Quantum Information Processing in Continuous Time

by

Andrew MacGregor Childs

B.S. in Physics, California Institute of Technology, 2000

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

Doctor of Philosophy in Physics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2004

Author . . . .
Department of Physics
16 April 2004

Certified by . . . .
Edward H. Farhi
Professor of Physics
Thesis Supervisor

Accepted by . . . .
Thomas J. Greytak
Professor of Physics
Associate Department Head for Education

Quantum Information Processing in Continuous Time
by
Andrew MacGregor Childs

Submitted to the Department of Physics
on 16 April 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Physics

Abstract

Quantum mechanical computers can solve certain problems asymptotically faster than any classical computing device. Several fast quantum algorithms are known, but the nature of quantum speedup is not well understood, and inventing new quantum algorithms seems to be difficult. In this thesis, we explore two approaches to designing quantum algorithms based on continuous-time Hamiltonian dynamics.

In quantum computation by adiabatic evolution, the computer is prepared in the known ground state of a simple Hamiltonian, which is slowly modified so that its ground state encodes the solution to a problem. We argue that this approach should be inherently robust against low-temperature thermal noise and certain control errors, and we support this claim using simulations. We then show that any adiabatic algorithm can be implemented in a different way, using only a sequence of measurements of the Hamiltonian. We illustrate how this approach can achieve quadratic speedup for the unstructured search problem.

We also demonstrate two examples of quantum speedup by quantum walk, a quantum mechanical analog of random walk. First, we consider the problem of searching a region of space for a marked item. Whereas a classical algorithm for this problem requires time proportional to the number of items regardless of the geometry, we show that a simple quantum walk algorithm can find the marked item quadratically faster for a lattice of dimension greater than four, and almost quadratically faster for a four-dimensional lattice. We also show that by endowing the walk with spin degrees of freedom, the critical dimension can be lowered to two. Second, we construct an oracular problem that a quantum walk can solve exponentially faster than any classical algorithm. This constitutes the only known example of exponential quantum speedup not based on the quantum Fourier transform.

Finally, we consider bipartite Hamiltonians as a model of quantum channels and study their ability to process information given perfect local control. We show that any interaction can simulate any other at a nonzero rate, and that tensor product Hamiltonians can simulate each other reversibly. We also calculate the optimal asymptotic rate at which certain Hamiltonians can generate entanglement.

Thesis Supervisor: Edward H. Farhi
Title: Professor of Physics

Acknowledgments

First and foremost, I would like to thank my advisor, Eddie Farhi, for invaluable advice and support. Eddie has been a great teacher and friend. I have also been fortunate to work with and learn from Jeffrey Goldstone and Sam Gutmann. I have enjoyed interacting with many other quantum information colleagues at the Center for Theoretical Physics, especially Wim van Dam, Enrico Deotto, Jason Eisenberg, and Andrew Landahl. And I would also like to thank the members of the greater quantum information community at MIT, especially Ken Brown, Ike Chuang, Aram Harrow, Bill Kaminsky, Seth Lloyd, and Peter Shor. In particular, I would like to thank Aram Harrow for commenting on Chapter 6 of this thesis.

During my thesis research, I enjoyed extended visits to the University of Queensland and the IBM T. J. Watson Research Center. I would like to thank many individuals at those institutions: at UQ, Michael Nielsen, Mick Bremner, Chris Dawson, Jennifer Dodd, Henry Haselgrove, Gerard Milburn, and Tobias Osborne; and at IBM, Nabil Amer, Charlie Bennett, Igor Devetak, Debbie Leung, Anthony Ndirango, and John Smolin.

The work described in this thesis is the product of collaborations with many people (as detailed in Section 0.4). I am grateful to Mick Bremner, Richard Cleve, Chris Dawson, Enrico Deotto, Jennifer Dodd, Eddie Farhi, Jeffrey Goldstone, Sam Gutmann, John Preskill, Andrew Landahl, Debbie Leung, Michael Nielsen, Daniel Spielman, Guifré Vidal, and Frank Verstraete for their contributions to this work.

I would also like to thank the many members of the quantum information community, in addition to those mentioned above, with whom I have had numerous illuminating discussions. I would especially like to thank Scott Aaronson, Dorit Aharonov, Andris Ambainis, Dave Bacon, Peter Høyer, Julia Kempe, Mike Mosca, Roman Orus, Jérémie Roland, Mario Szegedy, John Watrous, and Ronald de Wolf.

I would like to thank my fellow graduate students at the Center for Theoretical Physics, especially Michael Forbes, for four great years. I would also like to thank the staff of the CTP, Joyce Berggren, Scott Morley, Marty Stock, and Charles Suggs, for their generous help.

Finally, I would like to thank Sidney and Mom, Dad, and Ryan for their love and support.

My graduate studies were supported by a Fannie and John Hertz Foundation Fellowship. This work was also supported in part by the Cambridge–MIT Institute, by the Department of Energy under cooperative research agreement DE-FC02-94ER40818, and by the National Security Agency and Advanced Research and Development Activity under Army Research Office contract DAAD19-01-1-0656.

Contents

Chapter 0

Introduction

0.1 Quantum information processing

The fact that our world is quantum mechanical enables technologies that could not exist in an entirely classical world. In one sense, we are already surrounded by such wonders, from the transistors that make up our computers to the superconducting magnets that drive nuclear magnetic resonance imaging. But even these devices, while quantum mechanical at base, still operate using an old-fashioned, classical notion of information. In fact, quantum mechanics has the potential to revolutionize technology in a more fundamental way through a uniquely quantum mechanical concept of what information is and how it can be manipulated.

The notion of quantum information processing is an inevitable consequence of physical law. As Landauer put it, information is physical [122]: since information is stored and processed by physical systems, its basic properties must be consequences of physical principles. But physics, as we know from decades of experiment, is fundamentally quantum mechanical in nature. Therefore the notion of information, its representation and its manipulation, should be founded on quantum mechanics.

The ultimate apparatus for processing quantum information is the *universal quantum computer*, a quantum mechanical analog of the ubiquitous classical computer. The essential idea of a quantum computer was suggested in the early 1980s by Manin [137] and Feynman [84], who observed that an inherently quantum mechanical computer would be well-suited to the simulation of quantum systems, a problem that seems to be hard for classical computers. The notion of a quantum computer as a universal computational device was introduced in 1985 by Deutsch [63], who also gave the first example of a problem that could be solved faster by a quantum computer than by a classical computer. Deutsch's work was followed by a steady sequence of advances (described in Section 0.3), culminating in 1994 with Shor's discovery of efficient quantum algorithms for factoring integers and calculating discrete logarithms [165]. Since the security of most present-day cryptosystems is based on the presumed difficulty of factoring, Shor's algorithm demonstrates that the construction of a quantum computer would have significant practical applications.

The universal quantum computer is an abstract device, and could potentially be realized in wide variety of physical systems, just as classical computers can in principle be built using, for example, transistors, vacuum tubes, biological molecules, or mechanical components. Just a few of the proposed implementations of quantum computers include trapped ions [51], quantum dots [135], nuclear spins [55, 92], and Josephson junctions [147]. All of the current proposals for building quantum computers have their own advantages and disadvantages, and it is presently not clear which proposals will ultimately yield workable large-scale quantum computers, or if a fundamentally new idea is needed.

One of the problems that must be dealt with in any realistic implementation of a quantum computer is the inevitable occurrence of errors. No system can be perfectly isolated from its environment, and for quantum mechanical systems, such interactions will lead to a loss of quantum coherence. Furthermore, unless a system can be controlled exactly, simply performing computational operations will lead to errors. Fortunately, the concepts of quantum error correction [166, 170] and fault-tolerant quantum computation [167] have been introduced to cope with these problems. For the most part we will not address this issue, assuming that fault-tolerant techniques can be used to build a quantum device with essentially no errors. (One exception occurs in Section 2.4, where we consider a quantum computational paradigm with a certain intrinsic robustness against faults.)

Of course, the applications of quantum information processing are not limited to computation. There are many other kinds of tasks that can be performed better quantum mechanically, or that are impossible without the use of quantum systems. For example, relatively simple quantum devices can be used to securely distribute a secret key over long distances [21], a task that is impossible classically. As another example, entanglement between two separated quantum systems—a kind of uniquely quantum correlation with no classical counterpart—can be used to perform a variety of information processing tasks, such as increasing the rate at which classical information can be transmitted through a noiseless quantum channel [30]. Examples such as these also demonstrate that the development of quantum information processing technologies could have a significant impact.

This thesis is an exploration of the extent of the quantum information processing revolution: what can we do with quantum mechanical systems that we could not do without them? The presently known quantum algorithms show that quantum computers would be useful special-purpose devices, but the full power of quantum computation is not well understood. We approach this question by considering two recent approaches to quantum algorithms, quantum computation by adiabatic evolution (Chapter 2) and quantum walks on graphs (Chapters 3–5). We also explore some non-algorithmic information processing applications in bipartite quantum systems (Chapter 6).

0.2 Continuous time

Most of the algorithms and other information processing protocols presented in this thesis are described in terms of continuous time evolution. In quantum mechanics, states evolve according to the Schrödinger equation, the linear, first-order differential equation

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = H(t)|\psi(t)\rangle \tag{0.1}$$

where $\hbar = 1.054 \times 10^{-34}\ \mathrm{J\cdot s}$ is Planck's constant divided by 2π, $|\psi(t)\rangle$ is the state vector at time t, and $H(t)$ is a time-dependent Hermitian operator with units of energy, the *Hamiltonian.* Throughout, we use units in which $\hbar = 1$. If the Hamiltonian is in fact time-independent, then the time evolution is simply given by $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$. Otherwise, one must in general integrate (0.1) to determine the state at an arbitrary time.

Why do we consider continuous time evolution? Primarily, it is a matter of convenience. In keeping with the idea that information is physical, we will try to discover quantum advantages by considering physical effects that lend themselves to quantum speedup. Physical systems are naturally described by their Hamiltonians, and evolve continuously in time.

For the most part, the distinction between continuous and discrete is essentially semantic. Continuous processes can be approximated by discrete ones, so we certainly do not gain any computational power by considering continuous time evolution. However, certain processes may be more naturally defined in terms of Hamiltonian dynamics. The most dramatic example of this will be seen in Chapter 3, where we discuss a quantum analog of the classical notion of random walk. Whereas a continuous-time classical random walk can be viewed as a limiting case of a discrete-time random walk, the same is not true in the quantum case: continuous- and discrete-time quantum walks must be defined in fundamentally different ways, as we will see.

0.3 Quantum algorithms

The bulk of this thesis is concerned with the problem of discovering new *quantum algorithms*. Perhaps the most important open problem in the theory of quantum information processing is to understand the nature of quantum mechanical speedup for the solution of computational problems. What problems can be solved more rapidly using quantum computers than is possible with classical computers, and what ones cannot? To take full advantage of the power of quantum computers, we should try to find new problems that are amenable to quantum speedup. More importantly, we should try to broaden the range of available algorithmic techniques for quantum computers, which is presently quite limited.

The first examples of problems that can be solved faster with a quantum computer than with a classical computer were *oracular*, or black-box, problems. In standard computational problems, the input is simply a string of data such as an integer or the description of a graph. In contrast, in the black box model, the computer is given access to a black box, or oracle, that can be queried to acquire information about the problem. The goal is to find the solution to the problem using as few queries to the oracle as possible. This model has the advantage that proving lower bounds is tractable, which allows one to demonstrate provable speedup over classical algorithms, or to show that a given quantum algorithm is the best possible.

Deutsch's pioneering example of quantum speedup was an oracular problem that can be solved on a quantum computer using one query, but that requires two queries on a classical computer [63]. Deutsch and Jozsa generalized this problem to one that can be solved exactly on a quantum computer in polynomial time, but for which an exact solution on a classical computer requires exponential time [65]. However, the Deutsch-Jozsa problem can be solved with high probability in polynomial time using a probabilistic classical algorithm. Bernstein and Vazirani gave the first example of a superpolynomial separation between probabilistic classical and quantum computation [31], and Simon gave another example in which the separation is exponential [168]. This sequence of examples of rather artificial oracular problems led to Shor's aforementioned discovery of efficient quantum algorithms for the factoring and discrete log problems [165], two non-oracular computational problems with practical applications, for which no polynomial-time classical algorithm is known.

Shor's algorithm, like its predecessors, is based on the ability to efficiently implement a quantum Fourier transform [54]. More recently, numerous generalizations and variations of Shor's algorithm have been discovered for solving both oracular and non-oracular problems with superpolynomial speedup [118, 149, 62, 100, 58, 97, 109, 181, 99]. All of these algorithms are fundamentally based on quantum Fourier transforms.

A second tool used in quantum algorithms comes from Grover's algorithm for unstructured search [98]. In the unstructured search problem, one is given black box access to a list of N items, and must identify a particular marked item. Classically, this problem clearly requires $\Omega(N)$ queries, but Grover showed that it could be solved on a quantum computer using only $O(\sqrt{N})$ queries (which had previously been shown to be the best possible result by Bennett, Bernstein, Brassard, and Vazirani [19]). This speedup is more modest than the speedup of Shor's algorithm—it is only quadratic rather than superpolynomial—but the basic nature of unstructured search means that it can be applied to a wide variety of other problems. Grover's algorithm was subsequently generalized to the concept of amplitude amplification [33], and many extensions and applications have been found [37, 38, 107, 71].

The Shor and Grover algorithms and their relatives will surely be useful if large-scale quantum computers can be built. But they also raise the question of how broadly useful quantum computers could be. It appears to be difficult to design quantum algorithms, so it would be useful to have more algorithmic techniques to draw from, beyond the quantum Fourier transform and amplitude amplification. Here we investigate two such ideas, quantum computation by adiabatic evolution and quantum walks on graphs. Both are naturally described in terms of Hamiltonian dynamics. Adiabatic quantum computation, discussed in Chapter 2, works by keeping the quantum computer near the ground state of a slowly time-varying Hamiltonian. It can be used to approach a wide variety of combinatorial search problems, although its performance is typically difficult to analyze. Quantum walks on graphs, introduced in Chapter 3, are quantum analogs of Markov chains, which are extensively used in classical algorithms. Using quantum walks, we give two examples of algorithmic speedup: a quadratic speedup for a spatial version of the unstructured search problem in Chapter 4, and an exponential speedup for a carefully constructed oracular problem in Chapter 5.

0.4 Summary of results

This section summarizes the original results described in this thesis. Most of these results have previously appeared in published articles, as indicated.

In Chapter 1, we discuss the simulation of Hamiltonian dynamics on a universal quantum computer. After presenting some standard techniques for Hamiltonian simulation, we prove that any sufficiently sparse Hamiltonian can be efficiently simulated. This idea follows from results of [40] (joint work with Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel Spielman) together with classical results on graph coloring, and was also presented in [6].

In Chapter 2, we consider the idea of quantum computation by adiabatic evolution, a general approach to combinatorial search with a quantum computer. In Section 2.4, we argue that adiabatic quantum computation should be inherently robust against certain kinds of errors, and we investigate this claim through numerical simulations [45] (joint work with Edward Farhi and John Preskill). In Section 2.5, we show that a variant of the adiabatic algorithm can be implemented using only a sequence of measurements, which can be realized dynamically by Hamiltonian evolution. We also show that this measurement algorithm can solve the unstructured search problem in a similar way to Grover's algorithm [41] (joint work with Enrico Deotto, Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Andrew Landahl).

In Chapter 3, we describe the continuous-time quantum walk, a quantum analog of the classical continuous-time random walk. We present some basic examples and review some related work.

In Chapter 4, we apply quantum walk to the problem of locally searching a d-dimensional space for a single marked item. We show in Section 4.4 that the simplest approach finds the marked item in time $O(\sqrt{N})$ for $d > d^*$ and in time $O(\sqrt{N} \log N)$ for $d = d^*$, where the critical dimension is $d^* = 4$ [46] (joint work with Jeffrey Goldstone). In Section 4.5, we show that by adding a spin degree of freedom, the critical dimension can be lowered to $d^* = 2$ [47] (joint work with Jeffrey Goldstone). In Section 4.6, we describe how such algorithms can be implemented with the same run times in a model of local unitary transformations.

In Chapter 5, we explain how quantum walk can be used to achieve exponential speedup over classical processes. We begin in Section 5.2 with a simple example of a graph on which a quantum walk propagates between two designated vertices exponentially faster than the corresponding classical random walk [44] (joint work with Edward Farhi and Sam Gutmann). Then, in Section 5.3, we modify this example to construct an oracular problem that can be solved exponentially faster by quantum walk than by any classical algorithm [40] (joint work with Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel Spielman). This algorithm constitutes the only known example of exponential quantum speedup that is not based on the quantum Fourier transform.

Finally, in Chapter 6, we consider a bipartite Hamiltonian as a model of a quantum channel connecting two parties. In Section 6.3, we show that all nontrivial interactions can simulate each other at a nonzero rate [152] (joint work with Michael Nielsen, Michael Bremner, Jennifer Dodd, and Christopher Dawson). In Section 6.4, we show that furthermore, all tensor product Hamiltonians can simulate each other *reversibly*, i.e., with no loss of efficiency [50] (joint work with Debbie Leung and Guifré Vidal). Finally, in Section 6.5, we study the rate at which bipartite Hamiltonians can generate entanglement between the two parties. We compute the maximum possible rate for the Ising interaction [49] (joint work with Debbie Leung, Guifré Vidal, and Frank Verstraete), and consequently, for any product Hamiltonian.

Chapter 1

Simulating Hamiltonian dynamics

1.1 Introduction

In subsequent chapters, we present quantum algorithms defined in terms of continuous-time quantum dynamics according to the Schrödinger equation (0.1) with some specified Hamiltonian $H(t)$. But before doing so, we must pause to consider what Hamiltonians correspond to allowed computations. Surely not just any Hamiltonian can be allowed, since there are perfectly well-defined Hamiltonians that quickly produce solutions to uncomputable problems, not to mention intractable ones.

One reasonable definition of the set of allowed Hamiltonians is to allow any Hamiltonian that can be "easily" constructed as the Hamiltonian of an actual physical system. Such a definition is not entirely desirable since the details of what constitutes an easily constructible Hamiltonian will vary depending on the particular apparatus used. Nevertheless, this idea might naturally lead to a definition along the following lines. Suppose we consider a quantum system consisting of n two-level atoms. Then we might allow Hamiltonians of the form

$$H = \sum_j H_j \tag{1.1}$$

where each term H_j acts on a small number of atoms (e.g., no more than some constant number, independent of n). In addition, we might impose certain locality constraints, such as only allowing a coupling between atoms that are nearest neighbors in a two-dimensional or three-dimensional array.

Although such a definition may be reasonable, we will find it more convenient to use an equivalent definition expressed in terms of the quantum circuit model introduced by Deutsch [64]. In the quantum circuit model, the computer again consists of n two-level quantum systems (or *qubits*). The qubits are acted on by a sequence of unitary transformations acting on at most two qubits at a time [67]. Finally, the result of the computation is determined by measuring the qubits in the standard basis $\{|0\rangle, |1\rangle\}^{\otimes n}$, known as the *computational basis*. The quantum circuit model is universal in the sense that any unitary operator acting on the 2^n-dimensional Hilbert space of the n qubits can be realized to arbitrary precision by some (not necessarily short) sequence of two-qubit unitary transformations. In fact, the model can be universal even if the unitary transformations are chosen from a finite set. Of course, just as for Hamiltonians, generic unitary transformations are very difficult to implement. A sequence of poly(n) two-qubit unitary gates can only approximate a tiny fraction of the possible unitary transformations on n qubits.

The quantum circuit model has become the standard model of quantum computation because it is straightforward and easy to work with. It is certainly not unique, but any reasonable model of quantum computation is equivalent to the quantum circuit model. Many equivalent models of quantum computation are known; a few examples include the quantum Turing machine [63, 31, 188], quantum cellular automata [138, 180, 57], topological quantum field theories [88, 89], the one-way quantum computer [157], and adiabatic quantum computation [6, 5].[1]

Since the quantum circuit model is the standard model of quantum computation, we will find it useful to define the set of reasonable Hamiltonians as those that can be efficiently simulated using quantum circuits. More precisely, we define the set of efficiently simulable Hamiltonians as follows:

Definition 1.1. *A Hamiltonian H can be* efficiently simulated *if for any $t > 0$, $\epsilon > 0$ there is a quantum circuit U consisting of* $\mathrm{poly}(n, t, 1/\epsilon)$ *gates such that $\left\|U - e^{-iHt}\right\| < \epsilon$.*

In Sections 1.2 and 1.3, we present a collection of basic tools for efficiently simulating Hamiltonians. Using these techniques, the Hamiltonian for a continuous-time quantum algorithm can be viewed as a convenient shorthand for a sequence of unitary transformations. But we should stress that the Hamiltonian perspective may be useful for designing algorithms, as we will show through several examples.

The approach of simulating Hamiltonian dynamics using quantum circuits has the advantage that it can easily accommodate oracular problems. This setting will be useful in Section 5.3, where we use an oracular problem to prove an exponential separation between classical and quantum computers using a quantum walk.

The simulation techniques presented below only explicitly address time-independent Hamiltonians. However, time-dependent Hamiltonian dynamics can easily be simulated by discretizing the evolution into sufficiently small time steps (as will typically be necessary even when the Hamiltonian is time-independent), over which the Hamiltonian is approximately constant. Simulations of time-dependent Hamiltonians may be useful in an algorithmic context, for example in adiabatic quantum computation, as discussed in Chapter 2.

In this thesis, we are primarily interested algorithms for computational problems. However, the techniques described here may also be useful for simulating physical systems. We discuss this application briefly in Section 1.4.

Finally, in Section 1.5, we present a result of Feynman showing that even simple, time-independent Hamiltonians of a form such as (1.1) have at least as much computational power as the quantum circuit model. Therefore, we do not lose anything (at least in principle) by considering continuous-time quantum algorithms. Together with the simulations described in Section 1.2, Feynman's result shows that time-independent Hamiltonian dynamics can be viewed as a model of quantum computation in its own right, since it is computationally equivalent to the quantum circuit model.

1.2 Simulation rules

In this section, we present some standard techniques for efficiently simulating Hamiltonians using quantum circuits.

[1] The abundance of models of quantum computation parallels the situation in classical computation, which can equivalently be defined in terms of, for example, classical circuits, Turing machines, cellular automata, or the lambda calculus.

First, note that local Hamiltonians are easy to simulate.

Rule 1.1 (Local Hamiltonians). *If H acts on $O(1)$ qubits, then it can be efficiently simulated.*

This follows simply because any unitary evolution on a constant number of qubits can be approximated using a constant number of one- and two-qubit gates.

We can also rescale a Hamiltonian by any real constant, as long as that constant is not too large.

Rule 1.2 (Rescaling). *If H can be efficiently simulated, then cH can be efficiently simulated for any $c = \mathrm{poly}(n)$.*

Note that there is no restriction to $c > 0$, since any efficient simulation is expressed in terms of quantum gates, and can simply be run backward.

In addition, we can rotate the basis in which a Hamiltonian is applied using any unitary transformation with an efficient decomposition into basic gates.

Rule 1.3 (Unitary conjugation). *If H can be efficiently simulated and the unitary transformation U can be efficiently implemented, then $UHU^\dagger$ can be efficiently simulated.*

This rule follows from the simple identity

$$e^{-iUHU^\dagger t} = U e^{-iHt} U^\dagger \,. \tag{1.2}$$

Given two or more simulable Hamiltonians, we can produce further simulable Hamiltonians from them. For example, we have

Rule 1.4 (Addition). *If H_1 and H_2 can be efficiently simulated, then $H_1 + H_2$ can be efficiently simulated.*

If the two Hamiltonians commute, then this rule is trivial, since in that case $e^{-iH_1 t} e^{-iH_2 t} = e^{-i(H_1+H_2)t}$. However, in the general case where the two Hamiltonians do not commute, we can still simulate their sum as a consequence of the Lie product formula

$$e^{-i(H_1+H_2)t} = \lim_{m\to\infty} \left(e^{-iH_1 t/m} e^{-iH_2 t/m} \right)^m \,. \tag{1.3}$$

A simulation using a finite number of steps can be achieved by truncating this expression to a finite number of terms, which introduces some amount of error that must be kept small. Writing

$$e^{-iHt} = 1 - iHt + O(\|H\|^2 t^2) \,, \tag{1.4}$$

we find

$$\begin{aligned}
\left(e^{-iH_1 t/m} e^{-iH_2 t/m} \right)^m &= \left(1 - i(H_1+H_2)t/m + O(h^2 t^2/m^2)\right)^m && (1.5)\\
&= \left(e^{-i(H_1+H_2)t/m} + O(h^2 t^2/m^2) \right)^m && (1.6)\\
&= e^{-i(H_1+H_2)t} + O((ht)^2/m) \,, && (1.7)
\end{aligned}$$

where $h = \max\{\|H_1\|, \|H_2\|\}$. To achieve error ϵ, the total number of steps used in this simulation should be $2m = O((ht)^2/\epsilon)$, which is indeed polynomial in n, t, and $1/\epsilon$.[2]

[2] The requirement that H_1 and H_2 be efficiently simulable means that h can be at most $\mathrm{poly}(n)$.

However, it is somewhat unappealing that to simulate an evolution for time t, we need a number of steps proportional to t^2. Fortunately, the situation can be improved if we use higher-order approximations of (1.3). For example, a second-order expansion gives

$$\left(e^{-iH_1 t/2m} e^{-iH_2 t/m} e^{-iH_1 t/2m}\right)^m = e^{-i(H_1+H_2)t} + O((ht)^3/m^2)\,. \tag{1.8}$$

To achieve error ϵ, this approximation requires $2m+1 = O((ht)^{3/2}/\sqrt{\epsilon})$ steps. In general, using a pth order approximation (which can be constructed systematically [171, 36]), the error is $\epsilon = O((ht)^{p+1}/m^p)$, so that $O((ht)^{p/(p-1)}/\epsilon^{1/(p-1)})$ steps suffice to give error ϵ. In this expression, the big-O constant depends on p. Nevertheless, for any desired $\delta > 0$, there is a sufficiently large (but constant) p such that the number of steps in the simulation is $O(t^{1+\delta})$.[3] Whether the simulation can be done in truly linear time in the general case seems to be an open question.

Note that using the Baker-Campbell-Hausdorff formula, the error estimate in the first-order approximation (1.7) can be improved to use $h^2 = \|[H_1, H_2]\|$. In certain cases this allows for a more efficient simulation. We will see an example of such a case in Section 4.6.

A Hamiltonian that is a sum of polynomially many terms can be efficiently simulated by composing Rule 1.4. However, it is more efficient to directly use an approximation to the identity

$$e^{-i(H_1+\cdots+H_k)t} = \lim_{m\to\infty} \left(e^{-iH_1 t/m} \cdots e^{-iH_k t/m}\right)^m\,. \tag{1.9}$$

For example, in the first-order approximation, $O(k(ht)^2/\epsilon)$ steps suffice to give error ϵ, where $h^2 = \max_{j,j'} \left\|[H_j, H_{j'}]\right\|$. Just as in the case with only two terms, such approximations can be constructed systematically to arbitrarily high order, giving a simulation using $O(k(ht)^{p/(p-1)}/\epsilon^{1/(p-1)})$ steps, where $h = \max_j \|H_j\|$ [171, 36].

Combining Rules 1.2 and 1.4, we see that any real linear combination of Hamiltonians can be efficiently simulated. Another way of combining Hamiltonians comes from commutation:

Rule 1.5 (Commutation). *If H_1 and H_2 can be efficiently simulated, then $i[H_1, H_2]$ can be efficiently simulated.*

This rule is a consequence of the identity

$$e^{[H_1,H_2]t} = \lim_{m\to\infty} \left(e^{-iH_1 t/\sqrt{m}} e^{-iH_2 t/\sqrt{m}} e^{iH_1 t/\sqrt{m}} e^{iH_2 t/\sqrt{m}}\right)^m\,, \tag{1.10}$$

which can be approximated with a finite number of terms in a similar way to the approximation of (1.3). Using Rules 1.2, 1.4, and 1.5, it is possible to simulate any Hamiltonian in the Lie algebra generated by a set of Hamiltonians (although this simulation is not necessarily efficient).

Finally, we present a simple rule for the simulation of diagonal Hamiltonians:

Rule 1.6 (Diagonal Hamiltonians). *If H is diagonal in the computational basis and the diagonal element $d(a) = \langle a|H|a\rangle$ can be efficiently computed for any a, then H can be efficiently simulated.*

This rule follows from the simple quantum circuit shown in Figure 1-1. We assume that the diagonal element $d(a)$ is expressed as a binary number with k bits of precision. This circuit

[3] This result can be improved, but only very slightly, by using an order p that increases with t.

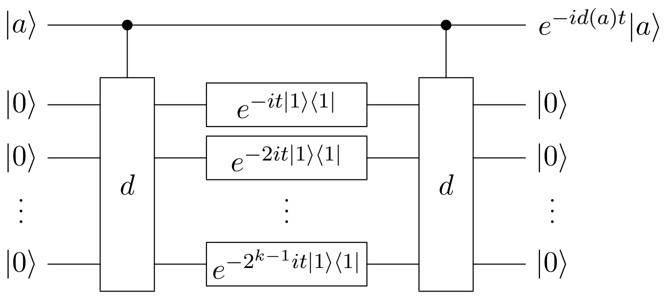

Figure 1-1: A circuit for implementing Rule 1.6.

transforms a computational basis state $|a\rangle$, together with a k-qubit ancilla state $|0\rangle$, as

$$\begin{aligned}|a,0\rangle &\to |a,d(a)\rangle && (1.11)\\ &\to e^{-itd(a)}|a,d(a)\rangle && (1.12)\\ &\to e^{-itd(a)}|a,0\rangle && (1.13)\\ &= e^{-iHt}|a\rangle|0\rangle\,, && (1.14)\end{aligned}$$

which shows that the circuit simulates H. Together with Rule 1.3, this allows us to simulate any Hamiltonian that can be efficiently diagonalized, and whose eigenvalues can be efficiently computed.

1.3 Sparse Hamiltonians

By combining some of the simple rules described above with classical results on local graph coloring, we arrive at a rule that allows us to simulate any Hamiltonian that is sufficiently sparse. This rule will be especially useful for implementing quantum walks on graphs, as discussed in Chapter 3, but it can also be applied to the simulation of physical systems and the implementation of other continuous-time quantum algorithms. The rule is

Rule 1.7 (Sparse Hamiltonians). *Suppose that for any a, one can efficiently compute all the values of b for which $\langle a|H|b\rangle$ is nonzero. Then H can be efficiently simulated.*

This rule was first explicitly stated by Aharonov and Ta-Shma [6]; it also follows immediately from Theorem 1 of [40] together with older (and stronger) classical results on local graph coloring [93, 128, 129, 173]. Coloring the graph of nonzero matrix elements of H allows us to break the Hamiltonian into simple pieces that can be easily simulated on their own. The classical result we need to prove Rule 1.7 is the following:

Lemma 1.1 (Linial [128, 129]). *Suppose we are given an undirected graph G with N vertices and maximum degree d, and that we can efficiently compute the neighbors of a given vertex. Then there is an efficiently computable function $c(a,b)=c(b,a)$ taking $O(d^2\log^2 N)$ values such that for all a, $c(a,b)=c(a,b')$ implies $b=b'$. In other words, $c(a,b)$ is a coloring of G.*

In fact, Linial proves that $O(d^2\log N)$ colors are sufficient if we do not require $c(a,b)$ to be efficiently computable, and comments that Example 3.2 from [74] gives an efficient construction with $O(d^2\log^2 N)$ colors. Instead, we prove this result using an argument along the lines of [6], streamlined to use only $O(d^2\log^2 N)$ colors instead of $O(d^2\log^6 N)$.

Proof. Number the vertices of G from 1 through N. For any vertex a, let $\operatorname{index}(a,b)$ denote the index of vertex b in the list of neighbors of a. Also, let $k(a,b)$ be the smallest k such that $a \neq b \pmod{k}$. Note that $k(a,b) = k(b,a)$, and $k = O(\log N)$.

For $a < b$, define the color of the edge ab to be the 4-tuple

$$c(a,b) := \big(\operatorname{index}(a,b), \operatorname{index}(b,a), k(a,b), b \bmod k(a,b)\big)\,. \tag{1.15}$$

For $a > b$, define $c(a,b) := c(b,a)$.

Now suppose $c(a,b) = c(a,b')$. There are four possible cases:

1. Suppose $a < b$ and $a < b'$. Then the first component of c shows that $\operatorname{index}(a,b) = \operatorname{index}(a,b')$, which implies $b = b'$.

2. Suppose $a > b$ and $a > b'$. Then the second component of c shows that $\operatorname{index}(a,b) = \operatorname{index}(a,b')$, which implies $b = b'$.

3. Suppose $a < b$ and $a > b'$. Then from the third and fourth components of c, $k(a,b) = k(a,b')$ and $a = b \pmod{k(a,b)}$, which is a contradiction.

4. Suppose $a > b$ and $a < b'$. Then from the third and fourth components of c, $k(a,b) = k(a,b')$ and $a = b' \pmod{k(a,b')}$, which is a contradiction.

Each case that does not lead to a contradiction gives rise to a valid coloring, which completes the proof. □

This result uses more colors than are absolutely necessary, since any graph can be colored with at most $d+1$ colors [178]. In practice, graphs with regular structure can typically be colored more efficiently than the above lemma suggests. Also, note that in Lemma 1.1, the color of a given vertex only depends on information about its nearest neighbors. By considering neighbors at a distance $O(\log^* N)$,[4] Linial shows how to color G using only $O(d^2)$ colors [128, 129], which yields an improved simulation of general sparse Hamiltonians [8].

Given Lemma 1.1, we can now complete the proof of Rule 1.7.

Proof. Write H as a diagonal matrix plus a matrix with zeros on the diagonal. The diagonal part can be simulated using Rule 1.6 and combined with the off-diagonal part using Rule 1.4. Therefore, we can assume H has zeros on the diagonal without loss of generality.

Now let G be the graph of nonzero matrix elements of H. The vertices of this graph consist of all the computational basis states, and two vertices have an edge between them if they are connected by a nonzero matrix element of H. Use Lemma 1.1 to color the edges of this graph, and let $v_c(a)$ be the vertex connected to a by an edge of color c (if there is no such vertex, it does not matter how $v_c(a)$ is defined). Also, let

$$x_c(a) := \operatorname{Re}\langle a|H|v_c(a)\rangle \tag{1.16}$$

$$y_c(a) := \operatorname{Im}\langle a|H|v_c(a)\rangle \tag{1.17}$$

when the vertex a has an incident edge of color c; otherwise, let $x_c(a) = y_c(a) = 0$.

[4] The extremely slowly growing function $\log^* x$ is defined to be the smallest integer y such that $\underbrace{\log\log\cdots\log}_{y} x \leq 1$.

Consider the state space $|a, b, z\rangle$, where the space on which H acts corresponds to states of the form $|a, 0, 0\rangle$. By assumption, we can efficiently implement unitary operators V_c, W_c defined by

$$V_c|a, b, z\rangle := |a, b \oplus v_c(a), z \oplus x_c(a)\rangle \tag{1.18}$$

$$W_c|a, b, z\rangle := |a, b \oplus v_c(a), z \oplus y_c(a)\rangle \,, \tag{1.19}$$

where $\oplus$ denotes bitwise addition modulo 2. We can also efficiently implement the inverse operations $V_c^\dagger, W_c^\dagger$, since they are simply $V_c^\dagger = V_c$ and $W_c^\dagger = W_c$. Furthermore, we can efficiently simulate the Hamiltonians S, T where

$$S|a, b, x\rangle := x|b, a, x\rangle \tag{1.20}$$

$$T|a, b, y\rangle := iy|b, a, -y\rangle \tag{1.21}$$

using Rules 1.3 and 1.6, since S and T are easily diagonalized. Therefore, we can efficiently simulate the Hamiltonian

$$\tilde{H} := \sum_c (V_c^\dagger S V_c + W_c^\dagger T W_c)\,. \tag{1.22}$$

When restricted to the subspace of states of the form $|a, 0, 0\rangle$, $\tilde{H}$ acts as H:

$$\tilde{H}|a, 0, 0\rangle = \sum_c [V_c^\dagger S|a, v_c(a), x_c(a)\rangle + W_c^\dagger S|a, v_c(a), y_c(a)\rangle] \tag{1.23}$$

$$= \sum_c [x_c(a) V_c^\dagger |v_c(a), a, x_c(a)\rangle + iy_c(a) W_c^\dagger |v_c(a), a, -y_c(a)\rangle] \tag{1.24}$$

$$= \sum_c [x_c(a) + iy_c(a)]|v_c(a), 0, 0\rangle \tag{1.25}$$

$$= H|a\rangle|0, 0\rangle\,, \tag{1.26}$$

where in the third line we have used the fact that $v_c(v_c(a)) = a$ when a has an incident edge of color c, and that $x_c(a) = y_c(a) = 0$ otherwise. This shows that H can be efficiently simulated. □

1.4 Simulating physical systems

In addition to implementing continuous-time quantum algorithms, the tools described above can also be useful for simulating quantum physics on a quantum computer. In fact, the simulation of quantum systems was the original inspiration for quantum computation, as advocated by Manin [137] and Feynman [84]. We briefly review some results on the simulation of physical systems to illustrate the utility of the rules presented above.

Lloyd pointed out that universal quantum computers can efficiently simulate the dynamics of quantum systems with local interactions [130]. This result follows from Rules 1.1 and 1.4 above. For example, a Hamiltonian of the form (1.1) can be efficiently simulated using quantum circuits. This also shows that we have not lost computational power by considering Hamiltonians satisfying Definition 1.1 rather than the "physically reasonable" Hamiltonians (1.1).

In [130], Lloyd comments that even with the first-order approximation (1.7), the amount of *time* for which the Hamiltonians H_1 and H_2 act is equal to twice the total simulation time t; in other words, it is linear in t. However, this remark is not entirely satisfying from a computational perspective since the total number of *steps* of the simulation is proportional to t^2. As discussed above, higher-order approximations can bring the number of computational steps down to $O(t^{1+\delta})$ for any arbitrarily small positive δ, but it is not clear if a general simulation can be performed in only $O(t)$ steps.

Further quantum simulation results were obtained by Weisner and Zalka, who considered the problem of simulating the dynamics of quantum mechanical particles [184, 189]. Their essential idea can be understood by looking at the simplest case, a single particle of mass m in a one-dimensional potential, with the Hamiltonian

$$H = \frac{p^2}{2m} + V(x)\,. \tag{1.27}$$

Here $p^2 = -\frac{\mathrm{d}^2}{\mathrm{d}x^2}$ is the square of the momentum operator, and the potential $V(x)$ is an arbitrary (but known) function of position. To simulate this Hamiltonian on a digital quantum computer, we must approximate the continuous degree of freedom x using a lattice. For example, we can let an n-qubit computational basis state $|x\rangle$ be the binary representation of the point at location lx, where l is the lattice spacing. Then we can approximate p^2 by the operator

$$P^2|x\rangle := \frac{1}{l^2}(2|x\rangle - |x+1\rangle - |x-1\rangle)\,, \tag{1.28}$$

a discrete approximation to minus the second derivative. The operator P^2 becomes a good approximation to the continuum operator p^2 in the limit $l \to 0$. The momentum operator (as well as its discretization) is diagonal in the Fourier basis, and the Fourier transform can be implemented efficiently using quantum circuits [54]. Therefore, the dynamics according to (1.27) can be simulated using Rules 1.3, 1.4, and 1.6.

Note that the simulation of the discretization of (1.27) also follows immediately from Rule 1.7, since the Hamiltonian is quite sparse. The graph of nonzero matrix elements is simply a line, so it can be colored using only two colors, corresponding to the momentum operator acting on even and odd sites. The resulting simulation does not make explicit use of the Fourier transform, and is actually more efficient than the one proposed in [184, 189].

This approach can be straightforwardly generalized to many-particle quantum dynamics, including interactions, in any number of dimensions. In principle, one can even simulate the dynamics of a quantum field theory, but the technicalities of such a simulation have not been worked out in detail. Although there is a substantial savings in memory over classical simulation, there are still technical obstacles to overcome. For example, the problem of fermion doubling is a discretization effect, and does not simply disappear by using a quantum computer. (We will encounter a phenomenon related to fermion doubling in Section 4.5.)

In this chapter, we have focused on the simulation of Hamiltonian *dynamics*, assuming that the final measurement of the system is a simple one such as a measurement in the computational basis. However, especially in a simulation of a physical system, one may be interested in measuring a more complicated observable. Fortunately, any Hermitian operator that can be efficiently simulated (viewing it as the Hamiltonian of a quantum system) can also be efficiently *measured* using von Neumann's formulation of the measurement process [150]. In fact, this idea is one of the essential ingredients in many of the known fast quantum algorithms, including Shor's factoring algorithm [165]. We will describe this connection in greater detail in Section 2.5, where we explain how measurement can be used to realize adiabatic quantum algorithms.

1.5 Time-independent local Hamiltonians are universal

In [85], Feynman presented a quantum mechanical model of a computer using local, time-independent Hamiltonian dynamics.[5] The motivation for this model was to show that quantum mechanics does not pose barriers to building a classical computer, despite quantum effects such as the uncertainty principle. Feynman showed that any sequence of reversible classical logic gates can be efficiently simulated using local Hamiltonian dynamics. However, his model applies equally well to simulate any quantum circuit.

Given a k-gate quantum circuit on n qubits, $U_k \cdots U_2 U_1$, let

$$H := \sum_{j=1}^{k} H_j \tag{1.29}$$

where

$$H_j := U_j \otimes |j\rangle\langle j-1| + U_j^\dagger \otimes |j-1\rangle\langle j| \,. \tag{1.30}$$

Here the first register consists of n qubits, and the second register stores a quantum state in a $(k+1)$-dimensional space spanned by states $|j\rangle$ for $j \in \{0, 1, \ldots, k\}$. The second register acts as a counter that records the progress of the computation. Later, we will show how to represent the counter using qubits, but for now, we treat it as a convenient abstraction.

If we start the computer in the state $|\psi\rangle|0\rangle$, then the evolution remains in the subspace spanned by the $k+1$ states $|\psi_j\rangle := U_j \cdots U_1 |\psi\rangle |j\rangle$. In this subspace, the nonzero matrix elements of H are

$$\langle \psi_j | H | \psi_{j\pm 1} \rangle = 1 \,, \tag{1.31}$$

so the evolution is the same as that of a free particle propagating on a discretized line segment. Such a particle moves with constant speed, so in a time proportional to k, the initial state $|\psi_0\rangle$ will evolve to a state with substantial overlap on the state $|\psi_k\rangle = U_k \cdots U_1 |\psi\rangle |k\rangle$, the final state of the computation. Arguments given in Sections 3.3.2 and 5.2 show that for large k,

$$|\langle \psi_k | e^{-iHk/2} | \psi_0 \rangle|^2 = O(k^{-2/3}) \,, \tag{1.32}$$

so that after time $k/2$, a measurement of the counter will yield the result k, and hence give the final state of the computation, with a substantial probability.

The success probability of Feynman's computer can be made close to 1 by a variety of techniques. The simplest approach is to repeat the process $O(k^{2/3})$ times. Alternatively, as Feynman suggests, the success probability can be made arbitrarily close to 1 in single shot by preparing the initial state in a narrow wave packet that will propagate ballistically without substantial spreading. But perhaps the best approach is to make the process perfect

[5] Feynman's model has also been useful for two other quantum computing applications, formulating a complete problem for a quantum analog of the complexity class NP [120] and showing the universality of adiabatic quantum computation [6, 5].

by changing the Hamiltonian to [95]

$$H' := \sum_{j=1}^{k} \sqrt{j(k+1-j)}\, H_j \,. \tag{1.33}$$

In this case, the choice $t = \pi$ gives the exact transformation $e^{-iH't}|\psi_0\rangle = |\psi_k\rangle$. This result can be understood by viewing $|\psi_j\rangle$ as a state of total angular momentum $\frac{k}{2}(\frac{k}{2}+1)$ with z component $j - \frac{k}{2}$. Then H' is simply the x component of angular momentum, which rotates between the states with z component $\pm\frac{k}{2}$ in time π. Alternatively, in the terminology of Section 5.2, H' can be viewed as the Hamiltonian of the column subspace of a hypercube.

In the Hamiltonians (1.30) and (1.33), the counter space is not represented using qubits. However, we can easily create a Hamiltonian expressed entirely in terms of $k+1$ qubits using a unary representation of the counter. Let

$$|j\rangle := |\underbrace{0\cdots0}_{j}1\underbrace{0\cdots0}_{k-j}\rangle\,; \tag{1.34}$$

then

$$|j+1\rangle\langle j| = \sigma_+^{(j+1)}\sigma_-^{(j)} \tag{1.35}$$

where

$$\sigma_+ := \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}\,, \quad \sigma_- := \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\,. \tag{1.36}$$

With these definitions, the action of H_j is unchanged. If the quantum circuit consists of one- and two-qubit gates, then the Hamiltonians (1.30) and (1.33) are local in the sense that the interactions involve at most four qubits. In other words, the Hamiltonian is “physically reasonable” in the spirit of (1.1) (albeit without spatial locality).

This construction shows that even a time-independent Hamiltonian of a particularly simple form can be universal for quantum computation. Of course, this does not mean that it is always best to think about quantum algorithms in such terms. However, it is not surprising that certain problems are naturally approached in a Hamiltonian formulation, as we will see.

Chapter 2

Adiabatic quantum computation

2.1 Introduction

In this chapter, we consider a class of Hamiltonian-based quantum algorithms known collectively as quantum computation by adiabatic evolution. This approach was proposed by Farhi, Goldstone, Gutmann, and Sipser as a general way of solving combinatorial search problems on a quantum computer [80].[1] Whereas a conventional quantum circuit is implemented as a sequence of discrete unitary transformations whose physical realization involve many energy levels of the computer, an adiabatic algorithm works by keeping the state of the quantum computer close to the instantaneous ground state of a Hamiltonian that varies continuously in time.

The adiabatic algorithm works by applying a time-dependent Hamiltonian that interpolates smoothly from an initial Hamiltonian whose ground state is easily prepared to a final Hamiltonian whose ground state encodes the solution to the problem. If the Hamiltonian varies sufficiently slowly, then the quantum adiabatic theorem guarantees that the final state of the quantum computer will be close to the ground state of the final Hamiltonian, so a measurement of the final state will yield a solution of the problem with high probability. This method will surely succeed if the Hamiltonian changes slowly. But how slow is slow enough?

Unfortunately, this question has proved difficult to analyze in general. Some numerical evidence suggests the possibility that the adiabatic algorithm might efficiently solve computationally interesting instances of hard combinatorial search problems, outperforming classical methods [77, 43, 79, 140, 106]. However, whether the adiabatic approach to combinatorial optimization provides substantial speedup over classical methods remains an interesting open question.[2] As discussed below, the time required by the algorithm for a particular instance can be related to the minimum gap Δ between the instantaneous ground state and the rest of the spectrum. Roughly speaking, the required time goes like $1/\Delta^2$. Thus, if $1/\Delta^2$ increases only polynomially with the size of the problem, then so does the time required to run the algorithm. However, determining Δ has not been possible in general.

[1] Closely related techniques have been proposed by Kadowaki and Nishimori [112] and Hogg [105]. The former approach has been tested experimentally (in conjunction with a cooling procedure) by Brooke, Bitko, Rosenbaum, and Aeppli [35].

[2] As mentioned in Section 1.1, there is a sense in which adiabatic quantum computation is computationally equivalent to any other model of quantum computation. However, while encouraging, this fact does not shed light on the question of whether the *algorithmic idea* of using adiabatic quantum computation to perform combinatorial search can provide speedup over classical methods.

Our goal in this chapter is not to address the computational power of adiabatic quantum computation. Rather, we describe two related ideas that may prove useful in the implementation of the adiabatic algorithm. In Section 2.4, we argue that adiabatic quantum computation enjoys an intrinsic robustness to certain kinds of errors, and we demonstrate this phenomenon through numerical simulations. In Section 2.5, we show how the essential idea of the adiabatic algorithm can be implemented in a conceptually simpler way using only a sequence of measurements, which in turn can be realized dynamically by Hamiltonian evolution. We also show how the measurement-based approach can be used to achieve quadratic speedup for unstructured search.

2.2 Review of adiabatic quantum computation

In this section, we briefly review the idea of adiabatic quantum computation. Let $h(z)$ be a function of n bits $z = (z_1, z_2, \ldots, z_n)$, and consider the computational problem of finding a value of z that minimizes $h(z)$. We will typically be interested in the case where this value of z is unique. We may associate with this function the Hermitian operator

$$H_P = \sum_{z=0}^{2^n-1} h(z)|z\rangle\langle z| \,, \tag{2.1}$$

so that the computational basis state $|z\rangle$ is an eigenstate of H_P with eigenvalue $h(z)$. Then the problem is to determine which state $|z\rangle$ is the ground state (the eigenstate with lowest eigenvalue) of H_P. We refer to H_P as the *problem Hamiltonian*.

The strategy for finding the ground state of H_P is to prepare the ground state of some other *beginning Hamiltonian* H_B and slowly interpolate to H_P. In other words, we consider a one-parameter family of Hamiltonians $\tilde{H}(s)$ for $s \in [0,1]$ such that $\tilde{H}(0) = H_B$, $\tilde{H}(1) = H_P$, and $\tilde{H}(s)$ is a smooth function of s. For example, one simple choice for $\tilde{H}(s)$ is linear interpolation,

$$\tilde{H}(s) = (1-s)H_B + sH_P \,. \tag{2.2}$$

We prepare the ground state of H_B at time $t = 0$, and then the state evolves from $t = 0$ to $t = T$ according to the Schrödinger equation (0.1), where the Hamiltonian is

$$H(t) = \tilde{H}(t/T) \,. \tag{2.3}$$

At time T (the *run time* of the algorithm), we measure the state in the computational basis. If we let $|E_0(1)\rangle$ denote the (unique) ground state of H_P for a given instance of the problem, then the *success probability* of the algorithm for this instance is $|\langle E_0(1)|\psi(T)\rangle|^2$.

Does the algorithm work? According to the quantum adiabatic theorem [115, 142], if there is a nonzero gap between the ground state and the first excited state of $\tilde{H}(s)$ for all $s \in [0,1]$, then the success probability approaches 1 in the limit $T \to \infty$. Furthermore, level crossings are non-generic in the absence of symmetries, so a non-vanishing gap is expected if H_B does not commute with H_P. Thus, the success probability of the algorithm will be high if the evolution time T is large enough. The question is, how large a T is large enough so that the success probability is larger than some fixed constant?

We can reformulate this question in terms of the quantities

$$\Delta := \min_{s\in[0,1]} \Delta(s) \,, \quad \Gamma := \max_{s\in[0,1]} \Gamma(s) \tag{2.4}$$

where

$$\Delta(s) := E_1(s) - E_0(s) \tag{2.5}$$

$$\Gamma^2(s) := \langle E_0(s)|(\tfrac{\mathrm{d}\tilde{H}}{\mathrm{d}s})^2|E_0(s)\rangle - \langle E_0(s)|\tfrac{\mathrm{d}\tilde{H}}{\mathrm{d}s}|E_0(s)\rangle^2 \,. \tag{2.6}$$

Here $|E_j(s)\rangle$ is the eigenstate of $\tilde{H}(s)$ with eigenvalue $E_j(s)$, with $E_0(s) \le E_1(s) \le \cdots \le E_{2^n-1}(s)$. By calculating the transition probability to lowest order in the adiabatic expansion [142], one finds that the probability of a transition from ground state to first excited state is small provided that the run time T satisfies

$$T \gg \frac{\Gamma}{\Delta^2} \,. \tag{2.7}$$

Note that the quantity Γ accounts for the possibility of transitions to all possible excited states. In general, the required run time T will be bounded by a polynomial in n so long as Δ and Γ are polynomially bounded. For most problems of interest, Γ is polynomially bounded, so we only have to consider the behavior of Δ.

The proof of the adiabatic theorem is somewhat involved, and will not be presented here. However, the basic idea can be understood as follows. By rescaling the time, we can think of the evolution as taking place in the unit time interval between $s = 0$ and $s = 1$, but in that case the energy eigenvalues are rescaled by the factor T. Roughly speaking, we can think of $\mathrm{d}\tilde{H}(s)/\mathrm{d}s$ as a perturbation that couples the levels of the instantaneous Hamiltonian $\tilde{H}(s)$, and that can drive a transition from $|E_0(s)\rangle$ to $|E_1(s)\rangle$. But if T is large, the effects of this perturbation are washed out by the rapid oscillations of the relative phase $\exp[-iT\int_0^s \mathrm{d}s'(E_1(s') - E_0(s'))]$.

To implement an adiabatic quantum computation, it is sufficient to build a universal quantum computer and employ the simulation techniques discussed in Chapter 1, assuming they apply to the desired Hamiltonian $H(t)$. However, to take advantage of the robustness properties we will describe in Section 2.4, it may be desirable to directly build a quantum system whose Hamiltonian can be smoothly modified from H_B to H_P. In this case, the Hamiltonian can be regarded as reasonable only if it is local, that is, if it can be expressed as a sum of terms, where each term acts on a constant number of qubits (a number that does not grow with n). Many combinatorial search problems (e.g., 3SAT) can be formulated as a search for a minimum of a function that is local in this sense. Along with a local choice of H_B, this results in a full $H(t)$ that is also local.

In fact, a direct physical implementation of the continuously varying $H(t)$ would presumably be possible only under a somewhat stronger locality condition. We might require that each qubit is coupled to only a few other qubits, or perhaps that the qubits can be physically arranged in such a way that the interactions are spatially local. Fortunately, there are interesting computational problems that have such forms, such as 3SAT restricted to having each bit involved in only three clauses or the problem of finding the ground state of a spin glass on a three-dimensional lattice [14]. However, for the purposes of the simulations described in Section 2.4, we will only consider small instances, and since we do not have a specific physical implementation in mind, we will not concern ourselves with the spatial arrangement of the qubits. We note that some of these implementation issues have been addressed in specific proposals [113, 114].

2.3 An example: The exact cover problem

To be concrete, we describe the details of the adiabatic algorithm for a particular combinatorial search problem known as *three-bit exact cover*, or EC3. This is the problem we will consider in our simulations in Section 2.4, and it also serves to illustrate the general idea. An n-bit instance of EC3 consists of a set of clauses, each of which specifies three of the n bits. A clause is said to be satisfied if and only if exactly one of its bits has the value 1. The problem is to determine if any of the 2^n assignments of the n bits satisfies all of the clauses.

For this problem, the function $h(z)$ is a sum

$$h(z) = \sum_C h_C(z_{i_C}, z_{j_C}, z_{k_C}) \tag{2.8}$$

of three-bit clauses, where

$$h_C(z_{i_C}, z_{j_C}, z_{k_C}) = \begin{cases} 0 & (z_{i_C}, z_{j_C}, z_{k_C}) \text{ satisfies clause } C \\ 1 & (z_{i_C}, z_{j_C}, z_{k_C}) \text{ violates clause } C\,. \end{cases} \tag{2.9}$$

The value of the function $h(z)$ is the number of clauses that are violated; in particular, $h(z) = 0$ if and only if z is an assignment that satisfies all the clauses.

To solve EC3 by the adiabatic algorithm, a sensible choice for the beginning Hamiltonian is

$$H_B = \sum_C H_{B,C}\,, \tag{2.10}$$

where

$$H_{B,C} = \frac{1}{2}\left(1 - \sigma_x^{(i_C)}\right) + \frac{1}{2}\left(1 - \sigma_x^{(j_C)}\right) + \frac{1}{2}\left(1 - \sigma_x^{(k_C)}\right)\,, \tag{2.11}$$

which has the ground state

$$|\psi(0)\rangle = \frac{1}{2^{n/2}} \sum_{z=0}^{2^n-1} |z\rangle\,. \tag{2.12}$$

Using the linear interpolation (2.2), the resulting $H(t)$ is local in the sense that it is a sum of terms, each of which acts on only a few qubits. A stronger kind of locality can be imposed by restricting the instances so that each bit is involved in at most a fixed number of clauses. The computational complexity of the problem is unchanged by this restriction.

Numerical studies of the adiabatic algorithm applied to this problem were reported in [77, 79]. Instances of EC3 with n bits were generated by adding random clauses until there was a unique satisfying assignment, giving a distribution of instances that one might expect to be computationally difficult to solve. The results for a small number of bits ($n \leq 20$) were consistent with the possibility that the adiabatic algorithm requires a time that grows only as a polynomial in n for typical instances drawn from this distribution. If this is the case, then the gap Δ does not shrink exponentially in n. Although the typical spacing between levels must be exponentially small, since there are an exponential number of levels in a polynomial range of energies, it is possible that the gap at the bottom is larger. For example, Figure 2-1 shows the spectrum of a randomly generated seven-bit instance of EC3. The gap at the bottom of the spectrum is reasonably large compared to the typical spacing. This feature is not specific to this one instance, but is characteristic of randomly

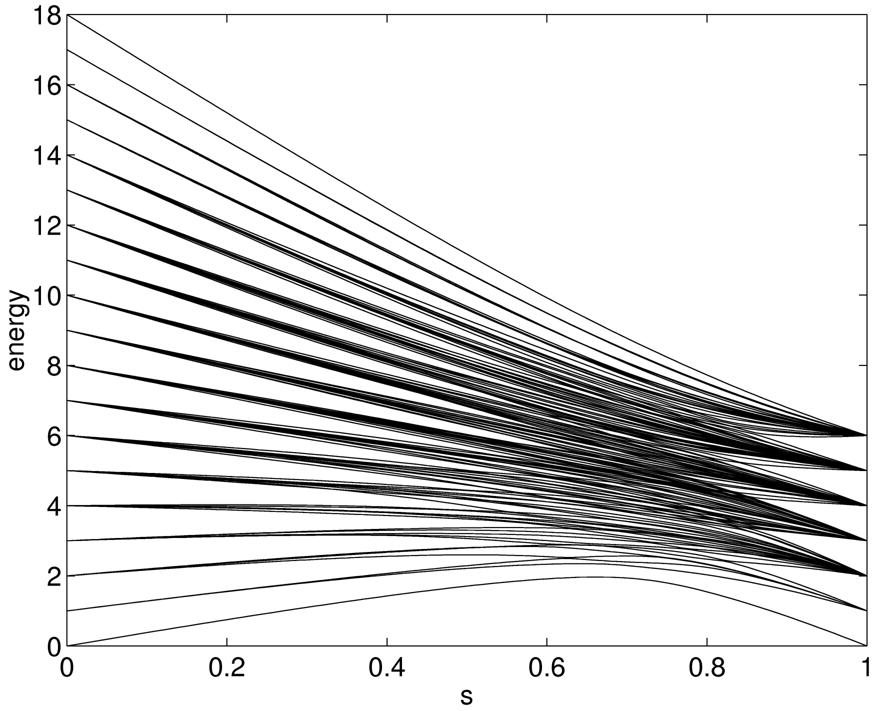

Figure 2-1: Spectrum of a randomly generated $n = 7$ bit instance of EC3 with a unique satisfying assignment. Note that the energy gap between the ground state and the first excited state is significantly larger than all other gaps. An expanded view would show that there are no level crossings anywhere in the spectrum (except for the degeneracies at $s = 0$ and $s = 1$).

generated instances, at least for $n \lesssim 10$, beyond which the repeated matrix diagonalization required to create a picture of the spectrum becomes computationally costly. A large gap makes an instance readily solvable by the adiabatic algorithm, and also provides robustness against thermal transitions out of the ground state, as described below.

2.4 Robustness

Since quantum computers are far more susceptible to making errors than classical digital computers, fault-tolerant protocols will be necessary for the operation of large-scale quantum computers. General procedures have been developed that allow any quantum algorithm to be implemented fault tolerantly on a universal quantum computer [167], but these involve a substantial computational overhead. Therefore, it would be highly advantageous to weave fault tolerance into the design of our quantum hardware.

In this section, we will regard adiabatic quantum computation not as a convenient language for describing a class of quantum circuits, but as a proposed physical implementation of quantum information processing. We do not cast the algorithm into the conventional quantum computing paradigm by approximating it as a sequence of discrete unitary transformations acting on a few qubits at a time. Instead, suppose we can design a physical device that implements the required time-dependent Hamiltonian with reasonable accuracy. We then imagine implementing the algorithm by slowly changing the parameters that control the physical Hamiltonian. How well does such a quantum computer resist decoherence, and how well does it perform if the algorithm is imperfectly implemented?

Regarding resistance to decoherence, we can make a few simple observations. The phase of the ground state has no effect on the efficacy of the algorithm, and therefore dephasing in the energy eigenstate basis is presumably harmless. Only the interactions with the environment that induce transitions between eigenstates of the Hamiltonian might cause trouble. In principle, these can be well controlled by running the algorithm at a temperature that is small compared to the minimum gap Δ.[3] If Δ decreases slowly as the size of the problem increases, then the resources required to run at a sufficiently low temperature may be reasonable. Since the adiabatic method is only efficient if Δ is not too small, we conclude that whenever the method works on a perfectly functioning quantum computer, it is robust against decoherence.

In addition to environmental decoherence, we must also consider the consequences of imperfect implementation. Our chosen algorithm may call for the time-dependent Hamiltonian $H(t)$, but when we run the algorithm, the actual Hamiltonian will be $H(t) + K(t)$, where $K(t)$ represents errors in the implementation. An interesting feature of adiabatic quantum computation is that $K(t)$ need not remain small during the evolution in order for the algorithm to work effectively. A reasonably large excursion away from the intended Hamiltonian is acceptable, as long as $K(t)$ is slowly varying and has initial and final values that are not too large. A very rapidly fluctuating $K(t)$ may also be acceptable, if the characteristic frequency of the fluctuations is large compared to the energy scale of $H(t)$.

Below, we will use numerical simulations to investigate the sensitivity of an adiabatic computer to decohering transitions and to a certain class of unitary perturbations induced by an error Hamiltonian $K(t)$. The results are consistent with the idea that the algorithm remains robust as long as the temperature of the environment is not too high and $K(t)$ varies either sufficiently slowly or sufficiently rapidly. Thus, the adiabatic model illustrates the principle that when the characteristics of the noise are reasonably well understood, it may be possible to design suitable quantum hardware that effectively resists the noise. However, note that some of the effects of decoherence and unitary control error may not be significant for the small problems we are able to study—especially in the case of decoherence, where the time required by the simulation restricts us to systems with only four qubits—and hence our data may not be indicative of the performance of the algorithm working on larger inputs.

In a different guise, the principles that make quantum adiabatic evolution robust also underlie the proposal by Kitaev [119] to employ non-abelian anyons for fault-tolerant quantum computation. The fact that adiabatic evolution incorporates a kind of intrinsic fault tolerance has also been noted in [156, 153, 190, 73, 131, 87].

2.4.1 Decoherence

Perhaps the most significant impediment to building a large-scale quantum computer is the problem of decoherence. No quantum device can be perfectly isolated from its environment, and interactions between a device and its environment will inevitably introduce noise. Fortunately, such effects can be countered using fault-tolerant protocols, but, as mentioned above, these protocols can be costly. Therefore, we would like to consider quantum systems with inherent resistance to decohering effects. If the ground state of our adiabatic quantum computer is separated from the excited states by a sizable energy gap, then we expect it to

[3] We use units in which Boltzmann's constant $k_B = 1$, so that temperature has units of energy.

exhibit such robustness. Here, we consider how the adiabatic algorithm for EC3 is affected by decoherence.

First, we briefly review the Markovian master equation formalism for describing the decohering effects of an environment on a quantum system. Suppose that our quantum computer is a collection of spin-$\frac{1}{2}$ particles interacting with each other according to the Hamiltonian H_S and weakly coupled to a large bath of photons. The total Hamiltonian of the quantum computer and its environment is

$$H = H_S + H_E + \lambda V \,, \tag{2.13}$$

where H_E is the Hamiltonian of its environment, V is an interaction that couples the quantum computer and the photon bath, and λ is a coupling constant. We may describe the state of the quantum computer alone by the density matrix ρ found by tracing over the environmental degrees of freedom. In general, the time evolution of ρ is complicated, but under reasonable assumptions, we can approximate its evolution using a Markovian master equation.

One way of deriving such a master equation is to consider the weak coupling limit, in which $\lambda \ll 1$ [60]. If the environment is very large and only weakly coupled to the quantum computer, it will be essentially unchanged by the interaction. Furthermore, in this limit, we can expect the evolution of the quantum computer to be Markovian, or local in time, if we filter out high-frequency fluctuations by some coarse-graining procedure. Assuming that the combined state of the quantum computer and its environment begins in a product state $\rho(0) \otimes \rho_E$, Davies derives the master equation

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -i[H_S, \rho] + \lambda^2 K^\natural \rho \,, \tag{2.14}$$

where

$$K\rho = -\int_0^\infty \mathrm{d}x \ \mathrm{tr}_E[U(-x)VU(x), [V, \rho \otimes \rho_E]] \tag{2.15}$$

$$K^\natural \rho = \lim_{x\to\infty} \frac{1}{x} \int_0^x \mathrm{d}y \, U(-y)\{K[U(y)\rho U(-y)]\}U(y) \tag{2.16}$$

with

$$U(x) = e^{-ix(H_S + H_E)} \,, \tag{2.17}$$

where we have (temporarily) assumed that H_S is time-independent. Although the $\natural$ operation defined by (2.16) does not appear in some formulations of the Markovian master equation, it appears to be essential for the equation to properly describe the weak coupling limit [69], and in particular, for it to capture the phenomenon of relaxation to thermal equilibrium. The master equation (2.14) has the property that if the environment is in thermal equilibrium at a given temperature, then the decohering transitions drive the quantum computer toward the Gibbs state of H_S at that temperature. While not an exact description of the dynamics, (2.14) should provide a reasonable caricature of a quantum computer in a thermal environment.

Note that (2.14) is derived assuming a time-independent Hamiltonian H_S; with a time-varying $H_S(t)$, we should expect the generator of time evolution at any particular time to depend on the Hamiltonian at all previous times [61]. However, if $H_S(t)$ is slowly varying, then it is a good approximation to imagine that the generator at any particular time depends only on H_S at that time [127]. In particular, since we are interested in nearly adiabatic evolution, $H_S(t)$ varies slowly, so (2.14) remains a good approximation, where at any given time t we compute $K^\natural$ using only $H_S(t)$. Note that with $H_S(t)$ time-dependent, $U(x)$ defined by (2.17) is not the time evolution operator; it depends on the time t only implicitly through $H_S(t)$.

For a system of spins coupled to photons, we choose the interaction

$$V = \sum_i \int_0^\infty \mathrm{d}\omega \, [g(\omega) a_\omega \sigma_+^{(i)} + g^*(\omega) a_\omega^\dagger \sigma_-^{(i)}] \,, \tag{2.18}$$

where $\sum_i$ is a sum over the spins, $\sigma_\pm^{(i)}$ are raising and lowering operators for the ith spin, a_ω is the annihilation operator for the photon mode with frequency ω, and $\lambda g(\omega)$ is the product of the coupling strength and spectral density for that mode. Note that if the coupling strength is frequency-dependent, we can absorb this dependence into $g(\omega)$, leaving λ as a frequency-independent parameter. With this specific choice for V, we can perform the integrals and trace in (2.14)–(2.17). If we assume that all spacings between eigenvalues of H_S are distinct, the resulting expression simplifies considerably, and we find

$$\begin{aligned}\frac{\mathrm{d}\rho}{\mathrm{d}t} =& - i[H_S, \rho] \\ &- \sum_{i,a,b} \left[N_{ba} |g_{ba}|^2 \langle a|\sigma_-^{(i)}|b\rangle\langle b|\sigma_+^{(i)}|a\rangle + (N_{ab}+1)|g_{ab}|^2 \langle b|\sigma_-^{(i)}|a\rangle\langle a|\sigma_+^{(i)}|b\rangle \right] \\ &\qquad \times \left[(|a\rangle\langle a|\rho) + (\rho|a\rangle\langle a|) - 2|b\rangle\langle a|\rho|a\rangle\langle b| \right] ,\end{aligned} \tag{2.19}$$

where the states $|a\rangle$ are the time-dependent instantaneous eigenstates of H_S with energy eigenvalues ω_a,

$$N_{ba} = \frac{1}{\exp[\beta(\omega_b - \omega_a)] - 1} \tag{2.20}$$

is the Bose-Einstein distribution at temperature $1/\beta$, and

$$g_{ba} = \begin{cases} \lambda g(\omega_b - \omega_a) & \omega_b > \omega_a \\ 0 & \omega_b \leq \omega_a \,. \end{cases} \tag{2.21}$$

We simulated the effect of thermal noise by numerically integrating the master equation (2.19) with a Hamiltonian H_S given by (2.3) and with the initial pure state density matrix $\rho(0) = |\psi(0)\rangle\langle\psi(0)|$ given by (2.12). For simplicity, we chose $g(\omega) = 1$ for $\omega \geq 0$ and zero otherwise. Although we would expect that $g(\omega) \to 0$ as $\omega \to \infty$, for the small systems we are able to simulate it should be a reasonable approximation to treat $g(\omega)$ as constant and tune the overall coupling strength using λ^2.

How should we expect the success probability $\langle E_0(1)|\rho(T)|E_0(1)\rangle$, where $|E_0(1)\rangle$ is the ground state of H_P, to depend on the run time T and the temperature? If the run time T is sufficiently long, then regardless of its initial state the quantum computer will come to thermal equilibrium; at the time of the final readout it will be close to the Gibbs state

$$\lim_{T\to\infty} \rho(T) = \frac{e^{-\beta H_P}}{\operatorname{tr} e^{-\beta H_P}} = \rho_P \tag{2.22}$$

of the problem Hamiltonian H_P, and therefore the success probability will be approximately

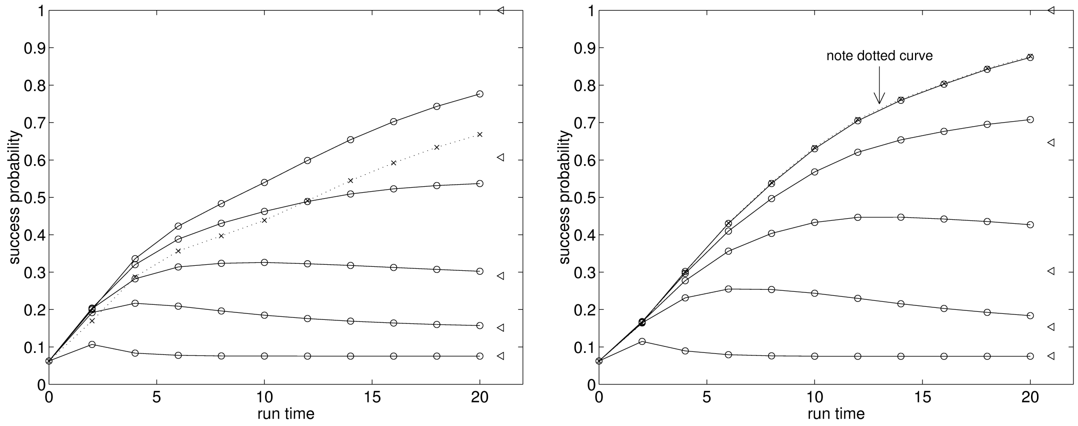

Figure 2-2: The success probability as a function of run time T for two instances of EC3 with $n = 4$ bits. The instance on the left has a gap of $\Delta_1 \approx 0.301$ and the instance on the right has a gap of $\Delta_2 \approx 0.425$. The dotted line shows the behavior of the algorithm with no decoherence, i.e., $\lambda^2 = 0$. Note that in the figure on the right, the dotted curve is partially obscured but can be seen slightly above the topmost solid curve. The solid lines show the behavior of the algorithm in the presence of decoherence with $\lambda^2 = 0.1$ for five different temperatures. The triangles at the far right show the thermal success probabilities $\langle E_0(1)|\rho_P|E_0(1)\rangle$ at each of these temperatures. From top to bottom, the temperatures are 1/10, 1/2, 1, 2, and 10.

$\langle E_0(1)|\rho_P|E_0(1)\rangle$. This probability may be appreciable if the temperature is small compared to the gap between the ground state and first excited state of H_P. Thus one way to find the ground state of H_P is to prepare the computer in any initial state, put it in a cold environment, wait a long time, and measure. However, this thermal relaxation method is not an efficient way to solve hard optimization problems. Although it may work well on some instances of a given problem, this method will not work in cases where the computer can get stuck in local minima from which downward transitions are unlikely. In such cases, the time for equilibration is expected to be exponentially large in n.

Consider an instance with a long equilibration time so that cooling alone is not an efficient way to find the ground state of H_P. It is possible that the minimum gap Δ associated with the quantum algorithm is not small, and the idealized quantum computer, running without decohering effects, would find the ground state of H_P in a short time. In this situation, if we include the coupling of the system to the environment and we run at a temperature much below Δ, then thermal transitions are never likely, and the adiabatic algorithm should perform nearly as well as in the absence of decoherence. But if the temperature is comparable to Δ, then the performance may be significantly degraded.

On the other hand, consider an instance for which the equilibration time is short, so that cooling alone is a good algorithm. Furthermore, suppose that the adiabatic algorithm would find the ground state of H_P in a short time in the absence of decohering effects. In this case, the combined effects of cooling and adiabatic evolution will surely find the ground state of H_P in a short time. But note that Δ alone does not control the success of the algorithm. Even if $H(t)$ changes too quickly for the evolution to be truly adiabatic so that a transition occurs when the gap is small, the system may be cooled back into its ground state at a later time.

Typical results of the simulation are shown in Figure 2-2 for two $n = 4$ bit instances of EC3 with unique satisfying assignments. These two instances have minimum gaps of $\Delta_1 \approx 0.301$ and $\Delta_2 \approx 0.425$. For each instance, we plot the success probability as a function of the run time T. With $\lambda^2 = 0.1$, we consider five temperatures: 1/10, 1/2, 1, 2, and 10. We also present the data with no decoherence ($\lambda^2 = 0$) for comparison.

Unfortunately, the time required to integrate (2.19) grows very rapidly with n. Whereas a state vector contains 2^n entries, the density matrix contains 4^n entries; and in addition, calculating $\mathrm{d}\rho/\mathrm{d}t$ at each timestep requires evaluating a double sum over 2^n energy eigenstates. For this reason, we were only able to consider instances with $n \le 4$.

The results are consistent with our general expectations. In the absence of decoherence, the success probability becomes appreciable for sufficiently long run times. This probability rises faster for the problem with a larger gap. When we add decoherence at high temperature, the success probability never becomes very large (note the lowest curves in Figure 2-2). As the temperature is decreased to a value of order one, the presence of decoherence has a less significant effect on the success probability. In fact, for sufficiently low temperatures, the success probability can actually be higher in the presence of decoherence than when there is no decoherence. This is because the primary effect of decoherence at low temperature is to drive transitions toward the ground state, improving performance.

However, these results do not illustrate a definitive connection between the minimum gap Δ and the temperature above which the algorithm no longer works. These simple $n = 4$ bit instances fall into the second category discussed above: the equilibration time is short, so cooling alone is a good algorithm. In other words, no sharp distinction can be drawn between the run time required for the adiabatic algorithm to perform well in the absence of decoherence and the run time required for equilibration. Accordingly, the dependence of the success probability on temperature and run time is similar for the two instances shown in Figure 2-2, even though the minimum gaps for these instances are somewhat different.

2.4.2 Unitary control error

We now consider how the performance of the adiabatic algorithm for EC3 is affected by adding three different kinds of perturbations to the Hamiltonian. Each perturbation we consider is a sum of single-qubit terms, where each term can be interpreted as a magnetic field pointing in a random direction. To simplify our analysis, we assume that the magnitude of the magnetic field is the same for all qubits, but its direction varies randomly from qubit to qubit. The perturbations we consider are

$$\tilde{K}_1(s) = C_1 s \sum_{i=1}^{n} \hat{m}_i \cdot \vec{\sigma}^{(i)} , \tag{2.23}$$

$$\tilde{K}_2(s) = C_2 \sin(\pi s) \sum_{i=1}^{n} \hat{m}_i \cdot \vec{\sigma}^{(i)} , \tag{2.24}$$

$$\tilde{K}_3(s) = \frac{1}{2} \sin(C_3 \pi s) \sum_{i=1}^{n} \hat{m}_i \cdot \vec{\sigma}^{(i)} , \tag{2.25}$$

which are added to (2.2) and give a time-dependent Hamiltonian according to (2.3). Each $\hat{m}_i$ is a randomly generated real three-component vector with unit length, C_1 and C_2 are real numbers, and C_3 is a nonnegative integer.

The adiabatic algorithm was simulated by numerically solving the time-dependent Schrödinger equation with initial state $|\psi(0)\rangle$ given by (2.12) and Hamiltonian $\tilde{H}(t/T) + \tilde{K}_j(t/T)$

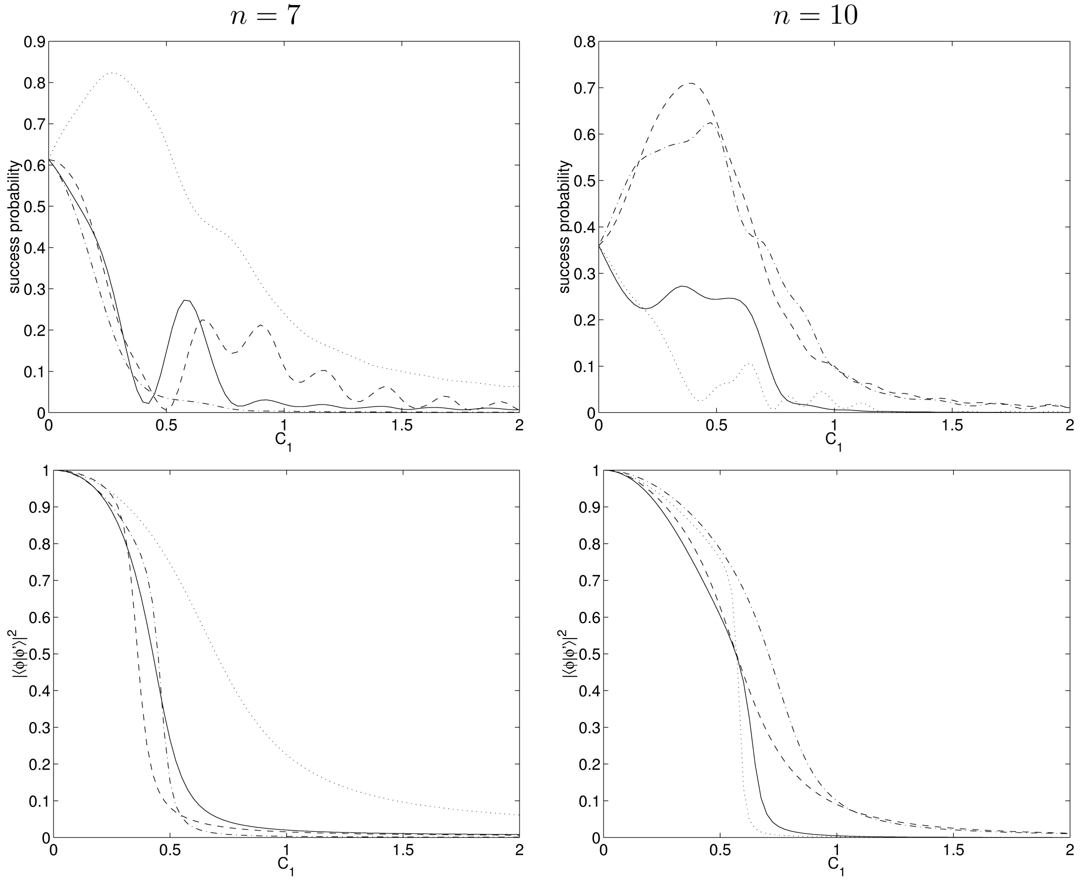

Figure 2-3: (Top) The success probability of the adiabatic algorithm for two randomly generated instances of EC3 with $n = 7$ bits (left) and $n = 10$ bits (right) under the perturbation K_1 defined by (2.23) for four different sets of magnetic field directions. For each n, the run time is the same for each random perturbation. (Bottom) The corresponding overlaps $|\langle E_0(1)|E_0(1)'\rangle|^2$ of the ground state $|E_0(1)\rangle$ of H_P with the perturbed ground state $|E_0(1)'\rangle$ at $s = 1$.

for a given $j \in \{1, 2, 3\}$. As in [77, 43, 79], we used a fifth-order Runge-Kutta method with variable step-size, and checked the accuracy by verifying that the norm of the state was maintained to one part in a thousand. For a specified value of n, we randomly generated an instance of EC3 with a unique satisfying assignment. Then we randomly generated several different values of the magnetic field directions $\{\hat{m}_i\}$. For each instance of the problem and the magnetic field, the run time was chosen so that the success probability without the perturbation was reasonably high. With this run time fixed, we then determined the success probability for varying values of the relevant C_j.

First, we consider the perturbation K_1. Since it turns on at a constant rate, this perturbation can be thought of as an error in H_P. Note that with $C_1 \neq 0$, the final Hamiltonian is not simply H_P, so the algorithm will not work exactly even in the adiabatic limit $T \to \infty$. This perturbation is potentially dangerous because of the way its effect scales with the number of bits n. Indeed, consider the case where H_P can be separated into a sum of Hamiltonians acting separately on each qubit. If adding K_1 reduces the overlap of the ground state $|E_0(1)\rangle$ of H_P with the perturbed ground state $|E_0(1)'\rangle$ by some fixed

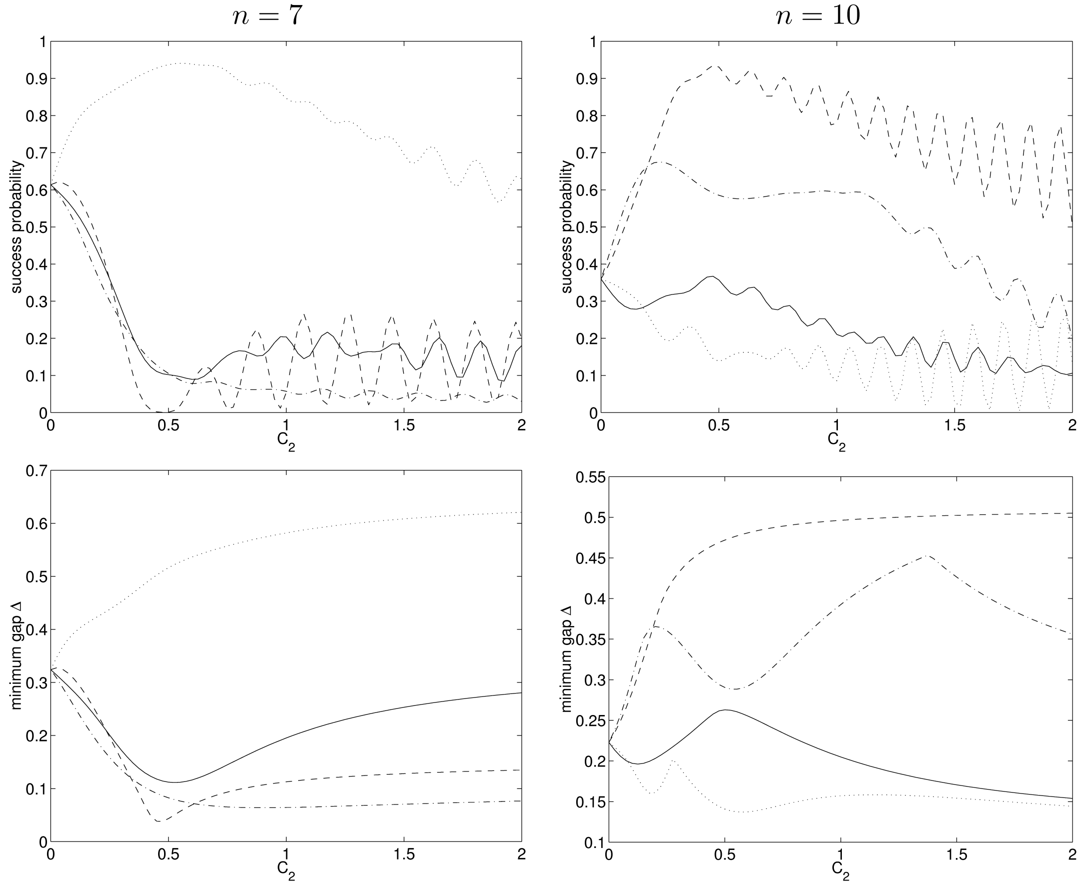

Figure 2-4: (Top) The success probability of the adiabatic algorithm for the same instances used in Figure 2-3 under the perturbation K_2 defined by (2.24). The four different magnetic field directions for each instance are also the same as in Figure 2-3. (Bottom) The minimum gap Δ in the perturbed problem.

value ϵ for each of the n qubits, then the total overlap is $(1-\epsilon)^n$, which is exponentially small in the number of bits. Thus the algorithm clearly fails in this factorized case. In general, if the magnitude of K_1 is independent of n, then we might expect the algorithm to fail. However, if the magnitude of K_1 falls as $1/n$ or faster, then the shift of the ground state may be small enough (as it would be in the factorized case) that the algorithm is not significantly affected. Note that for any n there is some value of C_1 that is small enough that the disadvantage of reduced overlap with the ground state of H_P may be overcome if the perturbation happens to increase the minimum gap Δ. For this reason, we expect to sometimes see an increase in success probability for small C_1 that goes away as C_1 is increased.

The effect of the perturbation K_1 is shown in Figure 2-3 for $n = 7$ and $n = 10$ bit instances of EC3, with four different randomly generated sets of magnetic field directions for each instance. The run time is chosen such that for $C_1 = 0$, the success probability is around 1/2. The top plots show that for small C_1, the success probability is not strongly suppressed; in fact, in some cases it is significantly enhanced. For large enough C_1, the success probability is heavily suppressed. The bottom plots show the overlap $|\langle E_0(1)|E_0(1)'\rangle|^2$ between the ground state of H_P and the actual ground state in the presence of the perturbation. As we expect, the suppression of the success probability is correlated with the amount of overlap. We also studied a similar perturbation in which s is replaced by $1 - s$, which can be thought of as an error in H_B. Unsurprisingly, the results were qualitatively similar.

Next, we consider the low-frequency perturbation K_2. The period of oscillation is chosen such that the perturbation vanishes at $t = 0$ and $t = T$, so the perturbation does not affect the algorithm in the adiabatic limit. Since the success probability is quite sensitive to the value of the minimum gap Δ, and it is not *a priori* obvious whether a perturbation will increase or decrease Δ, we can guess that turning on a nonzero value of C_2 can either increase the success probability or decrease it.

Figure 2-4 shows the effect of the perturbation K_2, using the same instances, magnetic field directions, and run times as in Figure 2-3. The top plots show the success probability as a function of C_2. As in the case of K_1, some perturbations can raise the success probability and some suppress it. Perhaps unsurprisingly, a particular set of magnetic field directions that can raise the success probability under K_1 is also likely to help when K_2 is applied. But unlike K_1, K_2 can improve the success probability even with $C_2 \simeq 2$, where the size of the perturbation is comparable to the size of the unperturbed Hamiltonian. The bottom plots show the minimum gap Δ when the perturbation is added. Note that there is a strong correlation between the success probability and Δ.

For both perturbations K_1 and K_2, similar results have been observed (with fewer data points) for instances with as many as $n = 14$ bits. Figures 2-3 and 2-4 present typical data. For example, for a given instance, typically one or two out of four sets of randomly chosen magnetic field directions led to an improvement in the success probability for some values of C_1 and C_2, compared to the unperturbed case.

Finally, we consider the perturbation K_3, in which the magnitude of the oscillating component is fixed, but we may vary its frequency by varying C_3. As for K_2, the frequency is chosen so that the perturbation vanishes at $t = 0$ and $t = T$. We expect that for C_3 of order one, the perturbation will be likely to excite a transition, and that the success probability will be small. But since both H_B and H_P have a maximum eigenvalue of order n, we can anticipate that for

$$C_3 \gg \frac{nT}{\pi}\,, \tag{2.26}$$

the perturbation will be far from any resonance. Then the probability that the perturbation drives a transition will be low, and the success probability should be comparable to the case where the perturbation vanishes.

Some representative plots of the dependence of the success probability on C_3 are shown in Figure 2-5. Each plot corresponds to a particular randomly generated instance of EC3 (with either $n = 8$ bits or $n = 10$ bits) and a randomly generated set of magnetic field directions. In the top row of plots, the run time is chosen so that the success probability is around 1/8 with the perturbation absent (i.e., $C_3 = 0$). In the bottom row, the run time is doubled. All of the data exhibit the expected qualitative trend. The leftmost point corresponds to $C_3 = 0$. For the smallest values of $C_3 > 0$, the success probability may not be too badly damaged; for somewhat larger values of C_3 it is heavily suppressed; and for sufficiently large C_3 it recovers to a value near the success probability in the absence of the perturbation. The value of nT/π is around 19 and 39 for the upper and lower $n = 8$ plots and is around 38 and 76 for the upper and lower $n = 10$ plots, so the estimate (2.26) turns out to be reasonable.

Another conspicuous feature of the plots in Figure 2-5 is that the success probability

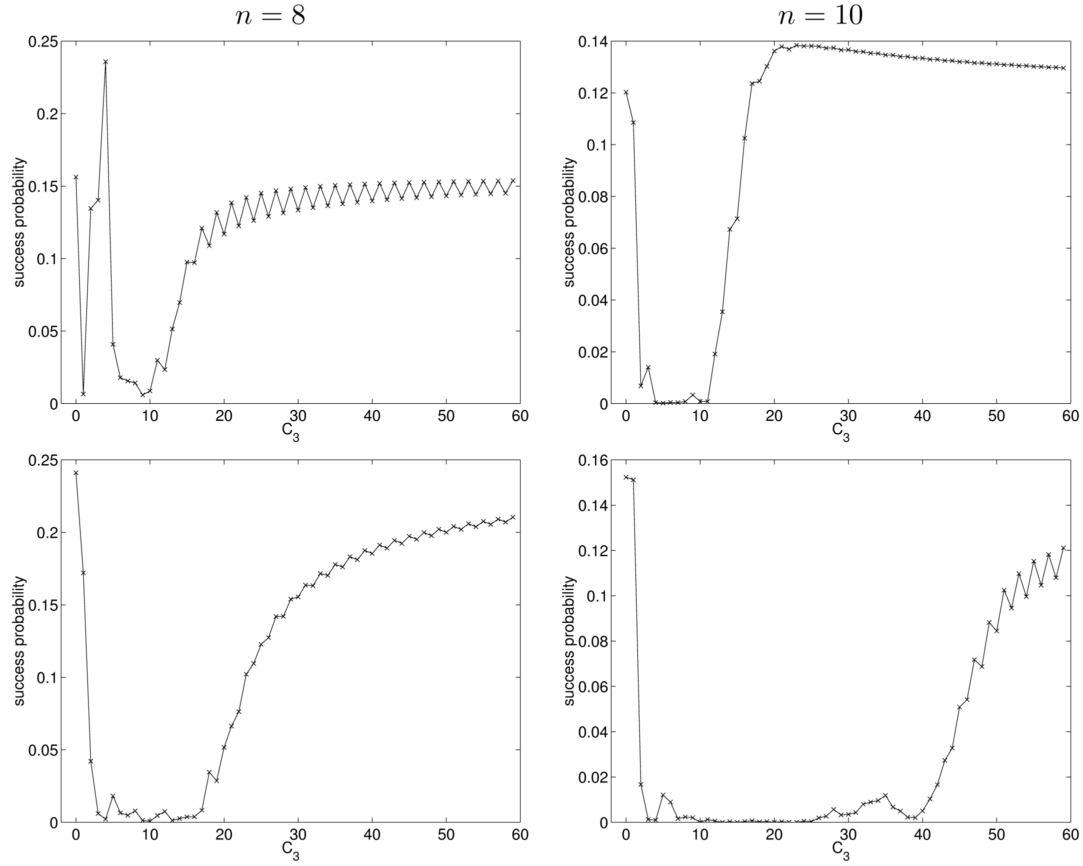

Figure 2-5: The success probability as a function of the frequency C_3 of the perturbation K_3 defined in (2.25). The data in each plot were obtained for a randomly generated instance of EC3 with randomly generated magnetic field directions. The data in the left column are for two instances with $n = 8$ bits, and the data in the right column are for two instances with $n = 10$ bits. For the top row, the run time is chosen so that the success probability is around 1/8 for $C_3 = 0$, and for the bottom row, the run time is twice as long. The leftmost points in each plot correspond to $C_3 = 0$, so the perturbation is absent for all t. C_3 takes integer values, so the lines are included only to guide the eye.

tends to oscillate between even and odd values of C_3, though whether even or odd values are favored varies from case to case. This occurs because the perturbation's time average vanishes for C_3 even, so that its integrated effect is weaker than for C_3 odd. Since a small perturbation might either help or hurt, the success probability is slightly enhanced for odd C_3 in some cases, and is slightly suppressed in other cases.

2.4.3 Discussion

We have conducted numerical simulations to investigate the fault tolerance of adiabatic quantum computation, and our results are consistent with the claim that this algorithm is robust against decoherence and certain kinds of random unitary perturbations. Thus, if a physical system could be engineered with interactions reasonably well described by a Hamiltonian that smoothly interpolates from an initial H_B to a final H_P corresponding to an interesting combinatorial search problem, and if the gap remains large throughout the interpolation, that system might be a powerful computational device.

Although we have viewed unitary perturbations as noise, the fact that they sometimes raise the success probability suggests a possible way to speed up the adiabatic algorithm. The algorithm finds the ground state of H_P by starting the system in the ground state of H_B. The quantum state evolves as the system Hamiltonian smoothly interpolates from H_B to H_P. However, there are many possible choices for H_B and many smooth paths from a given H_B to H_P. The choices (2.10) and (2.2) are convenient but arbitrary, so choosing an alternate route to H_P might speed up the algorithm. An example of this is seen in [158, 59], where it is shown that optimizing the time-dependent coefficients of H_B and H_P allows the adiabatic algorithm to achieve a square root speedup for the unstructured search problem. More generally, the interpolating Hamiltonian might involve terms that have nothing to do with H_B or H_P, but that increase Δ and therefore improve performance. For example, the perturbation K_2 sometimes increases the success probability, as seen in Figure 2-4. Rather than being thought of as a source of error, such a perturbation could be applied intentionally and might sometimes enhance the effectiveness of the adiabatic algorithm. This idea was applied in [78], where it was shown that a certain instance on which the adiabatic algorithm behaves badly can typically be solved efficiently by adding a random perturbation similar to K_2.

2.5 Search by measurement

In this section, we describe a measurement-based variant of adiabatic quantum computation. This approach is quite similar to standard adiabatic quantum computation, although the underlying mechanism is somewhat easier to understand than the usual adiabatic theorem. We also hope that exploring alternative means of computation can motivate new algorithmic ideas, as discussed in Section 0.3.

In the conventional circuit model of quantum computation described in Section 1.1, a quantum computation consists of a discrete sequence of unitary gates. Only at the end of the computation does one perform a measurement in the computational basis to read out the result. But another model of quantum computation allows measurement at intermediate stages. Indeed, recent work has shown that *measurement alone* is universal for quantum computation: one can efficiently implement a universal set of quantum gates using only measurements (and classical processing) [157, 151, 125]. Here, we describe an algorithm for solving combinatorial search problems that consists only of a sequence of measurements. Using a straightforward variant of the quantum Zeno effect (see for example [150, 7, 161]), we show how to keep the quantum computer in the ground state of a smoothly varying Hamiltonian $\tilde{H}(s)$. This process can be used to solve a computational problem by encoding the solution to the problem in the ground state of the final Hamiltonian just as in adiabatic quantum computation.

We begin in Section 2.5.1 by presenting the algorithm in detail and describing how measurement of $\tilde{H}(s)$ can be performed on a digital quantum computer. In Section 2.5.2, we estimate the running time of the algorithm in terms of spectral properties of $\tilde{H}(s)$, and in Section 2.5.3, we analyze the measurement process in detail. Then, in Section 2.5.4, we discuss how the algorithm performs on the unstructured search problem and show that by a suitable modification, Grover's quadratic speedup can be achieved by the measurement algorithm. Finally, in Section 2.5.6, we discuss the relationship between the measurement algorithm and quantum computation by adiabatic evolution.

2.5.1 The measurement algorithm

Our algorithm is conceptually similar to quantum computation by adiabatic evolution. Both algorithms operate by remaining in the ground state of a smoothly varying Hamiltonian $\tilde{H}(s)$ whose initial ground state is easy to construct and whose final ground state encodes the solution to the problem, as described in Section 2.2. However, whereas adiabatic quantum computation uses Schrödinger evolution under $\tilde{H}(s)$ to remain in the ground state, the present algorithm uses *only* measurement of $\tilde{H}(s)$.

To construct the measurement algorithm, we divide the interval $[0, 1]$ into M subintervals of width $\delta = 1/M$. So long as the interpolating Hamiltonian $\tilde{H}(s)$ is smoothly varying and δ is small, the ground state of $\tilde{H}(s)$ will be close to the ground state of $\tilde{H}(s + \delta)$. Thus, if the system is in the ground state of $\tilde{H}(s)$ and we measure $\tilde{H}(s + \delta)$, the post-measurement state is very likely to be the ground state of $\tilde{H}(s + \delta)$. If we begin in the ground state of $H_B = \tilde{H}(0)$ and successively measure $\tilde{H}(\delta), \tilde{H}(2\delta), \ldots, \tilde{H}((M - 1)\delta), \tilde{H}(1) = H_P$, then the final state will be the ground state of H_P with high probability, assuming δ is sufficiently small.

To complete our description of the measurement algorithm, we must explain how to measure the operator $\tilde{H}(s)$. The technique we use is motivated by von Neumann's description of the measurement process [150]. In this description, measurement is performed by coupling the system of interest to an ancillary system, which we call the *pointer*. Suppose that the pointer is a one-dimensional free particle and that the system-pointer interaction Hamiltonian is $\tilde{H}(s) \otimes p$, where p is the momentum of the particle. Furthermore, suppose that the mass of the particle is sufficiently large that we can neglect the kinetic term. Then the resulting evolution is

$$e^{-it\tilde{H}(s)\otimes p} = \sum_a \Big[|E_a(s)\rangle\langle E_a(s)| \otimes e^{-itE_a(s)p} \Big] \,, \qquad (2.27)$$

where $|E_a(s)\rangle$ are the eigenstates of $\tilde{H}(s)$ with eigenvalues $E_a(s)$. Suppose we prepare the pointer in the state $|x = 0\rangle$, a narrow wave packet centered at $x = 0$. Since the momentum operator generates translations in position, the above evolution performs the transformation

$$|E_a(s)\rangle \otimes |x = 0\rangle \to |E_a(s)\rangle \otimes |x = tE_a(s)\rangle \,. \qquad (2.28)$$

If we can measure the position of the pointer with sufficiently high precision that all relevant spacings $x_{ab} = t|E_a(s) - E_b(s)|$ can be resolved, then measurement of the position of the pointer—a fixed, easy-to-measure observable, independent of $\tilde{H}(s)$—effects a measurement of $\tilde{H}(s)$.

Von Neumann's measurement protocol makes use of a continuous variable, the position of the pointer. To turn it into an algorithm that can be implemented on a fully digital quantum computer, we can approximate the evolution (2.27) using r quantum bits to represent the pointer, as in the simulation of a particle in a potential discussed in Section 1.4 (and see also [184, 189]). The full Hilbert space is thus a tensor product of a 2^n-dimensional space for the system and a 2^r-dimensional space for the pointer. We let the computational basis of the pointer, with basis states $\{|z\rangle\}$, represent the basis of momentum eigenstates. The label z is an integer between 0 and $2^r - 1$, and the r bits of the binary representation of z specify the states of the r qubits. In this basis, the digital representation of p is simply given by

$$p = \sum_{j=1}^{r} 2^{-j} \frac{1-\sigma_z^{(j)}}{2} \,, \tag{2.29}$$

a sum of diagonal operators, each of which acts on only a single qubit. Here $\sigma_z^{(j)}$ is the Pauli z operator on the jth qubit. As we will discuss in the next section, we have chosen to normalize p so that

$$p|z\rangle = \frac{z}{2^r}|z\rangle \,, \tag{2.30}$$

which gives $\|p\| \approx 1$. If $\tilde{H}(s)$ is a sum of terms, each of which acts on at most k qubits, then $\tilde{H}(s) \otimes p$ is a sum of terms, each of which acts on at most $k+1$ qubits. If k is a fixed constant independent of the problem size n, such a Hamiltonian can be simulated efficiently on a universal quantum computer using Rules 1.1 and 1.4. Expanded in the momentum eigenbasis, the initial state of the pointer is

$$|x=0\rangle = \frac{1}{2^{r/2}} \sum_{z=0}^{2^r-1} |z\rangle \,. \tag{2.31}$$

The measurement is performed by evolving under $\tilde{H}(s) \otimes p$ for a total time τ. We discuss how to choose τ in the next section. After this evolution, the position of the simulated pointer could be measured by measuring the qubits that represent it in the x basis, i.e., the Fourier transform of the computational basis. However, note that our algorithm only makes use of the post-measurement state of the system, not of the measured value of $\tilde{H}(s)$. In other words, only the reduced density matrix of the system is relevant. Thus it is not actually necessary to perform a Fourier transform before measuring the pointer, or even to measure the pointer at all. When the system-pointer evolution is finished, one can either re-prepare the pointer in its initial state $|x=0\rangle$ or discard it and use a new pointer, and immediately begin the next measurement.

As an aside, note that the von Neumann measurement procedure described above is identical to the well-known phase estimation algorithm for measuring the eigenvalues of a unitary operator [118, 52], which can also be used to produce eigenvalues and eigenvectors of a Hamiltonian [3]. This connection has been noted previously in [189], and it has been pointed out that the measurement is a non-demolition measurement in [174]. In the phase estimation problem, we are given an eigenvector $|\psi\rangle$ of a unitary operator U and asked to determine its eigenvalue $e^{-i\phi}$. The algorithm uses two registers, one that initially stores $|\psi\rangle$ and one that will store an approximation of the phase ϕ. The first and last steps of the algorithm are Fourier transforms on the phase register. The intervening step is to perform the transformation

$$|\psi\rangle \otimes |z\rangle \to U^z|\psi\rangle \otimes |z\rangle \,, \tag{2.32}$$

where $|z\rangle$ is a computational basis state. If we take $|z\rangle$ to be a momentum eigenstate with eigenvalue z (i.e., if we choose a different normalization than in (2.30)) and let $U = e^{-iHt}$, this is exactly the transformation induced by $e^{-i(H\otimes p)t}$. Thus we see that the phase estimation algorithm for a unitary operator U is exactly von Neumann's prescription for measuring $i \ln U$.

2.5.2 Running time

The running time of the measurement algorithm is the product of M, the number of measurements, and τ, the time per measurement. Even if we assume perfect projective measurements, the algorithm is guaranteed to keep the computer in the ground state of $\tilde{H}(s)$ only in the limit $M \to \infty$, so that $\delta = 1/M \to 0$. Given a finite running time, the probability of finding the ground state of H_P with the last measurement will be less than 1. To understand the efficiency of the algorithm, we need to determine how long we must run as a function of n, the number of bits on which the function h is defined, so that the probability of success is not too small. In general, if the time required to achieve a success probability greater than some fixed constant (e.g., $\frac{1}{2}$) is poly(n), we say the algorithm is efficient, whereas if the running time grows exponentially, we say it is not.

To determine the running time of the algorithm, we consider the effect of the measurement process on the reduced density matrix of the system. Here, we simply motivate the main result; a detailed analysis is given in the following section.

Let $\rho^{(j)}$ denote the reduced density matrix of the system after the jth measurement; its matrix elements are

$$\rho_{ab}^{(j)} = \langle E_a(j\delta)|\rho^{(j)}|E_b(j\delta)\rangle \,. \tag{2.33}$$

The interaction with the digitized pointer effects the transformation

$$|E_a(s)\rangle \otimes |z\rangle \to e^{-iE_a(s)zt/2^r}|E_a(s)\rangle \otimes |z\rangle \,. \tag{2.34}$$

Starting with the pointer in the state (2.31), evolving according to (2.34), and tracing over the pointer, the quantum operation induced on the system is

$$\rho_{ab}^{(j+1)} = \kappa_{ab}^{(j+1)} \sum_{c,d} U_{ac}^{(j)} \rho_{cd}^{(j)} U_{bd}^{(j)*} \,, \tag{2.35}$$

where the unitary transformation relating the energy eigenbases at $s = j\delta$ and $s = (j+1)\delta$ is

$$U_{ab}^{(j)} = \langle E_a((j+1)\delta)|E_b(j\delta)\rangle \tag{2.36}$$

and

$$\kappa_{ab}^{(j)} = \frac{1}{2^r} \sum_{z=0}^{2^r-1} e^{i[E_b(j\delta)-E_a(j\delta)]zt/2^r} \,. \tag{2.37}$$

Summing this geometric series, we find

$$\left|\kappa_{ab}^{(j)}\right|^2 = |\kappa([E_b(j\delta) - E_a(j\delta)]t/2)|^2 \,, \tag{2.38}$$

where

$$|\kappa(x)|^2 = \frac{\sin^2 x}{4^r \sin^2(x/2^r)} \,. \tag{2.39}$$

This function is shown in Fig. 2-6 for the case $r = 4$. It has a sharp peak of unit height and width of order 1 at the origin, and identical peaks at integer multiples of $2^r\pi$.

If the above procedure were a perfect projective measurement, then we would have $\kappa_{ab} = 0$ whenever $E_a \neq E_b$. Assuming (temporarily) that this is the case, we find

$$\rho_{00}^{(j+1)} \geq \left|U_{00}^{(j)}\right|^2 \rho_{00}^{(j)} \tag{2.40}$$

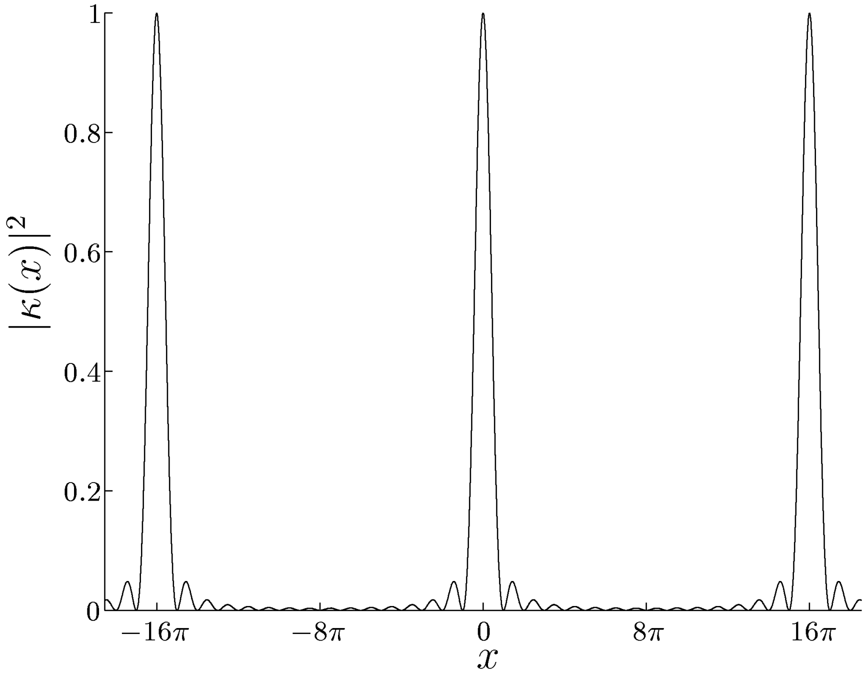

Figure 2-6: The function $|\kappa(x)|^2$ for $r = 4$.

with the initial condition $\rho_{00}^{(0)} = 1$ and $\rho_{ab}^{(0)} = 0$ otherwise. Perturbation theory gives

$$\left|U_{00}^{(j)}\right|^2 = 1 - \delta^2 \sum_{a\neq 0} \frac{|\langle E_a(s)|\frac{\mathrm{d}\tilde{H}}{\mathrm{d}s}|E_0(s)\rangle|^2}{(E_0(s) - E_a(s))^2}\Bigg|_{s=j\delta} + O(\delta^3) \tag{2.41}$$

$$\geq 1 - \frac{\Gamma(j\delta)^2\,\delta^2}{\Delta(j\delta)^2} + O(\delta^3)\,, \tag{2.42}$$

where $\Delta(s)$ and $\Gamma(s)$ are given in (2.5) and (2.6). In terms of the quantities Δ and Γ defined in (2.4), we find that according to (2.40), the probability of being in the ground state after the last measurement is at least

$$\rho_{00}^{(M)} \geq \left[1 - \frac{\Gamma^2}{M^2\Delta^2} + O(M^{-3})\right]^M \tag{2.43}$$

$$= \exp\left(-\frac{\Gamma^2}{M\Delta^2}\right) + O(M^{-2})\,. \tag{2.44}$$

The probability of success is close to 1 provided

$$M \gg \frac{\Gamma^2}{\Delta^2}\,. \tag{2.45}$$

When H_B and H_P are both sums of poly(n) terms, each of which acts nontrivially on at most a constant number of qubits, it is easy to choose an interpolation such as (2.2) so that Γ is only poly(n). Thus, as for the adiabatic algorithm, we are mainly interested in the behavior of Δ, the minimum gap between the ground and first excited states. We see that for the algorithm to be successful, the total number of measurements must be much larger than $1/\Delta^2$.

In fact, the simulated von Neumann procedure is not a perfect projective measurement. We must determine how long the system and pointer should interact so that the measurement is sufficiently good. In particular, the analysis of the following section shows that $|\kappa_{01}^{(j)}|^2$ should be bounded below 1 by a constant for all j. In other words, to sufficiently resolve the difference between the ground and first excited states, we must decrease the coherence between them by a fixed fraction per measurement. The width of the central peak in Figure 2-6 is of order 1, so it is straightforward to show that to have $|\kappa(x)|^2$ less than, say, 1/2, we must have $x \geq O(1)$. This places a lower bound on the system-pointer interaction time of

$$\tau \geq \frac{O(1)}{\Delta} \tag{2.46}$$

independent of r, the number of pointer qubits.

Putting these results together, we find that the measurement algorithm is successful if the total running time, $T = M\tau$, satisfies

$$T \gg \frac{\Gamma^2}{\Delta^3}\,. \tag{2.47}$$

This result can be compared to the corresponding expression for quantum computation by adiabatic evolution, (2.7).

The adiabatic and measurement algorithms have qualitatively similar behavior: if the gap is exponentially small, neither algorithm is efficient, whereas if the gap is only polynomially small, both algorithms are efficient. However, the measurement algorithm is slightly slower: whereas adiabatic evolution runs in a time that grows like $1/\Delta^2$, the measurement algorithm runs in a time that grows like $1/\Delta^3$. To see that this comparison is fair, recall that we have defined the momentum in (2.29) so that $\|p\| \approx 1$, which gives $\|\tilde{H}(s)\| \approx \|\tilde{H}(s) \otimes p\|$. Alternatively, we can compare the number m of few-qubit unitary gates needed to simulate the two algorithms on a conventional quantum computer. Using Rule 1.4 with the first-order expansion (1.7), we find $m = O(1/\Delta^4)$ for adiabatic evolution and $m = O(1/\Delta^6)$ for the measurement algorithm, in agreement with the previous comparison.

2.5.3 The measurement process

In this section, we analyze the measurement process in greater detail. First, we derive the bound (2.47) on the running time by demonstrating (2.45) and (2.46). We show rigorously that these bounds are sufficient as long as the gap is only polynomially small and the number of qubits used to represent the pointer is $r = O(\log n)$. Finally, we argue that $r = 1$ qubit should typically be sufficient.

Our goal is to find a bound on the final success probability of the measurement algorithm. We consider the effect of the measurements on the reduced density matrix of the system, which can be written as the block matrix

$$\rho = \begin{pmatrix} \mu & \nu^\dagger \\ \nu & \chi \end{pmatrix} \tag{2.48}$$

where $\mu = \rho_{00}$, $\nu_a = \rho_{a0}$ for $a \neq 0$, and $\chi_{ab} = \rho_{ab}$ for $a, b \neq 0$. Since $\operatorname{tr}\rho = 1$, $\mu = 1 - \operatorname{tr}\chi$. For ease of notation, we suppress j, the index of the iteration, except where necessary. The unitary transformation (2.36) may also be written as a block matrix. Define $\epsilon = \Gamma\delta/\Delta$.

Using perturbation theory and the unitarity constraint, we can write

$$U = \begin{pmatrix} u & -w^\dagger V + O(\epsilon^3) \\ w & V + O(\epsilon^2) \end{pmatrix} , \tag{2.49}$$

where $|u|^2 \geq 1 - \epsilon^2 + O(\epsilon^3)$, $\|w\|^2 \leq \epsilon^2 + O(\epsilon^3)$, and V is a unitary matrix. We let $\|\cdot\|$ denote the l_2 vector or matrix norm as appropriate. Furthermore, let

$$\kappa = \begin{pmatrix} 1 & k^\dagger \\ k & J \end{pmatrix} . \tag{2.50}$$

From (2.35), the effect of a single measurement may be written

$$\rho' = (U\rho U^\dagger) \circ \kappa \, , \tag{2.51}$$

where $\circ$ denotes the element-wise (Hadamard) product. If we assume $\|\nu\| = O(\epsilon)$, we find

$$\mu' = |u|^2\mu - w^\dagger V\nu - \nu^\dagger V^\dagger w + O(\epsilon^3) \tag{2.52}$$

$$\nu' = [V\nu + \mu w - V\chi V^\dagger w + O(\epsilon^2)] \circ k \, . \tag{2.53}$$

Now we use induction to show that our assumption always remains valid. Initially, $\nu^{(0)} = 0$. Using the triangle inequality in (2.53), we find

$$\left\|\nu'\right\| \leq [\|\nu\| + \epsilon + O(\epsilon^2)]\tilde{k} \, , \tag{2.54}$$

where

$$\tilde{k} = \max_{j,a} \left|k_a^{(j)}\right| . \tag{2.55}$$

So long as $\tilde{k} < 1$, we can sum a geometric series, extending the limits to go from 0 to ∞, to find

$$\left\|\nu^{(j)}\right\| \leq \frac{\epsilon}{1 - \tilde{k}} + O(\epsilon^2) \tag{2.56}$$

for all j. In other words, $\|\nu\| = O(\epsilon)$ so long as $\tilde{k}$ is bounded below 1 by a constant.

Finally, we put a bound on the final success probability $\mu^{(M)}$. Using the Cauchy-Schwartz inequality in (2.52) gives

$$\mu' \geq (1 - \epsilon^2)\mu - \frac{2\epsilon^2}{1 - \tilde{k}} + O(\epsilon^3) \, . \tag{2.57}$$

Iterating this bound M times with the initial condition $\mu^{(0)} = 1$, we find

$$\mu^{(M)} \geq 1 - \frac{\Gamma^2}{M\Delta^2}\left(1 + \frac{2}{1 - \tilde{k}}\right) + O(M\epsilon^3) \, . \tag{2.58}$$

If $\tilde{k}$ is bounded below 1 by a constant (independent of n), we find the condition (2.45).

The requirement on $\tilde{k}$ gives the bound (2.46) on the measurement time τ, and also gives a condition on the number of pointer qubits r. To see this, we must investigate properties of the function $|\kappa(x)|^2$ defined in (2.39) and shown in Fig. 2-6. It is straightforward to show that $|\kappa(x)|^2 \leq 1/2$ for $\pi/2 \leq x \leq \pi(2^r - 1/2)$. Thus, if we want $\tilde{k}$ to be bounded below 1 by a constant, we require

$$\pi/2 \le [E_a(s) - E_0(s)]t/2 \le \pi(2^r - 1/2) \tag{2.59}$$

for all s and for all $a \neq 0$. The left hand bound with $a = 1$ gives $t \ge \pi/\Delta$, which is (2.46). Requiring the right hand bound to hold for the largest energy difference gives the additional condition $2^r \gtrsim (E_{2^n-1} - E_0)/\Delta$. Since we only consider Hamiltonians $\tilde{H}(s)$ that are sums of poly(n) terms of bounded size, the largest possible energy difference must be bounded by a polynomial in n. If we further suppose that Δ is only polynomially small, this condition is satisfied by taking

$$r = O(\log n)\,. \tag{2.60}$$

Thus we see that the storage requirements for the pointer are rather modest.

However, for the purpose of the measurement algorithm, the pointer typically need not comprise even this many qubits. Since the goal of the measurement algorithm is to keep the system close to its ground state, it would be surprising if the energies of highly excited states were relevant. Suppose we take $r = 1$; then $|\kappa(x)|^2 = \cos^2(x/2)$. As before, (2.46) suffices to make $|\kappa_{01}|^2$ sufficiently small. However, we must also consider terms involving $|\kappa_{0a}|^2$ for $a > 1$. The algorithm will fail if the term $\mu w \circ k$ in (2.53) accumulates to be $O(1)$ over M iterations. This will only happen if, for $O(M)$ iterations, most of $\|w\|$ comes from components w_a with $(E_a - E_0)t$ close to an integer multiple of 2π. In such a special case, changing t will avoid the problem. An alternative strategy would be to choose t from a random distribution independently at each iteration.

2.5.4 The unstructured search problem

The unstructured search problem considered by Grover is to find a particular unknown n-bit string w using only queries of the form "is z the same as w?" [98]. In other words, one is trying to minimize a function

$$h_w(z) = \begin{cases} 0 & z = w \\ 1 & z \neq w. \end{cases} \tag{2.61}$$

Since there are 2^n possible values for w, the best possible classical algorithm uses $\Theta(2^n)$ queries. However, Grover's algorithm requires only $\Theta(2^{n/2})$ queries, providing a (provably optimal [19]) quadratic speedup. In Grover's algorithm, the winner is specified by an oracle U_w with

$$U_w|z\rangle = (-1)^{h_w(z)}|z\rangle\,. \tag{2.62}$$

This oracle is treated as a black box that one can use during the computation. One call to this black box is considered to be a single query of the oracle.

In addition to Grover's original algorithm, quadratic speedup can also be achieved in a time-independent Hamiltonian formulation [81] (and see also Chapter 4) or by adiabatic quantum computation [158, 59]. In either of these formulations, the winner is specified by an "oracle Hamiltonian"

$$H_w = 1 - |w\rangle\langle w| \tag{2.63}$$

whose ground state is $|w\rangle$ and that treats all orthogonal states (the non-winners) equivalently. One is provided with a black box that implements H_w, where w is unknown, and is asked to find w. Instead of counting queries, the efficiency of the algorithm is quantified in

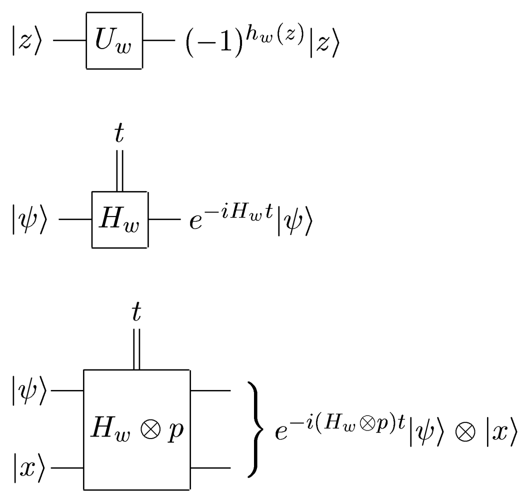

Figure 2-7: Oracles for the unstructured search problem. (a) Top: Grover's original oracle. (b) Center: An oracle that performs evolution according to H_w. The double line indicates a classical control parameter, the time for which the Hamiltonian is applied. (c) Bottom: An oracle that allows one to measure H_w.

terms of the total time for which one applies the oracle Hamiltonian.

Here, we show that by using a slightly different oracle Hamiltonian, we can achieve quadratic speedup using the measurement algorithm. We let the problem Hamiltonian be $H_P = H_w$ and we consider a one-parameter family of Hamiltonians $\tilde{H}(s)$ given by (2.2) for some H_B. Because we would like to *measure* this Hamiltonian, we will use a black box that evolves the system and a pointer according to $H_w \otimes p$, where p is the momentum of the pointer. This oracle is compared to the previous two in Fig. 2-7. By repeatedly alternating between applying this black box and evolving according to $H_B \otimes p$, each for small time, we can produce an overall evolution according to the Hamiltonian $[sH_B + (1-s)H_P] \otimes p$, and thus measure the operator $\tilde{H}(s)$ for any s. Note that either oracle Hamiltonian can be simulated efficiently using the unitary oracle (and vice versa), but we will find it convenient to use the continuous-time description.

Now consider the beginning Hamiltonian

$$H_B = \sum_{j=1}^{n} \frac{1 - \sigma_x^{(j)}}{2} \,, \tag{2.64}$$

where $\sigma_x^{(j)}$ is the Pauli x operator acting on the jth qubit. This beginning Hamiltonian is a sum of local terms, and has the easy-to-prepare ground state $|E_0(0)\rangle = 2^{-n/2} \sum_z |z\rangle$, the uniform superposition of all possible bit strings in the computational basis. If we consider the linear interpolation (2.2), then one can show [80] that the minimum gap occurs at

$$s^* = 1 - \frac{2}{n} + O(n^{-2}) \,, \tag{2.65}$$

where the gap takes the value

$$\Delta^* = \Delta(s^*) = 2^{1-n/2}[1 + O(n^{-1})] \,. \tag{2.66}$$

Naively applying (2.47) gives a running time $T = O(2^{3n/2})$, which is even worse than the classical algorithm.

However, since we know the value of s^* independent of w, we can improve on this approach by making fewer measurements. We observe that in the limit of large n, the ground state of $\tilde{H}(s)$ is close to the ground state $|E_0(0)\rangle$ of H_B for $s \lesssim s^*$ and is close to the ground state $|E_0(1)\rangle = |w\rangle$ of H_P for $s \gtrsim s^*$, switching rapidly from one state to the other in the vicinity of $s = s^*$. In Section 2.5.5, we show that up to terms of order $1/n$, the ground state $|\psi_+\rangle$ and the first excited state $|\psi_-\rangle$ of $\tilde{H}(s^*)$ are the equal superpositions

$$|\psi_\pm\rangle \simeq \frac{1}{\sqrt{2}}(|E_0(0)\rangle \pm |E_0(1)\rangle) \tag{2.67}$$

of the initial and final ground states (which are nearly orthogonal for large n). If we prepare the system in the state $|E_0(0)\rangle$ and make a perfect measurement of $\tilde{H}(s^*)$ followed by a perfect measurement of $\tilde{H}(1)$, we find the result w with probability $\frac{1}{2}$. The same effect can be achieved with an imperfect measurement, even if the pointer consists of just a single qubit. First consider the measurement of $\tilde{H}(s^*)$ in the state $|E_0(0)\rangle$. After the system and pointer have interacted for a time t according to (2.34) with $r = 1$, the reduced density matrix of the system in the $\{|\psi_+\rangle, |\psi_-\rangle\}$ basis is approximately

$$\frac{1}{2}\begin{pmatrix} 1 & e^{i\Delta^* t/4}\cos(\Delta^* t/4) \\ e^{-i\Delta^* t/4}\cos(\Delta^* t/4) & 1 \end{pmatrix}. \tag{2.68}$$

If we then measure $H(1)$ (i.e., measure in the computational basis), the probability of finding w is approximately

$$\frac{1}{2}\sin^2(\Delta^* t/4)\,. \tag{2.69}$$

To get an appreciable probability of finding w, we choose $t = \Theta(2^{n/2})$.

This approach is similar to the way one can achieve quadratic speedup with the adiabatic algorithm. Schrödinger time evolution governed by (2.2) does not yield quadratic speedup. However, because s^* is independent of w, we can change the Hamiltonian quickly when the gap is big and more slowly when the gap is small. Since the gap is only of size $\sim 2^{-n/2}$ for a region of width $\sim 2^{-n/2}$, the total oracle time with this modified schedule need only be $O(2^{n/2})$. This has been demonstrated explicitly by solving for the optimal schedule using a different beginning Hamiltonian that is not a sum of local terms [158, 59], but it also holds using the beginning Hamiltonian (2.64). We will return to this idea in Section 4.7.

Note that measuring $H(s^*)$ is not the only way to solve the unstructured search problem by measurement. More generally, we can start in some w-independent state, measure the operator

$$H' = H_w + K \tag{2.70}$$

where K is also independent of w, and then measure in the computational basis. For example, suppose we choose

$$K = 1 - |\psi\rangle\langle\psi|\,, \tag{2.71}$$

where $|\psi\rangle$ is a w-independent state with the property $|\langle w|\psi\rangle| \sim 2^{-n/2}$ for all w. (If we are only interested in the time for which we use the black box shown in Fig. 2-7(c), i.e., if we are only interested in the oracle query complexity, then we need not restrict K to be a sum of local terms.) In (2.71), the coefficient of -1 in front of $|\psi\rangle\langle\psi|$ has been fine-tuned so that $|\psi\rangle + |w\rangle$ is the ground state of $\tilde{H}$ (choosing the phase of $|w\rangle$ so that $\langle w|\psi\rangle$ is real and positive). If the initial state has a large overlap with $|\psi\rangle$, then the measurement procedure solves the unstructured search problem. However, the excited state $|\psi\rangle - |w\rangle$ is also an eigenstate of H', with an energy higher by of order $2^{-n/2}$. Thus the time to perform the measurement must be $\Omega(2^{n/2})$.

The measurement procedures described above saturate the well-known lower bound on the time required to solve the unstructured search problem. Using an oracle like the one shown in Fig. 2-7(a), Bennett, Bernstein, Brassard, and Vazirani showed that the unstructured search problem cannot be solved on a quantum computer using fewer than of order $2^{n/2}$ oracle queries [19]. By a straightforward modification of their argument, an equivalent result applies using the oracle shown in Fig. 2-7(c). Thus every possible H' as in (2.70) that can be measured to find w must have a gap between the energies of the relevant eigenstates of order $2^{-n/2}$ or smaller.

2.5.5 Eigenstates in the unstructured search problem

Here, we show that the ground state of $\tilde{H}(s^*)$ for the Grover problem is close to (2.67). Our analysis follows Section 4.2 of [80].

Since the problem is invariant under the choice of w, we consider the case $w = 0$ without loss of generality. In this case, the problem can be analyzed in terms of the total spin operators

$$S_a = \frac{1}{2}\sum_{j=1}^{n} \sigma_a^{(j)} \,, \tag{2.72}$$

where $a = x, y, z$ and $\sigma_a^{(j)}$ is the Pauli a operator acting on the jth qubit. The Hamiltonian commutes with $\vec{S}^2 = S_x^2 + S_y^2 + S_z^2$, and the initial state has $\vec{S}^2 = \frac{n}{2}(\frac{n}{2}+1)$, so we can restrict our attention to the $(n+1)$-dimensional subspace of states with this value of $\vec{S}^2$. In this subspace, the eigenstates of the total spin operators satisfy

$$S_a|m_a = m\rangle = m|m_a = m\rangle \tag{2.73}$$

for $m = -\frac{n}{2}, -\frac{n}{2}+1, \ldots, \frac{n}{2}$. Written in terms of the total spin operators and eigenstates, the Hamiltonian is

$$\tilde{H}(s) = (1-s)\left(\frac{n}{2} - S_x\right) + s\left(1 - \left|m_z = \frac{n}{2}\right\rangle\left\langle m_z = \frac{n}{2}\right|\right) \,. \tag{2.74}$$

The initial and final ground states are given by $|E_0(0)\rangle = |m_x = \frac{n}{2}\rangle$ and $|E_0(1)\rangle = |m_z = \frac{n}{2}\rangle$, respectively.

Projecting the equation $\tilde{H}(s)|\psi\rangle = E|\psi\rangle$ onto the eigenbasis of S_x, we find

$$\left\langle m_x = \frac{n}{2} - r\middle|\psi\right\rangle = \frac{s}{1-s}\frac{\sqrt{P_r}}{r-\lambda}\left\langle m_z = \frac{n}{2}\middle|\psi\right\rangle \,, \tag{2.75}$$

where we have defined $\lambda = (E-s)/(1-s)$ and $P_r = 2^{-n}\binom{n}{r}$. Now focus on the ground state $|\psi_+\rangle$ and the first excited state $|\psi_-\rangle$ of $\tilde{H}(s^*)$. By equation (4.39) of [80], these states have $\lambda_\pm = \mp\frac{n}{2}2^{-n/2}(1+O(1/n))$. Putting $r = 0$ in (2.75) and taking $s = s^*$ from (2.65), we find

$$\left\langle m_x = \frac{n}{2}\middle|\psi_\pm\right\rangle = \pm\left\langle m_z = \frac{n}{2}\middle|\psi_\pm\right\rangle(1+O(1/n)) \,. \tag{2.76}$$

For $r \neq 0$, we have

$$\left\langle m_x = \frac{n}{2} - r \middle| \psi_\pm \right\rangle = \frac{n}{2} \frac{\sqrt{P_r}}{r} \left\langle m_z = \frac{n}{2} \middle| \psi_\pm \right\rangle (1 + O(1/n)) \,. \tag{2.77}$$

Requiring that $|\psi_\pm\rangle$ be normalized, we find

$$1 = \sum_{r=0}^{n} \left| \left\langle m_x = \frac{n}{2} - r \middle| \psi_\pm \right\rangle \right|^2 \tag{2.78}$$

$$= \left| \left\langle m_z = \frac{n}{2} \middle| \psi_\pm \right\rangle \right|^2 \left(1 + \frac{n^2}{4} \sum_{r=1}^{n} \frac{P_r}{r^2} \right) (1 + O(1/n)) \tag{2.79}$$

$$= \left| \left\langle m_z = \frac{n}{2} \middle| \psi_\pm \right\rangle \right|^2 (2 + O(1/n)) \,, \tag{2.80}$$

which implies $|\langle m_z = \frac{n}{2} | \psi_\pm \rangle|^2 = \frac{1}{2} + O(1/n)$. From (2.76), we also have $|\langle m_x = \frac{n}{2} | \psi_\pm \rangle|^2 = \frac{1}{2} + O(1/n)$. Thus we find

$$|\psi_\pm\rangle \simeq \frac{1}{\sqrt{2}} \left(\left| m_x = \frac{n}{2} \right\rangle \pm \left| m_z = \frac{n}{2} \right\rangle \right) \tag{2.81}$$

up to terms of order $1/n$, which is (2.67).

2.5.6 Discussion

We have described a way to solve combinatorial search problems on a quantum computer using only a sequence of measurements to keep the computer near the ground state of a smoothly varying Hamiltonian. The basic principle of this algorithm is similar to quantum computation by adiabatic evolution, and the running times of the two methods are closely related. Because of this close connection, many results on adiabatic quantum computation can be directly applied to the measurement algorithm. We have also shown that the measurement algorithm can achieve quadratic speedup for the unstructured search problem using knowledge of the place where the gap is smallest, as in adiabatic quantum computation.

Although it does not provide a computational advantage over quantum computation by adiabatic evolution, the measurement algorithm is an alternative way to solve general combinatorial search problems on a quantum computer. The algorithm can be simply understood in terms of measurements of a set of operators, without reference to unitary time evolution. Nevertheless, we have seen that to understand the running time of the algorithm, it is important to understand the dynamical process by which these measurements are realized.

The robustness of the adiabatic algorithm to unitary control errors is shared to some extent by the measurement algorithm: again, the particular path from H_B to H_P is unimportant as long as the initial and final Hamiltonians are correct, the path is smoothly varying, and the minimum gap along the path is not too small. However, we would not necessarily expect the measurement algorithm to share the adiabatic algorithm's robustness to thermal transitions out of the ground state, since the Hamiltonian of the quantum computer during the measurement procedure is not simply $\tilde{H}(s)$.

Chapter 3

Quantum walk

3.1 Introduction

The concept of random walk has a long history in mathematics and physics. Furthermore, random walks on graphs, also known as Markov chains, are widely used in classical computer science—a few examples include space-bounded computation [9], constraint satisfaction problems [154, 160], graph coloring [141], approximate counting [169], and probability amplification [53, 108]. Thus, in the search for new quantum algorithms, there has been considerable interest in studying a quantum version of random walk.

Perhaps the best known type of random walk is the simple discrete-time random walk. In this process, one begins at some vertex of a graph (possibly selected at random) and at each time step has equal probability of moving to each adjacent vertex. These stochastic dynamics can be viewed as the deterministic evolution of a probability distribution over the vertices. The idea of a quantum walk is to replace the probability distribution with a quantum state, but to retain the notion of local evolution on the graph.

As we discuss in Section 3.4, there is a certain technical difficulty in defining a discrete-time quantum walk. Therefore, we will find it more natural to define a quantum walk in continuous time, in analogy to a continuous-time classical random walk (which we will do in Section 3.2). However, the spirit of the idea is the same: by replacing a probability distribution with complex amplitudes, we can create a walk that exhibits constructive and destructive interference, resulting in radically different behavior. We will give a few simple examples of quantum walks in Section 3.3, and we will present original results on how quantum walk can be used to achieve algorithmic speedup in Chapters 4 and 5. Some related results by other researchers are reviewed in Section 3.5.

To discuss quantum walk, we need some standard graph-theoretic notation. We write $a \in G$ to denote that the vertex a is in the graph G and $ab \in G$ to denote that the edge joining vertices a and b is in the graph. We let $\deg(a)$ denote the degree of vertex a, i.e., the number of edges incident on that vertex. Given an undirected graph G with N vertices and no self-loops, we define the $N \times N$ *adjacency matrix*

$$A_{ab} = \begin{cases} 1 & ab \in G \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

which describes the connectivity of G. In terms of this matrix, we also define the *Laplacian* $L = A - D$, where D is the diagonal matrix with $D_{aa} = \deg(a)$. The matrix L is called the

Laplacian because it is a discrete approximation to the continuum operator ∇^2 whenever G can be viewed as a discretization of a continuous manifold.

3.2 From random walk to quantum walk

A continuous-time quantum walk on a graph can be defined in direct analogy to a corresponding continuous-time classical random walk, as proposed by Farhi and Gutmann [82]. A continuous-time classical random walk is a Markov process on a graph. In this process, there is a fixed probability per unit time γ of moving to an adjacent vertex. In other words, from any vertex, the probability of jumping to any connected vertex in a time ϵ is $\gamma\epsilon$ (in the limit $\epsilon \to 0$). If the graph has N vertices, this classical random walk can be described by the $N \times N$ infinitesimal generator matrix $K = -\gamma L$, where L is the Laplacian of the graph. If $p_a(t)$ is the probability of being at vertex a at time t, then

$$\frac{\mathrm{d}p_a(t)}{\mathrm{d}t} = \sum_b K_{ab}\, p_b(t)\,. \tag{3.2}$$

Note that because the columns of K sum to zero, we have $\frac{\mathrm{d}}{\mathrm{d}t}\sum_a p_a(t) = 0$, so that an initially normalized distribution remains normalized, i.e., $\sum_a p_a(t) = 1$ for all t.

The continuous-time quantum walk on G takes place in an N-dimensional Hilbert space spanned by states $|a\rangle$, where a is a vertex in G. In terms of these basis states, we can write a general state $|\psi(t)\rangle$ in terms of the N complex amplitudes $q_a(t) = \langle a|\psi(t)\rangle$. If the Hamiltonian is H, then the dynamics of the system are determined by the Schrödinger equation (0.1), which we may rewrite as

$$i\frac{\mathrm{d}q_a(t)}{\mathrm{d}t} = \sum_b \langle a|H|b\rangle\, q_b(t)\,. \tag{3.3}$$

Note the similarity between (3.2) and (3.3). A natural quantum analog of the continuous-time classical random walk described above is given by the Hamiltonian with matrix elements

$$\langle a|H|b\rangle = K_{ab}\,, \tag{3.4}$$

i.e., $H = -\gamma L$.

Although this choice of the Hamiltonian is most closely analogous to the continuous-time classical random walk, it is not unique. Whereas (3.2) requires $\sum_a K_{ab}$ to conserve $\sum_a p_a(t)$, (3.3) requires $H = H^\dagger$ to conserve $\sum_a |q_a(t)|^2$, i.e., to be unitary. Any choice consistent with this requirement is legitimate as long as it retains the locality of the graph. For example, we can set the diagonal to zero and choose $H = \gamma A$. For regular graphs (i.e., graphs for which $\deg(a)$ is independent of a), these two choices differ by a multiple of the identity matrix, so they give rise to the same quantum dynamics. However, for non-regular graphs, the two choices will give different results. In the following chapters, we will have occasion to use both definitions depending on the situation.

In fact, one could reasonably view *any* time-independent Hamiltonian dynamics in an N-dimensional Hilbert space as a quantum walk on an N-vertex graph G provided $\langle a|H|b\rangle \neq 0$ if and only if $ab \in G$. In other words, we can associate arbitrary complex weights with the directed edges of the graph, so long as the value associated to the edge ab is the complex conjugate of the value associated to the edge ba. However, thinking of Hamiltonian evolution

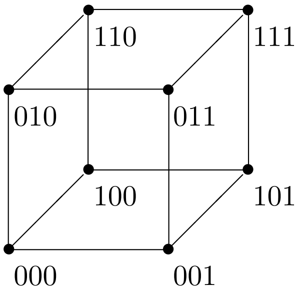

Figure 3-1: Three-dimensional hypercube, with vertices labeled by bit strings.

as a quantum walk on a graph can be helpful, since the underlying graph can provide a visual way of engineering quantum interference.

To be useful for algorithmic applications, the quantum walk must be efficiently implementable by a quantum computer. If the graph is sufficiently sparse, then its quantum walk can be implemented efficiently using Rule 1.7 from Chapter 1. For example, we will use this implementation in our demonstration of exponential algorithmic speedup by quantum walk in Chapter 5.

3.3 Examples

In this section, we present two simple examples of quantum walks on graphs, the n-dimensional hypercube and the infinite line. These examples serve to illustrate some basic features of quantum walk, including its relationship to classical random walk. They also introduce some concepts that will be useful in later chapters.

3.3.1 Quantum walk on a hypercube

The n-dimensional hypercube is a graph with $N = 2^n$ vertices. The familiar case $n = 3$ (the cube) is shown in Figure 3-1. The vertices of the hypercube can be labeled by n-bit strings, where two vertices are connected if they differ in exactly one bit. Thus the adjacency matrix can be written

$$A = \sum_{j=1}^{n} \sigma_x^{(j)} \tag{3.5}$$

where $\sigma_x^{(j)}$ is the Pauli x operator for the jth qubit. Since the hypercube is a regular graph, it does not matter if we use the adjacency matrix or the Laplacian as the generator of the quantum walk, so we use the adjacency matrix for simplicity.

Because A is a sum of commuting single-qubit operators, it is trivial to exponentiate. At time t (in units where $\gamma = 1$), the evolution of the quantum walk is described by the unitary operator

$$U(t) = e^{-iAt} \tag{3.6}$$

$$= \prod_{j=1}^{n} e^{-i\sigma_x^{(j)} t} \tag{3.7}$$

$$= \prod_{j=1}^{n} \left(\cos t - i \sigma_x^{(j)} \sin t \right) . \tag{3.8}$$

Note that after time $t = \pi/2$, we have $U(\pi/2) = \prod_{j=1}^{n} \sigma_x^{(j)}$ (up to an overall phase), i.e., the quantum walk evolves from any given state to its antipodal state. This can be contrasted with a classical random walk, which rapidly approaches a uniform distribution over the vertices, so that the probability of reaching the antipodal vertex is always exponentially small—a straightforward calculation shows that it is $[(1 - e^{-2t})/2]^n$. This is perhaps the simplest example of radically different behavior between a quantum walk and its classical counterpart.

3.3.2 Quantum walk on a line

Now consider the quantum walk on an infinite, translationally invariant line. In this graph, there is a vertex for every integer j, and the vertex j is connected to the two vertices $j + 1$ and $j - 1$, so the nonzero matrix elements of the adjacency matrix are

$$\langle j|A|j \pm 1\rangle = 1 \, . \tag{3.9}$$

Again, the graph is regular, so we choose $H = A$ for simplicity. The eigenstates of this Hamiltonian are the momentum eigenstates $|p\rangle$ with components

$$\langle j|p\rangle = \frac{1}{\sqrt{2\pi}} e^{ipj} \, , \quad -\pi \le p \le \pi \tag{3.10}$$

having energies

$$E(p) = 2 \cos p \, . \tag{3.11}$$

Using these expressions, it is straightforward to calculate the propagator, or Green's function, to go from j to k in time t:

$$\begin{aligned} G(j, k, t) &= \langle k|e^{-iHt}|j\rangle & (3.12) \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{ip(k-j) - 2it \cos p} \, \mathrm{d}p & (3.13) \\ &= (-i)^{k-j} J_{k-j}(2t) & (3.14) \end{aligned}$$

where $J_\nu(\cdot)$ is a Bessel function of order ν. By the well-known properties of the Bessel function, this shows that a state initially localized at a single vertex evolves as a left-moving and a right-moving wave packet, each propagating with speed 2. To see this, note that the Bessel function has the following asymptotic expansions for $\nu \gg 1$:

$$\begin{aligned} J_\nu(\nu \operatorname{sech} \xi) &\sim \frac{e^{-\nu(\xi - \tanh \xi)}}{\sqrt{2\pi\nu \tanh \xi}} & (3.15) \\ J_\nu(\nu + \xi \nu^{1/3}) &= (2/\nu)^{1/3} \operatorname{Ai}(-2^{1/3}\xi) + O(\nu^{-1}) & (3.16) \\ J_\nu(\nu \sec \xi) &= \sqrt{\frac{2}{\pi \nu \tan \xi}} \left\{ \cos[\tfrac{\pi}{4} - \nu(\xi - \tan \xi)] + O(\nu^{-1}) \right\} , \quad 0 < \xi < \frac{\pi}{2} & (3.17) \end{aligned}$$

where $\operatorname{Ai}(\cdot)$ is an Airy function [2]. These three relations show that for $|k-j| \gg 1$, $G(j, k, t)$ is exponentially small in $|k-j|$ for $t < 0.99 \cdot |k-j|/2$, of order $|k-j|^{-1/3}$ for t near $|k-j|/2$, and of order $|k-j|^{-1/2}$ for $t > 1.01 \cdot |k-j|/2$.

This result can be contrasted with the corresponding classical random walk. In time t, the continuous-time classical random walk on the line only moves a distance proportional to $\sqrt{t}$. This can be seen by analytically continuing the quantum walk (using the Laplacian rather than the adjacency matrix as the generator of the walk, since only the Laplacian gives rise to a probability-conserving classical Markov process). Including the appropriate phase factor and then putting $it \to t$ in (3.14), we find

$$\begin{aligned} G_c(j,k,t) &= [e^{-Lt}]_{kj} && (3.18) \\ &= e^{-2t} I_{k-j}(2t) && (3.19) \end{aligned}$$

for the probability of moving from j to k in time t, where $I_\nu(\cdot)$ is the modified Bessel function of order ν. This shows that the probability to move a distance x in time t is $e^{-2t}I_x(2t)$, which for large t is approximately $\frac{1}{\sqrt{4\pi t}}\exp(-x^2/4t)$, a Gaussian of width $\sqrt{2t}$, as can be shown using asymptotic properties of the modified Bessel function.

3.4 Discrete-time quantum walk

The model of quantum walk presented above is based on continuous time evolution. This definition is natural from the point of view of physics, but it also sidesteps a fundamental problem with defining a quantum analog of a discrete-time classical random walk. The essence of the difficulty can be seen by considering the infinite line. Suppose we attempt to define a translationally invariant rule that transforms a given vertex to a superposition of its neighbors. One is tempted to propose

$$|j\rangle \overset{?}{\to} \frac{1}{\sqrt{2}}(|j+1\rangle + |j-1\rangle)\,, \tag{3.20}$$

but this rule is clearly not unitary, since (for example) the orthogonal states $|1\rangle$ and $|-1\rangle$ evolve to non-orthogonal states. In fact, under very mild assumptions, Meyers proved that no discrete-time quantum walk on the vertices of the line exists [143]. This result can also be generalized to a lattice of any dimension [144], and furthermore, it can be shown that only graphs with special properties can possibly support local unitary dynamics, even without homogeneity restrictions [162].

However, despite this difficulty, it is possible to define a discrete-time quantum walk on a general graph if one is willing to use a state space other than the vertices of the graph.[1] In particular, one can define a discrete-time quantum walk on the directed edges of a graph, where each step maps an outgoing directed edge of a given vertex to a superposition of the outgoing directed edges of its neighbors [182, 11, 4] (and see also [104, 172] for related formulations). This model has been studied in some detail, and is generally quite similar in behavior to the continuous-time quantum walk. Some examples are mentioned in the following section, where we summarize known results on quantum walks.

[1] We are reminded of John Bell's remark that "...what is proved by impossibility proofs is lack of imagination" [17]. For another example of this phenomenon, see Section 4.5.

3.5 Discussion

Quantum walks on graphs have recently been widely studied. To conclude this chapter, we review some of the main results on both continuous- and discrete-time quantum walk, some of which will be discussed in the following two chapters. In our view, these results have only scratched the surface of the possible phenomenology and algorithmic implications of quantum walk.

The quantum walk on the hypercube, in both continuous and discrete time, was investigated by Moore and Russell [148]. They were primarily interested in determining how quickly the walk starting from a particular vertex spreads out over the entire graph. The discrete-time quantum walk on the line has been analyzed by many authors, including Ambainis, Bach, Nayak, Vishwanath, and Watrous [11]. Its behavior is qualitatively similar to the continuous-time quantum walk on the line described in Section 3.3.2.

Gerhardt and Watrous have analyzed the continuous-time quantum walk on certain Cayley graphs of the symmetric group using the tools of representation theory [90, 91]. Their results show that this quantum walk is very different from its classical counterpart. Unfortunately, it appears that, rather than reaching interesting faraway locations, the quantum walk tends to remain close to its starting point, a phenomenon that does not seem amenable to algorithmic applications.

Several examples are known of graphs for which the quantum walk moves rapidly between two particular vertices, but where the corresponding classical random walk does not. The first example of this kind was a continuous-time quantum walk on a certain tree presented by Farhi and Gutmann [82]. A much simpler example is the continuous-time walk on the hypercube, as discussed in Section 3.3.1. Kempe showed that the same phenomenon occurs in the discrete-time quantum walk [116]. Other examples, from [44, 40], are discussed in Chapter 5.

Regarding algorithmic applications, there have been three main developments. First, a quantum walk has been used to construct an oracular problem that can be solved exponentially faster by a quantum walk than by any classical algorithm [40]. This result is described in detail in Chapter 5. Second, quantum walks have been used to locally search a graph for a marked vertex quadratically faster than is possible classically [81, 164, 46, 12, 47], providing spatially local versions of Grover's algorithm [98]. The continuous-time algorithms of this sort are described in Chapter 4. Finally, Ambainis has recently used a discrete-time quantum walk to give an optimal $O(N^{2/3})$-query algorithm for the element distinctness problem [10] (and see also [42, 172]). This algorithm works in a very similar way to the spatial search algorithms, but walks on a different graph. The algorithm can be applied to a wide variety of problems sharing the feature that a solution can be identified by a small certificate [42]. For example, it has yielded improved quantum algorithms for finding substructures in graphs [136, 42].

Chapter 4

Spatial search by quantum walk

4.1 Introduction

Recall that Grover's quantum search algorithm [98] is one of the main applications of quantum computation. Given a black box function $f(x) : \{1, \dots, N\} \to \{0, 1\}$ satisfying

$$f(x) = \begin{cases} 0 & x \neq w \\ 1 & x = w \,, \end{cases} \tag{4.1}$$

Grover's algorithm can find the value of w using of order $\sqrt{N}$ queries, which is optimal [19]. On the other hand, no classical algorithm can do better than exhaustive search, which takes of order N queries. Grover's algorithm can be used to speed up brute force combinatorial search. It can also be used as a subroutine in a variety of other quantum algorithms.

Grover's algorithm is sometimes described as a way to search an unsorted database of N items in time $O(\sqrt{N})$. But the algorithm as originally proposed is not designed to search a physical database. Suppose we had N items stored in a d-dimensional physical space, and that these items could be explored in superposition by a quantum computer making local moves (a "quantum robot" in the terminology of Benioff [18]). Naively, it would seem that each step of the Grover algorithm should take time of order $N^{1/d}$, since this is the time required to cross the database. Performing $\sqrt{N}$ iterations, we find that the search takes time of order $N^{\frac{1}{2}+\frac{1}{d}}$, so no speedup is achieved in $d = 2$, and full speedup is achieved only in the limit of large d.

However, it is possible to do better than this naive approach suggests. In [1], Aaronson and Ambainis present a model of query complexity on graphs. Within this model, they give a recursive algorithm for the search problem that achieves full $\sqrt{N}$ speedup for a $d \geq 3$ dimensional lattice, and runs in time $\sqrt{N} \log^2 N$ in $d = 2$. (A straightforward argument shows that no algorithm can get speedup in $d = 1$, since it takes time $O(N)$ just to cross the database.)

In this chapter, we approach the spatial search problem using quantum walks. Quantum walks provide a natural framework for the spatial search problem because the graph can be used to model the locality of the database. We present a simple quantum walk search algorithm that can be applied to any graph. Our algorithm can be implemented within the model of [1], but is actually much simpler because it uses no auxiliary storage space. For the case of the complete graph, the resulting algorithm is simply the continuous-time search algorithm of Farhi and Gutmann [81]. On the hypercube, previous results can be used to show that the algorithm also provides quadratic speedup [80, 41] (and see Section 2.5.5). However, in both of these cases, the graph is highly connected. Here, we consider the case of a d-dimensional cubic periodic lattice, where d is fixed independent of N. We find full $\sqrt{N}$ speedup in $d > 4$ and running time $O(\sqrt{N}\log^{3/2} N)$ in $d = 4$. In $d < 4$, we find no speedup, so this simple continuous-time quantum walk algorithm is never faster than the Aaronson-Ambainis algorithm.

The spatial search problem can also be approached using a discrete-time quantum walk. This type of walk has been used to construct a fast search algorithm on the hypercube [164], and recently, on a d-dimensional lattice with $d \geq 2$ [12]. As discussed in Section 3.4, a discrete-time quantum walk cannot be defined on a state space consisting only of the vertices of a graph, so these discrete-time walk algorithms necessarily use additional degrees of freedom. In fact, by introducing additional degrees of freedom into the continuous-time quantum walk, we find a search algorithm that works better in lower dimensions. Although the continuous-time quantum walk can be defined without additional memory, we find that the additional degrees of freedom can improve the algorithm's performance. This modified algorithm draws from Dirac's quantum mechanical formulation of a relativistic particle, and helps to illuminate the similarities and differences between continuous- and discrete-time quantum walks.

This chapter is organized as follows. In Section 4.2 we discuss how the continuous-time quantum walk can be used to approach the search problem. In Section 4.3 we review the results in the high-dimensional cases (the complete graph and the hypercube), casting them in the language of continuous-time quantum walks. In Section 4.4 we present the results for finite dimensional lattices, and in Section 4.5, we show how these results can be improved in low dimensions by introducing a Dirac spin. In Section 4.6, we explain how the continuous-time algorithm can be simulated in linear time in the local unitary model of [1]. Finally, in Section 4.7, we conclude with a discussion of the results and some related algorithms.

4.2 Quantum walk algorithm

To approach the search problem with a quantum walk, we need to modify the usual quantum walk Hamiltonian so that the vertex w is special. Following [81], we introduce the *oracle Hamiltonian*[1]

$$H_w = -|w\rangle\langle w| \tag{4.2}$$

which has energy zero for all states except $|w\rangle$, which is the ground state, with energy -1. Solving the search problem is equivalent to finding the ground state of this Hamiltonian. Here, we assume that this Hamiltonian is given, and we want to use it for as little time as possible to find the value of w. As we noted before, this Hamiltonian can be simulated in the circuit model using the standard Grover oracle

$$U_w|j\rangle = (-1)^{\delta_{jw}}|j\rangle\,. \tag{4.3}$$

We will show how this can be done without compromising the speedup in Section 4.6. For now, we focus on the continuous-time description.

To construct an algorithm with the locality of a particular graph G, we consider the

[1] More precisely, we should use $H_w = -\omega|w\rangle\langle w|$ where ω is a fixed parameter with units of inverse time. However, we choose units in which $\omega = 1$. In these units, γ is a dimensionless parameter.

time-independent Hamiltonian

$$H = -\gamma L + H_w = -\gamma L - |w\rangle\langle w| \tag{4.4}$$

where L is the Laplacian of G. We begin in a uniform superposition over all vertices of the graph,

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle \,, \tag{4.5}$$

and run the quantum walk for time T. We then measure in the vertex basis. Our objective is to choose the parameter γ so that the success probability $|\langle w|\psi(T)\rangle|^2$ is as close to 1 as possible for as small a T as possible. Note that the coefficient of H_w is held fixed at 1 to make the problem fair (e.g., so that evolution for time T could be simulated with $O(T)$ queries of the standard Grover oracle (4.3)).

One might ask why we should expect this algorithm to give a substantial success probability for some values of γ, T. We motivate this possibility in terms of the spectrum of H. Note that regardless of the graph, $|s\rangle$ is the ground state of the Laplacian, with $L|s\rangle = 0$. As $\gamma \to \infty$, the contribution of H_w to H is negligible, so the ground state of H is close to $|s\rangle$. On the other hand, as $\gamma \to 0$, the contribution of L to H disappears, so the ground state of H is close to $|w\rangle$. Furthermore, since $|s\rangle$ is nearly orthogonal to $|w\rangle$, degenerate perturbation theory shows that the first excited state of H will be close to $|s\rangle$ as $\gamma \to 0$ for large N. We might expect that over some intermediate range of γ, the ground state will switch from $|w\rangle$ to $|s\rangle$, and could have substantial overlap on both for a certain range of γ. If the first excited state also has substantial overlap on both $|w\rangle$ and $|s\rangle$ at such values of γ, then the Hamiltonian will drive transitions between the two states, and thus will rotate the state from $|s\rangle$ to a state with substantial overlap with $|w\rangle$ in a time of order $1/(E_1 - E_0)$, where E_0 is the ground state energy and E_1 is the first excited state energy.

Indeed, we will see that this is a good description of the algorithm if the dimension of the graph is sufficiently high. The simplest example is the complete graph [81] which can be thought of roughly as having dimension proportional to N. A similar picture holds for the $(\log N)$-dimensional hypercube. When we consider a d-dimensional lattice with d independent of N, we will see that the state $|s\rangle$ still switches from ground state to first excited state at some critical value of γ. However, the $|w\rangle$ state does not have substantial overlap on the ground and first excited states unless $d > 4$, so the algorithm will not work for $d < 4$ (and $d = 4$ will be a marginal case).

4.3 High dimensions

In this section, we describe the quantum walk algorithm on "high dimensional" graphs, namely the complete graph and the hypercube. These cases have been analyzed in previous works [81, 80, 41] (and see Section 2.5.5). Here, we reinterpret them as quantum walk algorithms, which provides motivation for the case of a lattice in d spatial dimensions.

4.3.1 Complete graph

Letting L be the Laplacian of the complete graph, we find exactly the continuous-time search algorithm proposed in [81]. Adding a multiple of the identity matrix to the Laplacian gives

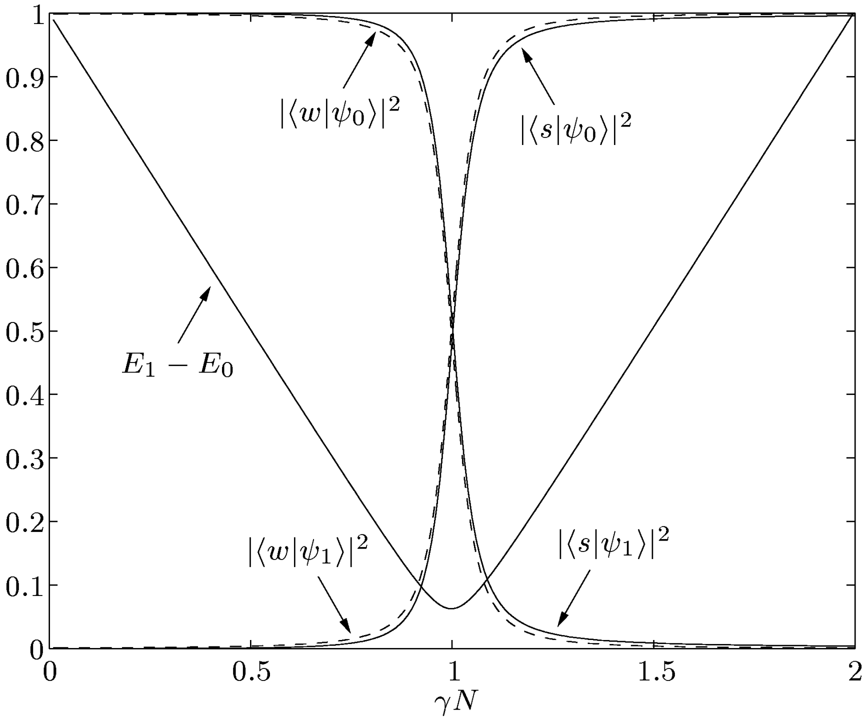

Figure 4-1: Energy gap and overlaps for the complete graph with $N = 1024$.

$$L + NI = N|s\rangle\langle s| = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} . \tag{4.6}$$

Therefore we consider the Hamiltonian

$$H = -\gamma N|s\rangle\langle s| - |w\rangle\langle w| . \tag{4.7}$$

This Hamiltonian acts nontrivially only on a two-dimensional subspace, so it is straightforward to compute its spectrum exactly for any value of γ. For $\gamma N \ll 1$, the ground state is close to $|w\rangle$, and for $\gamma N \gg 1$, the ground state is close to $|s\rangle$. In fact, for large N, there is a sharp change in the ground state from $|w\rangle$ to $|s\rangle$ as γN is varied from slightly less than 1 to slightly greater than 1. Correspondingly, the gap between the ground and first excited state energies is smallest for $\gamma N \sim 1$, as shown in Figure 4-1. At $\gamma N = 1$, for N large, the eigenstates are $\frac{1}{\sqrt{2}}(|w\rangle \pm |s\rangle)$ (up to terms of order $N^{-1/2}$), with a gap of $2/\sqrt{N}$. Thus the walk rotates the state from $|s\rangle$ to $|w\rangle$ in time $\pi\sqrt{N}/2$.

4.3.2 Hypercube

Now consider the n-dimensional hypercube with $N = 2^n$ vertices, as discussed in Section 3.3.1. In this case, we again find a sharp transition in the eigenstates at a certain critical value of γ, as shown in Figure 4-2. The Hamiltonian can be analyzed using essentially the same method we will apply in the next section, together with facts about spin operators. The energy gap is analyzed in Section 4.2 of [80], and the energy eigenstates are analyzed

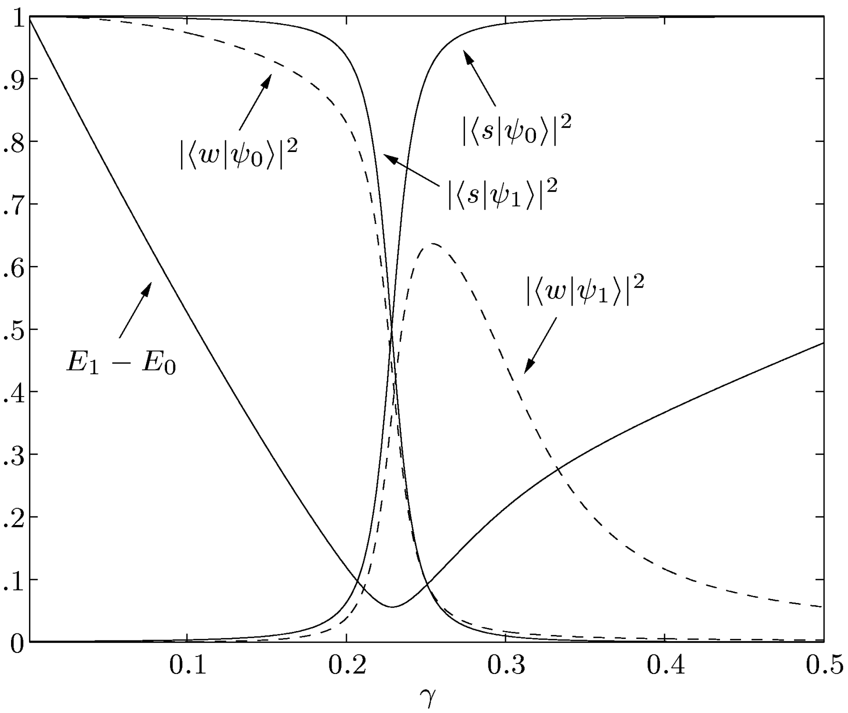

Figure 4-2: Energy gap and overlaps for the hypercube with $N = 2^{10} = 1024$.

in Section 2.5.5 above. The critical value of γ is

$$\gamma = \frac{1}{2^n} \sum_{r=1}^{n} \binom{n}{r} \frac{1}{r} = \frac{2}{n} + O(n^{-2}) \tag{4.8}$$

at which the energy gap is

$$\frac{2}{\sqrt{N}}[1 + O(n^{-1})]\,, \tag{4.9}$$

and the ground and first excited states are $\frac{1}{\sqrt{2}}(|w\rangle \pm |s\rangle)$ up to terms of order $1/n$. Again, we find that after a time of order $\sqrt{N}$, the probability of finding w is of order 1.

4.4 Finite dimensions

Having seen that the algorithm works in two cases where the dimension of the graph grows with N, we now consider the case of a d-dimensional cubic periodic lattice, where d is fixed independent of N. The minimum gap and overlaps of $|s\rangle, |w\rangle$ with the ground and first excited states are shown in Figure 4-3 for $d = 2, 3, 4, 5$ and $N \approx 1000$. In all of these plots, there is a critical value of γ where the energy gap is a minimum, and in the vicinity of this value, the state $|s\rangle$ changes from being the first excited state to being the ground state. In large enough d, the $|w\rangle$ state changes from being the ground state to having large overlap on the first excited state in the same region of γ. However, for smaller d, the range of γ over which the change occurs is wider, and the overlap of the $|w\rangle$ state on the lowest two eigenstates is smaller. Note that in all cases, $|s\rangle$ is supported almost entirely on the subspace of the two lowest energy states. Therefore, if the algorithm starting in the state

$|s\rangle$ is to work at all, it must work essentially in a two dimensional subspace.

In the rest of this section, we will make this picture quantitative. We begin with some general techniques for analyzing the spectrum of H using knowledge of the spectrum of the graph. We then show the existence of a phase transition in γ, and we show that for any d, the algorithm fails if γ is not close to a certain critical value. Next we consider what happens when γ is close to its critical value. In $d > 4$, we show that the algorithm gives a success probability of order 1 in time of order $\sqrt{N}$, and in $d = 4$, we find a success probability of order $1/\log N$ in time of order $\sqrt{N \log N}$. Finally, we investigate the critical point in $d < 4$ and show that the algorithm provides no speedup.

4.4.1 Preliminaries

In this section, we show how the spectrum of H can be understood in terms of the spectrum of L. An eigenvector of H, denoted $|\psi_a\rangle$, with eigenvalue E_a, satisfies

$$H|\psi_a\rangle = (-\gamma L - |w\rangle\langle w|)|\psi_a\rangle = E_a|\psi_a\rangle \,, \tag{4.10}$$

i.e.,

$$(-\gamma L - E_a)|\psi_a\rangle = |w\rangle\langle w|\psi_a\rangle \,. \tag{4.11}$$

The state $|\psi_a\rangle$ is normalized, so $|\langle\psi_a|\psi_a\rangle|^2 = 1$. Define

$$R_a = |\langle w|\psi_a\rangle|^2 \tag{4.12}$$

and choose the phase of $|\psi_a\rangle$ so that

$$\langle w|\psi_a\rangle = \sqrt{R_a} \,. \tag{4.13}$$

We wish to calculate the amplitude for success,

$$\langle w|e^{-iHt}|s\rangle = \sum_a \langle w|\psi_a\rangle\langle\psi_a|s\rangle e^{-iE_a t} \,, \tag{4.14}$$

so we only need those $|\psi_a\rangle$ with $R_a > 0$.

L is the Laplacian of a lattice in d dimensions, periodic in each direction with period $N^{1/d}$, with a total of N vertices. Each vertex of the lattice corresponds to a basis state $|x\rangle$, where x is a d-component vector with components $x_j \in \{0, 1, \ldots, N^{1/d} - 1\}$. The eigenvectors of $-L$ are the momentum eigenstates

$$|k\rangle = \frac{1}{\sqrt{N}} \sum_x e^{ik\cdot x}|x\rangle \,, \tag{4.15}$$

where k is a d-component vector with components

$$k_j = \frac{2\pi m_j}{N^{1/d}} \,, \quad m_j = \begin{cases} 0, \pm 1, \ldots, \pm\frac{1}{2}(N^{1/d} - 1) & N^{1/d} \text{ odd} \\ 0, \pm 1, \ldots, \pm\frac{1}{2}(N^{1/d} - 2), +\frac{1}{2}N^{1/d} & N^{1/d} \text{ even,} \end{cases} \tag{4.16}$$

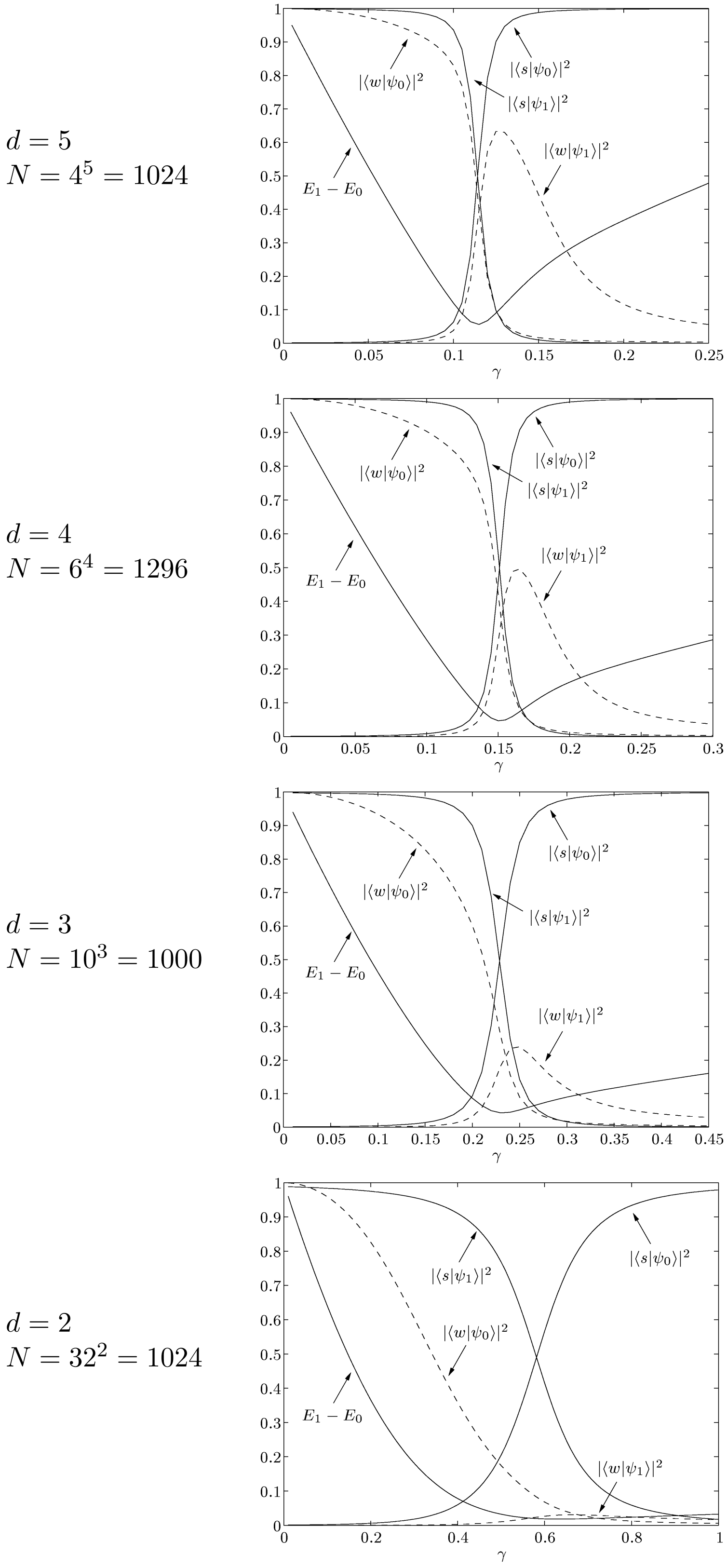

Figure 4-3: Energy gap and overlaps for d-dimensional lattices with $N \approx 1000$.

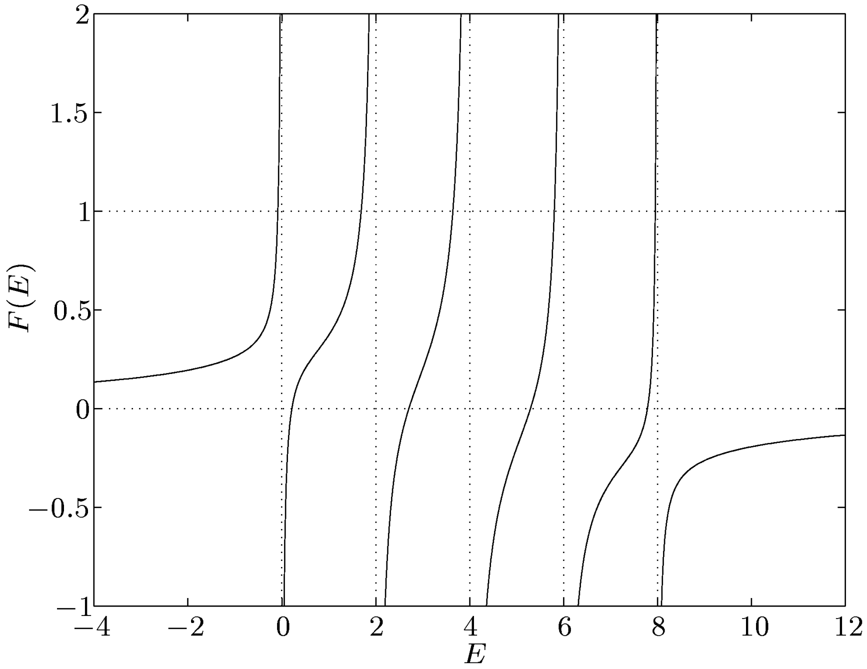

Figure 4-4: The function $F(E)$ for a $d = 2$ dimensional periodic lattice with $N = 16$ vertices, at $\gamma = 1$.

and the corresponding eigenvalues are

$$\mathcal{E}(k) = 2\left(d - \sum_{j=1}^{d} \cos(k_j)\right) . \tag{4.17}$$

Since $\langle k|w\rangle \neq 0$, from (4.11) we have

$$(\gamma\mathcal{E}(k) - E_a)\langle k|\psi_a\rangle \neq 0 \tag{4.18}$$

for any k. We can therefore rewrite (4.11), using (4.13), as

$$|\psi_a\rangle = \frac{\sqrt{R}_a}{-\gamma L - E_a}|w\rangle \, . \tag{4.19}$$

Consistency with (4.13) then gives the eigenvalue condition

$$\langle w|\frac{1}{-\gamma L - E_a}|w\rangle = 1 \, . \tag{4.20}$$

Using (4.15), this can be expressed as

$$F(E_a) = 1 \, , \quad F(E) = \frac{1}{N}\sum_k \frac{1}{\gamma\mathcal{E}(k) - E} \, . \tag{4.21}$$

A typical function $F(E)$ is shown in Figure 4-4. This function has poles where $E = \gamma\mathcal{E}(k)$. For $E \neq \gamma\mathcal{E}(k)$, (4.21) shows that $F'(E) > 0$, so there is an eigenvalue of H between every adjacent pair of eigenvalues of $-\gamma L$. Since $F(E) \to 0$ as $E \to \pm\infty$, there is also one negative eigenvalue of H (corresponding to the ground state). Note that in the case shown in Figure 4-4, the eigenvalues $\mathcal{E} = 2, 4, 6$ of $-\gamma L$ have degeneracies $4, 6, 4$ because of the symmetry of the lattice. It follows that there are $3, 5, 3$ eigenvectors of H with eigenvalues $E_a = 2, 4, 6$, all with $\langle w|\psi_a\rangle = 0$ and thus not relevant to our purpose. These 11 eigenvectors, together with the 5 relevant ones, make up the necessary total of 16.

The normalization condition on $|\psi_a\rangle$ gives

$$R_a \langle w| \frac{1}{(-\gamma L - E_a)^2} |w\rangle = 1\,, \tag{4.22}$$

i.e.

$$R_a = \frac{1}{F'(E_a)}\,. \tag{4.23}$$

We also need the overlap of $|\psi_a\rangle$ with $|s\rangle$. Since $L|s\rangle = 0$, from (4.19) we have

$$\langle s|\psi_a\rangle = -\frac{\sqrt{R_a}}{E_a}\langle s|w\rangle\,, \tag{4.24}$$

so that

$$|\langle s|\psi_a\rangle|^2 = \frac{1}{N}\frac{1}{E_a^2 F'(E_a)}\,. \tag{4.25}$$

Using (4.14), (4.19), and (4.20),

$$\langle w|e^{-iHt}|s\rangle = -\frac{1}{\sqrt{N}}\sum_a \frac{e^{-iE_a t}}{E_a F'(E_a)}\,. \tag{4.26}$$

At $t = 0$, this gives the sum rule

$$\sum_a \frac{1}{E_a F'(E_a)} = -1\,. \tag{4.27}$$

We will see that the spectrum of H depends significantly on the behavior of the sums

$$S_{j,d} = \frac{1}{N}\sum_{k\neq 0} \frac{1}{[\mathcal{E}(k)]^j}\,. \tag{4.28}$$

If $d > 2j$, then $S_{j,d}$ can be approximated by an integral as

$$S_{j,d} = I_{j,d} + o(1) \tag{4.29}$$

where

$$I_{j,d} = \frac{1}{(2\pi)^d}\int_{-\pi}^{\pi} \frac{\mathrm{d}^d k}{[\mathcal{E}(k)]^j}\,. \tag{4.30}$$

The condition $d > 2j$ is necessary for $I_{j,d}$ to converge at $k = 0$. The numerical values of $I_{1,d}$ and $I_{2,d}$ for $d \le 10$ are given in Table 4.1. Note that $I_{j,d}$ can also be calculated using the formula [145]

$$I_{j,d} = \frac{1}{(2d)^j}\int_0^\infty \mathrm{d}\alpha\, \frac{\alpha^{j-1} e^{-\alpha}}{(j-1)!}[I_0(\alpha/d)]^d \tag{4.31}$$

d	$I_{1,d}$	$I_{2,d}$
3	0.253	
4	0.155	
5	0.116	0.0184
6	0.0931	0.0105
7	0.0781	0.00697
8	0.0674	0.00504
9	0.0593	0.00383
10	0.0530	0.00301

Table 4.1: Numerical values of the convergent integrals. The result for $I_{1,3}$ is given exactly in [183]; the rest were computed numerically.

where $I_0(\cdot)$ is a modified Bessel function of the first kind.

On the other hand, if $d < 2j$, then $S_{j,d}$ can be well approximated by the contribution from values of k small enough that $\mathcal{E}(k)$ is approximately

$$\mathcal{E}(k) \approx k^2 = \frac{(2\pi m)^2}{N^{2/d}} \tag{4.32}$$

(where we have used the notation $k^2 = k_1^2 + \cdots + k_d^2$). Then

$$S_{j,d} \sim c_{j,d}\, N^{\frac{2j}{d}-1} \tag{4.33}$$

where

$$c_{j,d} = \frac{1}{(2\pi)^{2j}} \sum_{m \neq 0} \frac{1}{(m^2)^j}\,. \tag{4.34}$$

Here the sum is over all values of the d-component vector of integers m other than $m = 0$, and converges for large m^2. Numerically, we find

$$c_{2,2} = 0.00664\,, \quad c_{2,3} = 0.0265\,. \tag{4.35}$$

In the borderline case $d = 2j$, $I_{j,d}$ diverges logarithmically at k^2 small and $c_{j,d}$ diverges logarithmically at m^2 large. In this case

$$S_{j,2j} = \frac{1}{(4\pi)^j\, j!} \ln N + O(1)\,. \tag{4.36}$$

We will need

$$S_{1,2} = \frac{1}{4\pi} \ln N + A + O(N^{-1}) \tag{4.37}$$

$$S_{2,4} = \frac{1}{32\pi^2} \ln N + O(1) \tag{4.38}$$

where $A = 0.0488$ (the case $j = 1$, $d = 2$ is treated in greater detail in [146]).

4.4.2 Phase transition

In this section, we show that the overlap of the state $|s\rangle$ on the ground or first excited state of H exhibits a phase transition at a critical value of γ for any dimension d. In fact, away from the critical value, $|s\rangle$ is approximately an eigenstate of H, so Schrödinger evolution according to H does not change the state very much. In the next section, we will show that the algorithm indeed fails away from the critical value of γ, and in the following sections we will consider what happens near the critical point.

For γ larger than the critical value (which will be determined below), the ground state energy is very close to 0. This can be seen as follows. The eigenvalue condition (4.21) for the ground state energy E_0, which is negative, gives

$$1 = F(E_0) = \frac{1}{N|E_0|} + \frac{1}{N}\sum_{k\neq 0} \frac{1}{\gamma\mathcal{E}(k) + |E_0|} \tag{4.39}$$

$$< \frac{1}{N|E_0|} + \frac{1}{N}\sum_{k\neq 0} \frac{1}{\gamma\mathcal{E}(k)} \tag{4.40}$$

$$\approx \frac{1}{N|E_0|} + \frac{I_{1,d}}{\gamma} \tag{4.41}$$

where in the last line we have assumed $d > 2$. In this case, for $\gamma > I_{1,d}$ (which will turn out to be the critical value), up to small terms,

$$|E_0| < \frac{1}{N}\frac{\gamma}{\gamma - I_{1,d}}\,. \tag{4.42}$$

Using (4.25), we have

$$|\langle s|\psi_0\rangle|^2 = \Big[1 + E_0^2 \sum_{k\neq 0} (\gamma\mathcal{E}(k) + |E_0|)^{-2}\Big]^{-1} \tag{4.43}$$

$$> \left[1 + \frac{E_0^2}{\gamma^2}\sum_{k\neq 0}\frac{1}{[\mathcal{E}(k)]^2}\right]^{-1} \tag{4.44}$$

$$> 1 - \frac{E_0^2}{\gamma^2}\sum_{k\neq 0}\frac{1}{[\mathcal{E}(k)]^2}\,. \tag{4.45}$$

Inserting the behavior of $S_{2,d}$ from (4.28), (4.33), and (4.36) and using the bound (4.42), we find

$$1 - |\langle s|\psi_0\rangle|^2 < \frac{1}{(\gamma - I_{1,d})^2} \times \begin{cases} O(N^{-1}) & d > 4 \\ O(N^{-1}\log N) & d = 4 \\ O(N^{-2/3}) & d = 3\,. \end{cases} \tag{4.46}$$

This shows that if $\gamma = I_{1,d} + \epsilon$ for any $\epsilon > 0$, then $1 - |\langle s|\psi_0\rangle|^2$ approaches zero as $N \to \infty$.

If $d = 2$, then $I_{1,2}$ is logarithmically divergent, but using (4.37) in (4.40) we can apply a similar argument whenever $\gamma > \frac{1}{4\pi}\ln N + A$, in which case we have

$$|E_0| < \frac{1}{N}\frac{\gamma}{\gamma - \frac{1}{4\pi}\ln N - A} \tag{4.47}$$

and

$$1 - |\langle s|\psi_0\rangle|^2 < \frac{1}{(\gamma - \frac{1}{4\pi}\ln N - A)^2} \times O(1)\,. \tag{4.48}$$

This shows that if $\gamma > (\frac{1}{4\pi} + \epsilon)\ln N$, then $1 - |\langle s|\psi_0\rangle|^2 \leq 1/(\epsilon \ln N)^2$, which approaches zero as $N \to \infty$.

Similarly, for $d > 2$ and for $\gamma < I_{1,d}$, the first excited state $|\psi_1\rangle$, with energy $E_1 > 0$, is essentially $|s\rangle$. Here we find

$$1 = F(E_1) = -\frac{1}{NE_1} + \frac{1}{N}\sum_{k\neq 0} \frac{1}{\gamma\mathcal{E}(k) - E_1} \tag{4.49}$$

$$> -\frac{1}{NE_1} + \frac{1}{N}\sum_{k\neq 0} \frac{1}{\gamma\mathcal{E}(k)} \tag{4.50}$$

$$\approx -\frac{1}{NE_1} + \frac{I_{1,d}}{\gamma}\,, \tag{4.51}$$

so that, up to small terms,

$$E_1 < \frac{1}{N}\frac{\gamma}{I_{1,d} - \gamma}\,. \tag{4.52}$$

Again applying (4.25), we find

$$1 - |\langle s|\psi_1\rangle|^2 < \frac{1}{(I_{1,d} - \gamma)^2} \times \begin{cases} O(N^{-1}) & d > 4 \\ O(N^{-1}\log N) & d = 4 \\ O(N^{-2/3}) & d = 3\,. \end{cases} \tag{4.53}$$

We see that $\gamma = I_{1,d}$ is the critical point. In $d = 2$ we can apply similar reasoning to obtain that for $\gamma < \frac{1}{4\pi}\ln N + A$,

$$1 - |\langle s|\psi_1\rangle|^2 < \frac{1}{(\frac{1}{4\pi}\ln N - \gamma)^2} \times O(1)\,. \tag{4.54}$$

In this case $\gamma = \frac{1}{4\pi}\ln N + A$ is the critical point.

4.4.3 Failure of the algorithm away from the critical point

In this section we will show that the algorithm fails away from the critical point, regardless of dimension. The results (4.46) and (4.53) are actually sufficient to show that away from the critical point in $d > 4$, the algorithm can be no better than classical search, but we will give a different argument for consistency of presentation.

First we consider the regime where γ is larger than the critical value. In the previous section, we saw that in this case, the ground state energy E_0 is small. This is sufficient to imply that the success probability is small at all times. Combining (4.26) and (4.27), we see that the amplitude at an arbitrary time must satisfy

$$|\langle w|e^{-iHt}|s\rangle| \leq \frac{1}{\sqrt{N}}\left(\frac{2}{|E_0|F'(E_0)} - 1\right) \tag{4.55}$$

$$\leq \frac{2}{\sqrt{N}|E_0|F'(E_0)}\,. \tag{4.56}$$

Furthermore it is clear from the definition of $F(E)$ that

$$F'(E_0) \geq \frac{1}{NE_0^2}\,, \tag{4.57}$$

so

$$|\langle w|e^{-iHt}|s\rangle| \leq 2\sqrt{N}|E_0|\,. \tag{4.58}$$

Using (4.42), we find that for $d > 2$,

$$|\langle w|e^{-iHt}|s\rangle| \leq \frac{2}{\sqrt{N}}\frac{\gamma}{\gamma - I_{1,d}}\,. \tag{4.59}$$

This shows that if $\gamma = I_{1,d} + \epsilon$ for any $\epsilon > 0$, the success probability is never more than a constant factor larger than its initial value, no matter how long we run the algorithm. If $d = 2$, then $I_{1,2}$ is logarithmically divergent, but using (4.47) we find

$$|\langle w|e^{-iHt}|s\rangle| \leq \frac{2}{\sqrt{N}}\frac{\gamma}{\gamma - \frac{1}{4\pi}\ln N - A}\,. \tag{4.60}$$

This shows that the algorithm fails if $\gamma > (\frac{1}{4\pi} + \epsilon)\ln N$ for any $\epsilon > 0$.

Now we consider the case where γ is smaller than the critical value. For $d > 4$ and $E < 0$, we have

$$\begin{aligned}
F(E) &\approx \frac{1}{(2\pi)^d}\int \frac{\mathrm{d}^d k}{\gamma\mathcal{E}(k) + |E|} && (4.61)\\
&= \frac{1}{(2\pi)^d}\int \frac{\mathrm{d}^d k}{\gamma\mathcal{E}(k)} - \frac{|E|}{(2\pi)^d}\int \frac{\mathrm{d}^d k}{\gamma\mathcal{E}(k)[\gamma\mathcal{E}(k) + |E|]} && (4.62)\\
&> \frac{I_{1,d}}{\gamma} - \frac{|E|}{\gamma^2(2\pi)^d}\int \frac{\mathrm{d}^d k}{[\mathcal{E}(k)]^2} && (4.63)\\
&= \frac{I_{1,d}}{\gamma} - \frac{I_{2,d}}{\gamma^2}|E|\,. && (4.64)
\end{aligned}$$

Using the fact that $F(E_0) = 1$, this shows that

$$|E_0| > \frac{\gamma(I_{1,d} - \gamma)}{I_{2,d}}\,. \tag{4.65}$$

From (4.12) and (4.23), it is clear that $F'(E) > 1$, so using (4.56) gives

$$|\langle w|e^{-iHt}|s\rangle| < \frac{1}{\sqrt{N}}\frac{2I_{2,d}}{\gamma(I_{1,d} - \gamma)}\,. \tag{4.66}$$

A similar argument can be used for $d = 3, 4$. With $d = 4$, we have

$$\begin{aligned}
F(E) &\approx \frac{1}{(2\pi)^4}\int \frac{\mathrm{d}^4 k}{\gamma\mathcal{E}(k) + |E|} && (4.67)\\
&= \frac{1}{(2\pi)^4}\int \frac{\mathrm{d}^4 k}{\gamma\mathcal{E}(k)} - \frac{|E|}{(2\pi)^4}\int \frac{\mathrm{d}^4 k}{\gamma\mathcal{E}(k)[\gamma\mathcal{E}(k) + |E|]} && (4.68)
\end{aligned}$$

$$
\begin{aligned}
&> \frac{I_{1,4}}{\gamma} - \frac{|E|}{32\gamma}\int_0^{2\pi} \frac{k\mathrm{d}k}{\frac{4\gamma}{\pi^2}k^2+|E|} &(4.69)\\
&= \frac{I_{1,4}}{\gamma} - \frac{\pi^2|E|}{256\gamma^2}\ln\left(1+\frac{16\gamma}{|E|}\right)\,, &(4.70)
\end{aligned}
$$

where the third line follows because $\cos k \leq 1 - 2(k/\pi)^2$ for $|k| \leq \pi$, which implies $\mathcal{E}(k) \geq \frac{4}{\pi^2}k^2$. We have also used the fact that $k^2 \leq d\pi^2$ to place an upper limit on the integral. This shows that for any $\epsilon > 0$ (with $\epsilon \leq 1$), there exists a $c > 0$ such that

$$
F(E) > \frac{I_{1,4}}{\gamma} - \frac{c|E|^{1-\epsilon}}{\gamma^{2-\epsilon}}\,, \tag{4.71}
$$

so that

$$
|E_0| > c'\gamma(I_{1,d}-\gamma)^{1/(1-\epsilon)} \tag{4.72}
$$

for some $c' > 0$, and therefore

$$
|\langle w|e^{-iHt}|s\rangle| < \frac{1}{\sqrt{N}}\frac{2}{c'\gamma(I_{1,4}-\gamma)^{1/(1+\epsilon)}}\,. \tag{4.73}
$$

With $d = 3$, we have

$$
\begin{aligned}
F(E) &\approx \frac{1}{(2\pi)^3}\int \frac{\mathrm{d}^3k}{\gamma\mathcal{E}(k)+|E|} &(4.74)\\
&= \frac{1}{(2\pi)^3}\int \frac{\mathrm{d}^3k}{\gamma\mathcal{E}(k)} - \frac{|E|}{(2\pi)^3}\int \frac{\mathrm{d}^3k}{\gamma\mathcal{E}(k)[\gamma\mathcal{E}(k)+|E|]} &(4.75)\\
&> \frac{I_{1,3}}{\gamma} - \frac{|E|}{8\gamma}\int_0^{\infty} \frac{\mathrm{d}k}{\frac{4\gamma}{\pi^2}k^2+|E|} &(4.76)\\
&= \frac{I_{1,3}}{\gamma} - \frac{\pi^2}{32\gamma^{3/2}}\sqrt{|E|} &(4.77)
\end{aligned}
$$

where in the third line we have again used $\mathcal{E}(k) \geq \frac{4}{\pi^2}k^2$. In this case we find

$$
|E_0| > \frac{1024}{\pi^4}\gamma(I_{1,3}-\gamma)^2 \tag{4.78}
$$

which shows that

$$
|\langle s|e^{-iHt}|w\rangle| < \frac{1}{\sqrt{N}}\frac{2\pi^4}{1024\gamma(I_{1,3}-\gamma)^2}\,. \tag{4.79}
$$

Finally, with $d = 2$ we use a different argument. Here we have

$$
\begin{aligned}
F'(E) &\approx \frac{1}{(2\pi)^2}\int \frac{\mathrm{d}^2k}{[\gamma\mathcal{E}(k)+|E|]^2} &(4.80)\\
&> \frac{1}{2\pi}\int_0^{\pi} \frac{k\,\mathrm{d}k}{(\gamma k^2+|E|)^2} &(4.81)\\
&= \frac{\pi}{4|E|(|E|+\pi^2\gamma)} &(4.82)
\end{aligned}
$$

where the second line follows since $\cos k \geq 1 - \frac{1}{2}k^2$, which implies $\mathcal{E}(k) \leq k^2$. In the second line we have also used the fact that the entire disk $|k| \le \pi$ is included in the region of integration. Equation (4.82) shows that

$$|E|F'(E) > \frac{\pi}{4(|E| + \pi^2\gamma)} , \tag{4.83}$$

so that

$$|\langle w|e^{-iHt}|s\rangle| < \frac{1}{\sqrt{N}} \frac{8(|E_0| + \pi^2\gamma)}{\pi} , \tag{4.84}$$

which is $O(1/\sqrt{N})$ for $\gamma = O(1)$, and $O((\log N)/\sqrt{N})$ for any $\gamma < \frac{1}{4\pi} \ln N + A$.

The arguments for the case where γ is smaller than the critical value can be made tighter by a more refined analysis. For example, by considering the behavior of $F'(E)$, one can give a bound whose dependence on $I_{1,d} - \gamma$ is linear for all $d > 2$, not just for $d > 4$. Furthermore, the careful reader will note that our bounds for $d > 2$ all become useless as $\gamma \to 0$, but it is easy to see that the algorithm cannot be successful for small values of γ.

Altogether, we see that the algorithm cannot work any better than classical search if γ is not chosen close to its critical value. It remains to investigate what happens near the critical point.

4.4.4 The critical point in four dimensions and higher

In this section we investigate the region of the critical point in the cases where the algorithm provides speedup. First we consider the case $d > 4$. Separating out the $k = 0$ term in (4.21), we have

$$F(E) = -\frac{1}{NE} + \frac{1}{N} \sum_{k \ne 0} \frac{1}{\gamma \mathcal{E}(k) - E} . \tag{4.85}$$

If $|E| \ll \gamma\mathcal{E}(k)$ for all $k \ne 0$, then for large N, we can Taylor expand the second term to obtain

$$F(E) \approx -\frac{1}{NE} + \frac{1}{\gamma} I_{1,d} + \frac{E}{\gamma^2} I_{2,d} \tag{4.86}$$

which gives

$$F'(E) \approx \frac{1}{NE^2} + \frac{I_{2,d}}{\gamma^2} . \tag{4.87}$$

The critical point corresponds to the condition $\gamma = I_{1,d}$. At this point, setting (4.86) equal to 1 gives two eigenvalues,

$$E_0 \approx -\frac{I_{1,d}}{\sqrt{I_{2,d} N}} , \quad E_1 \approx +\frac{I_{1,d}}{\sqrt{I_{2,d} N}} , \tag{4.88}$$

which correspond to the ground and first excited state, with a gap of order $N^{-1/2}$. Since $\mathcal{E}(k) \approx (2\pi)^2 N^{-2/d}$ for $m^2 = 1$, we see that the assumption $E_0, E_1 \ll \gamma\mathcal{E}(k)$ holds for all $k \ne 0$. Furthermore, for the ground and first excited states at $\gamma = I_{1,d}$, (4.87) gives

$$F'(E_0) \approx F'(E_1) \approx \frac{2 I_{2,d}}{I_{1,d}^2} . \tag{4.89}$$

Now we want to use (4.26) to compute the time evolution of the algorithm. The contribution from all states above the first excited state is small, since as can be seen using

(4.27) we have

$$-\frac{1}{\sqrt{N}}\sum_{E_a>E_1}\frac{1}{E_a F'(E_a)}=\frac{1}{\sqrt{N}}\left(1+\frac{1}{E_0F'(E_0)}+\frac{1}{E_1F'(E_1)}\right). \tag{4.90}$$

Using (4.88) and (4.89), we see that the $O(\sqrt{N})$ contributions from the terms $1/E_0F'(E_0)$ and $1/E_1F'(E_1)$ cancel, so the right hand side of (4.90) is $o(1)$. Thus, using (4.26), we find

$$|\langle w|e^{-iHt}|s\rangle|\approx\frac{I_{1,d}}{\sqrt{I_{2,d}}}\left|\sin\left(\frac{I_{1,d}\,t}{\sqrt{I_{2,d}N}}\right)\right|. \tag{4.91}$$

The success probability is of order 1 at $t=\sqrt{I_{2,d}N}/I_{1,d}$. Straightforward analysis shows that a similar condition holds so long as $\gamma=I_{1,d}\pm O(N^{-1/2})$, exactly the width of the region that cannot be excluded based on the arguments of Section 4.4.3.

In $d=4$, $I_{2,d}$ does not converge, so the result is modified slightly. In this case (4.86) holds with $I_{2,d}$ replaced by $\frac{1}{32\pi^2}\ln N$, so the ground and first excited state energies are given by

$$E_0\approx-\frac{I_{1,4}}{\sqrt{\frac{1}{32\pi^2}N\ln N}}\,,\quad E_1\approx+\frac{I_{1,4}}{\sqrt{\frac{1}{32\pi^2}N\ln N}}\,, \tag{4.92}$$

and we find

$$F'(E_0)\approx F'(E_1)\approx\frac{\ln N}{16\pi^2 I_{1,4}^2}. \tag{4.93}$$

Therefore

$$|\langle w|e^{-iHt}|s\rangle|\approx\frac{I_{1,4}}{\sqrt{\frac{1}{32\pi^2}\ln N}}\left|\sin\left(\frac{I_{1,4}\,t}{\sqrt{\frac{1}{32\pi^2}N\ln N}}\right)\right|,$$

which shows that running for a time of order $\sqrt{N\log N}$ gives a success probability of order $1/\log N$. Using $O(\log N)$ repetitions to boost the success probability close to 1, we find a total run time $O(\sqrt{N}\log^{3/2}N)$. One can show that similar conditions hold so long as $\gamma=I_{1,4}\pm O(\sqrt{(\log N)/N})$.

In fact, we could improve the run time of the algorithm to $O(\sqrt{N}\,\log N)$ using amplitude amplification [33]. The same technique could also be used in $d=2,3$, but in those cases we would find that the resulting algorithm is still slower than $O(\sqrt{N})$ by some power of N.

For $d<4$, the expansion (4.86) fails to find states whose energies satisfy $E\ll\gamma\mathcal{E}(k)$. Indeed, we will see in the next section that the algorithm provides no speedup in these cases.

4.4.5 The critical point below four dimensions

To handle the case $d<4$, we rearrange the eigenvalue condition to extract the $O(1)$ contribution to $F(E)$:

$$F(E)=-\frac{1}{NE}+\frac{1}{N}\sum_{k\neq0}\frac{1}{\gamma\mathcal{E}(k)}+\frac{1}{N}\sum_{k\neq0}\frac{E}{\gamma\mathcal{E}(k)[\gamma\mathcal{E}(k)-E]}. \tag{4.94}$$

In $d=3$, we can replace the middle term by $I_{1,3}/\gamma$ for large N. To explore the neighborhood of the critical point in $d = 3$, we introduce rescaled variables a, x via

$$\gamma = I_{1,3} + \frac{a}{N^{1/3}} \tag{4.95}$$

$$E = \frac{4\pi^2 I_{1,3}}{N^{2/3}}\, x\,. \tag{4.96}$$

Since the sum in the third term of (4.94) only gets significant contributions from small energies, we use (4.32) to give the approximation

$$\gamma \mathcal{E}(k) \approx \frac{4\pi^2 I_{1,3} m^2}{N^{2/3}}\,, \tag{4.97}$$

and we can analyze the sum using the same techniques we applied to calculate $S_{j,d}$ in the case $d < 2j$. Then we have, for large N,

$$F(E) \approx 1 + \frac{G_3(x) - a}{I_{1,3} N^{1/3}} \tag{4.98}$$

where

$$G_3(x) = \frac{1}{4\pi^2} \left(\sum_{m \neq 0} \frac{x}{m^2(m^2 - x)} - \frac{1}{x} \right)\,. \tag{4.99}$$

Here the sum is over all integer values of m, as in (4.34), and similarly converges for large m^2. The eigenvalue condition in terms of x is $G_3(x) = a$, which has one negative solution x_0. Since $G_3(x)$ is independent of N, x_0 is independent of N, and the ground state energy E_0 is proportional to $N^{-2/3}$.

As we saw in Section 4.4.3, a very small ground state energy implies that the success probability is small at all times. Using (4.58), we find

$$|\langle w|e^{-iHt}|s\rangle| \le \frac{8\pi^2 I_{1,3} |x_0|}{N^{1/6}}\,. \tag{4.100}$$

Therefore the success probability is small no matter how long we run the algorithm. In fact, the small gap shows that we have to wait for a time of order $N^{2/3}$ even to get a probability of order $N^{-1/3}$.

Similar considerations hold in the case $d = 2$. In this case, the critical point is at $\gamma = \frac{1}{4\pi} \ln N + A$, so we choose

$$\gamma = \frac{1}{4\pi} \ln N + A + a \tag{4.101}$$

$$E = \frac{2\pi \ln N}{N}\, x\,. \tag{4.102}$$

In this case, we find

$$F(E) \approx 1 + \frac{G_2(x) - a}{\frac{1}{4\pi} \ln N}\,, \tag{4.103}$$

where $G_2(x)$ is defined as in (4.99), but with m having two components instead of three.

Again we find a solution $x_0 < 0$ that is independent of N, and applying (4.58) gives

$$|\langle w|e^{-iHt}|s\rangle| \le \frac{4\pi|x_0|\ln N}{\sqrt{N}}\,. \tag{4.104}$$

(Note that we could have reached a similar conclusion using (4.84).) Therefore the algorithm also fails near the critical point in $d = 2$.

4.5 The Dirac equation and an improved algorithm in low dimensions

So far, we have considered a quantum walk algorithm using no additional memory beyond the present location of the walker. We showed that this algorithm can find a single marked site in time $O(\sqrt{N})$ for dimensions $d > 4$ and in time $O(\sqrt{N}\log N)$ in four dimensions. We also showed that this algorithm fails to provide an interesting speedup for dimensions $d < 4$. After this result appeared [46], Ambainis, Kempe, and Rivosh found a discrete-time quantum walk algorithm that works in lower dimensions [12]. This algorithm runs in time $O(\sqrt{N})$ for $d > 2$ and in time $O(\sqrt{N}\log N)$ in two dimensions.

Because a discrete-time quantum walk cannot be defined on a state space consisting only of the vertices of a graph, as discussed in Section 3.4, the algorithm of [12] necessarily uses additional memory. In this section, we consider a continuous-time quantum walk using additional memory, and we show that it achieves the same running times as the discrete-time algorithm.

Dirac found that a consistent quantum mechanical description of a free relativistic particle requires the introduction of spin degrees of freedom [66]. Since this idea is essential to our construction, we outline it here. The relationship between the energy E, momentum p, and mass m of a relativistic particle is $E^2 = p^2 + m^2$ (in units where the speed of light is 1). To quantize such a particle, Dirac considered a Hamiltonian of the form

$$H = \sum_{j=1}^{d} \alpha_j p_j + \beta m \tag{4.105}$$

where the operators α_j and β act on the spin degrees of freedom, and $p = -i\hbar\frac{\mathrm{d}}{\mathrm{d}x}$ is the momentum operator. If the spin operators α_j and β satisfy the anticommutation relations

$$\{\alpha_j, \alpha_k\} = 2\delta_{j,k}\,, \quad \{\alpha_j, \beta\} = 0\,, \quad \beta^2 = 1\,, \tag{4.106}$$

then one indeed finds $H^2 = p^2 + m^2$. To write down the Dirac equation in d dimensions, we need $d+1$ anticommuting operators. The minimal representation of the algebra (4.106) uses $2^{\lceil d/2 \rceil}$-dimensional matrices, and hence there are $2^{\lceil d/2 \rceil}$ spin components.

Previously, we considered the Hamiltonian (4.4), which is the Hamiltonian of a free, spinless, non-relativistic particle plus a potential term at the marked site. Since the free Hamiltonian $-\gamma L$ is translationally invariant, its eigenstates are the momentum eigenstates $|k\rangle$ given in (4.15). For small k, the energy of the state $|k\rangle$ is $\mathcal{E}(k) \approx \gamma k^2$. As we saw in Section 4.4, this quadratic dispersion relation ultimately gives rise to the critical dimension $d = 4$.

To find an algorithm with a critical dimension of 2, we might expect to require a free

Hamiltonian with a linear dispersion relation. This can be achieved by introducing spin degrees of freedom and using the massless Dirac Hamiltonian, equation (4.105) with $m = 0$. On the lattice, the continuum operator p_j can be discretely approximated as

$$P_j|x\rangle = \frac{i}{2}(|x + e_j\rangle - |x - e_j\rangle)\,, \tag{4.107}$$

where e_j is a unit vector in the j direction. However, as we will see later, it turns out that simply taking the free Hamiltonian (4.105) using the lattice approximation (4.107) is insufficient. Instead, we will take[2]

$$H_0 = \omega \sum_j \alpha_j P_j + \gamma\beta L \tag{4.108}$$

where both ω and γ are adjustable parameters. For a Hamiltonian with spin degrees of freedom, translation invariance shows that the eigenstates have the form $|\eta, k\rangle$, where $|\eta\rangle$ is a (momentum-dependent) spin state. For (4.108), we find states with energies

$$\mathcal{E}(k) = \pm\sqrt{\omega^2 s^2(k) + \gamma^2 c^2(k)}\,, \tag{4.109}$$

where

$$s^2(k) = \sum_{j=1}^{d} \sin^2 k_j \tag{4.110}$$

$$c(k) = 2\sum_{j=1}^{d}(1 - \cos k_j)\,. \tag{4.111}$$

For small momenta, we have $\mathcal{E}(k) \approx \pm\omega|k|$, which leads to a better search algorithm in low dimensions.

The full algorithm is as follows. We begin in the state $|\eta, s\rangle$, where $|\eta\rangle$ is any spin state and $|s\rangle$ is the uniform superposition (4.5). We then evolve with the Hamiltonian

$$H = H_0 - \beta|w\rangle\langle w| \tag{4.112}$$

with parameters ω, γ to be determined in the analysis below, for a time T also determined below. The goal is to choose the parameters ω and γ so that for some T as small as possible, the spatial component of the evolved state has a substantial overlap on $|w\rangle$.

To analyze the algorithm, we would like to determine the spectrum of H using our knowledge of the spectrum of H_0. We do this using the same techniques we applied to the Hamiltonian (4.4) in Section 4.4.

An eigenvector of H, denoted $|\psi_a\rangle$, with eigenvalue E_a, satisfies

$$H|\psi_a\rangle = (H_0 - \beta|w\rangle\langle w|)|\psi_a\rangle = E_a|\psi_a\rangle\,. \tag{4.113}$$

Defining

$$\langle w|\psi_a\rangle = \sqrt{R_a}|\phi_a\rangle \tag{4.114}$$

[2] This choice is closely related to a standard remedy for the fermion doubling problem in lattice field theory [56, p. 27].

where $|\phi_a\rangle$ is a normalized spin state, and $\sqrt{R_a} > 0$ by choice of phases, we can rewrite (4.113) as

$$(H_0 - E_a)|\psi_a\rangle = \sqrt{R_a}\,\beta|\phi_a, w\rangle\,. \tag{4.115}$$

Assuming $H_0 - E_a$ is nonsingular, we can write the eigenstate of H as

$$|\psi_a\rangle = \frac{\sqrt{R_a}}{H_0 - E_a}\beta|\phi_a, w\rangle\,. \tag{4.116}$$

Consistency with (4.114) then gives the eigenvalue condition

$$|\phi_a\rangle = F(E_a)\beta|\phi_a\rangle \tag{4.117}$$

where

$$F(E) = \langle w|\frac{1}{H_0 - E}|w\rangle\,. \tag{4.118}$$

In other words, to find eigenvalues of H, we must look for values of E such that the spin operator $F(E)\beta$ has an eigenvector of eigenvalue 1.

In addition to finding eigenvalues of H, we need some facts about it eigenvectors. The normalization condition on $|\psi_a\rangle$ gives

$$\begin{align} R_a^{-1} &= \langle\phi_a, w|\beta\frac{1}{(H_0 - E_a)^2}\beta|\phi_a, w\rangle \tag{4.119}\\ &= \langle\phi_a|\beta F'(E_a)\beta|\phi_a\rangle\,. \tag{4.120}\end{align}$$

We also need the overlap of $|\psi_a\rangle$ with eigenvectors of H_0. From (4.116) we have

$$\langle\mathcal{E}|\psi_a\rangle = \frac{\sqrt{R_a}}{\mathcal{E} - E_a}\langle\mathcal{E}|\beta|\phi_a, w\rangle \tag{4.121}$$

where $|\mathcal{E}\rangle$ is an eigenvector of H_0 with eigenvalue $\mathcal{E}$.

For the free Hamiltonian (4.108), we find

$$\begin{align} F(E)\beta &= \langle w|\frac{H_0 + E}{H_0^2 - E^2}|w\rangle\beta \tag{4.122}\\ &= \frac{1}{N}\sum_k \frac{\gamma\, c(k) + \beta E}{\mathcal{E}(k)^2 - E^2} \tag{4.123}\\ &= -\frac{\beta}{NE} + U(E) + \beta\, E\, V(E) \tag{4.124}\end{align}$$

where in (4.123) we have canceled terms that are odd in k, and

$$\begin{align} U(E) &= \frac{1}{N}\sum_{k\neq 0} \frac{\gamma\, c(k)}{\mathcal{E}(k)^2 - E^2} \tag{4.125}\\ V(E) &= \frac{1}{N}\sum_{k\neq 0} \frac{1}{\mathcal{E}(k)^2 - E^2}\,. \tag{4.126}\end{align}$$

If $|E| \ll \mathcal{E}(k)$ for all $k \neq 0$, then for large N, we can Taylor expand $U(E)$ and $V(E)$ in powers of E. In fact, one can show that it is sufficient to keep only the leading order terms

$U(0)$ and $V(0)$. For large N, we have

$$U(0) \approx \frac{1}{(2\pi)^d} \int_{-\pi}^{\pi} \frac{\gamma\, c(k)\, \mathrm{d}^d k}{\mathcal{E}(k)^2}\,, \tag{4.127}$$

which is a convergent integral regardless of d. For $d > 2$ and N large, we can also write $V(0)$ as a convergent integral,

$$V(0) \approx \frac{1}{(2\pi)^d} \int_{-\pi}^{\pi} \frac{\mathrm{d}^d k}{\mathcal{E}(k)^2}\,. \tag{4.128}$$

In $d = 2$, this integral is logarithmically infrared divergent, and instead we find

$$V(0) = \frac{1}{4\pi} \ln N + O(1)\,. \tag{4.129}$$

Now suppose we choose ω and γ such that $U(0) = 1$. In this case, we are simply looking for a zero eigenvalue of $\beta(-\frac{1}{NE} + E\, V(0))$. We find such eigenvalues with

$$E_\pm \approx \pm \frac{1}{\sqrt{V(0)\, N}}\,, \tag{4.130}$$

which indeed satisfy the condition $|E_\pm| \ll \mathcal{E}(k)$ for all $k \neq 0$. These eigenvalues are degenerate in the spin space, i.e., any state $|\phi_\pm\rangle$ provides an eigenvector with the same eigenvalue.

The condition $U(0) = 1$ can be satisfied by choosing $u(\omega/\gamma) = \gamma$, where $u(\omega/\gamma) = \gamma\, U(0)$ is a function only of ω/γ. Figure 4-5 shows the critical curve in the (ω, γ) plane for $d = 2$ through 5. For any ω with $0 < \omega < \omega^*$, where ω^* is some dimension-dependent threshold value, there are two values of γ such that $U(0) = 1$. Note that with $\omega = 0$, we recover the results of Section 4.4. Also, with $\gamma = 0$, no solution of $U(0) = 1$ exists, so it was essential to include this additional term (and the ability to fine tune its coefficient).

Having found the relevant eigenvalues, we need to determine the corresponding eigenstates. Using (4.120) we find

$$R_\pm^{-1} \approx \frac{1}{N E_\pm^2} + V(0) \approx 2V(0)\,, \tag{4.131}$$

and using (4.121) we find

$$\langle \eta, s | \psi_\pm \rangle = -\frac{\sqrt{R_\pm}}{E_\pm \sqrt{N}} \langle \eta | \beta | \phi_\pm \rangle \approx \mp \frac{1}{\sqrt{2}}\,, \tag{4.132}$$

where we have chosen the eigenstate of H with $|\phi_\pm\rangle = \beta|\eta\rangle$. Therefore we have

$$|\eta, s\rangle \approx \frac{1}{\sqrt{2}}(|\psi_-\rangle - |\psi_+\rangle)\,, \tag{4.133}$$

and choosing $T = \pi/(2|E_\pm|)$ produces the state

$$e^{-iHT} |\eta, s\rangle \approx \frac{1}{\sqrt{2}}(|\psi_+\rangle + |\psi_-\rangle) \tag{4.134}$$

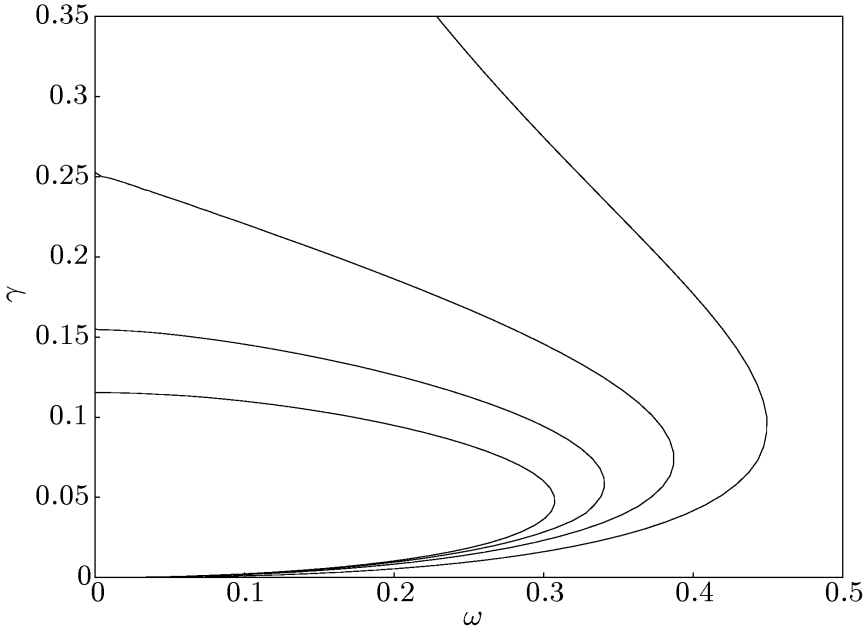

Figure 4-5: Critical values of (ω, γ) for various dimensions. From rightmost curve to leftmost curve, $d = 2, 3, 4, 5$.

which has an overlap on $|\eta, w\rangle$ of $\sqrt{2R_\pm}$.

For $d > 2$, we have shown that there is a $T = O(\sqrt{N})$ that gives a probability $O(1)$ of finding w. For $d = 2$, there is a $T = O(\sqrt{N \log N})$ that gives an amplitude $O(1/\sqrt{\log N})$, so that classical repetition can be used to find w with high probability in time $O(\sqrt{N} \log^{3/2} N)$, and amplitude amplification [33] can be used to find w with high probability in time $O(\sqrt{N} \log N)$.

This algorithm is closely related to the discrete-time quantum walk search algorithm of [12]. Very similar techniques to the ones we have used in this section (and Section 4.4) can also be applied to discrete-time quantum walks, as described in [42]. This analysis for the algorithm of [12] closely parallels the analysis above, which highlights the similarity between the two kinds of algorithms. However, there are a few important differences. The continuous-time algorithm requires fine tuning the parameters ω and γ, whereas there is (apparently) no equivalent fine tuning in the discrete-time algorithm. Also, the discrete-time algorithm has noticeably different behavior depending on whether $N^{1/d}$ is odd or even, a difference that is not seen in the continuous-time algorithm. In short, although the essential infrared features of the two kinds of algorithms are identical, their detailed behaviors differ.

In high dimensions, our algorithm is very wasteful in terms of the number of spin degrees of freedom: it uses a $2^{\lceil d/2 \rceil}$-dimensional spin space, whereas [46] shows that no spin degrees of freedom are required at all for $d > 4$. In comparison, the discrete-time quantum walk search algorithm in [12] uses $2d$ extra degrees of freedom. The Dirac particle in d dimensions cannot be represented with fewer than $2^{\lceil d/2 \rceil}$ degrees of freedom, but a continuous-time search algorithm with only $d + 1$ degrees of freedom can arise from reproducing the Dirac algebra (4.106) only on a subspace. If the operators α_j and β satisfy

$$\{\alpha_j, \alpha_k\}|\eta\rangle = 2\delta_{j,k}|\eta\rangle\,, \ \{\alpha_j, \beta\}|\eta\rangle = 0\,, \ \beta|\eta\rangle = |\eta\rangle \tag{4.135}$$

for some spin state $|\eta\rangle$, then the algorithm will work starting from the state $|\eta, s\rangle$. The condition (4.135) is sufficient to give $H_0^2|\eta, k\rangle = \mathcal{E}(k)^2|\eta, k\rangle$. The previous analysis then shows that

$$|\psi_a\rangle = \frac{\sqrt{R_a}}{H_0 - E_a}|\eta, w\rangle \tag{4.136}$$

is an eigenstate of H with eigenvalue E_a provided $-\frac{1}{NE_a} + U(E_a) + E_a V(E_a) = 1$, where $U(E)$ and $V(E)$ are as defined in equations (4.125) and (4.126). The rest of the analysis with two states $|\psi_\pm\rangle$ follows exactly as before. Finally, we see that (4.135) can be satisfied in a $(d+1)$-dimensional spin space with basis $|0\rangle, |1\rangle, \ldots, |d\rangle$, since in that case we can choose $\alpha_j = |0\rangle\langle j| + |j\rangle\langle 0|$, $\beta = 2|0\rangle\langle 0| - I$, and $|\eta\rangle = |0\rangle$.

4.6 Simulation in the local unitary model

For the spatial search problem, the Hamiltonian model might directly provide a reasonable physical model of a computational device. However, one can also express the problem in terms of a unitary gate model, where the gates are only allowed to move amplitude between adjacent locations on the graph [1]. In this section, we show that the continuous-time quantum walk algorithm can be efficiently simulated in such a model, without affecting the computational speedup. For simplicity, we discuss the spinless algorithm, although similar considerations apply to the Dirac-inspired algorithm.

The Hamiltonian (4.4) can easily be written as a sum of terms that can individually be simulated locally with only linear overhead. The only difficulty arises when we try to combine the terms using Rule 1.4. In Chapter 1, we were content with any polynomial-time simulation, but the quadratic overhead involved in naive implementation of Rule 1.4 apparently eliminates the quadratic speedup we worked so hard to achieve. Taking the expansion to higher orders as discussed in Section 1.4 improves the situation somewhat, but only allows us to achieve an algorithm that runs in time $O(N^{(1/2)+\epsilon})$ for arbitrarily small $\epsilon > 0$.

Fortunately, we can do better than this by using more detailed information about the expansion. Using the first order approximation (1.7) to simulate $H_1 + H_2$ with small error, the Baker-Campbell-Hausdorff theorem shows that we actually need only on the order of $\|[H_1, H_2]\|\, t^2$ alternations between the two simulations. For example, to simulate the search algorithm on the complete graph, note that $\|[|s\rangle\langle s|, |w\rangle\langle w|]\| = \sqrt{\frac{1}{N}(1-\frac{1}{N})}$, so that $O(\sqrt{N})$ simulation steps suffice [159].

For finite-dimensional lattices, $\|[L, |w\rangle\langle w|\|$ is of order 1, so the above argument does not immediately apply. However, we have seen that the algorithm works almost entirely in a subspace spanned by the uniform superposition $|s\rangle$ and another state $|\psi\rangle$ that is orthogonal to $|s\rangle$ (and that has substantial overlap on $|w\rangle$). Since we only need to accurately reproduce the evolution in this two-dimensional subspace, the simulation can be made sufficiently good using only $O(\|\Pi[L, |w\rangle\langle w|]\Pi\|\, t^2)$ alternations between the simulation of L and the simulation of $|w\rangle\langle w|$, where Π is a projector onto the subspace spanned by $|s\rangle$ and $|\psi\rangle$. A straightforward calculation shows that

$$\|\Pi[L, |w\rangle\langle w|]\Pi\| = \frac{|\langle w|L|\psi\rangle|}{\sqrt{N}}\,; \tag{4.137}$$

since $\|L\| = 4d$, this shows that $O(\sqrt{N})$ alternations suffice.

To simulate L using a sequence of local unitary operations, we can break it up into a sum of locally simulable pieces and combine those pieces using Rule 1.4. If $N^{1/d}$ is even, a simple decomposition that works is

$$L = \sum_{j=1}^{d} (L_j^{\text{even}} + L_j^{\text{odd}}) \,, \tag{4.138}$$

where

$$L_j^{\text{even}}|x\rangle = \begin{cases} |x+e_j\rangle - |x\rangle & x_j \text{ even} \\ |x-e_j\rangle - |x\rangle & x_j \text{ odd} \end{cases} \tag{4.139}$$

$$L_j^{\text{odd}}|x\rangle = \begin{cases} |x+e_j\rangle - |x\rangle & x_j \text{ odd} \\ |x-e_j\rangle - |x\rangle & x_j \text{ even.} \end{cases} \tag{4.140}$$

Since $L_j^{\text{even}}|s\rangle = L_j^{\text{odd}}|s\rangle = 0$, we find $\left\|\Pi[L_j^p, L_{j'}^{p'}]\Pi\right\| = 0$ for any j, j' and for any $p, p' \in \{\text{even}, \text{odd}\}$. This shows that a simulation of L that is sufficiently accurate in the relevant subspace can also be achieved with only linear overhead. Therefore, a local unitary discretization of the full algorithm can indeed find the marked item in $O(\sqrt{N})$ steps.

4.7 Discussion

In this chapter, we have presented a general approach to the spatial search problem using a continuous-time quantum walk on a graph. Using a straightforward quantum walk with no extra degrees of freedom, we showed that quadratic speedup can be achieved if the graph is a lattice of sufficiently high dimension ($d > 4$). Furthermore, by using Dirac's insight of introducing spin to take the square root in a relativistic dispersion relation, we showed that quadratic speedup can also be achieved for $d > 2$, and can nearly be achieved in $d = 2$.

Our algorithm begins in the state that is delocalized over the entire graph. One might demand instead that we start at a particular vertex of the graph. However, it is clear that the delocalized state $|s\rangle$ can be prepared from a localized state using $O(N^{1/d})$ local operations. In fact, we could also prepare $|s\rangle$ by running the quantum walk search algorithm backward from a known localized state for the same amount of time it would take to find $|w\rangle$ starting from $|s\rangle$.

The simple quantum walk search algorithm without spin degrees of freedom is closely related to a search algorithm using adiabatic evolution. In the adiabatic version of the search algorithm, the quantum computer is prepared in the state $|s\rangle$ (the ground state of H with γ large), and γ is slowly lowered from a large value to 0. If γ is changed sufficiently slowly, then the adiabatic theorem ensures that the quantum computer ends up near the final ground state $|w\rangle$, thus solving the problem. Recall from Section 2.2 that the time required to achieve a success probability of order 1 is inversely proportional to the square of the gap between the ground and first excited state energies. On the complete graph, the fact that the gap is only small (of order $1/\sqrt{N}$) for a narrow range of γ (also of order $1/\sqrt{N}$) means that γ can be changed in such a way that time $O(\sqrt{N})$ is sufficient to solve the problem [158, 59]. Since the gap has similar behavior for the hypercube and for d-dimensional lattices with $d > 4$, quadratic speedup can also be achieved adiabatically in these cases. In $d = 4$, the gap is of order $1/\sqrt{N \log N}$ for a range of γ of order $\sqrt{(\log N)/N}$, so the run time is $O(\sqrt{N}\log^{3/2} N)$, just as for the quantum walk search algorithm using classical repetition. In $d < 4$, no speedup can be achieved adiabatically.

In contrast, the Dirac-inspired algorithm from Section 4.5 cannot be turned into an adiabatic algorithm. In that case, the ground state of the free Hamiltonian (4.108) does not have $k = 0$; the zero-momentum states have zero energy, but this puts them in the middle of the spectrum. Although the adiabatic theorem applies to any eigenstate, not just the ground state, states near the middle of the spectrum of (4.112) with ω, γ small have very little overlap on $|w\rangle$, so that even perfectly adiabatic evolution produces a state far from the desired one.

The search problem can also be solved with high probability using a single measurement of H (followed by a measurement in the computational basis), as described in Section 2.5. The case of a hypercube was analyzed in Section 2.5.4, and our present results show that this algorithm can also be used when the graph is a lattice with $d > 4$ (or $d > 2$ if spin degrees of freedom are introduced). However, to realize the measurement dynamically, the Hamiltonian H must be coupled to a pointer variable, which must be represented using auxiliary space.

Finally, we note that the actual complexity of the spatial search problem in two dimensions is still an open question. A gap of $\log N$ remains between the best known algorithm and the lower bound of [19]. It would be interesting either to improve the algorithm further or to show that no such improvement is possible.

Chapter 5

Exponential speedup by quantum walk

5.1 Introduction

In this chapter, we show how quantum walks can be used to achieve exponential speedup over classical processes. We begin by presenting an example of a case in which a quantum walk can move through a graph exponentially faster than its classical counterpart. In particular, we consider a situation in which two vertices are designated as the ENTRANCE and EXIT, and we show that a quantum walk starting from the ENTRANCE has a substantial probability of reaching the EXIT much sooner than the corresponding classical random walk.

However, our ultimate goal is to find quantum algorithms that work faster than *any* classical algorithm, not just an algorithm of a particular type, such as a random walk. We would especially like to find algorithms that run exponentially faster (in contrast to polynomial speedups, such as the quadratic speedup we discussed in Chapter 4). Exponential quantum algorithmic speedup has been demonstrated for a number of different problems, but in each case, the quantum algorithm for solving the problem relies on the quantum Fourier transform. In this chapter, we demonstrate that exponential algorithmic speedup can be achieved using a quantum walk. In particular, we show how to solve an oracular computational problem exponentially faster than any classical algorithm could.

As discussed in Section 0.3, oracular problems provided the first examples of algorithmic speedup using a quantum instead of a classical computer, ultimately leading to Shor's factoring algorithm [165] and a number of related algorithms providing superpolynomial speedup over classical methods. However, all of these algorithms are fundamentally based on quantum Fourier transforms.

The idea of using quantum walks to achieve algorithmic speedup was explored by Farhi and Gutmann [82], who studied the behavior of quantum walks on decision trees. They also gave the first of several examples of graphs in which a quantum walk is exponentially faster than a classical walk. We present a simpler example of such graphs in Section 5.2. However, as we explain in Section 5.3, these results do not imply algorithmic speedup. Instead, we modify the example of Section 5.2 to construct an oracular problem that can be solved efficiently using a quantum walk, but that no classical algorithm can solve in subexponential time. We conclude in Section 5.4 with a discussion of the results and some remarks about open problems.

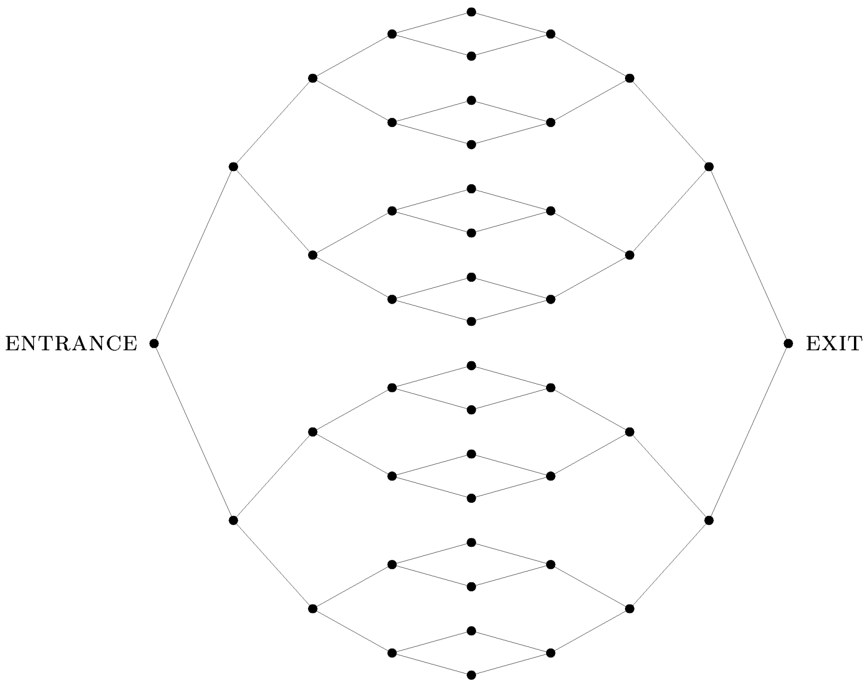

Figure 5-1: The graph G_4.

5.2 Speedup over classical random walk

In this section, we give an example of a sequence of graphs in which the behavior of a quantum walk is radically different from the behavior of the corresponding classical random walk. In particular, we show that the quantum walk can propagate from a designated ENTRANCE vertex to a designated EXIT vertex exponentially faster than the classical random walk.

We define a sequence of graphs G_n, where $n = 1, 2, 3, \ldots$. The number of vertices in G_n is $2^{n+1} + 2^n - 2$. In Figure 5-1 we show G_4. In general, G_n consists of two balanced binary trees of depth n with the 2^n leaves of the two trees pairwise identified. We will refer to the root of the left tree as the ENTRANCE, and to the root of the right tree as the EXIT.

For both the classical random walk and the quantum walk, we start at the ENTRANCE and want the probability as a function of time of being at the EXIT. In other words, we are interested in how long it takes to propagate from the ENTRANCE to the EXIT as a function of n.

Consider the classical case first. The vertices of G_n can be grouped in columns indexed by $j \in \{0, 1, \ldots, 2n\}$. Column 0 contains the ENTRANCE, column 1 contains the two vertices connected to the ENTRANCE, etc. Note that column n contains the 2^n vertices in the middle of the graph and column $2n$ is the EXIT.

To analyze the classical walk from the ENTRANCE to the EXIT, we need only keep track of the probabilities of being in the columns. In the left tree, for $0 < j < n$, the probability of stepping from column j to column $j + 1$ is twice as great as the probability of stepping from column j to column $j - 1$. However, in the right tree, for $n < j < 2n$, the probability of stepping from column j to column $j + 1$ is half as great as the probability of stepping from column j to column $j - 1$. This means that if you start at the ENTRANCE, you quickly move to the middle of the graph, but then it takes a time exponential in n to reach the

EXIT. More precisely, starting at the ENTRANCE, the probability of being at the EXIT after any number of steps is less than 2^{-n}. This implies that the probability of reaching the EXIT in a time that is polynomial in n must be exponentially small as a function of n.

We now analyze the quantum walk on G_n starting in the state corresponding to the ENTRANCE and evolving with the Hamiltonian given by $H = -\gamma L$. With this initial state, the symmetries of H keep the evolution in a $(2n+1)$-dimensional subspace of the $(2^{n+1}+2^n-2)$-dimensional Hilbert space. This subspace is spanned by states $|\mathrm{col}\, j\rangle$ (where $0 \le j \le 2n$), the uniform superposition over all vertices in column j, that is,

$$|\mathrm{col}\, j\rangle = \frac{1}{\sqrt{N_j}} \sum_{a \in \mathrm{column}\ j} |a\rangle\,, \tag{5.1}$$

where

$$N_j = \begin{cases} 2^j & 0 \le j \le n \\ 2^{2n-j} & n \le j \le 2n \end{cases} \tag{5.2}$$

is the number of vertices in column j. We refer to this subspace as the *column subspace.* In the column subspace, the non-zero matrix elements of H are

$$\langle \mathrm{col}\, j|H|(\mathrm{col}\, j \pm 1)\rangle = -\sqrt{2}\gamma \tag{5.3}$$

$$\langle \mathrm{col}\, j|H|\mathrm{col}\, j\rangle = \begin{cases} 2\gamma & j = 0, n, 2n \\ 3\gamma & \text{otherwise,} \end{cases} \tag{5.4}$$

which is depicted in Figure 5-2(a) (for $n = 4$) as a quantum walk on a line with $2n+1$ vertices.

Starting at the ENTRANCE in Figure 5-2(a), there is an appreciable probability of being at the EXIT after a time proportional to n. To see this, we will consider a number of examples that show the essential features of the walk. When we give an example of algorithmic speedup in the next section using a closely related graph, we will rigorously prove that the walk reaches the EXIT in time polynomial in n.

First consider quantum propagation on an infinite, translationally invariant line of vertices as depicted in Figure 5-2(b). This is simply a rescaled version of the quantum walk on the line discussed in Section 3.3.2. In this case, the energy for the momentum eigenstate $|p\rangle$ in (3.10) is given by

$$E(p) = 3 - 2\sqrt{2}\gamma \cos p\,. \tag{5.5}$$

Thus we find for the propagator between vertices j and k

$$G(j,k,t) = \langle k|e^{-iHt}|j\rangle \tag{5.6}$$

$$= e^{-i3\gamma t} i^{k-j} J_{k-j}(2\sqrt{2}\gamma t) \tag{5.7}$$

(cf. (3.14)). This corresponds to propagation with speed $2\sqrt{2}\gamma$.

To understand the effect of a defect, we now consider an infinite line with a different matrix element at site $j = n$, as shown in 5-2(c). For generality, we consider the case where the matrix element at the defect is α. We use standard scattering theory to calculate the transmission coefficient for an incident plane wave of momentum $p > 0$. Consider a state $|\psi\rangle$ consisting of an incident plane wave and a reflected plane wave on the left side of the

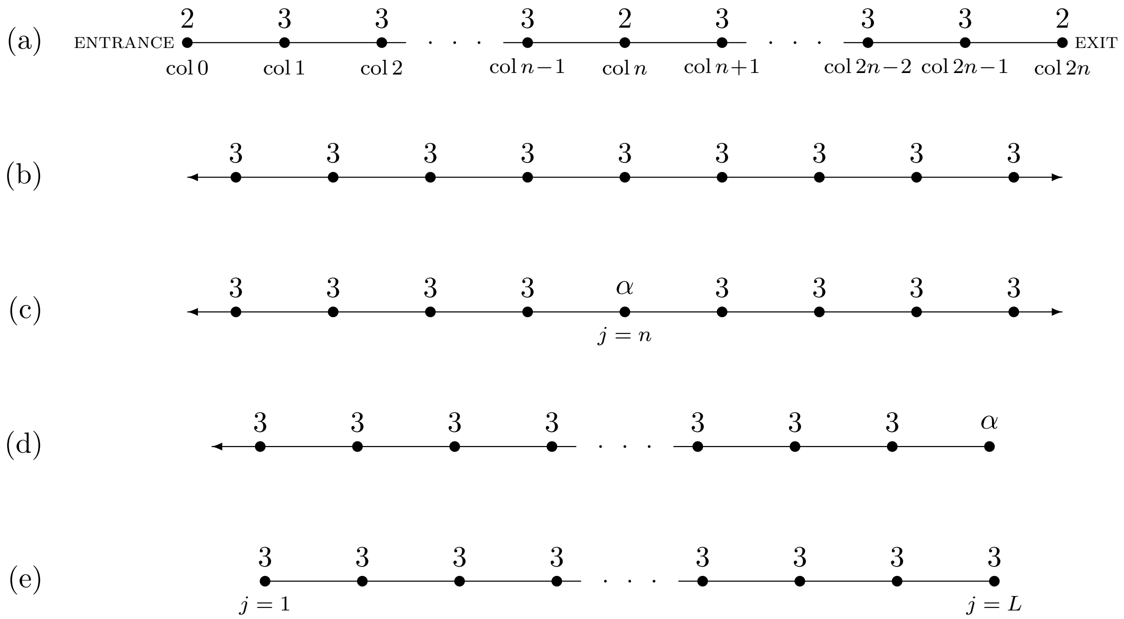

Figure 5-2: Quantum walks on lines, with matrix elements shown in units of γ. Each edge represents a matrix element of $-\sqrt{2}$ connecting two sites. (a) Reduction of the quantum walk on G_n to a quantum walk on a line. (b) Quantum walk on an infinite, translationally invariant line. (c) Quantum walk on an infinite line with a defect. (d) Quantum walk on a half-line with a defect at the end. (e) Quantum walk on a finite line with no defects.

defect and a transmitted plane wave on the right side:

$$\langle j|\psi\rangle = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{ipj} + \frac{\mathcal{R}}{\sqrt{2\pi}} e^{-ipj} & j \le n \\ \frac{\mathcal{T}}{\sqrt{2\pi}} e^{ipj} & j \ge n+1\,. \end{cases} \tag{5.8}$$

Here $\mathcal{R}$ is the reflection coefficient and $\mathcal{T}$ is the transmission coefficient. Requiring this to be an eigenstate of the Hamiltonian, we find

$$\mathcal{T}(p) = \frac{4\sin p}{4\sin p - i\sqrt{2}(3-\alpha)} \tag{5.9}$$

which gives

$$|\mathcal{T}(p)|^2 = \frac{8\sin^2 p}{1+8\sin^2 p} \tag{5.10}$$

for $\alpha = 2$, as shown in Figure 5-3(a). A narrow wave packet with momentum p propagates through the defect with probability $|\mathcal{T}(p)|^2$. The wave packet that results from a state initially localized at a particular site is spread out over a range of momenta, but since the transmission probability is appreciable over most of the range, it is clear that there will be substantial transmission through the defect.

To understand the effect of boundaries, now consider the half-line shown in Figure 5-2(d). A state $|\psi\rangle$ consisting of an incoming plane wave together with a reflected wave takes

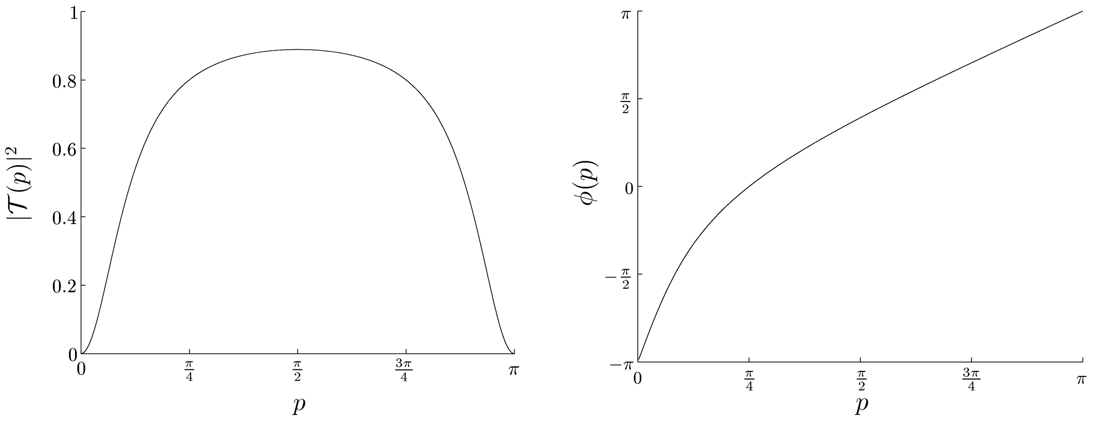

Figure 5-3: (a) The transmission probability $|\mathcal{T}(p)|^2$ as a function of momentum for the infinite line with a defect from Figure 5-2(c) (with $\alpha = 2$). (b) The phase shift $\phi(p)$ induced by reflection off the end of the half line from Figure 5-2(d) (again with $\alpha = 2$).

the form

$$\langle j|\psi\rangle = \frac{1}{\sqrt{2\pi}} e^{ipj} + \frac{\mathcal{R}}{\sqrt{2\pi}} e^{-ipj} \,. \tag{5.11}$$

Demanding that this state is an eigenstate of the Hamiltonian, we find

$$\mathcal{R}(p) = -\frac{\alpha - 3 + \sqrt{2}e^{ip}}{\alpha - 3 + \sqrt{2}e^{-ip}} \,. \tag{5.12}$$

By unitarity, $|\mathcal{R}(p)| = 1$, so we can write $\mathcal{R}(p) = e^{i\phi(p)}$. An incoming plane wave of momentum p is reflected from the end of the line with a momentum-dependent phase shift $\phi(p)$, as shown in Figure 5-3(b) for $\alpha = 2$. In the simple case $\alpha = 3$, we have a reflecting boundary condition, and we find $\mathcal{R}(p) = -e^{2ip}$. In this case, it is as if the incoming wave reflects off a site one unit to the right of the end of the half line, regardless of its momentum. The case of general α is similar, but the reflection is slightly more complicated due to the momentum dependence of the phase shift.

Finally, to understand the effect of having a finite system, consider the finite line (with no defects) shown in Figure 5-2(e). This problem can be treated by standard techniques of multiple reflection. The exact Green's function $\tilde{G}(j,k,t)$ for this finite line in terms of the Green's function $G(j,k,t)$ for the infinite line is

$$\tilde{G}(j,k,t) = \sum_{l=-\infty}^{\infty} \left[G(j,k+2l(L+1),t) - G(j,-k+2l(L+1),t)\right] \,, \quad 1 \le j,k \le L \,. \tag{5.13}$$

This can be interpreted as a sum of paths making reflections off the boundaries, as in the previous example. To verify this formula, one can check that it satisfies the Schrödinger equation, the boundary conditions, and the initial condition. The Schrödinger equation is satisfied because each term individually satisfies it for the infinite line. The boundary conditions $\tilde{G}(j,0,t) = \tilde{G}(j,L{+}1,t) = 0$ can be verified by substitution. The initial condition $\tilde{G}(j,k,0) = \delta_{jk}$ holds because $G(j,k,0) = \delta_{jk}$, and the only contribution at $t = 0$ comes from the first term of (5.13) with $l = 0$. We can now see, using the particular form of the propagator in terms of a Bessel function, that propagation from $j = 1$ to $j = L$ takes time proportional to L for $L \gg 1$. Since $J_j(t)$ is small for $|j| \gg t$, there are only four terms that contribute significantly for $t \gtrsim L/(2\sqrt{2}\gamma)$. They result from taking $l = 0, -1$ in the first term of (5.13) and $l = 0, 1$ in the second term. The resulting expression is

$$\tilde{G}(1, L, t) \approx G(1, L, t) + G(1, -L-2, t) - G(1, -L, t) - G(1, L+2, t)\,, \quad t \gtrsim L/(2\sqrt{2}\gamma)\,, \tag{5.14}$$

the magnitude of which is not small.

We have seen that propagation on a line occurs as a wave packet that moves with constant speed, that the wave packet is substantially transmitted through a defect, and that reflections off boundaries do not impede propagation. Taken together, these facts constitute compelling evidence that the quantum walk traverses the graph G_n in linear time. To verify this, we can numerically compute the probability $|\langle \mathrm{col}\, j|\psi(t)\rangle|^2$ of being in column j at various times t, where $|\psi(0)\rangle = |\textsc{entrance}\rangle$ and we choose $\gamma = 1$. This is shown in Figure 5-4 with $n = 500$ for $t = 100$, 250, and 400. These plots clearly show a wave packet that propagates with speed $2\sqrt{2}$, with the amplitude near the wavefront decreasing like $t^{-1/2}$. In the first plot, at $t = 100$, the leading edge of the distribution is at column $200\sqrt{2} \approx 283$. The packet has not yet encountered the small defect at the center, so it has a relatively simple shape. At $t = 250$, the wavefront has passed the center, and a small reflection can be seen propagating backward. However, the leading edge is relatively undisturbed, having propagated to column $500\sqrt{2} \approx 707$. The wavefront continues to propagate with speed $2\sqrt{2}$ until it reaches the EXIT, where the packet is reflected. The last plot, at $t = 400$, shows the distribution shortly after this first reflection. Even after the reflection, there is still an appreciable probability of being at the EXIT.

Note that if we had chosen $H = \gamma A$ instead of $H = -\gamma L$ as the Hamiltonian of our quantum walk, the reduced walk on the line would have been free of defects, and we could have calculated its propagation exactly, just as we did for the example of Figure 5-2(e). We chose to use the Laplacian since this choice is more closely analogous to the classical case. However, when we demonstrate algorithmic speedup in the next section, we will be free to choose $H = \gamma A$ to simplify the calculation.

5.3 Algorithmic speedup

In the previous section, we gave a simple example of exponential speedup of a quantum walk over a classical random walk. In this section, we modify the graphs in the previous example so that the quantum walk is exponentially better not only than the corresponding classical random walk, but also than *any classical algorithm* one can design to traverse the graph.

5.3.1 The problem

We begin by describing the computational problem in detail. The problem involves determining a property of a graph (a variant of G_n) whose structure is given in the form of black box. Before we describe the modified graphs, we provide a black box framework in which graphs can be specified, and where our notion of an algorithm traversing a graph can be made precise.

Let G be a graph with N vertices. To represent G as a black box, let m be such that $2^m > N$ and let k be at least as large as the maximum vertex degree in G. Assign each vertex $a \in G$ a distinct m-bit string as its name, except do not assign $11\ldots1$ as the name

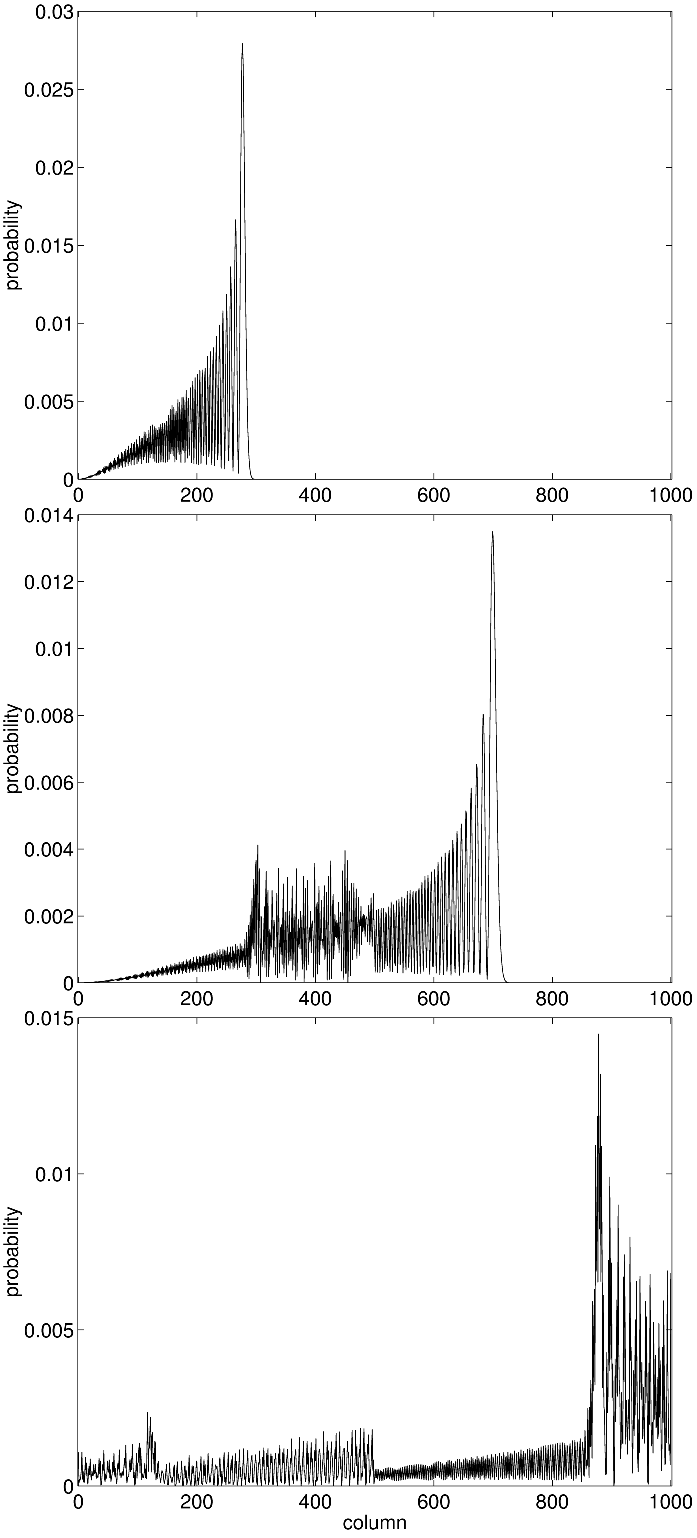

Figure 5-4: Propagation in G_{500} starting at the left root. From top to bottom, the times are $t = 100$, 250, and 400.

of any vertex. For each vertex $a \in G$, assign the outgoing edges of a labels from a set L of size k. For $a \in \{0,1\}^m$ and $c \in L$, define $v_c(a)$ as the adjacent vertex reached by following the outgoing edge of a labeled by c, if such an edge exists. If no such edge exists or if $a \notin G$, then $v_c(a) = 11\ldots1$. The resulting black box for G takes $c \in L$ and $a \in \{0,1\}^m$ as input and returns the value of $v_c(a)$. For quantum algorithms, this operation is defined as a unitary transformation U in the usual way. That is, for $a, b \in \{0,1\}^m$ and $c \in L$,

$$U|c, a, b\rangle = |c, a, b \oplus v_c(a)\rangle\,, \tag{5.15}$$

where $\oplus$ denotes bitwise addition modulo 2.

We now define a notion of traversing a graph G from its ENTRANCE to its EXIT:

Definition 5.1. *Let G be a graph and* ENTRANCE *and* EXIT *be two vertices of G. The input of the* traversal problem *is a black box for G and the name of the* ENTRANCE*. The output is the name of the* EXIT*.*

For the graphs G_n, with the ENTRANCE and EXIT defined as before, this instance of the traversal problem can be solved in time polynomial in n using a classical algorithm that is *not* a random walk. The key is that we can always tell whether a particular vertex is in the central column by checking its degree. We begin at the ENTRANCE. At each vertex, we query the oracle and move to one of the two unvisited adjacent vertices. After n steps, we reach the central column and then proceed moving to unvisited adjacent vertices in the right tree, checking the degree at each step. If we discover that after s steps we are again at a central vertex, we know that the wrong move happened after $s/2$ steps (s can only be even due to the structure of G_n) and we can backtrack to that point and take the other edge. After only $O(n^2)$ steps this procedure will reach the EXIT.[1]

We now describe how the graphs G_n are modified so that they cannot be traversed efficiently by any classical algorithm. We choose a graph G'_n at random from a particular distribution on graphs. A typical graph G'_n is shown in Figure 5-5 (for $n = 4$). The distribution is defined as follows. The graph again consists of two balanced binary trees of height n, but instead of identifying the leaves, they are connected by a random cycle that alternates between the leaves of the two trees. In other words, we choose a leaf on the left at random and connect it to a leaf on the right chosen at random. Then we connect the latter to a leaf on the left chosen randomly among the remaining ones. We continue with this procedure, alternating sides, until every leaf on the left is connected to two leaves on the right (and vice versa).

In Sections 5.3.2 and 5.3.3, we describe and analyze a quantum algorithm that solves the graph traversal problem for G'_n in polynomial time, and in Section 5.3.4, we show that any classical algorithm that solves this traversal problem with nonnegligible probability takes exponential time.

[1] As an aside, note that there is also a polynomial-time classical traversal algorithm for the n-dimensional hypercube (where one vertex is designated as the ENTRANCE and the EXIT is the unique vertex whose distance from ENTRANCE is n). For a description of the algorithm, see Appendix A of [40]. This means that there can be no exponential algorithmic speedup using quantum walks to traverse the hypercube, even though both continuous- and discrete-time quantum walks reach the EXIT in a polynomial number of steps (see Section 3.3.1 and [148, 116]).

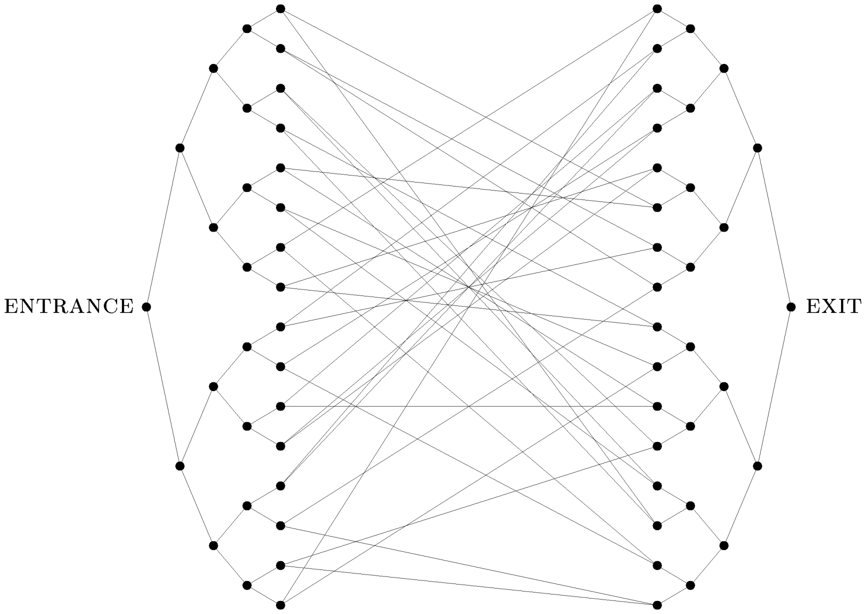

Figure 5-5: A typical graph G'_4.

5.3.2 Quantum walk algorithm

The traversal problem for any graph G'_n can be easily solved by a quantum walk. Just as for G_n, a quantum walk on G'_n starting from ENTRANCE rapidly traverses the graph to the EXIT. Given the black box (5.15) for computing the neighbors of a given vertex, this walk can be implemented efficiently in the quantum circuit model using Rule 1.7. To simplify the analysis, we define the walk using the adjacency matrix of the graph.

Similar arguments to the ones we gave in Section 5.2 show that the walk propagates from the ENTRANCE to the EXIT in linear time. In the remainder of this section, we explain how the walk on G'_n can be viewed as a walk on a finite line with a defect at the center, and we argue as before that the defect and the boundaries do not significantly affect the walk. In Section 5.3.3, we prove that the walk reaches the EXIT in polynomial time.

Just as the walk on G_n could be reduced to a walk on a line with $2n+1$ vertices, one for each column of the original graph, the walk on G'_n can be reduced to a walk on a line with $2n+2$ vertices. Consider the $(2n+2)$-dimensional column subspace spanned by the column states (5.1) for $j \in \{0, 1, \dots, 2n+1\}$, where now

$$N_j = \begin{cases} 2^j & 0 \le j \le n \\ 2^{2n+1-j} & n+1 \le j \le 2n+1\,. \end{cases} \tag{5.16}$$

Because every vertex in column j is connected to the same number of vertices in column $j+1$ and every vertex in column $j+1$ is connected to the same number of vertices in column j, applying H to any state in the column subspace results in another state in this subspace. Despite the random connections in G'_n, the column subspace is invariant under H. In particular, a quantum walk starting in the state corresponding to the ENTRANCE always remains in the column subspace. Thus, to understand the quantum walk starting from the ENTRANCE, we only need to understand how the Hamiltonian acts on the column

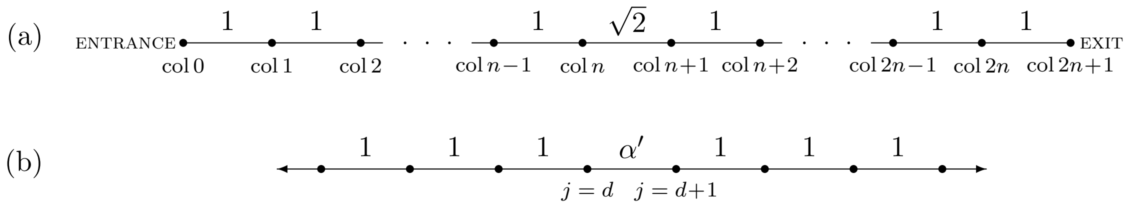

Figure 5-6: (a) Reduction of the quantum walk on G'_n to a quantum walk on a line. (b) Quantum walk on an infinite line with a defect between sites n and $n+1$.

subspace. In this subspace, the non-zero matrix elements of $H = \gamma A$ are

$$\langle \mathrm{col}\, j|H|\mathrm{col}(j+1)\rangle = \begin{cases} \sqrt{2}\gamma & 0 \le j \le n-1\,, \quad n+1 \le j \le 2n \\ 2\gamma & j = n \end{cases} \tag{5.17}$$

(and those deduced by Hermiticity of H). For simplicity, we set $\gamma = 1/\sqrt{2}$. The quantum walk in the column subspace is shown pictorially in Figure 5-6(a).

We claim that if the quantum state at $t = 0$ is $|\mathrm{col}\,0\rangle = |\textsc{entrance}\rangle$, then at a time of order $n/2$, there is an appreciable probability of being at $|\mathrm{col}(2n+1)\rangle = |\textsc{exit}\rangle$. By considering examples similar to those in Figure 5-2, it is clear that the walk propagates on the line with speed 2. In this case, since the diagonal elements are zero, we do not need to worry about defects at the ends of the line. However, since we have added edges in the middle of the graph, the reduction of G'_n contains a defect between sites $j = n$ and $j = n+1$. Therefore, analogous to Figure 5-2(c), we should consider an infinite line with a defect on one edge, as shown in Figure 5-6(b). For this problem, we can calculate a transmission coefficient just as we did in the previous section. If we consider a state $|\psi\rangle$ of the form

$$\langle j|\psi\rangle = \begin{cases} \frac{1}{\sqrt{2\pi}}e^{ipj} + \frac{\mathcal{R}}{\sqrt{2\pi}}e^{-ipj} & j \le d \\ \frac{\mathcal{T}}{\sqrt{2\pi}}e^{ipj} & j \ge d+1\,, \end{cases} \tag{5.18}$$

then we find a transmission coefficient

$$\mathcal{T}(p) = \frac{2i\alpha' \sin p}{(\alpha'^2 - 1)\cos p + i(\alpha'^2 + 1)\sin p}\,. \tag{5.19}$$

Coincidentally, even though the reduction of G'_n (using the adjacency matrix) has a different kind of defect from the reduction of G_n (using the Laplacian), for $\alpha' = \sqrt{2}$ we find the same transmission probability $|\mathcal{T}(p)|^2$ given in (5.10), as shown in Figure 5-3(a). As in the previous case, we see that the defect is not a substantial barrier to transmission.

Again, using simple arguments based on standard scattering theory, we see that a quantum walk traverses the graph G'_n in linear time. The exact propagator for the line shown in Figure 5-6(a) can be calculated using a more sophisticated version of these techniques [94]. This exact propagator, evaluated for t near n, is of order $n^{-1/3}$. We give a simpler proof of a bound that is not tight—but is nevertheless polynomial—in the following section.

5.3.3 Upper bound on the traversal time

Although the preceding section demonstrated beyond any reasonable doubt that the quantum walk traverses the graph G'_n in linear time, we now provide a simple proof that the traversal time is upper bounded by a polynomial.

For the purpose of this proof, it will be more convenient to consider the graph G'_{n-1}, which reduces to a line with $2n$ vertices. We label the vertices from 1 to $2n$, and the defect is on the edge between vertices n and $n+1$. With this labeling and $\gamma = 1/\sqrt{2}$, the Hamiltonian (5.17) is

$$\langle \mathrm{col}\, j|H|\mathrm{col}(j+1)\rangle = \begin{cases} 1 & 1 \le j \le n-1\,, \quad n+1 \le j \le 2n-1 \\ \sqrt{2} & j = n\,, \end{cases} \tag{5.20}$$

with Hermiticity of H giving the other nonzero matrix elements.

Define a reflection operator

$$R|\mathrm{col}\, j\rangle = |\mathrm{col}(2n+1-j)\rangle\,. \tag{5.21}$$

Note that $R^2 = 1$, so R has eigenvalues ± 1. R commutes with H on the column subspace, so we can find simultaneous eigenstates of R and H. These are of the form

$$\langle \mathrm{col}\, j|E\rangle = \begin{cases} \sin pj & 1 \le j \le n \\ \pm \sin(p(2n+1-j)) & n+1 \le j \le 2n\,, \end{cases} \tag{5.22}$$

which explicitly vanish at $j = 0$ and $j = 2n+1$. The eigenvalue corresponding to the eigenstate $|E\rangle$ is $E = 2\cos p$, and the quantization condition (to be discussed later) comes from matching at vertices n and $n+1$. The ENTRANCE vertex corresponds to $|\mathrm{col}\, 1\rangle$ and the EXIT vertex to $|\mathrm{col}\, 2n\rangle$.

Lemma 5.1. *Consider the quantum walk in G'_{n-1} starting at the* ENTRANCE. *Let the walk run for a time t chosen uniformly in $[0, \tau]$ and then measure in the computational basis. If $\tau \ge \frac{4n}{\epsilon \Delta E}$ for any constant $\epsilon > 0$, where ΔE is the magnitude of the smallest gap between any pair of eigenvalues of the Hamiltonian, then the probability of finding the* EXIT *is greater than $\frac{1}{2n}(1-\epsilon)$.*

Proof. The probability of finding the EXIT after the randomly chosen time $t \in [0, \tau]$ is

$$\begin{aligned} &\frac{1}{\tau}\int_0^\tau \mathrm{d}t\, |\langle \mathrm{col}\, 2n|e^{-iHt}|\mathrm{col}\, 1\rangle|^2 \\ &\quad = \frac{1}{\tau}\sum_{E,E'}\int_0^\tau \mathrm{d}t\, e^{-i(E-E')t}\langle \mathrm{col}\, 2n|E\rangle\langle E|\mathrm{col}\, 1\rangle\langle \mathrm{col}\, 1|E'\rangle\langle E'|\mathrm{col}\, 2n\rangle && (5.23) \\ &\quad = \sum_E |\langle E|\mathrm{col}\, 1\rangle|^2 |\langle E|\mathrm{col}\, 2n\rangle|^2 \\ &\qquad + \sum_{E\ne E'} \frac{1-e^{-i(E-E')\tau}}{i(E-E')\tau}\langle \mathrm{col}\, 2n|E\rangle\langle E|\mathrm{col}\, 1\rangle\langle \mathrm{col}\, 1|E'\rangle\langle E'|\mathrm{col}\, 2n\rangle\,. && (5.24) \end{aligned}$$

Because of (5.22), we have $\langle E|\mathrm{col}\,1\rangle = \pm\langle E|\mathrm{col}\,2n\rangle$. Thus the first term is

$$\sum_E |\langle E|\mathrm{col}\,1\rangle|^4 \geq \frac{1}{2n} \tag{5.25}$$

as is easily established using the Cauchy-Schwartz inequality. The second term can be bounded as follows:

$$\begin{aligned}\left|\sum_{E\neq E'} \frac{1-e^{-i(E-E')\tau}}{i(E-E')\tau}\langle \mathrm{col}\,2n|E\rangle\langle E|\mathrm{col}\,1\rangle\langle \mathrm{col}\,1|E'\rangle\langle E'|\mathrm{col}\,2n\rangle\right| \\ \leq \frac{2}{\tau\Delta E}\sum_{E,E'}|\langle E|\mathrm{col}\,1\rangle|^2|\langle E'|\mathrm{col}\,2n\rangle|^2 = \frac{2}{\tau\Delta E}\,.\end{aligned} \tag{5.26}$$

Thus we have

$$\frac{1}{\tau}\int_0^\tau \mathrm{d}t\,|\langle \mathrm{col}\,2n|e^{-iHt}|\mathrm{col}\,1\rangle|^2 \geq \frac{1}{2n} - \frac{2}{\tau\Delta E} \geq \frac{1}{2n}(1-\epsilon) \tag{5.27}$$

where the last inequality follows since $\tau \geq \frac{4n}{\epsilon\Delta E}$ by assumption. □

Now we need to prove that the minimum gap ΔE is only polynomially small.

Lemma 5.2. *The smallest gap between any pair of eigenvalues of the Hamiltonian satisfies*

$$\Delta E > \frac{2\pi^2}{(1+\sqrt{2})n^3} + O(1/n^4)\,. \tag{5.28}$$

Proof. To evaluate the spacings between eigenvalues, we need to use the quantization condition. We have

$$\langle \mathrm{col}\,n|H|E\rangle = 2\cos p\,\langle \mathrm{col}\,n|E\rangle \tag{5.29}$$

so that

$$\sqrt{2}\langle \mathrm{col}(n+1)|E\rangle + \langle \mathrm{col}(n-1)|E\rangle = 2\cos p\,\langle \mathrm{col}\,n|E\rangle \tag{5.30}$$

and using (5.22), we have

$$\pm\sqrt{2}\sin np + \sin((n-1)p) = 2\cos p\,\sin np \tag{5.31}$$

which simplifies to

$$\frac{\sin((n+1)p)}{\sin np} = \pm\sqrt{2}\,. \tag{5.32}$$

In Figure 5-7 we plot the left hand side of (5.32) for $n=5$. The intersections with $-\sqrt{2}$ occur to the left of the zeros of $\sin np$, which occur at $\pi l/n$ for $l = 1, 2, \ldots, n-1$. For the values of p that intersect $-\sqrt{2}$, we can write $p = (\pi l/n) - \delta$. Equation (5.32) with $-\sqrt{2}$ on the right hand side is now

$$-\sqrt{2}\sin n\delta = \sin\left(n\delta - \frac{l\pi}{n} + \delta\right)\,. \tag{5.33}$$

Write $\delta = (c/n) + (d/n^2) + O(1/n^3)$. Taking $n\to\infty$ in (5.33) gives $-\sqrt{2}\sin c = \sin c$, which

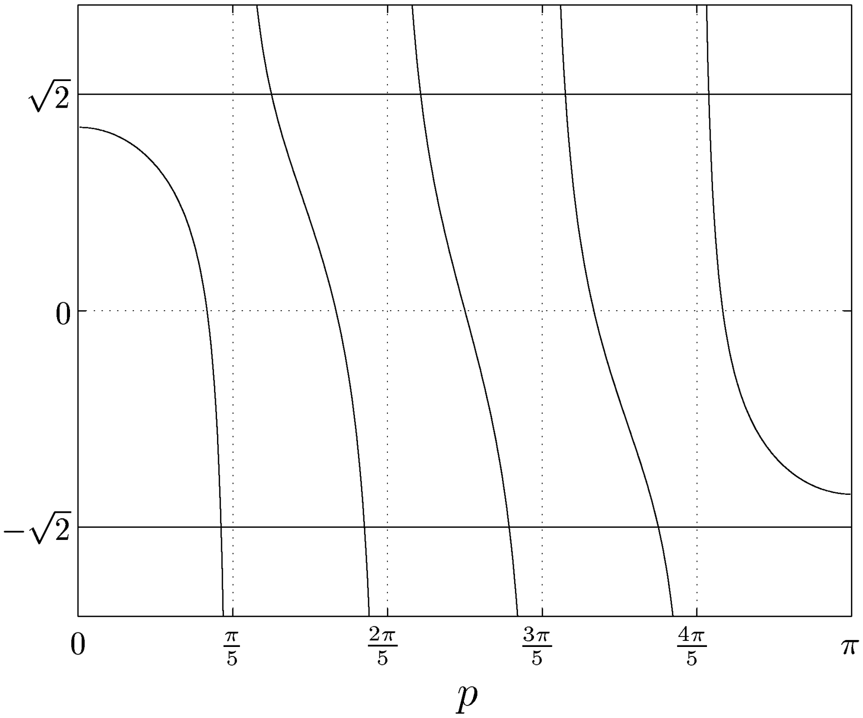

Figure 5-7: Left hand side of (5.32) for $n = 5$.

implies that $c = 0$. We then get

$$-\sqrt{2}\sin\left(\frac{d}{n} + O(1/n^2)\right) = \sin\left(\frac{d}{n} - \frac{l\pi}{n} + O(1/n^2)\right) \tag{5.34}$$

which gives, as $n \to \infty$,

$$d = \frac{l\pi}{1+\sqrt{2}}\,. \tag{5.35}$$

Thus we have that the roots of (5.32) with $-\sqrt{2}$ on the right hand side are of the form

$$p = \frac{l\pi}{n} - \frac{l\pi}{(1+\sqrt{2})n^2} + O(1/n^3)\,. \tag{5.36}$$

Let p' and p'' be the two roots of (5.32) closest to the root p just found, with $p' < p < p''$. From the figure we see that p' and p'' both are roots of (5.32) with $+\sqrt{2}$. (Note that the smallest p, corresponding to $l = 1$, does not have a p'.) We see that p'' lies to the right of the zero of $\sin np$ at $p = l\pi/n$. We also see that p' lies to the left of the zero of $\sin((n+1)p)$ at $l\pi/(n+1)$. Therefore we have

$$p' < \frac{l\pi}{n} - \frac{l\pi}{n^2} + O(1/n^3) \tag{5.37}$$

$$p'' > \frac{l\pi}{n}\,, \tag{5.38}$$

from which we conclude that

$$p - p' > \frac{l\pi\sqrt{2}}{(1+\sqrt{2})n^2} + O(1/n^3)\,, \quad l = 2, 3, \ldots, n-1 \tag{5.39}$$

$$p'' - p > \frac{l\pi}{(1+\sqrt{2})n^2} + O(1/n^3)\,, \quad l = 1, 2, \ldots, n-1\,. \tag{5.40}$$

Thus the smallest spacing is at least $\pi/[(1+\sqrt{2})n^2] + O(1/n^3)$.

Now for a given p, the corresponding eigenvalue is $2\cos p$. For small Δp, the spacing ΔE is related to the spacing Δp by

$$\Delta E = 2|\Delta p \, \sin p| + O\left((\Delta p)^2\right)\,. \tag{5.41}$$

The factor $\sin p = \sin(l\pi/n + O(1/n^2))$ is smallest when $l = 1$, so we have

$$\Delta E > \frac{2\pi^2}{(1+\sqrt{2})n^3} + O(1/n^4) > \frac{8}{n^3} \quad \text{for } n \text{ sufficiently large.} \tag{5.42}$$

The alert reader will note from the figure with $n = 5$ that there are only 8 roots, whereas the dimension of the reduced space is 10, so there are actually 10 eigenvalues. In general, there are $n-2$ roots of (5.32) with p real. If we let $p = ik$ with k real, we can have eigenstates of the form (5.22) with $\sin pj$ replaced by $\sinh kj$ and $\pm\sin(p(2n+1-j))$ replaced by $\pm\sinh(k(2n+1-j))$. The corresponding eigenvalue is $2\cosh k$ and the condition (5.32) becomes

$$\frac{\sinh((n+1)k)}{\sinh nk} = \pm\sqrt{2}. \tag{5.43}$$

As $n \to \infty$, the root of this equation is at $e^k = \sqrt{2}$, which corresponds to an eigenvalue $\sqrt{2} + \frac{1}{\sqrt{2}}$. To obtain the last eigenvalue, let $p = \pi + ik$. The eigenvalue is then $-2\cosh k$. The quantization condition is now the same as (5.43), and as $n \to \infty$, the eigenvalue is $-(\sqrt{2} + \frac{1}{\sqrt{2}})$. So we have found two eigenvalues at $\pm(\sqrt{2} + \frac{1}{\sqrt{2}})$ with corrections that vanish exponentially as $n \to \infty$. Since the other $n-2$ eigenvalues are all in the range $[-2, 2]$, our conclusion about the minimum spacing is unchanged. □

Using Lemma 5.1 and Lemma 5.2, we find

Theorem 5.3. *For n sufficiently large, running the quantum walk for a time chosen uniformly in $[0, \frac{n^4}{2\epsilon}]$ and then measuring in the computational basis yields a probability of finding the* EXIT *that is greater than $\frac{1}{2n}(1-\epsilon)$.*

To summarize, we have presented an efficient algorithm for traversing any graph G'_n using a quantum computer. The computer is prepared in the state corresponding to the ENTRANCE, and the quantum walk is simulated using Rule 1.7. After running the walk for a certain time t, the state of the computer is measured in the computational basis. The oracle can then be used to check whether the resulting vertex name corresponds to a vertex of degree 2 other than ENTRANCE, in which case it must be EXIT. Theorem 5.3 shows that by choosing an appropriate $t = \text{poly}(n)$, the probability of finding the name of the EXIT can be $O(1/n)$. By repeating this process poly(n) times, the success probability can be made arbitrarily close to 1. Combining this with the efficient implementation of the quantum walk described in Chapter 1, we see that the quantum walk algorithm finds the name of the EXIT with high probability using poly(n) calls to the oracle.

5.3.4 Classical lower bound

In this section, we show that any classical algorithm that solves the problem of traversing G'_n requires exponential time. More precisely, we have

Theorem 5.4. *Any classical algorithm that makes at most $2^{n/6}$ queries to the oracle finds the* EXIT *with probability at most $4 \cdot 2^{-n/6}$.*

We shall prove Theorem 5.4 by considering a series of games and proving relations between them. The first game is essentially equivalent to our problem, and each new game will be essentially as easy to win. Finally, we will show that the easiest game cannot be won in subexponential time.

Our problem is equivalent to the following game:

Game 1. *The oracle contains a random set of names for the vertices of the randomly chosen graph G'_n such that each vertex has a distinct $2n$-bit string as its name and the* ENTRANCE *vertex has the name 0. At each step, the algorithm sends a $2n$-bit string to the oracle, and if there exists a vertex with that name, the oracle returns the names of the neighbors of that vertex. The algorithm wins if it ever sends the oracle the name of the* EXIT *vertex.*

Note that there are two sources of randomness in this oracle: in the choice of a graph G'_n and in the random naming of its vertices. We first consider a fixed graph G and only draw implications from the random names. Throughout this section, G always refers to one of the graphs G'_n. For a game X with a graph G, the success probability of the algorithm A is defined as

$$\mathcal{P}_X^G(A) = \Pr_{\text{names}} [A \text{ wins game } X \text{ on graph } G] , \tag{5.44}$$

where $\Pr\limits_{\text{names}} [\cdot]$ means the probability is taken over the random naming of vertices.

In Game 1, the algorithm could traverse a disconnected subgraph of G'_n. But because there are exponentially many more strings of $2n$ bits than vertices in the graph, it is highly unlikely that any algorithm will ever guess the name of a vertex that it was not sent by the oracle. Thus, Game 1 is essentially equivalent to the following game:

Game 2. *The oracle contains a graph and a set of vertex names as described in Game 1. At each step, the algorithm sends the oracle the name of the* ENTRANCE *vertex or the name of a vertex it has previously been sent by the oracle. The oracle then returns the names of the neighbors of that vertex. The algorithm wins it ever sends the oracle the name of the* EXIT *vertex.*

The next lemma shows that, if the algorithms run for a sufficiently short time, then the success probabilities for Game 1 and Game 2 can only differ by a small amount.

Lemma 5.5. *For every algorithm A for Game 1 that makes at most t oracle queries, there exists an algorithm A' for Game 2 that also makes at most t oracle queries such that for all graphs G,*

$$\mathcal{P}_1^G(A) \le \mathcal{P}_2^G(A') + O(t/2^n) . \tag{5.45}$$

Proof. Algorithm A' simulates A, but whenever A queries a name it has not previously been sent by the oracle, A' assumes the result of the query is $11\ldots1$. The chance that A can discover the name of a vertex that it is not told by the oracle is at most $t(2^{n+2}-2)/(2^{2n}-1)$, and unless this happens, the two algorithms will have similar behavior. □

To obtain a bound on the success probability of Game 2, we will compare it with a simpler game, which is the same except that it provides an additional way to win:

Game 3. *The oracle contains a graph and a set of vertex names as described in Game 1. At each step, the algorithm and the oracle interact as in Game 2. The algorithm wins it ever sends the oracle the name of the* EXIT *vertex, or if the subgraph it has seen contains a cycle.*

Game 3 is clearly easier to win than Game 2, so we have

Lemma 5.6. *For all algorithms A for Game 2,*

$$\mathcal{P}_2^G(A) \le \mathcal{P}_3^G(A)\,. \tag{5.46}$$

Now we further restrict the form of the subgraph that can be seen by the algorithm unless it wins Game 3. We will show that the subgraph an algorithm sees must be a random embedding of a rooted binary tree. For a rooted binary tree T, we define an embedding of T into G to be a function π from the vertices of T to the vertices of G such that $\pi(\textsc{root}) = \textsc{entrance}$ and for all vertices u and v that are neighbors in T, $\pi(u)$ and $\pi(v)$ are neighbors in G. We say that an embedding of T is *proper* if $\pi(u) \neq \pi(v)$ for $u \neq v$. We say that a tree T *exits* under an embedding π if $\pi(v) = \textsc{exit}$ for some $v \in T$.

We must specify what we mean by a random embedding of a tree. Intuitively, a random embedding of a tree is obtained by setting $\pi(\textsc{root}) = \textsc{entrance}$ and then mapping the rest of T into G at random. We define this formally for trees T in which each internal vertex has two children (it will not be necessary to consider others). A random embedding is obtained as follows:

1. Label the ROOT of T as 0, and label the other vertices of T with consecutive integers so that if vertex i lies on the path from the root to vertex j then $i < j$.

2. Set $\pi(0) = \textsc{entrance}$.

3. Let i and j be the neighbors of 0 in T.

4. Let u and v be the neighbors of ENTRANCE in G.

5. With probability 1/2 set $\pi(i) = u$ and $\pi(j) = v$, and with probability 1/2 set $\pi(i) = v$ and $\pi(j) = u$.

6. For $i = 1, 2, 3, \ldots$, if vertex i is not a leaf, and $\pi(i)$ is not EXIT or ENTRANCE,

   (a) Let j and k denote the children of vertex i, and let l denote the parent of vertex i.

   (b) Let u and v be the neighbors of $\pi(i)$ in G other than $\pi(l)$.

   (c) With probability 1/2 set $\pi(i) = u$ and $\pi(j) = v$, and with probability 1/2 set $\pi(i) = v$ and $\pi(j) = u$.

We can now define the game of finding a tree T for which a randomly chosen π is an improper embedding or T exits:

Game 4. *The algorithm outputs a rooted binary tree T with t vertices in which each internal vertex has two children. A random π is chosen. The algorithm wins if π is an improper embedding of T in G'_n or T exits G'_n under π.*

As the algorithm A merely serves to produce a distribution on trees T, we define

$$\mathcal{P}^G(T) = \Pr_\pi\left[\pi \text{ is improper for } T \text{ or } T \text{ exits } G \text{ under } \pi\right], \tag{5.47}$$

and observe that for every distribution on graphs G and all algorithms taking at most t steps,

$$\max_A \mathop{\mathrm{E}}_G\left[\mathcal{P}_4^G(A)\right] \le \max_{\text{trees } T \text{ with } t \text{ vertices}} \mathop{\mathrm{E}}_G\left[\mathcal{P}^G(T)\right]. \tag{5.48}$$

(Here $\mathrm{E}_G[\cdot]$ means the expectation over graphs.) Game 3 and Game 4 are also equivalent:

Lemma 5.7. *For any algorithm A for Game 3 that uses at most t queries of the oracle, there exists an algorithm A' for Game 4 that outputs a tree of at most t vertices such that for all graphs G,*

$$\mathcal{P}_3^G(A) = \mathcal{P}_4^G(A'). \tag{5.49}$$

Proof. Algorithm A halts if it ever finds a cycle, exits, or uses t steps. Algorithm A' will generate a (random) tree by simulating A. Suppose that vertex a in graph G corresponds to vertex a' in the tree that A' is generating. If A asks the oracle for the names of the neighbors of a, A' generates two unused names b' and c' at random and uses them as the neighbors of a'. Now b' and c' correspond to b and c, the neighbors of a in G. Using the tree generated by A' in Game 4 has the same behavior as using A in Game 3. □

Finally, we bound the probability that an algorithm wins Game 4:

Lemma 5.8. *For rooted trees T of at most $2^{n/6}$ vertices,*

$$\max_T \mathop{\mathrm{E}}_G\left[\mathcal{P}^G(T)\right] \le 3 \cdot 2^{-n/6}. \tag{5.50}$$

Proof. Let T be a tree with t vertices, $t \le 2^{n/6}$, with image $\pi(T)$ in G'_n under the random embedding π. The vertices of columns $n+1, n+2, \ldots n+\frac{n}{2}$ in G'_n divide naturally into $2^{n/2}$ complete binary subtrees of height $n/2$.

1. It is very unlikely that $\pi(T)$ contains the root of any of these subtrees, i.e., that $\pi(T)$ includes any vertex in column $n+\frac{n}{2}$. Consider a path in T from the ROOT to a leaf. The path has length at most t, and there are at most t such paths. To reach column $n+\frac{n}{2}$ from column $n+1$, π must choose to move right $\frac{n}{2}-1$ times in a row, which has probability $2^{1-n/2}$. Since there are at most t tries on each path of T (from the ROOT to a leaf) and there are at most t such paths, the probability is bounded by $t^2 \cdot 2^{1-n/2}$.

2. If $\pi(T)$ contains a cycle, then there are two vertices a, b in T such that $\pi(a) = \pi(b)$. Let P be the path in T from a to b. Then $\pi(P)$ is a cycle in G'_n. Let c be the vertex in T closest to the root on the path $\pi(P)$, and let $\pi(P)$ consist of the path $\pi(P_1)$ from c to a and $\pi(P_2)$ from c to b.

   Let $S_1, S_2, \ldots, S_{2^{n/2}}$ denote the $2^{n/2}$ subtrees described above. Let $S'_1, S'_2, \ldots, S'_{2^{n/2}}$ denote the corresponding subtrees made out of columns $\frac{n}{2}+1$ to n. Without loss of generality, let $\pi(c)$ be in the left tree of G'_n, i.e., in a column $\le n$, as shown in Figure 5-8.

   $\pi(P_1)$ visits a sequence of subtrees $S'_{i_1}, S_{j_1}, S'_{i_2}, \ldots$ and, similarly, $\pi(P_2)$ visits a sequence of subtrees $S'_{k_1}, S_{l_1}, S'_{k_2}, \ldots$. Since $\pi(a) = \pi(b)$, the last subtree on these two

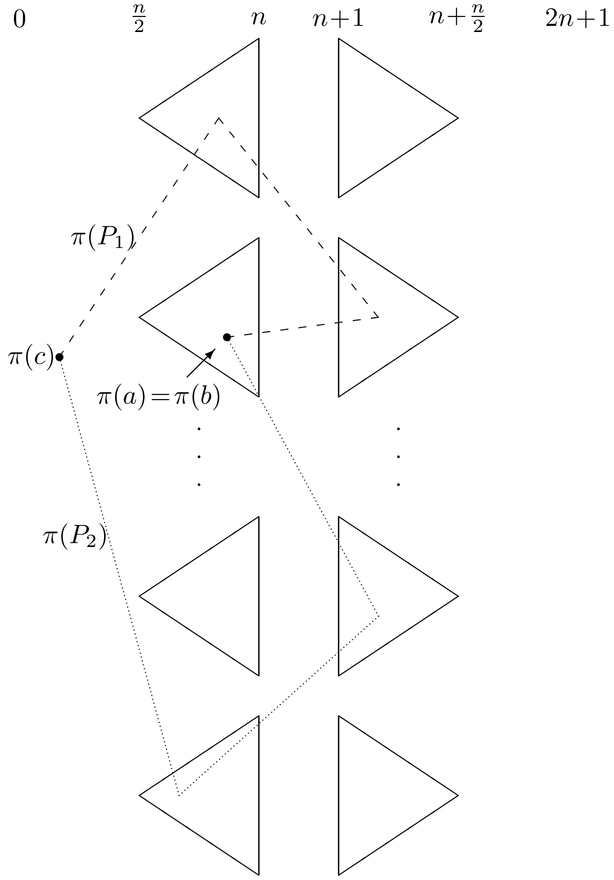

Figure 5-8: Graphical representation of part 2 of the proof of Lemma 5.8. The triangles represent the subtrees of G'_n of height $n/2$, the dashed line represents the path $\pi(P_1)$, and the dotted line represents the path $\pi(P_2)$. Together, these paths form a cycle in the graph.

lists must be the same. (The other possibility is that $\pi(a) = \pi(b)$ does not lie in any subtree, hence lies in columns 1 through $\frac{n}{2}$ or $n+\frac{n}{2}+1$ through $2n$. But the event that column $n + \frac{n}{2}$ is ever reached has already been shown to be unlikely in part 1. The same argument bounds the probability of a return to column $\frac{n}{2}$ after a visit to column $n + 1$.) At least one of the lists has more than one term (or all vertices visited are in the left tree, which can't make a cycle). The probability that the last terms on the two lists agree is bounded by $2^{n/2}/(2^n - t)$, by the construction of the random cycle that connected the two trees of G'_n. As long as $t \leq 2^{n-1}$, we have $2^{n/2}/(2^n - t) \leq 2 \cdot 2^{-n/2}$. Since there are $\binom{t}{2}$ paths P to be considered, the probability of a cycle is less than $t^2 \cdot 2^{-n/2}$.

Overall we have shown that

$$\mathop{\mathrm{E}}_{G}\left[\mathcal{P}^G(T)\right] \leq t^2 \cdot 2^{-n/2} + t^2 \cdot 2^{1-n/2} \tag{5.51}$$

$$\leq 3 \cdot 2^{-n/6} \tag{5.52}$$

if $t \leq 2^{n/6}$. □

This completes the proof of Theorem 5.4.

Since we are only interested in proving an exponential separation, we have not tried to optimize the analysis. By a slight improvement of Lemma 5.8, Theorem 5.4 can be improved to show that any algorithm making at most $2^{n/3}$ queries finds the EXIT with probability at most $O(n2^{-n/3})$ [83].

5.4 Discussion

In this chapter, we have shown examples of situations where quantum walks achieve exponential speedup over classical processes. We began with an example in which a quantum walk is exponentially faster than the corresponding classical walk. We then modified this example to produce a black box problem that can be solved efficiently using a quantum walk, but not by any classical algorithm.

The speedup of our algorithm is essentially the result of a fast quantum *hitting time*, the time to travel from one particular vertex to another. However, it can also be viewed as mitigating the results of [4] regarding the *mixing time* of a quantum walk, the time required to approach an appropriately defined limiting distribution. For discrete-time walks, [4] pointed out that since the mixing times of both classical and quantum walks are controlled by related eigenvalue gaps, a quantum walk cannot mix much faster than the corresponding classical walk. Similar considerations apply for continuous-time walks. However, this result does not say much about the relative behavior of classical and quantum walks, since the limiting distributions of the two kinds of walks can be radically different. This fact was used in the proof of Lemma 5.1, where we showed that the quantum walk reaches the EXIT because its time-averaged probability distribution rapidly approaches a uniform distribution on the *columns*, of which there are only polynomially many. In contrast, the corresponding classical random walk approaches a uniform distribution on the *vertices*, in which the probability of being at the EXIT is exponentially small.

Note that although our quantum algorithm finds the name of the EXIT, it does not find a particular path from ENTRANCE to EXIT. If the algorithm stored information about its path, then it would not exhibit constructive interference, and the speedup would be lost. It is not clear whether this phenomenon can be avoided. It would be interesting to find an efficient quantum algorithm for finding a path from ENTRANCE to EXIT, or to prove that no such algorithm exists.

Although it is convenient to express our results in terms of a graph traversal problem, the results can also be cast in terms of a graph *reachability* problem, where one is given a graph G and two vertices S and T, and the goal is to determine whether or not there is a path connecting S and T. The idea is to let G consist of two disjoint copies of G_n', and set S to be the ENTRANCE of one of the copies and T to be the EXIT of either the same copy or the other copy of G_n'. The quantum algorithm of Section 5.3.2 can be adapted to solve this problem in polynomial time, and the lower bound of Section 5.3.4 can be adapted to show that no classical algorithm can solve this problem in subexponential time. (See [185] for a survey of classical results about graph reachability problems in various contexts.)

Many computational problems can be recast as determining some property of a graph. A natural question is whether there are useful computational problems (especially non-oracular ones) that are classically hard (or are believed to be classically hard) but that can be solved efficiently on a quantum computer employing quantum walks.

Chapter 6

Bipartite Hamiltonians as quantum channels

6.1 Introduction

We now turn our attention from quantum algorithms to the exchange of quantum information between two systems. The fundamental resource for information processing is an interaction between two systems. Here we use Hamiltonian dynamics as a model of such an interaction, and we discuss some of its consequences for quantum information processing.

Any interaction Hamiltonian $H \neq H_A \otimes I_B + I_A \otimes H_B$ that is not a sum of local[1] terms couples the systems A and B. Together with local operations, such a coupling can be used for a variety of tasks, such as transmitting classical or quantum information [15, 26, 101, 32], generating entanglement between the two systems [70, 123, 26, 191, 121], or simulating the dynamics of some other bipartite Hamiltonian H' [68, 186, 117, 176, 187, 152, 39, 177, 139, 23, 175]. One of the goals of quantum information theory is to quantify the ability of an interaction to perform such information processing tasks.

To focus on the purely nonlocal properties of Hamiltonians, we will work in a framework of perfect local control. In other words, local control is regarded as a free resource. This involves the ability to perform arbitrarily fast local operations to modify the evolution, and may also include the use of local ancillary degrees of freedom.

The problem of simulating one bipartite Hamiltonian with another can be posed as follows. We consider two quantum systems A and B that interact according to some nonlocal Hamiltonian H. Separate parties in control of the two systems want to use H to produce an evolution according to some other bipartite Hamiltonian H'. They must do so using only the evolution H and their ability to control the system locally.[2] The goal of the simulation is not to produce the Hamiltonian evolution $e^{-iH't}$ for a particular time t, but rather to stroboscopically track the evolution $e^{-iH't}$ for arbitrarily closely spaced values of time. We discuss the simulation problem in more detail in Section 6.2, where we give some rules for

[1] In this chapter, Hamiltonians and other operations are referred to as local if they act *only* on one subsystem, i.e., on system A alone or system B alone. This can be contrasted with the use of the word "local" in earlier chapters to describe constraints on the couplings between various parts of a large quantum system, meaning for example that each qubit is only coupled to a few other qubits, that the connections between qubits are spatially local, or that the Hamiltonian only couples matrix elements corresponding to connected vertices in a sparse graph.

[2] In principle, we might allow classical communication between the two parties, but this would not increase the simulation rate [176].

bipartite Hamiltonian simulation similar to the rules from Chapter 1 for computational Hamiltonian simulation.

A basic result in bipartite Hamiltonian simulation is that *any* nonlocal Hamiltonian H can be used to simulate any other at a nonzero rate, even without the ability to control local ancillas.[3] This was shown for two-qubit systems in [70, 68, 23], for a pair of d-dimensional quantum systems in [152] and [23, Appendix C], and for an arbitrary pair of finite-dimensional quantum systems in [187]. In Section 6.3, we give a construction for the d-dimensional case along the lines of [152].

Unfortunately, the general simulation presented in Section 6.3 is typically quite inefficient. Ultimately, we would like to know the *optimal* rate $\gamma_{H'|H}$ at which any interaction Hamiltonian H can be used to simulate any other interaction Hamiltonian H', as well as a protocol for doing so. A method for optimal simulation of two-qubit Hamiltonians is given in [23], and the optimal rate in this case can be expressed in terms of a majorization condition [176]. However, little is known about optimal simulation beyond the two-qubit case.

The fact that any nonlocal Hamiltonian can simulate any other at some nonzero rate means that all interactions are *qualitatively* equivalent. A stronger, quantitative notion of equivalence between interactions comes from the possibility of performing a *reversible* simulation. We say that H and H' can simulate each other reversibly if we can use H to simulate H', and then use H' to simulate H back, with no overall loss in efficiency. In terms of simulation rates, reversible simulation amounts to the condition

$$\gamma_{H|H'}\, \gamma_{H'|H} = 1\,. \tag{6.1}$$

In Section 6.4, we show that all tensor product Hamiltonians of the form $H = H_A \otimes H_B$ can simulate each other reversibly. Thus, for this particularly simple class of Hamiltonians, we establish optimal simulation rates. Here we allow the use of local ancillas, since the Hamiltonians H and H' might not even act on spaces of the same dimension.

Understanding Hamiltonian simulation also provides insight into capacities for other information processing tasks. Let C_H denote the capacity of the Hamiltonian H to accomplish some task, again assuming perfect local control. The Hamiltonian capacity can be defined as

$$C_H := \sup_t \frac{C_{e^{-iHt}}}{t} = \lim_{t\to 0} \frac{C_{e^{-iHt}}}{t} \tag{6.2}$$

where $C_{e^{-iHt}}$ is the asymptotic capacity of the unitary gate e^{-iHt} to perform the given task [26]. If Hamiltonian H can be used to simulate H' at a rate $\gamma_{H'|H}$, then clearly

$$C_H \geq \gamma_{H'|H}\, C_{H'}\,, \tag{6.3}$$

since one could first use H to simulate H' and then use H' to accomplish the task. Equation (6.3) is a lower bound on the capacity of H, or equivalently, an upper bound on the capacity of H'. Of course, such bounds need not be tight. For example, the majorization condition for optimal simulation of two-qubit Hamiltonians [176] only provides a partial order on these Hamiltonians, and thus the resulting bounds on capacities—for example, on the entanglement capacity [26, 70, 49]—are not always tight. However, notice that if two Hamiltonians H and H' can simulate each other reversibly, then their capacities are related

[3] Note that the same is not true for multipartite Hamiltonians coupling more than two systems [176, 34].

by

$$C_H = \gamma_{H'|H}\, C_{H'}\,, \tag{6.4}$$

as can be seen by applying (6.3) in both directions. In general, if every pair of Hamiltonians in some given set can simulate each other reversibly, then simulation provides a total order on the set. Thus the nonlocal properties of the entire set can be studied by focusing on only one Hamiltonian in the set.

Finally, in Section 6.5, we address a different information processing task, the generation of entanglement between two systems. For this problem, even the two-qubit case is nontrivial because in general, the rate of entanglement generation can be increased through the use of ancillary systems, and no upper bound is known on the dimension of the ancillas needed to achieve the optimal rate. However, for a certain class of two-qubit Hamiltonians including the Ising interaction, we show that ancillary systems do not increase the capacity, so that it can be easily computed. Using (6.4), this allows us to compute the entanglement capacity of any interaction that can reversibly simulate the Ising interaction—in particular, all tensor product Hamiltonians.

6.2 Simulation rules

In this section, we present a list of rules for nonlocal Hamiltonian simulation. By composition, these rules give rise to all possible simulations achievable with local operations and ancillary systems. We present five basic rules, as well as three additional rules that can be obtained by combining the basic ones.

We use the shorthand notation $H \longrightarrow H'$ to represent the fact that H can be used to simulate H' at the rate $\gamma_{H'|H}$, and the notation $H \longleftrightarrow H'$ to indicate that, in addition, the simulation can be reversed with no loss of efficiency, as in (6.1). We say that two Hamiltonians are *locally equivalent* if they can simulate each other reversibly at unit rate, i.e., $\gamma_{H'|H} = \gamma_{H|H'} = 1$.

Many of these rules are analogous to those given in Chapter 1. However, they differ in several important ways. Whereas Chapter 1 described simulation in a computational setting, in this chapter we are using the model of perfect local control. Therefore we will allow potentially complicated operations to be performed arbitrarily quickly, as long as they are local. Additionally, since it turns out that any Hamiltonian can simulate any other at a nonzero rate, we are only interested in simulation rules with at most linear overhead.

The first two basic rules merely make precise the notion of Hamiltonian evolution. They do not involve any operational procedure, nor assume any ability to control the system. The first rule makes precise the notion of rescaling the evolution time: a Hamiltonian H can reversibly simulate another Hamiltonian $H' = cH$ that only differs by a *positive* multiplicative constant c.

Rule 6.1 (Rescaling). *For any $c > 0$,*

$$H \longleftrightarrow cH\,, \quad \gamma_{cH|H} = \frac{1}{c}\,. \tag{6.5}$$

Note that it is important that $c > 0$. In general, Hamiltonians H and $-H$ cannot simulate each other reversibly (see [23, 187] for examples). This contrasts with Rule 1.2, where every simulation was expressed in terms of a sequence of few-qubit unitary gates, and hence could be run backward.

The second rule makes precise what it means for a Hamiltonian to act on a subsystem. In the bipartite setting, the complete system can be described by subsystems A, B on which H acts and ancillary subsystems A', B' on which it acts trivially.

Rule 6.2 (Ancillas). *For any dimension of the ancillary Hilbert space $\mathcal{H}_{A'} \otimes \mathcal{H}_{B'}$,*

$$H \longleftrightarrow H \otimes I_{A'B'}\,, \quad \gamma_{H\otimes I_{A'B'}|H} = 1\,. \tag{6.6}$$

The next two basic rules arise from the possibility of switching on local Hamiltonians. A purely local Hamiltonian can be simulated without the use of any interaction:

Rule 6.3 (Local Hamiltonians). *Any local Hamiltonian of the form $H_0 = H_A \otimes I_B + I_A \otimes H_B$ can be produced at no cost.*

Also, by means of local unitaries, any Hamiltonian H is locally equivalent to any other that is obtained from it by local unitary conjugation. This follows from the simple identity

$$U e^{-iHt} U^\dagger = e^{-iUHU^\dagger t} \tag{6.7}$$

for any unitary transformation U.

Rule 6.4 (Local unitaries). *For any local unitary operation $U = U_A \otimes U_B$,*

$$H \longleftrightarrow UHU^\dagger\,, \quad \gamma_{UHU^\dagger|H} = 1\,. \tag{6.8}$$

Rules 6.1–6.4 allow us to produce Hamiltonians that differ from the original interaction H. The Lie product formula (1.3) tells us how two of these Hamiltonians can be combined into a new one by alternately simulating each of them individually. We suppose that we can choose to apply either H_1 or H_2, but that only one can be applied at any given time. Since we allow perfect local control, we need not be concerned with the error induced by approximating (1.3) by a finite number of terms. With the help of Rule 6.1, this gives the last basic rule.

Rule 6.5 (Convex combination). *For any H_1, H_2 and $0 \le p \le 1$, the simulation*

$$\left.\begin{array}{r} p\,H_1 \\ (1-p)H_2 \end{array}\right\} \longrightarrow H' = pH_1 + (1-p)H_2 \tag{6.9}$$

is possible with rate $\gamma_{H'|pH_1;(1-p)H_2} \ge 1$.

Here the notation $pH_1; (1-p)H_2$ indicates that we have considered the use of Hamiltonian H_1 for a total fraction of time p and Hamiltonian H_2 for a total fraction of time $1-p$, and the rate of simulating H' is computed by adding these two times together. In practice, the simulation is achieved by alternating the use of H_1 and H_2 many times. We stress that (6.9) assumes only the local ability to switch between the two constituent Hamiltonians, and that only one Hamiltonian is acting at a time.

Notice that Rule 6.5 is the only basic simulation rule where irreversibility may occur. Although we can always use H' to simulate back H_1 and H_2, in general we will incur an overall loss in efficiency by doing so.

These basic rules can be combined in various ways. We state three particularly useful combinations as additional rules. First, from Rules 6.3 and 6.5, a local Hamiltonian H_0 can be added to a given nonlocal Hamiltonian reversibly.

Rule 6.6 (Adding a local Hamiltonian).

$$H \longleftrightarrow H + H_0\,, \quad \gamma_{H+H_0|H} = 1\,. \tag{6.10}$$

Second, local unitary conjugation and convex combination can be composed into what we shall call a local unitary mixing of H.

Rule 6.7 (Local unitary mixing). *For any set of local unitary transformations $U_i = U_{A,i} \otimes U_{B,i}$ and any probability distribution p_i ($p_i \geq 0$ and $\sum_i p_i = 1$),*

$$H \longrightarrow H' = \sum_i p_i\, U_i H U_i^\dagger\,, \quad \gamma_{H'|H} \geq 1\,. \tag{6.11}$$

Rule 6.1 and Rules 6.3–6.7 do not require local control over ancillary degrees of freedom. In the two-qubit case, Rules 6.1, 6.6, and 6.7 describe all relevant simulations because local control over ancillas is known to be unnecessary for optimal simulations [23]. But in general, we may wish to allow local control over ancillas in our simulation model. By Rule 6.2, Rules 6.3–6.7 can be extended to include ancillas as well. Control over ancillas gives extra freedom in the simulation, and is known to improve the achievable simulation rates in some cases [176].

Our last rule is concerned with any simulation in which the original Hamiltonian H and the simulated Hamiltonian H' act on systems with different dimensions. Let $\mathcal{H}_A \otimes \mathcal{H}_B$ and $\mathcal{H}_{A'} \otimes \mathcal{H}_{B'}$ denote the Hilbert spaces on which H and H' act, with dimensions d_A, d_B and $d_{A'}, d_{B'}$, where $d_A \geq d_{A'}$ and $d_B \geq d_{B'}$. For simplicity, we assume that $H = H_A \otimes H_B$ is a product Hamiltonian. If it were not, then we could expand H as a linear combination of product Hamiltonians, $H = \sum_i H_{A,i} \otimes H_{B,i}$, and the following would hold for each of the terms in the expansion. Let vectors $|j\rangle_A$ ($1 \leq j \leq d_A$) denote an orthonormal basis in $\mathcal{H}_A$. We can express H_A as

$$H_A = \begin{pmatrix} J_\| & C^\dagger \\ C & J_\perp \end{pmatrix}\,, \tag{6.12}$$

where $J_\|$ is the restriction of H_A onto the subspace $\mathcal{H}_{A\|} \subseteq \mathcal{H}_A$ spanned by vectors $|j\rangle_A$ ($1 \leq j \leq d_{A'}$), and $J_\perp$ the restriction onto its orthogonal complement. Consider also an analogous decomposition for H_B. Then we have the following:

Rule 6.8 (Reduction to a local subspace). *The simulation*

$$H = \begin{pmatrix} J_\| & C^\dagger \\ C & J_\perp \end{pmatrix} \otimes \begin{pmatrix} K_\| & D^\dagger \\ D & K_\perp \end{pmatrix} \longrightarrow H' = J_\| \otimes K_\| \tag{6.13}$$

is possible with rate $\gamma_{H'|H} \geq 1$.

Proof. To establish this rule, we consider the simulation

$$H = \begin{pmatrix} J_\| & C^\dagger \\ C & J_\perp \end{pmatrix} \otimes H_B \longrightarrow H' = J_\| \otimes H_B\,. \tag{6.14}$$

Performing such a simulation twice, once in each direction, gives Rule 6.8.

We divide the simulation into two steps. First, by unitary mixing (Rule 6.7) with

$$p_1 = \frac{1}{2}\,, \quad U_1 = I_A \otimes I_B\,, \tag{6.15}$$

$$p_2 = \frac{1}{2}\,, \quad U_2 = \begin{pmatrix} I_\parallel & 0 \\ 0 & -I_\perp \end{pmatrix} \otimes I_B\,, \tag{6.16}$$

where $I_\parallel$ and $I_\perp$ denote restrictions of the identity operator, we achieve the simulation

$$H = \begin{pmatrix} J_\parallel & C^\dagger \\ C & J_\perp \end{pmatrix} \otimes H_B \longrightarrow H'' = \begin{pmatrix} J_\parallel & 0 \\ 0 & J_\perp \end{pmatrix} \otimes H_B \tag{6.17}$$

with unit rate, so $\gamma_{H''|H} \geq 1$.

Second, we use H'' to simulate $H' = J_\parallel \otimes H_B$ as follows. The goal is to evolve a state $|\psi\rangle_{A'B}$ according to $e^{-iH't}$. We assume system A is locally prepared in the state $|1\rangle_A$. Therefore the joint state of systems $AA'B$ is initially $|1\rangle_A|\psi\rangle_{A'B}$. Let $V_{AA'}$ denote a unitary transformation such that

$$V_{AA'}|1\rangle_A|j\rangle_{A'} = |j\rangle_A|1\rangle_{A'}\,, \quad 1 \leq j \leq d_{A'}\,. \tag{6.18}$$

Then the following three steps can be used to complete the desired simulation:

1. Apply the unitary operation $V_{AA'}$, placing $|\psi\rangle_{A'B}$ in the subspace $\mathcal{H}_{A\parallel} \otimes \mathcal{H}_B \subset \mathcal{H}_A \otimes \mathcal{H}_B$.

2. Let AB evolve according to H''. Notice that at all times $e^{-iH''t}|\psi\rangle_{AB}$ is supported in $\mathcal{H}_{A\parallel} \otimes \mathcal{H}_B$, and that H'' acts on this subspace as $J_\parallel \otimes H_B$.

3. Apply the unitary operation $V_{AA'}^\dagger$, so that the net evolution on $|\psi\rangle_{A'B}$ has been $e^{-iH't}$.

This completes the proof. □

6.3 Qualitative equivalence of all interaction Hamiltonians

In this section, we show that any interaction Hamiltonian between systems A and B, both of dimension d, can simulate any other interaction at a nonzero rate. The simulation given here does not require ancillary degrees of freedom. In fact, any nonlocal $(d_A \times d_B)$-dimensional bipartite Hamiltonian can simulate any other at a nonzero rate without the use of ancillas, as shown in [187] using different methods.

We will work in a basis of d-dimensional Pauli operators, as used by Gottesman in the investigation of stabilizer codes for d-level systems [96]. The d-dimensional Pauli group consists of all $d \times d$ matrices of the form $\omega^l X^j Z^k$, where $j, k, l \in \{0, 1, \ldots, d-1\}$, $\omega = e^{2\pi i/d}$, and

$$X|z\rangle = |z+1\rangle \tag{6.19}$$

$$Z|z\rangle = \omega^z|z\rangle \tag{6.20}$$

where addition is performed modulo d. Note that $X^d = Z^d = I$. These matrices satisfy the commutation relation

$$(X^j Z^k)(X^l Z^m) = \omega^{kl-jm}(X^l Z^m)(X^j Z^k)\,. \tag{6.21}$$

We will also use the simple fact

$$\frac{1}{d}\sum_{k=0}^{d-1}\omega^{jk} = \begin{cases} 1 & j=0 \\ 0 & \text{otherwise.} \end{cases} \tag{6.22}$$

Any d-dimensional operator M can be expanded in terms of the Pauli basis. Specifically,

$$M = \sum_{j,k} m_{j,k} X^j Z^k \tag{6.23}$$

where $m_{j,k} = \frac{1}{d}\operatorname{tr}(Z^{-k}X^{-j}M)$. If M is Hermitian, then $m_{j,k} = \omega^{jk} m^*_{-j,-k}$. This decomposition can be easily extended to multiple systems.

For a Hamiltonian H expressed in this form, it is easy to see how to simulate its negation (albeit not reversibly, as discussed above). For the purposes of this section, we will need the following additional simulation rule:

Rule 6.9 (Negation). *For any Hamiltonian H,*

$$H \longrightarrow -H\,, \quad \gamma_{-H|H} > 0\,. \tag{6.24}$$

Proof. By direct calculation,

$$\frac{1}{d}\sum_{j,k}(X^j Z^k)M(Z^{-k}X^{-j}) = (\operatorname{tr} M)\, I \tag{6.25}$$

for any operator M. Therefore,

$$\frac{1}{d^2}\sum_{(j,k,l,m)\neq(0,0,0,0)}(X^j Z^k \otimes X^l Z^m)H(Z^{-k}X^{-j}\otimes Z^{-m}X^{-l}) = -\frac{1}{d^2}H + (\operatorname{tr} H)\, I\otimes I\,, \tag{6.26}$$

so by Rules 6.1, 6.6, and 6.7, H can be used to simulate $-H$ at a nonzero rate. □

Further discussion of the problem of reversing Hamiltonian evolution can be found in [110, 124].

The construction we use to simulate any interaction Hamiltonian with any other is based on some simple number-theoretic ideas. First, we use Euclid's algorithm to show that every Pauli operator is isospectral to a diagonal one:

Lemma 6.1. *For any dimension d and for integers j,k with $1 \le j,k \le d-1$, there exists a unitary operator U such that $UX^jZ^kU^\dagger = Z^{\gcd(j,k)}$.*

Proof. We claim that there is a unitary operation U that performs the transformations

$$X \to XZ\,, \quad Z \to Z \tag{6.27}$$

when acting by conjugation on the Pauli operators. If d is odd, $U|z\rangle := \omega^{z(z-1)/2}|z\rangle$, and if d is even, $U|z\rangle := \omega^{z^2/2}$. Since the d-dimensional Fourier transformation takes $X \to Z$, $Z \to X^{-1}$, there is also a unitary operation U' that performs the transformation

$$X \to X\,, \quad Z \to XZ\,. \tag{6.28}$$

Therefore, there are unitary operations that perform the transformations

$$X^j Z^k \to X^j Z^{k+cj} \tag{6.29}$$

$$X^j Z^k \to X^{j+ck} Z^k \tag{6.30}$$

for any integer c.

Now we simply apply Euclid's algorithm to find $\gcd(j,k)$ [76, Book 7, Propositions 1 and 2]. If $j > k$, then we transform $X^j Z^k \to X^{j-ck} Z^k$) where $c = j \bmod k$ (the remainder left when j is divided by k). If $j < k$, then we transform $X^j Z^k \to X^j Z^{k-cj}$ where $c = k \bmod j$. Note that in each such step, the gcd of the two terms is unchanged. Repeating this transformation, we eventually reach either $X^{\gcd(j,k)}$ or $Z^{\gcd(j,k)}$, which are isospectral since they are related by the Fourier transform. □

We will also need the following simple lemma.

Lemma 6.2. *Suppose* $\gcd(p,q) = 1$. *Then* $aq = bp \pmod d$ *if and only if there exists an* n *such that* $a = np \pmod d$ *and* $b = nq \pmod d$.

Proof. The reverse implication follows by substitution. To prove the forward implication, suppose $aq = bp \pmod d$. Since $\gcd(p,q) = 1$, there exist integers r, s such that $rp + sq = 1$. Now choose $n = ar + bs$. Then we have

$$\begin{aligned} np &= arp + bsp \pmod d &\quad (6.31)\\ &= arp + asq \pmod d &\quad (6.32)\\ &= a(rp + sq) &\quad (6.33)\\ &= a \pmod d &\quad (6.34) \end{aligned}$$

as required. A similar calculation shows that $nq = b \pmod d$. □

Now we prove that any interaction Hamiltonian can simulate any other:

Theorem 6.3. *Let* H *be a bipartite Hamiltonian that is not simply a sum of local terms, and let* H' *be any bipartite Hamiltonian. Then the simulation* $H \longrightarrow H'$ *is possible at some rate* $\gamma_{H'|H} > 0$.

The idea of the proof is to reduce H to a simple form, and then build back up to an arbitrary Hamiltonian.

Proof. Given that H is nonlocal, it must contain at least one nonzero term $X^j Z^k \otimes X^l Z^m$ (and its Hermitian conjugate).

By Lemma 6.1 and Rule 6.4, $H \longleftrightarrow H_1$, where H_1 contains a nonzero term $Z^a \otimes Z^b$, where $a = \gcd(j,k)$ and $b = \gcd(l,m)$.

By Rule 6.7, $H_1 \longrightarrow H_2$ where

$$H_2 := \frac{1}{d} \sum_{r,s} (Z^r \otimes Z^s) H_1 (Z^{-r} \otimes Z^{-s}) \,. \tag{6.35}$$

Using the commutation relation (6.21), we have $Z^r (X^j Z^k) Z^{-r} = \omega^{rj} X^j Z^k$. Summing over r using (6.22), we see that all non-diagonal terms cancel. The same happens for system B, so that

$$H_2 = \sum_{j,k} \alpha_{j,k}\, Z^j \otimes Z^k \tag{6.36}$$

with $\alpha_{a,b} \neq 0$.

Now let $a/b = p/q$ with p coprime to q. Again by Rule 6.7, we have $H_2 \longrightarrow H_3$ where

$$\begin{align} H_3 &:= \frac{1}{d}\sum_l (X^{-q} \otimes X^p)^l H_2 (X^q \otimes X^{-p})^l \tag{6.37}\\ &= \sum_{j,k} \alpha_{j,k} \biggl(\frac{1}{d}\sum_l \omega^{(qj-pk)l}\biggr) Z^j \otimes Z^k \tag{6.38}\\ &= \sum_n \beta_n (Z^p \otimes Z^q)^n \tag{6.39}\end{align}$$

with $\beta_n := \alpha_{np,nq}$. In the last line we have used Lemma 6.2 to show that only multiples of p and q remain. Note that $\beta_f \neq 0$, where $f := a/p = b/q$.

Since p and q are relatively prime, we can choose r, s such that $rp + sq = 1$. Using Rule 6.7 a third time, we have $H_3 \longrightarrow H_4$ where

$$\begin{align} H_4 &:= \frac{1}{d}\sum_j (\omega^{jf} + \omega^{-jf})(X^{-r} \otimes X^{-s})^j H_3 (X^r \otimes X^s)^j \tag{6.40}\\ &= \sum_n \beta_n \biggl(\frac{1}{d}\sum_j \left[\omega^{j(n+f)} + \omega^{j(n-f)}\right]\biggr)(Z^p \otimes Z^q)^n \tag{6.41}\\ &= \beta_f\, Z^a \otimes Z^b + \beta_f^*\, Z^{-a} \otimes Z^{-b}\,. \tag{6.42}\end{align}$$

Thus we have managed to isolate a single nonzero term.

At this point, the proof given in [152] converts H_4 into a particular product Hamiltonian and uses Uhlmann's theorem to simulate an arbitrary product Hamiltonian, from which an arbitrary Hamiltonian can be built using Rule 6.5. Here we give an alternate construction that continues to use the Pauli basis. The goal of this construction is to use H_4 to produce an arbitrary term of the form $X^j Z^k \otimes X^l Z^m$ (plus its Hermitian conjugate). We will just show how to produce $X^j Z^k \otimes Z^b$ for arbitrary j, k; then the same argument can be used to modify system B.

The various terms $X^j Z^k$ can be separated into equivalence classes where the equivalence relation is similarity (i.e., the members of a class are related by a unitary transformation). If we show how to simulate one member of each class, then Rule 6.4 can be used to simulate an arbitrary term. Lemma 6.1 shows that the Pauli operators can be classified by looking only at the diagonal ones, of the form Z^k. Thus we see that the equivalence classes correspond exactly to the divisors of d. If d is prime, then Z^k and $Z^{k'}$ are isospectral for any $k, k' \neq 0$. In general, if d is composite, then Z^k and $Z^{k'}$ are isospectral if and only if $\gcd(k, d) = \gcd(k', d)$.

Starting from Z^a, we show how to simulate Z, and then how to simulate $Z^{a'}$ for any a'. Overall, this simulation uses Rule 6.7 four times. If a does not divide d, then clearly $Z^a \sim Z$, where $\sim$ denotes similarity. Otherwise, let $\gcd(a, d) = g$. The spectrum of Z^a is $1, \omega^g, \omega^{2g}, \ldots, \omega^{[(d/g)-1]g}$, where each eigenvalue has multiplicity g. In other words, $Z^a \sim Z^g_{d/g} \otimes I_g$, where a subscript on a Pauli matrix indicates its dimension (when different from d). Now let $P_{d/g}$ be a cyclic permutation of dimension d/g, and define the permutation

$$V_g := I_{d/g} \otimes |0\rangle\langle 0| + P_{d/g} \otimes (I_g - |0\rangle\langle 0|) \,. \tag{6.43}$$

Since

$$\frac{g}{d} \sum_{j=0}^{d/g-1} V_g^j (Z_{d/g}^g \otimes I_g) V_g^{-j} = Z_{d/g}^g \otimes |0\rangle\langle 0| \tag{6.44}$$

(using $\sum_{j=0}^{(d/g)-1} \omega^{gj} = 0$), we find $Z^a \longrightarrow Z_{d/g}^g \otimes |0\rangle\langle 0|$. This Hamiltonian can be used to simulate Z using the unitary mixing

$$\sum_{j=0}^{g-1} \omega^j (I \otimes P_g^j)(Z_{d/g}^g \otimes |0\rangle\langle 0|)(I \otimes P_g^{-j}) = \sum_{j=0}^{g-1} \omega^j (Z_{d/g}^g \otimes |j\rangle\langle j|) \tag{6.45}$$

$$= Z_{d/g}^g \otimes Z_g \sim Z \,. \tag{6.46}$$

If a' does not divide d, then $Z^{a'}$ is isospectral to Z, and we are done. Otherwise, let $\gcd(a', d) = g'$. Note that $Z \sim Z_{d/g'}^{g'} \otimes Z_{g'}$, and perform the simulation

$$\frac{g'}{d} \sum_{j=0}^{d/g'-1} V_{g'}^j (Z_{d/g'}^{g'} \otimes Z_{g'}) V_{g'}^{-j} = Z_{d/g'}^{g'} \otimes |0\rangle\langle 0| \,. \tag{6.47}$$

This Hamiltonian can then be used to simulate

$$\sum_{j=0}^{g'-1} (I \otimes P_{g'}^j)(Z_{d/g'}^{g'} \otimes |0\rangle\langle 0|)(I \otimes P_{g'}^{-j}) = \sum_{j=0}^{g'-1} (Z_{d/g'}^{g'} \otimes |j\rangle\langle j|) \tag{6.48}$$

$$= Z_{d/g'}^{g'} \otimes I_{g'} \sim Z^{a'} \,, \tag{6.49}$$

which is our desired term. Overall, we have shown

$$Z^a \sim Z_{d/g}^g \otimes I_g \longrightarrow Z_{d/g}^g \otimes |0\rangle\langle 0| \longrightarrow Z \longrightarrow Z_{d/g'}^{g'} \otimes |0\rangle\langle 0| \longrightarrow Z_{d/g'}^{g'} \otimes I_{g'} \sim Z^{a'} \tag{6.50}$$

for any a, a'.

Note that in this construction, we have had to multiply Pauli operators by powers of ω, which is not a real number for $d > 2$, in which case it is not allowed by Rule 6.1. Furthermore, when we produce some (complex) multiple of $X^j Z^k$, we need the ability to multiply it by an arbitrary complex number. However, this can easily be done by conjugating by any Pauli operator that does not commute with the original one. This will produce the same Pauli operator, but with a different complex phase that can be determined using (6.21). Since any complex number can be expressed as a real linear combination of two different phases, we are done. □

We have shown that any d-dimensional bipartite interaction Hamiltonian can simulate any other at a nonzero rate, even without the use of ancillas. This shows that all bipartite Hamiltonians can be viewed as qualitatively equivalent. Extensions of this idea can also be used to show that a fixed Hamiltonian consisting of pairwise coupling terms between many d-dimensional systems is computationally universal when assisted by local operations, as we discuss briefly in Section 6.6.

6.4 Reversible simulation of product Hamiltonians

In this section, we consider the set of bipartite Hamiltonians that can be written as a tensor product,

$$H = H_A \otimes H_B \,, \tag{6.51}$$

where H_A acts on system A and H_B acts on system B. We shall call such a Hamiltonian a *product Hamiltonian* for short. An example of a product Hamiltonian in a two-qubit system is the Ising interaction

$$H_{\text{Ising}} := \sigma_z \otimes \sigma_z \,. \tag{6.52}$$

The main result of this section is an explicit protocol for the reversible simulation of any product Hamiltonian by another. It follows that the nonlocal properties of a product Hamiltonian H depend entirely on a single parameter. We denote this parameter by $K_\otimes(H)$, and choose it to be the rate $\gamma_{H_{\text{Ising}}|H}$ at which H can simulate the Ising interaction. We find that

$$K_\otimes(H) = \frac{1}{4}\Delta_A \Delta_B \,, \tag{6.53}$$

where Δ_A (Δ_B) denotes the difference between the largest and the smallest eigenvalues of H_A (H_B). The optimal simulation rate between any two product Hamiltonians H and H' can be written in terms of $K_\otimes$ as

$$\gamma_{H'|H} = \frac{K_\otimes(H)}{K_\otimes(H')} \,, \tag{6.54}$$

so that any capacity known for just one product Hamiltonian can be easily computed for any other product Hamiltonian using (6.4) and (6.54). In particular, we can use the entanglement capacity of the Ising interaction computed in Section 6.5 to obtain a simple expression for the entanglement capacity of any product Hamiltonian.

To demonstrate the reversible simulation of tensor product Hamiltonians $H = H_A \otimes H_B$, we will consider product Hamiltonians in a certain standard form. Using Rule 6.4, we may diagonalize H_A and H_B, so we need only consider their eigenvalues. It will also be convenient to modify H_A so that the largest and smallest eigenvalues, $\lambda_A^{\max}$ and $\lambda_A^{\min}$, are equal in magnitude, and similarly for H_B. This can be done by adding a term proportional to the identity to each of H_A and H_B, i.e.,

$$(H_A + cI) \otimes (H_B + dI) = H_A \otimes H_B + c\, I \otimes H_B + d\, H_A \otimes I + cd\, I \otimes I \,. \tag{6.55}$$

The resulting Hamiltonian is locally equivalent to H since they differ only by local terms (Rule 6.6). Furthermore, since

$$(cH_A) \otimes (H_B/c) = H_A \otimes H_B \,, \tag{6.56}$$

we may assume $\lambda_A^{\max} - \lambda_A^{\min} = \lambda_B^{\max} - \lambda_B^{\min} = \Delta$ without loss of generality.

Having put all product Hamiltonians into a standard form, we are ready to show that they can reversibly simulate each other. By the transitivity of reversible simulation, it suffices to show that all product Hamiltonians can reversibly simulate the Ising interaction $H_{\text{Ising}} = \sigma_z \otimes \sigma_z$.

Theorem 6.4. *Any tensor product Hamiltonian $H = H_A \otimes H_B$ can reversibly simulate the*

Ising interaction:

$$H \longleftrightarrow H_{\text{Ising}}\,, \quad \gamma_{H_{\text{Ising}}|H} = \frac{1}{4}\Delta^2\,. \tag{6.57}$$

Proof. For any nonlocal H_1 and H_2, we have

$$\gamma_{H_1|H_2}\gamma_{H_2|H_1} \leq 1\,. \tag{6.58}$$

Otherwise we could use H_1 to simulate itself with simulation rate greater than 1, which is a contradiction.[4] It thus suffices to show that $\gamma_{H_{\text{Ising}}|H} \geq \Delta^2/4$ and $\gamma_{H|H_{\text{Ising}}} \geq 4/\Delta^2$.

Let H act on a Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$ with dimensions d_A and d_B. Since $H = H_A \otimes H_B$ is in the standard form, we may write

$$H = \frac{1}{4}\Delta^2 \operatorname{diag}(1, a_2, a_3, \ldots, a_{d_A-1}, -1) \otimes \operatorname{diag}(1, b_2, b_3, \ldots, b_{d_B-1}, -1) \tag{6.59}$$

where

$$1 = a_1 \geq a_2 \geq \ldots \geq a_{d_A} = -1 \tag{6.60}$$
$$1 = b_1 \geq b_2 \geq \ldots \geq b_{d_B} = -1\,, \tag{6.61}$$

and the corresponding eigenvectors are $|j\rangle_A$ $(1 \leq j \leq d_A)$ and $|j\rangle_B$ $(1 \leq j \leq d_B)$.

We can simulate the Ising interaction using H by restricting to the subspace spanned by the extremal eigenvectors of H_A and H_B, $\{|1\rangle_A|1\rangle_B,\ |d_A\rangle_A|1\rangle_B,\ |1\rangle_A|d_B\rangle_B,\ |d_A\rangle_A|d_B\rangle_B\}$, according to Rule 6.8. In this subspace, H acts as $(\Delta^2/4)H_{\text{Ising}}$. Therefore we have

$$\gamma_{H_{\text{Ising}}|H} \geq \Delta^2/4\,. \tag{6.62}$$

In order to show how to use the Ising interaction H_{Ising} to simulate H, we consider a concatenation of two simulations,

$$H_{\text{Ising}} \longrightarrow H'' \longrightarrow H\,. \tag{6.63}$$

Here $H'' = \frac{1}{4}\Delta^2 H''_{AA'} \otimes H''_{BB'}$ acts on local Hilbert spaces of dimensions $2d_A$ and $2d_B$, and reads

$$H'' = \frac{1}{4}\Delta^2 \operatorname{diag}(1, -1, a_2, -a_2, \cdots, -1, 1) \otimes \operatorname{diag}(1, -1, b_2, -b_2, \ldots, -1, 1)\,. \tag{6.64}$$

Clearly, we can use Rule 6.8 to simulate H by H'' with unit simulation rate. Therefore we need only focus on the simulation of H'' by H_{Ising}. In turn, this can be decomposed into two similar simulations,

$$H_{\text{Ising}} \longrightarrow H''_{AA'} \otimes \sigma_z \longrightarrow H''_{AA'} \otimes H''_{BB'}\,, \tag{6.65}$$

each one with unit rate. In order to simulate $H''_{AA'} \otimes \sigma_z$ using $\sigma_z \otimes \sigma_z$, we append a d_A-

[4] If $\gamma_{H_1|H_2}\gamma_{H_2|H_1} > 1$, then we could concatenate several simulations $H_1 \longrightarrow H_2 \longrightarrow H_1 \longrightarrow \cdots \longrightarrow H_2 \longrightarrow H_1$ to obtain that the optimal simulation rate $\gamma_{H_1|H_1}$ is infinite. Recalling that any bipartite nonlocal Hamiltonian can simulate any other one at finite rate, we would conclude that $\gamma_{H|H}$ is also infinite for any bipartite Hamiltonian. This would contradict, for instance, the results of [23] showing that $\gamma_{H|H} = 1$ for all nonlocal two-qubit Hamiltonians.

dimensional ancilla A to qubit A' (with H_{Ising} acting on $A'B'$) to obtain the Hamiltonian

$$\begin{aligned} H'_{\text{Ising}} &= (I \otimes \sigma_{z,A'}) \otimes \sigma_z &(6.66)\\ &= \operatorname{diag}(1,-1,1,-1,\ldots,1,-1) \otimes \sigma_z\,. &(6.67)\end{aligned}$$

We define

$$p_j := (a_j + 1)/2 \tag{6.68}$$

so that

$$1 = p_1 \geq p_2 \geq \ldots \geq p_{d_A} = 0\,. \tag{6.69}$$

Furthermore, we define d_A local unitary operations U_j ($1 \leq j \leq d_A$), where U_j exchanges the $(2j-1)$th and $(2j)$th basis vectors of AA'. To evolve under $H''_{AA'} \otimes \sigma_z$ for a small time δ, we apply each U_j at time $t = p_j\delta$ and $U_j^\dagger$ at time $t = \delta$. Equivalently, we can use Rule 6.7 with an appropriate probability distribution and set of unitaries. Thus we can use H_{Ising} to simulate $H''_{AA'} \otimes \sigma_z$ with unit efficiency. The second simulation in (6.65) is achieved similarly. The overall rate for H_{Ising} to simulate H'' or H is thus $4/\Delta^2$ by Rule 6.1. □

We have shown that any product Hamiltonian H can reversibly simulate the Ising interaction H_{Ising} with rate $\gamma_{H_{\text{Ising}}|H} = K_\otimes(H)$, where

$$K_\otimes(H) = \frac{1}{4}\Delta_A\Delta_B\,. \tag{6.70}$$

Therefore, any product Hamiltonian H can reversibly simulate any other product Hamiltonian H', with simulation rate given by

$$\gamma_{H'|H} = \frac{K_\otimes(H)}{K_\otimes(H')}\,. \tag{6.71}$$

As discussed previously, in general a bipartite Hamiltonian H cannot reversibly simulate $-H$. Similarly, in general H cannot reversibly simulate its complex conjugate H^*, nor the Hamiltonian $H^\leftrightarrow$ resulting from swapping systems A and B. However, for product Hamiltonians, all these Hamiltonians are locally equivalent: for any product Hamiltonian H,

$$K_\otimes(H) = K_\otimes(-H) = K_\otimes(H^*) = K_\otimes(H^\leftrightarrow)\,. \tag{6.72}$$

Theorem 6.4 can be extended to the case of a sum of bipartite product Hamiltonians acting on separate subsystems. If H_1 and H_2 are two Hamiltonians acting, respectively, on bipartite systems A_1B_1 and A_2B_2, we let $H_1 \boxplus H_2$ denote their sum.[5] In fact, $H_1 \boxplus H_2$ can reversibly simulate a product Hamiltonian H acting on a single bipartite system AB.

Corollary 6.5. *If H_1, H_2, and H' are product Hamiltonians, the simulation*

$$H_1 \boxplus H_2 \longleftrightarrow H' \tag{6.73}$$

can be achieved reversibly, with simulation rate

$$\gamma_{H'|H_1\boxplus H_2} = \gamma_{H'|H_1} + \gamma_{H'|H_2}\,. \tag{6.74}$$

[5] We use the symbol $\boxplus$ rather than $+$ to emphasize that the Hamiltonians being summed act on different pairs of systems. In other words, $H_{AB} \boxplus H_{A'B'} = H_{AB} \otimes I_{A'B'} + I_{AB} \otimes H_{A'B'}$.

Proof. Because of (6.71), we only need to show that the Hamiltonian $H' = c\, H_{\text{Ising}} \boxplus d\, H_{\text{Ising}}$, $c, d \in \mathbb{R}$, can reversibly simulate $H = (|c| + |d|) H_{\text{Ising}}$ at unit rate. In addition, (6.72) implies that we need only consider the case $c, d > 0$.

By Rule 6.2, $H = (c + d) H_{\text{Ising}}$ is locally equivalent to

$$J_1 = (c + d)\, \sigma_{zA} \otimes \sigma_{zB} \otimes I_{A'B'} \,. \tag{6.75}$$

In turn, using local unitaries to swap A with A' and B with B' (Rule 6.4), J_1 is locally equivalent to

$$J_2 = (c + d)\, I_{AB} \otimes \sigma_{zA'} \otimes \sigma_{zB'} \,. \tag{6.76}$$

Then we can simulate $H' = [c/(c + d)] J_1 + [d/(c + d)] J_2$ by convex combination (Rule 6.5) of J_1 and J_2, which shows that $\gamma_{H'|H} \geq 1$.

For the reverse simulation, note that H' is locally equivalent to each of the following four Hamiltonians:

$$\begin{array}{l} c(\sigma_{zA} \otimes I_{A'} \otimes \sigma_{zB} \otimes I_{B'}) + d(I_A \otimes \sigma_{zA'} \otimes I_B \otimes \sigma_{zB'}) \,, \\ c(I_A \otimes \sigma_{zA'} \otimes \sigma_{zB} \otimes I_{B'}) + d(\sigma_{zA} \otimes I_{A'} \otimes I_B \otimes \sigma_{zB'}) \,, \\ c(\sigma_{zA} \otimes I_{A'} \otimes I_B \otimes \sigma_{zB'}) + d(I_A \otimes \sigma_{zA'} \otimes \sigma_{zB} \otimes I_{B'}) \,, \\ c(I_A \otimes \sigma_{zA'} \otimes I_B \otimes \sigma_{zB'}) + d(\sigma_{zA} \otimes I_{A'} \otimes \sigma_{zB} \otimes I_{B'}) \,. \end{array} \tag{6.77}$$

Each of these Hamiltonians can be obtained from H' according to Rule 6.4 by swapping A with A' and B with B' as necessary. An equally weighted convex combination of these four Hamiltonians gives, after rearranging terms,

$$\frac{c + d}{4} (\sigma_{zA} \otimes I_{A'} + I_A \otimes \sigma_{A'}) \otimes (\sigma_{zB} \otimes I_{B'} + I_B \otimes \sigma_{B'}) \,, \tag{6.78}$$

a tensor product Hamiltonian with $\Delta^2 = 4\,(c + d)$. Therefore $\gamma_{H|H'} \geq 1$, which completes the proof. □

It follows from Corollary 6.5 that $K_\otimes$ is *additive* under the sum of product Hamiltonians acting on different pairs of systems,

$$K_\otimes(H_1 \boxplus H_2) = K_\otimes(H_1) + K_\otimes(H_2) \,. \tag{6.79}$$

More generally, for $H = \boxplus_i H_i$ and $H' = \boxplus_i H'_i$, where all H_i and H'_i are bipartite product Hamiltonians, we can perform the simulation

$$H = \boxplus_i H_i \longleftrightarrow H' = \boxplus_i H'_i \tag{6.80}$$

reversibly, with simulation rate given by

$$\gamma_{H'|H} = \frac{\sum_i K_\otimes(H_i)}{\sum_i K_\otimes(H'_i)} \,. \tag{6.81}$$

Finally, we present a case of reversible Hamiltonian simulation that is possible when in addition to local operations and ancillas, catalytic pre-shared entanglement is available. The simulation can be made reversible only in the presence of entanglement, but the entanglement is not used up during the simulation [175]. This simulation is possible for Hamiltonians that are a sum of two tensor product terms that share the same extremal eigenspace. Consider Hamiltonians of the form

$$H = J_A \otimes J_B + G_A \otimes G_B\,. \tag{6.82}$$

Let $J_A^{\rm e}$ denote the restriction of J_A to the subspace corresponding to its two extremal eigenvalues. By Rule 6.3, $J_A^{\rm e}$ can be assumed to be traceless. Let $G_A^{\rm e}$, $J_B^{\rm e}$, $G_B^{\rm e}$ be similarly defined. In terms of these Hamiltonians, we have

Corollary 6.6. *Given the resource of catalytic entanglement, $H = J_A \otimes J_B + G_A \otimes G_B$ is locally equivalent to $[(\Delta_J^2 + \Delta_G^2)/4]\, H_{\rm Ising}$ if the following conditions hold: (i) $J_A^{\rm e}$ and $G_A^{\rm e}$ are supported on the same 2-dimensional Hilbert space, and similarly for $J_B^{\rm e}$ and $G_B^{\rm e}$. (ii) $\operatorname{tr} J_A^{\rm e} G_A^{\rm e} = \operatorname{tr} J_B^{\rm e} G_B^{\rm e}$.*

Proof. $[(\Delta_J^2 + \Delta_G^2)/4] H_{\rm Ising}$ can simulate H termwise using Theorem 6.4 and Rule 6.5, with no need for catalytic entanglement.

The following procedure uses H to simulate $[(\Delta_J^2 + \Delta_G^2)/4]\, H_{\rm Ising}$:

1. Following Rule 6.8, Alice and Bob restrict to the extremal eigenspace, which is common to both terms in H by condition (i). This preserves the extremal eigenvalues. The resulting Hamiltonian is essentially a two-qubit Hamiltonian.

2. We can assume $J_A^{\rm e} \otimes J_B^{\rm e} = (\Delta_J^2/4)\, \sigma_{zA} \otimes \sigma_{zB}$ by a local change of basis. This can be chosen so that $G_A^{\rm e} \otimes G_B^{\rm e} = (\Delta_G^2/4)\,(\cos\theta \sigma_{zA} + \sin\theta \sigma_{xA}) \otimes (\cos\theta \sigma_{zB} + \sin\theta \sigma_{xB})$ for some θ because of condition (ii).

3. A further local change of basis takes $J_A^{\rm e} \otimes J_B^{\rm e} + G_A^{\rm e} \otimes G_B^{\rm e}$ to its normal form $(\Delta_x^2/4)\, \sigma_{xA} \otimes \sigma_{xB} + (\Delta_z^2/4)\, \sigma_{zA} \otimes \sigma_{zB}$ [70], where $\Delta_x^2 + \Delta_z^2 = \Delta_J^2 + \Delta_G^2$.

4. Finally, $(\Delta_x^2/4)\, \sigma_{xA} \otimes \sigma_{xB} + (\Delta_z^2/4)\, \sigma_{zA} \otimes \sigma_{zB}$ can simulate $[(\Delta_x^2 + \Delta_z^2)/4]\, \sigma_{zA} \otimes \sigma_{zB}$ using catalytic entanglement [175].

This completes the proof. □

To conclude, we have seen that all tensor product Hamiltonians can simulate each other reversibly, so that their nonlocal properties are characterized entirely by the quantity $K_\otimes(H)$ given in (6.53). This is an example of lossless interconversion of resources. A related example is the problem of communication through a one-way classical channel. By Shannon's noisy coding theorem [163] together with the reverse Shannon theorem [29], all classical channels can simulate each other reversibly (in the presence of free shared randomness), and hence they can be characterized entirely in terms of a single quantity, their capacity. Similarly, in the presence of free shared entanglement, all one-way quantum channels can simulate each other reversibly (at least on certain input ensembles [24]), and thus they are characterized entirely in terms of their entanglement-assisted capacity for sending classical information.

We noted that Theorem 6.4 can be used to extend results for two-qubit Hamiltonians to arbitrary product Hamiltonians. We will see this in the next section, where we calculate the entanglement capacity of the Ising interaction, thereby determining the entanglement capacity of all product Hamiltonians (as well as Hamiltonians that are a sum of product Hamiltonians acting on different subsystems, and those satisfying the conditions of Corollary 6.6). A similar extension can be obtained for the problem of using bipartite Hamiltonians to simulate bipartite unitary gates. In the case of two-qubit systems, it is known how to optimally produce any two-qubit gate using any two-qubit Hamiltonian [117, 177].

Since all product Hamiltonians are equivalent to some multiple of the Ising interaction, this result immediately provides the optimal way to use any product Hamiltonian to simulate any two-qubit unitary gate, such as the controlled-not gate.

In view of our results, it will be interesting to improve our understanding of the properties of the Ising interaction. For example, a calculation of the communication capacity of the Ising interaction (which is currently unknown) would provide a formula for the communication capacity of all product Hamiltonians.

Of course, the set of product Hamiltonians is clearly a special subset of all bipartite Hamiltonians, and thus may not be representative of the general problem of bipartite Hamiltonian simulation. For example, we have seen that product Hamiltonians admit a total order, whereas even in the two-qubit case, general Hamiltonians only admit a partial order. Also, note that for product Hamiltonians, H and $-H$ are locally equivalent, so that in particular, their capacities to generate entanglement are equal. However, while this is true for all two qubit Hamiltonians, numerical evidence suggests that it is not true in general [48]. Understanding optimal Hamiltonian simulation and the capacities of Hamiltonians in the general case remains an interesting open problem.

6.5 Entanglement capacity

We now focus on a particular application of bipartite Hamiltonian dynamics, the generation of entanglement between two quantum systems. Much experimental effort has been devoted to creating entangled states of quantum systems [86]. Determining the ability of a system to create entangled states provides a benchmark of the "quantumness" of the system (for example, through the violation of Bell inequalities [16, 13]). Furthermore, such states could ultimately be put to practical use in various quantum information processing tasks, such as superdense coding [30], quantum teleportation [22], quantum key distribution [72, 133], remote state preparation [132, 25, 27], and entanglement-assisted classical communication [28, 29, 26].

The problem of optimal entanglement generation by an interaction Hamiltonian can be approached in different ways. For example, [70] considers a *single-shot* scenario. This situation is of interest for present-day experiments attempting to create entangled states. For two-qubit interactions, assuming that ancillary systems are not available, [70] presents an expression for the maximum rate of entanglement generation and optimal protocols by which it can be achieved. In contrast, [123, 26] consider the *asymptotic* entanglement capacity, where a collective input state for many uses of the interacting systems (and local ancillas) can be used to produce entanglement, possibly at a higher rate than in the single-shot case. The asymptotic entanglement capacity is of interest in the context of understanding the ultimate limitations of quantum mechanical systems to process information. References [123, 26] show that if ancillas are allowed, the optimal single-shot rate of entanglement generation is equal to the asymptotic entanglement capacity. However, such capacities can nevertheless be difficult to calculate because the ancillary systems may in principle be arbitrarily large.

In this section, we calculate the asymptotic entanglement capacity of the Ising interaction $\sigma_z \otimes \sigma_z$, and more generally, of any two-qubit interaction that is locally equivalent to $\mu_x \, \sigma_x \otimes \sigma_x + \mu_y \, \sigma_y \otimes \sigma_y$. We consider the use of ancillary systems, and show that they do not increase the rate of entanglement generation for these interactions. Thus in these cases, the asymptotic capacity discussed in [123, 26] is in fact given by the expression presented in

[70]. Furthermore, the results of the previous section allow us to calculate the asymptotic entanglement capacity of an arbitrary product Hamiltonian.

We begin by reviewing some definitions and known results. Let $|\psi\rangle$ be a state of systems A and B. This state can always be written using the Schmidt decomposition [155],

$$|\psi\rangle := \sum_j \sqrt{\lambda_j}\, |\phi_j\rangle_A \otimes |\eta_j\rangle_B \,, \tag{6.83}$$

where $\{|\phi_i\rangle_A\}$ and $\{|\eta_i\rangle_B\}$ are orthonormal bases for the two systems, and $\lambda_j > 0$ with $\sum_j \lambda_j = 1$. The entanglement between A and B is defined as[6]

$$E(|\psi\rangle) := -\sum_j \lambda_j \log \lambda_j \,; \tag{6.84}$$

equivalently, it is the entropy of the reduced density matrix of either system A or B. We say that the state $|\psi\rangle$ contains $E(|\psi\rangle)$ *ebits* of entanglement between A and B. Two pure states with the same amount of entanglement are asymptotically interconvertible using local operations and an asymptotically vanishing amount of classical communication [20, 134, 132, 103, 102].

Reference [70] considers maximizing the rate of increase of entanglement when a pure state is acted on by e^{-iHt}, the evolution according to a time-independent bipartite Hamiltonian H. Ancillary systems cannot be used in the procedure. This maximal rate is referred to as the *entanglement capability*,

$$\Gamma_H := \max_{|\psi\rangle \in \mathcal{H}_{AB}} \lim_{t \to 0} \frac{E(e^{-iHt}|\psi\rangle) - E(|\psi\rangle)}{t} \,. \tag{6.85}$$

Here the rate of increasing entanglement is optimized over all possible pure initial states of the joint Hilbert space $\mathcal{H}_{AB}$ (without ancillary systems). In fact, the single-shot capacity may be higher if ancillary systems A' and B', not acted on by H, are used. For this reason, we may consider the alternative single-shot capacity

$$\Gamma'_H := \sup_{|\psi\rangle \in \mathcal{H}_{AA'BB'}} \lim_{t \to 0} \frac{E(e^{-iHt} \otimes I_{A'B'}|\psi\rangle) - E(|\psi\rangle)}{t} \,. \tag{6.86}$$

Note that in (6.85) and (6.86), the limit is the same from both sides even though it might in general be the case that $\Gamma_H \neq \Gamma_{-H}$ (and similarly for Γ'_H).

The single-shot entanglement capability (6.85) has been completely characterized for two-qubit Hamiltonians. Reference [70] shows that any two-qubit Hamiltonian H is locally equivalent to a Hamiltonian in the *canonical form*

$$\sum_{i=x,y,z} \mu_i \, \sigma_i \otimes \sigma_i \,, \quad \mu_x \geq \mu_y \geq |\mu_z| \,. \tag{6.87}$$

In terms of this canonical form, the entanglement capability of any two-qubit Hamiltonian is given by

$$\Gamma_H = \alpha(\mu_x + \mu_y) \,, \tag{6.88}$$

[6] We use log to denote the base 2 logarithm.

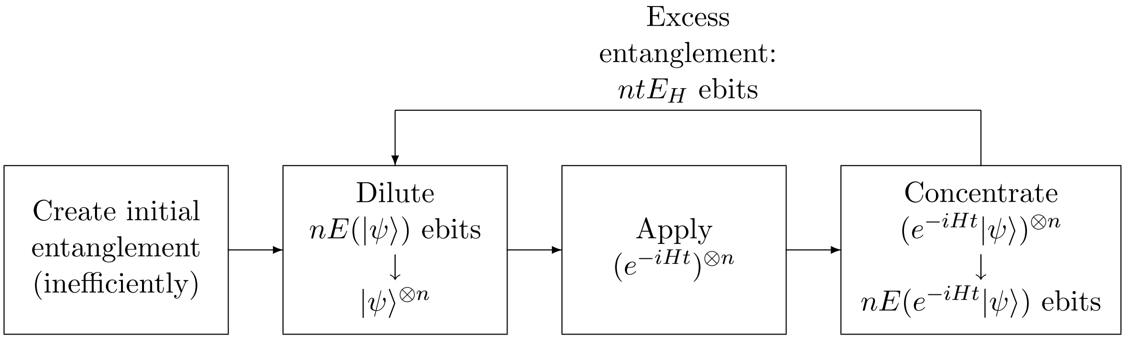
Excess
entanglement:
ntE_H ebits
Create initial entanglement (inefficiently)
Dilute
$nE(|\psi\rangle)$ ebits
$\downarrow$
$|\psi\rangle^{\otimes n}$
Apply
$(e^{-iHt})^{\otimes n}$
Concentrate
$(e^{-iHt}|\psi\rangle)^{\otimes n}$
$\downarrow$
$nE(e^{-iHt}|\psi\rangle)$ ebits

Figure 6-1: Cycle for optimal asymptotic entanglement generation.

where

$$\alpha := 2 \max_{x\in[0,1]} \sqrt{x(1-x)}\, \log\left(\frac{x}{1-x}\right) = 1.9123 \tag{6.89}$$

with the maximum obtained at $x_0 = 0.9168$. In addition, Γ'_H may be strictly larger than Γ_H when $|\mu_z| > 0$ [70].

References [123, 26] consider the *asymptotic* entanglement capacity E_H for an arbitrary Hamiltonian H. This capacity is defined as the maximum rate at which entanglement can be produced by using many interacting pairs of systems. These systems may be acted on by arbitrary collective local operations (attaching or discarding ancillary systems, unitary transformations, and measurements). Furthermore, classical communication between A and B and possibly mixed initial states are allowed. In fact, the asymptotic entanglement capacity in this general setting turns out to be just the single-shot capacity (6.86): $E_H = \Gamma'_H$ for all H [123, 26]. Note that the definition of E_H involves a supremum over both all possible states and all possible interaction times, but in fact it can be expressed as a supremum over states and a limit as $t \to 0$, with the limit and the supremum taken in either order.

Because the capacity E_H is equal to a single-shot capacity, there is a simple protocol for achieving it. Let $|\psi\rangle$ be the optimal input in (6.86). Assuming $|\psi\rangle$ is finite-dimensional, the entanglement capacity can be achieved by first inefficiently generating some EPR pairs, and then repeating the following three steps [70, 123, 26]:

1. Transform $nE(|\psi\rangle)$ EPR pairs into $|\psi\rangle^{\otimes n}$ [20, 134, 132, 103, 102],
2. Evolve each $|\psi\rangle$ according to H for a short time t, and
3. Concentrate the entanglement into $n[E(|\psi\rangle) + tE_H]$ EPR pairs [20].

This protocol is illustrated in Figure 6-1.

Below, we show that $E_K = \Gamma_K$ for any two-qubit Hamiltonian with canonical form

$$K = \mu_x\, \sigma_x \otimes \sigma_x + \mu_y\, \sigma_y \otimes \sigma_y\,, \quad \mu_x \geq \mu_y \geq 0\,, \tag{6.90}$$

so that in this case, all three rates of entanglement generation are equal:

$$E_K = \Gamma'_K = \Gamma_K\,. \tag{6.91}$$

The optimal input is therefore a two-qubit state, and the above protocol applies. In particular, for these Hamiltonians, which include the Ising interaction $\sigma_z \otimes \sigma_z$ and the anisotropic

Heisenberg interaction $\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y$, entanglement can be optimally generated from a two-qubit initial state $|\psi\rangle$ without ancillary systems $A'B'$. As mentioned above, this result is not generic, since ancillas increase the amount of entanglement generated by some two-qubit interactions, such as the isotropic Heisenberg interaction $\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y + \sigma_z \otimes \sigma_z$.

We will focus on computing the asymptotic entanglement capacity of the Ising interaction $H_{\text{Ising}} = \sigma_z \otimes \sigma_z$. This is sufficient to determine the asymptotic entanglement capacity of K in (6.90) since $(\mu_x + \mu_y)H_{\text{Ising}}$ can be used to simulate K at unit rate using Rule 6.5. According to (6.3), this demonstrates that $E_K \le (\mu_x + \mu_y)E_{H_{\text{Ising}}}$. After computing $E_{H_{\text{Ising}}}$, we will see that the protocol of [70] saturates this bound, so in fact $E_K = E_{H_{\text{Ising}}}$ with no need for ancillas to saturate either capacity.[7]

Consider the entanglement capacity of a general bipartite Hamiltonian H. We suppose that in addition to the systems A and B on which H acts, d-dimensional ancillas A' and B' are used, where d is arbitrary. The entanglement capacity of H is given by

$$E_H = \sup_{|\psi\rangle} \Big(-\frac{\mathrm{d}}{\mathrm{d}t} \sum_j \lambda_j \log \lambda_j \Big) \tag{6.92}$$

$$= \sup_{|\psi\rangle} \Big(-\sum_j \frac{\mathrm{d}\lambda_j}{\mathrm{d}t} \log \lambda_j \Big) \tag{6.93}$$

where the λ_j are the Schmidt coefficients of $|\psi\rangle$, from (6.83). The Schmidt coefficients are simply the eigenvalues of the reduced density matrix $\rho := \operatorname{tr}_{BB'} |\psi\rangle\langle\psi| = \sum_j \lambda_j |\phi_j\rangle\langle\phi_j|$. By first-order perturbation theory,

$$\begin{aligned}
\frac{\mathrm{d}\lambda_j}{\mathrm{d}t} &= \langle\phi_j|\frac{\mathrm{d}\rho}{\mathrm{d}t}|\phi_j\rangle && (6.94)\\
&= -i\langle\phi_j|(\operatorname{tr}_{BB'}[H, |\psi\rangle\langle\psi|])|\phi_j\rangle && (6.95)\\
&= -i \sum_{k,l,m} \sqrt{\lambda_l \lambda_m}\, \langle\phi_j, \eta_k|[H, |\phi_l, \eta_l\rangle\langle\phi_m, \eta_m|]|\phi_j, \eta_k\rangle && (6.96)\\
&= \sum_k \sqrt{\lambda_j \lambda_k}\, \operatorname{Im} \langle\phi_j, \eta_j|H|\phi_k, \eta_k\rangle\,, && (6.97)
\end{aligned}$$

where in the second line we have used the Schrödinger equation for the reduced density matrix, $i\frac{\mathrm{d}\rho}{\mathrm{d}t} = \operatorname{tr}_{BB'}[H, |\psi\rangle\langle\psi|]$. Putting this into (6.93), we see that

$$E_H = \sup_{|\psi\rangle} \sum_{j,k} \sqrt{\lambda_j \lambda_k} \log(\lambda_k/\lambda_j) B_{jk} \tag{6.98}$$

where

$$B_{jk} := \operatorname{Im} \langle\phi_j, \eta_j|H|\phi_k, \eta_k\rangle\,. \tag{6.99}$$

Now define $q_{jk} := \lambda_j/(\lambda_j + \lambda_k)$. We have

$$E_H = \sup_{|\psi\rangle} \sum_{j,k} \sqrt{q_{jk}(1-q_{jk})} \log\left(\frac{1-q_{jk}}{q_{jk}}\right) (\lambda_j + \lambda_k) B_{jk} \tag{6.100}$$

[7] Another argument that shows that K and $(\mu_x + \mu_y)H_{\text{Ising}}$ have the same capacity is to note that these interactions are *asymptotically equivalent*, in that K can simulate H_{Ising} given catalytic entanglement [175]. We will return to this argument below.

$$
\begin{aligned}
&\le \alpha \sup_{|\psi\rangle} \sum_{j,k} \frac{\lambda_j + \lambda_k}{2} |B_{jk}| && (6.101)\\
&= \alpha \sup_{|\psi\rangle} \sum_{j,k} \lambda_j |B_{jk}| && (6.102)\\
&\le \alpha \sup_{|\psi\rangle} \max_j \sum_k |B_{jk}|\,, && (6.103)
\end{aligned}
$$

where in the second line we have used (6.89) with $x = 1 - q_{jk}$.

The formula (6.98) and the bound (6.103) apply to any bipartite Hamiltonian H. Now we specialize to the Hamiltonian H_{Ising}. To simplify the argument, we find it convenient to add local terms using Rule 6.6 to produce the locally equivalent Hamiltonian

$$
H_{\text{Ising}} + \sigma_z \otimes I + I \otimes \sigma_z + I \otimes I = 4\,|0\rangle\langle 0| \otimes |0\rangle\langle 0|\,. \tag{6.104}
$$

On the full Hilbert space $\mathcal{H}_{ABA'B'}$, the Hamiltonian is $4\,\Pi_A \otimes \Pi_B$, where $\Pi_A := |0\rangle\langle 0| \otimes I_{A'}$ and $\Pi_B := |0\rangle\langle 0| \otimes I_{B'}$. Using the Cauchy-Schwartz inequality, we find that for this Hamiltonian,

$$
\sum_k |B_{jk}| \le 4 \left(\sum_{k \ne j} |\langle \phi_j | \Pi_A | \phi_k \rangle|^2 \times \sum_{k \ne j} |\langle \eta_j | \Pi_B | \eta_k \rangle|^2 \right)^{1/2}. \tag{6.105}
$$

Since

$$
\begin{aligned}
\sum_{k \ne j} |\langle \phi_j | \Pi_A | \phi_k \rangle|^2 &= \left(\sum_k \langle \phi_j | \Pi_A | \phi_k \rangle \langle \phi_k | \Pi_A | \phi_j \rangle \right) - \langle \phi_j | \Pi_A | \phi_j \rangle^2 && (6.106)\\
&= \langle \phi_j | \Pi_A | \phi_j \rangle (1 - \langle \phi_j | \Pi_A | \phi_j \rangle) && (6.107)\\
&\le 1/4 && (6.108)
\end{aligned}
$$

and similarly for the term involving Π_B, we find $\sum_k |B_{jk}| \le 1$. Putting this into (6.103), we find $E_{H_{\text{Ising}}} \le \alpha$. But since $E_{H_{\text{Ising}}} \ge \Gamma_{H_{\text{Ising}}} = \alpha$, this shows $E_{H_{\text{Ising}}} = \alpha$.

We have shown that ancillary systems are not needed to optimize the rate of entanglement generation for any two-qubit Hamiltonian with canonical form $\mu_x\, \sigma_x \otimes \sigma_x + \mu_y\, \sigma_y \otimes \sigma_y$. Furthermore, there is a universal optimal two-qubit initial state given by [70]

$$
|\psi_{\max}\rangle := \sqrt{x_0}|0\rangle_A \otimes |1\rangle_B - i\sqrt{1 - x_0}|1\rangle_A \otimes |0\rangle_B\,. \tag{6.109}
$$

As discussed above, ancillas are necessary to achieve the capacity in general. Although we do not have a closed-form expression for the capacity of an arbitrary two-qubit Hamiltonian, we can present partial results in this direction. The numerically optimized entanglement capacity of a general two-qubit Hamiltonian is shown in Figure 6-2. Numerically, we find that the optimum can be achieved with single-qubit ancillas on both sides. For Hamiltonians of the form $K_{\mu_{xy}} = \mu_{xy}(\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y) + \sigma_z \otimes \sigma_z$, we conjecture that the entanglement capacity is given by

$$
\begin{aligned}
E_{K_{\mu_{xy}}} = 2 \max \big\{ &\sqrt{\lambda_1 \lambda_2} \log(\lambda_1/\lambda_2)\, [\sin\theta + \mu_{xy} \sin(\varphi - \xi)] \\
&+ \sqrt{\lambda_2 \lambda_4} \log(\lambda_2/\lambda_4)\, [\sin\varphi + \mu_{xy} \sin(\theta - \xi)] \\
&+ \sqrt{\lambda_1 \lambda_4} \log(\lambda_1/\lambda_4)\, \mu_{xy} \sin\xi \big\}
\end{aligned} \tag{6.110}
$$

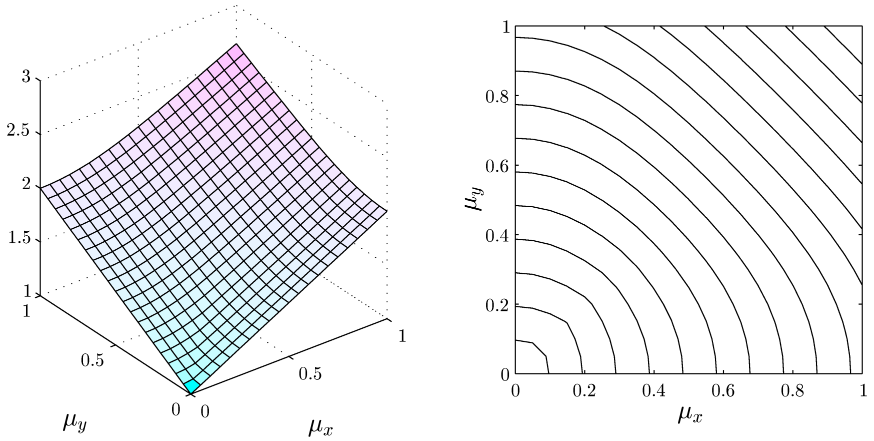

Figure 6-2: Numerically optimized entanglement capacity of the two-qubit Hamiltonian $\mu_x\, \sigma_x \otimes \sigma_x + \mu_y\, \sigma_y \otimes \sigma_y + \sigma_z \otimes \sigma_z$ with single-qubit ancillas on each side. The right figure is a contour plot of the left figure. The vertical axis in the left figure is in units of α.

where the maximum is taken over $\lambda_1 > 0$, $\lambda_2 > 0$, $\lambda_4 = 1 - \lambda_1 - 2\lambda_2 > 0$, and $\theta, \varphi, \xi \in [0, 2\pi)$. This expression was found by investigating the structure of the numerical optimum, and it agrees well with the numerical results. It does not seem possible to simplify this expression further, which suggests that in general, capacities may not have simple expressions, but can only be expressed as maximizations of multivariable transcendental functions. Nevertheless, it would be useful to show that this maximization can be taken over a finite number of parameters by proving an upper bound on the dimension of the ancillas.

Finally, using the results on reversible simulation from Section 6.4, we can now determine the entanglement capacity of any product Hamiltonian. Combining (6.4) and (6.71), we obtain an expression for the entanglement capacity of any product Hamiltonian H, since we have $E_H = K_\otimes(H)\, E_{H_{\mathrm{Ising}}}$. Thus, for any product Hamiltonian H,

$$E_H = \frac{\alpha}{4} \Delta_A \Delta_B \,. \tag{6.111}$$

Note that [179] reported the restricted case of this result in which H_A and H_B have eigenvalues ± 1, but we see that this property has no special significance.

In fact, (6.111) also corresponds to the single-shot capability Γ_H, since it can be obtained without using ancillas. In other words, for any product Hamiltonian H,

$$\Gamma_H = \frac{\alpha}{4} \Delta_A \Delta_B \,. \tag{6.112}$$

The explicit optimal input state is

$$|\psi\rangle = \sqrt{x_0} |+\rangle_A \otimes |+\rangle_B + i\sqrt{1 - x_0} |-\rangle_A \otimes |-\rangle_B \tag{6.113}$$

where

$$|\pm\rangle_A = \frac{1}{\sqrt{2}} (|1\rangle_A \pm |d_A\rangle_A) \tag{6.114}$$

and similarly for system B. Here $|1\rangle_A$ and $|d_A\rangle_A$ represent the eigenstates of H_A corresponding to the largest and smallest eigenvalues, respectively. That this state achieves Γ_H

can be seen by substitution into (6.98).

Similarly, we can compute $E_{H_1 \boxplus H_2}$ for the sum of two product Hamiltonians H_1 and H_2 acting on different pairs of systems:

$$E_{H_1 \boxplus H_2} = E_{H_1} + E_{H_2} \,. \tag{6.115}$$

This capacity can also be achieved without ancillas, because the protocol used in Corollary 6.5 does not involve ancillas:

$$\Gamma_{H_1 \boxplus H_2} = \Gamma_{H_1} + \Gamma_{H_2} \,. \tag{6.116}$$

Finally, we note that (6.111) can be extended to Hamiltonians that can be reversibly simulated using catalytic entanglement. In general, if H satisfies conditions (i) and (ii) of Corollary 6.6, then

$$E_H = \frac{\alpha}{4}(\Delta_J^2 + \Delta_G^2) \,. \tag{6.117}$$

This class of Hamiltonians includes, as a special case, the full set of two-qubit Hamiltonians of the form $\mu_x \, \sigma_x \otimes \sigma_x + \mu_y \, \sigma_y \otimes \sigma_y$ considered above. In the context of asymptotic entanglement capacity, catalytic resources need not be considered as additional requirements since the cost of first obtaining any catalytic resource can be made negligible [26]. However, it turns out that for these Hamiltonians, catalytic entanglement is actually not necessary to achieve the entanglement capacity. There is an input state that achieves E_H without making use of ancillas (and in particular, without using catalytic entanglement), so in fact $\Gamma_H = E_H$.

6.6 Discussion

In this chapter, we have studied properties of bipartite Hamiltonians, including their ability to simulate other bipartite Hamiltonians and to generate entanglement. We showed that any interaction can simulate any other at a nonzero rate and that tensor product Hamiltonians can simulate each other reversibly. We also computed the optimal asymptotic rate at which any product Hamiltonian (and a few other closely related Hamiltonians) can generate entanglement.

In addition to performing simulations in purely bipartite systems and bounding the capacities of various information processing tasks, understanding bipartite Hamiltonian simulation may also be useful for developing implementations of quantum computers. A bipartite Hamiltonian can serve as a model of the type of interaction used to perform two-qubit gates (or perhaps gates between a pair of higher-dimensional subsystems) in a quantum computer. In certain quantum systems, such as in nuclear magnetic resonance (NMR), it may be easy to apply local operations to individual particles, which interact only through a fixed many-particle Hamiltonian. By manipulating the system locally, it can be made to evolve under some different effective Hamiltonian, thereby implementing quantum gates or simulating another Hamiltonian of interest. Specific techniques along these lines are well known in NMR [75], and have been applied to the problem of implementing quantum logic gates [111, 126]. Building on these ideas using constructions like those given in 6.3, one can show that any system of n d-dimensional systems with a Hamiltonian consisting of two-body interactions can simulate any other such system efficiently using only local operations [152]. This provides an alternative universal model of computation in terms of Hamiltonian dynamics using different physical resources than the model discussed in Chapter 1. In the alternative model, there is no need to engineer the interactions, only to locally manipulate the individual particles.

Bibliography

[1] S. Aaronson and A. Ambainis, *Quantum search of spatial regions*, Proc. 44th IEEE Symposium on Foundations of Computer Science, pp. 200–209, 2003, quant-ph/0303041.

[2] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions*, Dover, New York, 1972.

[3] D. S. Abrams and S. Lloyd, *Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors*, Phys. Rev. Lett. **83** (1999), 5162–5165, quant-ph/9807070.

[4] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, *Quantum walks on graphs*, Proc. 33rd ACM Symposium on Theory of Computing, pp. 50–59, 2001, quant-ph/0012090.

[5] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, *Universality of adiabatic quantum computation with two-body interactions.*

[6] D. Aharonov and A. Ta-Shma, *Adiabatic quantum state generation and statistical zero knowledge*, Proc. 35th ACM Symposium on Theory of Computing, pp. 20–29, 2003, quant-ph/0301023.

[7] Y. Aharonov and M. Vardi, *Meaning of an individual "Feynman path"*, Phys. Rev. D **21** (1980), 2235–2240.

[8] G. Ahokas and R. Cleve, personal communication, July 2003.

[9] R. Aleliunas, R. Karp, R. Lipton, L. Lovász, and C. Rackoff, *Random walks, universal traversal sequences, and the time complexity of maze problems*, Proc. 20th IEEE Symposium on Foundations of Computer Science, pp. 218–223, 1979.

[10] A. Ambainis, *Quantum walk algorithm for element distinctness*, quant-ph/0311001.

[11] A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, and J. Watrous, *One-dimensional quantum walks*, Proc. 33rd ACM Symposium on Theory of Computing, pp. 37–49, 2001.

[12] A. Ambainis, J. Kempe, and A. Rivosh, *Coins make quantum walks faster*, quant-ph/0402107.

[13] A. Aspect, J. Dalibard, and G. Roger, *Experimental tests of Bell's inequalities using time-varying analyzers*, Phys. Rev. Lett. **49** (1982), 1804–1807.

[14] F. Barahona, *On the computational complexity of Ising spin-glass models*, J. Phys. A **15** (1982), 3241–3253.

[15] D. Beckman, D. Gottesman, M. A. Nielsen, and J. Preskill, *Causal and localizable quantum operations*, Phys. Rev. A **64** (2001), 052309, quant-ph/0102043.

[16] J. S. Bell, *On the Einstein-Podolsky-Rosen paradox*, Physics **1** (1964), 195–200.

[17] ———, *On the impossible pilot wave*, Found. Phys. **12** (1982), 989–999.

[18] P. Benioff, *Space searches with a quantum robot*, Quantum Computation and Information (S. J. Lomonaco and H. E. Brandt, eds.), AMS Contemporary Mathematics Series, vol. 305, AMS, Providence, RI, 2002, quant-ph/0003006.

[19] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, *Strengths and weaknesses of quantum computing*, SIAM J. Comput. **26** (1997), 1510–1523, quant-ph/9701001.

[20] C. H. Bennett, H. J. Bernstein, S. Popescu, and B. Schumacher, *Concentrating partial entanglement by local operations*, Phys. Rev. A **53** (1996), 2046–2052, quant-ph/9511030.

[21] C. H. Bennett and G. Brassard, *Quantum cryptography: public key distribution and coin tossing*, Proc. IEEE International Conference on Computers, Systems, and Signal Processing, pp. 175–179, 1984.

[22] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Phys. Rev. Lett. **70** (1993), 1895–1899.

[23] C. H. Bennett, J. I. Cirac, M. S. Leifer, D. W. Leung, N. Linden, S. Popescu, and G. Vidal, *Optimal simulation of two-qubit Hamiltonians using general local operations*, Phys. Rev. A **66** (2002), 012305, quant-ph/0107035.

[24] C. H. Bennett, I. Devetak, P. Shor, and A. Winter, in preparation.

[25] C. H. Bennett, D. P. DiVincenzo, P. W. Shor, J. A. Smolin, B. M. Terhal, and W. K. Wootters, *Remote state preparation*, Phys. Rev. Lett. **87** (2001), 077902, quant-ph/0006044.

[26] C. H. Bennett, A. W. Harrow, D. W. Leung, and J. A. Smolin, *On the capacities of bipartite Hamiltonians and unitary gates*, IEEE Trans. Inf. Theory **49** (2003), 1895–1911, quant-ph/0205057.

[27] C. H. Bennett, P. Hayden, D. W. Leung, P. W. Shor, and A. Winter, *Remote preparation of quantum states*, quant-ph/0307100.

[28] C. H. Bennett, P. W. Shor, J. A. Smolin, and A. V. Thapliyal, *Entanglement-assisted classical capacity of noisy quantum channels*, Phys. Rev. Lett. **83** (1999), 3081–3084, quant-ph/9904023.

[29] ———, *Entanglement-assisted capacity of a quantum channel and the reverse Shannon theorem*, IEEE Trans. Inf. Theory **48** (2002), 2637–2655, quant-ph/0106052.

[30] C. H. Bennett and S. J. Wiesner, *Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states*, Phys. Rev. Lett. **69** (1992), 2881–2884.

[31] E. Bernstein and U. Vazirani, *Quantum complexity theory*, Proc. 25th ACM Symposium on Theory of Computing, pp. 11–20, 1993.

[32] D. W. Berry and B. C. Sanders, *Relations for classical communication capacity and entanglement capability of two-qubit operations*, Phys. Rev. A **67** (2003), 040302(R), quant-ph/0205181.

[33] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, *Quantum amplitude amplification and estimation*, Quantum Computation and Information (S. J. Lomonaco and H. E. Brandt, eds.), AMS Contemporary Mathematics Series, vol. 305, AMS, Providence, RI, 2002, quant-ph/0005055.

[34] M. J. Bremner, J. L. Dodd, M. A. Nielsen, and D. Bacon, *Fungible dynamics: there are only two types of entangling multiple-qubit interactions*, Phys. Rev. A **69** (2004), 012313, quant-ph/0307148.

[35] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, *Quantum annealing of a disordered magnet*, Science **284** (1999), 779–781, cond-mat/0105238.

[36] K. Brown and A. W. Harrow, personal communication, October 2003.

[37] H. Buhrman, R. Cleve, and A. Wigderson, *Quantum vs. classical communication and computation*, Proc. 30th ACM Symposium on Theory of Computing, pp. 63–68, 1998, quant-ph/9802040.

[38] H. Buhrman, C. Dürr, M. Heiligman, P. Høyer, F. Magniez, M. Santha, and R. de Wolf, *Quantum algorithms for element distinctness*, Proc. 16th IEEE Conference on Computational Complexity, pp. 131–137, 2001, quant-ph/0007016.

[39] H. Chen, *Necessary conditions for the efficient simulation of Hamiltonians using local unitary operations*, Quantum Information and Computation **3** (2003), 249–257, quant-ph/0109115.

[40] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, *Exponential algorithmic speedup by quantum walk*, Proc. 35th ACM Symposium on Theory of Computing, pp. 59–68, 2003, quant-ph/0209131.

[41] A. M. Childs, E. Deotto, E. Farhi, J. Goldstone, S. Gutmann, and A. J. Landahl, *Quantum search by measurement*, Phys. Rev. A **66** (2002), 032314, quant-ph/0204013.

[42] A. M. Childs and J. M. Eisenberg, *Quantum algorithms for subset finding*, quant-ph/0311038.

[43] A. M. Childs, E. Farhi, J. Goldstone, and S. Gutmann, *Finding cliques by quantum adiabatic evolution*, Quantum Information and Computation **2** (2002), 181–191, quant-ph/0012104.

[44] A. M. Childs, E. Farhi, and S. Gutmann, *An example of the difference between quantum and classical random walks*, Quantum Information Processing **1** (2002), 35–43, quant-ph/0103020.

[45] A. M. Childs, E. Farhi, and J. Preskill, *Robustness of adiabatic quantum computation*, Phys. Rev. A **65** (2002), 012322, quant-ph/0108048.

[46] A. M. Childs and J. Goldstone, *Spatial search by quantum walk*, quant-ph/0306054, submitted to Phys. Rev. A.

[47] ———, *Spatial search and the Dirac equation*, manuscript in preparation.

[48] A. M. Childs, D. W. Leung, and J. A. Smolin, unpublished.

[49] A. M. Childs, D. W. Leung, F. Verstraete, and G. Vidal, *Asymptotic entanglement capacity of the Ising and anisotropic Heisenberg interactions*, Quantum Information and Computation **3** (2003), 97–105, quant-ph/0207052.

[50] A. M. Childs, D. W. Leung, and G. Vidal, *Reversible simulation of bipartite product Hamiltonians*, to appear in IEEE Trans. Inf. Theory **50** (2004), quant-ph/0303097.

[51] J. I. Cirac and P. Zoller, *Quantum computations with cold trapped ions*, Phys. Rev. Lett. **74** (1995), 4091–4094.

[52] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, *Quantum algorithms revisited*, Proc. Roy. Soc. London A **454** (1998), 339–354, quant-ph/9708016.

[53] A. Cohen and A. Wigderson, *Dispersers, deterministic amplification, and weak random sources*, Proc. 30th IEEE Symposium on Foundations of Computer Science, pp. 14–19, 1989.

[54] D. Coppersmith, *An approximate Fourier transform useful in quantum factoring*, Tech. Report RC 19642, IBM Research Division, Yorktown Heights, NY, 1994, quant-ph/0201067.

[55] D. G. Cory, A. F. Fahmy, and T. F. Havel, *Ensemble quantum computing by NMR spectroscopy*, Proc. Natl. Acad. Sci. **94** (1997), 1634–1639.

[56] M. Creutz, *Quarks, gluons, and lattices*, Cambridge University Press, Cambridge, 1983.

[57] W. van Dam, *A universal quantum cellular automaton*, Proc. PhysComp96 (T. Toffoli, M. Biafore, and J.Leão, eds.), pp. 323–331, 1996.

[58] W. van Dam, S. Hallgren, and L. Ip, *Quantum algorithms for some hidden shift problems*, Proc. ACM-SIAM Symposium on Discrete Algorithms, pp. 489–498, 2002, quant-ph/0211140.

[59] W. van Dam, M. Mosca, and U. Vazirani, *How powerful is adiabatic quantum computation?*, Proc. 42nd IEEE Symposium on Foundations of Computer Science, pp. 279–287, 2001, quant-ph/0206003.

[60] E. B. Davies, *Markovian master equations*, Comm. Math. Phys. **39** (1974), 91–110.

[61] E. B. Davies and H. Spohn, *Open quantum systems with time-dependent Hamiltonians and their linear response*, J. Stat. Phys. **19** (1978), 511–523.

[62] J. N. de Beaudrap, R. Cleve, and J. Watrous, *Sharp quantum vs. classical query complexity separations*, Algorithmica **34** (2002), 449–461, quant-ph/0011065.

[63] D. Deutsch, *Quantum theory, the Church-Turing principle, and the universal quantum computer*, Proc. Roy. Soc. London A **400** (1985), 97–117.

[64] ———, *Quantum computational networks*, Proc. Roy. Soc. London A **425** (1989), 73–90.

[65] D. Deutsch and R. Jozsa, *Rapid solution of problems by quantum computation*, Proc. Roy. Soc. London A **439** (1992), 553–558.

[66] P. A. M. Dirac, *The quantum theory of the electron*, Proc. Roy. Soc. London A **117** (1928), 610–624.

[67] D. P. DiVincenzo, *Two-bit gates are universal for quantum computation*, Phys. Rev. A **51** (1995), 1015–1022, cond-mat/9407022.

[68] J. L. Dodd, M. A. Nielsen, M. J. Bremner, and R. Thew, *Universal quantum computation and simulation using any entangling Hamiltonian and local unitaries*, Phys. Rev. A **65** (2002), 040301(R), quant-ph/0106064.

[69] R. Dümcke and H. Spohn, *The proper form of the generator in the weak coupling limit*, Z. Phys. B **34** (1979), 419–422.

[70] W. Dür, G. Vidal, J. I. Cirac, N. Linden, and S. Popescu, *Entanglement capabilities of non-local Hamiltonians*, Phys. Rev. Lett. **87** (2001), 137901, quant-ph/0006034.

[71] C. Dürr, M. Heiligman, P. Høyer, and M. Mhalla, *Quantum query complexity of some graph problems*, quant-ph/0401091.

[72] A. Ekert, *Quantum cryptography based on Bell's theorem*, Phys. Rev. Lett. **67** (1991), 661–663.

[73] A. Ekert, M. Ericsson, P. Hayden, H. Inamori, J. A. Jones, D. K. L. Oi, and V. Vedral, *Geometric quantum computation*, J. Mod. Opt. **47** (2000), 2501–2513, quant-ph/0004015.

[74] P. Erdős, P. Frankl, and Z. Füredi, *Familes of finite sets in which no set is covered by the union of r others*, Israel J. Math. **51** (1985), 79–89.

[75] R. R. Ernst, G. Bodenhausen, and A. Wokaun, *Principles of nuclear magnetic resonance in one and two dimensions*, Oxford University Press, Oxford, 1994.

[76] Euclid, *Elements*, Cambridge University Press, Cambridge, 1908, original version c. 300 BC, translated by Thomas L. Heath.

[77] E. Farhi, J. Goldstone, and S. Gutmann, *A numerical study of the performance of a quantum adiabatic evolution algorithm for satisfiability*, quant-ph/0007071.

[78] ———, *Quantum adiabatic evolution algorithms with different paths*, quant-ph/0208135.

[79] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, *A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem*, Science **292** (2001), 472–475, quant-ph/0104129.

[80] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, *Quantum computation by adiabatic evolution*, quant-ph/0001106.

[81] E. Farhi and S. Gutmann, *Analog analogue of a digital quantum computation*, Phys. Rev. A **57** (1998), 2403–2406, quant-ph/9612026.

[82] ———, *Quantum computation and decision trees*, Phys. Rev. A **58** (1998), 915–928, quant-ph/9706062.

[83] S. A. Fenner and Y. Zhang, *A note on the classical lower bound for a quantum walk algorithm*, quant-ph/0312230.

[84] R. P. Feynman, *Simulating physics with computers*, Int. J. Theor. Phys. **21** (1982), 467–488.

[85] ———, *Quantum mechanical computers*, Optics News **11** (1985), 11–20.

[86] Fortschr. Phys. **48**, no. 9–11 (2000), Special issue on experimental proposals for quantum computation.

[87] M. H. Freedman, A. Yu. Kitaev, M. J. Larsen, and Z. Wang, *Topological quantum computation*, Bull. Amer. Math. Soc. **40** (2003), 31–38, quant-ph/0101025.

[88] M. H. Freedman, A. Yu. Kitaev, and Z. Wang, *Simulation of topological field theories by quantum computers*, Comm. Math. Phys. **227** (2002), 587–603, quant-ph/0001071.

[89] M. H. Freedman, M. J. Larsen, and Z. Wang, *A modular functor which is universal for quantum computation*, Comm. Math. Phys. **227** (2002), 605–622, quant-ph/0001108.

[90] H. Gerhardt, *Continous-time quantum walks on the symmetric group*, Master's thesis, University of Calgary, Calgary, Alberta, December 2003.

[91] H. Gerhardt and J. Watrous, *Continuous-time quantum walks on the symmetric group*, Proc. RANDOM-APPROX (Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, eds.), Lecture Notes in Computer Science, vol. 2764, pp. 290–301, Springer-Verlag, 2003, quant-ph/0305182.

[92] N. Gershenfeld and I. L. Chuang, *Bulk spin-resonance quantum computation*, Science **275** (1997), 350–356.

[93] A. V. Goldberg and S. A. Plotkin, *Efficient parallel algorithms for* $(\delta + 1)$*-coloring and maximal independent set problems*, Proc. 19th ACM Symposium on Theory of Computing, pp. 315–324, 1987.

[94] J. Goldstone, personal communication, September 2002.

[95] ———, personal communication, January 2003.

[96] D. Gottesman, *Fault-tolerant quantum computation with higher-dimensional systems*, Proc. 1st NASA International Conference on Quantum Computing and Quantum Communications (C. P. Williams, ed.), Springer-Verlag, 1999, quant-ph/9802007.

[97] M. Grigni, L. Schulman, M. Vazirani, and U. Vazirani, *Quantum mechanical algorithms for the nonabelian hidden subgroup problem*, Proc. 33rd ACM Symposium on Theory of Computing, pp. 68–74, 2001.

[98] L. K. Grover, *Quantum mechanics helps in searching for a needle in a haystack*, Phys. Rev. Lett. **79** (1997), 325–328, quant-ph/9706033.

[99] S. Hallgren, *Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem*, Proc. 34th ACM Symposium on Theory of Computing, pp. 653–658, 2002.

[100] S. Hallgren, A. Russell, and A. Ta-Shma, *Normal subgroup reconstruction and quantum computation using group representations*, Proc. 32nd ACM Symposium on Theory of Computing, pp. 627–635, 2000.

[101] K. Hammerer, G. Vidal, and J. I. Cirac, *Characterization of non-local gates*, Phys. Rev. A **66** (2002), 062321, quant-ph/0205100.

[102] A. W. Harrow and H.-K. Lo, *A tight lower bound on the classical communication cost of entanglement dilution*, IEEE Trans. Inf. Theory **50** (2004), 319–327, quant-ph/0204096.

[103] P. Hayden and A. Winter, *On the communication cost of entanglement transformations*, Phys. Rev. A **67** (2003), 012326, quant-ph/0204092.

[104] M. Hillery, J. Bergou, and E. Feldman, *Quantum walks based on an interferometric analogy*, Phys. Rev. A **68** (2003), 032314, quant-ph/0302161.

[105] T. Hogg, *Quantum search heuristics*, Phys. Rev. A **61** (2000), 052311.

[106] ———, *Adiabatic quantum computing for random satisfiability problems*, Phys. Rev. A **67** (2003), 022314, quant-ph/0206059.

[107] P. Høyer, M. Mosca, and R. de Wolf, *Quantum search on bounded-error inputs*, Proc. 30th International Colloquium on Automata, Languages, and Programming, Lecture Notes in Computer Science, vol. 2719, pp. 291–299, 2003, quant-ph/0304052.

[108] R. Impagliazzo and D. Zuckerman, *How to recycle random bits*, Proc. 30th IEEE Symposium on Foundations of Computer Science, pp. 222–227, 1989.

[109] G. Ivanyos, F. Magniez, and M. Santha, *Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem*, Proc. 13th ACM Symposium on Parallelism in Algorithms and Architectures, pp. 263–270, 2001, quant-ph/0102014.

[110] D. Janzing, P. Wocjan, and T. Beth, *Complexity of decoupling and time-reversal for n spins with pair-interactions: Arrow of time in quantum control*, Phys. Rev. A **66** (2002), 042311, quant-ph/0106085.

[111] J. A. Jones and E. Knill, *Efficient refocussing of one spin and two spin interactions for NMR quantum computation*, Journal of Magnetic Resonance **141** (1999), 322–325, quant-ph/9905008.

[112] T. Kadowaki and H. Nishimori, *Quantum annealing in the transverse Ising model*, Phys. Rev. E **58** (1998), 5355–5363, cond-mat/9804280.

[113] W. M. Kaminsky and S. Lloyd, *Scalable architecture for adiabatic quantum computing of NP-hard problems*, Quantum Computing and Quantum Bits in Mesoscopic Systems (Anthony Leggett, Berardo Ruggiero, and Paolo Silvestrini, eds.), Kluwer, 2002, quant-ph/0211152.

[114] W. M. Kaminsky, S. Lloyd, and T. P. Orlando, *Scalable superconducting architecture for adiabatic quantum computation*, quant-ph/0403090.

[115] T. Kato, *On the adiabatic theorem of quantum mechanics*, Phys. Soc. Jap. **5** (1950), 435–439.

[116] J. Kempe, *Quantum random walks hit exponentially faster*, Proc. 7th International Workshop on Randomization and Approximation Techniques in Computer Science, pp. 354–369, 2003, quant-ph/0205083.

[117] N. Khaneja, R. Brockett, and S. J. Glaser, *Time optimal control in spin systems*, Phys. Rev. A **63** (2001), 032308, quant-ph/0006114.

[118] A. Yu. Kitaev, *Quantum measurements and the abelian stabilizer problem*, quant-ph/9511026.

[119] ______, *Fault-tolerant quantum computation by anyons*, quant-ph/9707021.

[120] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and quantum computation*, AMS, Providence, RI, 2002.

[121] B. Kraus and J. I. Cirac, *Optimal creation of entanglement using a two-qubit gate*, Phys. Rev. A **63** (2001), 062309, quant-ph/0011050.

[122] R. Landauer, *Information is physical*, Physics Today **44** (1991), 23–29.

[123] M. S. Leifer, L. Henderson, and N. Linden, *Optimal entanglement generation from quantum operations*, Phys. Rev. A **67** (2003), 012306, quant-ph/0205055.

[124] D. W. Leung, *Simulation and reversal of n-qubit Hamiltonians using Hadamard matrices*, J. Mod. Opt. **49** (2002), 1199–1217, quant-ph/0107041.

[125] ______, *Quantum computation by measurements*, International Journal of Quantum Information **2** (2004), 33–43, quant-ph/0310189.

[126] D. W. Leung, I. L. Chuang, F. Yamaguchi, and Y. Yamamoto, *Efficient implementation of selective recoupling in heteronuclear spin systems using Hadamard matrices*, Phys. Rev. A **61** (2000), 042310, quant-ph/9904100.

[127] G. Lindblad, *Non-equilibrium entropy and irreversibility*, Reidel, Dordrecht, 1983.

[128] N. Linial, *Distributive graph algorithms—global solutions from local data*, Proc. 28th IEEE Symposium on Foundations of Computer Science, pp. 331–335, 1987.

[129] ______, *Locality in distributed graph algorithms*, SIAM J. Comput. **21** (1992), 193–201.

[130] S. Lloyd, *Universal quantum simulators*, Science **273** (1996), 1073–1078.

[131] ———, *Quantum computation with abelian anyons*, Quantum Information Processing **1** (2002), 13–18, quant-ph/0004010.

[132] H.-K. Lo, *Classical-communication cost in distributed quantum-information processing: A generalization of quantum-communication complexity*, Phys. Rev. A **62** (2000), 012313, quant-ph/9912009.

[133] H.-K. Lo and H. F. Chau, *Unconditional security of quantum key distribution over arbitrarily long distances*, Science **283** (1999), 2050–2056, quant-ph/9803006.

[134] H.-K. Lo and S. Popescu, *The classical communication cost of entanglement manipulation: Is entanglement an inter-convertible resource?*, Phys. Rev. Lett. **83** (1999), 1459–1462, quant-ph/9902045.

[135] D. Loss and D. P. DiVincenzo, *Quantum computation with quantum dots*, Phys. Rev. A **57** (1998), 120–126, cond-mat/9701055.

[136] F. Magniez, M. Santha, and M. Szegedy, *An $\tilde{O}(n^{1.3})$ quantum algorithm for the triangle problem*, quant-ph/0310134.

[137] Yu. Manin, *Computable and uncomputable*, Sovetskoye Radio, 1980.

[138] N. Margolus, *Parallel quantum computation*, Complexity, Entropy, and the Physics of Information (W. H. Zurek, ed.), Addison-Wesley, Redwood City, 1990, pp. 273–287.

[139] Ll. Masanes, G. Vidal, and J. I. Latorre, *Time-optimal Hamiltonian simulation and gate synthesis using homogeneous local unitaries*, Quantum Information and Computation **2** (2002), 285–296, quant-ph/0202042.

[140] J. McBride, *An evaluation of the performance of the quantum adiabatic algorithm on random instances of k-SAT*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, May 2002.

[141] C. J. H. McDiarmid, *On a random recoloring method for graphs and hypergraphs*, Combin. Probab. Comput. **2** (1993), 363–365.

[142] A. Messiah, *Quantum mechanics*, vol. II, North-Holland, Amsterdam, 1961.

[143] D. A. Meyer, *From quantum cellular automata to quantum lattice gasses*, J. Stat. Phys. **85** (1996), 551–574, quant-ph/9604003.

[144] ———, *On the absence of homogeneous scalar unitary cellular automata*, Phys. Lett. A **223** (1996), 337–340, quant-ph/9604011.

[145] E. W. Montroll, *Random walks in multidimensional spaces, especially on periodic lattices*, J. Soc. Indust. Appl. Math. **4** (1956), 241–260.

[146] ———, *Random walks on lattices. III. Calculation of first-passage times with applications to exciton trapping on photosynthetic units*, J. Math. Phys. **10** (1969), 753–765.

[147] J. E. Mooij, T. P. Orlando, L. Levitov, L. Tian, C. H. van der Wal, and S. Lloyd, *Josephson persistent-current qubit*, Science **285** (1999), 1036–1039.

[148] C. Moore and A. Russell, *Quantum walks on the hypercube*, Proc. 6th International Workshop on Randomization and Approximation Techniques in Computer Science (J. D. P. Rolim and S. Vadhan, eds.), Lecture Notes in Computer Science, vol. 2483, pp. 164–178, Springer-Verlag, 2002, quant-ph/0104137.

[149] M. Mosca and A. Ekert, *The hidden subgroup problem and eigenvalue estimation on a quantum computer*, Proc. 1st NASA International Conference on Quantum Computing and Quantum Communication, Lecture Notes in Computer Science, vol. 1509, Springer-Verlag, 1999, quant-ph/9903071.

[150] J. von Neumann, *Mathematical foundations of quantum mechanics*, Princeton University Press, Princeton, NJ, 1955, original version 1932, translated by Robert T. Beyer.

[151] M. A. Nielsen, *Quantum computation by measurement and quantum memory*, Phys. Lett. A **308** (2003), 96–100, quant-ph/0108020.

[152] M. A. Nielsen, M. J. Bremner, J. L. Dodd, A. M. Childs, and C. M. Dawson, *Universal simulation of Hamiltonian dynamics for quantum systems with finite-dimensional state spaces*, Phys. Rev. A **66** (2002), 022317, quant-ph/0109064.

[153] W. Ogburn and J. Preskill, *Topological quantum computation*, Lecture Notes in Computer Science, vol. 1509, Springer-Verlag, Berlin, 1999, pp. 341–356.

[154] C. H. Papadimitriou, *On selecting a satisfying truth assignment*, Proc. 32nd IEEE Symposium on Foundations of Computer Science, pp. 163–169, 1991.

[155] A. Peres, *Quantum theory: Concepts and methods*, Kluwer, Dordrecht, 1995.

[156] J. Preskill, *Fault-tolerant quantum computation*, Introduction to Quantum Computation and Information (H.-K. Lo, S. Popescu, and T. Spiller, eds.), World Scientific, Singapore, 1998, quant-ph/9712048.

[157] R. Raussendorf and H. J. Briegel, *A one-way quantum computer*, Phys. Rev. Lett. **86** (2001), 5188–5191, quant-ph/0010033.

[158] J. Roland and N. J. Cerf, *Quantum search by local adiabatic evolution*, Phys. Rev. A **65** (2002), 042308, quant-ph/0107015.

[159] ———, *Quantum circuit implementation of the Hamiltonian versions of Grover's algorithm*, Phys. Rev. A **68** (2003), 062311, quant-ph/0302138.

[160] U. Schöning, *A probabilistic algorithm for k-SAT and constraint satisfaction problems*, Proc. 40th IEEE Symposium on Foundations of Computer Science, pp. 17–19, 1999.

[161] L. S. Schulman, A. Ranfagni, and D. Mugnai, *Characteristic scales for dominated time evolution*, Physica Scripta **49** (1994), 536–542.

[162] S. Severini, *On the digraph of a unitary matrix*, SIAM J. Matrix Anal. Appl. **25** (2003), 295–300, math.CO/0205187.

[163] C. E. Shannon, *A mathematical theory of communication*, Bell Syst. Tech. J. **27** (1948), 379–423, 623–656.

[164] N. Shenvi, J. Kempe, and K. B. Whaley, *A quantum random walk search algorithm*, Phys. Rev. A **67** (2003), 052307, quant-ph/0210064.

[165] P. W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proc. 35th IEEE Symposium on Foundations of Computer Science (S. Goldwasser, ed.), pp. 124–134, IEEE Press, 1994, quant-ph/9508027.

[166] ——, *Scheme for reducing decoherence in quantum computer memory*, Phys. Rev. A **52** (1995), 2493–2496.

[167] ——, *Fault-tolerant quantum computation*, Proc. 37th IEEE Symposium on Foundations of Computer Science, pp. 56–65, 1996, quant-ph/9605011.

[168] D. Simon, *On the power of quantum computation*, Proc. 35th IEEE Symposium on Foundations of Computer Science, pp. 116–123, 1994.

[169] A. Sinclair, *Algorithms for random generation and counting: A Markov chain approach*, Birkhauser, Boston, 1993.

[170] A. Steane, *Error correcting codes in quantum theory*, Phys. Rev. Lett. **77** (1996), 793–797.

[171] M. Suzuki, *General theory of higher-order decomposition of exponential operators and symplectic integrators*, Phys. Lett. A **165** (1992), 387–395.

[172] M. Szegedy, *Spectra of quantized walks and a $\sqrt{\delta\epsilon}$ rule*, quant-ph/0401053.

[173] M. Szegedy and S. Vishwanathan, *Locality based graph coloring*, Proc. 25th ACM Symposium on Theory of Computing, pp. 201–207, 1993.

[174] B. C. Travaglione, G. J. Milburn, and T. C. Ralph, *Phase estimation as a quantum nondemolition measurement*, quant-ph/0203130.

[175] G. Vidal and J. I. Cirac, *Catalysis in non-local quantum operations*, Phys. Rev. Lett. **88** (2002), 167903, quant-ph/0108077.

[176] ——, *Optimal simulation of nonlocal Hamiltonians using local operations and classical communication*, Phys. Rev. A **66** (2002), 022315, quant-ph/0108076.

[177] G. Vidal, K. Hammerer, and J. I. Cirac, *Interaction cost of non-local gates*, Phys. Rev. Lett. **88** (2002), 237902, quant-ph/0112168.

[178] V. G. Vizing, *On an estimate of the chromatic class of a p-graph*, Diskret. Analiz **3** (1964), 25–30.

[179] X. Wang and B. C. Sanders, *Entanglement capability of self-inverse Hamiltonian evolution*, Phys. Rev. A **68** (2003), 014301, quant-ph/0212035.

[180] J. Watrous, *On one-dimensional quantum cellular automata*, Proc. 36th IEEE Symposium on Foundations of Computer Science, pp. 528–537, 1995.

[181] ——, *Quantum algorithms for solvable groups*, Proc. 33rd ACM Symposium on Theory of Computing, pp. 60–67, 2001, quant-ph/0011023.

[182] ———, *Quantum simulations of classical random walks and undirected graph connectivity*, J. Computer and System Sciences **62** (2001), 376–391, cs.CC/9812012.

[183] G. N. Watson, *Three triple integrals*, Quart. J. Math. Oxford Scr. **10** (1939), 266–276.

[184] S. Wiesner, *Simulations of many-body quantum systems by a quantum computer*, quant-ph/9603028.

[185] A. Wigderson, *The complexity of graph connectivity*, Proc. 17th Mathematical Foundations of Computer Science Conf., Lecture Notes in Computer Science, vol. 629, pp. 112–132, 1992.

[186] P. Wocjan, D. Janzing, and T. Beth, *Simulating arbitrary pair-interactions by a given Hamiltonian: Graph-theoretical bounds on the time complexity*, Quantum Information and Computation **2** (2002), 117–132, quant-ph/0106077.

[187] P. Wocjan, M. Rotteler, D. Janzing, and T. Beth, *Universal simulation of Hamiltonians using a finite set of control operations*, Quantum Information and Computation **2** (2002), 133–150, quant-ph/0109063.

[188] A. C.-C. Yao, *Quantum circuit complexity*, Proc. 34th IEEE Symposium on Foundations of Computer Science, pp. 352–361, 1993.

[189] C. Zalka, *Simulating quantum systems on a quantum computer*, Proc. Roy. Soc. London A **454** (1998), 313–322, quant-ph/9603026.

[190] P. Zanardi and M. Rasetti, *Holonomic quantum computation*, Phys. Lett. A **264** (1999), 94–99, quant-ph/9904011.

[191] P. Zanardi, C. Zalka, and L. Faoro, *On the entangling power of quantum evolutions*, Phys. Rev. A **62** (2000), 030301(R), quant-ph/0005031.

Index