Quantum Algorithms and Complexity for Numerical Problems

Chi Zhang

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2011

ABSTRACT

Quantum Algorithms and Complexity for Numerical Problems

Chi Zhang

Quantum computing has attracted a lot of attention in different research fields, such as mathematics, physics and computer science. Quantum algorithms can solve certain problems significantly faster than classical algorithms. There are many numerical problems, especially those arising from quantum systems, which are notoriously difficult to solve using classical computers, since the computational time required often scales exponentially with the size of the problem. However, quantum computers have the potential to solve these problems efficiently, which is also one of the founding ideas of the field of quantum computing.

In this thesis, we explore five different computational problems, designing innovative quantum algorithms and studying their computational complexity.

First, we design an adiabatic quantum algorithm for the counting problem, i.e., approximating the proportion α, of the marked items in a given database. As the quantum system undergoes a designed cyclic adiabatic evolution, it acquires a Berry phase $2\pi\alpha$. By estimating the Berry phase, we can approximate α, and solve the problem. For an error bound ε, the algorithm can solve the problem with cost of order $\varepsilon^{-3/2}$, which is not as good as the optimal algorithm in the quantum circuit model, but better than the classical random algorithm. Moreover, since the Berry phase is a purely geometric feature, the result should be robust to decoherence and resilient to certain kinds of noise. Since the counting problem is the foundation of many other numerical problems, such as high-dimensional integration and path integration, our adiabatic algorithms can be directly generalized to these kinds of problems.

In addition, we study the quantum PAC learning model, offering an improved lower

bound on the query complexity. For a concept class with d-VC dimension, the lower bound is $\Omega(\varepsilon^{-1}(d^{1-\eta} + \log(1/\delta)))$, where ε is the required error bound, δ is the maximal failure possibility and η can be an arbitrarily small positive number. The lower bound is close to the best lower bound on query complexity known for the classical PAC learning model, which is $\Omega(\varepsilon^{-1}(d + \log(1/\delta)))$.

We also study the algorithms and the cost of simulating a system evolving with Hamiltonian $H = \sum_{j=1}^{m} H_j$, where the evolution of H_j can be implemented efficiently. We consider high order splitting methods that are particularly applicable in quantum simulation and obtain bounds on the number of exponentials required to approximate e^{-iHt} with error ε. Moreover, we derive the optimal order of convergence, given ε and the cost of the resulting algorithm. We compare our complexity estimates to previously known ones and show the resulting speedup.

Furthermore, we consider *randomized* algorithms for simulating the evolution of Hamiltonian H. The evolution is simulated by a product of exponentials of H_j in a random sequence and random evolution times. Hence the final state of the system is approximated by a mixed quantum state. First we provide a scheme to bound the error of the final quantum state in a randomized algorithm. Then we obtain randomized algorithms which have the same efficiency as certain deterministic algorithms but which are simpler to implement. Finally we provide a lower bound on the number of exponentials for both deterministic and randomized algorithms, when the evolution time is required to be positive.

We also apply the improved upper bound of Hamiltonian simulation in estimating the ground state energy of a multiparticle system with relative error ε, which is also known as the multivariate Sturm-Liouville eigenvalue problem. Since the cost of this problem grows exponentially with the number of particles using deterministic classical algorithms, it suffers from the curse of dimensionality. Quantum computers can vanquish the curse, and we exhibit a quantum algorithm that achieves relative error ε using $O(d \log \varepsilon^{-1})$ qubits with total cost (number of quantum queries and other quantum operations) $O(d\, \varepsilon^{-(3+\delta)})$, where $\delta > 0$ is arbitrarily small. Thus, the number of qubits and the total cost are linear in the number of particles.

The main result of Chapter 2 is based on the paper [127], published in Quantum Infor-

mation Proceeding. The result of Chapter 3 is the same as that of the paper [126], published in Information Processing Letters. The results of Chapter 4 and Chapter 6 have been submitted, and can be found in [88] and [84] separately. Chapter 5 from a talk in the 9th International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, has also been submitted and can be found in [125].

Table of Contents

Acknowledgments

First of all, I would like to thank my PhD advisors Anargyros Papageorgiou and Joseph F. Traub. It has been a great experience for me to work with them. With their guidance I have learned many things not only about theory but also about how to be a researcher.

I also would like to express my gratitude to Alfred V. Aho, Arthur G. Werschulz, Henryk Woźniakowski, my thesis committee members, who devoted their time and energy to this research.

I would like to thank Anargyros Papageorgiou for his efforts in our joint papers. By working with him, I gained a lot of experience in research and study. I also would like to thank Rocco Servedio, for his great advice and helpful discussions.

I am grateful to my colleagues from Department of Computer Science at Columbia University, for sharing academic experiences and many social activities.

Chapter 1

Introduction

In the early 1980s, Richard Feynman observed that computers built from quantum mechanical components would be ideally suited for simulating quantum mechanics. The perspective of quantum systems as information processing devices subsequently led to identification of concrete tasks, for which quantum computers have a quantifiable advantage. Numerical problems play a key role in science and engineering applications. By studying quantum algorithms and complexity for these problems, we try to exploit the superiority of quantum computers. In this thesis, we provide quantum algorithms that can significantly improve our ability in solving practical problems, such as the counting problem, PAC learning, simulation of composite systems and ground energy estimation.

In Chapter 2, we study adiabatic computing and provide an adiabatic algorithm for the counting problem. We can use this algorithm to solve other numerical problems, such as high-dimensional integrals and path integrals. In Chapter 3, we show that quantum computers cannot significantly reduce the required number of queries in the PAC learning model. In Chapter 4 and Chapter 5, we study the number of exponentials in Hamiltonian simulation, and provide several randomized algorithms for simulating the evolution of quantum systems. In Chapter 6, we work on how to apply Hamiltonian simulation to estimate the ground state energy of multiparticle systems.

1.1 Overview of the Introduction

Most continuous mathematical problems arising in science and engineering can only be solved numerically and therefore approximately. In this thesis, we often measure the "hardness" of these numerical problems using the tools of information-based complexity (IBC), a complexity theory that has been very successful in proving upper and lower bounds for numerous problems in scientific computing. We give an overview of the basic concepts of IBC in Section 1.2.

Quantum computing is a novel research field, studying algorithm design and analyzing complexity based on quantum computers, devices for computation that make direct use of quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. In Section 1.3, we introduce preliminaries of quantum mechanics, as the foundation of quantum computing.

Quantum computing takes advantage of the inherent parallelism, entanglement, interference and the analog nature of quantum mechanics. There are many quantum algorithms which can solve certain problems significantly faster than classical algorithms. We will give a brief introduction of quantum computing in Section 1.4.

Adiabatic computing is a model of quantum computing, which differs from the quantum circuit model. It is based on the adiabatic theorem in quantum mechanics. It is believed that the adiabatic model enjoys inherent robustness against decoherence. Because of this, adiabatic computing has attracted considerable attention. We introduce this model in Section 1.5.

Quantum algorithms also have wide applications in learning theory, which we will introduce in Section 1.6, particulary the PAC (Probably Approximately Correct) learning model.

In Section 1.7, we describe the problem of Hamiltonian simulation, and in Section 1.2.2, we describe the ground energy problem. They play key roles in quantum mechanics, and very difficult to solve by using classical computers. In this thesis, we show quantum algorithms which can efficiently solve these problems.

1.2 Information-Based Complexity

Information-based complexity (IBC) provides a comprehensive tool for analyzing the cost of solving continuous problems. Overviews of information-based complexity may be found in [81; 108; 109; 110].

In IBC a problem is given by a class of functions $F = \{f : D \to K\}$, a solution operator $S : F \to G$ and norms on F and G. Our goal is to approximate $S(f)$, i.e., we want to compute an approximation $\tilde{S}(f)$ whose accuracy is given by $\|S(f) - \tilde{S}(f)\|$, and is required within ε. The problem will be completely specified once we have fixed the domain and the range of S, as well as an error criterion for the approximation to the solution.

In order to analyze the complexity of the problem, we break it into two parts. First, we study the *information complexity* of approximating $S(f)$. We assume that the input f is given to us as a black box. We can evaluate f at points $x \in D$ and obtain $f(x)$. The information complexity represents how many evaluations we need such that an algorithm is at least in principle able to give an ε-accurate answer, even though the actual construction of the answer might still require a large number of computational steps. Secondly, we consider the total number of *computational steps* (including the evaluations of f). Obviously the information complexity gives a lower bound for the complexity. When the information cost dominates the computational cost of the algorithm (which happens often), the information cost also gives a good upper bound.

There are many important scientific, engineering and financial problems having continuous formulations, including high-dimensional integration [54; 55; 58; 59], path integration [85; 111], eigenvalue approximation [62; 83; 86], partial differential equations [30; 56; 57] and continuous optimization. In this thesis, we mainly focus on two continuous problems: the counting problem [22; 78] and the Sturm-Liouville eigenvalue problem [83; 86].

1.2.1 Counting Problem

The goal of the counting problem is to approximate the proportion α of marked items in an N-item database. In classical computation, the counting problem needs $O(N)$ evaluations

in the worst case setting, and $O(\varepsilon^{-2})$ in the randomized setting, for error ε, by using the Monte Carlo methods [75]. There exists a quantum algorithm solving the problem in $O(\varepsilon^{-1})$ evaluations [22]. Moreover, Nayak and Wu [78] showed that the quantum algorithm given in [22] is optimal in the quantum circuit model.

The counting problem is also central for many other continuous problems, such as high dimensional integration, path integration and eigenvalue approximation [54; 59].

For example, consider the problem of approximating the integration

$$I(f) = \int_{[0,1]^d} f(x)\, dx,$$

where $f \in F_d$, $F_d = \left\{ f : [0,1]^d \to R \mid \text{continuous and } |f(x)| \leq 1,\ x \in [0,1]^d \right\}$. Furthermore, assume that integrand class has *smoothness* r [110]. The query complexity in the worst case setting is

$$n(\varepsilon) = \Theta(\varepsilon^{-d/r}),$$

which leads to the *curse of dimensionality*. The curse can be broken by the Monte Carlo algorithm [25; 45] in the randomized setting, and yields the query complexity

$$n(\varepsilon) = \Theta(\varepsilon^{-\frac{1}{1/2+r/d}}).$$

In the quantum setting, the Monte Carlo algorithm can be replaced by the quantum counting algorithm, which yields a better query complexity

$$n(\varepsilon) = \Theta(\varepsilon^{-\frac{1}{1+r/d}}).$$

For more details see [85].

In this thesis, we consider an adiabatic algorithm for solving the counting problem. The cost of the algorithm is not as good as the optimal quantum algorithm in the circuit model, but is better than the cost of the best classical algorithm known. It will be discussed in Chapter 2.

1.2.2 Sturm-Liouville eigenvalue problem

The goal of the Sturm-Liouville eigenvalue problem is to approximate the smallest eigenvalue of $-\Delta + q$ with homogeneous Dirichlet boundary conditions on the d-dimensional unit

cube, where Δ is the Laplacian, and q is nonnegative and has continuous first order partial derivatives. We are interested in the minimal number $n(\varepsilon)$ of function evaluations or queries that are necessary to compute an ε-approximation to the smallest eigenvalue. This problem has been extensively studied both in classical settings and quantum settings [83; 86; 87].

Let $I_d = [0,1]^d$ and consider the class of functions

$$Q = \left\{ q : I_d \to [0,1] | q, D_j q := \frac{\partial q}{\partial x_j} \in C(I_d), \|D_j q\|_\infty \leq 1, \|q\|_\infty \leq 1 \right\},$$

where $\|\cdot\|_\infty$ denotes the supremum norm. For $q \in Q$, define $L_q := -\Delta + q$, where $\Delta = \sum_{j=1}^d \partial^2/\partial x_j^2$ is the Laplacian, and consider the eigenvalue problem

$$L_q u = \lambda u, x \in (0,1)^d,$$
$$u(x) = 0, x \in \partial I_d.$$

In the variational form, the smallest eigenvalue is given by

$$\lambda(q) = \min_{0 \neq u \in H_0^1} \frac{\int_{I_d} \sum_{j=1}^d [D_j u(x)]^2 + q(x) u^2(x) dx}{\int_{I_d} u^2(x) dx}.$$

For the eigenvalue problem, there is a perturbation formula relating the eigenvalues $\lambda(q)$ and $\lambda(\bar{q})$ for two functions q and $\bar{q}$, [83]. In particular,

$$\lambda(q) = \lambda(\bar{q}) + \int_{I_d} (q(x) - \bar{q}(x)) u_{\bar{q}}^2(x) dx + O(\|q - \bar{q}\|^2), \tag{1.1}$$

where $u_{\bar{q}}$ is the eigenfuction that corresponds to $\lambda(\bar{q})$. This equation relates the eigenvalue problems to the integration problems.

When $d = 1$, this is the classical univariate Sturm-Liouville eigenvalue problem. For univariate Sturm-Liouville eigenvalue problems, recall that $n(\varepsilon)$ denotes the number of queries needed to obtain an ε-approximation. It is known that

$$n(\varepsilon) = \Theta(\varepsilon^{-1/2})$$

in the (deterministic) worst case setting, and

$$n(\varepsilon) = \Theta(\varepsilon^{-2/5})$$

in the randomized setting; see [86]. In the quantum setting, there are two kinds of queries. One is called the *bit* query, which is similar to the oracle calls used in Grover's algorithm [52].

The other is the *power* query, which is used for a number of problems including phase estimation [86]. They are obtained by considering the propagator of the discretized system at a number of different time moments. Power queries allow us to use powers of the unitary matrix $e^{i\frac{1}{2}M}$, where M is an $n \times n$ matrix obtained from the standard discretization of the Sturm-Liouville differential operator. How to use bit queries to implement power queries is one of the topics in this thesis. To compute an ε-approximation with probability $\frac{3}{4}$ requires

$$n(\varepsilon) = \Theta(\varepsilon^{-1/3})$$

bit queries, or

$$n(\varepsilon) = \Theta(\log \varepsilon^{-1})$$

power queries [19].

For multivariate Sturm-Liouville eigenvalue problems, Papageorgiou [83] derived lower bounds for $n(\varepsilon)$ using known lower bounds for the information complexity of integration, which are

$$n(\varepsilon) = \Omega(\varepsilon^{-d})$$

in the worst case setting,

$$n(\varepsilon) = \Omega(\varepsilon^{-\frac{2d}{d+2}})$$

in the randomized setting, and

$$n(\varepsilon) = \Omega(\varepsilon^{-\frac{d}{d+1}})$$

in the quantum bit query setting.

By discretizing the continuous problem and solving the resulting matrix eigenvalue problem, there is an optimal deterministic algorithm, whose information complexity is

$$n(\varepsilon) = O(\varepsilon^{-d}),$$

and whose computational complexity is

$$\mathrm{comp}(\varepsilon) = \mathrm{O}(\mathbf{c}\,\varepsilon^{-\mathrm{d}} + \varepsilon^{-\mathrm{d}} \log \varepsilon^{-1}),$$

where $\mathbf{c}$ denotes the cost of one function evaluation. The best randomized algorithm known applies the perturbation formula, i.e., first approximate q by a function $\bar{q}$, then approximate

the first terms in the right hand side of Eq.1.1. The eigenvalue $\lambda(\bar{q})$ is approximated by using matrix discretization, and the weighted integration is approximated by using the Monte Carlo method. The information complexity is

$$n(\varepsilon) = O(\varepsilon^{-\max(2/3,d/2)}),$$

and the computational complexity is

$$\mathrm{comp}(\varepsilon) = \mathrm{O}(\mathbf{c}\,\varepsilon^{-\max(2/3,\mathrm{d}/2)} + \varepsilon^{-\mathrm{d}}\log\varepsilon^{-1}).$$

The randomized algorithm is only known to be optimal when $d \leq 2$; it is an open problem whether it is still optimal in the randomized setting for $d > 2$.

The multivariate eigenvalue problems can be solved with any constant probability arbitrarily close to 1 and relative error ε by a quantum algorithm based on the perturbation formula and quantum integration [55], which uses $O(\varepsilon^{-d/2})$ bit queries and $O(\varepsilon^{-d/2})$ classical function evaluations, plus $O(d^2\log^2\varepsilon^{-1})$ quantum operations and $O(\varepsilon^{-2d}\log\varepsilon^{-1})$ classical arithmetic operations. However, a better algorithm can be obtained by phase estimation, which uses

$$O(\varepsilon^{-6}\log^2\varepsilon^{-1})$$

bit queries and $O(d\,\varepsilon^{-6}\log^4\varepsilon^{-1})$ other quantum operations. The quantum algorithm is derived from the algorithm using power queries, which has $O(\log\varepsilon^{-1})$ power queries and $O(\log^2\varepsilon^{-1} + d\log\varepsilon^{-1})$ quantum operations.

In this thesis, we provide a better quantum algorithm for estimating the ground state energy, using $O(d\,\varepsilon^{-(3+\delta)})$ quantum operations, where $\delta > 0$ is arbitrarily small.

1.3 Quantum Mechanics

In this section we give a brief overview of quantum mechanics, for more details see [68; 80; 97]. A *quantum mechanical system* is a complex vector space with inner product (i.e. a Hilbert space) known as the *state space* of the system. The system is completely described by its *state*, a time-dependent function $|\psi\rangle$, which is *normalized* by the condition $\langle\psi|\psi\rangle = 1$.

A *qubit* is the fundamental building block of quantum computers [80]. Its state space is a 2-dimensional complex vector space $\mathcal{H} = C^2$. The state of a qubit is given by

$$|\psi\rangle = c_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + c_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = c_0|0\rangle + c_1|1\rangle,$$

where $|0\rangle$ and $|1\rangle$ are basis vectors in the usual Dirac notation and c_0, c_1 are complex numbers for which $|c_0|^2 + |c_1|^2 = 1$.

The evolution of quantum mechanical systems obeys the time-dependent Schrödinger equation [97]

$$i\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle, \tag{1.2}$$

where $H(t)$ is a Hermitian operator called the *Hamiltonian*. An example is a particle of mass m in a scalar time-independent potential V. If we denote the Laplacian operator by Δ, i.e., $\Delta = \sum_j \partial^2/\partial x_j^2$, then H is a differential operator

$$H = -\frac{\hbar^2}{2m}\Delta + V$$

and the Schrödinger equation is a partial differential equation

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = -\frac{\hbar^2}{2m}\Delta\psi(x,t) + V(x)\psi(x,t),$$

where $x = (x_1, \cdots, x_n) \in \Omega \subset R^n$.

Because the Hamiltonian is Hermitian, the time evolved state $|\psi(t)\rangle$ and the initial state $|\psi(0)\rangle$ are related by a unitary transformation $U_{t,0}$

$$|\psi(t)\rangle = U_{t,0}|\psi(0)\rangle.$$

For a time-independent H, this unitary transformation is

$$U_{t,0} = e^{-iHt},$$

so

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle.$$

When an experimentalist observes a quantum system, it is described by a quantum measurement [80]. Quantum measurements are described by a collection $\{M_m\}$ of measurement

operators, which are acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ before the measurement, then the probability that result m occurs is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle,$$

and the state of the system after the measurement is

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}.$$

The measurement operators satisfy the *completeness equation*,

$$\sum_m M_m^\dagger M_m = I.$$

For some applications the post-measurement state is not of interest. Then, there is a mathematical tool known as the POVM (Positive Operator Valued Measure), which is represented by a set of positive self-adjoint operators $\{\Pi_m\}$, where $\sum_m \Pi_m = I$, and the probability that result m occurs is

$$p(m) = \langle\psi|\Pi_m|\psi\rangle.$$

Quantum systems whose state is not completely known are described by mixed quantum states. More precisely, suppose a quantum system is in one of a number of states $|\psi_i\rangle$, with respective probability p_i, where i is an index. The mixed state for the system is defined by a density operator

$$\rho = \sum_i p_i|\psi_i\rangle\langle\psi_i|.$$

The evolution of the mixed quantum state ρ under an unitary operator U is

$$\rho = \sum_i p_i|\psi_i\rangle\langle\psi_i| \mapsto \sum_i p_i U|\psi_i\rangle\langle\psi_i|U^\dagger = U\rho U^\dagger.$$

In a POVM measurement $\{\Pi_m\}$, the probability of obtaining result m for a mixed quantum state ρ is

$$p(m) = \mathrm{Tr}(\Pi_m\rho).$$

1.4 Quantum Computing

Quantum Computing is a new computing paradigm which is based on "quantum computers" i.e., computing devices based on the principles of quantum mechanics. The concept of quantum computers was first proposed by Feynman for simulating quantum systems [44], which are extremely hard to carry out on classical computers. Interest in quantum computation grew enormously when Shor provided a quantum algorithm to solve the problem of factorization and discrete logarithm [101; 102] in polynomial running time in 1994, while all known classical algorithms take exponential time. In 1996 Grover discovered a quantum algorithm for searching unstructured database [51; 52]. The quantum algorithm can find an item in a database of size N with $O(\sqrt{N})$ queries, while any classical algorithm has to use $\Omega(N)$ queries. By now, there are three main applications of quantum algorithms: the hidden subgroup problem [2; 53; 69; 102], with Shor's factorization algorithm [102] one important example; search problems [23; 52; 54; 78; 82], including problems based on quantum walk [7; 41; 91; 95]; and simulation of quantum systems [14; 17; 64; 76; 89; 124].

Most quantum algorithms are formulated in the quantum circuit model [16; 35; 121]. In the quantum circuit model, the final state $|\psi_f\rangle$ is obtained from the initial state $|\psi_0\rangle$ through a sequence of unitary operations, which are comprised of certain *elementary quantum gates* - unitary operators acting on a small number of qubits. The most commonly used gates [80] are the following. The one-qubit gates are

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$$
$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

which are known as the Hadamard, phase, $\pi/8$, and Pauli X, Y, Z gates, respectively.

Together with the controlled-NOT gate,

$$C_{NOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(which acts on two qubits), they constitute a universal set for quantum computation. That is, any unitary operation can be approximated to arbitrary accuracy by a quantum circuit involving only these gates [37]. The computational cost of a quantum algorithm in the circuit model is defined as the number of elementary gates the algorithm uses.

For numerical problems, a *quantum algorithm* can be described as a sequence of unitary operations, i.e.,

$$|\psi_f\rangle := U_T Q_f U_{T-1} Q_f \cdots U_1 Q_f U_0 |\psi_0\rangle. \tag{1.3}$$

The unitary matrix Q_f is called a *quantum query* and is used to provide information to the algorithm about a function f. Q_f depends on n function evaluations $f(t_1), \ldots, f(t_n)$, at deterministically chosen points, with $n \leq 2^\nu$, where ν is the number of qubits used in the algorithm. The unitary operations $U_0, U_1, \ldots, U_T$ are unitary operations that do not depend on f. The integer T denotes the number of quantum queries [85]. Woźniakowski also consider quantum algorithms with randomized queries, where Q_f depends on randomized function evaluations [120].

For algorithms solving discrete problems, such as Grover's algorithm for the search of an unordered database [52], the input f is considered to be a Boolean function. The most commonly studied quantum query is the *bit* query. For a Boolean function

$$f : \{0, \ldots, 2^m - 1\} \to \{0, 1\},$$

the bit query is defined by

$$Q_f |j\rangle |k\rangle \; = \; |j\rangle |k \oplus f(j)\rangle. \tag{1.4}$$

where $\oplus$ denoting addition modulo 2. For a real function f, the query is constructed by taking the most significant bits of the function f evaluated at some points t_j. More precisely, as in [55], the bit query for f has the form

$$Q_f |j\rangle |k\rangle \; = \; |j\rangle |k \oplus \beta(f(\tau(j)))\rangle, \tag{1.5}$$

where the functions β and τ are used to discretize the domain $\mathcal{D}$ and the range $\mathcal{R}$ of f, respectively.

At the end of the quantum algorithm, the final state $|\psi_f\rangle$ is measured. The measurement produces one of M outcomes. Outcome $j \in \{0, 1, \ldots, M-1\}$ occurs with probability $p_f(j)$, which depends on j and the input f. Knowing the outcome j, we classically compute an approximation $\tilde{S}(j)$ to the solution $S(f)$. If for all f the probability to get an ε-estimate to $S(f)$ is

$$\sum_{j:\|S(f)-\tilde{S}(j)\|<\varepsilon} p_f(j) \geq \tfrac{3}{4},$$

then we say that the algorithm solves the problem with accuracy ε and probability $3/4$ in T queries. The probability $3/4$ is arbitrary; any number greater than $1/2$ would do. The success probability can then be boosted to become arbitrarily close to one. We are interested in the smallest number of queries such that a quantum algorithm solves the problem for all f with error at most ε.

Besides the quantum circuit model, there are several different quantum computing models, such as measurement-based quantum computation [6; 33; 90], and adiabatic quantum computation. The adiabatic quantum computation will be introduced in Section 1.5.

1.5 Adiabatic Computation and Berry Phase

One main obstacle to realize quantum algorithms is the decoherence induced by the coupling environment. To overcome the obstacle a novel quantum computation model, quantum adiabatic computation [3; 10; 26; 43; 73; 92; 93; 96], is a promising candidate. The model is believed to enjoy inherent robustness against the impact of decoherence [26; 73; 93]. Even though not all researchers share this view [10; 96], adiabatic quantum computation still attracts considerable attention.

Adiabatic computation is based on the adiabatic theorem. Consider a quantum system in a state $|\psi(t)\rangle$, $(0 \leq t \leq T)$, which evolves according to the Schrodinger equation (1.2). If the system is initially in its ground state, and the Hamiltonian varies slowly enough, it will remain close to the ground state of $H(t)$, at time t. Let $|E_0(t)\rangle$ be the ground state of the Hamiltonian, and let $E_0(t)$ be the corresponding eigenvalue. If $H(T) = H(0)$, then $|\psi(T)\rangle$

is close to $|\psi(0)\rangle$, with the exception of a global phase. The phase can be divided into two parts: the *dynamic phase*

$$\theta = -\int_0^T E_0(t)dt, \tag{1.6}$$

and the *geometric Berry phase*

$$\gamma = i\int_0^T \langle E_0|\frac{d}{dt}E_0\rangle dt. \tag{1.7}$$

The Berry phase depends only on the path taken, not on how fast the path is traversed. Hence, if we design a cyclic path of Hamiltonians, the Berry phase is totally determined [5; 18; 94; 103].

The quantum adiabatic theorem suggests a natural approach for optimization. The basic idea is to encode the solution of the problem in the ground state of a Hamiltonian and to adiabatically evolve into this ground state, starting from a known ground state. According to the adiabatic theorem, the probability of finding a solution will be high provided the evolution is sufficiently slow. From the adiabatic theorem, the running time of an adiabatic algorithm depends crucially on the gap $\Delta(s)$ between the ground and first excited state of $H(s)$.

Quantum adiabatic computation has been proved to be polynomially equivalent to the quantum circuit model [3]. There are also adiabatic algorithms that are as good as the quantum algorithms in the circuit model. An important example is the unstructured search problem. For a database with size N, the problem can be solved in time $O(\sqrt{N})$ by Grover's algorithm [52], but has classical query complexity $\Omega(N)$. There is an adiabatic algorithm that can also solve this problem with running time proportional to $\sqrt{N}$ [92].

In this thesis, we provide an adiabatic algorithm for the counting problem described in Section 1.2.1. This adiabatic algorithm is different from other adiabatic algorithms, such that the solution is encoded in the Berry phase of the final state, rather than the ground state of the final Hamiltonian. The final information is then obtained by estimating the relative phase, rather than by the usual quantum measurement.

1.6 Quantum PAC Learning

The PAC (Probably Approximately Correct) model of concept learning was introduced by Valiant [112] and has been extensively studied [21; 40; 66]. In the model, the learning algorithm has access to an example oracle $EX(c, D)$ where $c \in C$ is the unknown target concept, C is a known concept class and D is an unknown distribution over $\{0,1\}^n$. The oracle $EX(c, D)$ takes no inputs; when invoked, it generates a labeled example $(x, c(x))$ according to the distribution D. The goal of the learning algorithm is to provide a hypothesis $h : \{0,1\}^n \to \{0,1\}$, which is an ε-approximation for c under the distribution D, i.e., a hypothesis h such that $Pr_{x \in D}[h(x) \neq c(x)] \leq \varepsilon$. An algorithm A is a *PAC learning algorithm* for C if the following condition holds: given $0 < \varepsilon, \delta < 1$, for all $c \in C$ and all distributions D over $\{0,1\}^n$, with probability at least $1 - \delta$, the algorithm A outputs a hypothesis h, which is an ε-approximation to c under D. In the literature, ε is usually known as *accuracy* and $1 - \delta$ is known as *confidence*. The *query complexity* of the learning algorithm A for C is the maximum number of queries to $EX(c, D)$ algorithm A invokes for any $c \in C$ and any distribution D over $\{0,1\}^n$.

The quantum PAC learning model is a generalization of the classical PAC learning model to the quantum computing model [12; 24; 49; 50]. In the quantum PAC learning model, the algorithm has access to a quantum example oracle $QEX(c, D)$, which generates a superposition of all labeled examples, where each labeled example $(x, c(x))$ appears in the superposition with amplitude proportional to the square root of $D(x)$. More precisely, for a distribution D over $\{0,1\}^n$, the quantum example oracle $QEX(c, D)$ is a quantum gate that transforms the initial state $|0^n, 0\rangle$ as

$$|0^n, 0\rangle \mapsto \sum_{x \in \{0,1\}^n} \sqrt{D(x)} |x, c(x)\rangle.$$

The $QEX(c, D)$ is only defined on the initial state $|0^n, 0\rangle$, and it is required that a *quantum learning algorithm* only invokes the $QEX(c, D)$ oracle on the basis state $|0^n, 0\rangle$. A quantum learning algorithm is a quantum PAC learning algorithm for C if it has the following property: given $0 < \varepsilon, \delta < 1$, for all $c \in C$ and all distributions D over $\{0,1\}^n$, with probability $1 - \delta$ the algorithm outputs a representation of a hypothesis h, which is an ε-approximation to c under the distribution D. Similar to the classical PAC learning model,

ε is usually considered as accuracy and $1-\delta$ is considered as confidence. The *quantum query complexity* of a quantum PAC learning algorithm is the maximum number of queries to $QEX(c,D)$ for any $c \in C$ and any distribution D over $\{0,1\}^n$.

Because of the specialization of quantum circuits used in the quantum PAC learning model, they can be simplified as follows. First, since all the information used for computation comes from the oracle calls, the initial state of the quantum register can be assumed as $|0^n, 0\rangle$. Secondly, since $QEX(c,D)$ is required to be invoked only on the state $|0^n,0\rangle$, without loss of generality, all $QEX(c,D)$ calls in the quantum learning algorithm can be assumed to occur at the beginning of the algorithm. Finally, after a set of invocations to $QEX(c,D)$, the algorithm performs a sequence of unitary transformations and one measurement. Since we are chiefly interested in query complexity and not the circuit size, they can be replaced by a POVM measurement, which is represented by a set of positive self-adjoint operators $\{\Pi_h\}$, where $\sum_h \Pi_h = I$. More precisely, let

$$|\psi_c\rangle = \sum_{x\in\{0,1\}^n} \sqrt{D(x)}|x, c(x)\rangle.$$

A quantum learning algorithm with query complexity T can be represented as a POVM measurement $\{\Pi_h\}$ for $|\psi_c\rangle^{\otimes T}$. If the algorithm is a quantum PAC learning algorithm with accuracy ε and confidence $1-\delta$ for the concept class C, for any $c \in C$, then

$$\sum_{h\in H_c} \langle\psi_c|^{\otimes T} \Pi_h |\psi_c\rangle^{\otimes T} > 1-\delta. \tag{1.8}$$

where H_c is the set of hypothesis h, such that h is an ε-approximation to c.

In the PAC learning model, the Vapnik-Chervonenkis (VC) dimension plays a key role for measuring the complexity of a concept class [21; 113]. For $S \subseteq \{0,1\}^n$, we write $\Pi_C(S)$ to denote $\{c \cap S : c \in C\}$, so $|\Pi_C(S)|$ is the number of different *dichotomies* which the concepts in C induce on the points in S. A subset $S \subseteq \{0,1\}^n$ is said to be *shattered* by C if $|\Pi_C(S)| = 2^{|S|}$, i.e., if C induces every possible dichotomy on the points in S. For a concept class C, its VC-dimension is the cardinality d of the largest set S shattered by C [66]. In the classical PAC learning model, an algorithm with ε accuracy and $1-\delta$ confidence, for a concept class with VC-dimension d, requires at least

$$\Omega\left(\frac{1}{\varepsilon}(d + \log\frac{1}{\delta})\right)$$

examples [40]. The lower bound is nearly optimal, since an upper bound of

$$O\left(\frac{1}{\varepsilon}(d\log\frac{1}{\varepsilon}+\log\frac{1}{\delta})\right)$$

is given in [21]. Since a quantum PAC learning algorithm can be directly transferred to a classical PAC learning algorithm [80], the upper bound in the classical PAC learning model is also an upper bound in the quantum PAC learning model. On the other hand, in the quantum PAC learning model, the best known lower bound is given by Atici and Servedio [12], which is

$$\Omega\left(d+\frac{1}{\varepsilon}\left(\sqrt{d}+\log\frac{1}{\delta}\right)\right).$$

In this thesis, we improve the lower bound to

$$\Omega\left(\frac{1}{\varepsilon}\left(d^{1-\eta}+\log\frac{1}{\delta}\right)\right),$$

for any $\eta > 0$. Hence, we show that the lower bounds of query complexity are almost the same in both quantum and classical PAC learning model.

1.7 Hamiltonian Simulation

It is very hard to simulate a quantum system using classical computers, since the computational cost of simulating many particle quantum systems typically grows exponentially with the number of particles. However, Feynman [44] proposed that computers based on quantum mechanical principles could provide the most powerful technology for simulating quantum systems, which is also one of the founding ideas of the field of quantum computation.

In addition to predicting the behavior of physics systems [17; 44; 64; 76; 124], Hamiltonian simulation also has algorithmic applications. For example, the implementation of *power* queries by using *bit* queries in the Sturm-Liouville eigenvalue problem is based on a Hamiltonian simulation problem [83]. Other examples include adiabatic optimization [42], quantum walks [91], and the NAND tree evaluation algorithms [41].

In a Hamiltonian simulation problem, our goal is to simulate the unitary operator e^{-iHt} for a given Hamiltonian H and evolution time t. The accuracy of the simulation is measured by the trace distance [80] of the simulated final state and the desired final state, which is

noted by ε. The cost of a quantum algorithm is measured by the number of elementary gates in the quantum circuit, as stated in section 1.4. If it is $\text{poly}(n, t, 1/\varepsilon)$, the Hamiltonian H is said to be efficiently simulated for any $t > 0$, for an error bound ε.

There are a few cases where a Hamiltonian can obviously be simulated efficiently. For example, this is the case if H only acts nontrivially on a constant number of qubits, because any unitary evolution on a constant number of qubits can be approximated using a constant number of elementary gates [37].

Note that since we require a simulation for an arbitrary time t (with $\text{poly}(t)$ gates), we can rescale the evolution by any polynomial factor: if H can be efficiently simulated, then so can cH for any $c = \text{poly}(n)$. This holds even if $c < 0$, since any efficient simulation is expressed in terms of quantum gates, and can simply be run in reverse [80].

Another example is as follows. Suppose H is diagonal in the computational basis, and any diagonal element $d(a) = \langle a|H|a\rangle$ can be computed efficiently. Suppose for simplicity that the diagonal element $d(a)$ is expressed as a binary number with k bits of precision. Then H can be simulated efficiently as follows: For any input computational basis state $|a\rangle$, together with a k-qubit ancilla state initialized to $|0\rangle$,

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

For instance, the discretization of a bounded potential $V(x)$ on a mesh with size $h = 1/(N+1)$ can be efficiently simulated, assuming that the evaluations $V(x_i)$ is efficiently computable. Then, the query complexity is 2, and the ancilla computation is $O(\log(N))$.

In addition, we can rotate the basis in which a Hamiltonian is applied using any unitary transformation with an efficient decomposition into basic gates. In other words, if H can be efficiently simulated and the unitary transformation U can be efficiently implemented, then $UHU^\dagger$ can be efficiently simulated. This follows from the simple identity

$$e^{-iUHU^\dagger t} = Ue^{-iHt}U^\dagger.$$

For instance, consider the discretization of the negative Laplacian $-\Delta_h = S\Lambda S^\dagger$, where

Λ is a diagonal matrix. S is the sine transform of size $N \times N$, which can be implemented with $O(\log^2 N)$ quantum operations [69; 80]. Any diagonal element of Λ can also be calculated efficiently. Hence, $-\Delta_h$ can be implemented efficiently.

For a Hamiltonian $H = \sum_{j=1}^{m} H_j$, where H_j can be simulated efficiently, there are various algorithms for simulating e^{-iHt}. Obviously, if the H_j commute with each other, $e^{-iHt} = \prod_{j=1}^{m} e^{-iH_j t}$. Hence, in this thesis, we assume H_j do not commute. Without loss of generality, we also assume $\|H_1\| \geq \|H_2\| \geq \cdots \geq \|H_m\|$. Let $\tau = \|H_1\|t$, such that the evolution $e^{-iHt} = e^{-iH\tau/\|H_1\|}$, which is equivalent to the evolution of a bounded Hamiltonian with time τ.

Various algorithms have been studied in the literature. One of the simplest and widely used algorithms is based on the Trotter formula [80]

$$\left\| e^{-iH\Delta t} - \prod_{j=1}^{m} e^{-iH_j \Delta t} \right\| = O(\Delta t^2).$$

For an error bound ε, the algorithm based on Trotter formula has $O(\tau^2/\varepsilon)$ exponentials. A better algorithm is based on Strang's formula [104]

$$\left\| e^{-iH\Delta t} - \prod_{j=1}^{m} e^{-iH_j \Delta t/2} \prod_{j=m}^{1} e^{-iH_j \Delta t/2} \right\| = O(\Delta t^3).$$

The algorithm derived from Strang's formula has $O(\tau^{3/2}/\varepsilon^{1/2})$ exponentials. Based on Suzuki's decomposition [105; 106], where

$$S_2(\Delta t) = \prod_{j=1}^{m} e^{-iH_j \Delta t/2} \prod_{j=m}^{1} e^{-iH_j \Delta t/2},$$

and recursively

$$S_{2k}(\Delta t) = [S_{2k-2}(p_k \Delta t)]^2 S_{2k-2}((1-4p_k)\Delta t)[S_{2k-2}(p_k \Delta t)]^2, \quad k = 2, 3, \ldots,$$

where $p_k = (4 - 4^{1/(2k-1)})^{-1}$, there is a sequence of simulation algorithms. From

$$\left\| e^{-i\sum_{j=1}^{m} H_j \Delta t} - S_{2k}(\Delta t) \right\| = O(|\Delta t|^{2k+1}), \tag{1.9}$$

a set of upper bounds for the number of exponentials in simulating e^{-iHt} is derived

$$N \leq N_{\text{prev}} := 2m5^{2k}(m\tau)^{1+\frac{1}{2k}}(\frac{1}{\varepsilon})^{\frac{1}{2k}}, \tag{1.10}$$

for $k = 1, 2, \cdots$ [17].

From the algorithms given above, a Hamiltonian H can be simulated efficiently if it is sparse, i,e., it has poly$(\log N)$ nonzero entries in each row, and if there is an efficient means of computing the indices and the matrix elements of the nonzero entries in any given row. A simple algorithm is based on Vizing's theorem [29], which says that a graph of maximum degree d has an edge coloring with at most $d + 1$ colors. By using only local information about the graph, there is a polynomial overhead in the number of colors used. That is, given an undirected graph G with N vertices and maximum degree d, and that the neighbors of any given vertex can be efficiently computed. 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 $b = b'$. Hence, a sparse Hamiltonian can be written as the sum of polynomially many local Hamiltonians. From [17], the best upper bound known of query complexity for estimating the Hamiltonian H such that the nonzero elements in each column is at most d is

$$N = O\left(n(\log^* n)^2 d^2 5^{2k} (d^2\tau)^{1+1/(2k)} \varepsilon^{-1/(2k)}\right).$$

A recent paper [27] improves this bound to

$$N = O\left(5^{2k} d^2 (d + \log^* n)\tau \left(\frac{d\tau}{\varepsilon}\right)^{1/2k}\right).$$

Although the simulation of any Hamiltonian may be performed arbitrarily close to linear in the evolution time, however the scaling cannot be sublinear [17].

In this thesis, we obtain a series of upper bounds on the number of exponentials required to approximate e^{-iHt} with error ε, which are more accurate than those in [17]. Moreover, we derive the order of the splitting method that optimizes the cost of the resulting algorithm. We show significant speedups relative to previously known results.

We also consider a new model of Hamiltonian simulation, which is based on randomized algorithms. Such algorithms simulate the evolution of a Hamiltonian $H = \sum_{j=1}^{m} H_j$ for time t, by a product of exponentials of H_j in a random sequence, and random evolution times. Hence the final state of the system is approximated by a mixed quantum state.

Chapter 2

Adiabatic Counting with Geometric Phase

2.1 Introduction

Quantum adiabatic computation is a novel quantum computing model as introduced in Section 1.5. The model has been proved to be polynomially equivalent to the quantum circuit model [3; 43]. For instance, in the adiabatic model, searching an unordered database requires time of the same order of magnitude as Grover's algorithm [52; 92], but few adiabatic algorithms are known that have performance similar to that of the corresponding algorithm in the quantum circuit model. In this chapter, we show an adiabatic algorithm for the counting problem. The task of the counting problem is to approximate the proportion α of marked items in an N-item database. The counting problem plays a key role in numerical problems, as described in Section 1.2.1.

Our algorithm for the counting problem is based on the Berry phase acquired in the adiabatic evolution. When a quantum system undergoes a cyclic adiabatic evolution, it acquires a geometric phase, which is known as geometric phase or the Berry phase [18]. The phase depends only on the path taken, not on how fast that path is traversed, as long as it satisfies the adiabatic condition. The phenomenon was first observed by Berry [18], and then extensively studied [8; 47; 99; 122]. The Berry phase is a purely geometric feature, and resilient to certain small errors [20; 61; 79], so various methods are proposed to use it

when building quantum gates [39; 63; 128].

In our algorithm, we design an adiabatic evolution such that the resulting Berry phases encode the solution of the problem. Then we can solve the problem by estimating the Berry phase after the adiabatic evolution. Since Berry phases are global phases and cannot be measured directly, we let two parts of a superposition undergo the same cyclic adiabatic evolution in different directions. After the adiabatic evolution, the dynamic phases of the two parts cancel out, and the Berry phases are $\pm 2\pi\alpha$. Then, by estimating the relative phase between the two parts of the superposition, we can estimate α. The algorithm has a runtime of order $\varepsilon^{-3/2}$, which beats the optimal classical algorithm in the randomized setting. Usually, in adiabatic algorithms it is the final state that encodes the solution of the problem, while in our algorithm it is the Berry phase. Since the Berry phase is a purely geometric feature, i.e., it only depends on the path of evolution, and is independent of details of how the evolution is executed, the result is resilient to certain small errors. The details of adiabatic algorithms and the Berry phase are introduced in Section 1.5.

The remainder of the current chapter is organized as follows. In Section 2.2, we provide the basic adiabatic evolution used in the algorithm, which can encode the solution to a Berry phase of a quantum system. Then we give the adiabatic algorithm for the counting problem. In Section 2.3, we will show the relationship between the accuracy of the algorithm and the time it used. In Appendix A, we show the detailed derivation of the difference of real relative phase and the Berry phase in our algorithm.

2.2 The adiabatic algorithm for the counting problem

In this section, we will show how to encode the solution to the Berry phases. We use a function $f : \{0, \cdots, N-1\} \to \{0, 1\}$ to denote whether an item is marked, i.e.,

$$f(s) = \begin{cases} 1, & \text{if the } s\text{-th item is marked,} \\ 0, & \text{otherwise.} \end{cases}$$

For an error bound ε, the algorithm has $m = \log(1/\varepsilon)$ adiabatic evolutions. In each evolution, we use $n+1$ qubits, where $n = \log_2 N$. The quantum state can be divided into two systems: the control system, which has 1 qubit, and the computing system, which has n

qubits. The computing system is in an N-dimensional Hilbert space, whose basis states are denoted as $|k\rangle$, where $k = 0, \cdots, N-1$. We use an equal superposition of all basis states

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle \tag{2.1}$$

as the initial state in the computing system and use $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ as the initial state in the control system. Hence, the initial state of the whole system is

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\psi_0\rangle. \tag{2.2}$$

In each evolution, the initial Hamiltonian of the computing system is

$$H_0 = I - |\psi_0\rangle\langle\psi_0|. \tag{2.3}$$

Typically, in quantum algorithms, the fundamental oracle used is

$$|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle. \tag{2.4}$$

In our algorithm, we modify the oracle to

$$|x\rangle|y\rangle|z\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle|z \oplus (y \cdot f(x))\rangle. \tag{2.5}$$

We use a 2-qubit auxiliary state

$$\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

for quantum state $|k\rangle$, then the oracle works as follows,

$$\begin{aligned}
&|k\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
=&|k\rangle \otimes \frac{1}{2}(|00\rangle + i|10\rangle - |01\rangle - i|11\rangle) \\
\mapsto&\frac{1}{2}|k\rangle(|f(k)\rangle|0\rangle + i|f(k) \oplus 1\rangle|f(k)\rangle - |f(k)\rangle|1\rangle - i|f(k) \oplus 1\rangle|1 \oplus f(k)\rangle).
\end{aligned}$$

When $f(k) = 0$, the result is

$$\frac{1}{2}|k\rangle \otimes (|00\rangle + i|10\rangle - |01\rangle - i|11\rangle) = |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

and when $f(k) = 1$, the result is

$$\frac{1}{2}|k\rangle \otimes (|10\rangle + i|01\rangle - |11\rangle - i|00\rangle) = -i|0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Discarding the auxiliary qubits, we obtain $|k\rangle \mapsto (-i)^{f(k)}|k\rangle$. Hence, the operation of the oracle can be written as

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}|k\rangle \to \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\exp(-i\frac{\pi}{2}f(k))|k\rangle. \tag{2.6}$$

Let $\beta = 1-\alpha$, $M = \alpha N$, and

$$\begin{aligned} |\hat{0}\rangle &= \frac{1}{\sqrt{N-M}}\sum_{s:f(s)=0}|s\rangle, \\ |\hat{1}\rangle &= \frac{1}{\sqrt{M}}\sum_{s:f(s)=1}|s\rangle. \end{aligned} \tag{2.7}$$

The initial state can be rewritten as

$$|\psi_0\rangle = \sqrt{\beta}|\hat{0}\rangle + \sqrt{\alpha}|\hat{1}\rangle, \tag{2.8}$$

and the states after repeatedly using the oracle are

$$\begin{aligned} |\psi_1\rangle &= \sqrt{\beta}|\hat{0}\rangle - i\sqrt{\alpha}|\hat{1}\rangle, \\ |\psi_2\rangle &= \sqrt{\beta}|\hat{0}\rangle - \sqrt{\alpha}|\hat{1}\rangle, \\ |\psi_3\rangle &= \sqrt{\beta}|\hat{0}\rangle + i\sqrt{\alpha}|\hat{1}\rangle. \end{aligned}$$

In the adiabatic algorithm, we define four Hamiltonian oracles as

$$H_k = I - |\psi_k\rangle\langle\psi_k|, \tag{2.9}$$

for $k = 0, 1, 2, 3$. Then consider a linear interpolation between the four oracles, that is

$$H(t) = \sum_{k=0}^{3} s_k(t)H_k, \tag{2.10}$$

where $\sum s_k(t) = 1$, for any $0 \le t \le T$.

In the j-th evolution, we choose

$$\begin{aligned} s_0 &= \frac{1}{2}(1+\cos\theta_j), s_1 = \frac{1}{2}\sin\theta_j, \\ s_2 &= \frac{1}{2}(1-\cos\theta_j), s_3 = -\frac{1}{2}\sin\theta_j, \end{aligned} \tag{2.11}$$

where θ_j is a function from $[0, T]$ to $[0, 2^j\pi]$, satisfying $\theta_j(0) = 0$ and $\theta_j(T) = 2^j\pi$. Since the Berry phase only depends on the path of the evolution of the Hamiltonian, the choice

of the function $\theta(t)$ does not affect our result. Combining (2.9) and (2.11), the Hamiltonian of the j-th evolution is

$$\begin{aligned} H(\theta_j) =& I - \frac{1}{2}(1+\cos\theta_j)|\psi_0\rangle\langle\psi_0| - \frac{1}{2}\sin\theta_j|\psi_1\rangle\langle\psi_1| \\ & - \frac{1}{2}(1-\cos\theta_j)|\psi_2\rangle\langle\psi_2| + \frac{1}{2}\sin\theta_j|\psi_3\rangle\langle\psi_3| \\ =& I - |\psi(\theta_j)\rangle\langle\psi(\theta_j)|, \end{aligned} \tag{2.12}$$

where

$$|\psi(\theta_j)\rangle = \sqrt{\beta}|\hat{0}\rangle + e^{-i\theta_j}\sqrt{\alpha}|\hat{1}\rangle, \tag{2.13}$$

and $\theta_j \in [0, 2^j\pi]$. Clearly, the ground state of $H(\theta_j)$ is $|\psi(\theta_j)\rangle$. We set the Hamiltonian of the whole system in the j-th adiabatic evolution as

$$|0\rangle\langle 0| \otimes H(\theta_j) + |1\rangle\langle 1| \otimes H(-\theta_j). \tag{2.14}$$

Then the Berry phases of the computing system after the evolution are γ_j and $-\gamma_j$, where

$$\gamma_j = i\int_0^{2^j\pi} \langle\psi|\frac{d}{d\theta}\psi\rangle d\theta = \int_0^{2^j\pi} \alpha d\theta = 2^j\pi\alpha. \tag{2.15}$$

Since the dynamic phase is the same in both parts, the final state is

$$|\psi_j(T)\rangle = \frac{1}{\sqrt{2}}(e^{i\gamma_j}|0\rangle + e^{-i\gamma_j}|1\rangle) \otimes |\psi_0\rangle. \tag{2.16}$$

Hence, at the end of the evolution, the relative phase of the first qubit is $\Gamma_j = 2\gamma_j = 2\pi(2^j\alpha)$. In this way, we successfully encode the solution to the relative phases of a set of quantum states.

In the algorithm, we do not use the phase estimation procedure in [80] to estimate α from the Berry phases, in order to avoid unnecessary entanglement. We use Kitaev's equivalent procedure instead [68]. As described above, in the j-th adiabatic evolution of the algorithm, we prepare a quantum state whose relative phase is $2\pi(2^j\alpha)$, for $j = 1, \cdots, m$. A measurement for the first qubit in $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ basis gives the result $|+\rangle$ with probability

$$p = \cos^2(\gamma) = \cos^2(2^j\pi\alpha). \tag{2.17}$$

Hence, if we apply the process several times, we can approximate the probability. More precisely, let $q = r/R$ be the ratio between the number of r of results $|+\rangle$ and the number R of measurements. Then Chernoff's bound [77]

$$Pr(|p - q| \geq \delta) \leq 2e^{-2\delta^2 R} \tag{2.18}$$

shows that for a fixed δ, the error is smaller than ε for only $O(\log(1/\varepsilon))$ number of measurements. In this application, we only need an error that is smaller than $\pi/8$, which will give us an estimate of $2^j\alpha$, modulo 1, with error 1/16. Let

$$\alpha = \sum_{j=1}^{\infty} 2^{-j}\alpha_j, \tag{2.19}$$

for $\alpha_j \in \{0, 1\}$, and $\alpha_1 = 0$ since $\alpha < 1/2$. We also use $\overline{.\alpha_1 \cdots \alpha_p}$ to denote the binary fraction $\sum_{j=1}^{p} 2^{-j}\alpha_j$. For $j = 1, \cdots, m$, we replace the known approximate value of $2^j\alpha$ by η_j, the closest number from the set $\{0, 1/8, 2/8, \cdots, 7/8\}$. Hence, we have

$$|2^j\alpha - \eta_j|_1 < 1/16 + 1/16 = 1/8. \tag{2.20}$$

Since if $|y - 2\alpha|_1 < \delta < 1/2$, then $|y'_0 - \alpha|_1 < \delta/2$ or $|y'_1 - \alpha|_1 < \delta/2$, where y'_1, y'_2 are the solutions to the equation $2y' \equiv y(\mathrm{mod}1)$, we can start from $2^m\alpha$ and increase the precision in the following way: Set $\eta_m = \overline{.\alpha_m\alpha_{m+1}\alpha_{m+2}} = \eta_m$ and proceed by iteration:

$$\alpha_j = \begin{cases} 0 & \text{if} |\overline{.0\alpha_{j+1}\alpha_{j+2}} - \eta_j|_1 < 1/4, \\ 1 & \text{if} |\overline{.1\alpha_{j+1}\alpha_{j+2}} - \eta_j|_1 < 1/4, \end{cases} \tag{2.21}$$

for $j = m - 1, \cdots, 1$. By a simple induction, $\overline{.\alpha_1\alpha_2 \cdots \alpha_m}$ can estimate α with error less than $2^{-m} = \varepsilon$.

2.3 Running Time of the Adiabatic Algorithm

In this section, we consider the accuracy of the evolutions and the running time of the algorithm. It is easy to see that under the Hamiltonian given in (2.12) the actual state $|\varphi(\theta)\rangle$ in the computing system always stays in the subspace spanned by $\{|\hat{0}\rangle, |\hat{1}\rangle\}$. Then $H(\theta)$ can be rewritten as

$$H(\theta) = \begin{pmatrix} \alpha & -\sqrt{\alpha\beta}e^{i\theta} \\ -\sqrt{\alpha\beta}e^{-i\theta} & \beta \end{pmatrix} \tag{2.22}$$

in the subspace. Assume $\omega = d\theta/dt$ is constant, and $\omega \ll 1$. Let $|\varphi(t)\rangle$ be the t-time state in the system which is initially in $|\psi_0\rangle$, and evolving under $H(\omega t)$. By solving the Schrödinger equation (1.2), we attain

$$|\varphi(t)\rangle = e^{-i\frac{1}{2}t}(Ae^{i\omega_1 t} + Be^{i\omega_2 t})|0\rangle + e^{-i\frac{1}{2}t}(Ce^{-i\omega_1 t} + De^{-i\omega_2 t})|1\rangle, \tag{2.23}$$

where

$$\omega_{1,2} = \frac{\omega \pm \sqrt{(1-\omega)^2 + 4\alpha\omega}}{2}, \tag{2.24}$$

and

$$\begin{aligned} A &= \frac{(1-\omega)^2 - \alpha(1-3\omega) + (\beta-\omega)E}{(1-\omega)^2 + 4\alpha\omega + (\beta-\alpha-\omega)E}\sqrt{\beta}, \\ B &= \frac{\alpha(1+\omega) - \alpha E}{(1-\omega)^2 + 4\alpha\omega + (\beta-\alpha-\omega)E}\sqrt{\beta}, \\ C &= \frac{(1+\omega)^2 - \beta(1+3\omega) - (\alpha+\omega)E}{(1-\omega)^2 + 4\alpha\omega + (\beta-\alpha-\omega)E}\sqrt{\alpha}, \\ D &= \frac{\beta(1-\omega) + \beta E}{(1-\omega)^2 + 4\alpha\omega + (\beta-\alpha-\omega)E}\sqrt{\alpha}, \end{aligned} \tag{2.25}$$

where $E = \sqrt{(1-\omega)^2 + 4\alpha\omega}$, see Appendix A. On the other hand, denote the quantum state evolving under the Hamiltonian $H(-\theta) = H(-\omega t)$ by $|\varphi'(t)\rangle$, which can be obtained from $|\varphi\rangle$ by exchanging all ω by $-\omega$, i.e.,

$$|\varphi'(t)\rangle = e^{-i\frac{1}{2}t}(A'e^{i\omega'_1 t} + B'e^{i\omega'_2 t})|0\rangle + e^{-i\frac{1}{2}t}(C'e^{-i\omega'_1 t} + D'e^{-i\omega'_2 t})|1\rangle, \tag{2.26}$$

where

$$\omega'_{1,2} = \frac{-\omega \pm \sqrt{(1+\omega)^2 - 4\alpha\omega}}{2}, \tag{2.27}$$

and

$$\begin{aligned} A' &= \frac{(1+\omega)^2 - \alpha(1+3\omega) + (\beta+\omega)E'}{(1+\omega)^2 - 4\alpha\omega + (\beta-\alpha+\omega)E'}\sqrt{\beta}, \\ B' &= \frac{\alpha(1-\omega) - \alpha E'}{(1+\omega)^2 - 4\alpha\omega + (\beta-\alpha+\omega)E'}\sqrt{\beta}, \\ C' &= \frac{(1-\omega)^2 - \beta(1-3\omega) - (\alpha-\omega)E'}{(1+\omega)^2 - 4\alpha\omega + (\beta-\alpha+\omega)E'}\sqrt{\alpha}, \\ D' &= \frac{\beta(1+\omega) + \beta E'}{(1+\omega)^2 - 4\alpha\omega + (\beta-\alpha+\omega)E'}\sqrt{\alpha}, \end{aligned} \tag{2.28}$$

where $E' = \sqrt{(1+\omega)^2 - 4\alpha\omega}$.

Then the final state of the j-th evolution is

$$|\psi_j(T)\rangle = \frac{1}{\sqrt{2}}(|0\rangle|\varphi(T)\rangle + |1\rangle|\varphi'(T)\rangle), \tag{2.29}$$

where $T = 2^j\pi/\omega$, which is the time of the j-th evolution, for $j = 1, \cdots, m$. As indicated before, we will measure the relative phase of the first qubit, and use it as an approximation of $2\pi(2^j\alpha)$. Let $|\varphi_\perp\rangle$ be a state in the span space of $|\hat{0}\rangle$ and $|\hat{1}\rangle$, which is orthogonal to $|\varphi\rangle$ by (2.29), so that

$$|\psi_j(T)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \langle\varphi(T)|\varphi'(T)\rangle|1\rangle)|\varphi\rangle + \frac{1}{\sqrt{2}}(\langle\varphi_\perp(T)|\varphi'(T)\rangle)|1\rangle|\varphi_\perp(T)\rangle. \tag{2.30}$$

Hence, with probability

$$p_s = \frac{1 + |\langle\varphi(T)|\varphi'(T)\rangle|^2}{2}, \tag{2.31}$$

the relative phase will be the argument of $\langle\varphi(T)|\varphi'(T)\rangle$. It can be checked that

$$\langle\varphi(T)|\varphi'(T)\rangle = (AA'+DD')e^{i\mu_1}+(BB'+CC')e^{-i\mu_1}+(AB'+C'D)e^{i\mu_2}+(A'B+CD')e^{-i\mu_2}, \tag{2.32}$$

where

$$\begin{aligned}\mu_1 &= \frac{1}{2}(\sqrt{(1-\omega)^2+4\alpha\omega} - \sqrt{(1+\omega)^2-4\alpha\omega})T,\\ \mu_2 &= \frac{1}{2}(\sqrt{(1-\omega)^2+4\alpha\omega} + \sqrt{(1+\omega)^2-4\alpha\omega})T\end{aligned} \tag{2.33}$$

From the assumption $\omega \ll 1$, we have

$$\begin{aligned}AA' + DD' &= 1 - 3\alpha\beta\omega^2 + O(\omega^3),\\ BB' + CC' &= -\alpha\beta\omega^2 + O(\omega^3),\\ AB' + C'D &= 2\alpha\beta\omega^2 + O(\omega^3),\\ A'B + CD' &= 2\alpha\beta\omega^2 + O(\omega^3).\end{aligned} \tag{2.34}$$

Hence,

$$p_s = \frac{1 + |\langle\varphi(T)|\varphi'(T)\rangle|^2}{2} \geq 1 - 8\alpha\beta\omega^2, \tag{2.35}$$

and

$$|\mathrm{ang}(\langle\varphi(T)|\varphi'(T)\rangle) - \mu_1| \leq 8\alpha\beta\omega^2. \tag{2.36}$$

Moreover,

$$\begin{aligned}\mu_1 &= 2\alpha\omega T + 2\alpha\beta(\beta-\alpha)\omega^3 T + O(\omega^5 T) \\ &= 2\pi(2^j\alpha)(1+\beta(\beta-\alpha)\omega^2 + O(\omega^3)),\end{aligned} \tag{2.37}$$

in the j-th evolution. Hence, the difference between the expected relative phase and $2\pi(2^j\alpha)$ is

$$\Delta \leq 2\pi(1-p_s) + |\mathrm{ang}(\langle\varphi_+(T)|\varphi_-(T)\rangle) - \mu_1| + |\mu_1 - 2\pi(2^j\alpha)| = O(2^j\omega^2). \tag{2.38}$$

To estimate $2^j\alpha$ modulo 1 with error less than $1/16$, it is enough to make $\Delta < 2\pi/32$, and then estimate the relative phase to its expected value within error $2\pi/32$. To satisfy the first condition, in the j-th evolution we set

$$\omega = \omega_j = O(\frac{1}{\sqrt{2^j}}) \tag{2.39}$$

in (2.38). When estimating the relative phase, we can boost the success probability using repetitions. In our case, $O(m-j)$ repetitions yield overall success probability greater than $1/2$. Since the time of the j-th evolution is $2^j\pi/\omega_j$, the total running time is

$$T_{total} = \sum_{j=1}^{m} O\left(\frac{2^j}{\omega_j}(m-j)\right) = O\left((2^m)^{3/2}\right) = O\left(\left(\frac{1}{\varepsilon}\right)^{3/2}\right). \tag{2.40}$$

Therefore, the adiabatic algorithm has an $O(\varepsilon^{-3/2})$ running time. As we know the optimal quantum algorithm has a running time $O(1/\varepsilon)$, our adiabatic algorithm is not as good as the quantum algorithm given in the circuit model. However, it is better than the best classical algorithm, which needs $O(\varepsilon^{-2})$ running time.

2.4 Discussion

In conclusion, we have proposed an adiabatic algorithm for the quantum counting problem. The key idea of the algorithm is to construct Berry phases which equal $2\pi(2^j\alpha)$, for $j = 1, \cdots, m$, where α is the proportion of marked items in the database, $m = \log(1/\varepsilon)$. The algorithm has a running time of $O(\varepsilon^{-3/2})$, which beats the optimal classical random algorithm. Since the counting problem is the foundation of many numerical problems, such

as high-dimensional integration, path integration, and eigenvalue estimation, our algorithm provides a robust and efficient strategy to improve our ability to solve these problems.

There are some special characteristics of our algorithm. In particular, the solution of the problem is encoded in the phase difference of the final state, rather than the ground state of the final Hamiltonian. The final information is obtained by the phase estimation routine, rather than by the usual quantum measurement. In contrast to usual adiabatic algorithms, such as in [92], we use several Hamiltonians as oracles to construct the evolution path.

Chapter 3

Query Complexity for Quantum PAC Learning

3.1 Introduction

Some quantum algorithms are closely related to the field of learning theory. One example given by one of the earliest quantum algorithms, the Deutsch-Jozsa algorithm [36], which decides whether a boolean function over the domain $\{0,1\}^n$ is constant or balanced. To exactly decide the problem requires $\Theta(2^n)$ queries in the classical setting, but only $\Theta(1)$ queries in the quantum setting. Another example is the Bernstein-Vazirani algorithm [16], where the purpose is to learn a concept among the concept class of boolean functions $c_s(x) = x \cdot s \mod 2$, where $x, s \in \{0,1\}^n$. The problem has a query complexity of $\Omega(n)$ in the classical setting, while it can be solved using only 1 quantum query.

To systematically study learning problems in the quantum setting, Bshouty and Jackson [24] defined the quantum PAC learning model as a generalization of the standard PAC learning model. They also offered a polynomial-time algorithm for learning DNF in the quantum PAC learning model, while there is no polynomial-time algorithm known in the classical PAC learning model. The details about classical and quantum PAC models have been introduced in Section 1.6. The relationship between the number of quantum examples versus classical examples required for PAC learning was first studied by Servedio and Gortler [50], and then improved by Atici and Servedio [12]. In this chapter, I improve the

lower bound given in [12], offering a new quantum lower bound that is extremely close to the lower bound in the classical PAC learning model.

Before offering our main conclusion, we first review the lower bounds offered in [12].

Lemma 1. *Let C be any concept class with VC dimension $d+1$. Any quantum algorithm with a $QEX(c,D)$ oracle to learn a concept belonging to C with ε accuracy and $1-\delta$ confidence, must have quantum query complexity at least*

$$\Omega\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\right).$$

To prove Lemma 1, the authors in [12] consider two concepts c_0, c_1 such that $c_0(x_0) = c_1(x_0) = 0$, while $c_0(x_1) = 0, c_1(x_1) = 1$. Assuming the distribution D satisfies that $D(x_0) = 1-3\varepsilon$ and $D(x_1) = 3\varepsilon$. The states of the system after T queries of $QEX(c,D)$, which are denoted by $|\psi^{(0)}\rangle$ for c_0 and $|\psi^{(1)}\rangle$ for c_1, satisfy

$$\langle\psi^{(0)}|\psi^{(1)}\rangle = (1-3\varepsilon)^T.$$

From [100], if $(1-3\varepsilon)^T > 2\sqrt{\delta(1-\delta)}$, there is some output hypothesis which occurs with probability greater than δ, whether the target is c_0 or c_1; but this cannot be the case for ε-accuracy, $1-\delta$ confidence PAC learning algorithm. Thus $(1-3\varepsilon)^{2T} < 4\delta$ yielding

$$T = \Omega(\frac{1}{\varepsilon}\log\frac{1}{\delta}).$$

Lemma 2. *Let C be any concept class with VC dimension $d+1$. Any quantum algorithm with a $QEX(c,D)$ oracle to learn a concept belonging to C with ε accuracy and $4/5$ confidence, must have quantum query complexity at least*

$$\Omega\left(\frac{1}{\varepsilon}\sqrt{d}\right).$$

To prove Lemma 2, in [12], the authors used the following strategy. First, they designed a set of concepts, such that under a given distribution, the difference of each two concepts is no less than 2ε. Hence, the quantum states generated from such concepts, $|\psi_1\rangle, \cdots, |\psi_N\rangle$, should be distinguished with confidence $4/5$, where N is the number of concepts. Then, by constructing corresponding quantum states $|\phi_i\rangle$ that are close to the quantum states $|\psi_i\rangle^t$, for $i = 1, \cdots, N$, where t will be decided later, such that if $|\psi_i\rangle^{Tt}$ can be distinguished,

then so can $|\phi_i\rangle^T$. In this chapter, we use the notation $|\psi\rangle^t$ to denote $|\psi\rangle^{\otimes t}$. At the same time, from the polynomial-based argument [15], there is a lower bound for the copies of $|\phi_i\rangle$ needed to distinguish each other. Hence from the lower bound for $|\phi_i\rangle$ and the condition that $|\phi_i\rangle$ is close to $|\psi_i\rangle^t$, a lower bound for the copies of $|\psi_i\rangle$, i.e., tT, is derived in [12], which is $\Omega(\sqrt{d}/\varepsilon)$.

The remainder of the current chapter is organized as follows. In Section 3.2, by offering a simple example about how to improve the lower bound given in [12], we show our main idea in proving the general conclusion, while avoiding complicated notation and formulas. In Section 3.3, we provide our main conclusion, showing that the lower bound for query complexity in the quantum PAC learning model is almost the same as that in the classical PAC learning model.

3.2 A Simple Example for Improving Former Results

We find that if we change the quantum states $|\phi_i\rangle$ to be closer to $|\psi_i\rangle^t$, for $i = 1, \cdots, N$, we can derive improved lower bounds compared to the one in [12]. In this section, we will show how to derive a lower bound $\Omega(\varepsilon^{-1}(d/\ln d)^{2/3})$, as a simple example. In the next section, we will go on proving that for an arbitrarily small $\eta > 0$, there is a lower bound $\Omega(\varepsilon^{-1}d^{1-\eta})$ for the query complexity.

To start the proof, we begin with the Gilbert-Varshamov bound [74], which is well known in coding theory.

Lemma 3. *There exists a set $\{z_1, \cdots, z_N\}$ of d-bit binary strings such that for all $i \neq j$ the string z_i and z_j differ in at least $d/4$ bit positions, where*

$$N \geq \frac{2^d}{\sum_{j=0}^{d/4-1} \binom{d}{j}} \geq \frac{2^d}{\sum_{j=0}^{d/4} \binom{d}{j}} \geq 2^{d(1-H(1/4))} > 2^{d/6},$$

where $H(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function.

Theorem 1. *Let C be any concept class with VC dimension $d+1$. Any quantum algorithm with a $QEX(c, D)$ oracle to learn a concept belonging to C with ε accuracy and $4/5$ confidence, must have quantum query complexity at least*

$$\Omega\left(\frac{1}{\varepsilon}(\frac{d}{\ln d})^{2/3}\right).$$

Proof. Let $\{x_0, x_1, \cdots, x_d\}$ be a set of inputs which can be shattered by C. As in [12], we consider the same distribution D, that is

$$D(x_0) = 1 - 8\varepsilon,$$

and

$$D(x_i) = \frac{8\varepsilon}{d},$$

for $i = 1, \cdots, d$.

From Lemma 3, there are N strings, $z_1, \cdots, z_N$, whose length is d, whose pairwise Hamming distance is at least $d/4$, where $N > 2^{d/6}$. Here, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. For $i = 1, \cdots, N$, let c_i be a concept, such that $c_i(x_0) = 0$ for any i, and $c_i(x_j)$ equals to the j-th element in string z_i. Since for any $i \neq j$, we have

$$Pr_D[c_i(x) \neq c_j(x)] \geq \frac{8\varepsilon}{d}\frac{d}{4} = 2\varepsilon,$$

the quantum PAC learning algorithm should distinguish each quantum state with the confidence 4/5. The quantum states generated from $QEX(c, D)$ are

$$\begin{aligned} |\psi_i\rangle &= \sqrt{1-8\varepsilon}|x_0, 0\rangle + \sum_{j=1}^{d}\sqrt{\frac{8\varepsilon}{d}}|x_j, c_i(x_j)\rangle \\ &= \sqrt{1-8\varepsilon}|\alpha\rangle + \sqrt{8\varepsilon}|\beta_i\rangle, \end{aligned} \tag{3.1}$$

where $|\alpha\rangle = |x_0, 0\rangle$ and

$$|\beta_i\rangle = \frac{1}{\sqrt{d}}\sum_{j=1}^{d}|x_j, c_i(x_j)\rangle,$$

for $i = 1, \cdots, N$. Note that $|\alpha\rangle$ and $|\beta_i\rangle$ are all normalized.

Before we approximate $|\psi_i\rangle^{\otimes t}$, we define a series of non-normalized quantum states, which are not normalized. Let

$$|\mu_0(i, n)\rangle = |\alpha\rangle^n$$

and

$$|\mu_k(i, n)\rangle = \sum_{j=0}^{n-k}|\mu_{k-1}(i, n-1-j)\rangle \otimes |\beta_i\rangle \otimes |\alpha\rangle^j.$$

Note that $|\mu_k(i,n)\rangle$ has a direct physical meaning, the sum of all n-composed quantum states such that k of the subsystems are in the state $|\beta_i\rangle$, the others being in the state $|\alpha\rangle$. For example,

$$|\mu_1(i,t)\rangle = |\alpha\rangle^{t-1}|\beta_i\rangle + |\alpha\rangle^{t-2}|\beta_i\rangle|\alpha\rangle + \cdots + |\beta_i\rangle|\alpha\rangle^{t-1}.$$

and

$$\begin{aligned}|\mu_2(i,t)\rangle =&|\alpha\rangle^{t-2}|\beta_i\rangle^2 + |\alpha\rangle^{t-3}|\beta_i\rangle|\alpha\rangle|\beta_i\rangle + \cdots + |\beta_i\rangle|\alpha\rangle^{t-2}|\beta_i\rangle \\ &+ |\alpha\rangle^{t-3}|\beta_i\rangle^2|\alpha\rangle + |\alpha\rangle^{t-4}|\beta_i\rangle|\alpha\rangle|\beta_i\rangle|\alpha\rangle + \cdots + |\beta_i\rangle|\alpha\rangle^{t-3}|\beta_i\rangle|\alpha\rangle \\ &+ \cdots + |\beta_i\rangle^2|\alpha\rangle^{t-2}.\end{aligned}$$

From its physical meaning, it is easy to know that

$$\langle\mu_k(i,n)|\mu_l(i,n)\rangle = \binom{n}{k}\delta_{k,l}, \tag{3.2}$$

where $\delta_{k,l} = 1$ if $k = l$, and $\delta_{k,l} = 0$ if $k \neq l$. Since $|\psi_i\rangle = \sqrt{1-8\varepsilon}|\alpha\rangle + \sqrt{8\varepsilon}|\beta_i\rangle$,

$$|\psi_i\rangle^t = \sum_{k=0}^{t}(1-8\varepsilon)^{(t-k)/2}(8\varepsilon)^{k/2}|\mu_k(i,t)\rangle. \tag{3.3}$$

In the proof in [12], the authors define $|\phi_i\rangle$ as the first two terms in (3.3) with an additional orthogonal term, while in the current proof we define it from the first three terms in (3.3), i.e.,

$$|\phi_i\rangle = (1-8\varepsilon)^{t/2}|\mu_0(i,t)\rangle + (1-8\varepsilon)^{\frac{t-1}{2}}(8\varepsilon)^{1/2}|\mu_1(i,t)\rangle + (1-8\varepsilon)^{\frac{t-2}{2}}(8\varepsilon)|\mu_2(i,t)\rangle + |z\rangle, \tag{3.4}$$

for $i = 1, \cdots, N$, where $|z\rangle$ is an orthogonal term that normalizes $|\phi_i\rangle$.

Note that $|\phi_i\rangle$ is an approximation to $|\psi_i\rangle^t$ with

$$\begin{aligned}\langle\psi_i|^t|\phi_i\rangle \geq&(1-8\varepsilon)^t + (1-8\varepsilon)^{t-1}8\varepsilon t + (1-8\varepsilon)^{t-2}(8\varepsilon)^2\frac{t(t-1)}{2} \\ \geq&1 - 8t\varepsilon + \binom{t}{2}(8\varepsilon)^2 - \binom{t}{3}(8\varepsilon)^3 + 8t\varepsilon(1-8(t-1)\varepsilon) \\ &+ \frac{t(t-1)}{2}(8\varepsilon)^2 - \frac{t(t-1)(t-2)}{2}(8\varepsilon)^3 \\ =&1 - \frac{2t(t-1)(t-2)}{3}(8\varepsilon)^3.\end{aligned} \tag{3.5}$$

Since

$$|x_j, c_i(x_j)\rangle = (1 - c_i(x_j))|x_j, 0\rangle + c_i(x_j)|x_j, 1\rangle,$$
$$|\beta_i\rangle = \sum_{j=1}^{d}(1 - c_i(x_j))|x_j, 0\rangle + \sum_{j=1}^{d} c_i(x_j)|x_j, 1\rangle.$$

We see that $|\mu_k(i,t)\rangle$ is a superposition where each amplitude is a d-variate polynomial whose degree is at most k. Hence, the amplitudes of $|\phi_i\rangle$ are d-variate polynomial with degree at most 2. So, for any measurement of $|\phi_i\rangle^T$, there exist a set of d-variate $4T$-degree polynomials P_i, for $i = 1, \cdots, N$, such that $P_i(z_j)$ equals to the probability of attaining $|\phi_i\rangle$, while the real quantum state is $|\phi_j\rangle$, where $z_j = (c_j(x_1), c_j(x_2), \cdots, c_j(x_d))$.

Consider the scenario in which $|\phi_i\rangle$ can be distinguished with confidence $2/3$ by T copies. Consider an $N \times N$ matrix L, whose (i,j) element $L_{i,j} = P_i(z_j)$. Since

$$L_{i,i} \geq 2/3 > 1/3 \geq \sum_{j \neq i} L_{j,i},$$

we see that L is a strictly diagonally dominant matrix. From the Levy-Desplanques theorem [60], it is known that any strictly diagonally dominant matrix must be of full rank, so the rank of L is N. All the d-variate $4T$-degree polynomials are in the linear space spanned by monic multilinear monomials over d variables of degree at most $2T$. For instance, if $d = 2$ and $T = 1$, the space is spanned by $\{1,\ x_1,\ x_2,\ x_1^2,\ x_1x_2,\ x_2^2,\ x_1^3,\ x_1^2x_2,\ x_1x_2^2,\ x_2^3,\ x_1^4,\ x_1^3x_2,\ x_1^2x_2^2,\ x_1x_2^3,\ x_2^4\}$. The dimension of the linear space is

$$N_T = \sum_{k=0}^{4T} \binom{k+d-1}{k} = \binom{d+4T}{4T}$$

Since each P_i is in an N_T-dimensional space, so are the rows of L, which are

$$(P_i(z_1), P_i(z_2), \cdots, P_i(z_N)).$$

Hence, $N \leq N_T$, i.e,

$$N_T = \binom{d+4T}{4T} \geq N \geq 2^{d/6}. \tag{3.6}$$

If $T < \frac{1}{4}d$, then

$$\binom{d+4T}{4T} \leq (d+4T)^{4T} \leq (2d)^{4T}.$$

Consequently, $(2d)^{4T} \geq 2^{d/6}$, and so that

$$T \geq \frac{\ln 2}{24} \frac{d}{\ln d + \ln 2} \geq \frac{\ln 2}{48} \frac{d}{\ln d} \tag{3.7}$$

for $d \geq 2$. On the other hand, if $T \geq d/4$, T is also greater than $\ln 2/48 \cdot d/\ln d$. Hence, to distinguish $|\phi_i\rangle$, we need

$$T \geq \frac{ln2}{48} \frac{d}{\ln d} \tag{3.8}$$

copies.

If

$$\langle \phi_i |^{\otimes T} |\psi_i\rangle^{\otimes tT} \geq \frac{5}{6},$$

and $|\psi_i\rangle^{\otimes tT}$ can be distinguished with confidence $4/5$, then $|\phi_i\rangle^T$ can be distinguished with confidence $2/3$ by the same measurement. From (3.5), when

$$t = \frac{1}{8\varepsilon} \left(\frac{1}{4T}\right)^{1/3}, \tag{3.9}$$

the condition is satisfied, hence (3.8) holds. Combing (4.15) and (3.8), we see that

$$tT \geq \frac{1}{8} \left(\frac{1}{4}\right)^{1/3} \left(\frac{\ln 2}{48}\right)^{2/3} \left(\frac{1}{\varepsilon}\right) \left(\frac{d}{\ln d}\right)^{2/3} \geq 0.004 \left(\frac{1}{\varepsilon}\right) \left(\frac{d}{\ln d}\right)^{2/3}.$$

Hence there exists a lower bound for the query complexity, which is

$$\Omega\left(\frac{1}{\varepsilon} \left(\frac{d}{\ln d}\right)^{2/3}\right)$$

. □

In the analysis above, the lower bound for query complexity is improved by approximating $|\psi_i\rangle^t$ with more terms from (3.3). This idea offers us a strategy to further improve the lower bound by using more and more terms from (3.3) to construct $|\phi_i\rangle$, we show in the next section.

3.3 Lower bounds for query complexity in PAC learning model

We start this section with a lemma about the tails of the binomial distribution [31], which will be used in the proof of the main result of this chapter.

Lemma 4. *Consider a sequence of n Bernoulli trials, where success occurs with probability p. Let X be the random variable denoting the total number of successes. Then for $0 \leq k \leq n$, the probability of at least k successes is*

$$\Pr\{X \geq k\} = \sum_{i=k}^{n} b(i; n, p) \leq \binom{n}{k} p^k,$$

where

$$b(i; n, p) = \binom{n}{i} p^i (1-p)^{n-i}.$$

Proof. For $S \subseteq \{1, 2, \cdots, n\}$, we let N_s denote the event that the i-th trial is a success for every $i \in S$. Clearly $Pr\{N_s\} = p^k$ if $|S| = k$. We have

$$\begin{aligned}
\Pr\{X \geq k\} &= \Pr\{\text{there exists } S \subseteq \{1, 2, \cdots, n\} : |S| = k \text{ and } N_s\} \\
&= \Pr\{\bigcup_{S \subseteq \{1,2,\cdots,n\}:|S|=k} N_s\} \\
&\leq \sum_{S \subseteq \{1,2,\cdots,n\}:|S|=k} \Pr\{N_s\} \\
&= \binom{n}{k} p^k.
\end{aligned}$$

□

Then, we begin to prove the main result of this chapter.

Theorem 2. *Let C be any concept class of VC dimension $d+1$. Any quantum algorithm with a $QEX(c, D)$ oracle to learn a concept belonging to C with ε accuracy and $4/5$ confidence; must have quantum query complexity*

$$\Omega\left(\frac{1}{\varepsilon} d^{1-\eta}\right),$$

for an arbitrarily small constant $\eta > 0$.

Proof. Similarly to the proof of Theorem 1, let $\{x_0, x_1, \cdots, x_d\}$ be a set of inputs which can be shattered by C. We consider the same distribution D, and the same concepts $c_1, \cdots, c_N$ given in the proof of Theorem 1, where $N > 2^{d/6}$, then

$$Pr_D[c_i(x) \neq c_j(x)] \geq \frac{8\varepsilon}{d} \frac{d}{4} = 2\varepsilon.$$

Hence, the quantum PAC learning algorithm should successfully distinguish between any two concepts c_i and c_j with confidence 4/5, i.e., it can distinguish each quantum state generated by $QEX(c, D)$ with confidence 4/5, where the quantum states are

$$|\psi_i\rangle = \sqrt{1-8\varepsilon}|x_0, 0\rangle + \sum_{j=1}^{d} \sqrt{\frac{8\varepsilon}{d}}|x_j, c_i(x_j)\rangle.$$

As in the proof of Theorem 1, we will construct some quantum state $|\phi_i\rangle$ which is close to $|\psi_i\rangle^t$ such that if $|\psi_i\rangle^{Tt}$ can be distinguished, so can $|\phi_i\rangle^T$, where t will be decided later. Then, by providing a lower bound for the copies of $|\phi_i\rangle$ needed to distinguish each other, we can derive the lower bound for the copies of $|\psi_i\rangle$.

From (3.3), we see that

$$|\psi_i\rangle^t = \sum_{k=0}^{t} (1-8\varepsilon)^{\frac{t-k}{2}} (8\varepsilon)^{\frac{k}{2}} |\mu_k(i,t)\rangle. \tag{3.10}$$

for $i = 1, \cdots, N$, where $\mu_k(i,t)$ has the same definition as in the above section. Then we define quantum states $|\phi_i\rangle$,

$$|\phi_i\rangle = \sum_{k=0}^{s-1} (1-8\varepsilon)^{\frac{t-k}{2}} (8\varepsilon)^{\frac{k}{2}} |\mu_k(i,t)\rangle, \tag{3.11}$$

where $s = 2/\eta$ as an approximation of $|\psi_i\rangle^t$. Then

$$\langle\psi_i|^t|\phi_i\rangle \geq \sum_{k=0}^{s-1} \binom{t}{k} (8\varepsilon)^k (1-8\varepsilon)^{t-k} \geq 1 - \binom{t}{s} (8\varepsilon)^s. \tag{3.12}$$

The last equation comes from the tail bound of the binomial distribution as given in Lemma 4:

$$\sum_{i=k}^{n} b(i; n, p) \leq \binom{n}{k} p^k.$$

Consider the scenario that $|\phi_i\rangle$ can be distinguished with confidence 2/3 by T copies. Since each amplitude in $|\mu_k(i,t)\rangle$ is a d-variate polynomial whose degree is at most k, the amplitude of $|\phi_i\rangle$ are d-variate polynomials with degree at most s. Hence there exists d-variate $2sT$-degree polynomials P_i for $i = 1, \cdots, N$, such that $P_i(z_j)$ equals to the probability of attaining $|\phi_i\rangle$ when the real quantum state is $|\phi_j\rangle$, where $z_j = (c_j(x_1), c_j(x_2), \cdots, c_j(x_d))$.

Through similar analysis to that given in the proof of Theorem 1, we know that if $|\phi_i\rangle$ can be distinguished with confidence $2/3$ by T copies, then

$$N_T = \binom{d+2sT}{2sT} \geq N \geq 2^{d/6},$$

where N_T is the dimension of the linear space spanned by d-variate $2sT$-degree polynomials. So,

$$T \geq \frac{\ln 2}{24s}\frac{d}{\ln d}. \tag{3.13}$$

Moreover, if

$$\langle\phi_i|^{\otimes T}|\psi_i\rangle^{\otimes tT} \geq \frac{5}{6}, \tag{3.14}$$

and $|\psi_i\rangle$ can be distinguished with confidence $4/5$ by tT copies, $|\phi_i\rangle^{\otimes T}$ can be distinguished with confidence $2/3$ by the same measurement. To satisfy (3.14), from (3.12), it is enough to let

$$t = \frac{1}{8\varepsilon}(\frac{1}{6T})^{1/s}. \tag{3.15}$$

Combining (3.15) and (3.13), the lower bound of the query complexity is

$$Tt = \left(\frac{1}{6}\right)^{1/s}\left(\frac{\ln 2}{24s}\right)^{1-1/s}\left(\frac{1}{8\varepsilon}\right)\left(\frac{d}{\ln d}\right)^{1-1/s} = \Omega\left(\frac{1}{\varepsilon}\left(\frac{d}{\ln d}\right)^{1-1/s}\right).$$

Since $s = 2/\eta$, then the number of quantum queries is at least $Tt \geq \Omega(\varepsilon^{-1}\, d^{1-e})$. □

From Lemma 1, there is another lower bound for the number of queries of quantum PAC learning model, which is

$$\Omega(\frac{1}{\varepsilon}\log\frac{1}{\delta}).$$

Combining it with Theorem 2, we have the following theorem.

Theorem 3. *Let C be any concept class of VC dimension $d+1$. Any quantum algorithm with a $QEX(c,D)$ oracle to learn a concept belonging to C with ε accuracy and $1-\delta$ confidence, must have quantum query complexity*

$$\Omega\left(\frac{1}{\varepsilon}(d^{1-\eta} + \log\frac{1}{\delta})\right),$$

for an arbitrarily small constant $\eta > 0$.

3.4 Discussion

In this chapter, we have improved the lower bound of query complexity in quantum PAC learning model. The lower bound is quite close to the best lower bound known in the classical PAC learning model. In this sense, we seem to have almost optimally solved the problem.

Although the lower bounds of query complexity are almost the same in both the classical PAC learning model and the quantum PAC learning model, quantum computing still has its superiority in learning theory field considering computational complexity. Under the assumption that factoring is computationally hard for classical computers, polynomial-time quantum learning is provably more powerful than polynomial-time classical learning [49]. This follows directly from the observation that Shor's quantum factoring algorithm enables quantum algorithms to efficiently learn concept classes whose classical learnability is directly related to the hardness of factoring [9; 66; 102]. Moreover, if any one-way function exists, then there is a concept class C that is polynomial-time learnable from quantum membership queries but is not polynomial-time learnable from classical membership queries [98]. Here, a one-way function is a function that is easy to compute on every input, but hard to invert [48].

There are also other learning problems, such as learning DNF [24] and learning juntas [11], for which quantum learning algorithms beat classical learning algorithms observably. Here, juntas are Boolean functions which depend only on an unknown set out of the input variables.

Chapter 4

Efficiency of Hamiltonian Simulation

4.1 Introduction

In the Hamiltonian simulation problem one is given a Hamiltonian H, $t \in \mathbb{R}$ and an accuracy demand ε and the goal is to derive an algorithm approximating the unitary operator e^{-iHt} with error at most ε. The size of the quantum circuit realizing the algorithm is its cost. Assuming that H is a matrix of size $2^q \times 2^q$ the algorithm is efficient if its cost is a polynomial in q, t and ε^{-1}.

Lloyd [76] showed that local Hamiltonians can be simulated efficiently on a quantum computer. About the same time, Zalka [123; 124] showed that many-particle systems can be simulated efficiently on a quantum computer. Later, Aharonov and Ta-Shma [4] generalized Lloyd's results to sparse Hamiltonians. Berry et al. [17] extended the complexity results of [4] for sparse Hamiltonians.

Recall that *splitting formulas* such as the Lie-Trotter formula

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

have been extensively used in quantum simulation. From this we have a second order approximation

$$e^{-i(H_1+H_2)\Delta t} = e^{-iH_1\Delta t} e^{-iH_2\Delta t} + O(|\Delta t|^2).$$

A third order approximation is given by the Strang splitting

$$e^{-i(H_1+H_2)\Delta t} = e^{-iH_1\Delta t/2}e^{-iH_2\Delta t}e^{-iH_1\Delta t/2} + O(|\Delta t|^3).$$

Suzuki [105; 106] uses recursive modifications of this approximation to derive formulas of order $2k+1$, for $k = 1, 2, \ldots$. In the splitting methods, the unitary evolution e^{-iHt} is simulated by a product of exponentials of H_{j_s} for some sequence of j_s and intervals t_s, i.e., $\hat{U} = \prod_{s=1}^{N} e^{-iH_{j_s}t_s}$, where $H = \sum_{j=1}^{m} H_j$, $j_s \in \{1, 2, \cdots, m\}$ and assuming the Hamiltonians H_j do not commute. It is further assumed that the H_j can be implemented efficiently. Throughout this chapter we assume that the H_j are either Hermitian matrices or bounded Hermitian operators, so that $\|H_j\| < \infty$ for $j = 1, \ldots, m$.

In Section 1.7, we introduce Suzuki's high order splitting methods. A recent paper [17] shows that these methods can be used to derive upper bounds for the number N of exponentials, assuming the H_j are local Hamiltonians. These bounds are expressed in terms of the evolution time t, the norm $\|H\|$ of the Hamiltonian H, the order of the splitting method $2k+1$, the number of Hamiltonians m, and the error ε in the approximation of e^{-iHt}. In this chapter we will show how these bounds can be significantly improved.

Consider the Hamiltonians indexed with respect to the magnitude of their norms $\|H_1\| \geq \|H_2\| \geq \cdots \geq \|H_m\|$. Then the number of necessary exponentials N generally depends on H_1, but it must also depend explicitly on H_2 since only one exponential should suffice for the simulation if $\|H_2\| \to 0$. This observation is particularly important for the simulation of systems in physics and chemistry. To see this, suppose $m = 2$ and that H_1 is a discretization of the negative Laplacian $-\Delta$, while H_2 is a discretization of a uniformly bounded potential. Then $e^{-iH_1t_1}$ and $e^{-iH_2t_2}$ can be implemented efficiently for any t_1, t_2, and $\|H_2\| \ll \|H_1\|$. We will see that, not only in this case but in general, the number of exponentials is proportional to both $\|H_1\|$ and $\|H_2\|$; i.e., the Hamiltonian of the second largest norm plays an important role.

Let ε be sufficiently small. The previously known bound for the number of exponentials, according to [17], is

$$N \leq N_{\text{prev}} := 2m5^{2k}(m\|H\|t)^{1+\frac{1}{2k}}\varepsilon^{-1/(2k)}. \tag{4.1}$$

This bound does not properly reflect the dependence on H_2.

Performing a more detailed analysis of the approximation error by high order splitting formulas, it is possible to improve the bounds for N substantially. The new estimates lead to *optimal* splitting methods of significantly lower order which greatly reduces the cost of the algorithms.

The remainder of the current chapter is organized as follows. In Section 4.2, we consider the number of exponentials needed in simulating the sum of two Hamiltonians, illustrating the main idea of our approach. In Section 4.3, we provide a new bound for the number of exponentials N, given by

$$N \le N_{\text{new}} := (2m-1)\,5^{k-1}\left\lceil \|H_1\| t \left(\frac{4emt\|H_2\|}{\varepsilon}\right)^{1/(2k)} \frac{4me}{3}\left(\frac{5}{3}\right)^{k-1}\right\rceil.$$

The speedup factor compared to the best result known before [17] is

$$\frac{N_{\text{new}}}{N_{\text{prev}}} \le \frac{1}{3^k}\left(\frac{4e\|H_2\|}{\|H_1\|}\right)^{1/2k}.$$

In Section 4.4, we show that the *optimal* k^*_{new} that minimizes N_{new} is

$$k^*_{\text{new}} := \left\lceil \sqrt{\frac{1}{2}\log_{25/3}\frac{4emt\|H_2\|}{\varepsilon}}\right\rceil.$$

On the other hand, from [17] the bound for N_{prev} is minimized for

$$k^*_{\text{prev}} = \frac{1}{2}\sqrt{\log_5 \frac{m\|H\|t}{\varepsilon} + 1}.$$

For k^*_{new} the value of N_{new} satisfies

$$N^*_{\text{new}} \le \frac{8}{3}(2m-1)\,met\,\|H_1\|\; e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln\frac{4emt\|H_2\|}{\varepsilon}}}.$$

For k^*_{prev} the value of N_{prev} is

$$N^*_{\text{prev}} = 2m^2\|H\|t\cdot e^{2\sqrt{\ln 5\ln(m\|H\|t/\epsilon)}}.$$

Hence

$$\frac{N^*_{\text{new}}}{N^*_{\text{prev}}} \le \frac{8\,e}{3}\,e^{2\left(\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln\frac{4emt\|H_2\|}{\varepsilon}}-\sqrt{\ln 5\ln\frac{m\|H_1\|t}{\varepsilon}}\right)}.$$

4.2 Splitting methods for simulating the sum of two Hamiltonians

We begin this section by discussing the simulation of

$$e^{-i(H_1+H_2)t},$$

where H_1, H_2 are given Hamiltonians. Restricting the analysis to $m = 2$ will allow us to illustrate the main idea in our approach while avoiding the rather complicated notation needed in the general case for $m \geq 2$. The simulation of the Schrödinger equation of a p-particle system, where H_1 is obtained from the Laplacian operator and H_2 is the potential, requires one to consider an evolution operator that has the form above; see [64].

In the next section we deal with the more general simulation problem involving a sum of m Hamiltonians, $H_1, \ldots, H_m$, as Berry et al. [17] did, and we will show how to improve their complexity results.

Suzuki proposed methods for decomposing exponential operators in a number of papers [105; 106]. For sufficiently small Δt, starting from the formula

$$S_2(H_1, H_2, \Delta t) = e^{-iH_1\Delta t/2} e^{-iH_2\Delta t} e^{-iH_1\Delta t/2},$$

and proceeding recursively, Suzuki defines

$$S_{2k}(H_1, H_2, \Delta t) = [S_{2k-2}(H_1, H_2, p_k\Delta t)]^2 S_{2k-2}(H_1, H_2, (1-4p_k)\Delta t)[S_{2k-2}(H_1, H_2, p_k\Delta t)]^2,$$

for $k = 2, 3, \cdots$, where $p_k = (4 - 4^{1/(2k-1)})^{-1}$, and then proves that

$$\left\| e^{-i(H_1+H_2)\Delta t} - S_{2k}(H_1, H_2, \Delta t) \right\| = O(|\Delta t|^{2k+1}). \tag{4.2}$$

Suzuki was particularly interested in the order of his method, which is $2k + 1$, and did not address the size of the implied asymptotic factors in the big-O notation. However, these factors depend on the norms of H_1 and H_2 and can be very large, when H_1 and H_2 do not commute. For instance, when H_1 is obtained from the discretization of the Laplacian operator with mesh size h, $\|H_1\|$ grows as h^{-2}. Since $h = \varepsilon$, we get $\|H_1\| = O(\varepsilon^{-2})$. Hence, for fine discretizations $\|H_1\|$ is huge, which severely affects the error bound above.

Suppose $\|H_1\| \geq \|H_2\|$. Since

$$e^{-i(H_1+H_2)t} = e^{-i(\mathcal{H}_1+\mathcal{H}_2)\|H_1\|t},$$

where $\mathcal{H}_j = H_j/\|H_1\|$, for $j = 1, 2$, we can consider the simulation problem for $\mathcal{H}_1 + \mathcal{H}_2$ with an evolution time $\tau = \|H_1\|t$.

Unwinding the recurrence in Suzuki's construction yields

$$S_{2k}(\mathcal{H}_1, \mathcal{H}_2, \Delta t) = \prod_{\ell=1}^{K} S_2(\mathcal{H}_1, \mathcal{H}_2, z_\ell \Delta t) = \prod_{\ell=1}^{K} \left[e^{-i\mathcal{H}_1 z_\ell \Delta t/2} e^{-i\mathcal{H}_2 z_\ell \Delta t} e^{-i\mathcal{H}_1 z_\ell \Delta t/2} \right], \quad (4.3)$$

where $K = 5^{k-1}$ and each z_ℓ is defined according to the recursive scheme, $\ell = 1, \ldots, K$. In particular, $z_1 = z_K = \prod_{r=2}^{k} p_r$, and for the intermediate values of ℓ the z_ℓ is a product of $k-1$ factors and has the form $z_\ell = \prod_{r\in I_0} p_r \prod_{r\in I_1}(1-4p_r)$, where the products are over the index sets I_0, I_1 defined by traversing the corresponding to the ℓ path of the recursion tree.

Let $q_r = \max\{p_r, 4p_r - 1\}$, $r \geq 2$. Then $\{q_r\}$ is a decreasing sequence of positive numbers and from [118] we have that

$$\frac{3}{3^k} \leq \prod_{r=2}^{k} q_r \leq \frac{4k}{3^k}.$$

Thus

$$|z_\ell| \leq \frac{4k}{3^k} \quad \text{for all } \ell = 1, \ldots, K. \quad (4.4)$$

Equation (4.3) can be expressed in the more compact form, which we use to simplify the notation, namely,

$$S_{2k}(\mathcal{H}_1, \mathcal{H}_2, \Delta t) = e^{-i\mathcal{H}_1 s_0 \Delta t} e^{-i\mathcal{H}_2 z_1 \Delta t} e^{-i\mathcal{H}_1 s_1 \Delta t} \cdots e^{-i\mathcal{H}_2 z_K \Delta t} e^{-i\mathcal{H}_1 s_K \Delta t}, \quad (4.5)$$

where $s_0 = z_1/2$, $s_j = (z_j + z_{j+1})/2$, $j = 1, \ldots, K-1$, and $s_K = z_K/2$. Observe that $\sum_{j=0}^{K} s_j = 1$, $\sum_{j=1}^{K} z_j = 1$.

We need to bound $\sigma_k = \sum_{j=0}^{K} |s_j| + \sum_{j=1}^{K} |z_j|$ from above. From Eq.(4.4) we have

$$\sum_{j=1}^{K} |z_j| \leq \frac{4k5^{k-1}}{3^k},$$

and also

$$\sum_{j=0}^{K} |s_j| \leq \frac{4k5^{k-1}}{3^k}.$$

Thus

$$\sigma_k \leq \frac{8}{3}k\left(\frac{5}{3}\right)^{k-1} =: c_k \quad \text{for } k \geq 1. \tag{4.6}$$

(The inequality 4.6 trivially holds for $k = 1$.)

Expanding each exponential in Eq.(4.5) we obtain

$$\begin{aligned} S_{2k}(\mathcal{H}_1, \mathcal{H}_2, \Delta t) =& (I + \mathcal{H}_1 s_0(-i\Delta t) + \frac{1}{2}\mathcal{H}_1^2 s_0^2(-i\Delta t)^2 + \cdots + \frac{1}{k!}\mathcal{H}_1^k s_0^k(-i\Delta t)^k + \cdots) \\ & (I + \mathcal{H}_2 z_1(-i\Delta t) + \frac{1}{2}\mathcal{H}_2^2 z_1^2(-i\Delta t)^2 + \cdots + \frac{1}{k!}\mathcal{H}_2^k z_1^k(-i\Delta t)^k + \cdots) \\ & (I + \mathcal{H}_1 s_1(-i\Delta t) + \frac{1}{2}\mathcal{H}_1^2 s_1^2(-i\Delta t)^2 + \cdots + \frac{1}{k!}\mathcal{H}_1^k s_1^k(-i\Delta t)^k + \cdots) \\ & \vdots \\ & (I + \mathcal{H}_2 z_K(-i\Delta t) + \frac{1}{2}\mathcal{H}_2^2 z_K^2(-i\Delta t)^2 + \cdots + \frac{1}{k!}\mathcal{H}_2^k z_K^k(-i\Delta t)^k + \cdots) \\ & (I + \mathcal{H}_1 s_K(-i\Delta t) + \frac{1}{2}\mathcal{H}_1^2 s_K^2(-i\Delta t)^2 + \cdots + \frac{1}{k!}\mathcal{H}_1^k s_K^k(-i\Delta t)^k + \cdots). \end{aligned} \tag{4.7}$$

After carrying out the multiplications we see that S_{2k} is a sum of terms that has the form

$$\frac{s_0^{\alpha_0} s_1^{\alpha_1} \cdots s_K^{\alpha_K} z_1^{\beta_1} \cdots z_K^{\beta_K}}{\alpha_0!\alpha_1! \cdots \alpha_K!\beta_1! \cdots \beta_K!} \mathcal{H}_1^{\alpha_0}\mathcal{H}_2^{\beta_1}\mathcal{H}_1^{\alpha_1} \cdots \mathcal{H}_2^{\beta_K}\mathcal{H}_1^{\alpha_K}(-i\Delta t)^{\sum_{i=0}^K \alpha_i + \sum_{j=1}^K \beta_j}, \tag{4.8}$$

where the $\alpha_0, \alpha_1, \cdots, \alpha_K$ and the $\beta_1, \cdots, \beta_K$ are obtained by multiplying the denominators in the expansion of the exponentials.

The terms that do not contain $\mathcal{H}_2$ are those for which $\beta_1 = \beta_2 = \cdots = \beta_K = 0$, and their sum is

$$\begin{aligned} & \sum_{\alpha_0,\alpha_1,\cdots,\alpha_K} \frac{s_0^{\alpha_0} s_1^{\alpha_1} \cdots s_K^{\alpha_K}}{\alpha_0!\alpha_1! \cdots \alpha_K!} \mathcal{H}_1^{\sum_{j=0}^K \alpha_j}(-i\Delta t)^{\sum_{j=0}^K \alpha_j} \\ &= \sum_{\alpha_0} \frac{1}{\alpha_0!}\mathcal{H}_1^{\alpha_0}(-is_0\Delta t)^{\alpha_0} \cdot \sum_{\alpha_1} \frac{1}{\alpha_1!}\mathcal{H}_1^{\alpha_1}(-is_1\Delta t)^{\alpha_1} \cdot \cdots \cdot \sum_{\alpha_K} \frac{1}{\alpha_K!}\mathcal{H}_1^{\alpha_K}(-is_K\Delta t)^{\alpha_K} \\ &= \prod_{j=0}^K e^{-i\mathcal{H}_1 s_j \Delta t} = \exp(-i\sum_{j=0}^K \mathcal{H}_1 s_j \Delta t) = \exp(-i\mathcal{H}_1\Delta t). \end{aligned} \tag{4.9}$$

On the other hand, consider

$$e^{-i(\mathcal{H}_1+\mathcal{H}_2)\Delta t} = I + (-i(\mathcal{H}_1 + \mathcal{H}_2)\Delta t) + \cdots + \frac{1}{k!}(-i(\mathcal{H}_1 + \mathcal{H}_2)\Delta t)^k + \cdots. \tag{4.10}$$

The terms that do not contain $\mathcal{H}_2$ sum to

$$\sum_{k=0}^{\infty} \frac{1}{k!}\mathcal{H}_1^k(-i\Delta t)^k = e^{-i\mathcal{H}_1\Delta t}. \tag{4.11}$$

Let us now consider the bound in (4.2). Clearly the terms that do not contain $\mathcal{H}_2$ cancel out. Therefore, the error is proportional to $\|\mathcal{H}_2\||\Delta t|^{2k+1}$, i.e., it depends on the ratio $\|H_2\|/\|H_1\|$ of the norms of the original Hamiltonians. This fact will be used to improve the error and complexity results of Berry et al. [17]

Lemma 5. *For $k \in \mathbb{N}$, $c_k|\Delta t| \leq k+1$ (see, Eq.(4.6)) and $\|\mathcal{H}_2\| \leq \|\mathcal{H}_1\| = 1$ we have*

$$\|\exp(-i(\mathcal{H}_1+\mathcal{H}_2)\Delta t) - S_{2k}(\mathcal{H}_1,\mathcal{H}_2,\Delta t)\| \leq \frac{4\|\mathcal{H}_2\|}{(2k+1)!}(c_k|\Delta t|)^{2k+1}. \tag{4.12}$$

Proof. For notational convenience we use $S_{2k}(\Delta t)$ to denote $S_{2k}(\mathcal{H}_1,\mathcal{H}_2,\Delta t)$. Consider

$$\exp(-i(\mathcal{H}_1+\mathcal{H}_2)\Delta t) - S_{2k}(\Delta t) = \sum_{l=2k+1}^{\infty}\big[R_l(\Delta t) - T_l(\Delta t)\big], \tag{4.13}$$

where $R_l(\Delta t)$ is the sum of all terms in $\exp(-i(\mathcal{H}_1+\mathcal{H}_2)\Delta t)$ corresponding to Δt^l and $T_l(\Delta t)$ is the sum of all terms in $S_{2k}(\Delta t)$ corresponding to Δt^l. We know that the terms with only $\mathcal{H}_1$ cancel out. Hence, we can ignore the terms in $T_l(\Delta t)$ and $R_l(\Delta t)$ that contain only $\mathcal{H}_1$ (and not $\mathcal{H}_2$) as a factor. It follows that

$$R_l(\Delta t) = \frac{1}{l!}(\mathcal{H}_1+\mathcal{H}_2)^l(-i\Delta t)^l - \frac{1}{l!}\mathcal{H}_1^l(-i\Delta t)^l. \tag{4.14}$$

Then

$$\|R_l(\Delta t)\| \leq \frac{1}{l!}2^l\|\mathcal{H}_2\||\Delta t|^l, \tag{4.15}$$

since there are 2^l-1 terms, and they are bounded by $\frac{1}{l!}\|\mathcal{H}_2\||\Delta t|^l$.

Now consider the terms in $T_l(\Delta t)$. From Eq.(4.7,4.8)

$$T_l(\Delta t) = \sum_{\substack{\sum_{i=0}^K \alpha_i + \sum_{i=1}^K \beta_i = l \\ \sum_{i=1}^K \beta_i \neq 0}} \frac{s_0^{\alpha_0} s_1^{\alpha_1}\cdots s_K^{\alpha_K} z_1^{\beta_1}\cdots z_K^{\beta_K}}{\alpha_0!\alpha_1!\cdots\alpha_K!\beta_1!\cdots\beta_K!}\mathcal{H}_1^{\alpha_0}\mathcal{H}_2^{\beta_1}\mathcal{H}_1^{\alpha_1}\cdots\mathcal{H}_2^{\beta_K}\mathcal{H}_1^{\alpha_K}(-i\Delta t)^l, \tag{4.16}$$

where the condition $\sum_{i=1}^K \beta_i \neq 0$ holds because there are no terms containing $\mathcal{H}_1$ alone. Since the norm of $\mathcal{H}_1^{\alpha_0}\mathcal{H}_2^{\beta_1}\mathcal{H}_1^{\alpha_1}\cdots\mathcal{H}_2^{\beta_K}\mathcal{H}_1^{\alpha_K}$ is at most $\|\mathcal{H}_2\|$, we have

$$\|T_l(\Delta t)\| \leq \sum_{\sum_{i=0}^K \alpha_i + \sum_{i=1}^K \beta_i = l} \frac{|s_0^{\alpha_0} s_1^{\alpha_1}\cdots s_K^{\alpha_K} z_1^{\beta_1}\cdots z_K^{\beta_K}|}{\alpha_0!\alpha_1!\cdots\alpha_K!\beta_1!\cdots\beta_K!}\|\mathcal{H}_2\||\Delta t|^l. \tag{4.17}$$

Note that we relaxed the condition $\sum_{i=1}^K \beta_i \neq 0$ since it does not affect the inequality.

To calculate the sum

$$\sum \frac{|s_0^{\alpha_0} s_1^{\alpha_1} \cdots s_K^{\alpha_K} z_1^{\beta_1} \cdots z_K^{\beta_K}|}{\alpha_0!\alpha_1! \cdots \alpha_K! \beta_1! \cdots \beta_K!},$$

where $\sum_{i=0}^{K} \alpha_i + \sum_{i=1}^{K} \beta_i = l$, we first consider the equation

$$\begin{aligned}
&\exp(|s_0\Delta t|)\exp(|z_1\Delta t|)\exp(|s_1\Delta t|)\cdots\exp(|z_K\Delta t|)\exp(|s_K\Delta t|) \\
&= \left(\sum_{\alpha_0=0}^{\infty} \frac{1}{\alpha_0!}|s_0\Delta t|^{\alpha_0}\right)\cdot\left(\sum_{\beta_1=0}^{\infty} \frac{1}{\beta_1!}|z_1\Delta t|^{\beta_0}\right)\cdot\left(\sum_{\alpha_1=0}^{\infty} \frac{1}{\alpha_1!}|s_1\Delta t|^{\alpha_0}\right)\cdots \\
&\qquad \cdots\cdot\left(\sum_{\beta_K=0}^{\infty} \frac{1}{\beta_K!}|z_K\Delta t|^{\beta_K}\right)\cdot\left(\sum_{\alpha_K=0}^{\infty} \frac{1}{\alpha_K!}|s_K\Delta t|^{\alpha_K}\right) \\
&= \sum_{p=0}^{\infty}\sum_{\sum\alpha_j+\sum\beta_j=p} \frac{|s_0^{\alpha_0} s_1^{\alpha_1} \cdots s_K^{\alpha_K} z_1^{\beta_1} \cdots z_K^{\beta_K}|}{\alpha_0!\alpha_1! \cdots \alpha_K! \beta_1! \cdots \beta_K!}|\Delta t|^p.
\end{aligned} \tag{4.18}$$

Hence $\sum_{\sum\alpha_j+\sum\beta_j=l} \frac{|s_0^{\alpha_0} s_1^{\alpha_1} \cdots s_K^{\alpha_K} z_1^{\beta_1} \cdots z_K^{\beta_K}|}{\alpha_0!\alpha_1!\cdots\alpha_K!\beta_1!\cdots\beta_K!}$ is the coefficient of $|\Delta t|^l$ in the equation above. Similarly,

$$\begin{aligned}
&\exp(|s_0\Delta t|)\exp(|z_1\Delta t|)\exp(|s_1\Delta t|)\cdots\exp(|z_K\Delta t|)\exp(|s_K\Delta t|) \\
&= \exp((\sum_{i=0}^{K}|s_i| + \sum_{i=1}^{K}|z_i|)|\Delta t|) = \exp(\sigma_k|\Delta t|) \\
&= \sum_{p=0}^{\infty}\frac{1}{p!}\sigma_k^p|\Delta t|^p,
\end{aligned} \tag{4.19}$$

Recall that the bound for σ_k given in Eq.(4.6). Thus the coefficient of $|\Delta t|^l$ is bounded from above by $c_k^l/l!$. Therefore, we have

$$\|T_l(\Delta t)\| \leq \frac{c_k^l}{l!}\|\mathcal{H}_2\||\Delta t|^l. \tag{4.20}$$

We combine (4.15) and (4.20) to obtain

$$\begin{aligned}
\|\exp((\mathcal{H}_1+\mathcal{H}_2)\Delta t) - S_{2k}(\Delta t)\| &\le \sum_{l=2k+1}^{\infty} \|R_l(\Delta t) - T_l(\Delta t)\| \\
&\le \sum_{l=2k+1}^{\infty} \|R_l(\Delta t)\| + \|T_l(\Delta t)\| \\
&\le 2\sum_{l=2k+1}^{\infty} \frac{c_k^l}{l!}\|\mathcal{H}_2\||\Delta t|^l \\
&\le \frac{2}{(2k+1)!}\|\mathcal{H}_2\||c_k\Delta t|^{2k+1}\left(1-\frac{c_k|\Delta t|}{2k+2}\right)^{-1} \\
&\le \frac{4}{(2k+1)!}\|\mathcal{H}_2\||c_k\Delta t|^{2k+1},
\end{aligned} \tag{4.21}$$

where the last two inequalities follow from the assumption $c_k|\Delta t| \le k+1$ and an estimate of the tail of the Poisson distribution; see, e.g., [70]. □

Theorem 4. *Suppose that $\|H_2\| \le \|H_1\|$. Let $0 < \varepsilon \le 1$ be such that $\varepsilon \le 8et\|H_2\|$. The number N of exponentials for the simulation of $e^{-i(H_1+H_2)t}$ with accuracy ε is bounded by*

$$N \le 3\ 5^{k-1}\left\lceil \|H_1\|t\left(\frac{8et\|H_2\|}{\varepsilon}\right)^{1/(2k)}\frac{8e}{3}\left(\frac{5}{3}\right)^{k-1}\right\rceil,$$

for any $k \in \mathbb{N}$.

Proof. Let $M = |\Delta t|^{-1}$. Then using Lemma 5 and the fact that $\mathcal{H}_j = H_j/\|H_1\|$, $j = 1, 2$, we obtain

$$\begin{aligned}
\left\|e^{-i(H_1+H_2)t} - S_{2k}^{M\|H_1\|t}(\mathcal{H}_1,\mathcal{H}_2,1/M)\right\| &\le M\|H_1\|t\frac{4}{(2k+1)!}\|\mathcal{H}_2\|\left(\frac{c_k}{M}\right)^{2k+1} \\
&= 4t\|H_2\|\frac{c_k^{2k+1}}{(2k+1)!}\frac{1}{M^{2k}}.
\end{aligned}$$

Recall that c_k is defined in Eq.(4.6) and is used in Lemma 5. For accuracy ε, we obtain

$$M \ge \left(\frac{4t\|H_2\|c_k^{2k+1}}{\varepsilon(2k+1)!}\right)^{1/(2k)}.$$

Using Stirling's formula [1] for the factorial function, we have

$$(2k+1)! = \sqrt{2\pi}(2k+1)^{(2k+1)+1/2}e^{-(2k+1)+\theta/(12(2k+1))}, \quad 0 < \theta < 1,$$

which yields

$$[(2k+1)!]^{-1/(2k)} \le e^{1+1/(2k)}/(2k+1). \tag{4.22}$$

It is easy to check that

$$c_k^{1/(2k)} \leq 2^{1+1/(2k)}.$$

Thus it suffices to take

$$M \geq \left(\frac{8et\|H_2\|}{\varepsilon}\right)^{1/(2k)} \frac{2\,e\;c_k}{2k+1}.$$

So we define M to be lower bound of the expression above, i.e.,

$$M := \left(\frac{8et\|H_2\|}{\varepsilon}\right)^{1/(2k)} \frac{2e\;c_k}{2k+1}.$$

It is easy to check that

$$\frac{2e}{2k+1}(k+1) \geq e,$$

which, along with the condition $8et\|H_2\| \geq \varepsilon$, yields $M(k+1) \geq c_k$. This shows the assumptions of Lemma 5 are satisfied with this value of M.

From the recurrence relation, the number of required exponentials to implement S_{2k} in one subinterval is no more than $3 \cdot 5^{k-1}$. We need to consider two cases concerning $M\|H_1\|t$. If $M\|H_1\|t \geq 1$, then the number of subintervals is $\lceil M\|H_1\|t \rceil$, i.e., we partition the entire time interval into an integer number of subintervals, each of length at most M^{-1}. The total number of required exponentials is bounded by $3 \cdot 5^{k-1}\lceil M\|H_1\|t \rceil$. Substituting the values of M and c_k, we obtain the bound for N. In particular, we have

$$N \leq 3 \cdot 5^{k-1} \left\lceil \|H_1\|t \left(\frac{8et\|H_2\|}{\varepsilon}\right)^{1/(2k)} \frac{8e}{3} \left(\frac{5}{3}\right)^{k-1} \right\rceil. \tag{4.23}$$

If $M\|H_1\|t < 1$, then Lemma 5 can be used with $\Delta t = \|H_1\|t$, since $\|H_1\|t \leq M^{-1}$ and we have already seen that M is such that the assumptions of Lemma 5 are satisfied. Thus

$$\begin{aligned}
\left\|e^{-i(H_1+H_2)t} - S_{2k}(\mathcal{H}_1, \mathcal{H}_2, \|H_1\|t)\right\| &\leq \frac{4}{(2k+1)!}\|\mathcal{H}_2\|\,(c_k\|H_1\|t)^{2k+1} \\
&= 4t\|H_2\|\,\frac{c_k^{2k+1}}{(2k+1)!}(\|H_1\|t)^{2k} \\
&\leq 4t\|H_2\|\,\frac{c_k^{2k+1}}{(2k+1)!}(M)^{-2k} \\
&\leq \varepsilon,
\end{aligned}$$

where the last inequality holds by definition of M. In this case the total number of exponentials is simply

$$N \leq 3 \cdot 5^{k-1}. \tag{4.24}$$

Combining (4.23) and (4.24) we obtain

$$N \leq 3 \cdot 5^{k-1} \left\lceil \|H_1\| t \left(\frac{8et\|H_2\|}{\varepsilon} \right)^{1/(2k)} \frac{8e}{3} \left(\frac{5}{3} \right)^{k-1} \right\rceil .$$

This completes the proof. □

Lemma 5 and Theorem 4 indicate that when $\|H_2\|t \ll \varepsilon$, the number of exponentials N can be further improved. In this case it can be shown that high order splitting methods may lose their advantage. We do not pursue this direction in this chapter, since we assume that the H_j, $j = 1, \ldots, m$, are fixed and study N as $\varepsilon \to 0$.

4.3 Splitting methods for simulating the sum of many Hamiltonians

In this section we deal with the simulation of

$$e^{-i\sum_{j=1}^{m} H_j t},$$

where $H_1, \cdots, H_m$ are given non-commuting Hamiltonians. The analysis and the conclusions are similar to those of the previous section where $m = 2$, but the proofs are much more complicated and certainly tedious.

We use Suzuki's recursive construction once more [106]. In particular, for

$$S_2(H_1, \ldots, H_m, \Delta t) = \prod_{j=1}^{m} e^{-iH_j \Delta t/2} \prod_{j=m}^{1} e^{-iH_j \Delta t/2},$$

and

$$S_{2k}(H_1, \ldots, H_m, \Delta t) = [S_{2k-2}(p_k \Delta t)]^2 S_{2k-2}((1-4p_k)\Delta t)[S_{2k-2}(p_k \Delta t)]^2, \quad k = 2, 3, \ldots,$$

where for notational convenience we have used $S_{2k-2}(\Delta t)$ to denote $S_{2k-2}(H_1, \cdots, H_m, \Delta t)$, and $p_k = (4 - 4^{1/(2k-1)})^{-1}$, we have that

$$\left\| e^{-i\sum_{j=1}^{m} H_j \Delta t} - S_{2k}(H_1, \ldots, H_m, \Delta t) \right\| = O(|\Delta t|^{2k+1}). \tag{4.25}$$

Assuming again that $\|H_1\| \geq \|H_2\| \geq \cdots \geq \|H_m\|$ we normalize the Hamiltonians by setting $\mathcal{H}_j = H_j/\|H_1\|$, $j = 1, \ldots, m$, and consider the equivalent simulation problem

$$e^{-i\sum_{j=1}^{m} \mathcal{H}_j \tau},$$

where $\tau = \|H_1\|t$. Proceeding similarly to the previous section we derive the following lemma, whose proof can be found in the Appendix B.

Lemma 6. *For $k \in \mathbb{N}$, $d_k|\Delta t| \le k+1$, $d_k = m(4/3)k(5/3)^{k-1}$ and $\|\mathcal{H}_m\| \le \cdots \le \|\mathcal{H}_2\| \le \|\mathcal{H}_1\| = 1$, we have*

$$\|\exp(-i\sum_{j=1}^{m}\mathcal{H}_j\Delta t) - S_{2k}(\mathcal{H}_1,\ldots,\mathcal{H}_m,\Delta t)\| \le \frac{4\|\mathcal{H}_2\|}{(2k+1)!}(d_k|\Delta t|)^{2k+1}. \tag{4.26}$$

From Lemma 6, we have the following theorem.

Theorem 5. *Let $1 \ge \varepsilon > 0$ be such that $4met\|H_2\| \ge \varepsilon$. The number N of exponentials for the simulation of $e^{-i(H_1+\cdots+H_m)t}$ with accuracy ε is bounded by*

$$N \le (2m-1)\,5^{k-1}\left\lceil \|H_1\|t\left(\frac{4emt\|H_2\|}{\varepsilon}\right)^{1/(2k)}\frac{4me}{3}\left(\frac{5}{3}\right)^{k-1}\right\rceil,$$

for any $k \in \mathbb{N}$, where $\|H_m\| \le \cdots \le \|H_2\| \le \|H_1\|$.

Proof. The proof is similar to that of Theorem 4. Let $M = |\Delta t|^{-1}$. Then using Lemma 6 and $\mathcal{H}_j = H_j/\|H_1\|$, $j = 1,\ldots,m$, we obtain

$$\begin{aligned}\left\|e^{-i(H_1+\cdots+H_m)t} - S_{2k}^{M\|H_1\|t}(\mathcal{H}_1,\ldots,\mathcal{H}_m,1/M)\right\| &\le M\|H_1\|t\frac{4}{(2k+1)!}\|\mathcal{H}_2\|\left(\frac{d_k}{M}\right)^{2k+1}\\ &= 4t\|H_2\|\,\frac{d_k^{2k+1}}{(2k+1)!}\frac{1}{M^{2k}}.\end{aligned}$$

Recall that d_k is defined in Lemma 6. For accuracy ε we obtain

$$M \ge \left(\frac{4t\|H_2\|d_k^{2k+1}}{\varepsilon(2k+1)!}\right)^{1/(2k)}.$$

We use the estimate (4.22). It is easy to check that

$$d_k^{1/(2k)} \le 2m^{1/(2k)}.$$

Thus it suffices to take

$$M \ge \left(\frac{4emt\|H_2\|}{\varepsilon}\right)^{1/(2k)}\frac{2e\;d_k}{2k+1}.$$

So we define M to be the lower bound of the expression above, i.e.,

$$M := \left(\frac{4emt\|H_2\|}{\varepsilon}\right)^{1/(2k)}\frac{2e\;d_k}{2k+1}.$$

As in the proof of Theorem 4, it is straightforward to verify that $M(k+1) \geq d_k$. Therefore, the assumptions of Lemma 6 are satisfied for this value of M.

From the recurrence relation, we see that the number of required exponentials to implement S_{2k} in one subinterval is no more than $(2m-1) \cdot 5^{k-1}$. Again we distinguish two cases for $M\|H_1\|t$. We deal with the case $M\|H_1\|t < 1$ in the same way we did in the proof of Theorem 4, to conclude that

$$N \leq (2m-1) \cdot 5^{k-1}.$$

If $M\|H_1\|t \geq 1$, then the total number of required exponentials is

$$N \leq (2m-1) \cdot 5^{k-1} \lceil M\|H_1\|t \rceil.$$

Substituting the values of M and d_k, we obtain

$$N \leq (2m-1) \cdot 5^{k-1} \left\lceil \|H_1\|t \left(\frac{4emt\|H_2\|}{\varepsilon} \right)^{1/(2k)} \frac{4me}{3} \left(\frac{5}{3} \right)^{k-1} \right\rceil.$$

This completes the proof. □

The reader may wish to recall Remark 1 that applies in the case too.

Corollary 1. *If in addition to the assumptions of Theorem 5 either of the following two conditions holds:*

1. $4met\|H_1\| \geq 3$
2. *ε is sufficiently small such that*

$$\left(\ln \frac{4met\|H_1\|}{5} \right)^2 < 2 \ln \frac{5}{3} \ln \frac{4met\|H_2\|}{\varepsilon}$$

then the number of exponentials, N, for the simulation of $e^{-i(H_1+\cdots+H_m)t}$ with accuracy ε is bounded by

$$N \leq 2\,(2m-1)\,5^{k-1}\|H_1\|t \left(\frac{4emt\|H_2\|}{\varepsilon} \right)^{1/(2k)} \frac{4me}{3} \left(\frac{5}{3} \right)^{k-1}$$

for any $k \in \mathbb{N}$.

Proof. From the assumption of Theorem 5 we have $4emt\|H_2\|/\varepsilon \geq 1$. The argument of the ceiling function in the bound of Theorem 5 is greater than or equal to 1, if $4met\|H_1\| \geq 3$. Otherwise, we take its logarithm and multiply the resulting expression by k. This gives the quadratic polynomial

$$2k^2 \ln \frac{5}{3} + 2k \ln \frac{4met\|H_1\|}{5} + \ln \frac{4met\|H_2\|}{\varepsilon}.$$

When ε is sufficiently small and the discriminant is negative, i.e., when

$$\left(\ln \frac{4met\|H_1\|}{5}\right)^2 - 2 \ln \frac{5}{3} \ln \frac{4met\|H_2\|}{\varepsilon} < 0,$$

the polynomial is positive for all k. Hence, that argument of the ceiling function in the bound of Theorem 5 is greater than 1, for all $k \geq 1$.

In either case, we use $\lceil x \rceil \leq 2x$, for $x \geq 1$, to estimate N from above. □

4.4 Speedup

Let us now deal with the cost for simulating the evolution $e^{-i(\sum_{j=1}^{m} H_j)t}$. Berry et al. [17] show upper and lower bounds for the number of required exponentials. We concentrate on upper bounds and improve the estimates of [17].

We are interested in the number of exponentials required by the splitting formula that approximates the evolution with accuracy ε. From the results of the previous section we have that

$$N_{\text{new}} := 2\ (2m-1)\ 5^{k-1} \|H_1\| t \left(\frac{4emt\|H_2\|}{\varepsilon}\right)^{1/(2k)} \frac{4me}{3} \left(\frac{5}{3}\right)^{k-1}$$

exponentials suffice for error ε. The corresponding previously known estimate [17] is

$$N_{\text{prev}} = 2\ m\ 5^{2k}\ (m\|H_1\|t)^{1+1/(2k)} \left(\frac{1}{\epsilon}\right)^{1/(2k)},$$

where $H = \sum_{j=1}^{l} H_j$.

The ratio of the two estimates is

$$\frac{N_{\text{new}}}{N_{\text{prev}}} \leq \frac{1}{3^k} \left(\frac{4e\|H_2\|}{\|H_1\|}\right)^{1/(2k)}. \tag{4.27}$$

So for large k we have an improvement in the estimate of the cost of the algorithm. On the other hand, if $\|H_2\| \ll \|H_1\|$ we have an improvement in the estimate of the cost the algorithm, not just for large k but for all k. This is particularly significant when k is small. For instance, $k = 1$ for the Strang splitting S_2, which is frequently used in the literature.

Let us now consider the optimal k, i.e., the one minimizing N_{new}, for a given accuracy ε. It is obtaind from the solution of the equation

$$2k^2 \ln \frac{25}{3} - \ln \frac{4emt\|H_2\|}{\varepsilon} = 0.$$

Since we seek an integer k^*_{new} minimizing N_{new}, we set

$$k^*_{\text{new}} := \left\lceil \sqrt{\frac{1}{2} \log_{25/3} \frac{4emt\|H_2\|}{\varepsilon}} \right\rceil.$$

For k^*_{new} the number of exponentials N_{new} satisfies

$$N^*_{\text{new}} \leq \frac{8}{3} (2m-1)\, met\, \|H_1\|\, e^{2\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln\frac{4emt\|H_2\|}{\varepsilon}}}.$$

Berry et al. [17] find

$$k^*_{\text{prev}} = \text{round}\, \left(\frac{1}{2} \sqrt{\log_5 \frac{m\|H_1\|t}{\varepsilon} + 1} \right), \tag{4.28}$$

which minimizes N_{pre}. For k^*_{prev} the number of exponentials N_{prev} becomes

$$N^*_{\text{prev}} = 2m^2 \|H_1\| t\, e^{2\sqrt{\ln 5 \ln \frac{m\|H_1\|t}{\varepsilon}}}. \tag{4.29}$$

As a final comparison with N_{prev} we have

$$\frac{N^*_{\text{new}}}{N^*_{\text{prev}}} \leq \frac{8\,e}{3}\, e^{\,2\left(\sqrt{\frac{1}{2}\ln\frac{25}{3}\ln\frac{4emt\|H_2\|}{\varepsilon}} - \sqrt{\ln 5 \ln \frac{m\|H_1\|t}{\varepsilon}} \right)}.$$

Hence, there is an important difference between the previously derived optimal k and the one derived in the present chapter. In [17], the optimal k depends on $\|H_1\|$. More precisely, we show that the optimal k depends on $\|H_2\|$, the second largest norm of the Hamiltonians comprising H, which can be considerably smaller than $\|H_1\|$.

4.5 Discussion

Hamiltonian simulation plays an important role in scientific and engineering research. In this chapter, we have studied the splitting method, providing an improved upper bounds for the number of exponentials needed in the simulation of composite systems.

The advantage of our analysis is that we consider the influence of the second greatest Hamiltonian norm, $\|H_2\|$, on the computational complexity. In Chapter 6, we will use these results to design quantum algorithms that solve the ground state energy problem in multi-particle systems. Since the system is usually unbalanced, which means that the norm of one Hamiltonian is quite greater than that of the other, our conclusion improves the performance of the quantum algorithm significantly.

On the other hand, our study of Hamiltonian simulation focuses on deterministic splitting methods in this chapter. We also propose randomized algorithms for Hamiltonian simulation, which will be described in the next chapter.

Chapter 5

Randomized Hamiltonian Simulation

5.1 Introduction

Recall that, in a Hamiltonian simulation problem, the goal is to simulate the unitary operator e^{-iHt}, for some given time-independent Hamiltonian H and evolution time t. Often the accuracy ε of the simulation is measured by the trace distance [80] between the simulated final state and the desired final state. Various deterministic algorithms for this problem have been proposed as described in the previous chapter. However, as far as we know only deterministic algorithms have been considered.

In this chapter, we consider *randomized* algorithms for Hamiltonian simulation. By randomized we mean algorithms simulating $U = e^{-iHt}$ by $\hat{U}_\omega = \prod_{s=1}^{N_\omega} e^{-iH_{j_{s,\omega}} t_{s,\omega}}$ for random sequences of $j_{s,\omega}$ and intervals of $t_{s,\omega}$, occurring with probability p_ω. Consequently, the final state is approximated by a mixed quantum state.

This chapter is organized as follows. In Section 5.2, we provide an method to analyze the error bound of randomized algorithms in Hamiltonian simulation. Consider a randomized algorithm, where the unitary operator U is simulated by $\hat{U}_\omega$ with probability p_ω. Let the initial and final states for U be ρ_{init} and ρ_{final}, and the initial and final state of the simulation

process be $\tilde{\rho}_{\text{init}}$ and $\tilde{\rho}_{\text{final}}$, respectively. Let $D(\cdot)$ be the trace distance. Then

$$D(\rho_{\text{final}}, \tilde{\rho}_{\text{final}}) \leq D(\rho_{\text{init}}, \tilde{\rho}_{\text{init}}) + 2\|E(\hat{U}_\omega) - U\| + E(\|\hat{U}_\omega - U\|^2)$$

where $E(\cdot)$ denotes the expectation. In Section 5.3, we provide three randomized algorithms which are easier to implement than deterministic algorithms having the same accuracy. In Section 5.4, we prove that for positive evolution time, $t > 0$ any deterministic or randomized algorithm simulating e^{-iHt} with error ε must use $\Omega(t^{3/2}\varepsilon^{-1/2})$ exponentials.

5.2 The Randomized Model for Hamiltonian Simulation

In the randomized model, the sequence of unitary operators is selected randomly according to a certain probability distribution. The distribution can be realized either by "coin-flips" or by "control qubits". As a result, the algorithm is a product of a random sequence of unitary operators $\hat{U}_\omega = \prod_{s=1}^{N_\omega} e^{-iH_{j_{s,\omega}} t_{s,\omega}}$ selected with probability p_ω. Hence, for initial state $\rho_{\text{init}} = |\psi_0\rangle\langle\psi_0|$, the final state of the quantum algorithm is a mixed state $\sum_\omega p_\omega \hat{U}_\omega \rho_{\text{init}} \hat{U}_\omega^\dagger$. For more general cases, where the initial state of the simulation is not exactly ρ_{init}, but is a different mixed state $\tilde{\rho}_{\text{init}}$, the final state becomes

$$\tilde{\rho}_{\text{final}} = \sum_\omega p_\omega \hat{U}_\omega \tilde{\rho}_{\text{init}} \hat{U}_\omega^\dagger.$$

We now obtain an upper bound for the trace distance between the desired state and the one computed by a randomized algorithm.

Theorem 6. *Let U be the unitary operator being simulated by a set of random unitary operators $\{\hat{U}_\omega\}$ with probability distribution $\{p_\omega\}$. Assume the initial state for U is ρ_{init}, and the final state is $\rho_{\text{final}} = U\rho_{\text{init}}U^\dagger$. Let $\tilde{\rho}_{\text{init}}$ and $\tilde{\rho}_{\text{final}}$ respectively denote the initial and final states. Then the trace distance between ρ_{final} and $\tilde{\rho}_{\text{final}}$ is bounded from above by*

$$D(\rho_{\text{final}}, \tilde{\rho}_{\text{final}}) \leq D(\rho_{\text{init}}, \tilde{\rho}_{\text{init}}) + 2\|E(\hat{U}_\omega) - U\| + E(\|\hat{U}_\omega - U\|^2), \tag{5.1}$$

where $D(\cdot)$ denotes the trace distance, $E(\cdot)$ denotes the expectation, and $\|\cdot\|$ is the 2-norm.

Proof. First, we calculate the difference between ρ_{final} and $\tilde{\rho}_{\text{final}}$, which is

$$\begin{aligned}
\tilde{\rho_{\text{final}}} - \rho_{\text{final}} =& \sum_{\omega} p_\omega \hat{U}_\omega \tilde{\rho}_{\text{init}} \hat{U}_\omega^\dagger - U\rho_{\text{init}}U^\dagger \\
=& \sum_{\omega} p_\omega (U + \hat{U}_\omega - U)\tilde{\rho}_{\text{init}}(U + \hat{U}_\omega - U)^\dagger - U\rho_{\text{init}}U^\dagger \\
=& \sum_{\omega} p_\omega (\hat{U}_\omega - U)\tilde{\rho}_{\text{init}}U^\dagger + \sum_{\omega} p_\omega U\tilde{\rho}_{\text{init}}(\hat{U}_\omega - U)^\dagger \\
&+ \sum_{\omega} p_\omega (\hat{U}_\omega - U)\tilde{\rho}_{\text{init}}(\hat{U}_\omega - U)^\dagger + \sum_{\omega} p_\omega U\tilde{\rho}_{\text{init}}U^\dagger - U\rho_{\text{init}}U^\dagger \\
=& \left(\sum_{\omega} p_\omega \hat{U}_\omega - U\right)\tilde{\rho}_{\text{init}}U^\dagger + U\tilde{\rho}_{\text{init}}\left(\sum_{\omega} p_\omega \hat{U}_\omega - U\right)^\dagger \\
&+ \sum_{\omega} p_\omega (\hat{U}_\omega - U)\tilde{\rho}_{\text{init}}(\hat{U}_\omega - U)^\dagger + U(\tilde{\rho}_{\text{init}} - \rho_{\text{init}})U^\dagger.
\end{aligned}$$

Hence,

$$\begin{aligned}
D(\tilde{\rho}_{\text{final}}, \rho_{\text{final}}) =& \text{Tr}|\tilde{\rho}_{\text{final}} - \rho_{\text{final}}| \\
\leq& \text{Tr}|(\sum_{\omega} p_\omega \hat{U}_\omega - U)\tilde{\rho}_{\text{init}}U^\dagger| + \text{Tr}|U\tilde{\rho}_{\text{init}}(\sum_{\omega} p_\omega \hat{U}_\omega - U)^\dagger| \\
&+ \sum_{\omega} p_\omega \text{Tr}|(\hat{U}_\omega - U)\tilde{\rho}_{\text{init}}(\hat{U}_\omega - U)^\dagger| + \text{Tr}|\tilde{\rho}_{\text{init}} - \rho_{\text{init}}| \\
\leq& 2\left\|\sum_{\omega} p_\omega \hat{U}_\omega - U\right\| + \sum_{\omega} p_\omega \|\hat{U}_\omega - U\|^2 + D(\tilde{\rho}_{\text{init}}, \rho_{\text{init}}) \\
=& D(\tilde{\rho}_{\text{init}}, \rho_{\text{init}}) + 2\|E(\hat{U}_\omega) - U\| + E(\|\hat{U}_\omega - U\|^2).
\end{aligned}$$

□

Similarly to deterministic splitting algorithms, in a randomized algorithm the evolution time t is divided into K small intervals of size $\Delta t = t/K$, where K is decided later. The algorithm is comprised of K stages, and in each stage it approximates $e^{-iH\Delta t}$ by a product of unitary operators selected randomly according to a ceratin probability distribution. More precisely, in the k-th stage, the initial state is $\tilde{\rho}_{k-1}$, and the algorithm selects a product of exponentials randomly according to a certain probability distribution $\{p_\omega\}$ from $\{\hat{U}_\omega = \prod_{s=1}^{n_\omega} e^{-iH_{j_{s,\omega}}\Delta t_{s,\omega}}\}$. Then the final state of the k-th stage is

$$\tilde{\rho}_k = \sum_{\omega} p_\omega \hat{U}_\omega \tilde{\rho}_{k-1} \hat{U}_\omega^\dagger.$$

Obviously, $\tilde{\rho}_{\text{init}} = \rho_{\text{init}} = |\psi(0)\rangle\langle\psi(0)|$. The final state $\tilde{\rho}_K$ of the algorithm is used to approximate $\rho_{\text{final}} = |\psi(t)\rangle\langle\psi(t)|$. Then by choosing different unitary operator sets $\{\hat{U}_\omega\}$ and corresponding distributions $\{p_\omega\}$, we can provide different randomized algorithms with different efficiencies.

From Theorem 6, in each stage the increase of the trace distance is bounded from above by $\|E(\hat{U}_\omega) - U\|$ and $E(\|\hat{U}_\omega - U\|^2)$, modulo a constant. If both of these two terms are bounded from above by $O(\Delta t^{r+1})$ for some integer r, then the randomized algorithm yields a total error scaling as $O(t^{r+1}/K^r)$. Hence, the value of K required to achieve a given error bound ε scales as $O(t^{1+1/r}/\varepsilon^{1/r})$. When the number of exponentials in each stage can be considered constant, the number of exponentials equals

$$N = O(K) = O(t^{1+1/r}/\varepsilon^{1/r}).$$

We have the following corollary.

Corollary 2. *Consider a randomized algorithm for e^{-iHt}, where the evolution time t is divided into K small intervals of length Δt, and the evolution in each interval is simulated by the randomly selected $\hat{U}_\omega$ with probability p_ω. If*

$$\max\{\|E(\hat{U}_\omega) - e^{-iH\Delta t}\|, E(\|\hat{U}_\omega - e^{-iH\Delta t}\|^2)\} = O(\Delta t^{r+1})$$

for some integer r, then the total number of exponentials of the algorithm approximating e^{-iHt} is

$$O(t^{1+1/r}/\varepsilon^{1/r}).$$

5.3 Examples of Randomized Algorithms

In this section, we give several examples of randomized algorithms and use the lemma above to analyze their costs. The goal is to simulate the evolution of $H = \sum_{j=1}^m H_j$ for an evolution time t.

Algorithm 1:

1. *Divide the total evolution time t into K equal small segments of size Δt, where K will*

be defined later. The algorithm is comprised of K stages, and in the k-th stage, the initial state $\tilde{\rho}_{\text{init}}$ is denoted as $\tilde{\rho}_{k-1}$ and the final state $\tilde{\rho}_{\text{final}}$ is denoted as $\tilde{\rho}_k$, for $k = 1, \cdots, K$.

2. Let $\tilde{\rho}_0 = \rho_0 = |\psi_0\rangle\langle\psi_0|$ be the initial state of the first stage of the algorithm.

3. In the k-th stage of the algorithm, there are m substages. In the l-th substage, the initial state is $\tilde{\rho}_{k,l}$, and of course $\tilde{\rho}_{k,0} = \tilde{\rho}_{k-1}$.

3.1 In each substage, the algorithm chooses uniformly and independently at random an operator from $\{e^{-iH_1\Delta t}, \ldots, e^{-iH_m\Delta t}\}$, i.e., in the l-th substage, the operator would be $\hat{U}_{\omega_l} = e^{-iH_{\omega_l}\Delta t}$ with probability $p_{\omega_l} = 1/m$ for $\omega_l = 1, \cdots .m$. Taking into account all the alternatives, the final state of l-th substage in the k-th stage is

$$\tilde{\rho}_{k,l} = \sum_{j=1}^{m} \frac{1}{m} e^{-iH_j\Delta t} \tilde{\rho}_{k,l-1} e^{iH_j\Delta t}.$$

3.2 The final state of the k-th stage is $\tilde{\rho}_k = \tilde{\rho}_{k,m}$.

4. The final result of the algorithm is $\tilde{\rho}_K$, which is used to approximate the final quantum state.

In each stage of the algorithm, the operator $U_0 = e^{-iH\Delta t}$ is simulated by the product of random operators $\hat{U}_{\omega_l}$, for $l = 1, \cdots, m$ in m consecutive substages. Let $\tilde{U}_\omega = \prod_{l=1}^{m} \hat{U}_{\omega_l}$, then due to Theorem 6, the error of the algorithm in each stage is decided by two elements, $\|E(\tilde{U}_\omega) - U_0\|$ and $E(\|\tilde{U}_\omega - U_0\|^2)$. Since the selection of each operator is independent and uniform, we have

$$E(\tilde{U}_\omega) = \left(\frac{1}{m}\sum_{j=1}^{m} e^{-iH_j\Delta t}\right)^m = I - i\sum_{j=1}^{m} H_j\Delta t + O(\Delta t^2).$$

Hence,

$$\|E(\tilde{U}_\omega) - U_0\| = O(\Delta t^2).$$

Furthermore, for any ω, $\tilde{U}_\omega = I + O(\Delta t)$. Then

$$E(\|\tilde{U}_\omega - U_0\|^2) = O(\Delta t^2).$$

Thus the total error is $\varepsilon = O(K\Delta t^2)$ and the total number of exponentials used in the algorithm is

$$N = mK = O(t^2/\varepsilon).$$

We remark that this is equal modulo a constant to the cost of the deterministic algorithm that is based on a direct application of the Trotter formula $\prod_{j=1}^{m} e^{-iH_j\Delta t}$. However, **Algorithm 1** has a certain advantage over this deterministic algorithm. In each stage, the deterministic algorithm has a product of m exponentials in a precise sequence, hence it has to store the current index j of $e^{-iH_j\Delta t}$, for $j = 1, \cdots, m$. However, in **Algorithm 1**, the exponentials are random and independent of each other, hence the algorithm can be considered to be "memoryless".

Algorithm 2:

1. Divide the total evolution time t into K equal small segments of size Δt. The algorithm is comprised of K stages, and in the k-th stage, the initial state $\tilde{\rho}_{\text{init}}$ is denoted as $\tilde{\rho}_{k-1}$ and the final state $\tilde{\rho}_{\text{final}}$ is denoted as $\tilde{\rho}_k$, for $k = 1, \cdots, K$.

2. Let $\tilde{\rho}_0 = |\psi_0\rangle\langle\psi_0|$ be the initial state of the first stage of the algorithm.

3. In the k-th stage of the algorithm where the initial state is $\tilde{\rho}_{k-1}$, $k = 1, \ldots, K$. The algorithm selects a unitary operator uniformly and independently at random from $\{\hat{U}_1 = \prod_{j=1}^{m} e^{-iH_j\Delta t}, \hat{U}_2 = \prod_{j=m}^{1} e^{-iH_j\Delta t}\}$, i.e., in the k-th stage the operator would be $\hat{U}_\omega$ with probability $p_\omega = 1/2$, for $\omega = 1, 2$. Taking into account all the alternatives, the final state of the k-th stage is

$$\tilde{\rho}_k = \frac{1}{2}\left(\hat{U}_1\tilde{\rho}_{k-1}\hat{U}_1^\dagger + \hat{U}_2\tilde{\rho}_{k-1}\hat{U}_2^\dagger\right).$$

4. The final state of the algorithm is $\tilde{\rho}_K$ and is used to approximate the final quantum state.

In each stage, the algorithm simulates $U_0 = e^{-iH\Delta t}$ by $\hat{U}_1$ or $\hat{U}_2$ with equal probabil-

ity 1/2. Since

$$\begin{aligned}\hat{U}_1 =& \prod_{j=1}^{m}(I - iH_j\Delta t - \frac{1}{2}H_j^2\Delta t^2 + O(\Delta t^3)) \\ =& I - i\sum_{j=1}^{m} H_j\Delta t - \frac{1}{2}\sum_{j=1}^{m} H_j^2\Delta t^2 - \sum_{j_1<j_2} H_{j_1}H_{j_2}\Delta t^2 + O(\Delta t^3),\end{aligned}$$

and

$$\begin{aligned}\hat{U}_2 =& \prod_{j=m}^{1}(I - iH_j\Delta t - \frac{1}{2}H_j^2\Delta t^2 + O(\Delta t^3)) \\ =& I - i\sum_{j=1}^{m} H_j\Delta t - \frac{1}{2}\sum_{j=1}^{m} H_j^2\Delta t^2 - \sum_{j_1<j_2} H_{j_2}H_{j_1}\Delta t^2 + O(\Delta t^3),\end{aligned}$$

we see that $\|\hat{U}_\omega - U_0\| = O(\Delta t^2)$, for $\omega = 1, 2$, hence

$$E(\|\hat{U}_\omega - U_0\|^2) = O(\Delta t^4).$$

Moreover, since $E(\hat{U}_\omega) = I - iH\Delta t - \frac{1}{2}H^2\Delta t^2 + O(\Delta t^3)$, we have

$$\|E(\hat{U}_\omega) - U_0\| = O(\Delta t^3).$$

Due to Theorem 6, the error of this algorithm simulating $e^{-iH\Delta t}$ in each stage is $O(\Delta t^3)$. Hence, for a given accuracy ε, the total number of exponentials in **Algorithm 2** is

$$N = mK = O(t^{3/2}/\varepsilon^{1/2}).$$

We remark that this equals to the cost of a deterministic algorithm modulo a constant. The difference is that the deterministic algorithm is more complicated, since it is based on the Strang splitting formula

$$\hat{U} = \prod_{j=1}^{m} e^{-i\frac{1}{2}H_j\Delta t} \prod_{j=m}^{1} e^{-i\frac{1}{2}H_j\Delta t}.$$

As a result, in each stage, the deterministic algorithm has $2m - 1$ exponentials, but **Algorithm 2** only has m exponentials.

Next, we focus on the case that $m = 2$. In [105], the author provided the deterministic algorithm

$$\hat{U} = e^{-i\frac{1}{2}sH_1\Delta t}e^{-isH_2\Delta t}e^{-i\frac{1}{2}(1-s)H_1\Delta t}e^{-i(1-2s)H_2\Delta t}e^{-i\frac{1}{2}(1-s)H_1\Delta t}e^{-isH_2\Delta t}e^{-i\frac{1}{2}sH_1\Delta t}, \quad (5.2)$$

for simulating $e^{-iH\Delta t}$, where $s = 1/(2 - \sqrt[3]{2})$. The algorithm yields an error $O(\Delta t^4)$ in each stage. From Corollary 2, it has $O(t^{4/3}/\varepsilon^{1/3})$ exponentials. However, it requires irrational evolution times, which may be difficult to implement. We present a randomized algorithm with the same performance, but using fewer and simpler exponentials in each stage.

Algorithm 3:

1. Divide the total evolution time t into K equal small segments of size Δt. The algorithm is comprised of K stages, and in the k-th stage, the initial state $\tilde{\rho}_{\text{init}}$ is denoted as $\tilde{\rho}_{k-1}$ and the final state $\tilde{\rho}_{\text{final}}$ is denoted as $\tilde{\rho}_k$, for $k = 1, \cdots, K$.

2. Let $\tilde{\rho}_0 = |\psi_0\rangle\langle\psi_0|$ be the initial state of the first stage of the algorithm.

3. In the k-th stage of the algorithm where the initial state is $\tilde{\rho}_{k-1}$, $k = 1, \ldots, K$. The algorithm selects

$$\begin{aligned}
&\hat{U}_1 = e^{-i\frac{1}{2}H_1\Delta t}e^{-i\frac{1}{2}H_2\Delta t}e^{-i\frac{1}{2}H_1\Delta t}e^{-i\frac{1}{2}H_2\Delta t} && \text{with probability } p_1 = \frac{5}{12},\\
&\hat{U}_2 = e^{-i\frac{1}{2}H_2\Delta t}e^{-i\frac{1}{2}H_1\Delta t}e^{-i\frac{1}{2}H_2\Delta t}e^{-i\frac{1}{2}H_1\Delta t} && \text{with probability } p_2 = \frac{5}{12},\\
&\hat{U}_3 = e^{-i\frac{3}{2}H_1\Delta t}e^{i\frac{1}{2}H_2\Delta t}e^{i\frac{1}{2}H_1\Delta t}e^{-i\frac{3}{2}H_2\Delta t} && \text{with probability } p_3 = \frac{1}{12},\\
&\hat{U}_4 = e^{-i\frac{3}{2}H_2\Delta t}e^{i\frac{1}{2}H_1\Delta t}e^{i\frac{1}{2}H_2\Delta t}e^{-i\frac{3}{2}H_1\Delta t} && \text{with probability } p_4 = \frac{1}{12}.
\end{aligned}$$

i.e., the operator in the k-th stage is $\hat{U}_\omega$ with p_ω for $\omega = 1, 2, 3, 4$. Taking into account all the alternatives, the final state of stage k is

$$\tilde{\rho}_k = \sum_{\omega=1}^{4} p_\omega \hat{U}_\omega \tilde{\rho}_{k-1} \hat{U}_\omega^\dagger.$$

4. The final result of the algorithm is $\tilde{\rho}_K$ and is used to approximate the final quantum state.

In each stage, the algorithm simulates $U_0 = e^{-iH\Delta t}$ with U_ω by p_ω, for $\omega = 1, 2, 3, 4$. It

is easy to check, for any x,

$$\begin{aligned}
&e^{-ixH_1\Delta t}e^{-i(1-x)H_2\Delta t}e^{-i(1-x)H_1\Delta t}e^{-ixH_2\Delta t}\\
=&I-i(H_1+H_2)\Delta t-\left[\frac{1}{2}H_1^2+\frac{1}{2}H_2^2+x(2-x)H_1H_2+(1-x)^2H_2H_1\right]\Delta t^2\\
&+\{\frac{1}{6}H_1^3+\frac{1}{6}H_2^3+\left[x^2(\frac{3}{2}-x)+\frac{1}{2}x(1-x)^2\right]H_1^2H_2+\frac{1}{2}(1-x)^3H_2^2H_1\\
&+\left[\frac{1}{2}x(1-x)^2+x^2(\frac{3}{2}-x)\right]H_1H_2^2+\frac{1}{2}(1-x)^3H_2H_1^2\\
&+x(1-x)^2H_1H_2H_1+x(1-x)^2H_2H_1H_2\}\Delta t^3+O(\Delta t^4).
\end{aligned}\tag{5.3}$$

and

$$\begin{aligned}
&e^{-ixH_2\Delta t}e^{-i(1-x)H_1\Delta t}e^{-i(1-x)H_2\Delta t}e^{-ixH_1\Delta t}\\
=&I-i(H_1+H_2)\Delta t-\left[\frac{1}{2}H_1^2+\frac{1}{2}H_2^2+x(2-x)H_2H_1+(1-x)^2H_1H_2\right]\Delta t^2\\
&+\{\frac{1}{6}H_1^3+\frac{1}{6}H_2^3+\left[x^2(\frac{3}{2}-x)+\frac{1}{2}x(1-x)^2\right]H_2^2H_1+\frac{1}{2}(1-x)^3H_1^2H_2\\
&+\left[\frac{1}{2}x(1-x)^2+x^2(\frac{3}{2}-x)\right]H_2H_1^2+\frac{1}{2}(1-x)^3H_1H_2^2\\
&+x(1-x)^2H_2H_1H_2+x(1-x)^2H_1H_2H_1\}\Delta t^3+O(\Delta t^4).
\end{aligned}\tag{5.4}$$

Therefore, $\|\hat{U}_\omega-U_0\|=O(\Delta t^2)$, for $\omega=1,\cdots,4$, and

$$E(\|\hat{U}_\omega-U\|^2)=O(\Delta t^4).$$

Furthermore, from (5.3) and (5.4),

$$\begin{aligned}
E(\hat{U}_\omega)=&I-i(H_1+H_2)\Delta t-\frac{1}{2}(H_1+H_2)^2\Delta t^2\\
&+i[\frac{1}{2}(\frac{1}{2}-E(x(1-x)^2))(H_1H_2^2+H_2H_1^2+H_1^2H_2+H_2^2H_1)\\
&+E(x(1-x)^2)(H_1H_2H_1+H_2H_1H_2)]\Delta t^3+O(\Delta t^4),
\end{aligned}$$

where

$$E(x(1-x)^2)=2\times\frac{5}{12}\times\frac{1}{2}(1-\frac{1}{2})^2+2\times\frac{1}{12}\times\frac{3}{2}(1-\frac{3}{2})^2=\frac{1}{6}.$$

Hence,

$$\|E(\hat{U}_\omega)-U_0\|=O(\Delta t^4).$$

From Theorem 6, the error in each stage is bounded by $O(\Delta t^4)$, and the total number of exponentials used is $N=4K=O(t^{4/3}/\varepsilon^{1/3})$.

Note that **Algorithm 3** is not the only randomized algorithm that has $O(t^{4/3}/\varepsilon^{1/3})$ exponentials. In fact, if in each stage the algorithm selects

$$\hat{U}_\omega = e^{-ix_\omega H_1 \Delta t} e^{-i(1-x_\omega)H_2 \Delta t} e^{-i(1-x_\omega)H_1 \Delta t} e^{-ix_\omega H_2 \Delta t}$$

with probability $\frac{1}{2}p_\omega$ and

$$\hat{V}_\omega = e^{-ix_\omega H_2 \Delta t} e^{-i(1-x_\omega)H_1 \Delta t} e^{-i(1-x_\omega)H_2 \Delta t} e^{-ix_\omega H_1 \Delta t}$$

with the same probability $\frac{1}{2}p_\omega$, as long as $E(x_\omega(1-x_\omega)^2) = \sum_\omega p_\omega x_\omega (1-x_\omega)^2 = \frac{1}{6}$, then the algorithm also has $O(t^{4/3}/\varepsilon^{1/3})$ exponentials.

Compared to the deterministic algorithm that uses seven exponentials in each stage, **Algorithm 3** uses only four exponentials. Moreover, the exponentials in the deterministic algorithm have irrational factors in their evolution time, which certainly bring difficulties in implementation. However, all of the exponentials used in **Algorithm 3** have very simple factors in the evolution time. From [105], it is known that it requires at least six exponentials in each stage to simulate $e^{-iH\Delta t}$ within error bound $O(\Delta t^4)$ for deterministic algorithms. For the same efficiency, we have presented a randomized algorithm, i.e., **Algorithm 3**, which uses only four exponentials in each stage.

5.4 Lower Bounds for Positive Hamiltonian Simulation

Algorithms simulating e^{-iHt} by $\prod_{s=1}^{N} e^{-iH_{j_s}t_s}$ often require some of the t_s to be negative. Indeed, in splitting methods with order of convergence greater than or equal to three, some of the evolution times $\{t_s\}$ must be negative [106]. This may limit their applicability. For instance, such methods cannot be used for the simulation of diffusion operators, because there exists no inverse exponential diffusion operator, as noted by Suzuki [105].

In this section, we consider deterministic and *randomized* quantum algorithms simulating e^{-iHt} using only positive $\{t_s\}$. We call such a simulation a "*positive Hamiltonian simulation*". We provide a lower bound for the number of exponentials needed in positive Hamiltonian simulation as stated in the following theorem.

Theorem 7. *Any deterministic (randomized) algorithm simulating* e^{-iHt} *by* $\prod_{s=1}^{N} e^{-iH_{j_s}t_s}$, $t_s > 0$ ($\prod_{s=1}^{N} e^{-iH_{j_{s,\omega}}t_{s,\omega}}$, $t_{s,\omega} > 0$)*, with error* ε*, uses a number of exponentials bounded from below by* $\Omega(t^{3/2}\varepsilon^{-1/2})$.

Before proving the theorem, we start with the following lemma.

Lemma 7. *Let* $\sum_{i=1}^{N} x_i = 2$*, where* $0 \le x_i \le 1$ *for* $i = 1, \cdots, N$*. Let* S *be the sum of all elements in* $\{x_i x_j x_k : i < j < k, 2|k-i, 2 \nmid j-i\}$*. Then* $S < 1/3$.

Proof. For $N = 3$, it is easy to check $S \le (2/3)^3 < 1/3$. Assume that, for $N < M$, the conclusion holds.

For the case $N = M$, the global minimum of S will be achieved at a local minimum or the boundary. Moreover, if the global minimum is obtained at the boundary, it means some $x_i = 0$ or 1, and the problem reduces to the case $N < M$. Thus we only consider its local minimum and assume $x_i \neq 0$ for each i.

By using the Lagrange multiplier method, let $f(x) = S - \lambda(\sum_i^N x_i - 2)$. In any local minimum, $\partial f/\partial x_i = 0$, for each x_i. Then

$$\frac{\partial f}{\partial x_i} = \sum_{\substack{i<j<k \\ 2\nmid j-i \\ 2|k-i}} x_j x_k + \sum_{\substack{j<i<k \\ 2\nmid j-i \\ 2\nmid k-i}} x_j x_k + \sum_{\substack{j<k<i \\ 2|j-i \\ 2\nmid k-i}} x_j x_k - \lambda = 0,$$

and

$$\frac{\partial f}{\partial x_{i+2}} = \sum_{\substack{i+2<j<k \\ 2\nmid j-i \\ 2|k-i}} x_j x_k + \sum_{\substack{j<i+2<k \\ 2\nmid j-i \\ 2\nmid k-i}} x_j x_k + \sum_{\substack{j<k<i+2 \\ 2|j-i \\ 2\nmid k-i}} x_j x_k - \lambda = 0.$$

Subtracting these two equations, we have

$$x_{i+1}\Big(\sum_{\substack{k\ge i+2 \\ 2|k-i}} x_k + \sum_{\substack{k\le i-1 \\ 2\nmid k-i}} x_k - \sum_{\substack{k\ge i+3 \\ 2\nmid k-i}} x_k - \sum_{\substack{k\le i \\ 2|k-i}} x_k\Big) = 0.$$

From the assumption, $x_{i+1} \neq 0$, we have

$$\cdots + x_{i-3} + x_{i-1} + x_{i+2} + x_{i+4} + \cdots = \cdots + x_{i-2} + x_i + x_{i+3} + x_{i+5} + \cdots .$$

Then, we consider $\partial f/\partial x_{i+1} = 0$ and $\partial f/\partial x_{i+3} = 0$, which yields

$$\cdots + x_{i-1} + x_{i+1} + x_{i+4} + x_{i+5} + \cdots = \cdots + x_{i-2} + x_i + x_{i+3} + x_{i+5} + \cdots .$$

Combine them together, we have $x_{i+1} = x_{i+2}$. Therefore, $x_2 = x_3 = \cdots = x_{N-1}$. Then, by considering $\partial f/\partial x_1 = \partial f/\partial x_2$ and $\partial f/\partial x_N = \partial f/\partial x_{N-1}$, we have $x_1 = 0$ when N is even; and $x_1 = x_2 = \cdots = x_N$ when N is odd. Since the first case contradicts our assumption, we need only consider the case N is odd. Let $N = 2K + 1$, then each term in S is $(2/N)^3$, and there are $\frac{1}{6}K(K+1)(2K+1)$ terms. Therefore, the local minimum satisfies

$$S = \frac{1}{6}K(K+1)(2K+1)\left(\frac{2}{2K+1}\right)^3 = \frac{1}{3}\left(1 - \frac{1}{N^2}\right) < \frac{1}{3}.$$

Hence, for any N the conclusion holds. □

From the previous Lemma, we have the following theorem.

Theorem 8. *In each stage of deterministic or randomized algorithms, let $e^{-iH\Delta t}$ be simulated by a product of exponentials of H_j with positive evolution time, where $j = 1, \cdots, m$ and $H = \sum_{j=1}^{m} H_j$. Then, the trace distance in each stage increases no less than $\Omega(\Delta t^3)$.*

Proof. Since deterministic algorithms are special cases of randomized algorithms, it is enough to consider randomized algorithms. Assume $e^{-iH\Delta t}$ is simulated by U_ω with probability p_ω. Consider the evolutions of Hamiltonians H_1 and H_2 in U_ω. Let $\alpha_1\Delta t$, $\alpha_2\Delta t$, $\cdots$, $\alpha_K\Delta t$ be the overall evolution time of H_1 between two consecutive evolutions of H_2, while $\beta_1\Delta t, \beta_2\Delta t, \cdots, \beta_{K'}\Delta t$ are the overall evolution time of H_2 between two consecutive evolutions of H_1.

For example, if

$$U_\omega = e^{-iH_1\lambda_1\Delta t}e^{-iH_2\lambda_2\Delta t}e^{-iH_3\lambda_3\Delta t}e^{-iH_2\lambda_4\Delta t}e^{-iH_1\lambda_5\Delta t}e^{-iH_4\lambda_6\Delta t}e^{-iH_1\lambda_7\Delta t},$$

then $\alpha_1 = \lambda_1$, $\alpha_2 = \lambda_5 + \lambda_7$ and $\beta_1 = \lambda_2 + \lambda_4$.

Due to the alternation of the Hamiltonians in the algorithm, we have $|K - K'| \leq 1$. From Theorem 1, the difference of trace distance depends on $E(\|U_\omega - U_0\|^2)$ and $\|E(U_\omega) - U_0\|$. If $\sum_{j=1}^{K}\alpha_j \neq 1$ or $\sum_{j=1}^{K'}\beta_j \neq 1$ for some ω, $\|U_\omega - U_0\|^2 = \Omega(\Delta t^2)$, which follows by expanding the exponentials in the tower series. Thus, $E(\|U_\omega - U_0\|^2) = \Omega(\Delta t^2)$. Hence, we only need to consider the situation $\sum_{j=1}^{K}\alpha_j = 1$ and $\sum_{j=1}^{K'}\beta_j = 1$. Let us focus on the terms $iH_1H_2H_1\Delta t^3$ and $iH_2H_1H_2\Delta t^3$, in $e^{-iH\Delta t}$, each of which has coefficients 1/6. If the simulation has an error less than $O(\Delta t^3)$, their coefficients in $E(U_\omega)$ must also be 1/6,

therefore their sum should be 1/3. However, we will show in every U_ω, the sum of these two coefficients is less than 1/3. Without loss of generality, assume $K \geq K'$, let $x_{2j-1} = \alpha_j$, for $j = 1, \cdots, K$, and $x_{2j} = \beta_j$, for $j = 1, \cdots, K'$. Then the coefficient of $iH_1H_2H_1\Delta t^3$ is the sum of $x_j x_k x_l$, for $j < k < l$ where j, l are odd and k is even, while the coefficient of $iH_2H_1H_2\Delta t^3$ is the sum of $x_j x_k x_l$, for $j < k < l$ where j, l are even and k is odd. Lemma 7 implies that the sum of the coefficients is always less than 1/3, since $\sum_{j=1}^{K+K'} x_j = \sum_{j=1}^{K} \alpha_j + \sum_{j=1}^{K'} \beta_j = 2$. Since there always exists $\Theta(\Delta t^3)$ terms in any simulation, the error of any simulation is no less than $\Omega(\Delta t^3)$. □

We are ready to prove Theorem 7.

Proof of Theorem 7. Assume the simulation is comprised of K stages, and in the j-th stage, constant exponentials are used to simulate e^{-iHt_j}, where $\sum_{j=1}^{N} t_j = t$. From the previous theorem, the final error is $\Omega(\sum_{j=1}^{K} t_j^3)$, the minimum of which is $\Omega(t^3/K^2)$. Hence, to guarantee the final error is bounded by ε, we must have $K = \Omega(t^{3/2}\varepsilon^{-1/2})$. Therefore, $N = \Omega(K) = \Omega(t^{3/2}\varepsilon^{-1/2})$. □

It is straightforward to obtain the following two Corollaries.

Corollary 3. *The deterministic algorithm based on the Strang splitting is asymptotically optimal for strict Hamiltonian simulation.*

Corollary 4. *The randomized algorithm Algorithm 2 is asymptotically optimal for strict Hamiltonian simulation.*

5.5 Future Work

In this chapter, we have proposed a new model for Hamiltonian simulation. By selecting the local Hamiltonians randomly, we can design algorithms more easily, meanwhile maintaining the efficiency and accuracy.

For randomized Hamiltonian simulation, several questions remain open and should be addressed in future work. As we know, there exist deterministic algorithms based on Suzuki's decomposition that can simulate the evolution of a Hamiltonian with $O(t^{1+1/k}\varepsilon^{-1/k})$

exponentials, where k can be any small positive integer. However the implementation of this algorithm has two drawbacks: First, almost all of the factors in the algorithm are irrational. Second, there exists a huge constant term in the big-O, which is exponential in k. The huge constant arises from the number of exponentials used in each stage, which is $\Theta(5^k)$.

By giving **Algorithm 3**, we show that randomized algorithms may have the potential to overcome these two problems. **Algorithm 3** uses only rational factors to achieve $O(\Delta t^4)$ error bound in each stage, while all previous known deterministic algorithms must use irrational factors. Moreover, in each stage **Algorithm 3** uses only four exponentials, while deterministic algorithm with the same efficiency requires at least six exponentials. Hence, an interesting problem is to determine whether we can derive randomized algorithms, which achieve $O(\Delta t^k)$ error bound in each stage, use fewer exponentials in each stage, and keep all the factors rational, for any $k > 0$.

Chapter 6

Ground State Energy Estimation

6.1 Introduction

In this chapter, we study the problem of estimating the ground state energy of a time-independent Hamiltonian corresponding to a multiparticle system. The cost of solving such problems on a classical computer in the worst case setting is exponential in the number of particles. In particular, the number of state variables d is proportional to the number of particles and the cost to solve the problem with relative accuracy ε may grow as ε^{-d}. For these reasons researchers have been experimenting with quantum computers to solve eigenvalue problems in quantum chemistry with very encouraging results [38; 71]. See also [64; 65] and the references therein.

The eigenvalue problem considered in this chapter is called the time-independent Schrödinger equation in the physics literature and the Sturm-Liouville eigenvalue problem in the mathematics literature. It is described in Section 1.2.2. For convenience, we redefine the problem as follows. For a multiparticle system, if the potential is a function of only state variables then the ground state energy is given by the smallest eigenvalue E_1 of the equation

$$\begin{aligned} \left(-\tfrac{1}{2}\Delta + V\right)\Psi_1(x) &= E_1\Psi_1(x) \quad \text{for all } \ x \in I_d := (0,1)^d, \\ \Psi_1(x) &= 0 \quad \text{for all } \ x \in \partial I_d, \end{aligned}$$

where $\Delta = \sum_{j=1}^{d} \partial^2/\partial x_j^2$ is the Laplacian, ∂I_d denotes the boundary of the unit cube and Ψ_1 is a normalized eigenfunction. Moreover, V and its first order partial derivatives $\partial V/\partial x_j$,

for $j = 1, \cdots, d$, are continuous and uniformly bounded by 1.

We will exhibit a quantum algorithm that achieves relative error ε with cost $Cd\varepsilon^{-(3+\delta)}$, where $\delta > 0$ is an arbitrarily small positive number. The cost includes the number of queries plus all other quantum operations. The algorithm uses $C'd \log \varepsilon^{-1}$ qubits. The constants C and C', as well as all constants in our estimates throughout this chapter, are independent of d and ε.

The remainder of the chapter is organized as follows. In Section 6.2, we introduce the background knowledge about the ground state eigenvalue problem. In Section 6.3, we show how to transfer this problem from a continuous problem to a discrete problem, and the error introduced in the transfer. In Section 6.4, we provide our algorithm, and in Section 6.5, we analyze the cost of our algorithm.

6.2 Preliminary Analysis

The properties of the eigenvalues and eigenvectors of the problem are discussed extensively in [13; 32; 34; 46; 107], where it is shown that the eigenfunctions are continuous and have continuous partial derivatives of the first order, including the boundary of I_d.

The operator $\mathbb{L}_V = -\frac{1}{2}\Delta + V$ is symmetric with real eigenvalues and eigenvectors. The eigenvalues are positive, they can be indexed in nondecreasing order

$$0 < E_1(V) \leq E_2(V) \leq \cdots \leq E_k(V) \leq \cdots$$

and the sequence of eigenvalues tends to infinity. We denote the corresponding eigenvectors by $\Psi_k(\cdot; V)$, for $k = 1, 2, \cdots$.

The smallest eigenvalue $E_1(V)$ is simple, the corresponding eigenspace has dimension one, and the eigenvector $\Psi_1(\cdot; V)$ is uniquely determined up to its sign. It is convenient to assume that the $\Psi_k(\cdot; V)$ are normalized, i.e., that

$$\|\Psi_k(\cdot; V)\| := \left(\int_{I_d} \Psi_k(x, V)^2(x) dx \right)^{1/2} = 1,$$

for $k = 1, 2, \cdots$. Thus they form a complete orthonormal system. Then

$$\begin{aligned} E_1(V) &= \min_{\|\Psi\|=1} \int_{I_d} \sum_{j=1}^{d} \left(\frac{\partial \Psi}{\partial x_j}\right)^2 (x) + V(x)\Psi^2(x)dx \\ &= \int_{I_d} \sum_{j=1}^{d} \left(\frac{\partial}{\partial x_j}\Psi_1(x;V)\right)^2 + V(x)\Psi_1^2(x;V)dx \end{aligned} \tag{6.1}$$

For a constant potential $V(x) \equiv c$, we know that

$$E_1(c) = \frac{1}{2}d\pi^2 + c$$

and

$$\Psi_1(x_1, \cdots, x_d; c) = 2^{d/2} \prod_{j=1}^{d} \sin(\pi x_j).$$

It is also known that the eigenvalues of $\mathbb{L}_V$ are nondecreasing functions of V [32], i.e. if $V(x) \leq \bar{V}(x)$ for all $x \in [0,1]^d$, then $E_k(x;V) \leq E_k(x;\bar{V})$ for all x and k. Thus

$$\frac{1}{2}d\pi^2 = E_1(0) \leq E_1(V) \leq E_1(1) = \frac{1}{2}d\pi^2 + 1$$

For $d > 1$, the eigenvalues of $\mathbb{L}_V$ are generally not all simple. However, as in the case $d = 1$, the smallest eigenvalue $E_1(V)$ is simple and is well separated from the remaining eigenvalues. This is because of the nondecreasing property of the eigenvalues of $\mathbb{L}_V$ with respect to V and the fact that the second smallest eigenvalue of L_0 is equal to $E_2(0) = \frac{1}{2}(d+3)\pi^2$. Therefore, using $E_1(V) \leq \frac{1}{2}d\pi^2 + 1$, we obtain

$$E_k(V) - E_1(V) \geq E_2(V) - E_1(V) \geq \frac{3}{2}\pi^2 - 1, \tag{6.2}$$

for any k. From this fact, there is an estimate for the smallest eigenvalue by considering a perturbation of V.

Consider a potential function V and let $\bar{V}$ be a perturbation of V. Then the eigenvalue $E_1(V)$ and $E_1(\bar{V})$ are related according to the formula

$$E_1(V) = E_1(\bar{V}) + \int_{I_d} (V(x) - \bar{V}(x))\Psi_1^2(x;\bar{V})\, dx + O\big(\|V - \bar{V}\|_\infty^2\big).$$

This implies that approximating E_1 is at least as hard as approximating a multivariate integral in the worst case. As a result, any classical deterministic algorithm for the eigenvalue problem with accuracy ε must use a number of function evaluations of V that grows as ε^{-d}; see [83] for details. For convenience, in the remainder of this chapter we use E_1 to denote $E_1(V)$.

6.3 Discretization

Finite differences are often used for approximating E_1. The discretization of the operator $-\frac{1}{2}\Delta + V$ with mesh size $h = (m+1)^{-1}$ yields an $m^d \times m^d$ matrix $M_h := M_h(V) = -\frac{1}{2}\Delta_h + V_h$. Then one solves the corresponding matrix eigenvalue problem $M_h z_{h,1} = E_{h,1} z_{h,1}$. Note that Δ_h denotes the discretization of the Laplacian and V_h is a diagonal matrix whose entries are evaluations of the potential V at the m^d grid points. The reader may assume that Δ_h is obtained using a $2d+1$ stencil for the Laplacian; see, e.g., [72].

For instance, if $d = 2$, we have

$$-\Delta_h = h^{-2} \begin{pmatrix} T_h & -I & & & \\ -I & T_h & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T_h & -I \\ & & & -I & T_h \end{pmatrix},$$

which is an $m^2 \times m^2$ matrix, where I is the $m \times m$ identity matrix while

$$V_h = \begin{pmatrix} v_{11} & & & & \\ & \ddots & & & \\ & & v_{ij} & & \\ & & & \ddots & \\ & & & & v_{mm} \end{pmatrix},$$

where $v_{ij} = V(ih, jh)$, for $i, j = 1, \ldots, m$, and T_h is the $m \times m$ matrix given by

$$T_h = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}.$$

The matrix M_h is symmetric positive definite and sparse and has been extensively studied in the literature [34; 46; 72]. For a constant potential $v_{j,k} \equiv c$, the ground eigenvalue is

$$E_{h,1} = c + \frac{1}{2} \cdot 4dh^{-2} \sin^2(\pi h/2) = c + 2dh^{-2} \sin^2(\pi h/2)$$

and the ground eigenvector is $|\psi_1\rangle^{\otimes d}$, where the coordinates of $|\psi_1\rangle$ are

$$\psi_{1j} = \sqrt{2h}\sin(j\pi h), \quad j = 1, \ldots, m, \tag{6.3}$$

and $|\psi_1\rangle^{\otimes d}$ has unit length [34].

For V that has bounded first order partial derivatives, we use the results of [115; 116] to conclude that

$$|E_1 - E_{h,1}| \leq c_1 dh. \tag{6.4}$$

If $\hat{E}_{h,1}$ is such that $|E_{h,1} - \hat{E}_{h,1}| \leq c_2 dh$, we have relative error

$$|1 - \hat{E}_{h,1}/E_1| \leq c'h,$$

where c' is a constant. The inequality follows by observing that $2E_1$ is bounded from below by the smallest eigenvalue $4dh^{-2}\sin^2(\pi h/2)$ of the discretized Laplacian.

Such a discretization approach for a multiparticle system is not new; see e.g. [28]. The problem is that the size of the resulting matrix is exponential in d, and so is the cost of classical algorithms approximating its ground state eigenvalue.

6.4 Quantum algorithm

In this section, we describe our quantum algorithm in general terms. The key observation is that the discretization we outlined above and the estimation of the smallest eigenvalue of the resulting matrix can be implemented on a quantum computer with cost that does not grow exponentially with d. This is accomplished by modifying quantum phase estimation, a well known quantum algorithm for approximating an eigenvalue of a unitary matrix W, see e.g., [80]. First we provide the sketch of the algorithm.

Sketch of the algorithm

1. Consider the discretization $M_h = -\frac{1}{2}\Delta_h + V_h$ of $-\frac{1}{2}\Delta + V$ and let $h \leq \varepsilon$, where ε is the desired accuracy. The matrix

$$W = e^{iM_h/(2d)},$$

is unitary since M_h is Hermitian.

2. For W use phase estimation to approximate the phase corresponding to $e^{iE_{h,1}/(2d)}$ with the following modifications:

   (a) Use the approximate eigenvector

   $$|0\rangle^{\otimes b}|\psi_1\rangle^{\otimes d}$$

   as an initial state, where $|\psi_1\rangle^{\otimes d}$ is the ground state eigenvector of $-\Delta_h$.

   (b) Replace W^{2^t}, $t = 0, \ldots, b-1$, that are required in phase estimation, using approximations given by high order splitting formulas that deal with the exponentials of $-\frac{1}{2}\Delta_h$ and V_h separately,

Next we discuss the details of our algorithm. Assume h is the largest mesh size satisfying $h \leq \min(\varepsilon, 1/4)$. This leads to the desired accuracy while ensuring the discretization is not trivial. The eigenvalue of W that corresponds to $E_{h,1}$ is $e^{iE_{h,1}/(2d)} = e^{2\pi i\varphi_1}$, where

$$\varphi_1 = E_{h,1}/(4\pi d)$$

is the phase and belongs to the interval $[0, 1)$ since $E_{h,1} \leq 2dh^{-2}\sin^2(\pi h/2)+1 \leq d\pi^2/2+1$.

Quantum phase estimation approximates the phase φ_1 with b-bit accuracy, where $b = \lceil \log_2 h^{-1} \rceil$. The output of the algorithm is an index $j \in [0, 2^b-1]$ such that $|\varphi_1 - j\,2^{-b}| \leq 2^{-b}$. The algorithm approximates the ground state eigenvalue E_1 by

$$\hat{E}_{h,1} := 4\pi dj\,2^{-b}.$$

The initial state of our algorithm is

$$|0\rangle^{\otimes b}|\psi_1\rangle^{\otimes d}, \tag{6.5}$$

where $|\psi_1\rangle^{\otimes d}$ is the ground state eigenvector of the discretized Laplacian $-\Delta_h$ as described in (6.3). The quantum state $|\psi_1\rangle$ can be prepared by the quantum cosine transform [69].

The matrix exponentials W^{2^t} is approximated by a high-order splitting formula in Hamiltonian simulation. We simulate the evolution of a quantum system with Hamiltonian $H = M_h/(2d)$ for time 2^t, where $t = 0, 1, \ldots, b-1$. Let $H = H_1 + H_2$ where $H_1 = -\Delta_h/(4d)$ and $H_2 = V_h/(2d)$. Recall that h is the largest mesh size satisfying

$h \leq \min(\varepsilon, 1/4)$. The eigenvalues and eigenvectors of the discretized Laplacian are known and the evolution of a system with Hamiltonian H_1 can be implemented with $d \cdot O(\log^2 \varepsilon^{-1})$ quantum operations using the Fourier transform in each dimension. The evolution of a system with Hamiltonian H_2 can be implemented using two quantum queries and phase kickback. The details can be found in Section 1.2.2. The queries are similar to those in Grover's algorithm [80] and return function evaluations of V truncated to $O(\log \varepsilon^{-1})$ bits.

In particular, we use a splitting formula of order $2k + 1$, where $k \geq 1$, to approximate $W^{2^t} = e^{i(H_1+H_2)2^t}$ by

$$\prod_{\ell=1}^{N_t} e^{iH_{j_\ell} z_\ell}, \tag{6.6}$$

where $j_\ell \in \{1, 2\}$ and suitable z_ℓ that depend on t and k as described in chapter 4.

6.5 Cost of our Quantum Algorithm

In quantum phase estimation, since $b = \lceil \log_2 h^{-1} \rceil$, we have

$$|E_{h,1} - 4\pi dj\, 2^{-b}| \leq c_2 dh. \tag{6.7}$$

Combining (6.4)) and (6.7)) we conclude that

$$|E_1 - 4\pi dj\, 2^{-b}| \leq c_1 d\varepsilon + c_2 d\varepsilon = cd\varepsilon. \tag{6.8}$$

This estimate holds with probability at least $8/\pi^2$ [23] assuming that

- The initial state of the algorithm is $|0\rangle^{\otimes b}|z_{h,1}\rangle$, where $|z_{h,1}\rangle$ is the eigenvector of M_h that corresponds to $E_{h,1}$.
- We are given the matrix exponentials W^{2^t}, $t = 0, \ldots, b-1$.

In our case, however, we do not know $|z_{h,1}\rangle$ and we use an approximation $|\psi_1\rangle^{\otimes d}$. Similarly, we use approximations of the W^{2^t}, for $t = 0, \ldots, b-1$, to simulate the evolution of the quantum system that evolves with Hamiltonian $H = M_h/(2d)$. We will compute the cost to implement these approximations so that (6.8) holds. All these approximations affect the estimate $8/\pi^2$ of the success probability of phase estimation, but only by a small amount.

In our algorithm, the initial state $|0\rangle^{\otimes b}|\psi_1\rangle^{\otimes d}$ is prepared by quantum cosine transform. Since h is proportional to ε, the matrix M_h has size $m^d \times m^d$, with $m = \Theta(\varepsilon^{-1})$. Therefore, $|\psi_1\rangle^{\otimes d} \in C^{m^d}$ and can be represented using $\log_2 m^d = O(d \log_2 \varepsilon^{-1})$ qubits and can be implemented with $d \cdot O(\log^2 \varepsilon^{-1})$ quantum operations [69; 117]. We point out that here and elsewhere the implied constants in the big-O and Θ notation are independent of d and ε.

Expanding $|\psi_1\rangle^{\otimes d}$ using the eigenvectors of M_h we have

$$|\psi_1\rangle^{\otimes d} = \sum_{k=1}^{m^d} d_k |z_{h,k}\rangle.$$

The approximate initial state reduces the success probability of phase estimation by a factor equal to the square of the magnitude of the projection of $|\psi_1\rangle^{\otimes d}$ onto $|z_{h,1}\rangle$, to become $8|d_1|^2/\pi^2$; see, e.g., [2; 62].

We will see that $|d_1|^2 > \pi^2/10$. Indeed, we estimate $|d_1|$ using the approach in [119], which is based on the separation of the eigenvalues of M_h. In particular, we have

$$1 \geq (E_{h,2} - E_{h,1})^2 (1 - |d_1|^2),$$

where $E_{h,1}$ and $E_{h,2}$ are the smallest and second smallest eigenvalues of M_h. We estimate $E_{h,2} - E_{h,1}$ from below using the two smallest eigenvalues of $-\Delta_h$ to obtain

$$E_{h,2} - E_{h,1} \geq 2h^{-2}(\sin^2(\pi h) - \sin^2(\pi h/2)) - 1.$$

This yields that the success probability of phase estimation with the approximate ground state eigenvector is at least

$$\frac{8}{\pi^2}\left(1 - \frac{1}{(2h^{-2}(\sin^2(\pi h) - \sin^2(\pi h/2)) - 1)^2}\right) > \frac{4}{5}, \tag{6.9}$$

since $h \leq 1/4$.

Now let us turn to the cost of approximating the matrix exponentials by Hamiltonian simulation. From Chapter 4, the number of exponentials needed to approximate W^{2^t} by a splitting formula of order $2k+1$ with error ε_t, where $t = 0, \ldots, b-1$, is

$$N_t \leq 16e\|H_1\| 2^t \left(\frac{25}{3}\right)^{k-1} \left(\frac{8e\, 2^t \|H_2\|}{\varepsilon_t}\right)^{1/(2k)},$$

for any $k \geq 1$. The total number of exponentials required for the approximation of all the W^{2^t} is bounded from above by

$$\begin{aligned} N &= \sum_{t=0}^{b-1} N_t \leq 16e\|H_1\| \left(\frac{25}{3}\right)^{k-1} (8e\|H_2\|)^{1/(2k)} \\ &\qquad \times \sum_{t=0}^{b-1} 2^t \left(\frac{2^t}{\varepsilon_t}\right)^{1/(2k)} \\ &\leq 16e\|H_1\| 2^b \left(\frac{25}{3}\right)^{k-1} \left(160e\, 2^b \|H_2\|\right)^{1/(2k)}, \end{aligned} \tag{6.10}$$

where we obtained the last inequality by setting

$$\varepsilon_t = \tfrac{2^{t+1-b}}{40},$$

for $t = 0, \ldots, b-1$. It is easy to check that $\sum_{t=0}^{b-1} \varepsilon_t \leq 1/20$. Thus the success probability of phase estimation can be reduced by twice this amount [80]. Using (6.9) we conclude our algorithm succeeds with probability at least

$$\frac{8}{\pi^2}\left(1 - \frac{1}{(3\pi^2-2)^2}\right) - \frac{1}{10} \geq \frac{2}{3}.$$

Since $\|H_1\| \leq 4dh^{-2}/(4d) \leq \varepsilon^{-2}$ and $\|H_2\| \leq 1/(2d)$, the number N of exponentials of H_1 and H_2 that the algorithm uses satisfies

$$N \leq 16e \left(\frac{80e}{d}\right)^{1/(2k)} \left(\frac{25}{3}\right)^{k-1} \varepsilon^{-2}\, 2^{b(1+1/(2k))}.$$

Since we have chosen $2^b = \Theta(1/\varepsilon)$ we obtain

$$N \leq \widetilde{C} \left(\frac{80e}{d}\right)^{1/(2k)} \left(\frac{25}{3}\right)^{k-1} \varepsilon^{-(3+\frac{1}{2k})},$$

for any $k > 0$, where $\widetilde{C}$ is a constant.

The *optimal* k^*, i.e., the one minimizing the upper bound for N in (6.10), is obtained in Chapter 4 and is given by

$$k^* = \sqrt{\frac{1}{2}\log_{25/3}\frac{80e\, 2^b}{d}} = O\left(\sqrt{\ln\frac{1}{d\varepsilon}}\right) \quad \text{as } d\varepsilon \to 0,$$

The number of exponentials corresponding to k^* satisfies

$$N^* = O\left(\varepsilon^{-3} e^{\sqrt{\ln 1/d\varepsilon}}\right) \quad \text{as } d\varepsilon \to 0. \tag{6.11}$$

We remark that of the N^* matrix exponentials half involve H_1 and the other half involve H_2; see the detailed definition of the high order splitting formula in chapter 4. Since each exponential involving H_2 requires two queries the total number of queries is also $O\left(\varepsilon^{-3}e^{\sqrt{\ln 1/d\varepsilon}}\right)$.

Hence the number of quantum operations, excluding queries, to implement the initial state, the matrix exponentials involving H_1 and the inverse Fourier transform yielding the final state of phase estimation is

$$N^* \cdot O(d \log^2 \varepsilon^{-1}). \tag{6.12}$$

For the computational complexity, the depth of the quantum circuit realizing the algorithm grows as N^* which is given in (6.11). Clearly, $\varepsilon^{-3}e^{\sqrt{\ln 1/d\varepsilon}} \leq \varepsilon^{-3}e^{\sqrt{\ln 1/\varepsilon}}$, for any d. Thus N^* is bounded from above by a quantity independent of d. Recall that N^* is the total number of matrix exponentials the algorithm uses. Half of these exponentials involve the discretized Laplacian Δ_h and the other half involve the discretized potential V_h.

Each of the matrix exponentials involving Δ_h in d dimensions is implemented efficiently with cost proportional to $d \log^2 \varepsilon^{-1}$ using the quantum Fourier transform. Hence the cost of all matrix exponentials involving Δ_h is linear in d.

For the matrix exponentials involving V_h, each can be implemented with two quantum queries. The cost of each query is constant. Hence the cost of all matrix exponentials involving V_h is $2N^*$ times the cost of a quantum query.

Equations (6.10), (6.11) and (6.12) yield that the total cost of the algorithm, including the number of queries and the number of all other quantum operations, is

$$Cd\varepsilon^{-(3+\delta)},$$

where $\delta > 0$ is arbitrarily small and C is a constant.

Summarizing our results we see that both the number of qubits and the cost depend linearly on d. The algorithm uses phase estimation to approximate an eigenvalue of a matrix whose size is proportional to $\varepsilon^{-d} \times \varepsilon^{-d}$. The number of coordinates in the corresponding eigenvector is proportional to ε^{-d}, and is therefore represented using a number of qubits proportional to $d \log_2 \varepsilon^{-1}$.

Therefore, we obtain the following theorem.

Theorem 9. *Phase estimation with an approximate initial state and approximate powers of W with probability at least $\frac{2}{3}$ yields an estimate of E_1 with relative error ε and total cost*

$$Cd\,\varepsilon^{-(3+\delta)}$$

for any $\delta > 0$, using $C'd\log\varepsilon^{-1}$ qubits, where C and C' are constants.

This cost analysis reveals that the computational effort spent on solving the ground state eigenvalue problem unobscured by the cost of evaluating V (i.e., the the cost of a quantum query). It is not limited in any way, since for any particular choice of V when the actual cost of a query is known, it suffices to multiply it by the number of queries and add the product to (6.12) to obtain an aggregate cost estimate.

For multiparticle systems studied in physics and chemistry the number of dimensions d is directly proportional to the number of particles p. For instance, p particles in three dimensions yield $d = 3p$. Thus the number of qubits and the cost of the algorithm depends linearly on p. Nevertheless, we should point out that since $d \ll \varepsilon^{-1}$ in the interesting cases, precision plays the key role in the cost of approximating the ground state energy on a quantum computer.

6.6 Future Work

In this chapter, we have provided a quantum algorithm to estimate the ground state energy of a time-independent Hamiltonian corresponding to a multiparticle system. We assume that the potential V and its first order partial derivatives $\partial V/\partial x_j$, $j = 1, \cdots, d$, are continuous and uniformly bounded by 1. In our case we are able to efficiently obtain a rough but very useful approximation of the ground state eigenvector. Consequently, the cost of implementing and simulating the evolution of the Hamiltonian for the amount of time prescribed by the accuracy demand is polynomial in d and ε, which determines that we can approximate the ground state eigenvalue efficiently.

Observe that, an arbitrary potential ground state estimation problem is a QMA(Quantum Merlin Arthur)-complete problem [67], which is the quantum analog of the complexity class NP-complete problem or the probabilistic complexity class MA(Merlin Arthur)-complete problem [114].

On the other hand, there is still room for us to improve our algorithm. In the current version, the algorithm prepares the initial quantum state by $|0\rangle^{\otimes b}|\psi_1\rangle^{\otimes d}$, where $|\psi_1\rangle^{\otimes d}$ is the ground state eigenvector of $-\Delta_h$. It is possible to further reduce the cost of the initial state using the algorithm in [62] but we do not pursue this alternative in this chapter since the analysis of the algorithm becomes more involved. Another potential alternative is to prepare the initial state by adiabatic evolution introduced in Section 1.5. It is an interesting open problem to find the range of potentials in which quantum computers can substantially speedup the ground state eigenvalue problem relative to classical computers.

Moreover, in our algorithm the approximation of the matrix exponentials is simulated by high-order splitting methods. Quantum walk is an alternative way to simulate the evolution Hamiltonians, and in certain cases, the number of exponentials in the algorithm grows linearly with the Hamiltonian norm and the evolution time. It is an open question whether we can improve our algorithm by using the techniques of quantum walk.

Chapter 7

Conclusion and Future work

7.1 Conclusion

In this thesis, we studied five problems, including counting problem, quantum PAC learning, deterministic Hamiltonian simulation, randomized Hamiltonian simulation and ground state energy estimation.

Chapter 2 is about an application of adiabatic quantum algorithms to the counting problem. Adiabatic computation is a model of quantum computation different from the quantum circuit model. It applies adiabatic evolutions to find the solution of problems. We designed an adiabatic algorithm based on geometric phases to approximate the proportion of the marked items in a given database. The geometric phases depend only on the path of the evolution, not on how fast the path is traversed. Hence, the computation is robust to decoherence and resilient to certain kinds of noise. Since the counting problem is the foundation of many other numerical problems, such as high-dimensional integration and path integration, our adiabatic algorithms can be directly generalized to these kinds of problems.

Chapter 3 is about the query complexity of the quantum PAC learning model. For a concept class with d-VC dimension, we provide lower bound for query complexity, which is $\Omega(\varepsilon^{-1}(d^{1-\eta} + \log(1/\delta)))$, where ε is the required error bound, δ is the maximal failure possibility and η can be an arbitrarily small positive number. Since the best lower bound known in classical PAC learning model is $\Omega(\varepsilon^{-1}(d + \log(1/\delta)))$, we have almost optimally

solved this problem.

Chapter 4 is about Hamiltonian simulation, i.e., simulating the evolution of a quantum system under a given Hamiltonian. We focus on high-order splitting methods. For a given Hamiltonian H, where $H = \sum_{j=1}^{m} H_j$, we simulate the unitary operator e^{-iHt} by a product of exponentials of H_j. The efficiency of the simulation is measured by the number of exponentials. By providing an upper bound for the efficiency which depends explicitly on the second largest norm of the Hamiltonians comprising H, we significantly improved the previously known conclusions. We also point out that the optimal order of the splitting methods is decided by the second largest norm rather than the largest norm of the Hamiltonians.

In Chapter 5 we proposed the randomized model of Hamiltonian simulation. In a randomized algorithm, the evolution of Hamiltonian $H = \sum_{j=1}^{m} H_j$ is simulated by a product of exponentials of H_j in a random sequence and random evolution times. Hence the final state of the system is approximated by a mixed quantum state. In classical computation, the error is measured by different standards in deterministic algorithms and randomized algorithms. However, in Hamiltonian simulation the error is measured by the trace distance of the output quantum state and the desired state, in both deterministic algorithms and randomized algorithms. We provide an upper bound for the error in randomized algorithms, and then showed some randomized algorithms which have the same efficiency as certain deterministic algorithms but which are simpler to implement.

In Chapter 6 we showed an application of Hamiltonian simulation to the estimation of the ground state energy of a multiparticle system, which is also known as the multivariate Sturm-Liouville eigenvalue problem. The problem suffers from the curse of dimensionality in classical computation. We exhibited a quantum algorithm that achieves relative error ε using $O(d \log \varepsilon^{-1})$ qubits with total cost (number of quantum queries and other quantum operations) $O(d\, \varepsilon^{-(3+\delta)})$, where $\delta > 0$ is arbitrarily small. Thus, the number of qubits and the total cost are linear in the number of particles. The quantum algorithm is based on quantum phase estimation, and we use Hamiltonian simulation to implement the queries needed in phase estimation.

7.2 Future work

In Chapter 4, we studied splitting methods in Hamiltonian simulation and provided upper bounds for the number of exponentials. Our research focus is on the case where the Hamiltonian is time-independent. It would be an interesting research direction to generalize our results to the case where the Hamiltonian varies during the evolution time. In Chapter 4, we considered the problem of simulating the evolution of a Hamiltonian $H = \sum_{j=1}^{m} H_j$ with the exponentials of H_j, for $j = 1, \cdots, m$. But we didn't study how to decompose any given Hamiltonian H to the Hamiltonians which can be implemented efficiently. In [17], the author provide a method to decompose a sparse Hamiltonian to the sum of local Hamiltonians. However, for some Hamiltonians, it is not efficient to decompose it to local Hamiltonians. For instance, in Chapter 6, the Hamiltonian is decomposed as the sum of the Laplacian operator and a diagonal Hamiltonian, which improves the efficiency significantly compared to decomposing it to local Hamiltonians. Hence, an important problem in Hamiltonian simulation is to consider how to decompose a given Hamiltonian to make the simulation more efficient by applying the intrinsic characteristic of the Hamiltonian.

In Chapter 5, we proposed the model for randomized algorithms in Hamiltonian simulation. We provide some randomized algorithms to show the potential superiority of this new model. There are deterministic algorithms that can simulate the evolution of a Hamiltonian with $O(t^{1+1/k}\varepsilon^{-1/k})$ exponentials, where k can be any small positive integer. However the implementation of this algorithm has two drawbacks: First, almost all of the factors in the algorithm are irrational. Second, there exists a huge constant term in the big-O, which is exponential in k. The huge constant arises from the number of exponentials used in each stage, which is $\Theta(5^k)$. We provide a randomized algorithm, which uses only rational factors to achieve $O(\Delta t^4)$ error bound in each stage, while all previous known deterministic algorithms must use irrational factors. Moreover, in each stage the algorithm uses only fewer exponentials than any deterministic algorithm with the same efficiency. Hence, an interesting problem is to determine whether we can derive randomized algorithms, which achieve $O(\Delta t^k)$ error bound in each stage, use fewer exponentials in each stage, and keep all the factors rational, for any $k > 0$.

In Chapter 6, we provided a quantum algorithm to estimate the ground state energy of

a time-independent Hamiltonian corresponding to a multiparticle system. We assume that the potential V and its first order partial derivatives $\partial V/\partial x_j$, $j = 1, \cdots, d$, are continuous and uniformly bounded by 1. It is known that for certain potentials, the ground state energy estimation problem is a QMA(Quantum Merlin Arthur)-complete problem [67]. It is an interesting open problem to find the range of potentials for which quantum computers can substantially speedup the ground state energy problem relative to classical computers. Of particular interest is the Coulomb potential, which plays a key role in many scientific applications. Moreover, we consider the ground state energy for a multiparticle system where the masses of the particles are the same. A natural generalization is to estimate the ground state energy of the multiparticle system where the particles have different masses.

Bibliography

[1] M. Abramowitz and A. Stegun. *Handbook of Mathematical Functions.* Dover, New York, 1972.

[2] D. S. Abrams and S. Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.*, 83,5162, 1999.

[3] D. Aharonov, W. V. Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *Proceedings, 45th Annual Symposium on Fundamentals of Computer Science*, pages 42–51, 2004.

[4] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. *Proc. 35th Annual ACM Symp. on Theory of Computing*, pages 20–29, 2003.

[5] Y. Aharonov and J. Anandan. Phase change during a cyclic quantum evolution. *Phys. Rev. Lett.*, 58,1593, 1987.

[6] P. Aliferis and D. W. Leung. Computation by measurements: a unifying picture, 2004. http://arXiv.org/quant-ph/0404082.

[7] A. Ambainis. Quamtum walk algorithm for element distinctness. *Proceedings, 45th Annual Symposium on Fundamentals of Computer Science*, 2004.

[8] J. Anandan. The geometirc phase. *Nature*, 360:307–313, 1992.

[9] D. Angluin and M. Kharitonov. When wont membership queries help? *J. Comp. Syst. Sci.*, 50:336–355, 1995.

[10] S. Ashhab, J. R. Johansson, and F. Nori. Decoherence in a scalable adiabatic quantum computer. *Phys. Rev. A*, 74, 052330, 2006.

[11] A. Atici and R. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2007.

[12] A. Atici and R. A. Servedio. Improved bounds on quantum learning algorithms. *Quantum Information Processing*, 4(5), 2005.

[13] I. Babuska and J. Osborn. *Eigenvalue Problems*, volume II, pages 641–787. in Handbook of Numerical Analysis, P. G. Ciarlet and J. L. Lions, eds., North-Holland, Amsterdam, 1991.

[14] H. Barnum, E. Knill, G. Ortiz, R. Somma, and L. Viola. Quantum simulations of physics problems, 2003. http://arXiv.org/quant-ph/0304063.

[15] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Proc. 39th IEEE Symp. on Foundations of Computer Science*, pages 352–361, 1998.

[16] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997.

[17] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, 2000.

[18] M. V. Berry. Quantal phase factors accompanying adiabatic changes. *Proc. R. Soc. Lond. A*, 392:45–57, 1984.

[19] A. J. Bessen. A lower bound for phase estimation on a quantum computer. *Physical Review A*, 71(4),042313, 2005.

[20] A. Blais and A. M. S. Tremblay. Effect of noise on geometric logic gates for quantum computation. *Phys. Rev. A*, 67,012308, 2003.

[21] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

[22] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp. Quantum counting, 1998. http://arXiv.org/arXiv:quant-ph/9805082.

[23] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

[24] N. H. Bshouty and J. C. Jackson. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM J. Comput.*, 28(3):1136–1153, 1999.

[25] R. E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica. 7. Cambridge University Press*, pages 1–49, 1998.

[26] A. M. Childs, E. Farhi, and J. Preskill. Robustness of adiabatic quantum computation. *Phys. Rev. A*, 65, 012322, 2001.

[27] A. M. Childs and R. Kothari. Simulating sparse hamiltonians with star decompositions, 2010. http://arXiv.org/quant-ph/1003.3683.

[28] P. G. Ciarlet and C. Le Bris. *Handbook of Numerical Analysis, Special Volume Computational Chemistry*, volume X. North Holland, Amsterdam, 2003.

[29] R. Cole and U. Vishkin. Deterministic coin tossing with applications to optimal parallel list ranking. *Inform. and Control*, 70:32–53, 1986.

[30] L. Collatz. *The Numerical Treatment of Differential Equations.* Springer-Verlag, Berlin, 1960.

[31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* The MIT Press, 2001.

[32] C. Courant and D. Hilbert. *Methods of Mathematical Physics, Vol. I.* Wiley Classics Library, Willey-Interscience, New York, 1989.

[33] V. Danos, E. Kashefi, and P. Panangaden. The measurement calculus. *J. ACM*, 54(2), 2007.

[34] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[35] D. Deutsch. Quantum computational networks. *Proc. R. Soc. Lond. A*, 425:73, 1989.

[36] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc. R. Soc. Lond. A*, 439:553, 1992.

[37] D. P. DiVincenzo. Quantum gates and circuits. *Proc. R. Soc. London A*, 454:261–276, 1998.

[38] J. Du, N. Xu, X. Peng, P. Wang, S. Wu, and D. Lu. NMR implementation of a molecular hydrogen quantum simulation with adiabatic state preparation. *Phys. Rev. Lett.*, 104,030502, 2010.

[39] L. M. Duan, J. I. Cirac, and P. Zoller. Geometric manipulation of trapped ions for quantum computation. *Science*, 292,1695, 2001.

[40] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for leanring. *Information and Computation*, 82(3):247–251, 1989.

[41] E. Farhi, J. Goldstone, and S. Cutmann. A quantum algorithm for the hamiltonian NAND tree. *Theory of Computing*, 4, 2008.

[42] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution, 2000. http://arXiv.org/quant-ph/0001106.

[43] E. Farhi and S. Gutmann. Analog analogue of a digital quantum computation. *Phys. Rev. A*, 57, 2403, 1998.

[44] R. P. Feynman. Simulating physics with computers. *Int. J. Theoret. Phys*, 21:467–488, 1982.

[45] G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer, New York, 1995.

[46] G. E. Forsythe and W. R. Wasow. *Finite-Di?erence Methods for Partial Differential Equations*. Dover, New York, 2004.

[47] K. Fujikawa. Topological properties of Berry's phase. *Mod. Phys. Lett.*, A20:335–344, 2005.

[48] O. Goldreich. *Foundations of Cryptography.* Cambridge University Press, 2001.

[49] S. Gortler and R. A. Servedio. Quantum versus classical learnability. *6th Conference on Computational Complexity*, pages 138–148, 2001.

[50] S. Gortler and R. A. Servedio. Equivalences and separations between quantum and classical learnability. *SIAM J. Comput.*, 33(5), 2004.

[51] L. Grover. A fast quantum mechanical algorithm for database search. *28th ACM Symposium on Theory of Computing*, 1996.

[52] L. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79(2):325–328, 1997.

[53] S. Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. *J. ACM*, 54(1), 2007.

[54] S. Heinrich. Quantum summation with an application to integration. *J. Complexity*, 18(1):1–50, 2002.

[55] S. Heinrich. From monte carlo to quantum computation. *Proceedings of the 3rd IMACS Seminar on Monte Carlo Methods MCM2001, Salzburg*, 62:219–230, 2003.

[56] S. Heinrich. The quantum query complexity of elliptic PDE. *J. Complexity*, 22(5):691–725, 2006.

[57] S. Heinrich. The randomized complexity of elliptic PDE. *J. Complexity*, 22(2):220–249, 2006.

[58] S. Heinrich and E. Novak. Optimal summation and integration by deterministic, randomized, and quantum algorithms. *In 4th international conference on Monte Carlo and Quasi-Monte Carlo Methods, Hong Kong*, 2000.

[59] S. Heinrich and E. Novak. *Optimal summation by deterministic, randomized and quantum algorithms.* in Monte Carlo and Quasi-Monte Carlo Methods, 2000, K.-T. Fang, F. J. Hickernell and H. Niederreiter, eds, Springer-Verlag, Berlin, 2001.

[60] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[61] J. Hou. Effect of classical noise on the geometric quantum phase. *Phys. Rev. A*, 75,024103, 2007.

[62] P. Jaksch and A. Papageorgiou. Eigenvector approximation leading to exponential speedup of quantum eigenvalue estimation. *Phys. Rev. Lett.*, 91,257902, 2003.

[63] J. A. Jones, V. Vedral, A. Ekert, and G. Castagnoli. Geometric quantum computation using nuclear magnetic resonance. *Nature*, 403,869, 2000.

[64] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik. Polynomial-time quantum algorithm for the simulation of chemical dynamics. *Proc. Natl. Acad. Sci.*, 105,18681, 2008.

[65] I. Kassal, J. D. Witeld, A. Perdomo-Ortiz, M. H. Yung, and A. Aspuru-Guzik. Simulating chemistry using quantum computers. *Annu. Rev. Phys. Chem.*, 62:185, 2011.

[66] M. J. Kearns and U. V. Vazirani. *An introduction to computational learning theory.* The MIT Press, 1994.

[67] J. Kempe, A. Kitaev, and O. Regev. The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35 (5):1070C1097, 2010.

[68] A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation.* American Mathematical Society, 2002.

[69] A. Klappenecker and M. Rötteler. Discrete cosine transforms on quantum computers, 2001. http://arXiv.org/quant-ph/0111038.

[70] B. Klar. Bounds on tail probabilities of discrete distributions. *Probability in the Engineering and Informational Science*, 14:161–171, 2000.

[71] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik4, and A. G. White1. Towards quantum chemistry on a quantum computer. *Nature Chemistry*, 2,106, 2010.

[72] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM, Philadelphia, PA, 2007.

[73] D. A. Lidar. Towards fault tolerant adiabatic quantum computation. *Phys. Rev. Lett.*, 100, 160506, 2008.

[74] J. H. Van Lint. *Introduction to Coding Theory*. Springer-Verlag, New York, 1992.

[75] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York, 2001.

[76] S. Lloyd. Universal quantum simulators. *Science*, 273:1073–1078, 1996.

[77] M. Mitzenmacher and E. Upfal. *Probability and Computing: randomized algorithms and probabilistic analysis*. Cambridge University Press, Cambridge, UK, 2005.

[78] A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. *31st ACM Symposium on Theory of Computing*, pages 384–393, 1999.

[79] A. Nazir, T. P. Spiller, and W. J. Munro. Decoherence of geometric phase gates. *Phys. Rev. A*, 65,042303, 2002.

[80] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.

[81] E. Novak. Deterministic and stochastic error bounds in numerical analysis. *Lecture Notes in Mathematics, Springer Verlag*, 1349, 1988.

[82] A. Papageorgiou. Average case quantum lower bounds for computing the boolean mean. *J. Complexity*, 20(5):713–731, 2004.

[83] A. Papageorgiou. On the complexity of the multivariate sturm-liouville eigenvalue problem. *J. Complexity*, 23:802–827, 2007.

[84] A. Papageorgiou, I. Petras, J. F. Traub, and C. Zhang. A fast algorithm for approximating the ground state energy on a quantum computer, 2010. http://arXiv.org/quant-ph/1008.4294.

[85] A. Papageorgiou and J. F. Traub. *Quantum Algorithms and Complexity for Continuous Problems*. In Encyclopedia of Complexity and Systems Science, Springer, New York, 2009.

[86] A. Papageorgiou and H. Woźniakowski. Classical and quantum complexity of the sturm-liouville eigenvalue problem. *Quantum Information Processing*, 4(3):87–127, 2005.

[87] A. Papageorgiou and H. Woźniakowski. The sturm-liouville eigenvalue problem and np-complete problems in the quantum setting with queries. *Quantum Information Proceeding*, 6(2):101–120, 2007.

[88] A. Papageorgiou and C. Zhang. On the efficiency of quantum algorithms for hamiltonian simulation, 2010. http://arXiv.org/quant-ph/1005.1318.

[89] B. Paredes, F. Verstraete, and J. I. Cirac. Exploiting quantum parallelism to simulate quantum random many-body systems. *Phys. Rev. Lett.*, 95,140501, 2005.

[90] R. Raussendorf, D. E. Browne, and H. J. Briegel. Measurementbased quantum computation on cluster states. *Phys. Rev. A*, 68,022312, 2003.

[91] B. W. Reichardt and R. Spalek. Span-program-based quantum algorithm for evaluating formulas. *40th ACM Symposium on Theory of Computing*, 2008.

[92] J. Roland and N. J. Cerf. Quantum search by local adiabatic evolution. *Phys. Rev. A*, 65, 042308, 2002.

[93] J. Roland and N. J. Cerf. Noise resistance of adiabatic quantum computation using random matrix theory. *Phys. Rev. A*, 71, 032330, 2005.

[94] J. Samuel and R. Bhandari. General setting for Berry's phase. *Phys. Rev. Lett.*, 60, 2339, 1988.

[95] F. Santha, Nayak A, J. Roland, and M. Santha. Search via quantum walk. *39th ACM Symposium on Theory of Computing*, 2007.

[96] M. S. Sarandy and D. A. Lidar. Adiabatic quantum computation in open systems. *Phys. Rev. Lett.*, 95, 250503, 2005.

[97] L. I. Schiff. *Quantum Mechanics*. McGraw-Hill, New York.

[98] R. A. Servedio. Separating quantum and classical learning. *28th International Conference on Automata, Languages and Programming (ICALP)*, pages 1065–1080, 2001.

[99] A. Shapere and F. Wilczek. Geometric phases in physics. *World Scientific, Singapore*, 1989.

[100] Y. Shi. Lower bounds of quantum black-box complexity and degree of approximating polynomials by influence of boolean variables. *Information Processing Letters*, 75:79–83, 2000.

[101] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings, 35th Annual Symposium on Fundamentals of Computer Science*, 1994.

[102] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithm on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[103] B. Simon. Holonomy, the quantum adiabatic theorem, and Berry's phase. *Phys. Rev. Lett.*, 51,2167, 1983.

[104] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Num. Analysis*, pages 506–517, 1968.

[105] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. *Phys. Lett. A*, 146:319–323, 1990.

[106] M. Suzuki. General theory of fractal path integrals with application to many-body theories and statistical physics. *J. Math. Phys*, 32:400–407, 1991.

[107] E. C. Titschmarsh. *Eigenfunction Expansions Associated with Second-Order Differential Equations*. Oxford University Press, Oxford, UK, 1958.

[108] J. F. Traub. A continuous model of computation. *Physics Today*, pages 39–43, May 1998.

[109] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski. *Information-Based Complexity*. Academic Press, 1988.

[110] J. F. Traub and A. Werschulz. *Complexity and Information*. Cambridge University Press, Cambridge, UK, 1998.

[111] J .F. Traub and H. Woźniakowski. Path integration on a quantum computer. *Quantum Information Processing*, 1(5):365–388, 2002.

[112] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[113] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[114] J. Watrous. Quantum computational complexity, 2008. http://arXiv.org/quant-ph/0804.3401.

[115] H. F. Weinberger. Upper and lower bounds for eigenvalues by finite difference methods. *Comm. Pure Appl. Math*, 1956.

[116] H. F. Weinberger. Lower bounds for higher eigenvalues by finite difference methods. *Pacific J. Math.*, 8(2):339–368, 1958.

[117] M. V. Wickerhauser. *Adapted wavelet analysis from theory to software*. Wellesley, A.K. Peters eds, 1994.

[118] N. Wiebe, D. Berry, P. Hoyer, and B. C. Sanders. Higher order decompositions of ordered operator exponentials. *J. Phys. A: Math. Theor.*, 43,065203, 2010.

[119] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford, UK, 1965.

[120] H. Woźniakowski. The quantum setting with randomized queries for continuous problems. *Quantum Information Proceeding*, 5(2):83–130, 2006.

[121] A. C. Yao. Quantum circuit complexity. *Proceedings, 34th Annual Symposium on Fundamentals of Computer Science*, pages 352–361, 1993.

[122] X. Yi, L. Wang, and T. Zheng. Berry phase in a composite system. *Phys. Rev. Lett.*, 92:150406, 2004.

[123] C. Zalka. Efficient simulation of quantum systems by quantum computers. *Fortschritte der Physik*, 46, 887, 1998.

[124] C. Zalka. Simulating quantum systems on a quantum computer. *Proc. R. Soc. Lond. A*, 454:313–323, 1998.

[125] C. Zhang. Randomized algorithms and lower bounds for quantum simulation, 2009. http://arXiv.org/quant-ph/0910.4145.

[126] C. Zhang. An improved lower bound on query complexity for quantum PAC learning. *Information Processing Letters*, 111 (1), 2010.

[127] C. Zhang, Z. Wei, and A. Papageorgiou. Adiabatic quantum counting by geometric phase estimation. *Quantum Information Proceeding*, 9 (3), 2010.

[128] S. Zhu and Z. D. Wang. Implementation of universal quantum gates based on nonadiabatic geometric phases. *Phys. Rev. Lett.*, 89,097902, 2002.

Appendix A

Appendix of Adiabatic Counting

In this Appendix, we show the details of how to derive the quantum state under the Hamiltonian $H(\theta) = H(\omega t)$. The quantum system is initially in a state $|\psi\rangle = \left(\sqrt{\beta}, \sqrt{\alpha}\right)$, and then evolves under the Hamiltonian

$$H(\theta) = \begin{pmatrix} \alpha & -\sqrt{\alpha\beta}e^{i\theta} \\ -\sqrt{\alpha\beta}e^{-i\theta} & \beta \end{pmatrix}. \tag{A.1}$$

Let $|\varphi(t)\rangle$ be the t-time quantum state in the system. Then the Schrodinger equation (1.2) turns out to be,

$$i\frac{d}{dt}|\varphi\rangle = H|\varphi\rangle = \begin{pmatrix} \alpha & -\sqrt{\alpha\beta}e^{i\omega t} \\ -\sqrt{\alpha\beta}e^{-i\omega t} & \beta \end{pmatrix} |\varphi\rangle. \tag{A.2}$$

Let $|\varphi\rangle = \begin{pmatrix} x \\ y \end{pmatrix}$, then

$$i\frac{d}{dt}x = \alpha x - \sqrt{\alpha\beta}e^{i\omega t}y, \tag{A.3}$$

$$i\frac{d}{dt}y = -\sqrt{\alpha\beta}e^{-i\omega t}x + \beta y. \tag{A.4}$$

From (A.3), we have

$$\begin{aligned} y &= \sqrt{\frac{\alpha}{\beta}}e^{-i\omega t}x - i\frac{1}{\sqrt{\alpha\beta}}e^{-i\omega t}\frac{dx}{dt}, \\ \frac{dy}{dt} &= \frac{1}{\sqrt{\alpha\beta}}e^{-i\omega t}\left(-i\alpha\omega x + (\alpha-\omega)\frac{dx}{dt} - i\frac{d^2x}{dt^2}\right). \end{aligned} \tag{A.5}$$

Substituting (A.5) into (A.4), we obtain

$$\frac{d^2x}{dt^2} + i(1-\omega)\frac{dx}{dt} + \alpha\omega x = 0. \tag{A.6}$$

Similarly, we can derive

$$\frac{d^2y}{dt^2} + i(1+\omega)\frac{dy}{dt} - \beta\omega y = 0. \tag{A.7}$$

The roots of (A.6) and (A.7) are $-\frac{1}{2}i \pm i\omega_{1,2}$ separately, where

$$\omega_{1,2} = \frac{\omega \pm \sqrt{(1-\omega)^2 + 4\alpha\omega}}{2}. \tag{A.8}$$

Then

$$\begin{aligned} x &= e^{-i\frac{1}{2}t}(Ae^{i\omega_1 t} + Be^{i\omega_2 t}), \\ y &= e^{-i\frac{1}{2}t}(Ce^{-i\omega_1 t} + De^{-i\omega_2 t}). \end{aligned} \tag{A.9}$$

From the initial state $|\psi\rangle = \sqrt{\beta}|\hat{0}\rangle + \sqrt{\alpha}|\hat{1}\rangle$, we have

$$A + B = \sqrt{\beta}; C + D = \sqrt{\alpha}. \tag{A.10}$$

Substituting Eq.(A.9) into Eq.(A.3), it can be derived that

$$\frac{D}{A} = -\frac{B}{C} = \lambda = \frac{2\sqrt{\alpha\beta}}{(\beta - \alpha - \omega) + \sqrt{(1-\omega)^2 + 4\alpha\omega}}. \tag{A.11}$$

Then we attain that

$$\begin{aligned} A &= \frac{\lambda\sqrt{\alpha} + \sqrt{\beta}}{1+\lambda^2} = \frac{(1-\omega)^2 - \alpha(1-3\omega) + (\beta-\omega)E}{(1-\omega)^2 + 4\alpha\omega + (\beta-\alpha-\omega)E}\sqrt{\beta}, \\ B &= \lambda\frac{\lambda\sqrt{\beta} - \sqrt{\alpha}}{1+\lambda^2} = \frac{\alpha(1+\omega) - \alpha E}{(1-\omega)^2 + 4\alpha\omega + (\beta-\alpha-\omega)E}\sqrt{\beta}, \\ C &= \frac{\sqrt{\alpha} - \lambda\sqrt{\beta}}{1+\lambda^2} = \frac{(1+\omega)^2 - \beta(1+3\omega) - (\alpha+\omega)E}{(1-\omega)^2 + 4\alpha\omega + (\beta-\alpha-\omega)E}\sqrt{\alpha}, \\ D &= \lambda\frac{\lambda\sqrt{\alpha} + \sqrt{\beta}}{1+\lambda^2} = \frac{\beta(1-\omega) + \beta E}{(1-\omega)^2 + 4\alpha\omega + (\beta-\alpha-\omega)E}\sqrt{\alpha}, \end{aligned} \tag{A.12}$$

where $E = \sqrt{(1-\omega)^2 + 4\alpha\omega}$. On the other hand, denote the quantum state evolving under the Hamiltonian $H(-\theta) = H(-\omega t)$ by $|\varphi'(t)\rangle$, and it can be derived from $|\varphi\rangle$ by exchanging all ω by $-\omega$.

Then, the final state of the j-th evolution is

$$|\psi_j(T)\rangle = \frac{1}{\sqrt{2}}(|0\rangle|\varphi(T)\rangle + |1\rangle|\varphi'(T)\rangle), \tag{A.13}$$

where $T = 2^j\pi/\omega$, for $j = 1, \cdots, m$. Let $|\varphi_\perp\rangle$ be a state in the span space of $|\hat{0}\rangle$ and $|\hat{1}\rangle$, that is orthogonal to $|\varphi\rangle$. From (A.13), we find that

$$|\psi_j(T)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + \langle\varphi(T)|\varphi'(T)\rangle|1\rangle)|\varphi\rangle + \frac{1}{\sqrt{2}}\langle\varphi_\perp(T)|\varphi'(T)\rangle|1\rangle|\varphi_\perp(T)\rangle. \tag{A.14}$$

Hence, with probability

$$p_s = \frac{1 + |\langle\varphi(T)|\varphi'(T)\rangle|^2}{2}, \tag{A.15}$$

the relative phase will be the argument of $\langle\varphi(T)|\varphi'(T)\rangle$. We know that

$$\langle\varphi(T)|\varphi'(T)\rangle = (AA'+DD')e^{i\mu_1}+(BB'+CC')e^{-i\mu_1}+(AB'+C'D)e^{i\mu_2}+(A'B+CD')e^{-i\mu_2}, \tag{A.16}$$

where A', B', C', D' are derived from A, B, C, D by exchanging ω by $-\omega$, and

$$\begin{aligned}\mu_1 &= \frac{1}{2}(\sqrt{(1-\omega)^2+4\alpha\omega} - \sqrt{(1+\omega)^2-4\alpha\omega})T,\\ \mu_2 &= \frac{1}{2}(\sqrt{(1-\omega)^2+4\alpha\omega} + \sqrt{(1+\omega)^2-4\alpha\omega})T\end{aligned} \tag{A.17}$$

From the assumption $\omega \ll 1$, the relative phase can be estimated by $\langle\varphi(T)|\varphi'(T)\rangle$, from $AA' + DD'$, $BB' + CC'$, $AB' + C'D$ and $A'B + CD'$. Since the four terms above share the same denominator, we can first calculate the denominator F, then calculate their numerators separately. The denominator is

$$\begin{aligned}F =& \Big((1-\omega)^2 + 4\alpha\omega + (1-2\alpha-\omega)\sqrt{(1-\omega)^2+4\alpha\omega}\Big)\\ &\cdot\Big((1+\omega)^2 - 4\alpha\omega + (1-2\alpha+\omega)\sqrt{(1+\omega)^2-4\alpha\omega}\Big)\\ =& \big((1-\omega)^2+4\alpha\omega\big)\cdot\big((1+\omega)^2-4\alpha\omega\big)\\ &+ \big((1+\omega)^2-4\alpha\omega\big)(1-2\alpha-\omega)(1-(1-2\alpha)\omega+2\alpha\beta\omega^2)\\ &+ \big((1-\omega)^2+4\alpha\omega\big)(1-2\alpha+\omega)(1+(1-2\alpha)\omega+2\alpha\beta\omega^2)\\ &+ \big((1-2\alpha)^2-\omega^2\big)(1-(1-2\alpha)\omega+2\alpha\beta\omega^2)(1+(1-2\alpha)\omega+2\alpha\beta\omega^2)+O(\omega^3)\\ =& 1-2(1-8\alpha\beta)\omega^2+2(1-2\alpha)\big(1-2(1-5\alpha\beta)\omega^2\big)\\ &+(1-2\alpha)^2-\omega^2-(1-2\alpha)^2(1-8\alpha\beta)\omega^2+O(\omega^3)\\ =& 4\beta^2(1-2\beta(1-4\alpha)\omega^2)+O(\omega^3).\end{aligned} \tag{A.18}$$

Then

$$
\begin{aligned}
&(AA' + DD')F \\
=&\beta\left((1-\omega)^2 - \alpha(1-3\omega) + (1-\alpha-\omega)\sqrt{(1-\omega)^2+4\alpha\omega}\right) \\
&\cdot\left((1+\omega)^2 - \alpha(1+3\omega) + (1-\alpha+\omega)\sqrt{(1+\omega)^2-4\alpha\omega}\right) \\
&+\alpha\beta^2\left((1-\omega)+\sqrt{(1-\omega)^2+4\alpha\omega}\right)\cdot\left((1+\omega)+\sqrt{(1+\omega)^2-4\alpha\omega}\right) \\
=&\beta\left((1-\omega^2)^2 - 2\alpha(1-5\omega^2) + \alpha^2(1-9\omega^2) + \alpha\beta(1-\omega^2)\right) \\
&+\beta\left(\beta+\beta(2\beta-1)\omega+(2\alpha-1)\omega^2\right)\left(1-(1-2\alpha)\omega+2\alpha\beta\omega^2\right) \\
&+\beta\left(\beta-\beta(2\beta-1)\omega+(2\alpha-1)\omega^2\right)\left(1+(1-2\alpha)\omega+2\alpha\beta\omega^2\right) \\
&+\beta(\beta-\omega^2)(2\alpha-1)\omega^2(1-(1-2\alpha)\omega+2\alpha\beta\omega^2)(1+(1-2\alpha)\omega+2\alpha\beta\omega^2)+O(\omega^3) \\
=&4\beta^2(1-\beta(2-5\alpha)\omega^2)+O(\omega^3).
\end{aligned}
\tag{A.19}
$$

Hence,

$$
AA' + DD' = \frac{4\beta^2(1-\beta(2-5\alpha)\omega^2)+O(\omega^3)}{4\beta^2(1-2\beta(1-4\alpha)\omega^2)+O(\omega^3)} = 1 - 3\alpha\beta\omega^2 + O(\omega^3). \tag{A.20}
$$

$$
\begin{aligned}
&(BB' + CC')F \\
=&\alpha^2\beta(1+\omega-\sqrt{(1-\omega)^2+4\alpha\omega})(1-\omega-\sqrt{(1+\omega)^2-4\alpha\omega}) \\
&+\alpha\left((1+\omega)^2-\beta(1+3\omega)-(\alpha+\omega)\sqrt{(1-\omega)^2+4\alpha\omega}\right) \\
&\cdot\left((1-\omega)^2-\beta(1-3\omega)-(\alpha-\omega)\sqrt{(1+\omega)^2-4\alpha\omega}\right) \\
=&\alpha\left(\alpha+(9\alpha\beta+\beta^2-2)\omega^2\right) \\
&+\left(\alpha+(1-2\alpha)\alpha\omega+(1-2\alpha)\omega^2\right)\left(1-(1-2\alpha)\omega+2\alpha\beta\omega^2\right) \\
&+\left(\alpha-(1-2\alpha)\alpha\omega+(1-2\alpha)\omega^2\right)\left(1+(1-2\alpha)\omega+2\alpha\beta\omega^2\right) \\
&+\alpha(\alpha-\omega^2)(1-(1-2\alpha)\omega+2\alpha\beta\omega^2)\cdot(1+(1-2\alpha)\omega+2\alpha\beta\omega^2)+O(\omega^3) \\
=&-4\alpha\beta^3+O(\omega^3).
\end{aligned}
\tag{A.21}
$$

Hence,

$$
BB' + CC' = \frac{-4\alpha\beta^3+O(\omega^3)}{4\beta^2(1-2\beta(1-4\alpha)\omega^2)+O(\omega^3)} = -\alpha\beta\omega^2 + O(\omega^3). \tag{A.22}
$$

Moreover,

$$\begin{aligned}
&(AB' + C'D)F \\
=&\alpha\beta\left((1-\omega)^2 - \alpha(1-3\omega) + (\beta-\omega)\sqrt{(1-\omega)^2+4\alpha\omega}\right)\cdot(1-\omega-\sqrt{(1+\omega)^2-4\alpha\omega}) \\
&+\alpha\beta\left((1-\omega)^2 - \beta(1-3\omega) + (\alpha-\omega)\sqrt{(1+\omega)^2-4\alpha\omega}\right)\cdot(1-\omega+\sqrt{(1-\omega)^2+4\alpha\omega}) \\
=&\alpha\beta(1-\omega)(1-\omega+2\omega^2) \\
&+\alpha\beta(1-(3-2\beta)\omega+2\omega^2)(1-(1-2\alpha)\omega+2\alpha\beta\omega^2) \\
&-\alpha\beta(1+(3-2\beta)\omega+2\omega^2)(1+(1-2\alpha)\omega+2\alpha\beta\omega^2) \\
&-(1-2\omega)(1-(1-2\alpha)\omega+2\alpha\beta\omega^2)\cdot(1+(1-2\alpha)\omega+2\alpha\beta\omega^2)+O(\omega^3) \\
=&8\alpha\beta^3\omega^3+O(\omega^3).
\end{aligned} \tag{A.23}$$

Hence,

$$AB' + C'D = \frac{8\alpha\beta^3\omega^3+O(\omega^3)}{4\beta^2(1-2\beta(1-4\alpha)\omega^2)+O(\omega^3)} = 2\alpha\beta\omega^2+O(\omega^3). \tag{A.24}$$

Since $A'B + CD' = (AB' + C'D)'$,

$$A'B + CD' = 2\alpha\beta\omega^2+O(\omega^3). \tag{A.25}$$

Hence,

$$\begin{aligned}
\langle\varphi(T)|\varphi'(T)\rangle =&(1-3\alpha\beta\omega^2)e^{i\mu_1} - \alpha\beta\omega^2 e^{-i\mu_1} \\
&+4\alpha\beta\omega^2\cos(\mu_2)+O(\omega^3).
\end{aligned} \tag{A.26}$$

So

$$p_s = \frac{1+|\langle\varphi(T)|\varphi'(T)\rangle|^2}{2} \geq 1-8\alpha\beta\omega^2, \tag{A.27}$$

and

$$|arg(\langle\varphi(T)|\varphi'(T)\rangle) - \mu_1| \leq 8\alpha\beta\omega^2. \tag{A.28}$$

$$T = O\left((\frac{1}{\epsilon})^{3/2}\right) \tag{A.29}$$

Appendix B

Appendix of Hamiltonian Simulation

Proof of Lemma 6. Unwinding the recurrence for S_{2k} we see that

$$S_{2k}(\mathcal{H}_1,\ldots,\mathcal{H}_m,\Delta t) = \prod_{\ell=1}^{K} S_2((\mathcal{H}_1,\ldots,\mathcal{H}_m,z_\ell\Delta t) = \prod_{\ell=1}^{K}\left[\prod_{j=1}^{m} e^{-i\mathcal{H}_j z_\ell \Delta t/2}\prod_{j=m}^{1} e^{-i\mathcal{H}_j z_\ell \Delta t/2}\right],$$

where $K = 5^{k-1}$ and each z_ℓ is defined according to the recursive scheme, $\ell = 1,\ldots,K$. For the details, see the part of the text that follows (4.3). The bound (4.4), namely,

$$|z_\ell| \le \frac{4k}{3^k} \quad \text{for all } \ell = 1,\ldots,K,$$

holds independently of m, because it depends on the $k-1$st levels of the recursion tree and not on the leaf, $S_2((\mathcal{H}_1,\ldots,\mathcal{H}_m,z_\ell\Delta t)$, which ends the path corresponding to ℓ.

In the expression of $S_2((\mathcal{H}_1,\ldots,\mathcal{H}_m,z_\ell\Delta t)$ the sum of the magnitudes of the factors multiplying the Hamiltonials in the exponents is $m|z_\ell|\cdot|\Delta t|$, for all $\ell = 1,\ldots,K$. Thus in the expression of S_{2k} above, the sum of the magnitudes of all factors multiplying the Hamiltonians in the exponents is

$$\sum_{\ell=1}^{K}(m|z_\ell|\cdot|\Delta t|) \le 5^{k-1} m\frac{4k}{3^k}|\Delta t|.$$

Define

$$d_k := m\frac{4}{3}k\left(\frac{5}{3}\right)^{k-1} \quad k \ge 1. \tag{B.1}$$

Equivalently, one can view the expression for S_{2k} above as a product of exponentials of the form $e^{\mathcal{H}_j r_{j,n}\Delta t}$, where $\sum_{n=1}^{N_j} r_{j,n} = 1$, $j = 1, \cdots, m$, and N_j is the number of occurrences of $\mathcal{H}_j$ in S_{2k}. Recall that for $m = 2$ we used s_n to denote $r_{1,n}$ and z_n to denote $r_{2,n}$. With this notation and using (B.1) we have

$$\sum_{j,n} |r_{j,n}| \leq d_k. \tag{B.2}$$

(Recall the derivation of (4.6).)

Expanding the factors of S_{2k} in a power series individually, and then carrying out the multiplications amongst them, we conclude that S_{2k} is given by an infinite sum whose terms have the form

$$\prod_{(j,n)} \frac{1}{\gamma_{j,n}!}\, \mathcal{H}_j^{\gamma_{j,n}}\, [-i\, r_{j,n}\, \Delta t]^{\gamma_{j,n}}. \tag{B.3}$$

The factors of these products are specified by the Hamiltonians H_j and the order of their occurrences after unwinding the recurrence for S_{2k}, where $j = 1, \ldots, m$ and $\gamma_{j,n} = 0, 1, 2, \ldots$, for all $n = 1, \ldots, N_j$.

Consider the terms that contain only $\mathcal{H}_1$ and, therefore, have $\gamma_{j,n} = 0$, for $n = 1, \ldots, N_j$ and $j = 2, \ldots, m$. The sum of these terms is

$$\begin{aligned}
&\sum_{\gamma_{j,n}=0 \text{ for } j\neq 1} \prod_{(j,n)} \frac{1}{\gamma_{j,n}!}\, \mathcal{H}_j^{\gamma_{j,n}}\, [-i\, r_{j,n}\, \Delta t]^{\gamma_{j,n}} \\
=&\sum_{\gamma_{1,1}=\cdots=\gamma_{1,N_1}=0}^{\infty} \prod_{(1,n)} \frac{1}{\gamma_{1,n}!}\, \mathcal{H}_1^{\gamma_{1,n}}\, [-i\, r_{1,n}\, \Delta t]^{\gamma_{1,n}} \\
=&\prod_{n=1}^{N_1} \sum_{\gamma_{1,n}} \frac{1}{\gamma_{1,n}!} H_1^{\gamma_{1,n}}\, [-i r_{1,n}\, \Delta t]^{\gamma_{1,n}} = \prod_{n=1}^{N_1} e^{-i\mathcal{H}_1 r_{1,n}\Delta t} \\
=&e^{-i\sum_n r_{1,n} H_1 \Delta t} = e^{-i\mathcal{H}_1\Delta t}.
\end{aligned} \tag{B.4}$$

On the other hand,

$$e^{-i\sum_{j=1}^m \mathcal{H}_j\Delta t} = I + \left(-i\sum_{j=1}^m \mathcal{H}_j\Delta t\right) + \cdots + \frac{1}{k!}\left(-i\sum_{j=1}^m \mathcal{H}_j\Delta t\right)^k + \cdots, \tag{B.5}$$

and the terms that contain only $\mathcal{H}_1$ have the sum

$$\sum_{k=0}^{\infty} \frac{1}{k!}\mathcal{H}_1^k(-i\Delta t)^k = e^{-i\mathcal{H}_1\Delta t}. \tag{B.6}$$

Let us now consider the error bound in (4.25). The sum of the terms with only $\mathcal{H}_1$ in S_{2k+1} and $\exp(\sum_{j=1}^m H_j\Delta t)$ is the same and cancels out when we subtract one from the other. Moreover, in $\exp(-i\sum_{j=1}^m \mathcal{H}_j\Delta t) - S_{2k}(\Delta t)$ we know that the terms of order up to $2k$ also cancel out, see (4.25). From this we conclude that the error is proportional to $\|\mathcal{H}_2\||\Delta t|^{2k+1}$.

Consider

$$\exp(-i(\mathcal{H}_1+\cdots+\mathcal{H}_m)\Delta t) - S_{2k}(\mathcal{H}_1,\ldots,\mathcal{H}_m,\Delta t) = \sum_{l=2k+1}^{\infty}\big[R_l(\Delta t) - T_l(\Delta t)\big], \quad \text{(B.7)}$$

where $R_l(\Delta t)$ is the sum of all terms in $\exp(-i(\mathcal{H}_1+\cdots+\mathcal{H}_m)\Delta t)$ corresponding to Δt^l and $T_l(\Delta t)$ is the sum of all terms in S_{2k} corresponding to Δt^l. We can ignore the terms in $T_l(\Delta t)$ and $R_l(\Delta t)$ that contain only $\mathcal{H}_1$ (and not $\mathcal{H}_2$) as a factor.

Then

$$\|R_l(\Delta t)\| = \left\|\frac{1}{l!}\left(\sum_{j=1}^m \mathcal{H}_j\Delta t\right)^l - \frac{1}{l!}\mathcal{H}_1^l\Delta t^l\right\| \le \frac{m^l}{l!}\|\mathcal{H}_2\||\Delta t|^l, \quad \text{(B.8)}$$

because there are m^l-1 terms in R_l and each norm is at most $\|\mathcal{H}_2\||\Delta t|^l/l!$.

From Eq.(B.3) we have

$$T_l(\Delta t) = \sum_{\sum\gamma_{(j,n)}=l} \frac{\prod_{(j,n)} r_{j,n}^{\gamma_{j,n}}}{\prod_{(j,n)}\gamma_{j,n}!}\prod_{(j,n)}\mathcal{H}_j^{\gamma_{j,n}}\Delta t^l, \quad \text{(B.9)}$$

where $\sum_n \gamma_{1,n} \neq l$, i.e., there is no terms containing only $\mathcal{H}_1$. So, $\|\prod_{(j,n)}\mathcal{H}_j^{\gamma_{j,n}}\| \le \|\mathcal{H}_2\|$ and

$$\|T_l(\Delta t)\| \le \sum_{\sum\gamma_{j,n}=l} \frac{\prod_{j,n}|r_{j,n}|^{\gamma_{j,n}}}{\prod_{j,n}\gamma_{j,n}!}\|\mathcal{H}_2\||\Delta t|^l. \quad \text{(B.10)}$$

To calculate the coefficients of the sum, we consider

$$\begin{aligned}\prod_{(j,n)}\exp(|r_{j,n}\Delta t|) &= \prod_{(j,n)}\sum_{\gamma_{j,n}=0}^{\infty}\frac{1}{\gamma_{j,n}!}|r_{j,n}\Delta t|^{\gamma_{j,n}} \\ &= \sum_{l=0}^{\infty}\sum_{\sum\gamma_{j,n}=l}\frac{\prod_{j,n}|r_{j,n}|^{\gamma_{j,n}}}{\prod_{j,n}\gamma_{j,n}!}|\Delta t|^l.\end{aligned} \quad \text{(B.11)}$$

Hence the coefficient of $|\Delta t|^l$ in Eq.(B.10) is equal to that in Eq.(B.11). Also

$$\prod_{j,n}\exp(|r_{j,n}\Delta t|) = \exp(\sum_{j,n}|r_{j,n}\Delta t|). \quad \text{(B.12)}$$

From Eq.(B.2) we obtain

$$\|T_l(\Delta t)\| = \frac{d_k^l}{l!}\|\mathcal{H}_2\||\Delta t|^l. \tag{B.13}$$

Therefore,

$$\begin{aligned}
\|\exp(\sum_{j=1}^{m}\mathcal{H}_j\Delta t) - S_{2k}(\Delta t)\| &\leq \sum_{l=2k+1}^{\infty}\|R_l(\Delta t)\| + \|T_l(\Delta t)\| \\
&\leq 2\sum_{l=2k+1}^{\infty}\frac{d_k^l}{l!}\|\mathcal{H}_2\||\Delta t|^l \\
&= 2\|\mathcal{H}_2\|\sum_{l=2k+1}^{\infty}\frac{1}{l!}|d_k\Delta t|^l \\
&\leq \frac{2}{(2k+1)!}\|\mathcal{H}_2\||d_k\Delta t|^{2k+1}\left(1-\frac{d_k|\Delta t|}{2k+2}\right)^{-1} \\
&\leq \frac{4}{(2k+1)!}\|\mathcal{H}_2\||d_k\Delta t|^{2k+1},
\end{aligned} \tag{B.14}$$

where the last two inequalities follow from the assumption $d_k|\Delta t| \leq k+1$ and an estimate of the tail of the Poisson distribution; see, e.g., [70]. □