Effects of Control Error on an Adiabatic Quantum Algorithm

by

Edward L. Platt

Submitted to the Department of Physics
in partial fulfillment of the requirements for the degree of

Bachelor of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Author .....
Department of Physics
May 18, 2006

Certified by .....
Edward Farhi
Professor, Department of Physics
Thesis Supervisor

Accepted by .....
David E. Pritchard
Senior Thesis Coordinator, Department of Physics

Effects of Control Error on an Adiabatic Quantum Algorithm

by

Edward L. Platt

Submitted to the Department of Physics
on May 18, 2006, in partial fulfillment of the
requirements for the degree of
Bachelor of Science

Abstract

Noise in adiabatic quantum computation can be modelled as a perturbation of the problem Hamiltonian. For a type of noise called control error, the perturbation can be considered to have the same structure as the problem Hamiltonian. If the problem Hamiltonian, and therefore the noise, are 2-local, then the result of the adiabatic algorithm can be simulated somewhat more efficiently than an algorithm with an arbitrary problem Hamiltonain. Using optimized numerical methods, I present an analysis of the effect of 1-local and 2-local control error on the success of an adiabatic algorithm that solves the agree problem. Furthermore, I examine how the maximum allowable noise, or success threshold, scales with the number of qubits. These analyses suggest the existence of a minimum success threshold for the particular algorithm considered in the presence of only 2-local noise on an arbitrarily large number of qubits, as well as a polynomial decrease in success threshold with the number of qubits.

Thesis Supervisor: Edward Farhi
Title: Professor, Department of Physics

Acknowledgments

I would like to thank: Edward Farhi for guidance, enlightening discussions, feedback, and taking me on as an advisee, Stephen Jordan, Peter Shor, Shay Mozes, and Jay Gill for enlightening discussions, Aram Harrow and Andrew Cross for recommending possible thesis topics and enlightening discussions, Brandon Willard for enlightening discussions and feedback, Seth Lloyd and Paul Penfield Jr. for introducing me, and many MIT frosh, to the intriguing world of information and entropy, Gerald Jay Sussman for guidance, inspiration, and stimulating discussions, the staff of Toscanini's for stimulating beverages, wireless network access and friendly service, Ronald H. Cordover for generously establishing the Cordover Scholar in the Arts scholarship, which has allowed me to attend MIT, the Free Software Foundation and the open source community for providing much of the software used for this thesis, Charles Gosdzinski for providing me with the opportunities that paved the way for my study of science, and my parents for encouragement and generous contributions of lentil soup and other necessities.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A *classsical computer* is a device based on the laws of classical mechanics and electrodynamics that carries out algorithms describable in terms of mathematical logic. A *quantum computer* is a device based on the laws of quantum mechanics capable of carrying out similar algorithms. The laws of quantum mechanics are appealing for the basis of a computational system, because they are believed to describe how nature behaves on a fundamental level. Furthermore, several quantum algorithms have been devised to solve computational problems more efficiently than any known classical algorithms. Such algorithms include the Deutsch-Jozsa problem, the Grover search problem, and the prime factorization of integers [1]. However, a large scale quantum computer has yet to be built. One difficulty in building a quantum computer, is that large scale systems tend to behave classically. Any such effect preventing a quantum computer from behaving as expected is generally referred to as *noise*, and methods for characterizing and correcting for noise are an important step towards large-scale quantum computation. In this thesis, I consider a particular type of noise in one model of quantum computation known as *Adiabatic quantum computation.*

Adiabatic quantum computation is an alternative to *standard quantum computation.* In all models of quantum computation, information is stored in quantum bits,

or *qubits*. These qubits are represented by vectors of the form:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1.1}$$

$$|\alpha|^2 + |\beta|^2 = 1, \tag{1.2}$$

where the orthonormal basis states $\{|0\rangle, |1\rangle\}$ are referred to as the *computational basis*. In standard quantum computation, qubits are manipulated by *quantum gates*, which are unitary operators that act on quantum states representing qubits. Standard quantum computation has been formulated in considerable detail [1], but many open questions remain. Alternatively, adiabatic quantum computation is a *Hamiltonian based* model of quantum computation, in which qubits are manipulated by changing the Hamiltonian of the quantum system composing the quantum computer.

The work done for this thesis is a numerical study of noise in a particular adiabatic quantum algorithm. The general formalism of adiabatic quantum computation and a discussion of noise are presented in chapters 2 and 3. The specific adiabatic algorithm studied numerically is presented in chapter 4. The methods and results of the numerical simulations are presented in chapters 5 and 6. The conclusions drawn from the results are summarized in chapter 7.

Chapter 2

Adiabatic Quantum Computation

As a Hamiltonian-based model, adiabatic quantum computation is performed by varying a parameter of the Hamiltonian of a quantum system. The parameter is varied in such a way as to evolve the state from a known gound state of one Hamiltonian to the unknown ground state of the other, which encodes to solution to a computational problem.

2.1 History of Adiabatic Quantum Computation

The first quantum algorithms achieved by adiabatic evolution were presented in [2], including an algorithm to solve the Grover search problem in the same running time as standard Grover algorithm [3]. Small adiabatic algorithms have been experimentally realized using nuclear magnetic resonance [4, 5]. Architectures for adiabatic quantum computation have also been proposed using superconducting qubits [6] and phase modulated laser pulses [7].

Although fault-tolerance techniques are well developed for standard quantum computation, they have yet to be fully explored for adiabatic quantum computation. Since it was first suggested that adiabatic quantum computation could be inherently resistant to certain types of noise [8], the topic has received much attention [9, 10, 11, 12, 13, 14, 15]. An error-correction scheme based on stabilizer codes has also been proposed [16].

Adiabatic quantum computation was shown to be as powerful as standard quantum computation in [17]. Given that adiabatic quantum computation appears to be resistant to certain types of noise, is as powerful as statndard quantum computation, and has already been realized on a small scale, it appears to be a good candidate for the realization of a large-scale quantum computer.

2.2 Formulation of Adiabatic Quantum Computation

The formulation of adiabatic quantum computation used in this thesis is based on that of [2].

2.2.1 The Adiabatic Theorem

Adiabatic quantum computation is based on the quantum *adiabatic theorem* [18]. The adiabatic theorem applies to Hamiltonians with a parameter, $s(t)$, that is varied slowly over a time period $0 \leq t \leq T$. For example, the parameter could be:

$$s(t) = \frac{t}{T}. \tag{2.1}$$

Let $H(s(t))$ be a time-dependent Hamiltonian with instantaneous eigenstates $|l; s(t)\rangle$ and ordered instantaneous eigenenergies $E_l(s(t))$ such that:

$$H(s(t))|l; s(t)\rangle = E_l(s(t))|l; s(t)\rangle \tag{2.2}$$

$$E_0(s(t)) < E_1(s(t)) < \ldots \tag{2.3}$$

Denoting the state at time t as $|\psi(t)\rangle$, the adiabatic theorem states that if

$$|\psi(t=0)\rangle = |l=0; s(t)=0\rangle \tag{2.4}$$

and

$$E_1(s(t)) - E_0(s(t)) > 0 \text{ for all } 0 \leq t \leq T, \tag{2.5}$$

then

$$\lim_{T\to\infty} |\langle 0; s(T)|\psi(T)\rangle| = 1. \tag{2.6}$$

Further analysis suggests that given

$$\gamma = \min_{0\leq t\leq T} (E_1(s(t)) - E_0(s(t))) \tag{2.7}$$

$$\mathcal{E} = \max_{0\leq t\leq T} \left| \left\langle l = 1; s(t) \left| \frac{dH}{ds}(s(t)) \right| l = 0; s(t) \right\rangle \right| \tag{2.8}$$

if

$$T \gg \frac{\mathcal{E}}{\gamma^2} \tag{2.9}$$

then

$$|\langle l = 0; s(T)|\psi(T)\rangle| \tag{2.10}$$

can be made arbitrarily close to 1.

2.2.2 Quantum Computation With the Adiabatic Theorem

Adiabatic quantum computation inovlves encoding a computational problem as a Hamiltonian, and finding its ground state using adiabatic time-evolution. This is achieved by preparing a system with an *initial Hamiltonian* H_B with an easily computable ground state, then varying a parameter of the Hamiltonian slowly to change it into the *problem Hamiltonian* H_P (the ground state of which encodes the solution to a problem). The adiabatic theorem implies that if the system is prepared in the (known) ground state of H_B, it will remain in the ground state as the Hamiltonian is slowly changed to H_P. The issue of encoding a problem as Hamiltonian is addressed in Chapter 4.

One possible choice for the initial Hamiltonian in a system with n qubits is:

$$H_B = \sum_{j=0}^{n-1} \frac{1}{2}(1 - \sigma_x^{(j)}). \tag{2.11}$$

The eigenstates and eigenvalues of σ_x are:

$$\sigma_x|+\rangle = \sigma_x \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = (1)|+\rangle \tag{2.12}$$

$$\sigma_x|-\rangle = \sigma_x \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = (-1)|-\rangle. \tag{2.13}$$

Since each term in (2.11) acts on a different qubit, they all commute and the ground state of H_B is the tensor product of the ground states of each term. The ground state of each term can be seen to be $|+\rangle$ with eigenvalue 0. The initial state is thus:

$$|\psi(0)\rangle = |+\rangle^{\otimes n} \tag{2.14}$$

$$= \frac{1}{2^{n/2}}(|0\rangle + |1\rangle)^{\otimes n}, \tag{2.15}$$

which should be easy to prepare.

The adiabatic algorithm can be applied to any slowly time-varying Hamiltonian beginning in H_B and ending in H_P. One such Hamiltonian is:

$$H(t) = \left(1 - \frac{t}{T}\right) H_B + \frac{t}{T} H_P. \tag{2.16}$$

Letting $s = t/T$, (2.16) is equivalent to:

$$\tilde{H}(s) = (1 - s)H_B + sH_P. \tag{2.17}$$

Since $\tilde{H}(0) = H_B$,

$$|\psi(0)\rangle = |l = 0; s(0)\rangle, \tag{2.18}$$

satisfying (2.4). Denoting the (time-independent) ground state of H_P as $|\phi\rangle = |l = 0; s(T)\rangle$, so as long as (2.9) is satisfied, (2.10) implies that the system ends in the

ground state of the problem Hamiltonian:

$$|\langle\psi(T)|\phi\rangle| \approx 1. \tag{2.19}$$

2.2.3 Running Time

Since adiabatic time-evolution is dependent upon (2.9) begin satisfied, it defines the *running time* T of the algorithm. Equation (2.9) gives a lower bound on the running time proportional to the squared inverse of the minimum gap γ between the ground and first excited energy level at any point in the computation. Calculating the minimum gap for an adiabatic quantum algorithm thus gives a lower bound on the running time. However, as of the writing of this thesis, no general method exists to find the gap.

2.3 2-Local Hamiltonians

A Hamiltonian is called *k-local* if it can be written as a sum of products of Pauli operators with at most k non-identity operators in each product. Hamiltonians that are k-local for some small k, which I refer to simply as *local* Hamiltonians, are important because interactions involving many particles are difficult to realize physically. The work in this thesis focuses on 2-local Hamiltonians in particular.

Although 2-local Hamiltonians are a small subset of all Hamiltonians, they are still computationally powerful in the context of adiabatic quantum computation. In fact, it has been shown that adiabatic quantum computation with 2-local Hamiltonians is computationally equivalent to standard quantum computation [17, 19].

Chapter 3

Noise in Adiabatic Quantum Computation

Any physical implementation of a quantum computer will be imperfect and will have to stand up to unideal conditions, generally known as *noise*. This chapter details possible sources of noise, possible approaches to tolerating noise, and the parameters that are likely to be important for handling noise in adiabatic quantum computation.

3.1 Sources of Noise

There are two sources of noise likely to be important in a quantum computer: the environment, and control error. The environment could affect a quantum computer in many ways. For instance, the qubits in the computer could be coupled to particles outside of the quantum computer, or there could be interactions between the qubits beyond those considered in designing the quantum computer. Furthermore, the adiabatic theorem is formulated for closed systems and coupling to an outside environment could conceivably result in a deviation from adiabatic time-evolution. Even in the absence of environmental noise, any physical implementation of a quantum computer will only meet its specifications to some tolerance. The imperfections in the implementation of a computation will cause noise, which I refer to as *control error*.

3.2 Fault Tolerance

Given that some level of noise is inevitable in any physical system, it is desireable to have methods for guaranteeing a successful computation even in the presence of noise. Such methods are generally known as *fault-tolerance* techniques. Fault-tolerance techniques have been well developed in both classical computation and standard quantum computation [1]. In both cases, noise is protected against using *redundancy*. More specifically, the bits/qubits involved in the computation are encoded as logical bits/qubits. Each logical bit/qubit is itself comprised of many physical bits/qubits, creating redundancy. Computational operations are then performed on the logical bits/qubits. The goal of this encoding procedure is to allow some of the fundamental building blocks of the system to behave incorrectly, an event known as a *fault*, leading to a discrepancy in the physical bits/qubits known as an *error*, without the computation producing an incorrect result, known as a *failure*. There are two types of such encoding schemes: *error correcting codes* which can correct errors, and *error detecting codes* which can be used to determine when the physical bits/qubits are in error, but not to correct them.

The goal of fault-tolerance is to allow a computation of any size to succeed with high probability when reasonable conditions on the amount of noise are met. In standard quantum computation, the conditions for fault-tolerance are summarized by the *threshold theorem* [1]. The threshold theorem is centered on the idea of *concatenated codes*. In a concatenated code, if a single application of an error correcting code is not sufficient, then the encoded logical bits/qubits are themselves encoded, and so on. Each round of encoding provides redundancy, but also introduces more components that can have errors, so it is not obvious when this scheme is beneficial and when it is harmful. In concatenated codes, operations on the unencoded bits/qubits are used to construct logical operations on the encoded bits/qubits. The threshold theorem states that for any error correcting code, there exists a constant threshold p_0, such that if the probability of a single fundamental operation failing p is less than p_0, then

the probability of the an encoded logical operation can be made to be

$$p_{\text{fail}} < p_0 \left(\frac{p}{p_0}\right)^{2^k}, \tag{3.1}$$

where k is the number of code concatenations used. One possible goal for fault-tolerance in quantum computation is to find an analagous result, allowing an arabitrary computation to be done adiabatically regardless of size as long as the probability of a single qubit being in error is less than some constant. However, other techniques may also be possible, such as *error-resistance* which preemptively stops errors from occurring at all.

3.3 Modelling Noise as a Perturbation

Before attempting to correct or prevent errors, it is necessary to have a model for them. The model considered in this thesis is a *perturbative model* in which noise is treated as a perturbation added to the Hamiltonian.

A perturbative noise model is useful for modelling some types of noise and not others. There are several ways in which a perturbation can be considered during the process of an adiabatic algorithm, and each method is relevant for different types of noise. A time-dependent perturbation could be applied during the entire adiabatic evolution, a constant perturbation could be applied during the entire evolution, or a constant perturbation could be considered only at the end of the evolution.

A time-dependent perturbation is a good model for statistical fluctuations that could be caused by environmental noise. This type of perturbation is particularly complicated because the rate and manner in which the perturbation varies is important in studying its effects. On the other hand, a constant perturbation during the entire time-evolution is a good model for systematic noise, such as control error. Though valid error models, the work in this thesis is focused on another model.

A constant perturbation at the end of the time-evolution is relevant to both systematic and statistical noise, under the assumption that time-evolution remained

adiabatic throughout the running of the algorithm. Assuming that the system began in the ground state of the noisy Hamiltonian and evolved adiabatically, at the end of the computation the adiabatic theorem implies that the state of the system will be the ground state of the noisy final Hamltonian. The noisy final Hamiltonian may thus be modelled as the problem Hamiltonian plus a perturbation regarless of whether the perturbation is a result of a constant or time-varying process earlier in the computation.

3.3.1 Structure of the Perturbation

The structure of a perturbation also determines what types of noise can be used to model. For instance, there is no reason to expect noise from the environment to be local. On the other hand, it is reasonable to assume that control error will not change the structure of the interactions between qubits, just their magnitude and types. In other words, control error in a k-local Hamiltonian can be reasonably modelled as a k-local perturbation with interactions between the same groups of qubits as the unperturbed Hamiltonian. The noise considered in this thesis is a perturbation of the problem Hamiltonian with the same structure, and is thus primarily a model of control error.

3.3.2 Noise Magnitude

Before continuing, the topic of classifying the magnitude of a perturbation requires further discussion. For example, it is not immediately clear whether $\sigma_x^{(1)}$ and $\sigma_x^{(1)}\sigma_x^{(2)}$ should be considered to be the same magnitude. The question of the magnitude of noise is really a question of what the appropriate *matrix norm* on the corresponding perturbation is. When considering the perturbation V to be chosen uniformly at random, a reasonable norm is the expectation of the magnitude of V acting on a random vector $\vec{v}$:

$$\| V \|^2 = \langle (V\vec{v})^\dagger (V\vec{v}) \rangle. \tag{3.2}$$

It can be shown that (3.2) gives a matrix norm equal to the root-mean-square of the eigenvalues of the matrix:

$$\| V \| = \sqrt{\frac{\lambda_0^2 + \lambda_1^2 + \ldots + \lambda_{d-1}^2}{d}} \tag{3.3}$$

$$= \sqrt{\frac{1}{d}\mathrm{Tr}(V^2)}, \tag{3.4}$$

where d is the dimension of V.

3.4 Relevant Properties of the Problem Hamiltonian

Several properties of the problem Hamiltonian can be used to gain insight into the bahvior of noise. The relevance of several of these properties can be seen using perturbation theory [20]. For an n-qubit Hamiltonian H_0 with perturbation V of magnitude ϵ and eigenstates $|k\rangle$ with energies E_k:

$$H = H_0 + \epsilon V, \tag{3.5}$$

the non-normalized projector onto the perurbed ground state $|\tilde{0}\rangle$ is given by:

$$S = \sum_{k=1}^{2^n-1} \frac{1}{E_0 - E_k} |k\rangle\langle k| \tag{3.6}$$

$$S^0 = -|0\rangle\langle 0| \tag{3.7}$$

$$A^{(m)} = (-1)^{m-1} \sum_{k_1+k_2+\ldots+k_{m+1}=m} S^{k_1} V S^{k_2} V \cdots V S^{k_{m+1}} \tag{3.8}$$

$$|\tilde{0}\rangle\langle\tilde{0}| = |0\rangle\langle 0| + \sum_{m=1}^{\infty} \epsilon^m A^m. \tag{3.9}$$

One relevant property is the gap, γ, between the ground state and first excited state. Note that the mth order term in (3.9) contains m powers of S, which scales each original energy eigenstate by the inverse of the difference between its energy

and the ground state's. The energy gap between the ground state and state $|k\rangle$ must be greater than or equal to γ, so the magnitude of the mth order correction to the ground state is bounded by

$$\left(\frac{1}{\gamma}\right)^n. \tag{3.10}$$

The gap is also important for thermodynamic reasons. At nonzero temperature T, if the system is in equilibrium, the probability of the ground state being thermally excited to state $|j\rangle$ is proportional to:

$$e^{-\frac{E_j - E_0}{k_B T}}. \tag{3.11}$$

The most probable state to be thermally excited to is thus the first excited state with probability:

$$e^{-\frac{\gamma}{k_B T}}. \tag{3.12}$$

Another important property of the problem Hamiltonian is the density of states in its energy spectrum. The density of states is important for the simple reason that even if each state in the first excited energy level has a small amplitude in the perturbed ground state, if there are many such states then the probability of the state being excited to the first energy level may still be high.

Finally, the locality of the perturbation gives further insight into the effects of noise. A 2-local perturbation acting on a basis state $|k\rangle$ produces a linear combination of basis states no more than Hamming distance 2 away from $|k\rangle$. As (3.8) shows, $A^{(m)}$ has $m+1$ different "k" variables that must sum to m in each term, which implies that at least one must be equal to 0. Because S^0 is the projector onto the unperturbed ground state, Equation (3.9) implies that for a 2-local perturbation, $\langle k|\tilde{0}\rangle\langle\tilde{0}|k\rangle$ is non-zero only for terms of order d and higher, where d is the Hamming distance between $|0\rangle$ and $|k\rangle$. Generally speaking, the Hamming distance between the unperturbed ground state and another state $|k\rangle$ determines the order of perturbation theory at which $|k\rangle$ may become a component of the perturbed ground state due to noise.

3.5 Success Critereon

To classify the effects of noise on the success of an adiabatic quantum computation, it is necessary to have a critereon for determining whether a computation was successful. Since the ideal result of the computation is the ground state of the problem Hamiltonian, the success critereon should somehow compare the ground state of the noisy Hamiltonian to that of the ideal problem Hamiltonian. The standard measure of the "distance" between two states is the *fidelity* [1]. The fideltity between two pure states $|\psi\rangle$ and $|\phi\rangle$ is given by:

$$F(|\psi\rangle, |\phi\rangle) = |\langle\psi|\phi\rangle|. \tag{3.13}$$

One reasonable definition of a successful computation would be when $F(|0\rangle, |\tilde{0}\rangle) > 1/2$. When this critereon is met, performing the computation k times and choosing the most commonly observed computational basis state, called *majority voting*, will give the ground state of the ideal problem Hamiltonian with high probability.

It is more realistic to expect that majority voting will be performed on a qubit by qubit basis, motivating another success metric. Letting $|\phi_k\rangle$ be the state of qubit k in the ground state of the ideal problem hamiltonian, the probability of observing qubit k in state $|\phi\rangle$ is given by:

$$p_k = \langle\phi|(I \otimes |\phi_k\rangle\langle\phi_k| \otimes I)|\phi\rangle. \tag{3.14}$$

The probability of a successful computation is thus the product of (3.14) for each qubit:

$$p_{\text{success}} = \prod_{k=0}^{n-1} p_k. \tag{3.15}$$

In this thesis, a computation is considered successful with the value of (3.15) is greater than $1/2$.

Chapter 4

Agree on a Line

The adiabatic algorithm studied in this thesis is one designed to solve a the *agree* problem on a line of qubits. The agree problem is a constraint satisfaction problem in which the value of each qubit is constrained to be equal to that of its nearest neighbors, and a single qubit is constrained to have the value $|0\rangle$. The solution is thus always the bit string of zeros. The agree problem can be solved using an adiabatic algorithm with a problem hamiltonian consisting of 2-local couplings between nearest neighbors, which I shall refer to as the *neighbor-coupling* algorithm. Neighbor-coupling is useful for numerical studies of errors for several reasons. Although nieghbor-coupling is particularly simple, it contains many features characteristic of more complicated algorithms. Furthermore, the energy spectrum of the problem Hamiltonian is easily calculated and exhibits properties that could be important in classifying the effects of error.

4.1 Problem Hamiltonian

The neighbor-coupling problem Hamiltonian on a line of n qubits consists of a term for each pair of nearest neighbors and a single term for qubit 0. The nearest neighbor terms assign an energy penalty if the neighbors disagree. The term for qubit 0 assigns

an energy penalty for the state $|1\rangle$. The resulting Hamiltonian is:

$$\frac{1}{2}(I - \sigma_z^{(0)}) + \frac{1}{2}\sum_{k=0}^{n-2}(I - \sigma_z^{(k)}\sigma_z^{(k+1)}). \tag{4.1}$$

4.2 Analysis of Problem Hamiltonian

Although quite simple, the neighbor-coupling algorithm is closely related to significantly more complex algorithms. A simple problem Hamiltonian is desirable for comparisons of its properties at different numbers of qubits. For neighbor-coupling, adding a qubit simply means adding a term to the problem Hamiltonian. Although neighbor-coupling is simple, it is still potentially useful to study. Each term in the Hamiltonian corresponds to a constraint in the agree problem. Such problems as 3-SAT are also constraint satisfaction problems that can be solved adiabatically (although not necessarily efficiently) by adding a term to the problem Hamiltonian for each constraint. This suggests that although the neighbor-coupling algorithm is trivial, its behavior may give insight into the behavior of more complicated algorithms, even those that could conceivably solve NP-complete problems efficiently.

The energy spectrum of the problem Hamiltonian is easy to analyze because all terms are diagonal in the computational basis, and therefore commute. Furthermore, the eigenenergies of each term are 0 for computational basis states that satisfy the corresponding constraint and 1 for states that do no. Given these facts, the eigenstates are the computational basis states and their corresponding energies are simply the number of unsatisfied constraints, and the degeneracy of each energy level is the number of distinct ways to violate a given number of constraints.

The ground state and first excited states have some interesting properties. The ground state is the unique computational basis states satisfying all constraints: $|0\rangle^{\otimes n}$. The next energy level contains all computational basis states violating a single constraint. There are two types of such states. There is a single state, $|1\rangle^{\otimes n}$ violating only the constraint on qubit 0. There are also $n-1$ states violating a single constraint on nearest neighbors. The state which violates the constraint on qubits k and $k+1$ is

given by $|0\rangle^k|1\rangle^{n-k}$. These lower energy states are interesting because they represent two opposite extremes. In one case, there is a single state, very far from the ground state in Hamming distance. In the other case there are many states very close to the ground state in Hamming distance.

Chapter 5

Simulation Techniques

In this thesis, I have used Monte Carlo simulations to study the effects of control error on an adiabatic quantum algorithm. A closed-form mathematical description of such effects would be very useful, but is not currently known and might not exist at all. In the absence of a closed-form description, Monte Carlo techniques provide an alternative means of gaining insight into the effects of control error. In general, simulations of quantum computers appear to require resources exponential in the number of qubits. However, small numbers of qubits can be simulated on classical computers, and may still allow relevant trends to be identified. I have developed an optimized algorithm to simulate the effects of control error on as many qubits as possible, given time and space limitiations, in order to determine how these effects scale with the number of qubits.

5.1 Overview

In this thesis, control error is treated as a perturbation of the ideal problem Hamiltonian. The noisy problem Hamiltonian H is thus:

$$H = H_p + V, \tag{5.1}$$

where H_p is the ideal problem Hamiltonian and V is a perturbation. An alternative model would be to consider the noisy Hamiltonian to be a unitary transformation of the ideal Hamiltonian:

$$H = U^\dagger H_p U. \tag{5.2}$$

However, (5.1) is more general and includes Hamiltonians of the form (5.2). This can be seen as follows:

$$\begin{aligned} H &= U^\dagger H_p U && (5.3)\\ &= H_p - H_p + U^\dagger H_p U && (5.4)\\ &= H_p + V && (5.5)\\ V &= U^\dagger H_p U - H. && (5.6) \end{aligned}$$

If it is assumed that the adiabatic theorem still governs the time-evolution of the system, the final state of the system is the ground state of the perturbed Hamiltonian. For this thesis, I assume that time-evolution is still adiabatic. Characterizing the effects of control error is thus a matter of finding the ground state of the perturbed problem Hamiltonian and comparing it to that of the ideal problem Hamiltonian.

5.1.1 Space Complexity

A system of n qubits has a Hilbert space of dimension 2^n, so in general, an n-qubit Hamiltonian is a $2^n \times 2^n$ Hermitian matrix. In its most explicit form, such a matrix has 4^n complex entries, which may be specified by $2 \cdot 4^n$ real numbers (the real and imaginary parts of the complex entries). However, the amount of information necessary to represent a Hamiltonian can be significantly less, especially for some classes of Hamiltonians. Noting that Hamiltonians must be Hermitian, the entry in row i and column j, $H_{i,j}$, must be the complex conjugate of $H_{j,i}$. Each diagonal entry is thus real, and may be represented by a single real number. Each off diagonal entry $H_{i,j}$ may be specified by two real numbers. Thus, once $H_{i,j}$ is specified, $H_{j,i}$ is also specified, so each off diagonal pair my be specified by two real numbers. The number

of real numbers required to specify an arbitrary Hermitian matrix is thus just the number of entries, 4^n. For example, a 1-qubit Hamiltonian can be specified by 4 real numbers a, b, c, d as follows:

$$H = \begin{bmatrix} a & b+ic \\ b-ic & d \end{bmatrix}. \tag{5.7}$$

Local Hamiltonians can be specified with even less information. A Hamiltonian is fully specified by how it acts on each computational basis vector. The action of a Hamiltonian H on a basis vector $|m\rangle$ can be seen easily using the outer product notation:

$$H = \sum_{j=0}^{n-1}\sum_{k=0}^{n-1} H_{j,k}|j\rangle\langle k| \tag{5.8}$$

$$H|m\rangle = \sum_{j=0}^{n-1}\sum_{k=0}^{n-1} H_{j,k}|j\rangle\langle k|m\rangle \tag{5.9}$$

$$= \sum_{j=0}^{n-1} H_{j,m}|j\rangle. \tag{5.10}$$

In other words, when a Hamiltonian acts on a computational basis state $|m\rangle$, the components of each computational basis state in the resulting state vector are given by column m of H. A 2-local Hamiltonian is a sum of terms with at most two non-identity Pauli operators, so when acting on a basis state, the result must be a sum of basis states with two or fewer bits different from the original state. Therefore, any column of an n-qubit, 2-local Hamiltonian has at most

$$\binom{n}{2} + \binom{n}{1} + \binom{n}{0} = \frac{1}{2}n(n-1) + n + 1 \tag{5.11}$$

$$= \frac{1}{2}n^2 + \frac{1}{2}n + 1 \tag{5.12}$$

$$\tag{5.13}$$

nonzero entries. The entire Hamiltonian thus has at most

$$2^n \left(\frac{1}{2} n^2 + \frac{1}{2} n + 1 \right) \tag{5.14}$$

entries. Although still exponential in the number of qubits, this improvement is very significant at small numbers of qubits.

The number of real numbers necessary to fully specify a 2-local Hamiltonian is not necessarily the same as the number of nonzero entries. In fact, it is much less. Since an n-qubit 2-local Hamiltonian is a sum of products of two Pauli operators, the Hamiltonian is fully specified by specifying the real coefficient[1] multiplying every possible product of two Pauli operators. There are $\binom{n}{2}$ distinct pairs of qubits. For each pair, there are 3 choices for the Pauli operator acting on the lower index qubit, and 3 for the operator acting on the higher index qubit. The number of real numbers necessary to specify an n-qubit 2-local Hamiltonian is thus:

$$9\binom{n}{2} = \frac{9}{2} n(n-1). \tag{5.15}$$

5.1.2 Time Complexity

The time required to construct a Hamiltonian is dependent upon how it is represented. Representing a Hamiltonian as a 2-dimensional array necessarily requires each entry to be set at least once, taking $\Omega(4^n)$ worst-case time to construct n qubits. However, if the Hamiltonian is stored as a sparse matrix, only the nonzero entries need to be set. Given (5.14), the worst-case time necessary to construct a 2-local Hamiltonian as a sparse matrix is thus

$$\Omega\left(2^n \left(\frac{1}{2} n^2 + \frac{1}{2} n + 1 \right) \right). \tag{5.16}$$

Again, for small numbers of qubits, this is a significant improvement.

[1] If the coefficient had an imaginary component, the resulting matrix would not be Hermitian.

5.2 Implementation

All simulations for this thesis were written in MATLAB[2] and optimized for 2-local Hamiltonians. The ideal problem Hamiltonian and a random perturbation are first generated in a representation that requires size polynomial in the number of qubits and that allows additions to be done in polynomial time. The noisy Hamiltonian is then constructed as a sparse matrix from these, requiring time and space exponential in the number of qubits. Finally the lowest eigenstate is found using standard techniques.

5.2.1 Pauli Representation

The noisy Hamiltonian is initially constructed in a representation consisting of the coefficients of each possible 2-local Pauli operator, which I shall refer to as the *Pauli representation.* The most general form of an n-qubit 2-local Hamiltonian is:

$$H = \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} \sum_{l=1}^{3} \sum_{m=l}^{3} c_{j,k,l,m} \sigma_l^{(j)} \sigma_m^{(m)}. \tag{5.17}$$

Note that the index of the fourth sum begins at l to avoid double counting. The coefficients $c_{j,k,l,m}$ are stored in six matrices: $C^{XX}, C^{XY}, C^{XZ}, C^{YY}, C^{YZ}, C^{ZZ}$. The matrix C^{XX} contains the coefficients $c_{j,k,1,1}$, C^{XY} contains the coefficients $c_{j,k,1,2}$, and so on. Row i and column j of each matrix correspond to qubits i and j. For instance, the element $C^{XY}_{j,k}$ is the coefficient $c_{j,k,1,2}$. Note that when $j = k$, the coefficent corresponds to the product of two Pauli operators on the same qubit. The $j = k$ case is used to achieve the identity operator and 1-local Pauli operators.

The manipulation of the Pauli representation matrices is hidden behind an abstraction consisting of the functions `mk2loc`, `add2loc`, and `add1loc`. The function `mk2loc` creates and returns a data structure containing the Pauli representation of an n-qubit, 2-local Hamiltonian. The function `add2loc` returns a copy of a Pauli representation structure with a 2-local term added to it. The function `add1loc` returns a

[2]the code is available at `http://alum.mit.edu/www/elplatt/programs`.

Operator	$\|0\rangle$ Multiplier	$\|1\rangle$ Multiplier	Bit Flip?
σ_x	1	1	yes
σ_y	i	$-i$	yes
σ_z	1	-1	no

Table 5.1: The action of the Pauli operators on computational basis states in terms of bit flips and scalar multiplications.

copy of a Pauli representation structure with a 1-local term added to it. The ideal problem Hamiltonian and random perturbation are created using these functions.

5.2.2 Sparse Matrix Construction

Once the noisy Hamiltonian has been constructed as a Pauli representation data structure, it must be converted into a sparse matrix to find the lowest eigenstate. The algorithm used for the conversion is based on the observation that a Pauli operator's action on a computational basis state can be described as a conditional scalar multiplication and/or a bit flip. The action of the Pauli operators in these terms is summarized in Table 5.1. The algorithm begins with an empty $2^n \times 2^n$ sparse matrix H. It then loops through each computational basis state $|k\rangle$, qubit pair, and 2-local operator type. For each combination, it performs the necessary bit flips and scalar multiplications to find the resulting basis state $|j\rangle$ and its coefficient, then adds the coefficient to the sparse matrix element $H_{j,k}$. Pseudocode for the algorithm used to construct the sparse matrix is given in Figure 5.2.2. Once again, the code used in this thesis hides the algoirithm behind and abstraction provided by the `matrix2loc` procedure.

After the sparse matrix has benn constructed, all that remains is to find the eigenvector corresponding to the smallest eigenvalue. I use the standard MATLAB procedure `eigs` to determine the ground state eigenvector of the final sparse matrix.

```
allocate sparse matrix H

for each basis state |j⟩
  for each qubit l
    for each qubit m

      z ← C_{l,m}^{YX}

      // apply multiplication corresponding to σ_y
      if bit l of j equals 0 then
        z ← iz
      else
        z ← −iz
      end if

      // apply bit flip corresponding to σ_y
      |k⟩ ← flip bit l of |j⟩

      // apply bit flip corresponding to σ_x
      |k⟩ ← flip bit m of |k⟩

      // update sparse matrix
      H_{j,k} ← H_{j,k} + z

      // repeat for other Pauli operators
      ...

    end for
  end for
end for
```

Figure 5-1: Pseudocode for an algorithm to construct a sparse matrix from a Pauli operator representation of a 2-local Hamiltonian.

5.3 Simulations Performed

The techniques described in this chapter were used to determine the fraction of successful runs of the neighbor-coupling algorithm for agree on a line for different types of noise, different numbers of qubits and different noise magnitudes. Simulations were performed with 1-local noise, and again with 2-local noise. Qubit numbers ranged between 3 and 9. For each type of noise and number of qubits, many trials were carried out for a range of noise magnitudes. The results of these simulations are presented and analyzed in the next chapter.

Chapter 6

Simulation Results and Analysis

This chapter presents results from the simulation techniques described in the previous chapter applied to determine how the probability of a successful computation varies with noise magnitude and the number of qubits. A summary of the simulations run is given in Table 6.1.

6.1 Success Fraction vs Noise Magnitude

Figures 6-1 and 6-2 show the fraction of successful trials, f, for 1-local and 2-local noise respectively. Each trial yielded a binary value corresponding to whether the simulated algorithm would find the solution to the agree problem. The success fraction was approximated as:

$$f = \frac{k}{N}, \tag{6.1}$$

where k is the number of successful trials, and N is the total number of trials. The error bars shown were determined by assuming that at each noise magnitude, there is some fixed probability of a successful computation, p. A series of several trials is thus described by a binomial distribution[21] with variance:

$$\sigma_b^2 = n^2 p(1-p), \tag{6.2}$$

k-Local Noise	Qubits	Min Noise	Max Noise	Divisions	Trials / Division
1	3	0	2.12	30	2500
1	4	0	2.12	30	1000
1	5	0	2.12	30	500
1	6	0	2.12	30	300
1	7	0	2.12	30	250
1	8	0	2.12	15	250
1	9	0	2.12	15	150
1	3	70.7	70.7	1	2500
1	4	70.7	70.7	1	1000
1	5	70.7	70.7	1	500
1	6	70.7	70.7	1	400
1	7	70.7	70.7	1	250
1	8	70.7	70.7	1	200
1	9	70.7	70.7	1	150
2	3	0.5	3	30	2500
2	4	0.5	3	30	1000
2	5	0.5	3	30	500
2	6	0.5	3	30	300
2	7	0.5	3	30	250
2	8	0.5	3	15	200
2	9	0.5	3	15	150
2	3	50	50	1	2500
2	4	50	50	1	1000
2	5	50	50	1	500
2	6	50	50	1	300
2	7	50	50	1	250
2	8	50	50	1	150
2	9	50	50	1	150

Table 6.1: Summary of simulations performed.

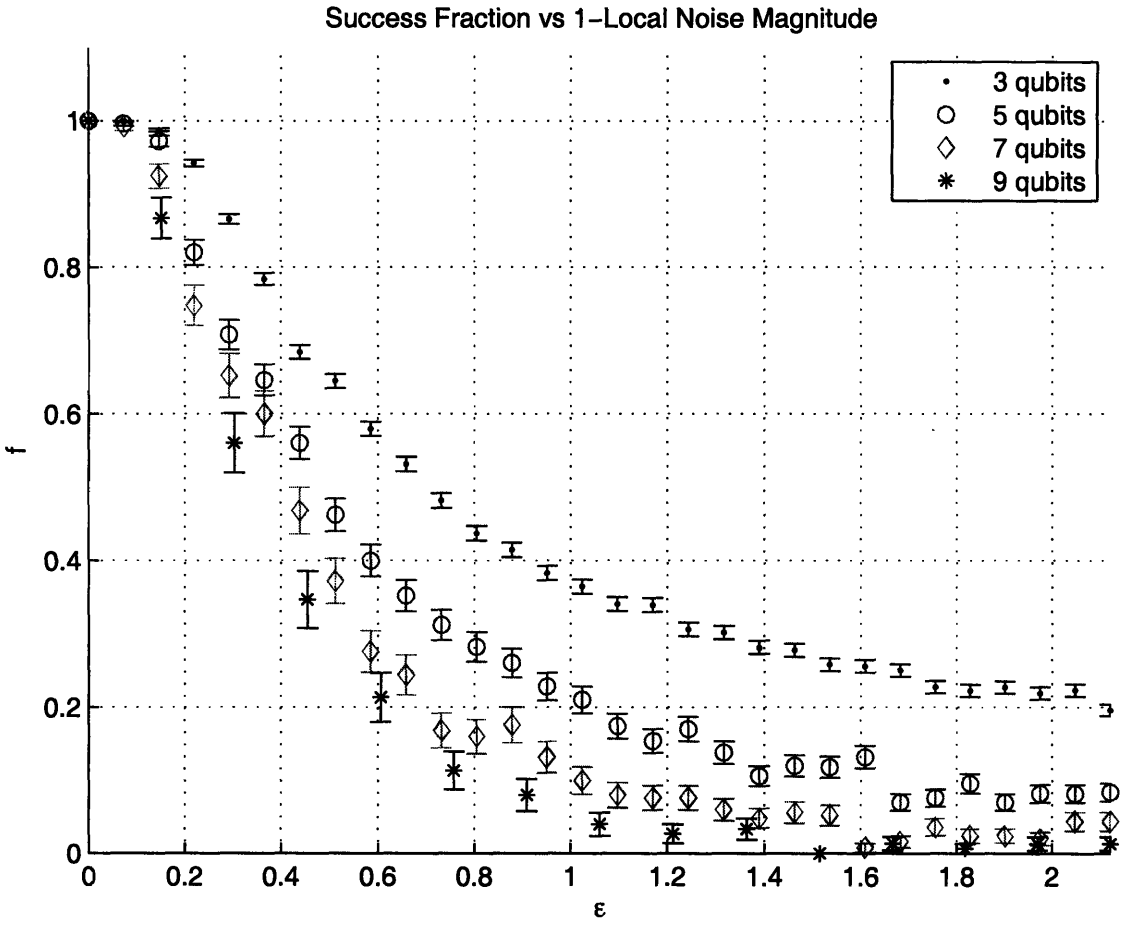

Figure 6-1: Success fraction vs noise magnitude for 1-local noise.

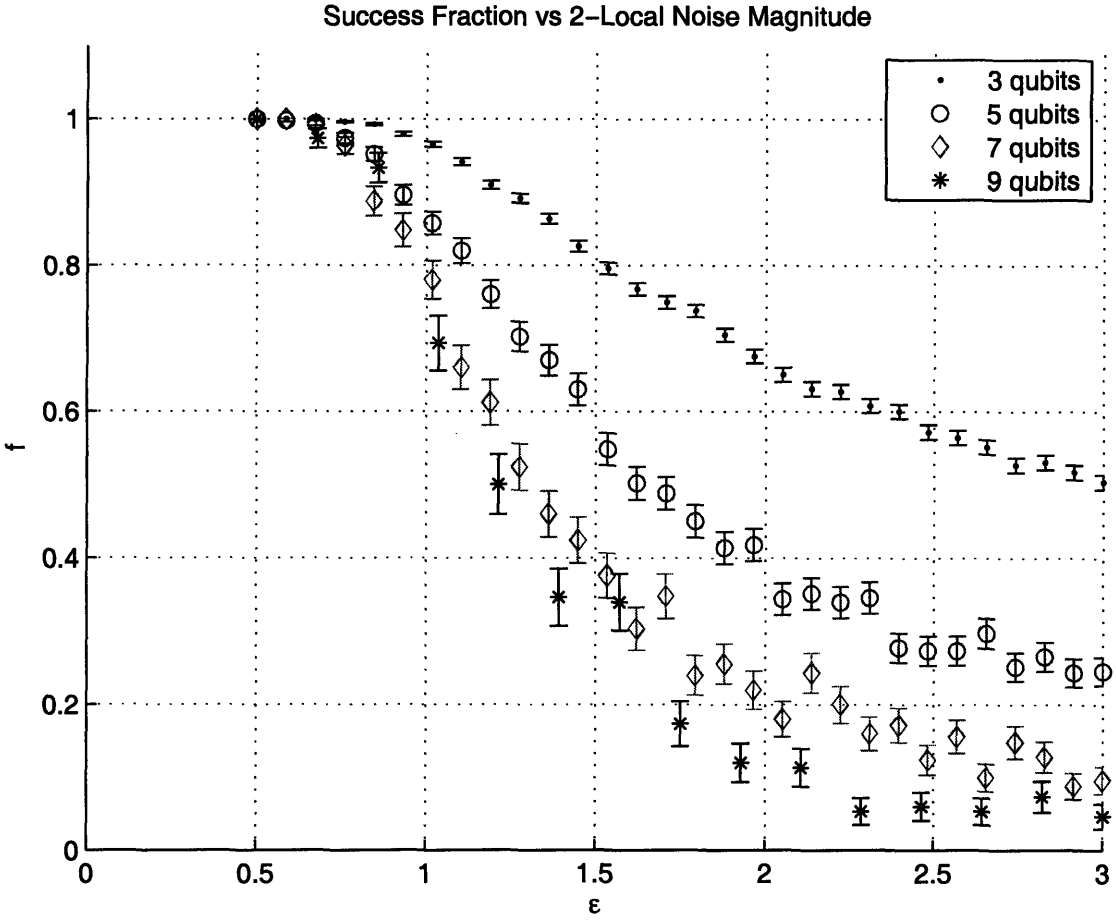

Figure 6-2: Success fraction vs noise magnitude for 2-local noise.

where n is the number of simulations. The variance of the estimate of the mean from n trials is given by [21]:

$$\sigma^2 = \frac{\sigma_b^2}{n}. \tag{6.3}$$

Combining (6.2) with (6.3), and assuming $p \approx f$, the standard deviation of the estimate is:

$$\sigma_f = \sqrt{\frac{f(1-f)}{n}}. \tag{6.4}$$

One inference that can be made from the simulation data is that for large noise magnitudes, the fraction of successful trials approaches a constant. Classically, large random noise would imply that the value of each bit was random. If this is also the case in the simulations, one would expect the probability of computing a given bit correctly to be $1/2$ which gives the limiting success probability:

$$\lim_{\epsilon \to \infty} p_{\text{success}} = \left(\frac{1}{2}\right)^n. \tag{6.5}$$

Figures 6-3 and 6-4 show the high-noise success fraction as a function of the number of qubits for 1-local and 2-local noise. The 1-local noise data matches 6.5 well. However, the 2-local noise data suggests a limiting success fraction of:

$$\lim_{\epsilon \to \infty} p_{\text{success}} = \left(\frac{1}{2}\right)^{n-1}. \tag{6.6}$$

Figure 6-2 also demonstrates that 2-local noise does not significantly hinder the computation until near $\epsilon = 0.5$. Both behaviors of the simulations of 2-local noise can be explained by considering the neighbor-coupling algorithm in more detail.

Just as 2-local terms are used in the neighbor-coupling algorithm to enforce agreement between neighboring qubits, 2-local noise terms in the algorithm affect only the agreement of neighboring qubits, not their individual values. In other words, 2-local noise terms do not bias individual qubit values. Since the neighbor-coupling algorithm with 2-local noise has a single 1-local term (on qubit 0), no other terms in the Hamiltonian bias the value of qubit 0 and it is always computed correctly. The high-noise probability of success given by (6.6) now makes sense because only $n-1$ qubits

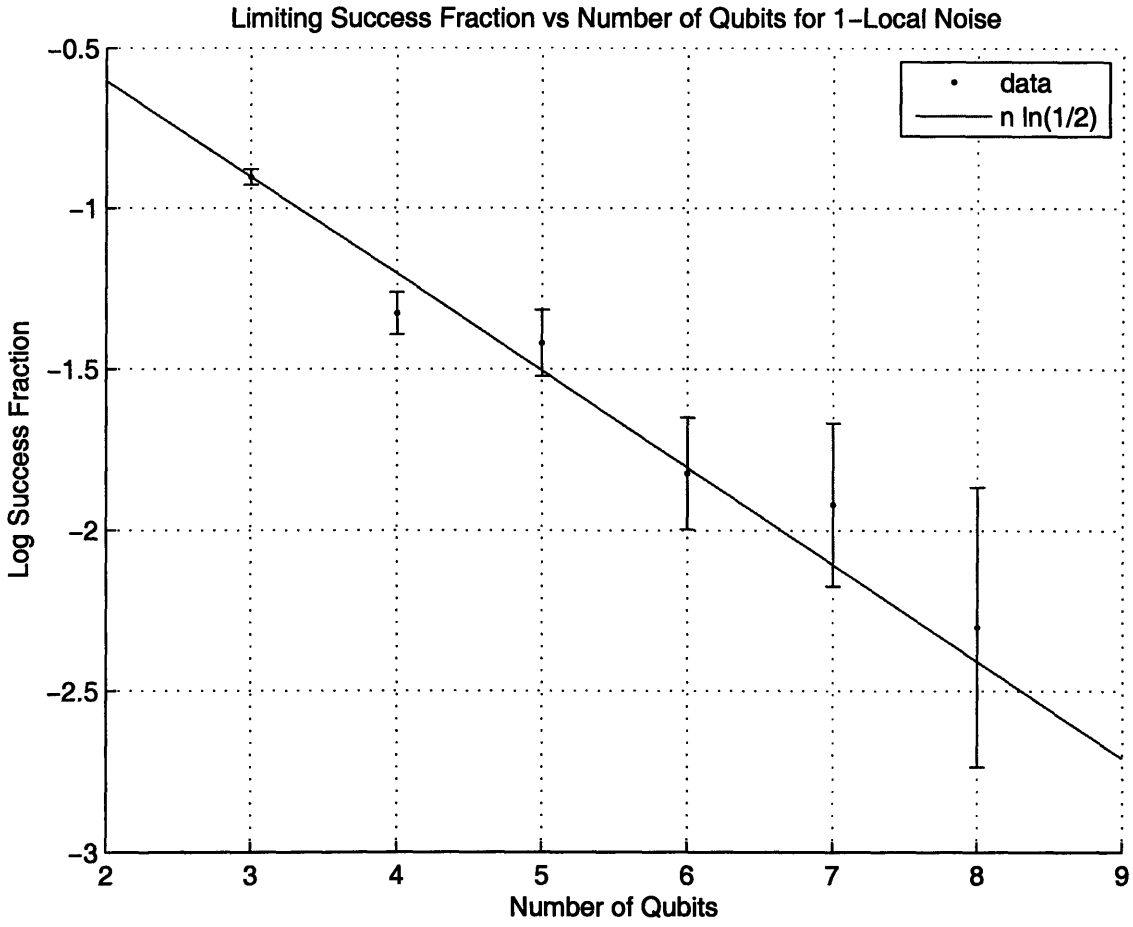

Figure 6-3: High-noise limiting success fraction for 1-local noise plotted on a semi-log scale.

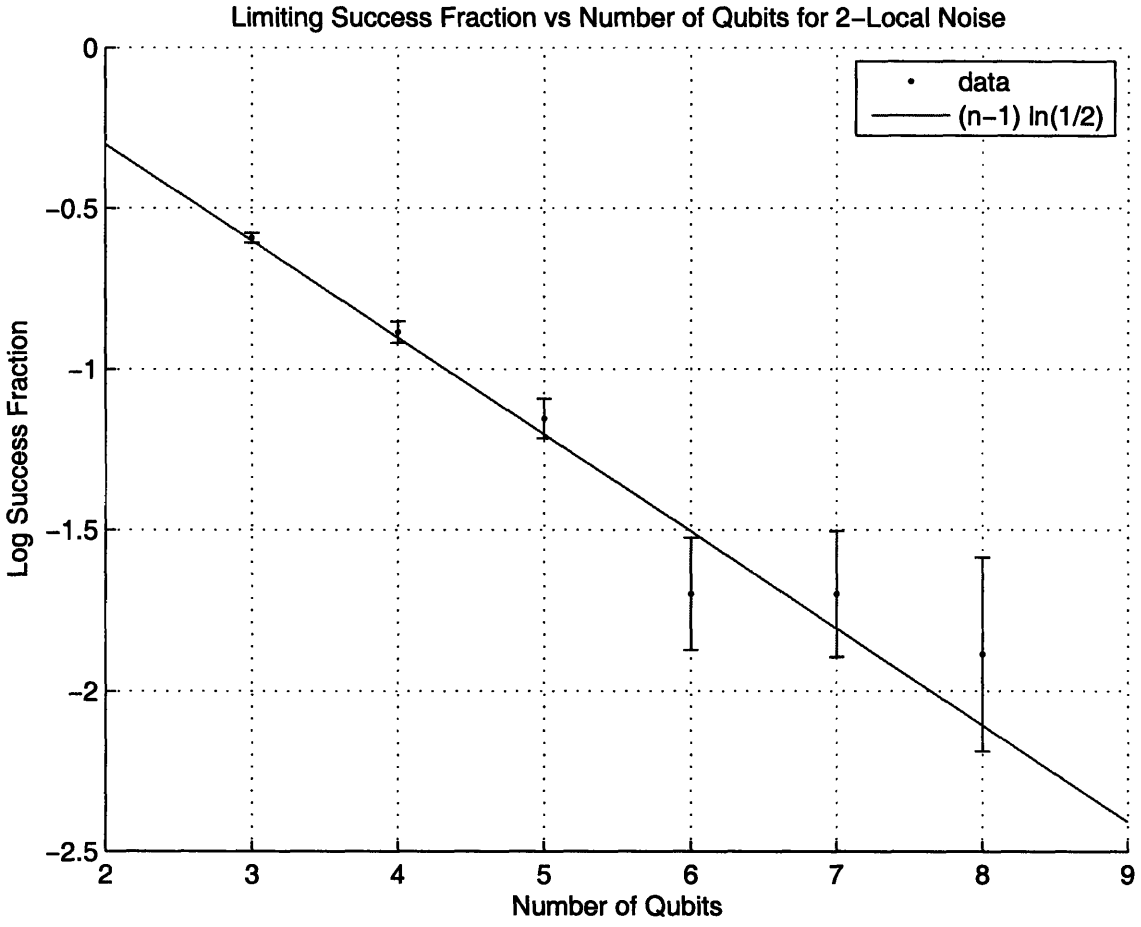

Figure 6-4: High-noise limiting success fraction for 2-local noise plotted on a semi-log scale.

have random values. For the same reason, the value of qubit 1 is only biased by its coupling to qubit 0, so its value can only be incorrect when the noise magnitude is greater than the magnitude of the coupling term between qubits 0 and 1, explaining the behavoior in Figure 6-2.

6.2 Success Threshold vs Qubit Number

The simulation data show how the success probability of the algorithm varies with noise, but the important question for fault-tolerance is how the success probability varies with the number of qubits. If there is no small finite noise magnitude for which an arbitrarily large computation can succeed, the algorithm is not inherently fault-tolerant. To determine how the the success probability scales with the number of qubits, I have found the largest allowable noise, or *success threshold*, for a successful computation for a range of qubit numbers.

Since the success probability must be estimated by repeated simulations at a fixed noise magnitude, the success thresholds were determined by fitting a function to the success fraction curves and numerically finding where it crosses 1/2. It is important to note that although the functional form of the success curves is not known, a function consistent with the data points will provide a reliable estimate of the success threshold.

The functional form for the fit was chosen based on several considerations. The success probability must be identically 1 for a noise magnitude of $\epsilon = 0$ and must level off to a constant for high noise magnitude. Both of these features are met by a decaying exponential, however the success probability appears to remain close to 1 for small ϵ rather than decaying exponentially. By multiplying a decaying exponential by a function that is approximately a growing exponential for small ϵ and a constant for large ϵ, a plateau for small ϵ can be achieved. The low and high ϵ limits of the hyperbolic tangent meet this requirement, suggesting:

$$g(\epsilon) = e^{-\epsilon}\left(1 + \tanh(\epsilon)\right) \tag{6.7}$$

Qubits	3	4	5	6	7	8	9
χ^2_ν	0.89	1.34	1.21	1.25	1.66	0.74	0.67

Table 6.2: Reduced chi squared values for fits of the success fraction data for 1-local noise.

Qubits	3	4	5	6	7	8	9
χ^2_ν	0.61	0.62	0.69	0.76	0.98	1.15	1.54

Table 6.3: Reduced chi squared values for fits of the success fraction data for 2-local noise.

as a possible fitting function. To allow for more flexibility in the fit I have used a polynomial in ϵ for the argument of the hyperbolic tangent. Inserting fit parameters and scaling the function to level off to a constant gives the fitting function:

$$f(\epsilon) = C + (1-C)e^{-a\epsilon}\left(1 + b\frac{\sinh(c\epsilon^3 + d\epsilon^2 + f\epsilon)}{\cosh(g\epsilon^3 + h\epsilon^2 + i\epsilon)}\right), \tag{6.8}$$

where C is the high-noise success probability and a, b, c, d, f, g, h, i are fit parameters. An example of the success fraction data fit to (6.8) is shown in Figure 6-5. The reduced chi squared values (χ^2_ν)for each fit are shown in Tables 6.2 and 6.2. In general, the χ^2_ν values are all near 1, confirming that the fits are consistent with the data.

The success threshold was determined from the fits using a simple numerical technique. The fit function was evaluated for 10,000 evenly spaced noise magnitudes in the same range as the simulations. The smallest noise magnitude for which the value of the fit function was less than 1/2 was chosen to be the success fraction. The uncertainty σ_f in value of the success fraction at that noise magnitude was determined by applying the standard error propagation formula [21] to the the uncertainties of the fit parameters as calculated by a standard nonlinear fitting routine. The uncertainty of the success threshold σ_t is thus given by dividing by the magnitude of the slope of the fit function (6.8):

$$\sigma_t = \sigma_f / \left|\frac{df}{d\epsilon}\right|. \tag{6.9}$$

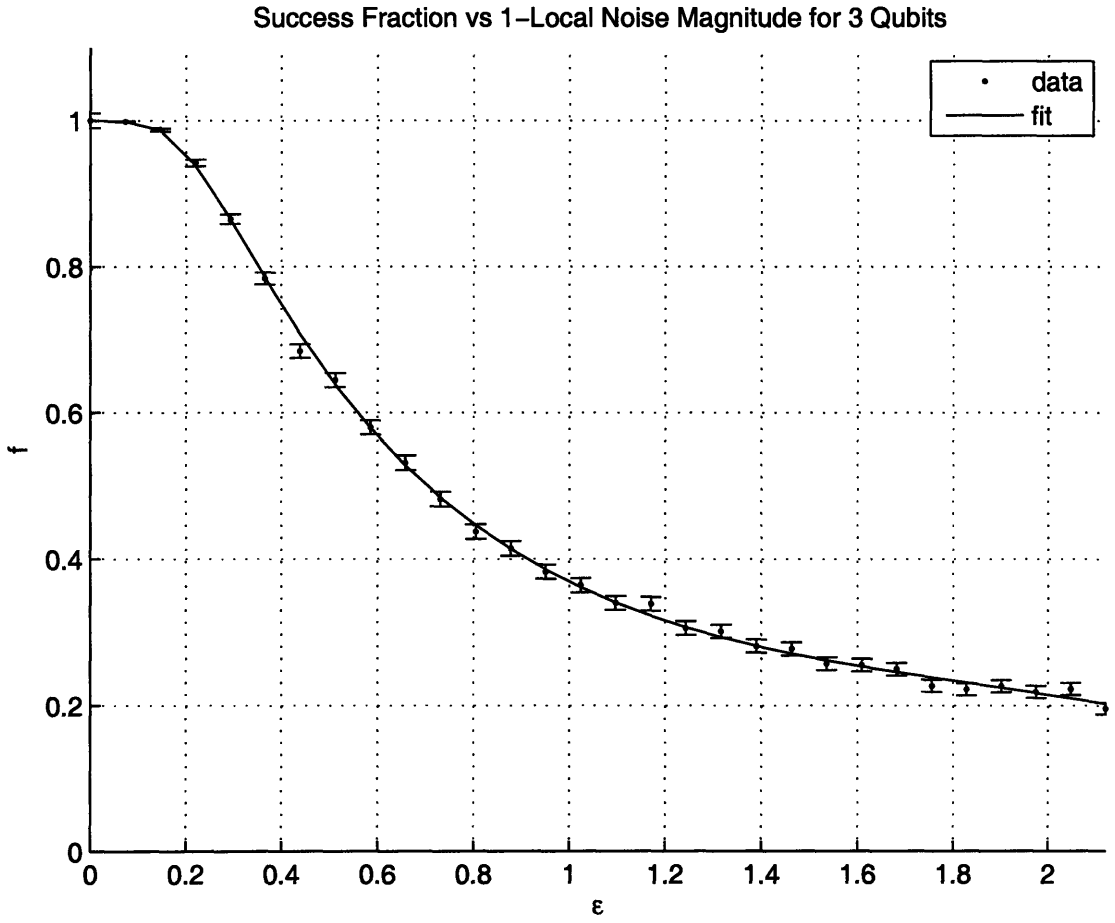

Figure 6-5: Example fit of success fraction curve.

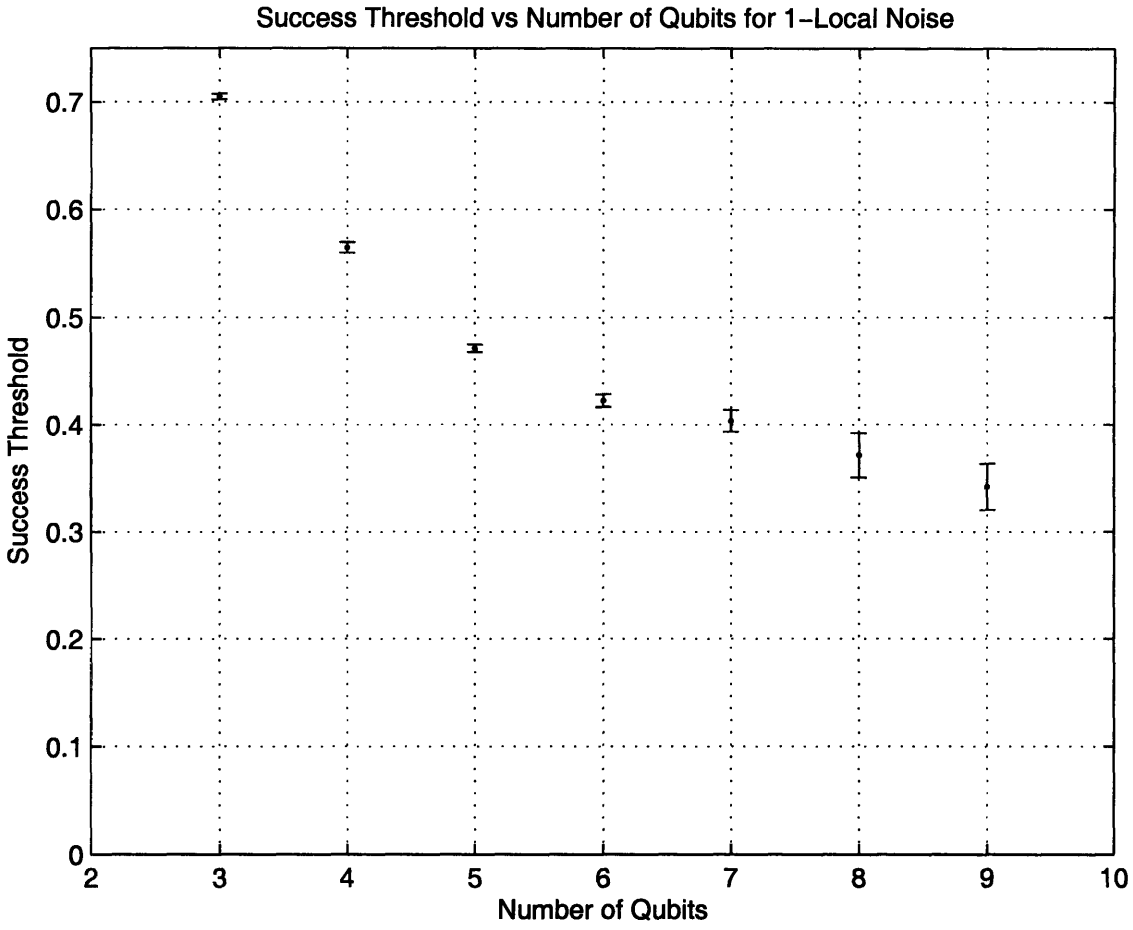

Figure 6-6: Success threshold as a function of the number of qubits for 1-local noise.

The resulting success thresholds are shown in figures 6-6 and 6-7. The success thresholds are also shown plotted on a semi-log scale in figures 6-8 and 6-9, as well as a log-log scale in figures 6-10 and 6-11. If the decline in success threshold is exponential, the data should be linear on the semi-log scale. If the decline is polynomial, the data should be linear on the log-log scale.

Comparing figures 6-8 and 6-9 with figures 6-10 and 6-11 shows that the data much closer to linear on a log-log scale, suggesting that the success threshold for the neighbor-coupling algorithm decreases polynomially in the number of qubits.

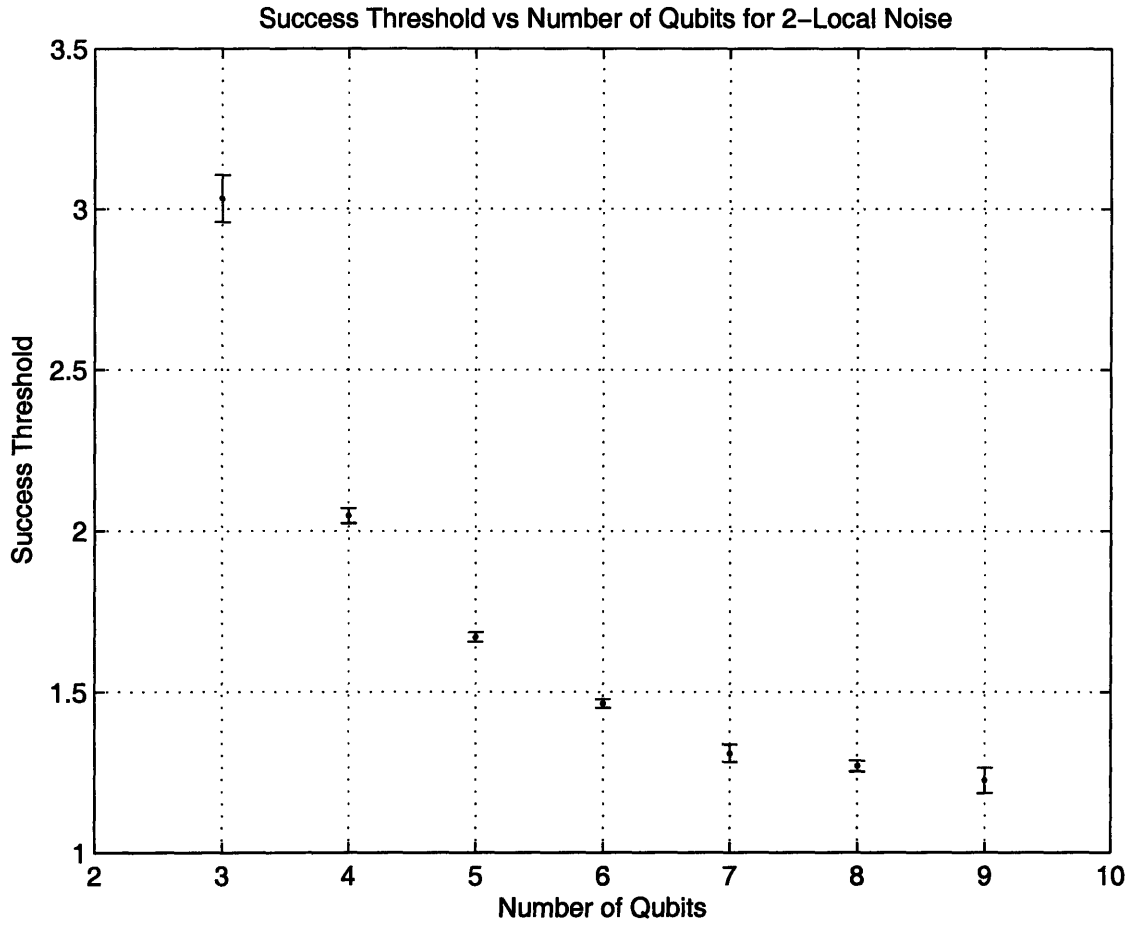

Figure 6-7: Success threshold as a function of the number of qubits for 2-local noise.

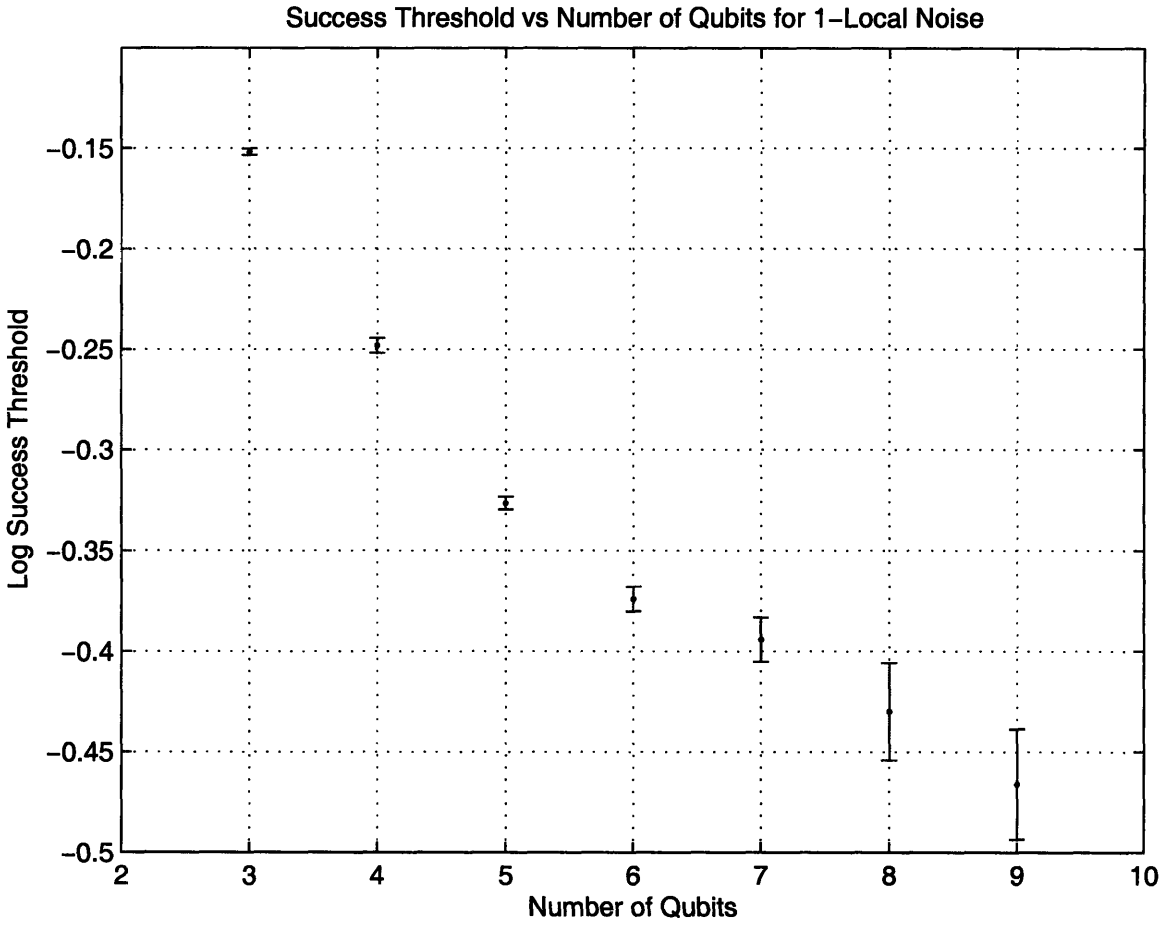

Figure 6-8: Success threshold as a function of the number of qubits for 1-local noise ploted on a semi-log scale.

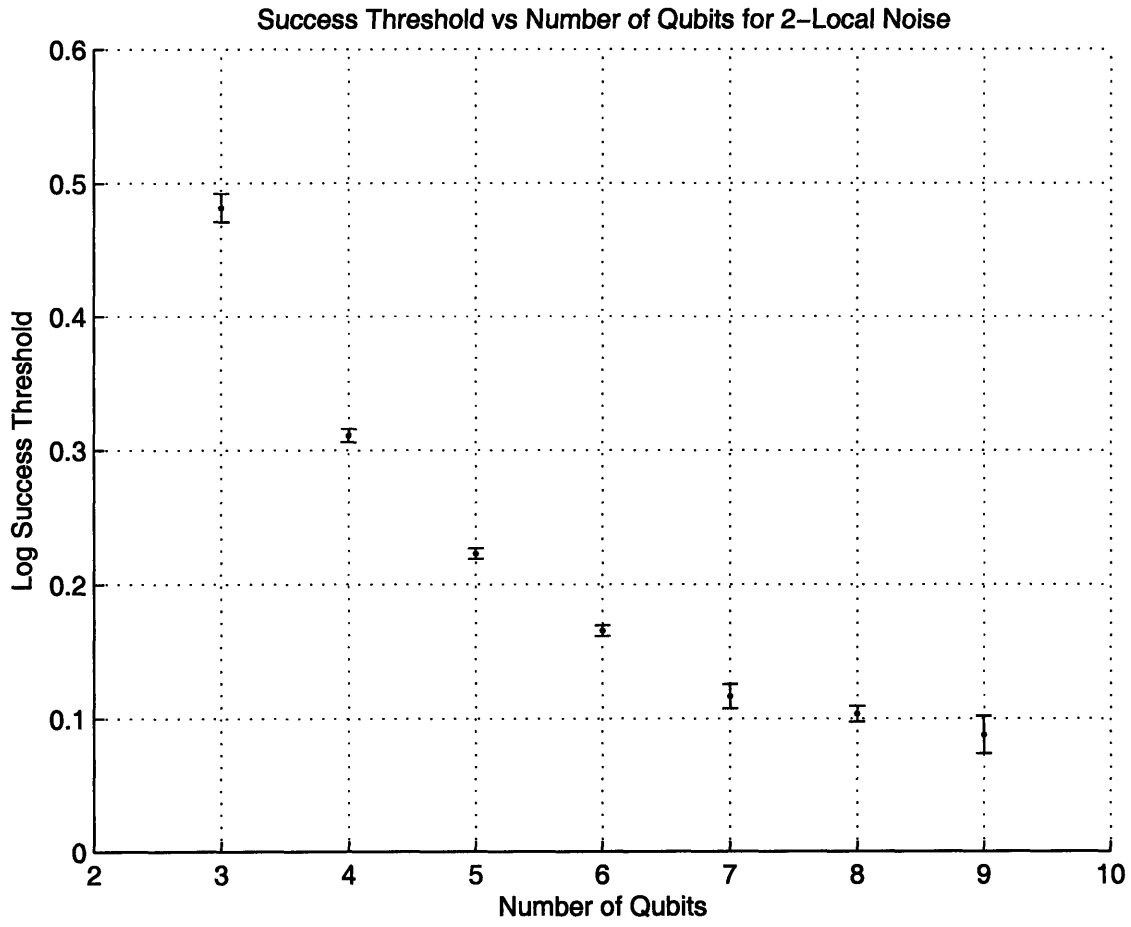

Figure 6-9: Success threshold as a function of the number of qubits for 2-local noise, plotted on a semi-log scale.

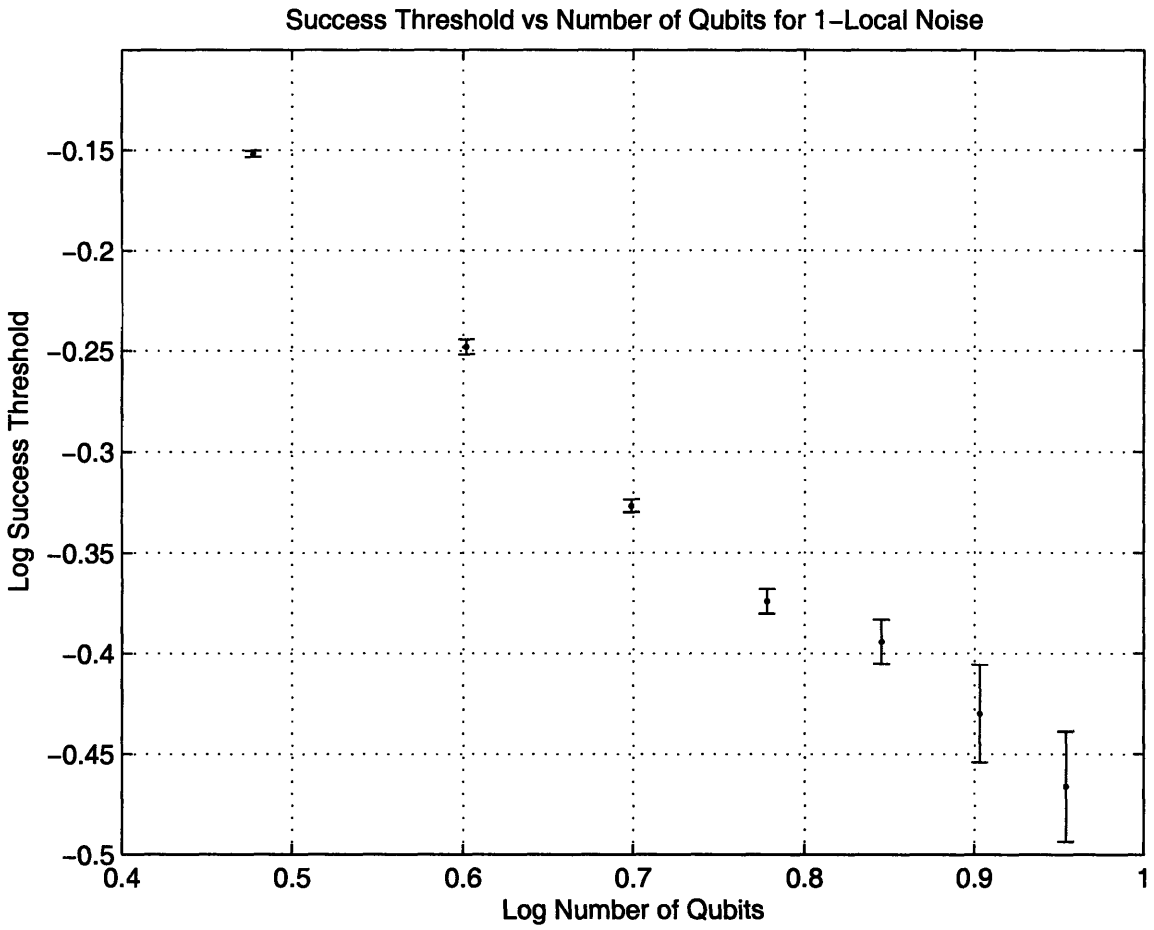

Figure 6-10: Success threshold as a function of the number of qubits for 1-local noise ploted on a log-log scale.

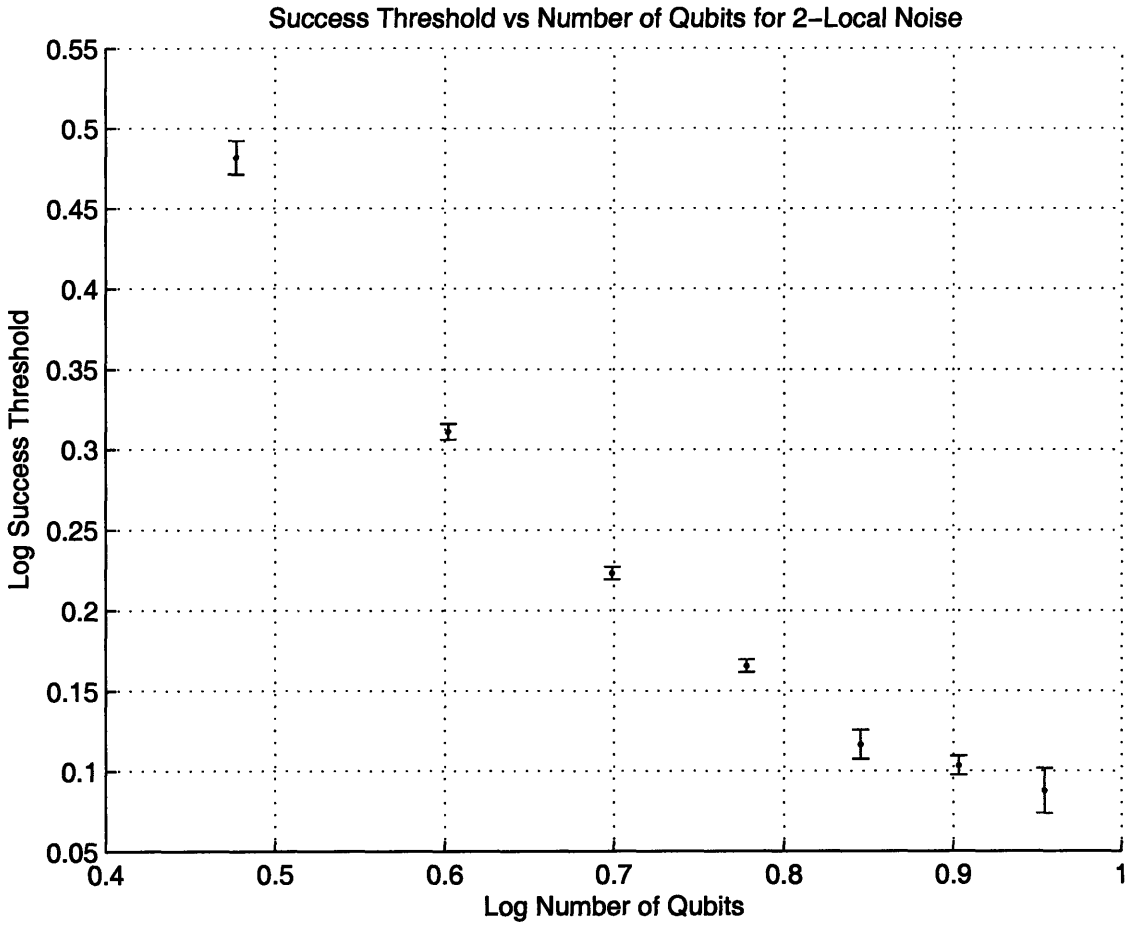

Figure 6-11: Success threshold as a function of the number of qubits for 2-local noise, plotted on a log-log scale.

Chapter 7

Conclusion

The numerical simulations performed for this thesis give insight into the behavior of the neighbor-coupling algorithm for the agree problem on a line of qubits in the presence of control error. One of the most interesting results is the apparent existence of a minimum success threshold when only 2-local noise is present. The existence of such a threshold implies that a neighbor-coupling algorithm of arbitrary size will succeed as long as the noise in the system is 2-local, shares the structure of the problem Hamiltonian and has a magnitude less than that of the terms in the problem Hamiltonian.

The simulations have also shown that the successs threshold for the neighbor-coupling algorithm decreases with the number of qubits for all types of noise considered. The simulation data for both 1 and 2-local noise suggest that the success threshold scales polynomially with the number of qubits rather than exponentially.

Further studies to determine how the success threshold scales with the number of qubits in the neighbor-coupling algorithm would be insightful. It would also be insightful to consider how the success threshold changes as the magnitude 1-local term on qubit 0 and the 2-local terms between the other qubits are varied independently. Finally, the existence of a noise threshold for the case of 2-local noise suggests an approach for constructing fault tolerant adiabatic algorithms when 1-local noise can be ignored. As such, it would be useful to determine when 1-local noise can be ignored, as well as what the computational power of such algorithms would be.

Bibliography

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

[2] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000.

[3] Lov K. Grover. A fast quantum mechanical algorithm for database search, 1996.

[4] Matthias Steffen, Wim van Dam, Tad Hogg, Greg Breyta, and Isaac Chuang. Experimental implementation of an adiabatic quantum optimization algorithm. *Physical Review Letters*, 067903, 2003.

[5] Avik Mitra, Arindam Ghosh, Ranabir Das, Apoorva Patel, and Anil Kumar. Experimental implementation of local adiabatic evolution algorithms by an nmr quantum information processor, 2005.

[6] William M. Kaminsky, Seth Lloyd, and Terry P. Orlando. Scalable superconducting architecture for adiabatic quantum computation, 2004.

[7] Debabrata Goswami. Adiabatic quantum computing with phase modulated laser pulses, 2005.

[8] Andrew M. Childs, Edward Farhi, and John Preskill. Robustness of adiabatic quantum computation. *Physical Review A*, 65:012322, 2002.

[9] Yu Shi and Yong-Shi Wu. Perturbative formulation and non-adiabatic corrections in adiabatic quantum computing schemes. *Physical Review A*, 69:024301, 2004.

[10] David R. Mitchell, Christoph Adami, Waynn Lue, and Colin P. Williams. A random matrix model of adiabatic quantum computing. *Physical Review A*, 71:052324, 2005.

[11] Ryan Requist, John Schliemann, Alexander G. Abanov, and Daniel Loss. Double occupancy errors in quantum computing operations: Corrections to adiabaticity. *Physical Review B*, 71:115315, 2005.

[12] Jeremie Roland and Nicolas J. Cerf. Noise resistance of adiabatic quantum computation using random matrix theory. *Physical Review A*, 71:032330, 2005.

[13] M. S. Sarandy and D. A. Lidar. Adiabatic quantum computation in open systems. *Physical Review Letters*, 95:250503, 2005.

[14] Gernot Schaller, Sarah Mostame, and Ralf Schutzhold. General error estimate for adiabatic quantum computing, 2005.

[15] Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler. Bounds for the adiabatic approximation with applications to quantum computation, 2006.

[16] Stephen P. Jordan, Edward Farhi, and Peter W. Shor. Error correcting codes for adiabatic quantum computation, 2005.

[17] Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation, 2004.

[18] Albert Messiah. *Quantum Mechanics*, volume II. Wiley, 1976.

[19] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *SIAM JOURNAL OF COMPUTING*, 35:1070, 2006.

[20] Tosio Kato. On the convergence of the perturbation method. i. *Progress of Theoretical Physics*, IV(4), 1949.

[21] Philip R. Bevington and D. Keith Robinson. *Data Reduction and Error Analysis for the Physical Sciences.* McGraw-Hill, 2003.