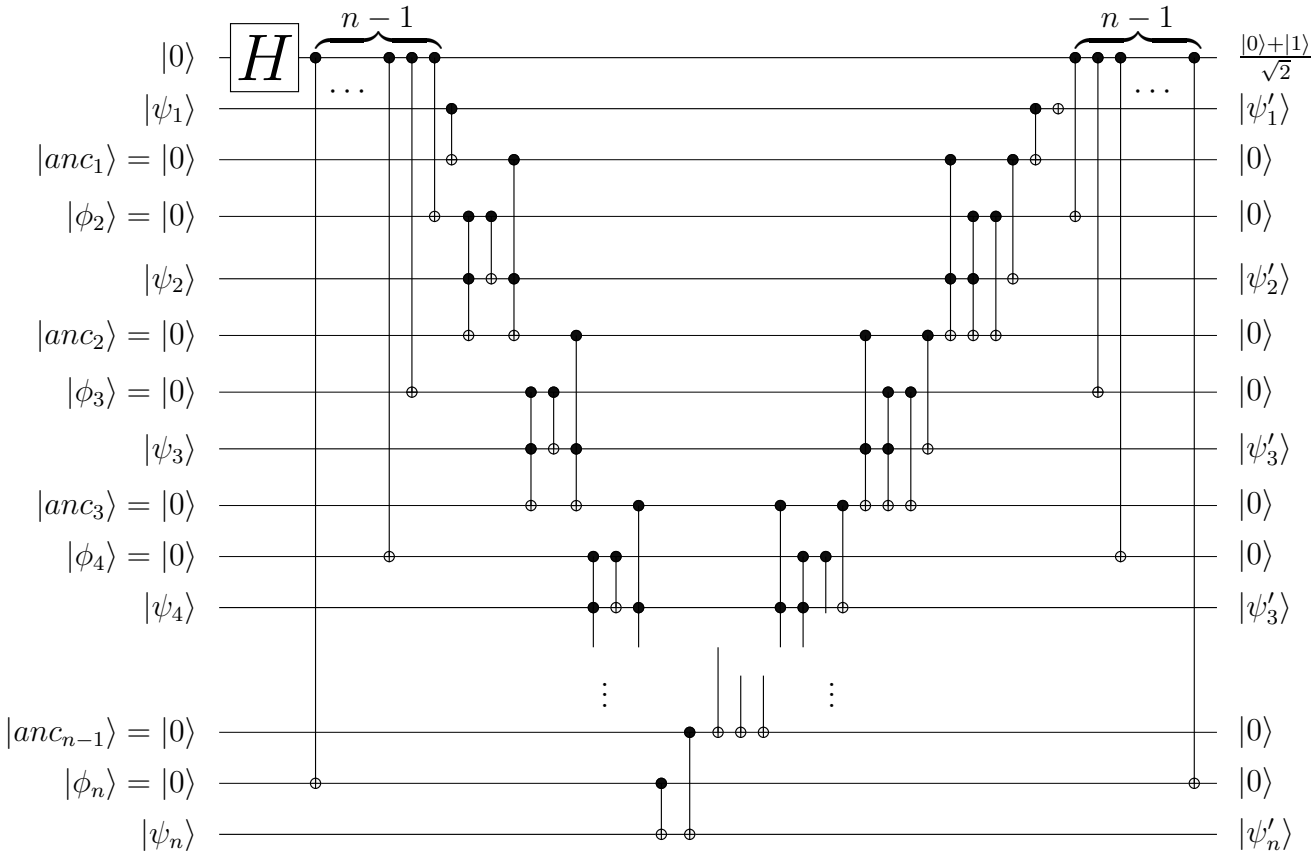

Figure 3.3: Circuit diagram for one time step in an n-qubit random walker

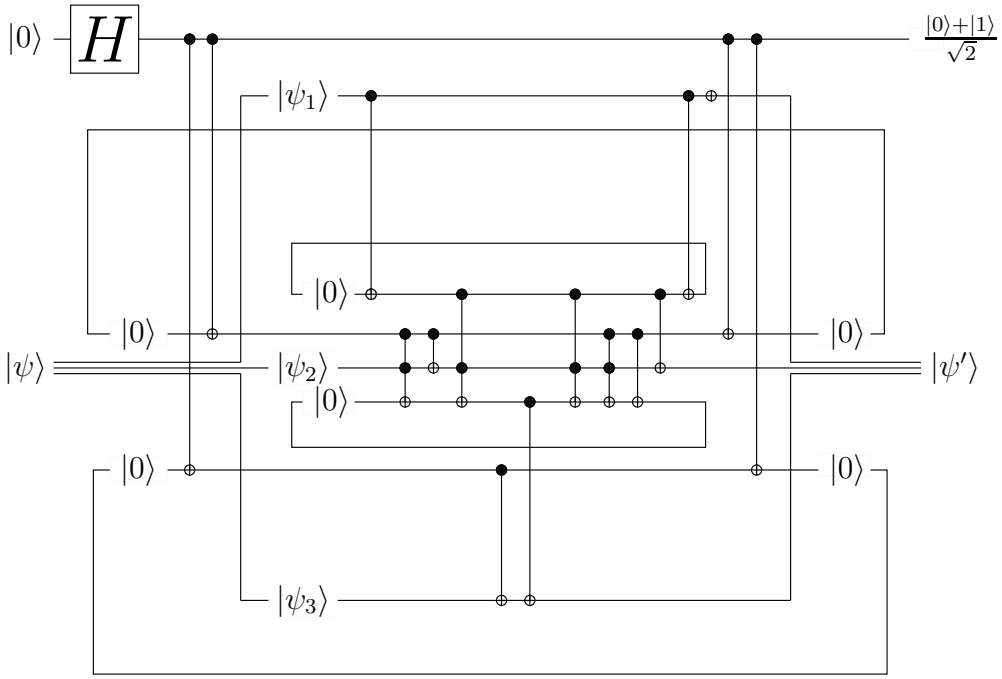

Figure 3.4: Random walk circuit on three qubits

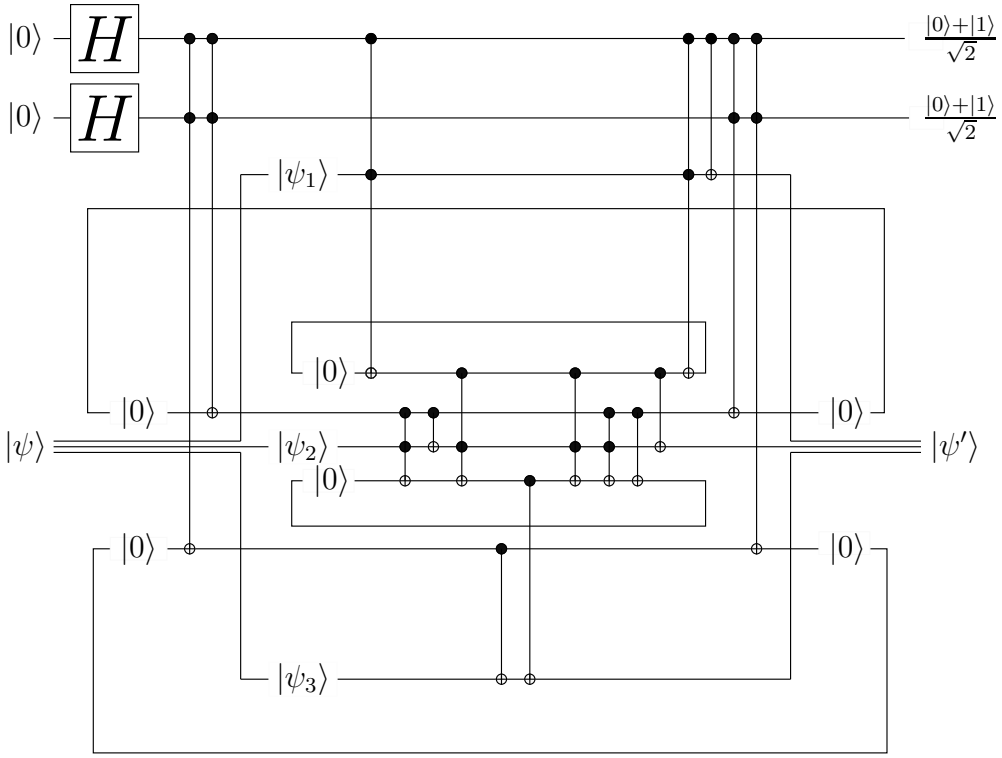

Figure 3.5: Alternative random walk circuit on three qubits

We now address the issue of how to iterate this circuit. We cannot simply feed the output from our circuit directly back into itself because we get unfortunate overlap and cancellation which distorts the probability distribution. To demonstrate, consider a four bit register input into 3.4 (three for the particle position and one for the control bit). This gives us eight possible particle positions. We begin with the register initialized to $|\psi_0\rangle = |0\rangle|001\rangle$. Note the particle starts in the second leftmost position on our line. After one iteration, we get $|\psi_1\rangle = \frac{1}{\sqrt{2}}[|0\rangle|010\rangle + |1\rangle|000\rangle]$. Our second and third iterations yield

$$\begin{aligned}
|\psi_2\rangle &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}[|0\rangle|011\rangle + |1\rangle|001\rangle + |0\rangle|001\rangle - |1\rangle|111\rangle]\right) \\
&= \frac{1}{2}[|0\rangle|011\rangle + (|1\rangle + |0\rangle)|001\rangle - |1\rangle|111\rangle] \\
|\psi_3\rangle &= \frac{1}{2}\Bigg(\frac{1}{\sqrt{2}}\big[|0\rangle|100\rangle + |1\rangle|010\rangle + |0\rangle|010\rangle - |1\rangle|000\rangle + |0\rangle|010\rangle + \\
&\quad |1\rangle|000\rangle - |0\rangle|000\rangle + |1\rangle|110\rangle\big]\Bigg) \\
&= \frac{1}{2\sqrt{2}}[|0\rangle|100\rangle + |1\rangle|010\rangle + 2(|0\rangle|010\rangle) - |0\rangle|000\rangle + |1\rangle|110\rangle]
\end{aligned}$$

After the third iteration, the particle will be in position $|010\rangle$ with probability $\frac{5}{8}$, and positions $|110\rangle, |100\rangle, |000\rangle$ each with probability $\frac{1}{8}$. This does not give the correct distribution on a one-dimensional lattice with eight possible positions. Positions $|000\rangle$ and $|010\rangle$ should each have probability $\frac{3}{8}$.

It seems that if we could prevent overlap and keep the circuit unitary, this problem would be alleviated (note that since the circuit is unitary cancellation will be corrected at the same time). One obvious way to do this is to introduce one control bit for each desired time step. This has obvious drawbacks - for any large number of time steps it is unwieldy. Consider the circuit shown in figure 3.4. If we label this circuit A, then Figure 3.6 shows the difficulty associated with this solution.

I made an attempt to correct the overlap using a modified version of the Hadamard gate.

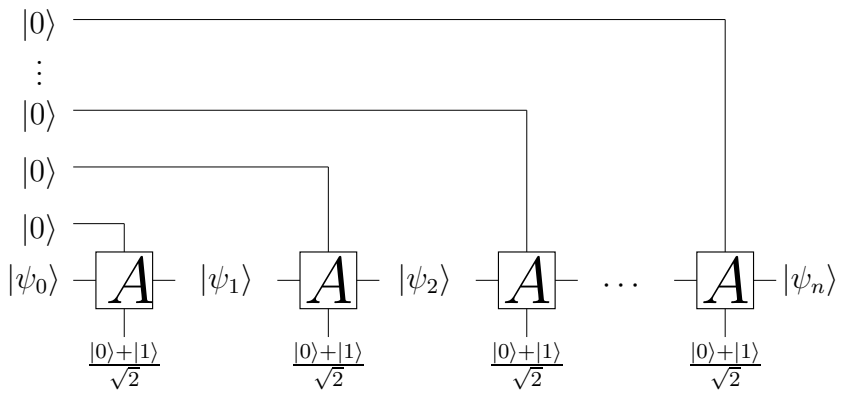

Figure 3.6: Using n control qubits for n time steps

The basic approach was to *tag* each element of the superposition with a unique phase, thus preventing the unwanted overlap. Unfortunately, this modified the probability distribution in another way and was therefore unacceptable as a solution to the theoretical dilemma. However, as is wont to happen in mathematics, it appears as though the idea may have interesting and useful applications in a somewhat unrelated area. See Appendix A for more details.

In practice (with the small quantum computers that have been developed already) another approach is to reset the value of the control bit to $|0\rangle$ after each time step by a nonunitary process similar to but different from measurement. In particular, this new process does not collapse the superposition of the qubit as measurement would. Thus we *discard* the superposition of the control bit after every iteration without measuring it. There is no satisfactory theoretical model for resetting a bit unconditionally. Nonetheless laboratory experimentalists are able to *reset* a qubit on NMR quantum computers in the manner described above, and this is the method which Troy Borneman intends to use for implementing the circuit.

Chapter 4

Implementing the Circuit with NMR

Once we had designed a circuit that behaves asymptotically as the heat equation, we began to look into how to implement this circuit on the nuclear magnetic resonance (NMR) quantum computers at MIT. Here is a brief overview of how NMR works. For a more complete treatment of this subject see [8].

Certain molecules have nuclei which have a measurable *spin* with regard to the field. The laws of quantum physics tell us that spin-$\frac{1}{2}$ nuclei may be in one of two basic states, called *spin up* and *spin down*. The orientation of the spin can be changed by hitting the nucleus with electromagnetic waves at specific radio frequencies. In fact we can find radio frequencies which will place the spin of the nucleus in an arbitrary superposition of the two basic states. The two basic states have slightly different amounts of energy and magnetization, a quantity which can be measured to determine the orientation of the spin. When we measure the orientation, it collapses into one of the basic states. If we then identify *spin up* with $|1\rangle$ and *spin down* with $|0\rangle$, we have the makings of a primitive quantum computer. There are many complications which make implementing a computer of significant size extremely difficult, not to mention expensive. I will address some of these difficulties in the order which we encountered them.

4.1 Output Extraction

The first issue we encountered was both a strength and a weakness. When building a circuit for NMR, we develop a molecule which contains the proper number of qubits, and whose nuclei will interact with each other appropriately with respect to the logic gates desired in the circuit. Then the molecule is hit with the appropriate radio waves to run the circuit. Needless to say, it would be difficult to do this for just one molecule. Instead, a small vial with large quantities of the same molecule is put into the NMR quantum computer at one time. In effect, we have a massive parallel computation happening. Millions of copies of our circuit are being run independently at the same time, one on each molecule. What a beautiful and very efficient way to get a Monte Carlo estimate for the probability distribution of a dynamical system!

The catch comes with the measurement techniques. Since it would be impossible to try to measure individual molecules, we instead get an average of the output over all molecules for each qubit. To demonstrate the problem this presents, lets look at an example. Suppose that we have a three-qubit register with initial conditions of $|\psi\rangle = |001\rangle$ (i.e., with the particle in the second lattice position from the left). Then upon measuring the output of our three-qubit circuit after t time steps through NMR, we learn that the most significant qubit has an average value of $\frac{7}{16}$, the middle qubit an average value of $\frac{1}{2}$, and the least significant qubit an average value of 1. What does this tell us about the average position of our particle? At first glance, not much. It is entirely possible that two different probability distributions could give us this same output. Perhaps we are unable to observe that $|\psi\rangle$ has a value of $|001\rangle$ with probability $\frac{7}{16}$, $|011\rangle$ with probability $\frac{1}{8}$, $|101\rangle$ with probability $\frac{1}{16}$, and $|111\rangle$ with probability $\frac{3}{8}$. This gives us the correct average of expected values for most- and least-significant qubits, but in fact is not the distribution which the correct circuit should produce (for the correct distribution, see last time step in Figure 4.1).

Since our ultimate goal is to develop a circuit which calculates the probability distribution of a particle location in a complex system where we do not know what the answer will be, this seems to be a rather insurmountable obstacle. In fact, it may be. However if we allow the assumptions that

- The circuit is for a three-qubit register (i.e., an eight-position lattice)
- This circuit is operating correctly

I was able find a way to extract the correct probability distribution after any (known) number of time steps.

Let $P(t, |\alpha\rangle)$ be the probability at time t that a particle is at a specific location (on a one-dimensional lattice with eight possible locations) represented by the three qubit register $|\alpha\rangle = |\alpha_1\alpha_2\alpha_3\rangle$. Suppose we start the particle at location $|001\rangle$. If we calculate the probability of the particle's location after the first few time steps, we see in Figure 4.1 that the probabilities quickly fall into a repeating pattern. For $t = 2n + 1, n \geq 1$, we have

$$P(t, |001\rangle) = P(t, |011\rangle) = P(t, |101\rangle) = P(t, |111\rangle) = 0 \tag{4.1}$$

$$P(t, |000\rangle) = P(t, |010\rangle) \tag{4.2}$$

$$P(t, |100\rangle) = P(t, |110\rangle). \tag{4.3}$$

Further, if $t = 2n, n \geq 1$, we have

$$P(t, |000\rangle) = P(t, |010\rangle) = P(t, |100\rangle) = P(t, |110\rangle) = 0 \tag{4.4}$$

$$P(t, |011\rangle) = P(t, |111\rangle). \tag{4.5}$$

Let $P_M(t, |\psi\rangle)$ be the probability that the most significant qubit of $|\psi\rangle = |1\rangle$. Let $P_I(t, |\psi\rangle)$ be the probability that the middle (I for *intermediate*) qubit of $|\psi\rangle = |1\rangle$. Then using this information together with the known pattern of the distribution over the one-dimensional sample space, we can deduce that at odd time step t,

$$P_M(t, |\psi\rangle) = 2P(t, |100\rangle) = 2P(t, |110\rangle) \tag{4.6}$$

$$\Rightarrow P(t, |100\rangle) = P(t, |110\rangle) = \frac{1}{2}(P_M(t, |\psi\rangle)) \tag{4.7}$$

$$P_I(t, |\psi\rangle) = 2P(t, |000\rangle) = 2P(t, |010\rangle) \tag{4.8}$$

$$\Rightarrow P(t, |000\rangle) = P(t, |010\rangle) = \frac{1}{2}(P_I(t, |\psi\rangle)) \tag{4.9}$$

Similarly, for even time step t,

$$P_M(t, |\psi\rangle) = 2P(t, |011\rangle) = 2P(t, |111\rangle) \tag{4.10}$$

$$\Rightarrow P(t, |011\rangle) = P(t, |111\rangle) = \frac{1}{2}(P_I(t, |\psi\rangle)) \tag{4.11}$$

$$P_M(t, |\psi\rangle) = P(t, |101\rangle) + P(t, |111\rangle) \tag{4.12}$$

$$\Rightarrow P(t, |101\rangle) = P_M(t, |\psi\rangle) - P(t, |111\rangle) = P_M(t, |\psi\rangle) - \frac{1}{2}(P_I(t, |\psi\rangle)) \tag{4.13}$$

$$P(t, |001\rangle) = 1 - P_I(t, |\psi\rangle) - P(t, |101\rangle) \tag{4.14}$$

As promised, we now have a way to extract the probability distribution after t time steps for this particular circuit. However, we already know how to find the correct probability distribution for this particular experiment. Perhaps a more practical question at this stage would be this: For an arbitrary number of time steps t, and for a circuit with an arbitrary number of qubits n, can we easily verify that the circuit is working correctly?

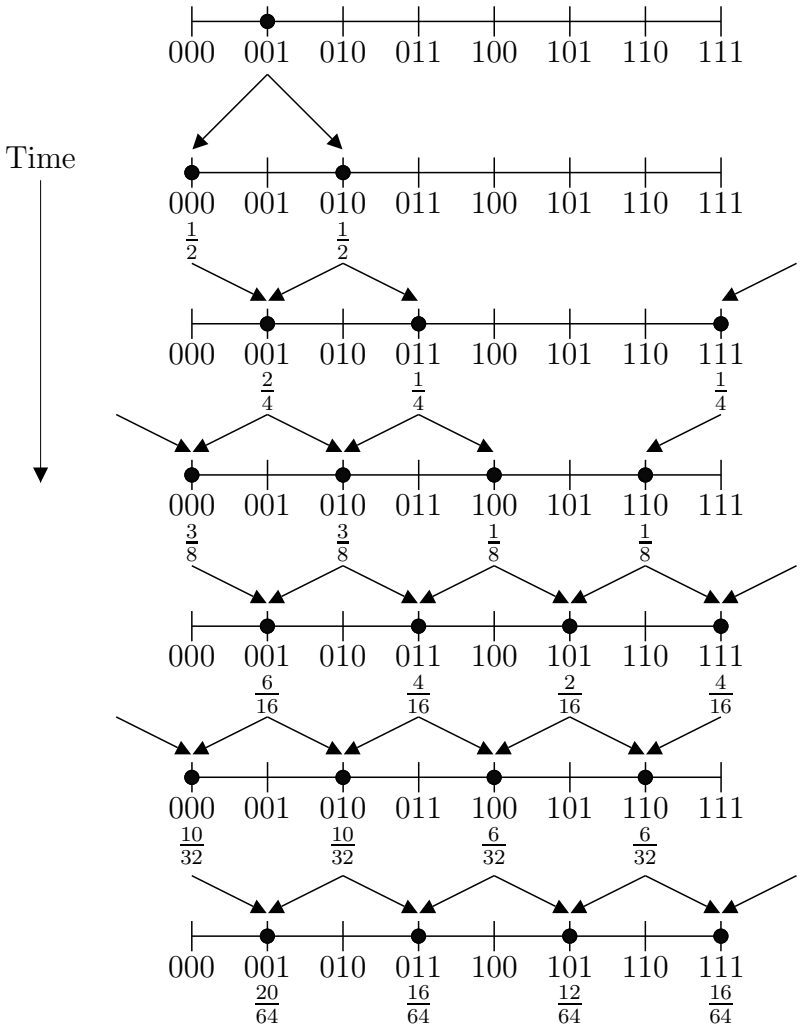

Figure 4.1: Symmetry of probabilities with three qubits

4.2 General Formula for a Periodic Probability Distribution

A general solution to this question (the answer is yes) was worked out in a meeting and appears here courtesy of Professor Boghosian [4]. If we can quickly and easily calculate the correct probability distribution on a lattice with periodic boundary conditions, then we can also quickly and easily calculate the expected value of the register's qubits. This gives us a way to check with a reasonable amount of certainty that the circuit is working. When n is the number of qubits in the register of our quantum circuit, let $N = 2^n$. In other words, N is the number of possible particle positions on our one-dimensional lattice. On an infinite lattice, the probability distribution would just be given by the binomial theorem. Because we are using periodic boundary conditions, we will get a *wrap around* effect which we wish to account for. We will need the following lemma to find a function of N and the number of time steps which gives us the correct probability distribution on a lattice with periodic boundary conditions.

Lemma 1. *Let $N, \beta \in \mathbb{Z}$, with N even. Then*

$$\frac{1}{N}\sum_{\alpha=0}^{N-1} e^{\left(\frac{2\pi i\alpha}{N}\right)\beta} = \begin{cases} 1 & \text{if } \beta \equiv 0 \pmod{N} \\ 0 & \text{if } \beta \not\equiv 0 \pmod{N} \end{cases}$$

$$= \delta_{\beta,0 \pmod{N}}$$

.

Proof. Consider the sum

$$\sum_{\alpha=0}^{N-1} e^{\left(\frac{2\pi i\alpha}{N}\right)}.$$

Note that we are simply adding up the N^{th} roots of unity. Because N is even, for each

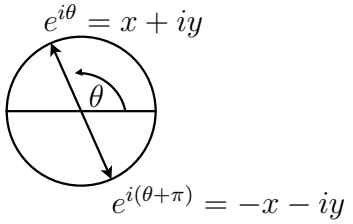

Figure 4.2: Proof that $e^{i\theta} + e^{i(\theta+\pi)} = 0$

root $e^{i\theta}$, $e^{i(\theta+\pi)}$ is also an N^{th} root. Figure 4.2 shows that $e^{i\theta} + e^{i(\theta+\pi)} = 0$. Hence the terms of the above sum cancel each other out. Further, multiplying the exponent by an integer simply permutes the N^{th} roots, and so when $\beta \neq 0$ the lemma is true. When $\beta \equiv 0 \pmod{N}$, the result is obvious.

□

By Lemma 1, we have

$$\sum_{k=0}^{N-1} P(x,t)e^{\frac{2\pi i}{N}k} = 0.$$

Picking $0 \leq x' \leq N-1$ and modifying this sum slightly, we get

$$\sum_{k=0}^{N-1} P(x,t)e^{\frac{2\pi i}{N}k(x'-x)} = \begin{cases} NP(x,t) & \text{if } x' = x \\ 0 & \text{if } x' \neq x \end{cases}, \tag{4.15}$$

so

$$\sum_{x=0}^{N-1}\sum_{k=0}^{N-1} P(x,t)e^{\frac{2\pi i}{N}k(x'-x)} = NP(x',t), \tag{4.16}$$

or

$$\frac{1}{N}\sum_{k=0}^{N-1}\sum_{x=0}^{N-1} P(x,t)e^{\frac{2\pi i}{N}k(x'-x)} = P(x',t), \tag{4.17}$$

or

$$\frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{x=0}^{N-1} P(x,t)e^{-\frac{2\pi ixk}{N}}\right)e^{\frac{2\pi ix'k}{N}} = P(x',t). \tag{4.18}$$

Let $\tilde{P}(k,t) = \sum_{x=0}^{N-1} P(x,t)e^{-\frac{2\pi ixk}{N}}$. Then equation 4.18 becomes

$$\frac{1}{N}\sum_{k=0}^{N-1} \tilde{P}(k,t)e^{\frac{2\pi ixk}{N}} = P(x,t) \tag{4.19}$$

Lemma 2. *Let $\tilde{P}(k,t)$ be defined as above. Then*

$$\tilde{P}(k,t) = \left[\cos\left(\frac{2\pi k}{N}\right)\right]^t \tilde{P}(k,0) \tag{4.20}$$

.

Proof.

$$\tilde{P}(k,t+1) = \sum_{x=0}^{N-1} P(x,t+1)e^{-\frac{2\pi ikx}{N}} \tag{4.21}$$

$$= \sum_{x=0}^{N-1} \left(\frac{1}{2}P(x-1,t) + \frac{1}{2}P(x+1,t)\right) e^{-\frac{2\pi ikx}{N}} \tag{4.22}$$

$$= \frac{1}{2}\sum_{x=0}^{N-1} P(x-1,t)e^{-\frac{2\pi ikx}{N}} + \frac{1}{2}\sum_{x=0}^{N-1} P(x+1,t)e^{-\frac{2\pi ikx}{N}}. \tag{4.23}$$

But we are using periodic boundary conditions, which means that $\sum_{x=0}^{N-1} P(x \pm 1, t) = \sum_{x=0}^{N-1} P(x,t)$. Hence

$$\tilde{P}(k,t+1) = \frac{1}{2}\sum_{x=0}^{N-1} P(x,t)e^{-\frac{2\pi ik(x+1)}{N}} + \frac{1}{2}\sum_{x=0}^{N-1} P(x,t)e^{-\frac{2\pi ik(x-1)}{N}} \tag{4.24}$$

$$= \frac{1}{2}\sum_{x=0}^{N-1} P(x,t)e^{-\frac{2\pi ikx}{N}}e^{-\frac{2\pi ik}{N}} + \frac{1}{2}\sum_{x=0}^{N-1} P(x,t)e^{-\frac{2\pi ikx}{N}}e^{\frac{2\pi ik}{N}} \tag{4.25}$$

$$= \left(\frac{e^{\frac{2\pi ik}{N}} + e^{-\frac{2\pi ik}{N}}}{2}\right) \sum_{x=0}^{N-1} P(x,t)e^{-\frac{2\pi ikx}{N}} \tag{4.26}$$

$$\tilde{P}(k,t+1) = \cos\left(\frac{2\pi k}{N}\right)\tilde{P}(k,t). \tag{4.27}$$

Therefore $\tilde{P}(k,t) = \left[\cos\left(\frac{2\pi k}{N}\right)\right]^t \tilde{P}(k,0)$.

□

Finally we are in a position to derive a formula to calculate the expected probability distribution. Beginning with Equation 4.19 and using Lemma 2,

$$P(x,t) = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{P}(k,t) e^{\frac{2\pi i x k}{N}} \tag{4.28}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} \left(\left[\cos\left(\frac{2\pi k}{N}\right) \right]^t \tilde{P}(k,0) \right) e^{\frac{2\pi i k x}{N}} \tag{4.29}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} \left(\left[\frac{e^{\frac{2\pi i k}{N}} + e^{-\frac{2\pi i k}{N}}}{2} \right]^t \sum_{x=0}^{N-1} P(x,0) e^{-\frac{2\pi i k x}{N}} \right) e^{\frac{2\pi i k x}{N}}. \tag{4.30}$$

Suppose x_0 is the original position of the particle at time $t = 0$. Then $P(x, 0)$ gives the probability that $x = x_0$ at time $t = 0$, which is 1 when $x = x_0$ and 0 when $x \neq x_0$. Hence $\sum_{x=0}^{N-1} P(x,0) e^{-\frac{2\pi i k x}{N}} = e^{-\frac{2\pi i k x_0}{N}}$. Using this fact and the binomial formula to expand the exponential form of the cosine term, Equation 4.30 becomes

$$P(x,t) = \frac{1}{2^t}\frac{1}{N}\sum_{k=0}^{N-1} e^{\frac{2\pi i k (x-x_0)}{N}} \sum_{l=0}^{t} \binom{t}{l} \left(e^{\frac{2\pi i k}{N}}\right)^{t-l} \left(e^{-\frac{2\pi i k}{N}}\right)^{l} \tag{4.31}$$

$$= \frac{1}{2^t}\frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{t} \binom{t}{l} \left(e^{\frac{2\pi i k}{N}(x-x_0+t-2l)}\right) \tag{4.32}$$

$$= \frac{1}{2^t}\sum_{l=0}^{t} \binom{t}{l} \frac{1}{N}\sum_{k=0}^{N-1} \left(e^{\frac{2\pi i k}{N}(x-x_0+t-2l)}\right) \tag{4.33}$$

$$= \frac{1}{2^t}\sum_{l=0}^{t} \binom{t}{l} \delta_{x-x_0+t-2l,0 \pmod N} \qquad \text{(by Lemma 1).} \tag{4.34}$$

When $\delta = 1$, $x - x_0 + t - 2l \equiv 0 \pmod N$. In addition to the observation that this implies that $x - x_0 + t$ must be even, we note that $x - x_0 + t \equiv 2l \pmod N$ and $x - x_0 + t = 2l + jN$ for some $j \in \mathbb{Z}$. From this it follows that

$$l = \frac{x - x_0 + t - jN}{2}. \tag{4.35}$$

Since we are summing over all values of l between 0 and t, we want to include all values of j which satisfy

$$\begin{array}{ccccc} 0 & \leq & l & \leq & t \\ 0 & \leq & \frac{x-x_0+t-jN}{2} & \leq & t \\ 0 & \leq & x-x_0+t-jN & \leq & 2t \\ -(t+x-x_0) & \leq & jN & \leq & 2t-(t+x-x_0) \\ \frac{-t-x+x_0}{N} & \leq & j & \leq & \frac{t-x+x_0}{N} \end{array} . \tag{4.36}$$

Let $a = \lceil \frac{-t-x+x_0}{N} \rceil$, and let $b = \lfloor \frac{t-x+x_0}{N} \rfloor$. Using the inequality 4.36, Equation 4.34 becomes

$$P(x,t) = \begin{cases} 0 & \text{if } x-x_0+t \text{ is odd} \\ \sum_{j=a}^{b} \binom{t}{\frac{x-x_0+t+jN}{2}} & \text{if } x-x_0+t \text{ if even} \end{cases} . \tag{4.37}$$

4.3 Optimizing the Circuit

In December, 2004 we met with Professor David Cory from MIT, who generously agreed to let us spend some time on his NMR equipment trying to implement our circuit. During our first meeting it became evident that even with an eight position lattice, our circuit was too big to implement using current technology. As you can see in Figure 3.4, the circuit uses three qubits for the lattice, one qubit for a control qubit, and four ancillary qubits (for carry information, etc). This gives a total of eight qubits, plus a relatively large number (16) of controlled `not` gates. According to Professor Cory, we needed to reduce this to a *maximum* of four qubits and approximately ten controlled `not` gates. So the development priority became optimizing our circuit.

The most significant optimization came from a computer science observation usually referred to as *two's compliment*. Specifically, if ψ is a n qubit binary number, then (modulo

n) $-\psi$ can be obtained by applying a logical *not* to each qubit of ψ, and then adding one to the result. For example, let $\psi = 101$. Then $-\psi = 010 + 001 = 011$. To verify for this example, note that $\psi + (-\psi) = 101 + 011 = 000$. The more general case is proved in the same manner:

Lemma 3. *Let $\psi = \alpha_1\alpha_2\cdots\alpha_n$ be an n qubit binary register, where α_n is the least significant qubit and α_1 is the most significant qubit. Let $\overline{\psi} = \overline{\alpha}_1\overline{\alpha}_2\cdots\overline{\alpha}_n$ be the result of applying a logical* not *to each of the individual qubits. In other words, for each $1 \leq i \leq n$,*

$$\overline{\alpha}_i = \begin{cases} 0 & \textit{if } \alpha_i = 1 \\ 1 & \textit{if } \alpha_i = 0 \end{cases}. \tag{4.38}$$

Then $\overline{\psi} + 1 = -\psi$.

Proof. From the definition of $\overline{\alpha}_i$ in Equation 4.38, we observe that $\overline{\alpha}_i + \alpha_i = 1$ for each $1 \leq i \leq n$. Hence $\overline{\psi} + \psi = 111\cdots1$. But $111\cdots1 + 1 = 0 \pmod 2$. Therefore

$$\overline{\psi} + \psi + 1 = 0 \tag{4.39}$$

$$\overline{\psi} + 1 = -\psi. \tag{4.40}$$

□

Lemma 3 allows us to subtract one in our circuit without using an n qubit number to do so. The main idea is summarized by the next lemma.

Lemma 4. *Let ψ be an n qubit number. Let $\overline{\psi}$ be the result of applying a logical* not *to each of the n qubits in ψ independently. Then $\psi - 1 = \overline{(\overline{\psi} + 1)}$.*

Proof. Starting from the result of Lemma 3,

$$\begin{aligned} -\psi &= \overline{\psi} + 1 \\ \psi &= -(-\psi) \\ &= -(\overline{\psi} + 1) \\ &= \overline{(\overline{\psi} + 1)} + 1 \\ \psi - 1 &= \overline{(\overline{\psi} + 1)}. \end{aligned}$$

□

Using Lemma 4, we can immediately shave off n qubits from our original circuit in Figure 3.3. The original circuit needs an entire n-qubit register *just to hold the value of negative one* so that we can add it in the event we need to subtract one. Lemma 4 allows us to subtract one by adding one and using judiciously placed controlled `not` gates. Furthermore, Figure 4.3 demonstrates that we can lose a fair number of controlled `not` gates in the process. In the event that the control qubit is one, there are $2n$ gates to `not` the qubits of $|\psi\rangle$ before and after adding one. There are also $2n - 1$ gates setting (and afterwards unsetting) the carry qubits. In addition there are n gates adding carry qubits to the register qubits, plus the Hadamard gate. Finally there are n ancillary qubits (including the control qubit) and the n qubits in the register $|\psi\rangle$. This cuts our total complexity from $12n - 7$ before optimization almost in half to $7n - 1$. Though the order of complexity is still linear, for smaller scale implementations we have made a significant gain.

This simple optimization reduces the mess of Figure 3.4 to the much nicer Figure 4.4. But we needed to do even better, since the limits of current NMR technology demand that we use only four qubits and about ten gates. Because our register is so small, we can use logic instead of carry qubits to help us add. In Figure 4.5 we can see that (inside the `not` gates dictated by Lemma 4) adding one is actually quite simple with just three qubits. If

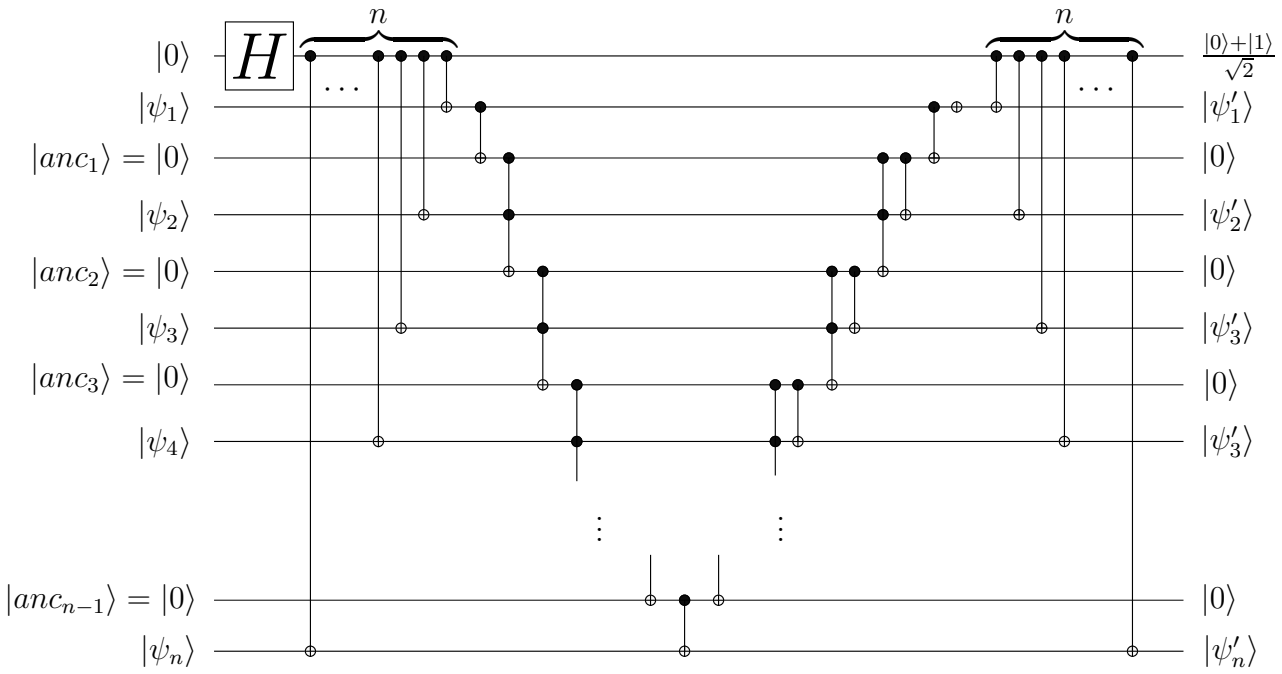

Figure 4.3: Optimized random walk circuit on n qubits

the first two qubits are both in state $|1\rangle$, then we will carry to the third and must `not` it accordingly. If the first qubit has value one, then we must `not` the second. Also, adding one means that we will always `not` the first. Hence the circuit in Figure 3.4 has been reduced to one with four qubits and only ten logical gates. During a meeting with Professor Cory, Troy Borneman was able to call upon Professor Cory's experience to give us the final optimizations realized in Figure 4.6. This circuit is currently under implementation by Troy Borneman in Professor Cory's laboratory. When run for t time steps, this circuit has a complexity of $11t$.

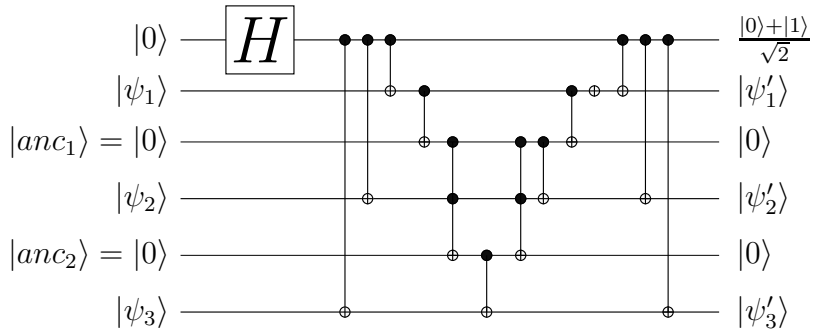

Figure 4.4: Optimized random walk circuit on three qubits

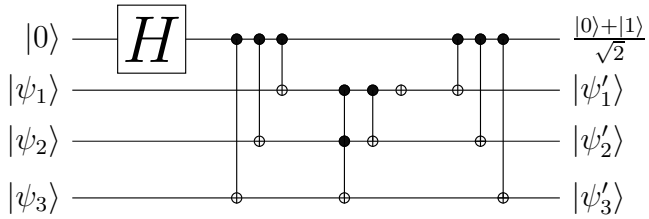

Figure 4.5: Second optimization of random walk circuit on three qubits

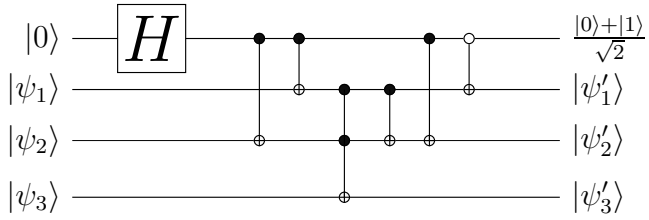

Figure 4.6: Third optimization of random walk circuit on three qubits

Chapter 5

Conclusion

We have shown that it is possible to simulate the heat equation in one dimension using a circuit whose complexity scales logarithmically with respect to the number of discrete positions on a one-dimensional lattice. As mentioned in the introduction, this is very exciting since we could conceivably simulate a random walk on a lattice with 2^{100} positions (our circuit will only have complexity on the order of 100), a feat well beyond what classical computers are currently capable of. Furthermore, we are not limited to one dimension. Using methods similar to the ones described in this paper, it should be possible to generalize this statement to the diffusion of heat in any fixed finite number of dimensions.

While implementing a small scale version of our circuit, we learned that it is very possible that certain circuit components may be less available than others. For classical complexity, generally it is assumed that a bit has approximately the same complexity as a primitive logical gate. This relationship does not necessarily hold for the complexity of a quantum circuit. For an example, consider the circuit in Figure 4.5. We learned in the course of implementing our circuit that the controlled `not` gate between $|0\rangle$ and $|\psi_3\rangle$ is something like three times the complexity of the controlled `not` gate between $|0\rangle$ and $|\psi_2\rangle$. Along these lines, it is worth noting that should qubits and controlled `not` gate turn out to have significant

enough differences in their complexities or availability, we have presented two different circuit models which might accommodate either direction.

The circuit in Figure 4.3 maintains linearity with respect to n (approximately $7n - 1$ gates and qubits). In contrast, a circuit generalizing the one shown in Figure 4.5 to a lattice with 2^n positions will trade $n - 1$ carry qubits for $\frac{n(n-1)}{2} + 2n$ controlled `not` gates. While we don't maintain linearity, it is possible that if the number of qubits is severely restricted for some reason, this trade-off might be worthwhile.

In Section 4.1 we illustrate that extracting a full probability distribution from our circuit is not trivial. It is worth mentioning that there may be other important questions we could answer more easily. For example, perhaps we have developed a modified circuit which models some very complicated behavior of a particle over the interval $[0, 1]$. Then we can easily determine the probability that this particle lands in a position greater than one half by simply measuring the average state of the most significant qubit.

To conclude, I would like to list some interesting questions still wanting answers. First, it remains to outline clearly the higher-dimensional circuits. In addition, we believe it is possible to modify this circuit slightly and obtain similar results for non-homogeneous random walks (i.e. random walks where the probability of moving in a particular direction is a function of the particle location). Finally, it would be nice to find a general way to modify the Hadamard gate such that we can get an arbitrary probability distribution between two possible outcomes.

Appendix A

When I encountered the problem of overlap with my original circuit, I attempted to find alternatives which would use the idea of the Hadamard gate to give a binomial distribution but avoid the problems which that gate appeared to possess for my purposes. A possible compromise which was suggested by Professor Boghosian was to *decohere* the control bit after every iteration. This involves a procedure which changes the spin of the control bit in such a way that the resulting superposition is random. This turned out not to work, however it led to the idea of using a modified version U of the Hadamard gate.

$$U = \left(\frac{1}{\sqrt{2}}\right) \begin{pmatrix} e^{i\alpha_n} & 1 \\ 1 & -e^{-i\alpha_n} \end{pmatrix}$$

With the restriction that $\alpha \neq k\pi, \forall k \in \mathbb{Z}$. Note that this matrix is unitary:

$$
\begin{aligned}
UU^{\dagger} &= \left(\frac{1}{\sqrt{2}}\right)^2 \begin{pmatrix} e^{i\alpha_n} & 1 \\ 1 & -e^{-i\alpha_n} \end{pmatrix} \begin{pmatrix} e^{-i\alpha_n} & 1 \\ 1 & -e^{i\alpha_n} \end{pmatrix} \\
&= \left(\frac{1}{2}\right) \begin{pmatrix} e^{i\alpha_n - i\alpha_n} + 1 & e^{i\alpha_n} - e^{i\alpha_n} \\ e^{-i\alpha_n} - e^{-i\alpha_n} & 1 + e^{-i\alpha_n + i\alpha_n} \end{pmatrix} \\
&= \left(\frac{1}{2}\right) \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
\end{aligned}
$$

This time, we will use ψ_+, ψ_{+2}, etc., to indicate we have added one, two, etc., to the register. Similarly, we will use ψ_-, ψ_{-2}, etc., for subtraction. On cursory inspection, our iterations through the circuit yields something more satisfactory.

$$\psi_0 = |0\rangle|\psi\rangle \tag{A.1}$$

$$\psi_1 = \left(\frac{1}{\sqrt{2}}\right)\left(e^{i\alpha_1}|0\rangle|\psi_+\rangle + |1\rangle|\psi_-\rangle\right) \tag{A.2}$$

$$\psi_2 = \left(\frac{1}{2}\right)\left(e^{i\alpha_1}\left(e^{i\alpha_2}|0\rangle|\psi_{+2}\rangle + |1\rangle|\psi\rangle\right) + \left(|0\rangle|\psi\rangle - e^{-i\alpha_2}|1\rangle|\psi_{-2}\rangle\right)\right) \tag{A.3}$$

$$= \left(\frac{1}{2}\right)\left(e^{i(\alpha_1+\alpha_2)}|0\rangle|\psi_{+2}\rangle + e^{i\alpha_1}|1\rangle|\psi\rangle + |0\rangle|\psi\rangle - e^{-i\alpha_2}|1\rangle|\psi_{-2}\rangle\right) \tag{A.4}$$

$$\psi_3 = \left(\frac{1}{2\sqrt{2}}\right)\Bigg[e^{i(\alpha_1+\alpha_2)}\left(e^{i\alpha_3}|0\rangle|\psi_{+3}\rangle + |1\rangle|\psi_{+1}\rangle\right) + e^{i\alpha_1}\left(|0\rangle|\psi_+\rangle - e^{-i\alpha_3}|1\rangle|\psi_-\rangle\right) + e^{i\alpha_3}|0\rangle|\psi_+\rangle + |1\rangle|\psi_-\rangle - e^{-i\alpha_2}\left(|0\rangle|\psi_-\rangle - e^{-i\alpha_3}|1\rangle|\psi_{-3}\rangle\right)\Bigg] \tag{A.5}$$

$$= \left(\frac{1}{2\sqrt{2}}\right)\Bigg[e^{i(\alpha_1+\alpha_2+\alpha_3)}|0\rangle|\psi_{+3}\rangle + e^{i(\alpha_1+\alpha_2)}|1\rangle|\psi_+\rangle + e^{i\alpha_1}|0\rangle|\psi_+\rangle - e^{i(\alpha_1-\alpha_3)}|1\rangle|\psi_-\rangle + e^{i\alpha_3}|0\rangle|\psi_+\rangle + |1\rangle|\psi_-\rangle - e^{-i\alpha_2}|0\rangle|\psi_-\rangle - e^{-i(\alpha_2+\alpha_3)}|1\rangle|\psi_{-3}\rangle\Bigg] \tag{A.6}$$

Provided that we can guarantee that each sum $S_i = (\alpha_{i_1} + \alpha_{i_2} + \cdots + \alpha_{i_n}$ will be unique and that $0 < S_i < \pi$, then complete cancellation of terms will no longer occur. Let $\alpha_n = \left(\frac{1}{\phi_n}\right)^2$, where ϕ_n is the n^{th} prime number starting with 2. Our choice of α_n is motivated by the following lemma. Using the lemma, we can show that will never have a cancellation or overlap of terms.

Lemma 5. *Let $\alpha_x = \left(\frac{1}{\phi_x}\right)^2$, where ϕ_x is the x^{th} prime number. Let $\alpha_{i_1} + \alpha_{i_2} + \cdots + \alpha_{i_m}$ and $\alpha_{j_1} + \alpha_{j_2} + \cdots + \alpha_{j_n}$ be two sequences of such fractions. Let $a_x, b_x \in \mathbb{Z}$ such that $a_x^2\alpha_{i_x} < 1$ and $b_x^2\alpha_{j_x} < 1$ for all $x \in \mathbb{Z}$. In particular, this means that $\phi_x \nmid a_x$ for all $x \in \mathbb{Z}$. Then*

$$a_1\alpha_{i_1} + a_2\alpha_{i_2} + \cdots + a_m\alpha_{i_m} = b_1\alpha_{j_1} + b_2\alpha_{j_2} + \cdots + b_n\alpha_{j_n}$$

if and only if the following are all true

1. $m = n$

2. $i_x = j_x$ *for all* $1 \leq x \leq m = n$ *(possibly rearranging the indices), and*

3. $a_x = b_x$ *for all* $1 \leq x \leq m = n$.

Proof. The reverse direction is obvious. To prove the forward direction, assume the two sequences have the same value, and that the variables a, b, and ϕ are as described above. We proceed by induction. For the base case,

$$\frac{a_1}{\phi_{i_1}^2} = \frac{b_1}{\phi_{j_1}^2} + \frac{b_2}{\phi_{j_2}^2} + \cdots + \frac{b_n}{\phi_{j_n}^2} \tag{A.7}$$

Let $\hat{\phi_{j_x}^2} = (\phi_{j_1}^2)(\phi_{j_2}^2)(\cdots)(\phi_{j_{x-1}}^2)(\phi_{j_{x+1}}^2)(\cdots)(\phi_{j_z}^2)$. Then

$$a_1 = \frac{\phi_{i_1}^2(b_1\hat{\phi_{j_1}^2} + b_2\hat{\phi_{j_2}^2} + \cdots + b_n\hat{\phi_{j_n}^2})}{(\phi_{j_1}^2)(\phi_{j_2}^2)(\cdots)(\phi_{j_n}^2)}. \tag{A.8}$$

and

$$(a_1)(\phi_{j_1}^2)(\phi_{j_2}^2)(\cdots)(\phi_{j_n}^2) = \phi_{i_1}^2(b_1\hat{\phi_{j_1}^2} + b_2\hat{\phi_{j_2}^2} + \cdots + b_n\hat{\phi_{j_n}^2}). \tag{A.9}$$

By hypothesis $\phi_{i_1} \nmid a_1$. Thus by the Fundamental Theorem of Arithmetic, $\phi_{j_x} = \phi_{i_1}$ for some x. Without loss of generality, suppose that $x = 1$ and subtract $\frac{b_1}{\phi_{j_1}^2}$ from both sides of Equation A.7. Then manipulating this new equation in the same manner as A.8, we get

$$a_1 - b_1 = \frac{\phi_{j_1}^2(b_2\hat{\phi_{j_2}^2} + b_3\hat{\phi_{j_3}^2} + \cdots + b_n\hat{\phi_{j_n}^2})}{(\phi_{j_2}^2)(\phi_{j_3}^2)(\cdots)(\phi_{j_n}^2)}, \tag{A.10}$$

and

$$(a_1 - b_1)(\phi_{j_2}^2)(\phi_{j_3}^2)(\cdots)(\phi_{j_n}^2) = \phi_{j_1}^2(b_2\hat{\phi_{j_2}^2} + b_3\hat{\phi_{j_3}^2} + \cdots + b_n\hat{\phi_{j_n}^2}). \tag{A.11}$$

By hypothesis, $\phi_{j_1} \nmid \phi_{j_x}$ for all x such that $2 \leq x \leq n$. Also by hypothesis, $|a_1|, |b_1| < \phi_{j_1} \Rightarrow |a_1 - b_1| < 2\phi_{j_1}$. Therefore, if $\phi_{j_1} \mid (a_1 - b_1)$, then either $a_1 = b_1$ or $|a_1 - b_1| = \phi_{j_1}$.

In the former case, the proof is finished since the sequences must be the same. In the latter case, we can cancel ϕ_{j_1} from both sides of Equation A.11 to get

$$\pm(\phi_{j_2}^2)(\phi_{j_3}^2)(\cdots)(\phi_{j_n}^2) = \phi_{j_1}(b_2\hat{\phi_{j_2}^2} + b_3\hat{\phi_{j_3}^2} + \cdots + b_n\hat{\phi_{j_n}^2}). \tag{A.12}$$

The Fundamental Theorem of Arithmetic ensures that $\phi_{j_x} = \phi_{j_1}$ for some x such that $2 \leq x \leq n$. This contradicts the choice of ϕ_{j_x} for $x > 1$. Hence $n = 1$, $a_1 = b_1$, $i_1 = j_1$, and all three conditions of the lemma are satisfied. Now for the induction step, assume that

$$\frac{a_1}{\phi_{i_1}^2} + \frac{a_2}{\phi_{i_2}^2} + \cdots + \frac{a_{m-1}}{\phi_{i_{m-1}}^2} = \frac{b_1}{\phi_{j_1}^2} + \frac{b_2}{\phi_{j_2}^2} + \cdots + \frac{b_n}{\phi_{j_n}^2}$$

if and only if all three conditions of Lemma 5 are met. We need to show that the same holds for

$$\frac{a_1}{\phi_{i_1}^2} + \frac{a_2}{\phi_{i_2}^2} + \cdots + \frac{a_{m-1}}{\phi_{i_{m-1}}^2} + \frac{a_m}{\phi_{i_m}^2} = \frac{b_1}{\phi_{k_1}^2} + \frac{b_2}{\phi_{k_2}^2} + \cdots + \frac{b_s}{\phi_{k_s}^2}$$

To see this, simply clear one term from the left hand side of the equation:

$$\begin{aligned}\frac{a_1}{\phi_{i_1}^2} + \frac{1}{\phi_{i_2}^2} + \cdots + \frac{a_{m-1}}{\phi_{i_{m-1}}^2} &= \frac{b_1}{\phi_{k_1}^2} + \frac{b_2}{\phi_{k_2}^2} + \cdots + \frac{b_s}{\phi_{k_s}^2} - \frac{a_m}{\phi_{i_m}^2} \\ &= \frac{b_1}{\phi_{k_1}^2} + \frac{b_2}{\phi_{k_2}^2} + \cdots + \frac{b_t}{\phi_{k_t}^2}\end{aligned}$$

for some $t \in \mathbb{Z}$, where we have combined $\frac{a_m}{\phi_{i_m}^2}$ with another term on the right hand side if appropriate. By our induction hypothesis, these two sequences must be the same.

□

Note that this sequence $\{\alpha_i\}$ will work for any number of time steps through our circuit,

since

$$\begin{aligned}\sum_{i=1}^{\infty}\alpha_i &= \sum_{i=1}^{\infty}\left(\pm\frac{1}{\phi_i}\right)^2 \\ &\leq \sum_{x=1}^{\infty}\left(\frac{1}{2}\right)\left(\frac{1}{x}\right)^2 \\ &\leq \frac{1}{2}\int_1^{\infty}\frac{1}{x^2}dx \\ &\leq 1\end{aligned}$$

shows that the sum of any number of terms will never reach π.

Unfortunately, on closer inspection if we calculate the probability distribution of Equation A.6, the phases introduced distort the binomial distribution given by the Hadamard gate. For example, the Measurement Postulate tells us that we will observe the outcome of $|\psi_{-3}\rangle$ with a probability of

$$|-e^{-i(\alpha_2+\alpha_3)}|^2, \tag{A.13}$$

which at the very least is not obviously equal to the correct probability of $\frac{1}{8}$. Hence as a way to avoid resetting the control bit, this is not a viable solution without further work. However, the method of applying a unique *tag* to each potential particle path is reminiscent of Feynman Path integrals. For this reason we believe that this deserves further study.

Bibliography

[1] Tom M. Apostol. *Mathematical Analysis*. Addison Wesley Longman, second edition, 1974.

[2] Paul E. Black. http://hissa.nist.gov/ black/quantum/genadder.html. *National Institute of Standards and Technology*, 11/20/03.

[3] Bruce M. Boghosian. Untitled. Notes from a Partial Differential Equations course at Tufts University, 2004.

[4] Bruce M. Boghosian. Untitled. Notes from a course in Quantum Computation, 2004.

[5] C. Henry Edwards and David E. Penney. *Differential Equations and Boundary Value Problems*. Prentice-Hall, Inc., second edition, 2000.

[6] Mika Hirvensalo. *Quantum Computing*. Springer, second edition, 2004.

[7] NATO Advanced Workshop. *Quantum Computing Classical Physics*, Discrete Simulation of Fluid Dynamics: New Trends, New Perspectives, Cargese, France, July 2001.

[8] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[9] Marco A. Pravia, Zhiying Chen, Jeffrey Yepez, and David G. Cory. Experimental demonstration of quantum lattice gas computation. *Quantum Information Processing*, 2(1-2), April 2003.