Contributions to Information-Based Complexity and to Quantum Computing

Iasonas Petras

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

COLUMBIA UNIVERSITY

2013

ABSTRACT

Contributions to Information-Based Complexity and to Quantum Computing

Iasonas Petras

Multivariate continuous problems are widely encountered in physics, chemistry, finance and in computational sciences. Unfortunately, interesting real world multivariate continuous problems can almost never be solved analytically. As a result, they are typically solved numerically and therefore approximately.

In this thesis we deal with the approximate solution of multivariate problems. The complexity of such problems in the classical setting has been extensively studied in the literature. On the other hand the quantum computational model presents a promising alternative for dealing with multivariate problems. The idea of using quantum mechanics to simulate quantum physics was initially proposed by Feynman in 1982. Its potential was demonstrated by Shor's integer factorization algorithm, which exponentially improves the cost of the best classical algorithm known.

In the first part of this thesis we study the tractability of multivariate problems in the worst and average case settings using the real number model with oracles. We derive necessary and sufficient conditions for weak tractability for linear multivariate tensor product problems in those settings.

More specifically, we initially study necessary and sufficient conditions for weak tractability on linear multivariate tensor product problems in the worst case setting under the absolute error criterion. The complexity of such problems depends on the rate of decay of the squares of the singular values of the solution operator for the univariate problem. We show a condition on the singular values that is sufficient for weak tractability. The same condition is known to be necessary for weak tractability.

Then, we study linear multivariate tensor product problems in the average case setting

under the absolute error criterion. The complexity of such problems depends on the rate of decay of the eigenvalues of the covariance operator of the induced measure of the one dimensional problem. We derive a necessary and sufficient condition on the eigenvalues for such problems to be weakly tractable but not polynomially tractable.

In the second part of this thesis we study quantum algorithms for certain eigenvalue problems and the implementation and design of quantum circuits for a modification of the quantum NAND evaluation algorithm on k-ary trees, where k is a constant.

First, we study quantum algorithms for the estimation of the ground state energy of the multivariate time-independent Schrödinger equation corresponding to a multiparticle system in a box. The dimension d of the problem depends linearly to the number of particles of the system. We design a quantum algorithm that approximates the lowest eigenvalue with relative error ε for a non-negative potential V, where V as well as its first order partial derivatives are continuous and uniformly bounded by one. The algorithm requires a number of quantum operations that depends polynomially on the inverse of the accuracy and linearly on the number of the particles of the system. We note that the cost of any classical deterministic algorithm grows exponentially in the number of particles. Thus we have an exponential speedup with respect to the dimension of the problem d, when compared to the classical deterministic case.

We extend our results to convex non-negative potentials V, where V as well as its first order partial derivatives are continuous and uniformly bounded by constants C and C' respectively. The algorithm solves the eigenvalue problem for a sequence of convex potentials in order to obtain its final result. More specifically, the quantum algorithm estimates the ground state energy with relative error ε a number of quantum operations that depends polynomially on the inverse of the accuracy, the uniform bound C on the potential and the dimension d of the problem. In addition, we present a modification of the algorithm that produces a quantum state which approximates the ground state eigenvector of the discretized Hamiltonian within δ. This algorithm requires a number of quantum operations that depends pollynomially on the inverse of ε, the inverse of δ, the uniform bound C on the potential and the dimension d of the problem.

Finally, we consider the algorithm by Ambainis et.al. that evaluates balanced binary

NAND formulas. We design a quantum circuit that implements a modification of the algorithm for k-ary trees, where k is a constant. Furthermore, we design another quantum circuit that consists exclusively of Clifford and T gates. This circuit approximates the previous one with error ε using the Solovay-Kitaev algorithm.

Table of Contents

List of Figures

List of Tables

Acknowledgments

It is imperative to start by acknowledging the instrumental role the following people had to the completion of my thesis. First, I would like to thank my advisor, Prof. Joseph F. Traub, for his encouragement, guidance and support throughout my studies at Columbia University as well as his focus on research with impact. I would also like to thank Dr. Anargyros Papageorgiou for the considerable time and effort he allocated into my development as a researcher, through numerous research meetings and insightful discussions that sharpened my skills. Last but not least, I am grateful to Prof. Henryk Woźniakowski initially for introducing me, in my first days in New York, to information-based complexity, but most importantly for always being open to research discussions and offering advice throughout the years.

In addition, I would like to thank Prof. Alfred V. Aho and Prof. Arthur G. Werschulz for the time and effort they dedicated on my work, as members of my dissertation committee.

The work in Chapters 2 and 3 was motivated by discussions with Anargyros Papageorgiou and Henryk Woźniakowski on the tractability of linear tensor product problems.

The research in Chapters 4 and 5 was stimulated by the previous work on Sturm-Liouville eigenvalue problems by Anargyros Papageorgiou and Henryk Woźniakowski, as well as the work on simulating Hamiltonians by Anargyros Papageorgiou and Chi Zhang.

Furthermore, the study of implementation methods in chapter 6 was initiated during my internship at NEC Laboratories America. I would like to thank Dr. Martin Rötteler and the members of the Quantum IT Group at NEC for many thought-provoking and insightful discussions.

Finally, I would like to thank my wife and my family for their support, understanding and motivation throughout the years.

This thesis is dedicated to my family, for always being there for me.

Chapter 1

Introduction

In this thesis we study algorithms for multivariate problems. A great number of the important real life multivariate problems are extremely difficult to solve. Analytical solutions cannot be generally derived. Hence numerical methods are employed to approximate the solution.

Some problems are intrinsically difficult to solve. There are known lower bounds showing that the amount of computational resources required to solve them is huge. On the other hand, there are many interesting problems for which lower and upper bounds have an exponential gap depending on the dimension of the problem. Even more interestingly, the choice of the setting can make a significant difference. For example, there are problems that are very hard in the worst case but are easy when we allow randomization, such as high dimensional integration problems, see [8; 66].

Quantum computers offer a promising alternative. Richard Feynman initiated research in computing based on quantum mechanics [27]. More specifically he proposed to take advantage of the features of quantum mechanical systems that make it hard to simulate using classical computers. Peter W. Shor later developed a quantum algorithm for prime factorization that is exponentially faster than any classical algorithm known [61]. This result further motivated the use of quantum computers in dealing with problems that are or appear to be computationally hard for the classical computer.

This thesis is divided into two parts. In the first part, we study the complexity of linear multivariate tensor product problems in two settings; the worst case (Chapter 2) and the average case (Chapter 3). We show necessary and sufficient conditions for the problems to

be weakly tractable[1].

In the second part, we switch to a different computational model, the quantum model, and derive upper bounds on the computational resources required to solve eigenvalue problems. In addition, we estimate the resources required to implement a modification of the algorithm in [4, Fig. 2]. In Chapter 4, we present a quantum algorithm approximating the lowest eigenvalue of a certain class of Hamiltonians of a quantum system, an important problem in physics and chemistry. In Chapter 5, we derive another quantum algorithm for a modification of the previous class of Hamiltonians. Finally, in Chapter 6, we design a quantum circuit implementing a modification of the quantum algorithm in [4, Fig. 2]. The modified algorithm corresponds to complete k-ary trees, where $k \in \{2, 3, \ldots\}$ is a constant.

1.1 Information-based complexity

Computational complexity studies the intrinsic difficulty of approximating the solution of a problem and is independent of the algorithm used. Information-based complexity (IBC) is used to analyze the computational complexity of continuous problems. The computational complexity of a problem is defined as the minimal number of information and combinatory operations required to solve the problem within error ε. The information complexity, typically denoted by $n(\varepsilon)$, is the minimal number of information operations required to solve the problem within error ε. As a result, the information complexity is a lower bound on the computational complexity of the problem. For many problems, the information complexity is proportional to the computational complexity, even though there are exceptions. IBC focuses on the study of the information complexity of a problem in different settings, such as the worst case, the average case, the randomized and probabilistic settings. Recently, IBC was extended to the quantum setting [35].

A problem is typically defined by a class of functions $F = \{f : A \to B\}$, and a solution operator $S : F \to G$, where F and G are normed spaces. We approximate the solution $S(f)$ by $U_n(f)$, an algorithm that uses n information operations. The goal is to compute an

[1]For the definition of weak tractability see Section 1.2

ε-approximation of the solution element $S(f)$, namely

$$\|S(f) - U_n(f)\|_G \leq \varepsilon \quad \forall f \in F,$$

where ε is the error constraint.

Let

$$N(f) = [L_1(f), L_2(f), \ldots, L_n(f)]$$

be the information we have about f, where $L_1, L_2, \ldots, L_n$ are continuous linear functionals on F. The permissible information operations are either arbitrary linear functionals $(L_1(f), L_2(f), \ldots, L_n(f) \in \Lambda^{\text{all}})$ or just function evaluations $(L_1(f), L_2(f), \ldots, L_n(f) \in \Lambda^{\text{std}})$.

We use the real number model with queries. In this model we can store and perform arithmetic operations on real numbers exactly with unit cost. Under fairly modest assumptions, such as the stability assumption and the error demand assumption [49, pg. 101], results in this model are predictive of results in fixed precision arithmetic.

Consider algorithms that use n information operations of the form

$$A_n(f) = \phi(N(f)),$$

where $\phi : N(F) \to G$.

We need to choose the error criterion, as well as the setting, to completely specify the problem. The worst case error under the absolute error criterion is defined as

$$e^{\text{wor}}(A_n, S, F) = \sup_{f \in F} \|S(f) - A_n(f)\|_G.$$

The nth minimal error

$$e^{\text{wor}}(n) = e^{\text{wor}}(n, S, F) = \inf_{A_n} e^{\text{wor}}(A_n, S)$$

is the minimal error we can achieve for any algorithm $A_n(f)$ that uses n information operations. Let $n^{\text{wor}}(\varepsilon, S, F)$ denote the minimal number n for which there exists an algorithm A_n having worst case error ε, so that

$$n^{\text{wor}}(\varepsilon, S, F) = \min\{n \mid \text{there exists } A_n \text{ with } \mathrm{e}^{\text{wor}}(A_n, S, F) \leq \varepsilon\}.$$

This is the information complexity of S in the worst case.

For the normalized error criterion the error becomes $e^{\mathrm{wor}}(A_n, S, F)/e_0^{\mathrm{wor}}$, where e_0^{wor} denotes the initial error

$$e_0^{\mathrm{wor}} = e_0^{\mathrm{wor}}(S, F) = \inf_{g \in G} \sup_{f \in F} \|S(f) - g\|_G,$$

namely the minimal worst case error of a constant algorithm $A_n(f) = g$, $\forall f \in F$.

In the average case setting we consider the average performance of an algorithm with respect to a probability measure μ defined on Borel sets of F. The space F is a subset of the separable Banach space $\tilde{F}$. Similarly to what we did above, we consider a solution operator $S : \tilde{F} \to G$, where G is a separable Banach space. Then $\nu = \mu S^{-1}$ is a probability measure on the set of solution elements $S(f)$.

Consider a measurable algorithm $A = \phi \circ N : F \to G$. The average case error of the algorithm is

$$e^{\mathrm{avg}}(A, S, F) = \left(\int_F \|S(f) - A(f)\|^2 \mu(df) \right)^{1/2}.$$

So far, we have assumed that the information N has fixed cardinality. Alternatively, the total number $n(f)$ of information operations on the problem element f can be obtained through computing successive values $y_i = L_i(f)$. In particular, suppose we have already calculated the values $y_1 = L_1(f)$ up to $y_i = L_i(f, y_1, y_2, \ldots, y_{i-1})$. Based on those values, one decides whether another functional L_{i+1} is needed. If not, $n(f) = i$ and $N(f) = [y_1, y_2, \ldots, y_i]$. Otherwise, $y_{i+1} = L_{i+1}(f, y_1, y_2, \ldots, y_i)$ is evaluated.

Now consider boolean functions of the form $\mathrm{ter}_i : \mathbb{R} \to \{0, 1\}$ known as *termination functions*, that decide whether on the ith step another functional is needed ($\mathrm{ter}_i(y_1, y_2, \ldots, y_i) = 1$) or not ($\mathrm{ter}_i(y_1, y_2, \ldots, y_i) = 0$). As a result, the cardinality $n(f)$ is

$$n(f) = \min\{i \,|\, \mathrm{ter}_i(y_1, y_2, \ldots, y_i) = 1\},$$

with the convention that $\min \emptyset = \infty$.

In the average case we define the average cardinality of N to be

$$\mathrm{card}^{\mathrm{avg}}(N) = \int_{\mathrm{F}} n(f) \mu(df).$$

We consider the average cardinality to be a measure of the cost of the algorithm A. Thus we can compare different algorithms with average cardinality at most n. Define

$$e^{\mathrm{avg}}(n) = \inf\{e^{\mathrm{avg}}(\phi \circ N, S, F) \,|\, \mathrm{card}^{\mathrm{avg}}(N) \leq n\}.$$

For the absolute error, we define the information complexity of S

$$n^{\text{avg}}(\varepsilon, S, F) = \min\{n \,|\, \text{there exists } A = \phi \circ N \text{ with } \text{card}^{\text{avg}}(N) \leq n, \text{e}^{\text{avg}}(A, S, F) \leq \varepsilon\}.$$

Similarly, for the normalized error the information complexity of S is

$$n^{\text{avg}}(\varepsilon, S, F) =$$
$$\min\{n \,|\, \text{there exists } A = \phi \circ N \text{ with } \text{card}^{\text{avg}}(N) \leq n, \text{e}^{\text{avg}}(A, S, F) \leq \varepsilon \, \text{e}^{\text{avg}}(0)\},$$

where $e^{\text{avg}}(0)$ is the average error of the optimal constant algorithm, which is considered to be the initial cost in the average case setting.

For more information on the error definitions and complexity in various settings and error criteria, the reader is referred to [49; 66; 67].

1.2 Tractability of multivariate problems

Multivariate continuous problems are common in chemistry, finance, physics and computational sciences. They are continuous problems defined on spaces of functions with d variables, where d is considered large. For example, a path integral can be approximated by another d-dimensional integral, provided that d is sufficiently large. The Schrödinger equation for p particles in $\mathbb{R}^3$ is another important example of a multivariate continuous problem, where $d = 3p$. Since the number of particles is usually large, the dimension of the wavefunction solving the Schrödinger equation is large as well.

Recall the notions we presented in Section 1.1 such as error and information complexity in the worst and average case, with the only difference being the inclusion of the dimension d of the problem S_d. We present definitions and examples of multivariate problems in Sections 2.2 and 3.1. For more details see [49].

Tractability provides a characterization of the intrinsic difficulty of the problem as expressed by the information complexity of the problem $n(\varepsilon, d) := n(\varepsilon, S_d)$. It is important that $n(\varepsilon, d)$ should not grow exponentially in either ε^{-1} or d, since the problem is *intractable* otherwise. Moreover, if the information complexity is exponential in d we follow R. E. Bellman [10] in saying that the problem suffers from the *curse of dimensionality*. Thus a necessary

condition for tractability is

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln n(\varepsilon, d)}{\varepsilon^{-1} + d} = 0. \tag{1.1}$$

When the condition (1.1) holds, the problem is *weakly tractable*.

To gain a better understanding of the difficulty of the problem, other types of tractability are widely used as well. The most commonly used type of tractability is *polynomial tractability*, introduced in [73], for which

$$n(\varepsilon, d) \leq C\varepsilon^{-p} d^q,$$

for $C, p, q \geq 0$ constants. This means that we have polynomial tractability when the information complexity is upper-bounded by a polynomial in ε^{-1} and d. When $q = 0$, we have *strong polynomial tractability*, which in turn implies that the problem's intrinsic difficulty does not depend on the number of variables.

Another type of tractability, *quasi-polynomial tractability*, has attracted interest [29]. It originates from the need to further categorize the type of tractability for weakly tractable problems. The computational complexity of a weakly tractable problem might be greater than any polynomial and less than any exponential in d and ε^{-1}, but a more accurate characterization might be critical.

Finally, one can consider other tractability criteria. For example one may require that tractable problems are those whose information complexity $n(\varepsilon, d)$ is not exponential in d or $\left(\log \varepsilon^{-1}\right)^{\kappa}$, with $\kappa \geq 1$ a constant. This criterion implies that "efficient" algorithms are only those with "sub-exponential" cost in both the dimension of the problem d and any power of the number of bits required for the accuracy ε.

Weak tractability is represented in this case by the notion of $\ln^{\kappa}$-weak tractability, with $\kappa \geq 1$ a constant. The necessary condition for $\ln^{\kappa}$-weak tractability is

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln n(\varepsilon, d)}{(\ln \varepsilon^{-1})^{\kappa} + d} = 0. \tag{1.2}$$

Furthermore, a problem is polylog tractable when the information complexity is upper-bounded by a polynomial in $\ln \varepsilon^{-1}$ and d. Similarily, the problem is strongly polylog tractable if the information complexity does not depend on d and depends polynomially in $\ln \varepsilon^{-1}$.

Results based on these new types of tractability have been presented for linear and linear tensor product problems in the worst case [52] and for the integration of weighted Korobov spaces [43].

1.3 Quantum Computing

1.3.1 Quantum mechanics

A quantum mechanical system is described by a complete complex vector space $\mathcal{H} = \{\psi : \Omega \times \mathbb{R} \to \mathbb{C}\}$ equipped with an inner product structure $\langle \cdot | \cdot \rangle$, i.e. a Hilbert space. The state of the quantum system corresponds to an element of the space $|\psi\rangle \in \mathcal{H}$, where the element is normalized, i.e. $\|\psi\| = \sqrt{\langle \psi | \psi \rangle} = 1$. For example, the state of a one qubit system is an element of $\mathbb{C}^2$, while a state of an n qubit system is represented by an element in the tensor product space $\mathcal{H}_n = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 = \left(\mathbb{C}^2\right)^{\otimes n}$.

The evolution of the state of a closed quantum system is described by a partial differential equation, the *Schrödinger equation*

$$i\hbar \frac{\partial}{\partial t} |\Psi(\mathbf{x}, t)\rangle = H(t) \, |\Psi(\mathbf{x}, t)\rangle, \tag{1.3}$$

where $H(t)$ is the time-dependent Hermitian operator known as the Hamiltonian of the system, $i = \sqrt{-1}$ and $\hbar$ is the reduced Planck constant, following the notation in [60]. Since $H(t)$ is Hermitian, the state evolves according to

$$|\Psi(t)\rangle = U_t \, |\Psi(0)\rangle,$$

where U_t is a unitary transformation.

One method to solve the Schrödinger equation analytically is by separation of variables. Namely, one can look for solutions $|\Psi(\mathbf{x}, t)\rangle$ that are products of the form

$$|\Psi(\mathbf{x}, t)\rangle = |\psi(\mathbf{x})\rangle |\phi(t)\rangle.$$

This leads to

$$H|\psi(\mathbf{x})\rangle = E \, |\psi(\mathbf{x})\rangle, \tag{1.4}$$

for H a time-independent Hamiltonian operator, $E \in \mathbb{R}$ and $\mathbf{x}$ the state variable, see [30]. Note that E is an eigenvalue of H and $|\psi(\mathbf{x}, t)\rangle$ is an eigenfunction corresponding to E.

Equation (1.4) is known as the *time-independent Schrödinger equation*[2] and its solution $|\Psi(\mathbf{x},t)\rangle$ is

$$|\Psi(\mathbf{x},t)\rangle = e^{-itH/\hbar} \cdot |\Psi(\mathbf{x},0)\rangle.$$

Observe that $e^{-itH/\hbar}$ is unitary, since H is Hermitian.

For a p particle quantum system with masses $m_1, m_2, \ldots, m_p$ the above equation becomes

$$\left(-\hbar^2 \sum_{j=1}^{p} \frac{1}{2m_j} \sum_{k=1}^{3} \frac{\partial^2}{\partial x_{j,k}^2} + V(\mathbf{x})\right) |\psi(\mathbf{x})\rangle = E\,|\psi(\mathbf{x})\rangle, \tag{1.5}$$

where $\{x_{1,1}, x_{1,2}, x_{1,3}, \ldots, x_{p,1}, x_{p,2}, x_{p,3}\}$ denotes the set of the state variables, assuming that the state of each particle belongs to $\mathbb{R}^3$.

We can simplify equation (1.5) if we assume that all particles have the same mass m with $\hbar^2/m = 1$. The time independent Schrödinger equation then becomes

$$\left(-\frac{1}{2}\Delta + V(\mathbf{x})\right) |\psi(\mathbf{x})\rangle = E|\psi(\mathbf{x})\rangle, \tag{1.6}$$

where $\Delta = \sum_{j=1}^{3p} \frac{\partial^2}{\partial x_j^2}$ is the Laplacian operator.

The Schrödinger equation determines several important properties of the quantum system. Firstly, the solution of the time dependent form (1.3) corresponds to the state $|\Psi(\mathbf{x},t)\rangle$ of the system at time $t > 0$, given the initial state $|\Psi(\mathbf{x},0)\rangle$. For a multiparticle quantum system, finding the solution of (1.5) is an extremely hard computational problem given arbitrary initial state and Hamiltonian. The cost of any algorithm solving the problem in its generic form is at least exponential in the dimension of the problem and consequently in p. There are cases however where this problem is easy. In addition, the lowest eigenvalue of the Hamiltonian operator in (1.4) is the ground state energy of the quantum system, which is widely used in physics and chemistry. Later in this thesis we present algorithms to approximate the ground state energy for Hamiltonians that correspond to smooth, bounded potentials.

For more details on the subject, we refer the reader to [30; 39; 47; 60].

1.3.2 Quantum algorithms

The idea of using quantum mechanical systems for computational tasks originates in Richard Feynman's proposal to use such systems to simulate quantum mechanics [27]. Feynman's

[2] also known as Sturm-Liouville equation in mathematics [54].

main idea is based on the belief that those exact features that hinder the simulation of the quantum system by a classical one can be beneficial when used by a quantum-based computational machine. Deutsch, based on Feynman's suggestion, developed a *quantum Turing machine* (QTM) that is computationally equivalent to the classical Turing machine [20]. This implies that a quantum computer can solve the same problems a classical one can. Even more importantly there are problems for which the quantum algorithm is exponentially faster than any deterministic classical algorithm; e.g. the Deutsch and Josza algorithm [21] and performing a continuous time walk on a specific graph [14]. However these are toy problems.

In mid 90's two very important quantum algorithms where developed. Peter W. Shor showed a quantum algorithm for factoring any large number N into its prime factors [61] with poly($\log N$) running time. Any known classical algorithm for the same problem is exponentially slower than the quantum algorithm. A few years later, Lov K. Grover presented an oracle based quantum algorithm finding a specific element in an unstructured database [31]. His algorithm finds that element with high probability after $O(\sqrt{N})$ queries to the oracle, while any classical algorithm requires $\Omega(N)$ queries.

These developments have led to a surge of interest on quantum algorithms for both discrete and continuous mathematical problems. Examples include approximate counting [12], approximating the median [46], solving linear systems of equations [33], evaluating NAND formulas [4; 25], implementing Markov based (Szegedy) quantum walks [44], finding the lowest eigenvalue of the Sturm-Liouville equation [50; 55], quantum summation and integration[34; 35; 36; 48], path integration [68], numerical gradient estimation [38], performing a walk on a graph in both the quantum and the classical setting [16], triangle finding [45] and element distinctness [3].

A quantum algorithm on n qubits is described by a series of unitary matrices applied in sequential order

$$|\psi\rangle = U_K Q_f U_{K-1} Q_f \cdots U_1 Q_f U_0 |\psi_0\rangle,$$

where $U_0, \ldots, U_K$ do not depend on the function f. The unitary matrix Q_f is a *quantum query*, and depends on m function evaluations of f computed at some points x_i, $i = 0, 1, \ldots m$ in the domain of the function, where $m \leq 2^n$ and n the number of qubits. K denotes the number of quantum queries the algorithm requires.

For discrete problems f is usually considered to be a Boolean function $f : \{0, 1, \ldots, 2^n\} \to \{0, 1\}$. A *bit query* [31] is defined by

$$Q_f|j\rangle|k\rangle = |j\rangle|k \oplus f(j)\rangle,$$

where $|j\rangle \in \mathcal{H}_n$, $|k\rangle \in \mathcal{H}_1$ and $\oplus$ denotes addition modulo 2.

On the other hand, for continuous problems we generally assume that f is a real valued function, namely $f : \mathcal{D} \to \mathbb{R}$. Here we use different types of queries, such as

$$Q_f|j\rangle|k\rangle = |j\rangle|k \oplus_{2^m} \alpha(f(\beta(j)))\rangle, \tag{1.7}$$

where $|j\rangle \in \mathcal{H}_n$, $|k\rangle \in \mathcal{H}_m$, $\beta : \mathcal{H}_n \to \mathcal{D}$, $\alpha : \mathbb{R} \to \mathcal{H}_m$ and $\oplus_{2^m}$ is addition modulo 2^m [35; 36].

Abrams and Williams used the query

$$\begin{aligned} Q_f|j\rangle|0\rangle &= \sqrt{1 - f(j)^2}|j\rangle|0\rangle + f(j)|j\rangle|1\rangle \\ Q_f|j\rangle|1\rangle &= -f(j)|j\rangle|0\rangle + \sqrt{1 - f(j)^2}|j\rangle|1\rangle \end{aligned}$$

in their studies for integration [2]. A similar query was used by Novak in his study [48] of the complexity of integration for the Hölder class

$$\begin{aligned} Q_f|j\rangle|0\rangle &= \sqrt{1 - f(j)}|j\rangle|1\rangle + \sqrt{f(j)}|j\rangle|0\rangle \\ Q_f|j\rangle|1\rangle &= \sqrt{f(j)}|j\rangle|1\rangle - \sqrt{1 - f(j)}|j\rangle|1\rangle \end{aligned}$$

Whenever bit queries are mentioned throughout the thesis, it is implied that they are of the type defined in (1.7).

At the end of the quantum algorithm a measurement is performed on the final state $|\psi\rangle$. We obtain a result j, with certain probability $P(j)$, based on the final state of the register on which we perform the measurement.

The cost of a quantum algorithm is usually measured with respect to the number of quantum queries required to approximate the solution of a problem to within error ε and with probability $p > 3/4$. In fact, any probability $p > 1/2$ would also suffice.

1.3.3 The quantum gate model

Apart from the query-based computational model presented in Section 1.3.2, another widespread model of quantum computation is the quantum gate model [22]. It has been proven to be polynomially equivalent to the Quantum Turing Machine [74]; see also [20] for more on the Quantum Turing Machine model of computation. The quantum systems are represented by qubits, namely vectors in the two-dimensional Hilbert space $\mathbb{C}^2$, with the unit vector basis $\{|0\rangle, |1\rangle\}$. This model introduces *elementary gates*, transformations that are considered easy to execute. Quantum transformations acting on a small number of qubits are usually considered elementary, for example (see [47])

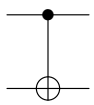
$$H = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}, S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix},$$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The gates X, Y, Z, are the Pauli matrices, while H, T, $S = T^2$, are the Hadamard, $\pi/8$ and phase gates respectively. All of these are single qubit gates. A commonly used two qubit gate is the "CNOT" gate

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

which acts as $\text{CNOT}\,|a\rangle|b\rangle = |a\rangle|b \oplus a\rangle$, i.e. it flips the second qubit if the first qubit is 1, otherwise it does nothing. It is represented by the circuit



One of the most important properties of the quantum gate model is that there are sets of gates comprised of a finite number of single and double qubit gates that are *universal*, namely any unitary operation can be approximated by gates of the set [23]. For example, $\{H, T, \text{CNOT}\}$ is a universal set of gates for quantum computation [47] and is known as the

Clifford[3] and T gate set. In the case where one deals with single qubit unitaries the Clifford gates are comprised of the H and S gates. On the other hand, in the case of multiple qubit unitaries the CNOT gate is included in the set. Clifford gates are generally considered easier to implement than general unitaries in many technologies and can be made universal with the addition of a single non-Clifford gate (in our case the T gate). It is common to design quantum circuits for algorithms using gates from this set as building blocks.

[3]The Clifford set of gates contains the Hadamard, phase and CNOT gates. Hence the Clifford and T gate set contains the Hadamard, phase, CNOT and T gates. However, since the phase gate is equivalent to two T gates acting one after the other, it is sometimes ommited. We include the phase gate in the Clifford and T gate set throughout the thesis.

Part I

Contributions to Information-Based Complexity

Chapter 2

On the tractability of linear tensor product problems in the worst case

2.1 Introduction

Consider the approximation of a problem $S = \{S_d\}$, where each of the S_d, $d \geq 1$, is a continuous linear operator defined on a space of functions f of d variables. Such problems are known as *linear multivariate problems*. Moreover, consider algorithms that approximate $S_d(f)$ using finitely many evaluations of arbitrary continuous linear functionals.

As already stated, the information complexity (for brevity, the complexity) is the minimal number of evaluations $n(\varepsilon, d)$ required to approximate S_d with accuracy ε. We remark that there are a variety of error criteria that one may consider for the accuracy of the algorithms but we limit ourselves to the worst case setting and the absolute error criterion.

In general S_{i+1} is not necessarily related to S_i, for $i = 1, 2, \ldots, d-1$. This is not the case though for linear multivariate tensor product problems. These are linear multivariate problems obtained by taking the tensor product of d copies of a single univariate linear problem. Thus

$$S_d = S_1^{\otimes d},$$

where S_1 is a given continuous linear operator.

In this case, the complexity of approximating S_d with accuracy ε depends on the singular

values of S_1 and, particularly, on their rate of decay [49, Ch. 5.2].

The squares of the singular values of S_1 are the eigenvalues, $\{\lambda_i\}_{i\in\mathbb{N}}$, of the operator $S_1^*S_1$, where the eigenvalues are indexed in non increasing order. Moreover, the relationship between the tractability of $S = \{S_d\}$ and the $\{\lambda_i\}_{i\in\mathbb{N}}$ is studied in detail in [49, Thm. 5.5]. In particular, we know that if a problem is weakly tractable with $\lambda_1 = 1$ and $\lambda_2 \in (0, 1)$ then $\lambda_n = o((\ln n)^{-2})$, as $n \to \infty$.

Proving the converse is Open Problem 26 in [49], which we solve.

2.2 Linear Tensor Product Problems in the wost case setting

A linear tensor product problem is defined in [49, Ch. 5.2] as a tensor product of a single univariate linear problem.

Let H_1 be an infinite dimensional separable Hilbert space of real univariate functions with its inner product denoted by $\langle\cdot,\cdot\rangle_{H_1}$, and let G_1 be an arbitrary Hilbert space.

Assume that $S_1 : H_1 \to G_1$ is a compact linear operator. The operator

$$W_1 := S_1^*S_1 : H_1 \to H_1$$

is positive semi-definite, self-adjoint and compact. Let us denote its ordered eigenvalues by $\{\lambda_i\}$, where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_i \geq \ldots$. They are the squares of the singular values of S_1. We denote the eigenpairs of W_1 by $\{(\lambda_i, e_i)\}_{i\in\mathbb{N}}$.

For $d \geq 1$, define $H_d = \bigotimes_{j=1}^d H_1$ to be the tensor product of the space H_1. This is a space of real functions of d variables. Similarly, let $G_d = \bigotimes_{j=1}^d G_1$. The linear tensor product problem is defined by considering the operator

$$S_d := \bigotimes_{j=1}^d S_1 : H_d \to G_d.$$

Observe that S_d is compact and that $\|S_d\|_{H_d} = \prod_{j=1}^d \left[\lambda_1\right]^{1/2}$. The problem $S = \{S_d\}$ is called the *linear tensor product problem.*

The non-negative definite, self adjoint and compact operator

$$W_d := S_d^*S_d : H_d \to H_d$$

has eigenpairs

$$\{\lambda_{d,i}, e_{d,i}\}_{i\in\mathbb{N}^d} \text{ with } \lambda_{d,i} = \prod_{j=1}^{d} \lambda_{i_j} \text{ and } e_{d,i} = \bigotimes_{j=1}^{d} e_{i_j},$$

for all $i = [i_1, i_2, \ldots, i_d] \in \mathbb{N}^d$. Let λ_{d,β_j} denote the j-th largest of all the $\lambda_{d,i}$ and let e_{d,β_j} denote the corresponding eigenvector. Clearly, $\lambda_{d,\beta_1} = \lambda_{d,1,\ldots,1} = \lambda_1^d$.

Suppose we can use arbitrary linear continuous functionals as information operations. Then it is known, see e.g. [66], that the algorithm

$$A_{n,d}(f) = \sum_{j=1}^{n} \langle f, e_{d,\beta_j}\rangle_{H_d} S_d e_{d,\beta_j}$$

minimizes the worst case error among all possible algorithms using at most n information operations. The worst case error is defined as

$$e(A_{n,d}) = \sup_{f\in H_d,\, \|f\|_{H_d}\le 1} \|S_d f - A_{n,d}(f)\|_{G_d}.$$

It is also known that $e(A_{n,d}) = \sqrt{\lambda_{d,\beta_{n+1}}}$.

For accuracy ε, the worst case information complexity of the problem S_d for the absolute error criterion is defined as the minimal number of information operations needed to guarantee that the worst case error is at most ε, and is given by

$$n(\varepsilon, d) = |\{i \in \mathbb{N}^d \ : \ \lambda_{d,i} > \varepsilon^2 \}|,$$

where $|\{\cdot\}|$ denotes the cardinality of the set.

2.3 Prior work

The relationship between the complexity $n(\varepsilon, d)$ of linear tensor product problems and the singular values of S_1 is extensively studied in [49, Thm. 5.5]. More precisely, the complexity depends on the eigenvalues $\{\lambda_i\}_{i\in\mathbb{N}}$ of the operator W_1. The problem $S = \{S_d\}$ is intractable when $\lambda_1 > 1$ and $\lambda_2 > 0$. Furthermore, the problem remains intractable even when $\lambda_1 = \lambda_2 = 1$.

When $\lambda_1 = 1$ and $\lambda_2 \in (0, 1)$ the problem is weakly tractable as long as the remaining eigenvalues decay sufficiently fast. More specifically, if a problem is weakly tractable then $\lambda_n = o((\ln n)^{-2})$, as $n \to \infty$. It is also demonstrated that $\lambda_n = o((\ln n)^{-2}(\ln\ln n)^{-2})$, as

$n \to \infty$ is a sufficient condition. This condition is stronger than necessary, as we demonstrate in the following section.

2.4 Weak Tractability

Open Problem 26 in [49] asks whether $\lambda_n = o(\ln^{-2} n)$ is a necessary and sufficient condition for a problem to be weakly tractable. We give an affirmative answer below.

Theorem 1. *Consider the linear tensor product problem in the worst case setting $S = \{S_d\}$ with $\lambda_1 = 1$ and $\lambda_2 \in (0, 1)$ with the absolute error criterion. Then S is weakly tractable iff*

$$\lambda_n = o((\ln n)^{-2}) \text{ as } n \to \infty.$$

Proof. We know that $\lambda_n = o((\ln n)^{-2})$ is a necessary condition for weak tractability of S [49, Thm. 5.5]. We show that it is also a sufficient condition.

When $\lambda_n = o(\ln^{-2} n)$ one may proceed as in [49] to obtain $\ln n(\varepsilon, 1) = o(\varepsilon^{-1})$. Indeed, $n(\varepsilon, 1) = \min\{n : \lambda_{n+1} < \varepsilon^2\} \leq \min\{n : \ln n = o(\varepsilon^{-1})\}$.

When $\lambda_2 \leq \varepsilon^2$ we know that $n(\varepsilon, 1) \leq 1$ and so we consider the case $\lambda_2 > \varepsilon^2$.

For $d \geq 2$, we are interested in eigenvalue products satisfying

$$\lambda_{j_1} \lambda_{j_2} \cdots \lambda_{j_d} > \varepsilon^2. \tag{2.1}$$

Let k be the number of indices $j_i \geq 2$, i.e., $\lambda_{j_i} < 1$. The inequality above implies

$$\lambda_2^k > \varepsilon^2, \tag{2.2}$$

and we know that $k \leq a_d(\varepsilon)$, where

$$a_d(\varepsilon) = \min\left(d, \left\lceil 2\frac{\ln \varepsilon^{-1}}{\ln \lambda_2^{-1}} \right\rceil - 1\right),$$

see [49, Thm. 5.5] for the details.

There are $\binom{d}{a_d(\varepsilon)}$ ways to select the $(d - a_d(\varepsilon))$ indices j_r that must be equal to 1, i.e., $\lambda_{j_r} = 1$, due to (2.1) and (2.2).

Let $j_{\max}$ be the largest index of the eigenvalues in (2.1). Then $\lambda_{j_{\max}} \geq \lambda_{j_1} \cdots \lambda_{j_d} > \varepsilon^2$, which implies $j_{\max} \leq n(\varepsilon, 1)$. Note that there are no more than $a(d) \leq d$ choices for the location of the largest index.

Consider now the second largest index $j'_{\max}$ of the eigenvalues in (2.1). Then we have $\lambda^2_{j'_{\max}} \geq \lambda_{j'_{\max}} \lambda_{j_{\max}} \geq \lambda_{j_1} \cdots \lambda_{j_d} > \varepsilon^2$, which implies that $\lambda_{j'_{\max}} > \varepsilon$ and so $j'_{\max} \leq n(\varepsilon^{1/2}, 1)$.

(Similarly, we see that the i-th largest index is at most $n(\varepsilon^{1/i}, 1)$.)

Thus, we estimate $n(\varepsilon, d)$ by

$$n(\varepsilon, d) \leq \binom{d}{a_d(\varepsilon)} [n(\varepsilon^{1/2}, 1)]^{a_d(\varepsilon)-1} n(\varepsilon, 1) d.$$

Taking logarithm we obtain

$$\begin{aligned}
\ln n(\varepsilon, d) &\leq \ln \left[\binom{d}{a_d(\varepsilon)} [n(\varepsilon^{1/2}, 1)]^{a_d(\varepsilon)-1} n(\varepsilon, 1) d \right] \\
&= \ln \binom{d}{a_d(\varepsilon)} + (a_d(\varepsilon) - 1) \ln n(\varepsilon^{1/2}, 1) + \ln n(\varepsilon, 1) + \ln d \\
&\leq a_d(\varepsilon) \ln d - \ln(a_d(\varepsilon)!) + a_d(\varepsilon) \ln n(\varepsilon^{1/2}, 1) + \ln n(\varepsilon, 1) + \ln d \\
&\leq a_d(\varepsilon) \ln d + a_d(\varepsilon) \ln n(\varepsilon^{1/2}, 1) + \ln n(\varepsilon, 1) + \ln d.
\end{aligned}$$

Dividing by $(\varepsilon^{-1} + d)$ and taking the limit as $\varepsilon^{-1} + d \to \infty$ yields

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln n(\varepsilon, d)}{\varepsilon^{-1} + d} \leq \lim_{\varepsilon^{-1}+d\to\infty} \left[\frac{a_d(\varepsilon) \ln d}{\varepsilon^{-1} + d} + \frac{a_d(\varepsilon) \ln[n(\varepsilon^{1/2}, 1)]}{\varepsilon^{-1} + d} + \frac{\ln n(\varepsilon, 1)}{\varepsilon^{-1} + d} + \frac{\ln d}{\varepsilon^{-1} + d} \right].$$

Using $\ln n(\varepsilon, 1) = o(\varepsilon^{-1})$ and $a_d(\varepsilon) = \Theta(\min(d, \ln \varepsilon^{-1}))$, we consider the limit of each of the four terms in the right hand side above.

The limit of the first term is zero. Indeed, as in [49], if $x = \max(d, \varepsilon^{-1})$, then

$$\min(d, \ln \varepsilon^{-1}) \leq \ln x$$

and

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\min(d, \ln \varepsilon^{-1}) \ln d}{\varepsilon^{-1} + d} \leq \lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln^2 x}{x} = 0.$$

The limit of the second term is zero since

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\min(d, \ln \varepsilon^{-1}) \cdot o(\varepsilon^{-1/2})}{\varepsilon^{-1} + d} = 0.$$

Observe that if we had $o(\varepsilon^{-1})$ instead of $o(\varepsilon^{-1/2})$ in the numerator, then for $d = \Theta(\varepsilon^{-1})$ the limit would not be zero, which was the complicating factor in [49].

For the third term, it is easy to see that

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln n(\varepsilon, 1)}{\varepsilon^{-1} + d} = \lim_{\varepsilon^{-1}+d\to\infty} \frac{o(\varepsilon^{-1})}{\varepsilon^{-1} + d} = 0.$$

Finally, the limit of the fourth term is trivially zero.

Hence,

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln n(\varepsilon, d)}{\varepsilon^{-1} + d} = 0,$$

and the problem is weakly tractable. □

This result carries over to the normalized error criterion. The information complexity in this case is

$$n(\varepsilon, d) = \left| \{[j_1, j_2, \ldots, j_d] \in \mathbb{N}^d | \prod_{k=1}^{d} \lambda_{j_k} > \varepsilon^2 \lambda_1^d \} \right|,$$

since the initial error of the optimal algorithm is $\sqrt{\lambda_{d,1}} = \lambda_1^{d/2}$. If we define $\lambda'_j = \lambda_j/\lambda_1$, we have

$$n(\varepsilon, d) = \left| \{[j_1, j_2, \ldots, j_d] \in \mathbb{N}^d | \prod_{k=1}^{d} \lambda'_{j_k} > \varepsilon^2 \} \right|.$$

This corresponds to the absolute error criterion for the univariate eigenvalues λ'_j, with $\lambda'_1 = 1$. Hence we obtain tractability conditions analogous to those for the absolute error criterion for the case $\lambda_1 = 1$. As a result the Theorem 5.6 [49] is modified to

Theorem 5.6. *Consider the linear tensor product problem $S = \{S_d\}$ for the normalized error criterion in the worst case setting and for the class Λ^{all} with $\lambda_2 > 0$.*

- *Let $\lambda_1 = \lambda_2$. Then S is intractable and for all $\varepsilon \in (0, 1)$ we have*

$$n(\varepsilon, d) \geq 2^d$$

- *Let $\lambda_2 < \lambda_1$. Then S is polynomially intractable.*

- *S is weakly tractable iff*

$$\lambda_2 < \lambda_1 \quad \text{and} \quad \lambda_{\mathrm{n}} = \mathrm{o}((\ln \mathrm{n})^{-2}) \text{ as } \mathrm{n} \to \infty.$$

Similarily as above, our result carries over to linear weighted tensor product problems in the worst case setting. Theorem 5.8 in [49] lists tractability conditions for such problems for the normalized error criterion and the class Λ^{all}. We modify it for the case where $\lambda_2 < \lambda_1$

Theorem 5.8. *Consider the linear weighted tensor product problem in the worst case setting* $S_\gamma = \{S_{d,\gamma}\}$ *for compact linear* $S_{d,\gamma} : H_{d,\gamma} \to G_d$ *defined over Hilbert spaces* $H_{d,\gamma}$ *and* G_d *with* $\lambda_2 > -0$. *We study the problem* S_γ *for the normalized error criterion and for the class* Λ^{all}. *The weight sequence* $\gamma = \{\gamma_{d,u}\}$ *satisfies*

$$\gamma_{d,\emptyset} = 1 \text{ and } \gamma_{d,u} =\in [0,1] \text{ for all non} - \text{empty } u \subseteq \{1,2,\ldots,d\}.$$

Let $n(\varepsilon, d) = n(\varepsilon, S_{d,\gamma})$ *denote the information complexity of* $S_{d,\gamma}$ *and* $\lambda_2 < \lambda_1$.

- *If there is a non-zero weight* $\gamma_{d,u}$ *for a non-empty* u *and* S_γ *is weakly tractable, then* $\lambda_n = o\left((\ln n)^{-2}\right)$ *as* $n \to \infty$.
- *If* $\lambda_n = o\left((\ln n)^{-2}\right)$ *then* S_γ *is weakly tractable for all weight sequences.*

The reader is referred to [49, Ch. 5.3] for the definition and details on weighted linear tensor product problems.

2.5 Future work

A research direction worth pursuing is to study how different tractability criteria affect tractability requirements. For example, one may require that tractable problems are not exponential in both d and any power of $\ln \varepsilon^{-1}$, as discussed in Section 1.2. Preliminary results regarding necessary and sufficient conditions for $\ln^\kappa$-weak and (strong) polylog tractability have been presented in [52].

Chapter 3

Tractability of tensor product problems in the average case setting

3.1 Introduction

In this chapter we study the complexity of linear tensor product problems in the average case setting under the absolute error criterion. More specifically, we are interested in determining whether there are cases of multivariate problems where weak tractability holds but polynomial tractability does not. This problem is also known as Open Problem 28 in [49].

A linear problem $S = \{S_d\}$ is obtained though a sequence of continuous linear operators S_d, each defined on a space of functions of $d \geq 1$ variables [49]. As we saw in Section 2.2, the tensor product structure is introduced in the worst case setting by setting

$$S_d = S_1^{\otimes d},$$

where S_1 is defined on a space of univariate functions. This construction is modified for the average case setting. For the solution operator $S_d : F_d \to G_d$, only the target space G_d needs to be a tensor product space $G_d = G_1^{\otimes d}$, where G_1 is a Hilbert space. The space F_d is equipped with a Gaussian measure that is *derived* from a given Gaussian measure on F_1. We will go over the details later.

We are interested in algorithms approximating the operator S_d using n evaluations of arbitrary linear functionals and we consider their average error. The information complexity

is the minimal number of evaluations needed to approximate S_d to within accuracy ε. To underline the dependency on ε and d, we denote the complexity by $n(\varepsilon, d)$.

We remind the reader that a problem is *polynomially* tractable iff $n(\varepsilon, d)$ grows as a polynomial in d and ε^{-1}. In particular, when $n(\varepsilon, d)$ is bounded by a quantity independent of d and polynomial in ε^{-1} the problem S is *strongly polynomially* tractable.

A problem is *weakly tractable* iff

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln n(\varepsilon, d)}{\varepsilon^{-1} + d} = 0,$$

otherwise the problem is intractable. Hence, a problem is weakly tractable if its complexity is not exponential in either ε^{-1} or d.

The complexity of linear tensor product problems in the average case setting is characterized by the eigenvalues of the covariance operator of the induced measure on the space G_1. These eigenvalues, due to the tensor product structure, determine the rate of decay of the eigenvalues of the covariance operator in the d-dimensional problem and, through them, they determine the (average) error of optimal algorithms.

3.2 Linear Tensor Product Problems in the Average Case Setting

We briefly introduce linear tensor product problems in the average case setting as defined in [49, Ch. 6].

For $d \geq 1$, let

$$S_d : F_d \to G_d,$$

be a linear operator mapping a separable Banach space F_d to a separable Hilbert space G_d. We assume the space G_d is the tensor product of d copies of a separable Hilbert space G_1, i.e., $G_d = \otimes_{i=1}^d G_1$. Thus G_d is spanned by $\otimes_{i=1}^d g_i$, $g_i \in G_1$, and has an inner product structure defined by

$$\langle \otimes_{i=1}^d g_i, \otimes_{i=1}^d h_i \rangle_{G_d} = \prod_{i=1}^d \langle g_i, h_i \rangle_{G_1} \quad \text{for} \quad g_i, h_i \in G_1.$$

Hence,

$$S_d f = \sum_{j \in \mathbb{N}^d} \langle S_d f, \eta_{d,j} \rangle_{G_d} \eta_{d,j} \quad \text{for } f \in F_d,$$

where

$$\eta_{d,j} = \otimes_{k=1}^{d} \eta_{j_k} \quad j = [j_1, j_2, \ldots, j_d] \in \mathbb{N}^d, \tag{3.1}$$

and $\{\eta_i\}_{i \in \mathbb{N}}$ is an orthonormal system in G_1.

Consider a zero-mean Gaussian measure μ_d on F_d with $\int_{F_d} \|S_d f\|_{G_d}^2 \, \mu_d(df) < \infty$. Let $\nu_d = \mu_d S_d^{-1}$ be the induced measure on G_d, which is also a zero-mean Gaussian measure. Let C_{ν_d} denote the covariance operator of ν_d and let $(\lambda_{d,j}, \eta_{d,j})$, $j \in \mathbb{N}^d$, be its eigenvalues and the corresponding eigenvectors.

We also assume that the eigenvalues satisfy the conditions below, to preserve the tensor product structure of G_d and its orthonormal system $\{\eta_{d,j}\}_{j \in \mathbb{N}^d}$. For $d = 1$, we have $\lambda_{1,j} = \lambda_j$, with $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ and

$$\sum_{j=1}^{d} \lambda_j = \text{trace}(C_{\nu_1}) < \infty.$$

For $d \geq 1$, we assume

$$\lambda_{d,j} = \Pi_{k=1}^{d} \lambda_{j_k} \text{ for all } j = [j_1, j_2, \ldots, j_d] \in \mathbb{N}^d, \tag{3.2}$$

and

$$\text{trace}(C_{\nu_d}) = \sum_{j \in \mathbb{N}^d} \lambda_{d,j} = \left(\sum_{i=1}^{\infty} \lambda_j \right)^d.$$

A *linear tensor product problem* in the *average case setting* is the multivariate problem $S = \{S_d\}$ with the eigenpairs of the covariance operator C_{ν_d} satisfying the conditions (3.1) and (3.2).

For notational convenience, let us now reindex the eigenvalues and eigenvectors to obtain $\{\lambda_{d,j}\}_{j \in \mathbb{N}^d} = \{\lambda_{d,i}\}_{i \in \mathbb{N}}$ and $\{\eta_{d,j}\}_{j \in \mathbb{N}^d} = \{\eta_{d,i}\}_{i \in \mathbb{N}}$, respectively. Also assume the eigenvalues are ordered, so that $\lambda_{d,1} \geq \lambda_{d,2} \geq \cdots \geq 0$.

Suppose that we can use arbitrary linear functionals on F_d as information operations, i.e., we can use functionals from the class Λ^{all}, as denoted in [49; 66]. Then it is known, see e.g. [66], that the algorithm

$$A_{d,n}(f) = \sum_{i=1}^{n} \langle S_d f, \eta_{d,i} \rangle_{G_d} \eta_{d,i} \tag{3.3}$$

minimizes the average error

$$e(A_{d,n}) = \left(\int_{F_1} \|S_d f - A_{d,n}(f)\|_{G_d}^2 \, \mu(df) \right)^{1/2},$$

among all possible algorithms using at most n information operations. It is also known that the error of this optimal algorithm is obtained from the truncated trace of C_{ν_d}, with

$$e(A_{d,n}) = \left(\sum_{i=n+1}^{\infty} \lambda_{d,i} \right)^{1/2}. \tag{3.4}$$

The information complexity of the problem S_d for accuracy ε with the absolute error criterion is the minimal number of information operations needed to guarantee that the average case error is at most ε, and is given by

$$n(\varepsilon, d) = \min \left\{ n : \sum_{i=n+1}^{\infty} \lambda_{d,i} \le \varepsilon^2 \right\}.$$

3.3 Prior work

Linear tensor product problems in the average case setting are discussed in [49, Ch. 6]. We briefly review some of the results, which motivate Open Problem 28 in the book [49]. Additional details can be found in [49, Th. 6.5, Th. 6.6].

Recall that we deal only with the absolute error criterion, since linear tensor product problems are intractable in the average case setting with the normalized error criterion for $\lambda_2 > 0$.

If the one-dimensional eigenvalues satisfy $\sum_{j=1}^{\infty} \lambda_j \ge 1$ then the linear tensor product problem $S = \{S_d\}$ is intractable. From this point on we consider the case

$$\sum_{j=1}^{\infty} \lambda_j < 1, \qquad \lambda_2 > 0.$$

Then the following are equivalent:

1. S is polynomially tractable.

2. S is strongly polynomially tractable.

3. There exists a $\tau \in (0,1)$ such that $\sum_{j=1}^{\infty} \lambda_j^{\tau} \le 1$.

Moreover, if $\lambda_j = O(j^{-p})$ with $p > 1$, the following are equivalent:

1. S is weakly tractable.
2. S is polynomially tractable.
3. S is strongly polynomially tractable.
4. $\sum_{j=1}^{\infty} \lambda_j < 1$.

Combining the above with

$$\sum_{j=1}^{\infty} \lambda_j^{\tau} \leq 1 \quad \text{for} \quad \tau \in (0,1) \quad \text{iff} \quad \sum_{j=1}^{\infty} \lambda_j < 1 \quad \text{and} \quad \lambda_j = \mathcal{O}(j^{-p}) \quad \text{for} \quad p > 1,$$

(see, [49, p. 258] for the proof) we conclude that the only possibility for a linear tensor product problem to be weakly tractable but not polynomially tractable is if

$$\lambda_j = O\left(\frac{1}{j \ln^q(j+1)}\right) \quad q > 1.$$

We are interested in finding for which q the linear tensor product problem S is weakly tractable but not polynomially tractable. This is the Open Problem 28 in [49].

3.4 Tractability

We are interested in estimating the information complexity $n(\varepsilon, d)$. Let $a = \sum_{j=1}^{\infty} \lambda_j$. From (3.2) and $\sum_{j \in \mathbb{N}^d} \lambda_{d,j} = \left(\sum_{j=1}^{\infty} \lambda_j\right)^d$, the error of the zero algorithm, which does not use any information at all, is $a^{d/2}$. Hence, the only remaining case to be studied is when the required accuracy satisfies $\varepsilon^2 < a^d$.

Lemma 1. *Consider the eigenvectors of C_{ν_d} given by*

$$\eta_{d,j} = \eta_{j_1} \otimes \cdots \otimes \eta_{j_d},$$

where $j = [j_1, j_2, \ldots, j_d]$, for $j_k = 1, \ldots, m$, and $k = 1, \ldots, d$. The average error of the algorithm

$$\phi_{d,m^d}(f) = \sum_{j_1,\ldots,j_d=1}^{m} \langle S_d(f), \eta_{d,j} \rangle \eta_{d,j}$$

is bounded from above by

$$e^2(\phi_{d,m^d}) \leq d\, a^{d-1}\, t_m,$$

where $t_m = \sum_{j=m+1}^{\infty} \lambda_j$.

Proof. The error of ϕ_{d,m^d} satisfies

$$\begin{aligned}
e^2(\phi_{d,m^d}) &= \sum_{j_1,\ldots j_d \geq 1} \lambda_{j_1} \ldots \lambda_{i_d} - \sum_{j_1,\ldots j_d=1}^{m} \lambda_{j_1} \ldots \lambda_{i_d} \\
&= \sum_{j_1>m, j_2,\ldots,j_d \geq 1} \lambda_{j_1} \ldots \lambda_{j_d} + \sum_{j_1 \leq m, j_2,\ldots,j_d \geq 1} \lambda_{j_1} \ldots \lambda_{j_d} - \sum_{j_1,\ldots j_d=1}^{m} \lambda_{j_1} \ldots \lambda_{j_d} \\
&= t_m a^{d-1} + \sum_{j_1 \leq m, j_2,\ldots,j_d \geq 1} \lambda_{j_1} \ldots \lambda_{j_d} - \sum_{j_1,\ldots j_d=1}^{m} \lambda_{j_1} \ldots \lambda_{j_d} \\
&\leq 2t_m a^{d-1} + \sum_{j_1,j_2 \leq m, j_3,\ldots,j_d \geq 1} \lambda_{j_1} \ldots \lambda_{j_d} - \sum_{j_1,\ldots j_d=1}^{m} \lambda_{j_1} \ldots \lambda_{j_d} \\
&\vdots \\
&\leq da^{d-1} t_m.
\end{aligned}$$

We remark that the algorithm ϕ_{d,m^d} minimizes the average error among all algorithms that use the information $\langle S_d(f), \eta_{d,j} \rangle$ although this information is not optimal, in general. The reason is that the eigenvectors $\eta_{d,j}$ do not correspond to the m^d largest eigenvalues. Hence, if m is large enough and ϕ_{d,m^d} satisfies the accuracy demand ε then m^d is an upper bound of $n(\varepsilon, d)$. □

Theorem 2. *Consider the linear tensor product problem* $S = \{S_d\}$ *in the average case setting with* $\sum_{j=1}^{\infty} \lambda_j < 1$, $\lambda_2 > 0$, *for the absolute error criterion and the class of* Λ^{all}.

- *S is weakly tractable iff*

$$t_n = \sum_{j>n} \lambda_j = o\left(\frac{1}{\ln^2(n+1)}\right).$$

- *Suppose that*

$$\ell = \lim_{j\to\infty} \lambda_j j \ln^3(j+1)$$

exists. Then S is weakly tractable iff $\ell = 0$.

Proof. We begin by showing that

$$t_n = \sum_{j>n} \lambda_j = o\left(\frac{1}{\ln^2(n+1)}\right)$$

is a sufficient condition for weak tractability. Let ε^{-1} and/or d be sufficiently large. The error of the algorithm ϕ_{d,m^d} of Lemma 1 satisfies

$$e^2(\phi_{d,m^d}) \le da^{d-1} t_m = da^{d-1} \frac{s_m}{\ln^2(m+1)},$$

where $s_m = o(1)$.

Let $m = m(\varepsilon, d)$ be the smallest integer such that

$$e^2(\phi_{d,m^d}) \le da^{d-1} \frac{s_m}{\ln^2 m} \le \varepsilon^2 < a^d.$$

Then $m \to \infty$ as $\varepsilon \to 0$ and/or $d \to \infty$. Clearly $n(d,\varepsilon) \le m^d$ and

$$\ln m \ge (da^{d-1} s_m)^{1/2} \varepsilon^{-1}.$$

By definition of $m(\varepsilon, d)$, there exists a constant c such that

$$\ln m(\varepsilon, d) \le c\,(da^{d-1} s_{m(\varepsilon,d)-1})^{1/2} \varepsilon^{-1}.$$

Hence,

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln n(\varepsilon,d)}{\varepsilon^{-1}+d} \le \lim_{\varepsilon^{-1}+d\to\infty} \frac{d \ln m(\varepsilon,d)}{\varepsilon^{-1}+d} = \lim_{\varepsilon^{-1}+d\to\infty} \frac{c\, d^{3/2} \left[a^{d-1} s_{m(\varepsilon,d)-1}\right]^{1/2} \varepsilon^{-1}}{\varepsilon^{-1}+d} = 0.$$

On the other hand, it is relatively easy to show that

$$t_n = \sum_{j>n} \lambda_j = o\left(\frac{1}{\ln^2(n+1)}\right),$$

is a necessary condition for weak tractability. One can use the same proof used in [49, p. 178] for the worst case. For completeness, we include it here. Assume S is weakly tractable, i.e.,

$$\lim_{\varepsilon^{-1}+d\to\infty} \frac{\ln n(\varepsilon,d)}{\varepsilon^{-1}+d} = 0.$$

Setting $d = 1$, we get $\left(\varepsilon^{-1}+1\right)^{-1} = o(\ln^{-1} n(\varepsilon,1))$ as $\varepsilon \to 0$, so that, $\varepsilon = o(\ln^{-1} n(\varepsilon,1))$. Also

$$\varepsilon^2 \ge e^2(A_{1,n(\varepsilon,1)}) \ge t_{n(\varepsilon,1)}.$$

So

$$t_n = o(\ln^{-2}(n+1)).$$

This completes the proof of the first part of the theorem.

For the second part of the theorem it is easy to see that $\ell = 0$ is a necessary condition for weak tractability. Indeed, let $d = 1$ and let ε be sufficiently small. Assume that the problem S is weakly tractable and that there exists a constant c such that $\ell = \lim_{j\to\infty} \lambda_j j \ln^3(j+1) \geq c > 0$. Then

$$t_n = e^2(A_{1,n}) \geq c \sum_{j>n} \frac{1}{j \ln^3(j+1)} = \Omega\left(\frac{1}{\ln^2(n+2)}\right)$$

and we have a contradiction.

We now show that the condition $\ell = 0$ is sufficient for weak tractability. Let

$$\lambda_j = \frac{g(j)}{j \ln^3(j+1)}.$$

Since $\ell = 0$ we have $g(j) = o(1)$. Let ε^{-1} and/or d be sufficiently large. We have

$$t_n = \sum_{j>n} \frac{g(j)}{j \ln^3(j+1)} \leq \frac{s_n}{\ln^2(n+1)},$$

where $s_n = \sup_{j>n} g(j) = o(1)$. Hence $t_n = o(\ln^{-2}(n+1))$, and the first part of the theorem yields that S is weakly tractable. □

Remark. In the second part of Theorem 2 we assumed that the limit of $\lambda_j j \ln^3(j+1)$ exists as $j \to \infty$ and we showed a necessary and sufficient condition for weak tractability. On the other hand, if this limit does not exist the problem may still be weakly tractable.

Indeed, the condition $t_n = \sum_{j>n} \lambda_j = o(\ln^{-2}(n+1))$, implies that $n\lambda_{2n} = o(\ln^{-2}(n+1))$. Therefore,

$$\lambda_n = o\left(\frac{1}{n \ln^2(n+1)}\right)$$

is a necessary condition for weak tractability. Moreover, proceeding in a way similar to that in the proof of Theorem 2, we can show a second necessary condition, namely

$$\liminf_{n\to\infty} \lambda_n n \ln^3(n+1) = 0.$$

It is interesting to observe that as long as the slower converging subsequence of eigenvalues does not contribute excessively to t_n the problem can be weakly tractable. We illustrate this by an example.

Let k_0 be a sufficiently large integer. For $k = k_0, k_0+1, \ldots$ let $j = \lceil e^{k^2} \rceil$, and

$$\lambda_i = \frac{1}{j \ln^{3+\gamma}(j+1)} \quad i = j+1, \ldots, \lceil j + j\ln(j+1) \rceil,$$

with $\gamma \in (1/2, 1)$. So we have a segment of $\lceil j\ln(j+1) \rceil$ eigenvalues that are equal and the first eigenvalue in the segment, λ_j, goes to zero faster than the last $\lambda_{\lceil j+j\ln(j+1)\rceil}$. Furthermore, since k_0 is large enough the segments are disjoint. We define the remaining eigenvalues by

$$\lambda_j = \frac{1}{j \ln^{3+\gamma}(j+1)}.$$

Hence, $\lambda_n = o(n^{-1}\ln^{-2}(n+1))$ and $\liminf_{n\to\infty} \lambda_n n \ln^3(n+1) = 0$. However

$$\limsup_{n\to\infty} \lambda_n n \ln^3(n+1) = \infty$$

since $\gamma < 1$. Thus the limit ℓ of Theorem 2 does not exist. Nevertheless, S is weakly tractable.

Indeed,

$$\sum_{i=j+1}^{\lceil j+j\ln(j+1)\rceil} \lambda_i \le c' \frac{1}{\ln^{2+\gamma}(j+1)},$$

where c' is an absolute constant. The contribution of all such segments starting at $j = \lceil e^{k^2} \rceil$, where $k \in \mathbb{N}$, to t_n is

$$c' \sum_{j=\lceil e^{k^2}\rceil > n,\ k\in\mathbb{N}} \frac{1}{\ln^{2+\gamma}(j+1)} \le c' \frac{1}{\ln^{2+\gamma}(n+1)} + c'' \int_{x^2 > \ln n} \frac{dx}{x^{2(2+\gamma)}} = o\left(\frac{1}{\ln^2(n+1)}\right),$$

where c'' is an absolute constant and the last equality holds since $\gamma > 1/2$. It is easy to see that the contribution to t_n of the remaining eigenvalues is also $o(\ln^{-2}(n+1))$. Since $t_n = o(\ln^{-2}(n+1))$ the problem S is weakly tractable, as claimed.

Finally, it is relatively easy to see that a problem can be weakly tractable even though it is not polynomially tractable. We state this fact in the following corollaries.

Corollary 1. *Consider the linear tensor product problem $S = \{S_d\}$ in the average case setting with $\sum_{j=1}^{\infty} \lambda_j < 1$ for the absolute error criterion and the class of $\Lambda^{\rm all}$. If $\lambda_j = \Theta\left((j\ln^q(j+1))^{-1}\right)$ then the problem is weakly tractable if and only if $q > 3$.*

Proof. This follows directly from Theorem 2. □

Corollary 2. *Consider the linear tensor product problem $S = \{S_d\}$ in the average case setting with $\sum_{j=1}^{\infty} \lambda_j < 1$ and $\lambda_2 > 0$ for the absolute error criterion and the class of Λ^{all}. Then S is weakly tractable but not polynomially tractable iff*

$$t_n = \sum_{j>n} \lambda_j = o\left(\frac{1}{\ln^2(n+1)}\right).$$

and

$$\limsup_{j\to\infty} \lambda_j j^p = \infty \quad \text{for all } p > 1.$$

Proof. The proof is immediate from Theorem 2 and [49, Th. 6.7]. □

3.5 Future work

A research direction worth pursuing is to examine how other tractability criteria affect the tractability requirements for linear tensor product problems in the average case setting. One, for example, may study tractability conditions for $\ln^{\kappa}$-weak tractability or (strong) polylog tractabilityfor linear and linear tensor product problems.

Part II

Contributions to Quantum Computing

Chapter 4

A fast algorithm for approximating the ground state energy on a quantum computer

4.1 Introduction

In this chapter and the following one, we study algorithms estimating the ground state energy of a time-independent Hamiltonian corresponding to a multiparticle system; see Section 1.3.1 for details on the time-independent Schrödinger equation. More specifically, we consider the time-independent Schrödinger equation for p particles in the d dimensional unit cube with Dirichlet boundary conditions If the potential is a function of only state variables then the ground state energy is given by the smallest eigenvalue E_0 of the equation

$$\begin{aligned} (-\tfrac{1}{2}\Delta + V)\psi_0(x) &= E_0\psi_0(x) \quad \text{for all } \ x \in I_d := (0,1)^d, \\ \psi_o(x) &= 0 \quad \text{for all } \ x \in \partial I_d, \end{aligned}$$

where ψ_0 is a normalized eigenfunction and E_0 denotes the ground state energy, which is the smallest eigenvalue of the Hamiltonian H. For simplicity we assume that all masses and the normalized Planck constant are one.

This eigenvalue problem is called the time-independent Schrödinger equation in the physics literature and the Sturm-Liouville eigenvalue problem in the mathematics literature. We want

to approximate E_0 with relative error ε.

Here, Δ is the d-dimensional Laplacian and $V \geq 0$ is a function of d variables. The dimension is proportional to the number of particles, e.g. $d = 3p$. For many applications the number of particles p and hence d is huge. We consider algorithms that approximate E_0 using finitely many function evaluations of V. Moreover, we assume that V and its first order partial derivatives $\partial V/\partial x_j$, $j = 1, \ldots, d$, are continuous and uniformly bounded by 1.

Such eigenvalue problems can be solved by discretizing the Hamiltonian operator and approximating the corresponding matrix eigenvalue. Eigenvalue problems involving symmetric matrices are conceptually easy and methods such as the bisection method can be used to solve them with cost proportional to the matrix size, modulo polylog factors. The problem with the time independent Schrödinger equation, especially the one corresponding to a multi-particle system, is that the discretization leads to a matrix of dimension that grows exponentially in d. Thus the resulting cost is prohibitive when d is large. Moreover, as we will see later for the potentials considered here, it can be shown that any deterministic classical algorithm must have cost that grows exponentially in d, i.e. the problem suffers from the curse of dimensionality.

On the other hand, in certain cases, quantum algorithms may be able to compute accurate eigenvalue estimates with cost that does not grow exponentially in d, even though any classical deterministic algorithm in the worst case suffers from the curse of dimensionality. This is illustrated later in this chapter and in [54], where we show that if the potential is smooth, nonnegative and uniformly bounded by a relatively small constant then there exists a quantum algorithm approximating the ground state energy with relative error ε and cost proportional to $d\varepsilon^{-(3+\eta)}$, where η is an arbitrary positive constant.

We stress that we are not dealing with an arbitrary eigenvalue problem. Estimating the ground state energy of general local Hamiltonians is a QMA complete problem [40]. QMA is the quantum analogue of the complexity class NP which is equipped with deterministic

verifiers, or the complexity class MA which is equipped with probabilistic verifiers[1].

4.2 Phase estimation

Quantum algorithms are a sequence of unitary transformations; see Section 1.3.2 for details. The eigenvalues $\{\lambda_j\}_{j=0,1,\ldots,N-1}$ of any $N \times N$ unitary matrix W can be expressed in the form $\lambda_j = e^{2\pi i \phi_j}$ for $\phi_j \in [0, 1)$. Phase estimation is a quantum algorithm that approximates the phase ϕ_j of an eigenvalue λ_j of the unitary matrix W. It requires the implementation of the corresponding eigenvector $|u_j\rangle$ of W and the implementation of matrix exponentials W^{2^t}, for $t = 0, 1, \ldots, b-1$, where b is a parameter related to the accuracy requirement and the probability of success of the algorithm. The circuit that implements phase estimation is presented in the following figure.

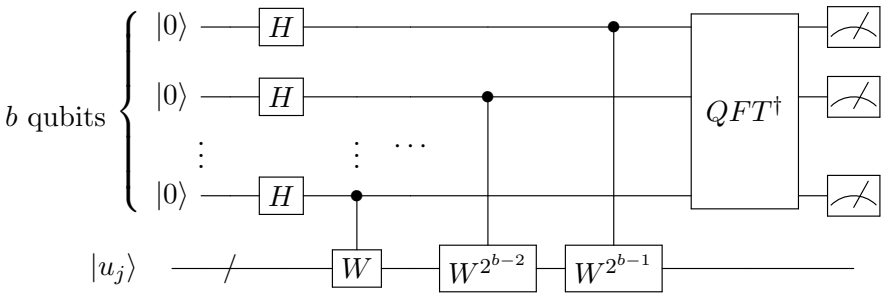


Figure 4.1: Quantum circuit implementing the phase estimation procedure. We assume that the eigenstate $|u_j\rangle$ and the controlled exponentials W^{2^t} are implemented exactly, for $t = 0, 1, \ldots, b-1$.

In many problems the eigenvector $|u_j\rangle$ and/or the matrix exponentials cannot be implemented exactly. We typically deal with this problem by constructing approximations to the eigenvector and/or the matrix exponentials. This results in decreased probability of success for the phase estimation procedure.

More details on phase estimation can be found in [47, pg. 221–226] and [1; 37].

[1]The relation of QMA to other complexity classes is depicted in the following inclusions

$$\text{P} \subseteq \text{NP} \subseteq \text{MA} \subseteq \text{QMA} \subseteq \text{PSPACE}$$

4.3 Discretization error

The finite difference method is frequently used to discretize partial differential equations, and approximate their solutions. The method with mesh size $h = (m+1)^{-1}$ yields an $m^d \times m^d$ matrix $M_h = -\frac{1}{2}\Delta_h + V_h$, where Δ_h denotes the discretized Laplacian and V_h the diagonal matrix whose entries are the evaluations of the potential on a regular grid with mesh size h.

M_h is a symmetric positive definite and sparse matrix. For a potential function V that has bounded first order partial derivatives, we have [69; 70]

$$|E_0 - E_{h,0}| \leq c_1 dh, \tag{4.1}$$

where $E_{h,0}$ is the smallest eigenvalue of M_h. Consider $\hat{E}_{h,0}$ such that

$$|E_{h,0} - \hat{E}_{h,0}| \leq c_2 dh. \tag{4.2}$$

Then we have $|1 - \hat{E}_{h,0}/E_0| \leq C'h$. This inequality follows by observing that $E_0 \geq d\pi^2/2$, for any $V \geq 0$.

4.4 Prior work

The ground state eigenvalue problem has been extensively studied in [50; 55], where upper and lower bounds on the complexity of the problem under the absolute error criterion have been presented.

For the univariate version of the problem it is known [55] that

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

in the deterministic worst case setting, while

$$n^{\text{rand}}(\varepsilon, 1) = \Theta(\varepsilon^{-2/5})$$

in the randomized setting. In the quantum setting there are two different types of information operators (queries) we can use for this problem, the *bit query* (see (1.7)) and the *power query*. The power queries refer to powers of an exponential of the form e^{izM_h}, where M_h is the

Hermitian matrix obtained from the discretization of the Hamiltonian operator and $z \in \mathbb{R}$ is a coefficient. We can approximate the eigenvalue using

$$n^{\text{bit query}}(\varepsilon, 1) = \Theta(\varepsilon^{-1/3})$$

bit queries or

$$n^{\text{power query}}(\varepsilon, 1) = \Theta(\log \varepsilon^{-1})$$

power queries [55]. The number of power queries is optimal [11].

Let us now consider V and a perturbation $\overline{V}$. Then the eigenvalues $E_0(V)$ and $E_0(\overline{V})$ are related according to

$$E_0(V) = E_0(\overline{V}) + \int_{I_d} (V(x) - \overline{V}(x))\psi_1^2(x; \overline{V})dx + O(\|V - \overline{V}\|_\infty^2), \tag{4.3}$$

where $\psi_1(\cdot; \overline{V})$ denotes the eigenfunction corresponding to $E_0(\overline{V})$, see [50].

For the multivariate version of the problem, equation (4.3) relates the eigenvalue problems to multivariate integration. Using lower bounds on multivariate integration in the worst case, randomized setting [66] and quantum setting [48] the information complexity lower bounds are

$$\begin{aligned} n^{\text{worst}}(\varepsilon, d) &= \Omega(\varepsilon^{-d}) \\ n^{\text{rand}}(\varepsilon, d) &= \Omega\left(\varepsilon^{-2d/(d+2)}\right) \\ n^{\text{bit queries}}(\varepsilon, d) &= \Omega\left(\varepsilon^{-d/(d+1)}\right) \\ n^{\text{power queries}}(\varepsilon, d) &= \Omega(\log \varepsilon^{-1}) \end{aligned}$$

Upper bounds on the information and computational complexity can be obtained in the worst case by discretizing the problem and approximating the eigenvalue using the bisection method [72]

$$\begin{aligned} n^{\text{worst}}(\varepsilon, d) &= O(\varepsilon^{-d}) \\ \text{comp}^{\text{worst}}(\varepsilon, d) &= O(c\varepsilon^{-d} + \varepsilon^{-d} \log \varepsilon^{-1}) \end{aligned}$$

where c depends on d, see [50].

For the randomized setting, equation (4.3) can be used to derive upper bounds on both the information and the computational complexity. Once the problem is discretized, Monte Carlo is employed to approximate the weighted integral. Thus

$$\begin{aligned} n^{\text{rand}}(\varepsilon, d) &= O(\varepsilon^{-\max(2/3, d/2)}) \\ \text{comp}^{\text{rand}}(\varepsilon, d) &= O(c\varepsilon^{-\max(2/3, d/2)} + \varepsilon^{-d} \log \varepsilon^{-1}) \end{aligned}$$

Observe that the algorithm is optimal only when $d \leq 2$. It is an open problem whether it is optimal when $d > 2$.

A quantum algorithm can be derived modifying the algorithm for the randomized case. In order to approximate the integral in (4.3) amplitude amplification [28] is used in the place of Monte Carlo. The algorithm approximates E_1 with accuracy ε and probability at least $3/4$ using $O(\varepsilon^{-d/2})$ classical function evaluations, $O(\varepsilon^{-d/2})$ bit queries, $O(d^2 \log^2 \varepsilon^{-1})$ quantum operations and $O(\varepsilon^{-2d} \log \varepsilon^{-1})$ classical arithmetic operations.

The upper bound in the quantum setting can be significantly improved using a phased estimation based quantum algorithm. More specifically, this algorithm uses

$$n^{\text{bit queries}}(\varepsilon, d) = O(\varepsilon^{-6} \log^2 \varepsilon^{-1})$$

bit queries, $O(d \log \varepsilon^{-1})$ qubits and $O(d\varepsilon^{-6} \log^4 \varepsilon^{-1})$ other quantum operations. Alternatively, the algorithm uses

$$n^{\text{power queries}}(\varepsilon, d) = O(\log \varepsilon^{-1})$$

power queries and $O(\log^2 \varepsilon^{-1} + d \log \varepsilon^{-1})$ other quantum operations.

4.5 Quantum algorithm

We assume that $\varepsilon < 2/(d\pi^2)$ since otherwise we can approximate the smallest eigenvalue with relative error ε with constant cost. Indeed, for V uniformly bounded by one, the smallest eigenvalue $E_1(V)$ satisfies $E_0(0) \leq E_0(V) \leq E_0(0) + 1$, where $E_0(0) = \frac{1}{2} d\pi^2$ is the smallest eigenvalue of $-\frac{1}{2}\Delta$. Thus,

$$\frac{|E_0(V) - E_0(0)|}{E_0(V)} \leq \frac{1}{E_0(0)} = \frac{2}{d\pi^2}.$$

Therefore it suffices to deal only with the case $\varepsilon < 2/(d\pi^2)$.

First we discuss our algorithm in general terms and then we provide a complete analysis. 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. First we provide a high level description of the algorithm and then give all its details and the resulting error and cost estimates.

Sketch of the algorithm:

1. Consider the discretization $M_h = -\frac{1}{2}\Delta_h + V_h$ of $-\frac{1}{2}\Delta + V$ and choose $h \leq \varepsilon$ leading to 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 $e^{iE_{h,0}/(2d)}$ with the following modifications:

   (a) Use the approximate eigenvector

$$|0\rangle^{\otimes b}|\psi_0\rangle^{\otimes d}$$

   as an initial state, where $|\psi_0\rangle^{\otimes d}$ is the ground state eigenvector of $-\Delta_h$ and can be implemented efficiently; see the discussion following (4.6) below for details.

   (b) For $t = 0, \ldots, b-1$, replace W^{2^t} 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 and can be implemented efficiently; see the discussion leading to (4.9) below for details.

The effect of the modifications is to somewhat decrease the success probability while increasing the cost of phase estimation. Nevertheless, the resulting success probability is at least $\frac{2}{3}$, and the cost for implementing the initial state and the approximate powers of W does not suffer from the curse of dimensionality. (The actual value of the success probability

is not important since it exceeds $\frac{1}{2}$ and can be boosted to become arbitrarily close to one; see [47, pg. 153] for details.)

Theorem 3. *Phase estimation with an approximate initial state and approximate powers of W with probability at least $\frac{2}{3}$ yields an estimate of E_0 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. The pseudocode for the algorithm is listed in Algorithm 1.

Next we discuss the details of our algorithm, which will lead us to the proof of the theorem. Let $h = (m+1)^{-1}$, where $m = 2^{\lceil -\log_2 \varepsilon\rceil} - 1$. Clearly, $h \le \varepsilon < 2/(d\pi^2) < 1/4$ due to our assumption at the beginning of this section. This leads to the desired accuracy while ensuring the discretization is not trivial. The eigenvalue of W that corresponds to $E_{h,0}$ is $e^{iE_{h,0}/(2d)} = e^{2\pi i\varphi_0}$, where

$$\varphi_0 = E_{h,0}/(4\pi d)$$

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

Quantum phase estimation approximates the phase φ_0 with b-bit accuracy, where $b = \lceil -\log_2\varepsilon\rceil$. The output of the algorithm is an index $j \in [0, 2^b - 1]$ such that $|\varphi_0 - j\,2^{-b}| \le 2^{-b}$. Hence,

$$|E_{h,0} - 4\pi dj\,2^{-b}| \le c_2 d\varepsilon. \tag{4.4}$$

Combining (4.1) and (4.4) we conclude

$$|E_0 - 4\pi dj\,2^{-b}| \le c_1 d\varepsilon + c_2 d\varepsilon = cd\varepsilon. \tag{4.5}$$

Hence the algorithm approximates the ground state eigenvalue E_0 by

$$\widehat{E}_{h,0} := 4\pi dj\,2^{-b}.$$

The estimate $\widehat{E}_{h,0}$ holds with probability at least $8/\pi^2$ (see, e.g., [28]) assuming:

- The initial state of the algorithm is $|0\rangle^{\otimes b}|z_{h,0}\rangle$, where $|z_{h,0}\rangle$ is the eigenvector of M_h that corresponds to $E_{h,0}$.

Algorithm 1 GroundStateEnergy(ε, d, V)

Require: d to be a positive integer.

Require: $\varepsilon \in (0, 2/\pi^2 d)$. Note that for relative error $\varepsilon \geq 2/\pi^2 d$ the problem can be solved with constant cost.

Require: $V : [0,1]^d \to [0,1]$ to be provided by an oracle (black box).

1: $b \leftarrow \lceil -\log_2 \varepsilon \rceil$

2: $m \leftarrow 2^b - 1$

3: $h \leftarrow (m+1)^{-1}$

4: Initial state : $|0\rangle^{\otimes b}|\psi_0\rangle^{\otimes d}$ {The right register holds the eigenvector of the discretized Laplacian, with mesh size h. The corrdinates of $|\psi_0\rangle$ are given in equation (4.7)}

5: ApproxW(b, h, m, d, V, $\widetilde{W}$) {This subroutine call returns $\widetilde{W}$ which is a list of the approximations of the exponentials $W^{2^t} = e^{(-\frac{1}{2}\Delta_h + V_h)2^t/(2d)}$, $t = 0, \ldots b-1$. These approximations are denoted by $\widetilde{W}_t$, $t = 0, \ldots, b-1$; see Algorithm 2 for details.}

6: Apply Hadamard to the left b-qubit register:

$$\left(H^{\otimes d} \otimes I\right)|0\rangle^{\otimes d}|\psi_0\rangle^{\otimes d} \to \frac{1}{2^{b/2}} \sum_{i_0, i_1, \ldots, i_{b-1}=0}^{1} |i_{b-1} i_{b-2} \cdots i_1 i_0\rangle |\psi_0\rangle^{\otimes d}$$

7: Apply $\widetilde{W}_0 \ldots \widetilde{W}_{b-1}$, controlled by the left b qubits, to the state above:

$$\to \frac{1}{2^{b/2}} \sum_{i_0, i_1, \ldots, i_{b-1}=0}^{1} |i_{b-1} j_{b-2} \cdots i_1 i_0\rangle \widetilde{W}_{b-1}^{i_{b-1}} \cdots \widetilde{W}_1^{i_1} \widetilde{W}_0^{i_0} |\psi_0\rangle^{\otimes d}$$

8: Apply the inverse Fourier transform $FT^\dagger$ to the register holding the leftmost b qubits:

$$\to (FT^\dagger \otimes I) \left(\frac{1}{2^{b/2}} \sum_{i_0, i_1, \ldots, i_{b-1}=0}^{1} |i_{b-1} i_{b-2} \cdots i_1 i_0\rangle \widetilde{W}_{b-1}^{i_{b-1}} \cdots \widetilde{W}_1^{i_1} \widetilde{W}_0^{i_0} |\psi_0\rangle^{\otimes d} \right)$$

9: Measure the first b qubits in the computational basis: outcome $(j_{b-1}, \ldots, j_1, j_0)$

10: $j \leftarrow \sum_{k=0}^{b-1} j_k 2^k$

11: $\widehat{E}_{h,0} \leftarrow 4\pi d j / 2^b$

12: **return** $\widehat{E}_{h,0}$

Algorithm 2 ApproxW(d, b, h, m, V, $\widetilde{W}$)

Require: d, b, m are positive integers; m and b are defined in Algorithm 1.

Require: $V : [0,1]^d \to [0,1]$ to be provided by an oracle (black box).

Require: $\widetilde{W}$ to be a list where this subroutine will hold the approximations $\widetilde{W}_t$ for $t = 0, \ldots, b-1$, that it computes. This list is returned to the calling program.

Let V_h be the $m^d \times m^d$ diagonal matrix obtained by discretizing the function V on a grid with mesh size $h = (m+1)^{-1}$.

$\mathrm{Norm}_1 \leftarrow h^{-2} \sin^2(\pi m/(2(m+1)))$ {The norm of $H_1 = -\Delta_h/(4d)$.}

$\mathrm{Norm}_2 \leftarrow 1/(2d)$ {Upper bound of the norm of $H_2 = V_h/(2d)$.}

$k \leftarrow \left\lfloor \sqrt{\frac{1}{2}\log_{25/3}(80\, e\, 2^b/d)} + \frac{1}{2} \right\rfloor$ {Note $k \geq 1$ by definition of b.}

$c_k \leftarrow \frac{8}{3} k \left(\frac{5}{3}\right)^{k-1}$ {See also [56, Eq. 7].}

$\mathcal{H}_1 = -\Delta_h/(4d \cdot \mathrm{Norm}_1)$

$\mathcal{H}_2 = V_h/(2d \cdot \mathrm{Norm}_1)$

for $t = 0$ to $b-1$ **do**

$\varepsilon_t \leftarrow 2^{t+1-b}/40$

$M \leftarrow \left(\frac{8\, e\, 2^t\, \mathrm{Norm}_2}{\varepsilon_t}\right)^{1/(2k)} \frac{2\, e\, c_k}{2k+1}$

$\mathrm{NumberOfIntervals} \leftarrow \lceil M\ \mathrm{Norm}_1\ 2^t \rceil$

$\mathrm{IntervalSize} \leftarrow \mathrm{Norm}_1 \cdot 2^t / \mathrm{NumberOfIntervals}$

$p_k \leftarrow (4 - 4^{1/2k-1})^{-1}$

$S_2(\mathrm{IntervalSize}) \leftarrow e^{-i\mathcal{H}_1 \mathrm{IntervalSize}/2} e^{-i\mathcal{H}_2 \mathrm{IntervalSize}} e^{-i\mathcal{H}_1 \mathrm{IntervalSize}/2}$

for $j = 2$ to k **do**

Let

$$\begin{aligned} S_{2j}(\mathrm{IntervalSize}) \;&\leftarrow\; [S_{2j-2}(p_k \mathrm{IntervalSize})]^2\, [S_{2j-2}((1-4p_k)\mathrm{IntervalSize})] \\ &\times\; [S_{2j-2}(p_k \mathrm{IntervalSize})]^2 \end{aligned}$$

end for

$\widetilde{W}_t \leftarrow [S_{2k}(\mathrm{IntervalSize})]^{\mathrm{NumberOfIntervals}}$

end for

return $\widetilde{W} = (\widetilde{W}_0, \widetilde{W}_1, \ldots, \widetilde{W_{b-1}})$

- We are given the matrix exponentials W^{2^t}, $t = 0, \ldots, b-1$.

However, we do not know $|z_{h,0}\rangle$ in our case, so we use an approximation. 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 (4.5) holds. All these approximations affect the estimate $8/\pi^2$ of the success probability of phase estimation, but only by a small amount.

The initial state of our algorithm is

$$|0\rangle^{\otimes b}|\psi_0\rangle^{\otimes d}, \tag{4.6}$$

where $|\psi_0\rangle^{\otimes d}$ is the ground state eigenvector of the discretized Laplacian. We know [19] that the coordinates of $|\psi_0\rangle$ are

$$\psi_{0j} = \sqrt{2h}\sin(j\pi h), \quad j = 1, \ldots, m, \tag{4.7}$$

and $|\psi_0\rangle^{\otimes d}$ has unit length. Since h is proportional to ε, the matrix M_h has size $m^d \times m^d$, with $m = \Theta(\varepsilon^{-1})$. Therefore, $|\psi_0\rangle^{\otimes d} \in \mathbb{C}^{m^d}$ 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 using the Fourier transform; see e.g., [41; 71]. We point out that here and elsewhere the implied constants in the big-O and Θ notation are independent of d and ε. (From a practical standpoint, it is possible to further reduce the cost of the initial state using the algorithm in [37] but we do not pursue this alternative since the analysis of the algorithm becomes more involved.)

Expanding $|\psi_0\rangle^{\otimes d}$ using the eigenvectors of M_h we have

$$|\psi_0\rangle^{\otimes d} = \sum_{k=0}^{m^d-1} d_k |z_{h,k}\rangle.$$

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 $|\psi_0\rangle^{\otimes d}$ onto $|z_{h,0}\rangle$, to become $|d_0|^2 \cdot 8/\pi^2$; see, e.g., [1; 37].

We will see that $|d_0|^2 > \pi^2/10$. Indeed, we estimate $|d_0|$ using the approach in [72, pg. 172], which is based on the separation of the eigenvalues of M_h. In particular, we have

$$1 \geq (E_{h,1} - E_{h,0})^2(1 - |d_0|^2),$$

where $E_{h,0}$ and $E_{h,1}$ are the smallest and second smallest eigenvalues of M_h. We estimate $E_{h,1} - E_{h,0}$ from below using the two smallest eigenvalues of $-\Delta_h$ to obtain $E_{h,1} - E_{h,0} \geq 2h^{-2}(\sin^2(\pi h) - \sin^2(\pi h/2)) - 1$.

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

$$\frac{8}{\pi^2}\left(1 - \frac{1}{(2h^{-2}(\sin^2(\pi h) - \sin^2(\pi h/2)) - 1)^2}\right) > \frac{4}{5}, \tag{4.8}$$

where $h \leq 1/4$. (The overall success probability of the algorithm is also affected by the approximation of the exponentials; once we address that we will provide a final estimate.)

Now let us turn to the approximation of the matrix exponentials. We simulate the evolution of a quantum system with Hamiltonian $H = M_h/(2d)$ for time 2^t for $t = 0, 1, \ldots, b-1$. 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; see e.g., [47, pg. 209]. The evolution of a system with Hamiltonian H_2 can be implemented using two quantum queries and phase kickback. The queries are similar to those in Grover's algorithm [47] and are defined in (1.7).

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

$$\prod_{\ell=1}^{N_t} e^{iA_\ell z_\ell}, \tag{4.9}$$

where $A_\ell \in \{H_1, H_2\}$ and suitable z_ℓ that depend on t and k.

The splitting formula S_{2k} is due to Suzuki [63; 64]. It is used to approximate $e^{i(B+C)\Delta t}$, where B and C are Hermitian matrices. This formula is defined recursively by

$$\begin{aligned} S_2(B, C, \Delta t) &= e^{iB\Delta t/2} e^{iC\Delta t} e^{iB\Delta t/2} \\ S_{2k}(B, C, \Delta t) &= [S_{2k-2}(B, C, p_k \Delta t)]^2 S_{2k-2}(B, C, (1-4p_k)\Delta t) \\ &\qquad \times [S_{2k-2}(B, C, p_k \Delta t)]^2, \end{aligned}$$

where $p_k = (4 - 4^{1/(2k-1)})^{-1}$ for $k = 2, 3, \ldots$.

Unfolding the recurrence above and combining it with [56, Thm. 1] we obtain that the approximation of W^{2^t} has the form

$$\widetilde{W}^{2^t} = e^{iH_1 a_{t,0}} e^{iH_2 b_{t,1}} e^{iH_1 a_{t,1}} \cdots e^{iH_2 b_{t,L_t}} e^{iH_1 a_{t,L_t}}, \tag{4.10}$$

where $s_{t,0}, \ldots, s_{t,L_t}$ and $z_{t,1}, \ldots, z_{t,L_t}$ and L_t are parameters for $t = 0, \ldots, b-1$. The number of exponentials involving H_1 and H_2 in the expression above is $N_t = 2L_t + 1$. The precise definition of the $\widetilde{W}^{2^t}$ for $t = 0, \ldots, b-1$ is presented in pseudocode listing Algorithm 2.

Let $\|\cdot\|$ be the matrix norm induced by the Euclidean vector norm. From [56, Thm. 1 and Cor. 1] the number N_t of exponentials needed to approximate W^{2^t} by a splitting formula of order $2k+1$ with error ε_t for $t = 0, \ldots, b-1$ 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 as follows

$$\begin{aligned} N &= \sum_{t=0}^{b-1} N_t \leq 16e\|H_1\| \left(\frac{25}{3}\right)^{k-1} (8e\|H_2\|)^{1/(2k)} \\ &\qquad \times \sum_{t=0}^{b-1} 2^t \left(\frac{2^t}{\varepsilon_t}\right)^{1/(2k)} \\ &\leq 16e\|H_1\| 2^b \left(\frac{25}{3}\right)^{k-1} \left(160e\, 2^b \|H_2\|\right)^{1/(2k)}, \end{aligned} \tag{4.11}$$

where we obtained the last inequality by setting $\varepsilon_t = (2^{t+1-b})/40$ for $t = 0, \ldots, b-1$. It is easy to check that $\sum_{t=0}^{b-1} \varepsilon_t \leq \frac{1}{20}$. Thus the success probability of phase estimation can be reduced by twice this amount [47, pg. 195]. Using (4.8) we conclude our algorithm succeeds with probability at least

$$\frac{4}{5} - \frac{1}{10} > \frac{2}{3}.$$

The largest eigenvalue of $-\Delta_h$ is $4dh^{-2}\sin^2(\pi m h/2)$. Since $H_1 = -\Delta_h/(4d)$ we have $\|H_1\| \leq \frac{4dh^{-2}}{4d} \leq \varepsilon^{-2}$. Since V is uniformly bounded by one and $H_2 = V_h/(2d)$ we have $\|H_2\| \leq 1/(2d)$. Hence, the algorithm uses a number of exponentials of H_1 and H_2 that 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 $b = \lceil -\log_2 \varepsilon \rceil$ we obtain

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

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

The *optimal* k^*, i.e., the one minimizing the upper bound for N in (4.11), is obtained in [56, Sec. 5] and is given by

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

by the definition of b. The number of exponentials corresponding to k^* satisfies

$$N^* = O\left(\varepsilon^{-3} e^{\sqrt{\ln(1/(d\varepsilon))}}\right) \quad \text{as } d\varepsilon \to 0. \tag{4.12}$$

We remark that of the N^* matrix exponentials, roughly half involve H_1 and the remaining involve H_2; see (4.10). Since each exponential involving H_2 requires two queries the total number of queries is also N^*.

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

$$N^* \cdot O(d \log^2 \varepsilon^{-1}). \tag{4.13}$$

Equations (4.11), (4.12) and (4.13) yield that the total cost of the algorithm, including the number of queries and the number of all other quantum operations, is

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

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

Summarizing our results we see that the dependence on d in the number of qubits is linear, as is the cost. As far as the number of qubits is concerned, this is not really surprising. 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 of the corresponding eigenvector is proportional to ε^{-d} and therefore is represented using a number of qubits proportional to $d \log_2 \varepsilon^{-1}$.

We now turn to the cost. The depth of the quantum circuit realizing the algorithm grows as N^* which is given in (4.12). 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. Roughly half of these exponentials involve the discretized Laplacian Δ_h and the rest involve the discretized potential V_h.

Each of the matrix exponentials involving the d-dimensional Δ_h 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 depends linearly on d.

We consider the cost of the matrix exponentials involving V_h. Each exponential can be implemented with two quantum queries. We assume 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.

Thus the sum of the cost of all matrix exponentials and, therefore, the cost of the algorithm depends linearly on d.

4.6 Discussion

This cost analysis has the advantage that it reveals the computational effort spent on solving the ground state eigenvalue problem unobscured by the actual cost of evaluating V (i.e., the the cost of a quantum query). The analysis 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 the cost of the query by the number of queries and add the product to (4.13) 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 dependence on p of the number of qubits and the cost of the algorithm is linear.

Finally, our analysis assumes a perfect physical realization of a quantum computer. However, for the implementation of the algorithm, one needs to address decoherence and other sources of error for a specific underlying architecture. This may significantly increase the required computational resources. Such a study exists for phase estimation and the Abrams and

Lloyd algorithm [1] applied to the ground state eigenvalue of the transverse Ising model [17]; see also the references therein and [32]. This study is broad enough to cover Shor's algorithm and conveys the general idea in our case as well. It concludes that for the current state of the art in quantum logic array architectures the existing fault tolerance and error correction techniques impose significant resource requirements in the implementation of these algorithms.

4.7 Future work

There are several interesting research directions motivated by this work.

Firstly, one may consider the ground state energy problem for p particles of different masses $m_1, m_2, \ldots, m_p$. The time-independent Schrödinger equation related to the problem is (1.5). We can consider Dirichlet boundary conditions, just as we did in this chapter. The problem is more complicated though, since the discretized Laplacian is replaced by a weighted version, with weights depending on the masses m_j.

Another important research question relevant to our work is the approximation of energies corresponding to excited eigenstates of the Hamiltonian. The related eigenvalues of H are not necessarily approximated within $O(dh)$ by the corresponding eigenvalues of M_h. However it could be possible to use phase estimation to approximate the smallest energies with an approach similar to ours.

Chapter 5

Estimating the ground state energy of the Schrödinger equation for convex potentials

5.1 Introduction

In the previous chapter, we discussed the ground state estimation problem for potentials that are smooth under certain conditions, and uniformly bounded by a relatively small constant. Here, we design a quantum algorithm that overcomes the restriction on the magnitude of the uniform bound on the potential.

Once again, we consider the smallest eigenvalue E_0 given by

$$\left(-\frac{1}{2}\Delta + V\right)\psi_0(x) = E_0\,\psi_0(x) \text{ for all } x \in I_d = (0,1)^{\mathrm{d}}, \tag{5.1}$$

$$\psi_0(x) = 0 \text{ for all } x \in \partial I_d, \tag{5.2}$$

where x is the d-dimensional spatial variable and ψ_0 is a normalized eigenfunction. The boundary conditions are for p particles in a box, where $d = 3p$. Furthermore we assume that all the masses and the normalized Planck constant are one for simplicity and that the potential V satisfies certain conditions.

It is natural to investigate conditions for V beyond those of Chapter 4 and [54] where quantum algorithms, possibly implementing perturbation methods, approximate the ground

state energy without suffering from the curse of dimensionality. Indeed, in this chapter we assume that V and its first order partial derivatives $\partial V/\partial x_j$ for $j = 1, \ldots, d$ are continuous and uniformly bounded from above by constants $C > 1$ and $C' > 0$ respectively, in addition to V being non-negative and convex.

Our algorithm solves the eigenvalue problem for a sequence of Hamiltonians $H_\ell = -\frac{1}{2}\Delta + V_\ell$, for $\ell = 1, 2, \ldots, L$, where $V_\ell = \ell \cdot V/L$. In each of the L stages, the algorithm produces an approximate ground state eigenvector of H_ℓ that is passed on to the next stage. The fact that V is convex allows us to use the bounds on the fundamental gap [6] and to select L accordingly so that the ground state eigenvectors of the successive Hamiltonians have a large enough "overlap" between them. This means that the (approximate) ground state eigenvector of H_ℓ is also an approximate ground state eigenvector of $H_{\ell+1}$. Our algorithm uses a measurement at every stage, which produces with a certain probability an outcome that approximates the ground state energy of the Hamiltonian H with relative error ε. We select the parameters of the algorithm so that the total success probability is at least $3/4$. The resulting cost (including quantum queries and quantum operations) is proportional to

$$c(k) \cdot \varepsilon^{-(3+\frac{1}{2k})} \cdot C^{\frac{4-2\eta}{1-\eta}+\frac{5-2\eta}{2k(1-\eta)}} \cdot d^{1+\frac{4-2\eta}{1-\eta}+\frac{3}{2k(1-\eta)}}$$

and the number of qubits is

$$3\log\varepsilon^{-1} + \frac{2-\eta}{1-\eta} \cdot \log(Cd) + \Theta\left(d \cdot \log\varepsilon^{-1}\right).$$

In the expressions above k is a parameter such that the order of the splitting formula that we use for Hamiltonian simulation is $2k+1$, $c(k)$ increases with k, and $\eta > 0$ is arbitrary.

A direct consequence of our algorithm is that the state produced, approximates[1] the ground state of the discretized Hamiltonian within $1 - O((1/Cd)^2)$. We modify the algorithm to derive approximations of the ground state of the discretized Hamiltonian within $O(\delta)$, where $\delta = o(1/(Cd)^2)$. The resulting cost is proportional to

$$c(k) \cdot C^{1+\frac{1}{2k}} \cdot d^{2-\frac{1}{2k}} \cdot \varepsilon^{-\left(3+\frac{1}{2k}\right)} \cdot \delta^{-1-\frac{1}{2k}-\frac{1}{2-\eta}-\frac{1}{k(2-\eta)}}$$

and the number of qubits is

$$3\log\varepsilon^{-1} + \log\delta^{-1} + \Theta\left(d\log\varepsilon^{-1}\right).$$

[1]We use the expression "a state $|a\rangle$ approximates $|b\rangle$ within δ" to denote that $|\langle a|b\rangle|^2 \geq 1 - \delta$.

Results in this chapter are based on work in [51].

5.2 Discretization error

The discretization of the time-independent Schrödinger equation for smooth uniformly bounded potentials is studied in Section 4.3. We include the details here for completeness.

The finite difference method is frequently used to discretize partial differential equations, and approximate their solutions. The method with mesh size $h = \frac{1}{n+1}$ yields an $n^d \times n^d$ matrix $M_h = -\frac{1}{2}\Delta_h + V_h$, where Δ_h denotes the discretized Laplacian and V_h the diagonal matrix whose entries are the evaluations of the potential on a regular grid with mesh size $h = \frac{1}{n+1}$.

M_h is a symmetric positive definite and sparse matrix. For a potential function V that has bounded first order partial derivatives, we have [69; 70]

$$|E_0 - E_{h,0}| \leq c_1 dh, \tag{5.3}$$

where $E_{h,0}$ is the smallest eigenvalue of M_h. Consider $\hat{E}_{h,0}$ such that

$$|E_{h,0} - \hat{E}_{h,0}| \leq c_2 dh. \tag{5.4}$$

Then we have $|1 - \frac{\hat{E}_{h,0}}{E_0}| \leq c'h$. The inequality follows by observing that $E_0 \geq d\pi^2/2$, for any $V \geq 0$.

5.3 Quantum Algorithm

We consider the Hamiltonian $H_\ell = -\frac{1}{2}\Delta + \ell\, V/L$ and the discretized Hamiltonian $M_{h,\ell} = -\frac{1}{2}\Delta_h + \ell\, V_h/L$, where the value of L will be chosen appropriately later. We proceed in L stages. In the ℓth stage, we solve the eigenvalue problem for H_ℓ (and $M_{h,\ell}$) and pass the results to the next stage. The eigenvalue problem is solved using phase estimation.

In the following section we present some properties of phase estimation. In Sections 5.3.2 and 5.3.3 we present quantum algorithms for estimating the ground state energy of H and the ground state eigenvector of M_h respectively.

5.3.1 Eigenvector approximation through phase estimation

Let A, $\|A\| \leq R$, be an $n^d \times n^d$ Hermitian matrix. Then the eigenvalues of $U = e^{iA/R}$ have the form $e^{i\lambda R}$, where λ denotes an eigenvalue of A. Equivalently $e^{i\lambda/R} = e^{2\pi i\phi_\lambda}$, where $\phi_\lambda = \lambda/(2\pi R) \in (0,1)$ is the phase corresponding to λ.

Consider the phase estimation algorithm presented in Section 4.2 and [47, Fig. 5.2, 5.3] Besides the (approximate) eigenvector, phase estimation uses matrix exponentials of the form $U^\tau = e^{iA\tau/R}$ to accomplish its task. Frequently, approximations $\tilde{U}_\tau$ are used instead. For instance, when A is given as a sum of Hamiltonians each of which can be implemented efficiently one can use a splitting formula [63; 65] to approximate U_τ. Let the initial state and the matrix exponentials in phase estimation be as follows:

- *Initial state:* We have $|0\rangle^{\otimes b}$ in the top register, that deals with the accuracy, and $|\psi_{\text{in}}\rangle$ in the bottom register.

- *Matrix exponentials:* We have a unitary matrix $\tilde{U}_{2^t}$ approximating $U^{2^t} = e^{iA2^t/R}$, for $t = 0, 1, \ldots, b-1$. Assume that the total error in the approximation of the exponentials is bounded by ε_H, i.e.
$$\sum_{j=0}^{b-1} \|U^{2^j} - \tilde{U}_{2^j}\| \leq \varepsilon_H, \tag{5.5}$$
which implies that
$$\|U^t - \tilde{U}_t\| \leq \varepsilon_H, \text{ for } t = 0, 1, \ldots, 2^b - 1$$

Denoting by $\{\lambda_j, |u_j\rangle\}_{j=0,1,\ldots,n^d-1}$ the eigenpairs of A we have
$$|\psi_{\text{in}}\rangle = \sum_j c_j |u_j\rangle. \tag{5.6}$$

Proposition 1. *Consider the phase estimation with initial state $|0\rangle^{\otimes b} \otimes |\psi_{\text{in}}\rangle$ and the unitaries $\tilde{U}_t$, for $t = 1, 2, \ldots, 2^b - 1$. Let m be the measurement outcome of phase estimation and $|\psi_m\rangle$ the final state after the measurement on the bottom register. Let $c_0' = \langle\psi_m|u_0\rangle$ and $c_0 = \langle\psi_{\text{in}}|u_0\rangle$ where $|u_0\rangle$ is the ground state eigenvector. If*

- *b is such that the phases satisfy $|\phi_j - \phi_0| > \frac{5}{2^b}$ for all $j = 1, 2, \ldots, n^d - 1$ and*

- *$|c_0|^2 \geq \frac{\pi^2}{16}$,*

then with probability

$$p \geq |c_0|^2 \cdot \frac{4}{\pi^2} - 2\sum_{j=0}^{b-1} \|U_{2^j} - \tilde{U}_{2^j}\|$$

we get a result m such that

- $\left|\phi_0 - \frac{m}{2^b}\right| \leq \frac{1}{2^{b+1}}$

and

- *if* $1 - |c_0|^2 \leq \gamma\varepsilon_H$ *then* $1 - |c_0'|^2 \leq (\gamma + 14)\varepsilon_H$,
- *if* $1 - |c_0|^2 \geq \gamma\varepsilon_H^{1-\eta}$, *for* $\eta \in (0, 1)$, *then* $|c_0'| \geq |c_0|$,

where γ is a positive constant.

Proof. After the application of $H^{\otimes b}$ on the first register the state becomes

$$\frac{1}{2^{b/2}} \sum_{k=0}^{2^b-1} |k\rangle \sum_{j=0}^{n^d-1} c_j |u_j\rangle$$

The state of the system after the application of the controlled $\tilde{U}_t$, $t = 0, 1, \ldots, b-1$ is

$$\sum_{j=0}^{n^d-1} c_j \frac{1}{2^{b/2}} \sum_{k=0}^{2^b-1} |k\rangle \tilde{U}_k |u_j\rangle = \frac{1}{2^{b/2}} \sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^b-1} |k\rangle \left(U^k|u_j\rangle + D_k|u_j\rangle\right),$$

where $D_k = \tilde{U}_k - U^k$. Then $\|D_k\| \leq \varepsilon_H$. Since $|u_j\rangle$, $j = 0, 1, \ldots, n^d - 1$, are the eigenvectors of U, the state can be written as $|\psi_1\rangle + |\psi_2\rangle$, where

$$|\psi_1\rangle = \frac{1}{2^{b/2}} \sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^b-1} |k\rangle U^k |u_j\rangle = \frac{1}{2^{b/2}} \sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^b-1} |k\rangle e^{2\pi i k\phi_j} |u_j\rangle,$$

and

$$|\psi_2\rangle = \frac{1}{2^{b/2}} \sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^b-1} |k\rangle D_k |u_j\rangle = \frac{1}{2^{b/2}} \sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^b-1} |k\rangle |x_{j,k}\rangle,$$

where $|x_{j,k}\rangle := D_k|u_j\rangle$. Clearly $\||x_{j,k}\rangle\| \leq \varepsilon_H$, for all $k = 0, 1, \ldots, 2^b - 1$ and $j = 0, 1, \ldots, 2^b - 1$.

The next step in the phase estimation is to apply $\mathbb{F}^H \otimes I$, where $\mathbb{F}^H$ is the inverse Fourier transform. The state becomes $|\psi_{\mathbb{F}^H}\rangle = \left|\psi_{1,\mathbb{F}^H}\right\rangle + \left|\psi_{2,\mathbb{F}^H}\right\rangle$, where

$$\begin{aligned}\left|\psi_{1,\mathbb{F}^H}\right\rangle &= \frac{1}{2^{b/2}} \sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^b-1} \mathbb{F}^H|k\rangle e^{2\pi i k\phi_j}|u_j\rangle \\ &= \frac{1}{2^b} \sum_{j=0}^{n^d-1} c_j \sum_{k,l=0}^{2^b-1} e^{2\pi i k\left(\phi_j - \frac{l}{2^b}\right)}|l\rangle|u_j\rangle \\ &= \sum_{j=0}^{n^d-1} c_j \sum_{l=0}^{2^b-1} \alpha(l,\phi_j)|l\rangle|u_j\rangle\end{aligned}$$

where

$$\alpha(l,\phi_j) := \frac{1}{2^b} \sum_{k=0}^{2^b-1} e^{2\pi i k\left(\phi_j - \frac{l}{2^b}\right)} \tag{5.7}$$

and

$$\left|\psi_{2,\mathbb{F}^H}\right\rangle = \sum_{j=0}^{n^d-1} c_j \frac{1}{2^{b/2}} \sum_{k=0}^{2^b-1} \mathbb{F}^H|k\rangle|x_{j,k}\rangle = \sum_{j=0}^{n^d-1} c_j \frac{1}{2^b} \sum_{k,l=0}^{2^b-1} e^{-2\pi i l k/2^b}|l\rangle|x_{j,k}\rangle.$$

Finally we measure the top register on the computational basis states. The resulting state is

$$|\psi_m\rangle = \frac{|\psi_{1,m}\rangle + |\psi_{2,m}\rangle}{\||\psi_{1,m}\rangle + |\psi_{2,m}\rangle\|},$$

where

$$|\psi_{1,m}\rangle = \frac{1}{2^b} \sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^b-1} e^{2\pi i k\left(\phi_j - \frac{m}{2^b}\right)}|m\rangle|u_j\rangle = \sum_{j=0}^{n^d-1} c_j \alpha(m,\phi_j)|m\rangle|u_j\rangle,$$

and

$$|\psi_{2,m}\rangle = \frac{1}{2^b} \sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^b-1} e^{-2\pi i m k/2^b}|m\rangle|x_{j,k}\rangle \tag{5.8}$$

We now consider the magnitude of the projection of the resulting state $|\psi_m\rangle$ on the ideal state, namely $|c_0'| = |\langle\psi_m|m,u_0\rangle|$.

We have

$$|\langle\psi_m|m,u_0\rangle|^2 = \frac{\left|c_0\alpha(m,\phi_0) + \frac{1}{2^b}\sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^b-1} e^{-2\pi i m k/2^b}\langle u_0|x_{j,k}\rangle\right|^2}{\||\psi_{1,m}\rangle + |\psi_{2,m}\rangle\|^2} \tag{5.9}$$

Without accounting for the simulation error, with probability at least $|c_0|^2|\alpha(m,\phi_0)| \geq 4|c_0|^2/\pi^2$ we get a result m such that $\left|\phi_0 - \frac{m}{2^b}\right| \leq \frac{1}{2^{b+1}}$, see [28, Thm. 11]. If we account for

the simulation error, the probability of getting such a result becomes at least

$$|c_0|^2 \cdot \frac{4}{\pi^2} - 2\sum_{j=0}^{b-1} \|U_{2^j} - \tilde{U}_{2^j}\|,$$

see [47, pg. 195]. From now on, the analysis considers that specific result m.

From Lemma 4 we have $\||\psi_{2,m}\rangle\| \leq \varepsilon_H$. In addition

$$\langle\psi_{2,m}|m,u_0\rangle \leq \||\psi_{2,m}\rangle\| \cdot \||m,u_0\rangle\| \leq \varepsilon_H.$$

Hence, according to Lemma 3 and for $\varepsilon_H < \sqrt{8}/\pi^2$, equation (5.9) becomes

$$\begin{aligned} |c_0'| &> |c_0| \left(\frac{|\alpha(m,\phi_0)|}{\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 \cdot |\alpha(m,\phi_j)|^2}} - 7\frac{\varepsilon_H}{|c_0|} \right) \\ &= |c_0| \left(\frac{1}{\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 \cdot \frac{|\alpha(m,\phi_j)|^2}{|\alpha(m,\phi_0)|^2}}} - 7\frac{\varepsilon_H}{|c_0|} \right) \\ &= |c_0| \left(\frac{1}{\sqrt{|c_0|^2 + \sum_{j=1}^{n^d-1} |c_j|^2 \cdot \frac{|\alpha(m,\phi_j)|^2}{|\alpha(m,\phi_0)|^2}}} - 7\frac{\varepsilon_H}{|c_0|} \right) \end{aligned} \tag{5.10}$$

Note that $|\alpha(m,\phi_0)|^2 \geq 8/(2\pi^2) = 4/\pi^2$, see [28]. Let

$$k := \{j : |\alpha(m,\phi_j)|^2 = \max_{i\geq 1} |\alpha(m,\phi_i)|^2\}.$$

Then $|\alpha(m,\phi_k)|^2 \leq \frac{1}{(2\cdot 2^b \cdot 2^{-b+2})^2} = 1/64$, see [28, Thm. 11], where the size of the grid is 2^b and the minimum distance of any phase ϕ_j (where $j = 1, 2, \ldots n^d - 1$ from $m/2^b$) is at least 2^{-b+2}, according to the assumptions of Proposition 1. Hence (5.10) becomes

$$\begin{aligned} |c_0'| &> |c_0| \left(\frac{1}{\sqrt{|c_0|^2 + \frac{\pi^2}{256} \cdot \sum_{j=1}^{n^d-1} |c_j|^2}} - 7\frac{\varepsilon_H}{|c_0|} \right) \\ &= |c_0| \left(\frac{1}{\sqrt{|c_0|^2 + \frac{\pi^2}{256} \cdot (1 - |c_0|^2)}} - 7\frac{\varepsilon_H}{|c_0|} \right), \end{aligned} \tag{5.11}$$

since $\sum_{j=0}^{n^d-1} |c_j|^2 = 1$.

Now examine the different cases, depending on the magnitude of $|c_0|$.

Case 1: $1-|c_0|^2 \leq \gamma\varepsilon_H$, for a constant γ. Then (5.11) becomes

$$|c_0'| > |c_0|\left(1-7\frac{\varepsilon_H}{\sqrt{1-\gamma\varepsilon_H}}\right),$$

since $f(x)=\left(x+\frac{\pi^2}{256}(1-x)\right)^{-1/2}$ is a monotonically decreasing function for $x\in[0,1]$. Hence

$$\begin{aligned}
|c_0'|^2 &> |c_0|^2\left(1-\frac{14}{\sqrt{1-\gamma\varepsilon_H}}\varepsilon_H+\frac{49}{1-\gamma\varepsilon_H}\varepsilon_H^2\right)\\
&\geq (1-\gamma\varepsilon_H)\cdot\left(1-\frac{14}{\sqrt{1-\gamma\varepsilon_H}}\varepsilon_H+\frac{49}{1-\gamma\varepsilon_H}\varepsilon_H^2\right)\\
&= 1-\gamma\varepsilon_H-14\varepsilon_H\sqrt{1-\gamma\varepsilon_H}+49\varepsilon_H^2\\
&\geq 1-\gamma\varepsilon_H-14\varepsilon_H+49\varepsilon_H^2\geq 1-(\gamma+14)\varepsilon_H,
\end{aligned}$$

since $1-\gamma\varepsilon_H<1$. This concludes the first part of the theorem.

Case 2: $1-|c_0|^2\geq\gamma\varepsilon_H^{1-\eta}$, for $\eta\in(0,1)$ and $\gamma>0$. Then (5.11) becomes

$$|c_0'| > |c_0|\left(\frac{1}{\sqrt{1-\left(1-\frac{\pi^2}{256}\right)\gamma\varepsilon_H^{1-\eta}}}-7\frac{\varepsilon_H}{\pi/4}\right),$$

since $f(x)=\left(x+\frac{\pi^2}{256}(1-x)\right)^{-1/2}$ is a monotonically decreasing function for $x\in[0,1-\gamma\varepsilon_H^{1-\eta}]$ and $|c_0|^2\geq\frac{\pi^2}{16}$.

Note that $(1-a)^{-1/2}\geq\sqrt{1+a}$, for $|a|\leq 1$. Hence

$$\begin{aligned}
|c_0'|^2 &> |c_0|^2\left(1+\left(1-\frac{\pi^2}{256}\right)\gamma\varepsilon_H^{1-\eta}-\frac{56}{\pi}\varepsilon_H\sqrt{1+\left(1-\frac{\pi^2}{256}\right)\gamma\varepsilon_H^{1-\eta}}+\frac{28^2}{\pi^2}\varepsilon_H^2\right)\\
&> |c_0|^2\left(1+\left(1-\frac{\pi^2}{256}\right)\gamma\varepsilon_H^{1-\eta}-O(\varepsilon_H)\right)>|c_0|^2,
\end{aligned}$$

for ε_H sufficiently small. □

We now present a modified version of Proposition 1, where we have extended the first register with t_0 qubits. Similarly as before, we assume that (5.5) holds.

Theorem 4. *Let $|\psi_{m'}\rangle$ be the final state in the bottom register after the measuring m' on the top register of the phase estimation procedure with initial state $|0\rangle^{\otimes(b+t_0)}|\psi_{in}\rangle$ and unitaries $\tilde{U}_t$, for $t=1,\ldots,2^{b+t_0}-1$ and $t_0\geq 1$. Let $c_0'=\langle\psi_{m'}|u_0\rangle$ and $c_0=\langle\psi_{m'}|u_0\rangle$, where $|u_0\rangle$ is the ground state eigenvector. If*

- *b is such that the phases satisfy $|\phi_j - \phi_0| > \frac{5}{2^b}$ for all $j = 1, 2, \ldots, n^d - 1$.*
- $|c_0|^2 \geq \frac{\pi^2}{16}$.

Then with probability

$$p \geq |c_0|^2 \left(1 - \frac{1}{2(2^{t_0} - 1)}\right) - \left(\frac{5\pi^2}{2^5} + \frac{1 - \frac{\pi^2}{16}}{2^5}\right) \cdot \frac{1}{2^{t_0}} - 2 \sum_{j=0}^{b+t_0-1} \|U_{2^j} - \tilde{U}_{2^j}\|$$

we get a result m' satisfying

- $m' \in \mathcal{G}$, *with* $\mathcal{G} = \left\{m \in \{0, 1, \ldots 2^{b+t_0} - 1\} \ : \ \left|\phi_0 - \frac{m'}{2^{b+t_0}}\right| \leq \frac{1}{2^b}\right\}$

and

- *if* $1 - |c_0|^2 \leq \gamma \varepsilon_H$ *then* $1 - |c_0'|^2 \leq (\gamma + 14)\varepsilon_H$
- *if* $1 - |c_0|^2 \geq \gamma \varepsilon_H^{1-\eta}$, *for* $\eta \in (0, 1)$, *then* $|c_0'| \geq |c_0|$

where γ denotes a positive constant.

Proof. Just as in Proposition 1 we reach a similar version of equation (5.9)

$$|\left\langle \psi_{m'} | m', u_0 \right\rangle|^2 = \frac{\left| c_0 \alpha(m', \phi_0) + \frac{1}{2^b} \sum_{j=0}^{n^d-1} c_j \sum_{k=0}^{2^{b+t_0}-1} e^{-2\pi i m' k / 2^{b+t_0}} \left\langle u_0 | x_{j,k} \right\rangle \right|^2}{\| \left| \psi_{1,m'} \right\rangle + \left| \psi_{2,m'} \right\rangle \|^2},$$

Without accounting for the simulation error, with probability at least

$$|c_0|^2 \geq |c_0|^2 \cdot \left(1 - (2(2^{t_0} - 1))^{-1}\right)$$

we get a result m' such that $m' \in \mathcal{G}$, with

$$\mathcal{G} = \left\{m \in \{0, 1, \ldots 2^{b+t_0} - 1\} \ : \ \left|\phi_0 - \frac{m'}{2^{b+t_0}}\right| \leq \frac{2^{t_0}}{2^{b+t_0}} = \frac{1}{2^b}\right\},$$

see [28, Thm. 11]. Moreover, according to Lemma 5 the probability of getting a result $m' \in \mathcal{G}$ with $\frac{|\alpha(m', \phi_j)|^2}{|\alpha(m', \phi_0)|^2} \leq \frac{\pi^2}{32}$ for all $j \geq 1$ is at least $|c_0|^2 \left(1 - \frac{1}{2(2^{t_0} - 1)}\right) - \left(\frac{5\pi^2}{2^5} + \frac{1 - \frac{\pi^2}{16}}{2^5}\right) \cdot \frac{1}{2^{t_0}}$.

Accounting for the simulation error, the probability of getting such results is at least

$$|c_0|^2 \left(1 - \frac{1}{2(2^{t_0} - 1)}\right) - \left(\frac{5\pi^2}{2^5} + \frac{1 - \frac{\pi^2}{16}}{2^5}\right) \cdot \frac{1}{2^{t_0}} - 2 \sum_{j=0}^{b+t_0-1} \|U_{2^j} - \tilde{U}_{2^j}\|,$$

see [47, pg. 195]. From now on we consider only such results.

As in Proposition 1, we reach equation (5.10). For the results of interest, we have $\frac{|\alpha(m',\phi_j)|^2}{|\alpha(m',\phi_0)|^2} \leq \frac{\pi^2}{32}$, for $j \geq 1$ Hence (5.10) becomes

$$\begin{aligned} |c_0'| &> |c_0| \left(\frac{1}{\sqrt{|c_0|^2 + \frac{\pi^2}{32} \cdot \sum_{j=1}^{n^d-1} |c_j|^2}} - 7\frac{\varepsilon_H}{|c_0|} \right) \\ &= |c_0| \left(\frac{1}{\sqrt{|c_0|^2 + \frac{\pi^2}{32} \cdot (1 - |c_0|^2)}} - 7\frac{\varepsilon_H}{|c_0|} \right), \end{aligned} \tag{5.12}$$

since $\sum_{j=0}^{n^d-1} |c_j|^2 = 1$.

Now examine the different cases, depending on the magnitude of $|c_0|$.

Case 1: $1 - |c_0|^2 \leq \gamma\varepsilon_H$, for a constant γ. Then (5.12) becomes

$$|c_0'| > |c_0| \left(1 - 7\frac{\varepsilon_H}{\sqrt{1 - \gamma\varepsilon_H}} \right),$$

since $f(x) = \left(x + \frac{\pi^2}{32}(1-x) \right)^{-1/2}$ is a monotonically decreasing function and $|c_0|^2 \leq 1$. Hence

$$\begin{aligned} |c_0'|^2 &> |c_0|^2 \left(1 - \frac{14}{\sqrt{1 - \gamma\varepsilon_H}}\varepsilon_H + \frac{49}{1 - \gamma\varepsilon_H}\varepsilon_H^2 \right) \\ &\geq (1 - \gamma\varepsilon_H) \cdot \left(1 - \frac{14}{\sqrt{1 - \gamma\varepsilon_H}}\varepsilon_H + \frac{49}{1 - \gamma\varepsilon_H}\varepsilon_H^2 \right) \\ &= 1 - \gamma\varepsilon_H - 14\varepsilon_H\sqrt{1 - \gamma\varepsilon_H} + 49\varepsilon_H^2 \\ &\geq 1 - \gamma\varepsilon_H - 14\varepsilon_H + 49\varepsilon_H^2 \geq 1 - (\gamma + 14)\varepsilon_H, \end{aligned}$$

since $1 - \gamma\varepsilon_H < 1$. This concludes the first part of the theorem.

Case 2: $1 - |c_0|^2 \geq \gamma\varepsilon_H^{1-\eta}$, for some $\eta \in (0,1)$ and $\gamma > 0$. Then (5.12) becomes

$$|c_0'| > |c_0| \left(\frac{1}{\sqrt{1 - \left(1 - \frac{\pi^2}{32}\right)\gamma\varepsilon_H^{1-\eta}}} - 7\frac{\varepsilon_H}{\pi/4} \right),$$

since $f(x) = \left(x + \frac{\pi^2}{32}(1-x) \right)^{-1/2}$ is a monotonically decreasing function and $|c_0|^2 \geq \pi^2/16$.

Note that $(1-a)^{-1/2} \geq \sqrt{1+a}$, for $|a| \leq 1$. Hence

$$\begin{aligned} |c_0'|^2 &> |c_0|^2 \left(1 + \left(1 - \frac{\pi^2}{32}\right)\gamma\varepsilon_H^{1-\eta} - \frac{56}{\pi}\varepsilon_H\sqrt{1 + \left(1 - \frac{\pi^2}{32}\right)\gamma\varepsilon_H^{1-\eta}} + \frac{28^2}{\pi^2}\varepsilon_H^2 \right) \\ &> |c_0|^2 \left(1 + \left(1 - \frac{\pi^2}{32}\right)\gamma\varepsilon_H^{1-\eta} - O(\varepsilon_H) \right) \end{aligned}$$

This concludes the proof, since we can discard the $O(\varepsilon_H)$ terms. □

5.3.2 A quantum algorithm estimating the ground state energy

Let $M_{h,\ell} = -\frac{1}{2}\Delta_h + \frac{\ell}{L}V_h$ with $|u_{0,\ell}\rangle$ the corresponding ground state eigenvectors, for $\ell = 1, 2, \ldots, L$. At each stage we approximate the minimum eigenvalue of $M_{h,\ell}$ within relative error ε.

We first introduce some useful notation. Phase estimation requires two quantum registers [47, Fig. 5.2, 5.3]. The upper register determines the accuracy and the probability of success of the algorithm and the lower register holds an approximation of the ground state of $M_{h,\ell}$. Let $|\psi_{\text{in},\ell}\rangle$ the initial state on the lower register at the ℓth stage of the algorithm and $|\psi_{\text{out},\ell}\rangle$ the state on the same register once the stage is complete.

We use as initial state for our algorithm the state $|\psi_{\text{in},1}\rangle = |\psi_{-\Delta_h}\rangle$, i.e. the eigenstate of the discretized Laplacian. By choosing an appropriately large L, and lower bounds on the eigenvalue gap between the first and the second eigenvalues of Hamiltonians involving convex potentials, we ensure that the initial state of the algorithm has good overlap with the ground state of $M_{h,1}$. Using the results of Theorem 4 we maintain this good overlap throughout all the stages with high probability. As in the Theorem, we use $b + t_0$ qubits on the upper register. The b qubits are used to control the accuracy of the method and the t_0 qubits are used to boost the probability of success of each stage.

We provide an overview of the algorithm.

1. Run phase estimation on $W_{h,\ell} = e^{-iM_{h,\ell}/R}$ for $\ell = 1, 2, \ldots, L$, starting from $W_{h,1}$, with R a parameter to be defined later. We modify phase estimation as follows:

   - *Number of qubits:* The upper register has $b + t_0$ qubits, while the lower register has $d \log_2 h^{-1}$ qubits.
   - *Input state:* The upper register is initialized to $|0\rangle^{\otimes(b+t_0)}$. The lower register of the first stage is initialized to $|\psi_{-\Delta_h}\rangle$. Furthermore we set $|\psi_{\text{in},\ell}\rangle := |\psi_{\text{out},\ell-1}\rangle$ for $\ell = 2, 3, \ldots, L$.
   - *Implementation of exponentials:* Implement each exponential $W_{h,\ell}^{2^j}$ participating on the ℓth stage, for $j = 0, 1, \ldots, t_0 + b - 1$ using Suzuki splitting formulas [63; 65].

2. *Output:* Let $j \in \{0, 1, \ldots, 2^{t_0+b} - 1\}$ be the result of the measurement on the upper register after the last stage. Output $\hat{E}_{h,0} = 2\pi \cdot R \cdot j \cdot 2^{-(b+t_0)}$

Let $\lambda_{j,\ell}$ be the jth eigenvalue of $M_{h,\ell}$. The phase corresponding to this eigenvalue is

$$\phi_{j,\ell} = \frac{\lambda_{j,\ell}}{2\pi R}.$$

Set $R = 3dh^{-2} \gg 2dh^{-2} + C = \| - \frac{1}{2}\Delta_h + V_h\|$. This choice of R guarantees that $\phi_{j,\ell} \in [0, 1)$ for all $j = 0, 1, \ldots, n^d - 1$ and $\ell = 1, 2, \ldots, L$.

We refer to the algorithm as *repeated phase estimation* or RPE in short, since the algorithm uses repetitions of the phase estimation procedure.

5.3.2.1 Error analysis

We know (eq. (5.3), (5.4)) that we can achieve relative error $O(h)$ if we approximate the ground energy of $M_{h,L}$ with error at most dh. This implies that the algorithm has to approximate the eigenvalues $\lambda_{0,\ell}$ within error dh, for all $\ell = 0, 1, \ldots, L$, which in turn requires $\phi_{0,L}$ to be approximated with error $\frac{dh}{2\pi R}$. This translates to $2^{-b} \leq \frac{dh}{2\pi R}$, which in turn leads to

$$b = \left\lceil \log \frac{2R\pi}{dh} \right\rceil = \left\lceil \log(6\pi h^{-3}) \right\rceil = \log \Theta\left(h^{-3}\right). \tag{5.13}$$

5.3.2.2 Preliminary Analysis

We need to quantify how the results of one stage of the algorithm affect the success probability of the next stage in the case of the Schrödinger equation with convex potential.

The fundamental gap for Hamiltonians of the form $-\frac{1}{2}\Delta + V$, where V is a convex potential, is at least $3\pi^2/(2d)$, see [6]. The gap between the first and second eigenvalues of $M_{h,\ell}$, for $\ell = 1, 2, \ldots, L$, is reduced by $O(dh)$, see [69; 70]. For $h = o(d^{-2})$, the gap is at least $\frac{3\pi^2}{2d} - o(d^{-1}) \geq \frac{\pi^2}{d}$. Set $2^{-b} < \frac{1}{5} \cdot \frac{\pi^2}{d \cdot 2\pi R}$, which implies $h < \frac{2\pi^2}{5} \cdot \frac{1}{d^2}$ according to (5.13). This leads to $|\phi_0 - \phi_j| \geq \frac{5}{2^b}$, for all $j \geq 1$. As a result, for $h = o(d^{-2})$ the requirements of Theorem 4 hold.

Let $L = \omega(d)$ to be specified later. Consider the $(\ell-1)$th stage, with initial state $|\psi_{\text{in},\ell-1}\rangle$ and Hamiltonian $M_{h,\ell-1}$. Assume $|\langle\psi_{\text{out},\ell-1}|u_{0,\ell-1}\rangle| = 1 - \kappa\delta$, where $\kappa > 0$ a constant and $\delta \in [0, 1)$ a quantity that is $\delta = \omega((Cd)^2/L^2)$. That means that $|\psi_{\text{in},\ell-1}\rangle$ is not a good

approximation of $|u_{0,\ell-1}\rangle$. In addition, assume that $\varepsilon_H = o(\delta)$. Then the magnitude of the projection of the resulting state $|\psi_{\text{out},\ell-1}\rangle$ of this stage onto the ground state eigenvector follows from Theorem 4, as shown below.

Corollary 3. *Let* $|\langle\psi_{\text{in},\ell-1}|u_{0,\ell-1}\rangle|^2 = 1-\kappa\delta$, *where* $\kappa > 0$, $\delta \in [0,1)$ *and* $\delta = \omega(\varepsilon_H)$. *Then*

$$|\langle\psi_{\text{out},\ell-1}|u_{0,\ell-1}\rangle|^2 \geq 1 - \frac{\pi^2+1}{32}\kappa\delta, \quad \ell \geq 2.$$

Proof. We reconsider the case 2 of Theorem 4. Retracing the steps, we reach to

$$\begin{aligned}
|\langle\psi_{\text{out},\ell-1}|u_{0,\ell-1}\rangle|^2 &> |\langle\psi_{\text{in},\ell-1}|u_{0,\ell-1}\rangle|^2 \left(1 + \left(1 - \frac{\pi^2}{32}\right)\kappa\delta - O(\varepsilon_H)\right) \\
&= (1-\kappa\delta)\left(1 + \left(1 - \frac{\pi^2}{32}\right)\kappa\delta - O(\varepsilon_H)\right) \\
&= 1 - \frac{\pi^2}{32}\kappa\delta - O(\varepsilon_H) + \kappa\delta \cdot O(\varepsilon_H) \\
&\geq 1 - \frac{\pi^2+1}{32}\kappa\delta,
\end{aligned}$$

where the last inequality is due to $\delta \in [0,1)$ and $\delta = \omega(\varepsilon_H)$. □

Note that after stage $\ell-1$, phase estimation has improved the approximation of $|u_{0,\ell-1}\rangle$. In addition, $|\langle\psi_{\text{in},\ell}|u_{0,\ell}\rangle| = |\langle\psi_{\text{out},\ell-1}|u_{0,\ell}\rangle|$ determines the probability of success of the ℓth stage. In order to calculate this probability, we need to consider the projection of $|u_{0,\ell-1}\rangle$ onto $|u_{0,\ell}\rangle$.

Taking into account the lower bound on the gap between the first two eigenvalues of $M_{h,\ell}$, we express $|u_{0,\ell-1}\rangle$ in terms of the eigenstates of $M_{h,\ell}$ to get

$$\|V/L\|_\infty^2 \geq (1 - |\langle u_{0,\ell-1}|u_{0,\ell}\rangle|^2)\left(\frac{\pi^2}{d}\right)^2 \Rightarrow |\langle u_{0,\ell-1}|u_{0,\ell}\rangle|^2 \geq 1 - \left(\frac{Cd}{\pi^2 L}\right)^2, \tag{5.14}$$

for $\ell = 2,\ldots,L$, see [50].

Lemma 2. *Let*

$$\begin{aligned}
|\langle\psi_{\text{out},\ell-1}|u_{0,\ell-1}\rangle|^2 &\geq 1-\kappa'\delta \\
|\langle u_{0,\ell-1}|u_{0,\ell}\rangle|^2 &\geq 1 - \left(\frac{Cd}{\pi^2 L}\right)^2,
\end{aligned}$$

with $\delta = \omega\left(\left(\frac{Cd}{L}\right)^2\right)$. *Then*

$$|\langle\psi_{\text{out},\ell-1}|u_{0,\ell}\rangle|^2 \geq 1 - \kappa'\delta - o(\delta) \quad \ell \geq 2,$$

where $\kappa' > 0$ a constant.

Proof. Let $\theta_1 := \arccos |\langle \psi_{\text{out},\ell-1} | u_{0,\ell-1} \rangle|$ and $\theta_2 = \arccos |\langle u_{0,\ell-1} | u_{0,\ell} \rangle|$.

Then $|\langle \psi_{\text{out},\ell-1} | u_{0,\ell} \rangle|^2 \geq \sin^2(\theta_1 + \theta_2)$. Note that

$$\cos^2(\theta_1 + \theta_2) = \frac{1}{2}[1 + \cos(2(\theta_1 + \theta_2))] = \frac{1}{2}[1 + \cos(2\theta_1)\cos(2\theta_2) - \sin(2\theta_1)\sin(2\theta_2).$$

Now $\cos^2(\theta_1) = \frac{1}{2}[1 + \cos(2\theta_1)] \geq 1 - \kappa'\delta$, which leads to

$$\cos(2\theta_1) \geq 1 - 2\kappa'\delta.$$

Similarly

$$\cos(2\theta_2) \geq 1 - 2\left(\frac{Cd}{\pi^2 L}\right)^2.$$

Furthermore

$$\sin^2(2\theta_1) = 1 - \cos^2(2\theta_1) \leq 1 - \left[1 - 2\kappa'\delta\right]^2 \leq 4\kappa'\delta,$$

and similarly $\sin^2(2\theta_2) \leq 4\left(\frac{Cd}{\pi^2 L}\right)^2$. According to the above

$$\begin{aligned}
\cos^2(\theta_1 + \theta_2) &\geq \frac{1}{2}\left[1 + \left(1 - 2\kappa'\delta\right) \cdot \left(1 - 2\left(\frac{Cd}{\pi^2 L}\right)^2\right) - \sqrt{4\kappa'\delta} \cdot \sqrt{4\left(\frac{Cd}{\pi^2 L}\right)^2}\right] \\
&\geq 1 - \kappa'\delta + 2\kappa'\delta\left(\frac{Cd}{\pi^2 L}\right)^2 - \left(\frac{Cd}{\pi^2 L}\right)^2 - 2\sqrt{\kappa'} \cdot \sqrt{\delta \cdot \left(\frac{Cd}{\pi^2 L}\right)^2} \\
&\geq 1 - \kappa'\delta - o(\delta)
\end{aligned}$$

since $\left(\frac{Cd}{\pi^2 L}\right)^2 = o(\delta)$ and $\sqrt{\delta\left(\frac{Cd}{\pi^2 L}\right)^2} = \sqrt{\delta \cdot o(\delta)} = o(\delta)$. □

Consider errors $\varepsilon_{j,\ell}^S$ for the exponentials $W_{h,\ell}^{2^j}$ such that the total error for each stage is $\sum_{j=0}^{t_0+b-1} \varepsilon_{j,\ell}^S \leq \varepsilon_H = O\left(\left(\frac{Cd}{L}\right)^2\right)$. We use Corollary 3, and Lemma 2 to get

$$|\langle \psi_{\text{out},\ell-1} | u_{0,\ell} \rangle|^2 = 1 - \frac{\pi^2 + 1}{32}\kappa\delta - o(\delta) \geq 1 - \left(\frac{\pi^2 + 1}{32} + 10^{-3}\right)\kappa\delta,$$

where $\kappa > 0$ is a constant, $\delta = \omega\left(\left(\frac{Cd}{L}\right)^2\right)$ and $\ell \geq 2$.

5.3.2.3 Initial state

The initial state of our algorithm is the ground state eigenvector $|\psi_{-\Delta_h}\rangle$ of the discretized Laplacian. We have $|\psi_{-\Delta_h}\rangle = |z\rangle^{\otimes d}$, where $|z\rangle$ is the ground state eigenvector of the $n \times n$ matrix corresponding to the one dimensional problem. The coordinates of $|z\rangle$ are

$$z_j = \sqrt{2h}\sin(j\pi h), \quad \text{for } j = 1, 2, \ldots, n.$$

and it can be implemented in a number quantum operations proportional to $\log^2 h^{-1}$, see [41]. Thus, the implementation of the initial state of the algorithm $|\psi_{\Delta_h}\rangle$ requires a number of quantum operations proportional to $d\log^2 h^{-1}$.

According to (5.14) we have

$$|\langle\psi_{\text{in},1}|u_{0,1}\rangle|^2 \geq 1 - \left(\frac{Cd}{\pi^2 L}\right)^2, \tag{5.15}$$

since $|\psi_{\text{in},1}\rangle := |u_{0,1}\rangle$.

5.3.2.4 Success probability

According to the analysis in Sections 5.3.2.2, 5.3.2.3, we have

$$|\langle\psi_{\text{in},\ell}|u_{0,\ell}\rangle|^2 \geq 1 - \delta,$$

for any $\delta = \omega\left(\left(\frac{Cd}{L}\right)^2\right)$ and $\ell = 1, 2, \ldots, L$, as long as the total error due to the approximation of exponentials at each stage is $O\left(\left(\frac{Cd}{L}\right)^2\right)$. As a result,

$$|\langle\psi_{\text{in},\ell}|u_{0,\ell}\rangle|^2 \geq 1 - \kappa_1\left(\frac{Cd}{L}\right)^{2-\eta}, \tag{5.16}$$

for $\kappa_1 > 0$ a constant and any $\eta > 0$.

According to Theorem 4, the total probability of success of the algorithm after L steps is

$$P_{\text{total}} \geq \left(\min_{\ell=1,2,\ldots,L-1} |\langle\psi_{\text{in},\ell}|u_{0,\ell}\rangle|^2 \cdot \left(1 - \frac{1}{2(2^{t_0}-1)}\right) - \left(\frac{5\pi^2}{2^5} + \frac{1-\pi^2/16}{2^5}\right)\frac{1}{2_0^t} - 2\sum_{j=0}^{b+t_0-1} \|W_{h,\ell}^{2^j} - \widetilde{W_{h,\ell}^{2^j}}\|\right)^L \tag{5.17}$$

We use Suzuki splitting formulas to approximate the exponentials $W_{h,\ell}^{2^j}$, for $j = 0, 1, \ldots, t_0 + b - 1$, see [63; 65]. Let $\varepsilon_{j,l}^S = \|W_{h,\ell}^{2^j} - \widetilde{W_{h,\ell}^{2^j}}\|$. We pick

$$\varepsilon_{j,l}^S := 2^{j-(b+t_0)} \cdot \left(\frac{Cd}{L}\right)^2,$$

The total error at each stage is

$$\sum_{j=0}^{b+t_0-1} \varepsilon_{j,l}^S \leq \left(\frac{Cd}{L}\right)^2$$

and the choice of $\varepsilon_{j,l}^S$ implies that

$$\|W_{h,\ell}^k - \widetilde{W_{h,\ell}^k}\| \leq \left(\frac{Cd}{L}\right)^2$$

for $k = 0, 1, \ldots 2^{t_0+b} - 1$. Using (5.16) and the inequality above, (5.17) becomes

$$\begin{aligned} P_{\text{total}} \geq \Bigg(\Bigg(1 - \kappa_1 \left(\frac{Cd}{L}\right)^{2-\eta}\Bigg) \cdot \left(1 - \frac{1}{2(2^{t_0}-1)}\right) - \\ \left(\frac{5\pi^2}{2^5} + \frac{1-\pi^2/16}{2^5}\right)\frac{1}{2_0^t} - 2\left(\frac{Cd}{L}\right)^2\Bigg)^L \end{aligned} \tag{5.18}$$

Set t_0 such that $\frac{1}{2(2^{t_0}-1)} \leq \left(\left(\frac{Cd}{L}\right)^{2-\eta}\right)$, i.e.

$$t_0 = \left\lceil \log\left(\frac{L}{Cd}\right)^{2-\eta} \right\rceil. \tag{5.19}$$

Then (5.18) becomes

$$\begin{aligned} P_{\text{total}} &\geq \left[\left(1 - \kappa_1 \left(\frac{Cd}{L}\right)^{2-\eta}\right) \cdot \left(1 - \left(\frac{Cd}{L}\right)^{2-\eta}\right) - \frac{6\pi^2}{2^5}\left(\frac{Cd}{L}\right)^{2-\eta} - 2\left(\frac{Cd}{L}\right)^2\right]^L \\ &\geq \left[1 - \left(\kappa_1 + 3 + \frac{6\pi^2}{2^5}\right)\left(\frac{Cd}{L}\right)^{2-\eta}\right]^L \end{aligned}$$

Let $\kappa_2 := \kappa_1 + 3 + \frac{6\pi^2}{2^5}$ for brevity. Then

$$P_{\text{total}} \geq \left[1 - \kappa_2 \left(\frac{Cd}{L}\right)^{2-\eta}\right]^L \geq \left(1 - \frac{(\kappa_2 \cdot (Cd)^{2-\eta}/L^{1-\eta})^2}{L}\right) \cdot e^{-\kappa_2 \cdot (Cd)^{2-\eta}/L^{1-\eta}},$$

since $\left(1 - \frac{x}{n}\right)^n \geq \left(1 - \frac{x^2}{n}\right) \cdot e^{-x}$ for $|x| \leq n$ and $n > 1$. For this case we take $x = \kappa_2 \cdot \frac{(Cd)^{2-\eta}}{L^{1-\eta}}$, $n = L$ and the requirements are satisfied when L is sufficiently large, which we will choose below.

Set the number of stages to

$$L = (Cd)^{(2-\eta)/(1-\eta)}. \tag{5.20}$$

Then the probablity of success becomes

$$P_{\text{total}} \geq \left(1 - \frac{\kappa_2^2}{L}\right) \cdot e^{-\kappa_2} = \left(1 - \frac{\kappa_2^2}{(Cd)^{(2-\eta)/(1-\eta)}}\right) \cdot e^{-\kappa_2} \geq \frac{1}{2} \cdot e^{-\kappa_2} = \Omega(1),$$

for d sufficiently large. Hence, the choice of L in (5.20) guarantees that the algorithm has constant probability of success. Moreover, we can get $P_{\text{total}} \geq 3/4$ if we repeat the algorithm and choose the median as the final result.

5.3.2.5 Cost

The Suzuki splitting method expresses the exponentials $W_{h,\ell}^{2^j} = e^{-iM_{h,\ell}/R}$, in terms of exponentials involving either $-\Delta_h$ or V_h. The exponentials involving $-\Delta_h$ can be implemented in $O(d \log^2 h^{-1})$ quantum operations [41]. Furthermore the exponentials involving V_h can be implemented with two bit queries. Approximately half of the exponentials involve $-\Delta_h$ and half involve V_h. Consequently the number of exponentials provides a good estimate on the cost of the algorithm.

Let $N_{j,\ell}$ be the number of exponentials used to simulate $W_{h,\ell}^{2^j}$, let N_ℓ be the number of the exponentials required for the ℓth stage and let N be the total number of exponentials required for the algorithm. Then

$$N_\ell = \sum_{j=0}^{t_0+b-1} N_{j,\ell} \leq \sum_{j=0}^{t_0+b-1} 2 \cdot 5 \cdot 5^{k-1} \cdot \| -\Delta_h/R \|_2 \cdot 2^j \left(\frac{4e \cdot 2 \cdot 2^j \| \frac{\ell}{L} \cdot \frac{V_h}{R} \|_2}{\varepsilon_{j,\ell}^S} \right)^{1/(2k)} \cdot \frac{4e \cdot 2}{3} \left(\frac{5}{3}\right)^{k-1},$$

where k determines the order of the Suzuki splitting method [56]. However $\| -\Delta_h/R \|_2 \leq 1$ and $\| \frac{\ell}{L} \cdot \frac{V_h}{R} \| \leq \frac{\ell}{L} \cdot \frac{C}{3dh^{-2}}$, which leads to

$$\begin{aligned} N_\ell &\leq \sum_{j=0}^{t_0+b-1} \frac{80e}{3} \cdot 5^{k-1} \cdot \left(\frac{5}{3}\right)^{k-1} \cdot 1 \cdot 2^j \cdot \left(\frac{4e \cdot 2 \cdot 2^j \cdot \frac{\ell}{L} \cdot \frac{C}{3dh^{-2}}}{2^{j-(b+t_0)} \cdot (Cd)^{-2/(1-\eta)}} \right)^{1/(2k)} \\ &\leq \frac{80e}{3} \cdot 5^{k-1} \cdot \left(\frac{5}{3}\right)^{k-1} \cdot \left(\frac{8e}{3}\right)^{1/(2k)} \cdot \left(2^{b+t_0}\right)^{1/(2k)} \cdot C^{\left(1+\frac{2}{1-\eta}\right)/(2k)} \cdot d^{\left(\frac{2}{1-\eta}-1\right)/(2k)} \\ &\quad \cdot h^{1/k} \cdot \sum_{j=1}^{t_0+b-1} 2^j \end{aligned}$$

$$\leq \frac{80e}{3} \cdot \left(\frac{25}{3}\right)^{k-1} \cdot \left(\frac{8e}{3}\right)^{1/(2k)} \cdot (24\pi)^{1+\frac{1}{2k}} \cdot C^{\frac{2-\eta}{1-\eta}+\frac{5-2\eta}{2k(1-\eta)}} \cdot d^{\frac{2-\eta}{1-\eta}+\frac{3}{2k(1-\eta)}} \cdot h^{-(3+\frac{1}{2k})} \cdot \left(\frac{\ell}{L}\right)^{1/(2k)},$$

since $\sum_{j=0}^{t_0+b-1} 2^j \leq 2^{t_0+b} \leq 24\pi \cdot (Cd)^{\frac{2-\eta}{1-\eta}} \cdot h^{-3}$. Denote by $c(k)$ the constant on the expression above that depends on k, namely $c(k) := \frac{80e}{3} \cdot \left(\frac{25}{3}\right)^{k-1} \cdot \left(\frac{8e}{3}\right)^{1/(2k)} \cdot (24\pi)^{1+\frac{1}{2k}}$. We have

$$\begin{aligned} N = \sum_{\ell=1}^{L} N_\ell &\leq c(k) \cdot C^{\frac{2-\eta}{1-\eta}+\frac{5-2\eta}{2k(1-\eta)}} \cdot d^{\frac{2-\eta}{1-\eta}+\frac{3}{2k(1-\eta)}} \cdot h^{-(3+\frac{1}{2k})} \cdot \sum_{\ell=1}^{L} \left(\frac{\ell}{L}\right)^{1/(2k)} \\ &\leq c(k) \cdot C^{\frac{2-\eta}{1-\eta}+\frac{5-2\eta}{2k(1-\eta)}} \cdot d^{\frac{2-\eta}{1-\eta}+\frac{3}{2k(1-\eta)}} \cdot h^{-(3+\frac{1}{2k})} \cdot L \\ &\leq c(k) \cdot C^{\frac{4-2\eta}{1-\eta}+\frac{5-2\eta}{2k(1-\eta)}} \cdot d^{\frac{4-2\eta}{1-\eta}+\frac{3}{2k(1-\eta)}} \cdot h^{-(3+\frac{1}{2k})}. \end{aligned}$$

For *relative error* $O(\varepsilon)$, it suffices to set $h \leq \varepsilon$. In that case

$$N \leq c(k) \cdot \varepsilon^{-(3+\frac{1}{2k})} \cdot C^{\frac{4-2\eta}{1-\eta}+\frac{5-2\eta}{2k(1-\eta)}} \cdot d^{\frac{4-2\eta}{1-\eta}+\frac{3}{2k(1-\eta)}}. \tag{5.21}$$

The eigenvalues and eigenvectors of the discretized Laplacian are known and the evolution of a system with a Hamiltonian involving $-\Delta_h$ can be implemented with $d \cdot O(\log^2 \varepsilon^{-1})$ quantum operations using the Fourier transform in each dimension; see e.g., [47, pg. 209]. The evolution of a system with a Hamiltonian involving V_h can be implemented using two quantum queries and phase kickback. Hence the number of quantum operations required to implement the algorithm is proportional to

$$c(k) \cdot \varepsilon^{-(3+\frac{1}{2k})} \cdot C^{\frac{4-2\eta}{1-\eta}+\frac{5-2\eta}{2k(1-\eta)}} \cdot d^{1+\frac{4-2\eta}{1-\eta}+\frac{3}{2k(1-\eta)}}.$$

The analysis leads to the following theorem.

Theorem 5. *Consider the ground state estimation problem for the time-independent Schrödinger equation* (5.1), (5.2). *The repeated phase estimation procedure of* $L = (Cd)^{(2-\eta)/(1-\eta)}$ *stages with*

- *Number of qubits: The upper register has* $q = 3\log\varepsilon^{-1} + \frac{2-\eta}{1-\eta}\log(Cd) + O(1)$ *qubits, while the lower register has* $\Theta\left(d\log\varepsilon^{-1}\right)$ *qubits.*
- *Input state: The upper register is initialized to* $|0\rangle^{\otimes q}$. *The lower register of the first stage is initialized to* $|\psi_{-\Delta_h}\rangle$. *Furthermore we set* $|\psi_{\text{in},\ell}\rangle := |\psi_{\text{out},\ell-1}\rangle$ *for* $\ell = 2, 3, \ldots, L$.

- *Implementation of exponentials: Implement each exponential* $W_{h,\ell}^{2^j}$ *using Suzuki splitting formulas of order* $2k+1$ *with simulation error* $\varepsilon_{j,\ell}^S = 2^{j-q} \cdot (Cd)^{-2/(1-\eta)}$*, for* $j = 0, 1, \ldots, q-1$*, and* $\ell = 1, 2, \ldots, L$*.*

approximates the ground state energy E_0 *with relative error* $O(\varepsilon)$*, for* $\varepsilon = o(d^{-2})$*, using a number of bit queries proportional to*

$$c(k) \cdot \varepsilon^{-(3+\frac{1}{2k})} \cdot C^{\frac{4-2\eta}{1-\eta}+\frac{5-2\eta}{2k(1-\eta)}} \cdot d^{\frac{4-2\eta}{1-\eta}+\frac{3}{2k(1-\eta)}}$$

and a number of quantum operations proportional to

$$c(k) \cdot \varepsilon^{-(3+\frac{1}{2k})} \cdot C^{\frac{4-2\eta}{1-\eta}+\frac{5-2\eta}{2k(1-\eta)}} \cdot d^{1+\frac{4-2\eta}{1-\eta}+\frac{3}{2k(1-\eta)}},$$

where $c(k) := \frac{80e}{3} \cdot \left(\frac{25}{3}\right)^{k-1} \cdot \left(\frac{8e}{3}\right)^{1/(2k)} \cdot (24\pi)^{1+\frac{1}{2k}}$*, with constant probability of success.*

The final state on the lower register $|\psi_{\text{out,L}}\rangle$ has overlap

$$|\langle \psi_{\text{out,L}} | u_0 \rangle|^2 \geq 1 - O\left((Cd)^{-\frac{2-\eta}{1-\eta}}\right).$$

with the ground state eigenvector $|u_0\rangle$ of M_h.

5.3.3 A quantum algorithm approximating the ground state eigenvector

In Section 5.3.2 we demonstrated a quantum algorithm estimating the ground state energy of the Hamiltonian H (eq. (5.1),(5.2)). It turns out we can use the same algorithm with different parameters, to estimate the ground state eigenvector of the discretized Hamiltonian M_h, while we also estimate the ground state energy of H with relative error $O(\varepsilon)$.

The following algorithm

- estimates the ground state energy of the Hamiltonian H with relative error ε,
- approximates the ground state $|u_0\rangle$ of the discretized Hamiltonian M_h with a state $|\psi\rangle$ such that

$$|\langle u_0 | \psi \rangle|^2 \geq 1 - O(\delta),$$

where $0 < \delta < 1$ a parameter.

We remark that for $\delta = \Omega((Cd)^{-\frac{2-\eta}{1-\eta}})$ we can use the algorithm in Section 5.3.2. From now on we assume that $\delta = o((Cd)^{-\frac{2-\eta}{1-\eta}})$.

5.3.3.1 Error analysis

We work as in Section 5.3.2.1 to get the same number of qubits b as in (5.13), namely

$$b = \left\lceil \log \frac{2R\pi}{dh} \right\rceil = \left\lceil \log(10\pi h^{-3}) \right\rceil = \log \Theta\left(h^{-3}\right).$$

5.3.3.2 Success probability

Equations (5.18) and (5.19) remain the same. What changes is the number of stages L. Since we require

$$|\langle \psi_{\text{out,L}} | u_0 \rangle|^2 \geq 1 - O(\delta)$$

we set

$$(Cd/L)^{2-\eta} = \delta \Rightarrow L = Cd/\delta^{1/(2-\eta)}. \tag{5.22}$$

Just as before, the success probability of the algorithm after L stages is

$$P_{\text{total}} \geq \left(1 - \frac{(\kappa_2 \cdot (Cd)^{2-\eta}/L^{1-\eta})^2}{L}\right) \cdot e^{-\kappa_2 \cdot (Cd)^{2-\eta}/L^{1-\eta}} = \left(1 - \frac{o(1)}{L}\right) \cdot e^{-o(1)} \geq 3/4,$$

according to our choice of L in (5.22) and the fact that $\delta = o\left((Cd)^{-\frac{2-\eta}{1-\eta}}\right)$, since for larger δ we can use the algorithm in Section 5.3.2 as we pointed out.

5.3.3.3 Simulation error

Just like before, we pick

$$\varepsilon^S_{j,\ell} := 2^{j-(b+t_0)}/(L/Cd)^2,$$

which according to our choice of L becomes

$$\varepsilon^S_{j,\ell} = 2^{j-(b+t_0)}\delta^{2/(2-\eta)}.$$

Note that $2 \cdot \sum_{j=0}^{b+t_0-1} \varepsilon^S_{j,l} = O\left(\delta^{2/(2-\eta)}\right)$, which is asymptotically smaller than $O\left(\left(\frac{Cd}{L}\right)^{2-\eta}\right) = O(\delta)$.

5.3.3.4 Cost

We work as before. Using the bounds on the number of exponentials [56] required to simulate $W^{2^j}_{h,\ell}$ we have

$$N_\ell = \sum_{j=0}^{t_0+b-1} N_{j,\ell} \leq \sum_{j=0}^{t_0+b-1} 2 \cdot 5 \cdot 5^{k-1} \cdot \|\Delta_h/R\|_2 \cdot 2^j \left(\frac{4e \cdot 2 \cdot 2^j \|\frac{\ell}{L} \cdot \frac{V_h}{R}\|_2}{\varepsilon^S_{j,\ell}}\right)^{1/(2k)} \cdot \frac{4e \cdot 2}{3}\left(\frac{5}{3}\right)^{k-1},$$

where k determines the order of the Suzuki splitting method. Since $\|\Delta_h/R\|_2 \leq 1$ and $\|\frac{\ell}{L} \cdot \frac{V_h}{R}\| \leq \frac{\ell}{L} \cdot \frac{C}{3dh^{-2}}$ we have

$$\begin{aligned}
N_\ell &\leq \frac{80e}{3} \cdot 5^{k-1} \cdot \left(\frac{8e}{3}\right)^{1/2k} \cdot \left(\frac{5}{3}\right)^{k-1} \cdot 2^{(b+t_0)/(2k)} \cdot \left(\frac{\ell}{L}\right)^{1/2k} \cdot \left(\frac{C/(dh^{-2})}{\delta^{2/(2-\eta)}}\right)^{1/(2k)} \\
&\quad \cdot \sum_{j=0}^{t_0+b-1} 2^j \\
&\leq \frac{80e}{3} \cdot 5^{k-1} \cdot \left(\frac{8e}{3}\right)^{1/2k} \cdot \left(\frac{5}{3}\right)^{k-1} \cdot 2^{\frac{b+t_0}{2k}\left(1+\frac{1}{2k}\right)} \cdot \left(\frac{\ell}{L}\right)^{1/(2k)} \cdot C^{1/(2k)} \cdot d^{-1/(2k)} \cdot h^{1/k} \\
&\quad \cdot \delta^{-\frac{1}{k(2-\eta)}} \\
&\leq \frac{80e}{3} \cdot 5^{k-1} \cdot \left(\frac{8e}{3}\right)^{1/2k} \cdot \left(\frac{5}{3}\right)^{k-1} \cdot (24\pi)^{1+\frac{1}{2k}} \cdot \left(\frac{\ell}{L}\right)^{1/(2k)} \cdot C^{1/(2k)} \cdot d^{-1/(2k)} \\
&\quad \cdot h^{-\left(3+\frac{1}{2k}\right)} \cdot \delta^{-\frac{1}{k(2-\eta)}},
\end{aligned}$$

since $2^b \leq 12\pi h^{-3}$ and $2^{t_0} \leq 2\delta^{-1}$. Once again, denote by $c(k)$ the expression $c(k) := \frac{80e}{3} \cdot 5^{k-1} \cdot \left(\frac{8e}{3}\right)^{1/2k} \cdot \left(\frac{5}{3}\right)^{k-1} \cdot (24\pi)^{1+\frac{1}{2k}}$. The total number of exponentials required is

$$\begin{aligned}
N = \sum_{\ell=1}^{L} N_\ell &\leq c(k) \cdot C^{1/(2k)} \cdot d^{-1/(2k)} \cdot h^{-\left(3+\frac{1}{2k}\right)} \cdot \delta^{-\frac{1}{k(2-\eta)}} \cdot \sum_{\ell=1}^{L} \left(\frac{\ell}{L}\right)^{1/(2k)} \\
&\leq c(k) \cdot C^{1/(2k)} \cdot d^{-1/(2k)} \cdot h^{-\left(3+\frac{1}{2k}\right)} \cdot \delta^{-\frac{1}{k(2-\eta)}} \cdot L \\
&\leq c(k) \cdot C^{1+\frac{1}{2k}} \cdot d^{1-\frac{1}{2k}} \cdot h^{-\left(3+\frac{1}{2k}\right)} \cdot \delta^{-1-\frac{1}{2k}-\frac{1}{2-\eta}-\frac{1}{k(2-\eta)}}
\end{aligned}$$

For *relative error* $O(\varepsilon)$, it suffices to set $h \leq \varepsilon$. In that case

$$N \leq c(k) \cdot C^{1+\frac{1}{2k}} \cdot d^{1-\frac{1}{2k}} \cdot \varepsilon^{-\left(3+\frac{1}{2k}\right)} \cdot \delta^{-1-\frac{1}{2k}-\frac{1}{2-\eta}-\frac{1}{k(2-\eta)}} \tag{5.23}$$

The analysis above leads to the following theorem.

Theorem 6. *Consider the ground state energy eigenvector and ground state eigenvector estimation problem for the time-independent Schrödinger equation* (5.1),(5.2)*. Assume where* $\delta = o((Cd)^{-\frac{2-\eta}{1-\eta}})$*. The repeated phase estimation procedure of* $L = \Theta(Cd \cdot \delta^{-1/(2-\eta)})$ *stages with*

- *Number of qubits: The upper register has* $q = 3\log\varepsilon^{-1} + \log\delta^{-1} + O(1)$ *qubits, while the lower register has* $\Theta(d\log_2\varepsilon^{-1})$ *qubits.*

- *Input state: The upper register is initialized to* $|0\rangle^{\otimes q}$*. The lower register of the first stage is initialized to* $|\psi_{-\Delta_h}\rangle$*. Furthermore we set* $|\psi_{\text{in},\ell}\rangle := |\psi_{\text{out},\ell-1}\rangle$ *for* $\ell = 2, 3, \ldots, L$*.*

- *Implementation of exponentials: Implement each exponential* $W_{h,l}^{2^j}$ *using Suzuki splitting formulas of order* $2k+1$ *with simulation error* $\varepsilon_{j,l}^{S} = 2^{j-q} \cdot \delta^{2/(2-\eta)}$*, for* $j = 0, 1, \ldots, q-1$*.*

approximates the ground state energy E_0 *with relative error* $O(\varepsilon)$*, for* $\varepsilon = o(d^{-2})$*, using a number of bit queries proportional to*

$$c(k) \cdot C^{1+\frac{1}{2k}} \cdot d^{1-\frac{1}{2k}} \cdot \varepsilon^{-\left(3+\frac{1}{2k}\right)} \cdot \delta^{-1-\frac{1}{2k}-\frac{1}{2-\eta}-\frac{1}{k(2-\eta)}}$$

and a number of quantum operations proportional to

$$c(k) \cdot C^{1+\frac{1}{2k}} \cdot d^{2-\frac{1}{2k}} \cdot \varepsilon^{-\left(3+\frac{1}{2k}\right)} \cdot \delta^{-1-\frac{1}{2k}-\frac{1}{2-\eta}-\frac{1}{k(2-\eta)}},$$

where $c(k) := \frac{80e}{3} \cdot 5^{k-1} \cdot \left(\frac{8e}{3}\right)^{1/2k} \cdot \left(\frac{5}{3}\right)^{k-1} \cdot (24\pi)^{1+\frac{1}{2k}}$*, with probability of success at least* $3/4$*. The final state on the lower register* $|\psi_{\text{out,L}}\rangle$ *has overlap*

$$|\langle u_0|\psi_{\text{out,L}}\rangle|^2 \geq 1 - O(\delta),$$

with the ground state eigenvector $|u_0\rangle$ *of* M_h*.*

5.4 Future work

The repeated phase estimation algorithm could be applicable in other classes of potentials. There is, however, an important requirement a potential function must satisfy in order to be a good candidate for the algorithm; there are known lower bounds on the fundamental gap[2] of the Hamiltonians H_ℓ the algorithm runs phase estimation on at each stage. These lower bounds determine the values of the algorithm's parameters. However, there are cases where the established lower bounds on the fundamental gap might not be large enough, forcing the repeated phase estimation algorithm to perform poorly. The number of stages L of the algorithm have to be at least proportional to the reciprocal of the eigenvalue gap. As a result, lower bounds that are exponentially small in d are problematic and result in a

[2]difference between the smallest and second smallest eigenvalues

number of stages of RPE that is exponential in d. If, however, one relinquishes the convexity requirement, one can still derive lower bounds on the fundamental gap, but may involve more parameters about the geometry of the domain, or the potential function [7]. For more information on the fundamental gap see [6; 7; 75].

Another important problem is to estimate the ground state energy of the time-independent Schrödinger equation corresponding to (1.5), as discussed in Chapter 4. However, lower bounds on the fundamental gap could be more difficult to derive.

Finally, given lower bounds on the fundamental gap for other classes of potentials, one may study other ways to choose the Hamiltonians $H_1, H_2, \ldots, H_L$ for each stage of the algorithm. We remark that classes of potentials with larger bounds on the fundamental gap are better candidates for our method. Hence, one may approach the target Hamiltonian initially considering Hamiltonians with larger bounds on the fundamental gap, and for the final stages use Hamiltonians involving the potentials with lower bounds in the fundamental gap. This approach could provide better bounds on the cost of the algorithm.

Chapter 6

Circuit design for the quantum NAND evaluation algorithm

6.1 Introduction

Consider a boolean NAND formula ϕ with variables $x_1, x_2, \ldots, x_N$, with N a positive integer. Such formulas consist of NAND gates and variables $x_i \in \{0, 1\}$, for $i = 1, 2, \ldots, N$. If the same variable appears multiple times in the formula, we treat each occurrence separately. Thus N counts multiply occurring variables multiple times. Every boolean formula corresponds to a tree whose internal vertices are NAND gates.

The quantum algorithm for balanced boolean formulas [4, Fig. 2] evaluates a balanced NAND formula of input size N, using a quantum walk on a binary tree. We implement a modified version of the algorithm on a tree of fanin-k, with $k \in \{2, 3, \ldots\}$ a constant.

We design two quantum circuits for the above version of the algorithm. The circuit in Section 6.4 implements the algorithm using arbitrary single qubit gates and their controlled variants, CNOT and Toffoli gates. The total number of gates it requires is $O(\sqrt{N} \cdot \log N)$. The circuit in Section 6.5 uses gates from the Clifford group and the $\pi/8$ gate. When gates from Section 6.4 can be implemented exactly using gates from the Clifford group and the $\pi/8$ gate, we provide the quantum circuits. Otherwise, we employ the Solovay-Kitaev algorithm to implement the single gates approximately. In order to approximate a single qubit transformation with error ε_0, Solovay-Kitaev requires $O(\log^c 1/\varepsilon_0)$ gates, where $c = 3.97$; see

[18]. Our design requires $O(\sqrt{N} \cdot \log N \cdot \log^c(N/\varepsilon))$ gates from the Clifford group and T gates.

The results of this chapter are based on work in [57].

6.2 Prior work

One can classically compute the evaluation of a balanced binary AND-OR tree with zero error in expected time $O(N^{0.754})$ using alpha-beta pruning; see [58; 62]. This algorithm is optimal, even for bounded error classical algorithms [59].

In 2008, Farhi et.al. presented a continuous-time quantum walk based algorithm [25] for the balanced NAND formulas corresponding to a complete binary tree. The algorithm has a runtime of $O(\sqrt{N})$ in the Hamiltonian oracle model [26] and is optimal in the continuous-time query setting, since the lower bound on the quantum setting is $\Omega(\sqrt{N})$; see [9]. Moreover, it can be converted in the more conventional discrete quantum oracle query model, with cost $O(N^{1/2+o(1)})$; see [15]. Later, in 2010, Ambainis et.al. showed an optimal discrete-time coin walk based quantum algorithm evaluating a formula ϕ corresponding to a complete binary tree using $O(\sqrt{N})$ queries to the input oracle [4, Fig. 2].

Recently the implementation of quantum walks has attracted interest. In 2009, Douglas and Wang demonstrated quantum circuits for quantum walks on graphs such as circle and glued trees [24]. Later, Chiang et.al. demonstrated a method implementing quantum walks corresponding to arbitrary sparse classical random walks [13].

6.3 Discrete-time quantum coin walk

The algorithm presented in [4] evaluates a balanced NAND formula of input size N, using a quantum walk on a binary tree. We implement a modified version of the algorithm on a tree of fanin-k. For $k = 2$ the following algorithm is the one presented in [4, Fig. 2] for binary NAND trees.

The algorithm of interest runs phase estimation[1] on top of a quantum walk as follows:

[1] see Section 4.2 for more on the phase estimation algorithm

- **Initialization**: Let $M = O(\sqrt{N})$. Prepare three quantum registers in the state

$$\left(\frac{1}{\sqrt{M}}\sum_{i=1}^{M-1}(-i)^t|t\rangle\right)\otimes|r''\rangle|\text{rightmost}\rangle.$$

  The first register holds the counter for the phase estimation, the second register holds the index for the vertex, and the final register holds the direction. There are $k+1$ possible directions, k directions "upwards", denoted by $c_i \in \text{UP} = \{\text{rightmost}, \ldots, \text{leftmost}\}$, and one down direction.

- **Quantum walk**. If the first register is $|t\rangle$, perform t steps of the following discrete-time quantum walk. Let $|v\rangle|c\rangle$ the last two registers.

  1. If the vertex is a leaf, then apply the phase flip $(-1)^{f(u)}$, using the controlled oracle $O_f|j\rangle|b\rangle = |j\rangle|b \oplus f(j)\rangle$, where $j = 0, \ldots, N-1$ enumerates the leaves, and f is the input function.
  2. If $|v\rangle$ is an internal degree $k+1$ vertex, then apply the operator

$$R_{|u\rangle} = 2|u\rangle\langle u| - I,$$

     where $|u\rangle$ is the equal superposition among all possible directions, namely

$$|u\rangle = \frac{1}{\sqrt{k+1}}(|\text{down}\rangle + |\text{leftmost}\rangle + \cdots + |\text{rightmost}\rangle).$$

  3. If $|v\rangle = |r'\rangle$ apply the diffusion operator

$$R_{|u'\rangle} = 2|u'\rangle\langle u'| - I,$$

     where

$$|u'\rangle = \frac{1}{N^{1/4}}|\text{down}\rangle + \sqrt{1 - \frac{1}{N^{1/2}}}|\text{rightmost}\rangle.$$

  4. If $|v\rangle = |r''\rangle$ do nothing.

- **Walk step**.

  – If $|c\rangle = |\text{down}\rangle$, then walk down to the parent of the vertex and set $|c\rangle$ to the direction it came from.

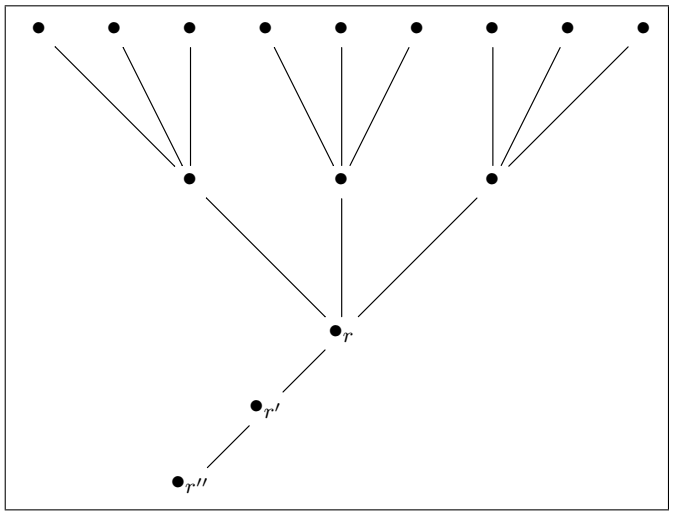


Figure 6.1: An example of a graph the algorithm in Section 6.3 is applied on.

  - If $|c\rangle = |c_i\rangle$ with $c_i \in \mathrm{UP}$, then walk 'up' to the corresponding child of $|v\rangle$ and set $|c\rangle = |\mathrm{down}\rangle$.

- **Phase estimation**. Apply the inverse Fourier transform on the first register and measure in the computational basis. Return 0 if and only if the outcome is 0 or $M/2$.

The algorithm is applied to a graph consisting of a complete k-ary tree and a line of two vertices, r' and r''. The vertex r' is connected to the root of the tree r, and r'' is connected to the vertex r'. An example of a graph of this form is provided in Figure 6.1.

6.4 Quantum circuit using arbitrary single qubit, CNOT and Toffoli gates

In this section we present an asymptotically optimal quantum circuit that implements the algorithm using arbitrary single qubit gates and their controlled versions, CNOT and Toffoli gates.

6.4.1 Labelling scheme

The coin is labelled using $b := \lceil \log_2(k+1) \rceil$ as in Table 6.1.

Let $\mathcal{C}$ denote the set of the coin labels. Clearly $\mathcal{C} \subseteq \{0,1\}^b$.

Direction	Value	Binary Representation
down	$2^b - 1$	1^b
rightmost	0	0^b
$\vdots$	$\vdots$	$\vdots$
leftmost	$k-1$	$\mathrm{Bin}(k-1)$

Table 6.1: Encoding of the coin states for the coin walk on a k-ary tree. $\mathrm{Bin}(x)$ denotes the binary representation of the value x.

Now we label the vertices of the tree. Let L denote the total levels of the tree, not counting the root, namely $L = \log_k N$. Each vertex is assigned a basis state of the space $\mathbb{C}^{(L+1)b+1}$. We set the first qubit to $|0\rangle$ for any tree vertex, in order to differentiate it from r' and r'' whose first qubits are set to $|1\rangle$. We label each vertex using the labels of the directions in the path from the root to the vertex. Namely, the vertex label represents the path from the root leading to it.

For example, a vertex in the j-th level of the tree is represented by $(L+1)b+1$ qubits of the form

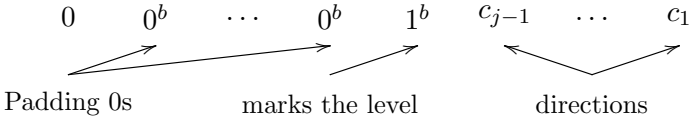


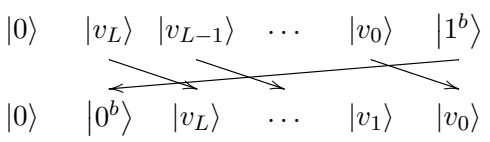
Figure 6.2: Label of a vertex in the j-th level of the tree.

where $c_i \in \mathcal{C}$, for $i = 0, \ldots, j-1$.

6.4.2 Coin walk on a complete k-ary tree

Let $|v\rangle = |0 v_L v_{L-1} \ldots v_0\rangle$, where $v_i \in \mathcal{C}$ for $i = 0, 1, \ldots, L$, be the label of a vertex in the tree and $|c\rangle$ the coin. According to the labelling scheme, the following transformations describe the walk on the tree

- Moving down



- Moving up towards direction c_i (the first $b+1$ qubits must be 0, since v belongs in the tree, and simultaneously $v =$ leaf and $c =$ down is impossible).

$$\begin{array}{ccccccc} |0\rangle & |0^b\rangle & |v_{L-1}\rangle & \cdots & |v_0\rangle & |c_i\rangle & \\ |0\rangle & |v_{L-1}\rangle & |v_{L-2}\rangle & \cdots & |c_i\rangle & |1^b\rangle & \end{array}$$

The circuit implementing the walk on the tree is presented below. It requires one auxiliary qubit.

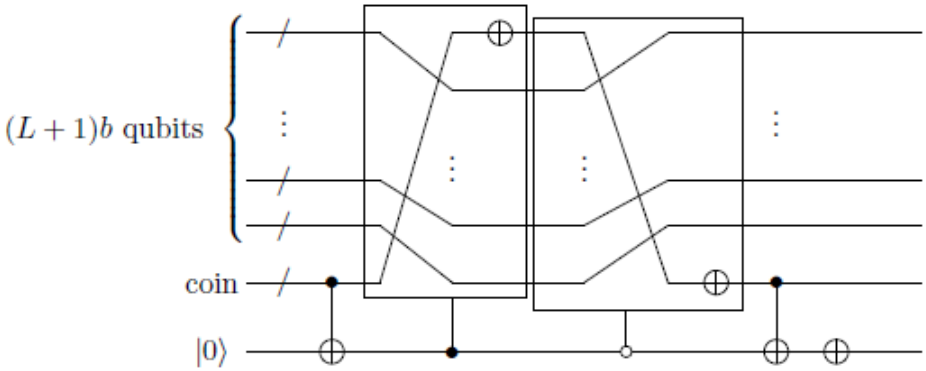


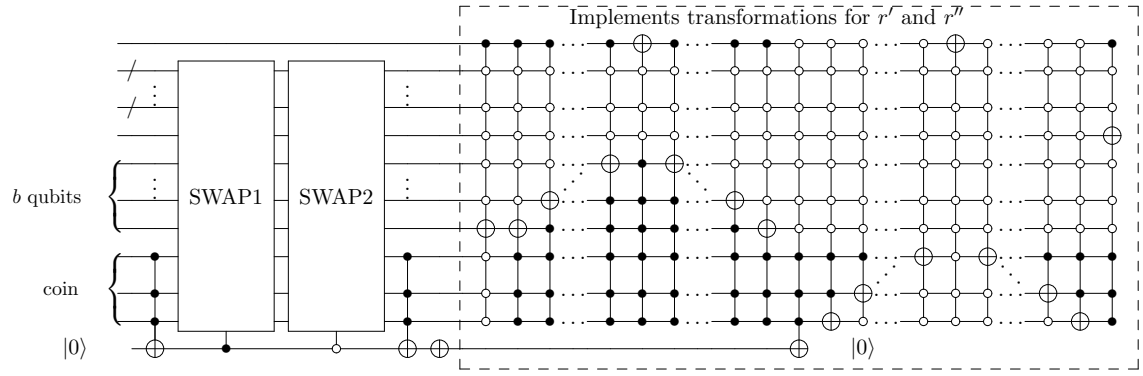
Figure 6.3: The circuit implementing the quantum walk on a tree.

6.4.3 Walking on the graph

Next we implement the walk on the vertices r' and r''. First we label $|r'\rangle$ and $|r''\rangle$ by $\left|10^{(L+1)b}\right\rangle$ and $\left|10^{(L+1)b-1}1\right\rangle$ respectively. The transformations that are required are

1. $|r'\rangle|\text{down}\rangle$ is transformed to $\left|10^{(L+1)b}\right\rangle\left|0^b\right\rangle$. It should be replaced with $\left|10^{(L+1)b-1}1\right\rangle\left|0^b\right\rangle$.
2. $|r'\rangle|\text{rightmost}\rangle$ is transformed to $\left|10^{(L+1)b}\right\rangle\left|1^b\right\rangle$. It should be replaced with $\left|0^{Lb+1}1^b\right\rangle\left|1^b\right\rangle$.
3. $|\text{root}\rangle|\text{down}\rangle$ is transformed to $\left|0^{(L+1)b+1}\right\rangle\left|1^b\right\rangle$. It should be replaced with $\left|10^{(L+1)b}\right\rangle\left|0^b\right\rangle$ and the auxiliary qubit has to be reset from $|1\rangle$ to $|0\rangle$.
4. $|r''\rangle|\text{rightmost}\rangle$ is transformed to $\left|10^{Lb-1}10^b\right\rangle\left|1^b\right\rangle$. It should be replaced with $\left|10^{(L+1)b}\right\rangle\left|1^b\right\rangle$.

Since there are only two vertices in the line, the transformations can be implemented using Gray codes [47, pp. 191–194]. Hence the circuit implementing one step of the discrete-time quantum walk is



6.4.4 Diffusion

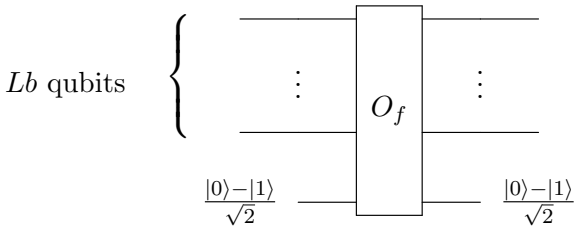
There are three cases of diffusion depending on the vertex $|v\rangle$ of the tree the algorithm operates on.

6.4.4.1 Diffusion on the leaves

In order to implement the diffusion on the leaves, an input oracle O_f,

$$O_f|j\rangle|0\rangle = |j\rangle|f(j)\rangle, \quad j = 0, 1, \ldots, N-1$$

is required. Note that after proper re-indexing, the order of the leafs is revealed by the Lb least significant qubits of the vertex label. We use phase kickback in order to introduce the $(-1)^{f(j)}$ factor in the amplitude [39, Chp. 6.2]. The circuit is

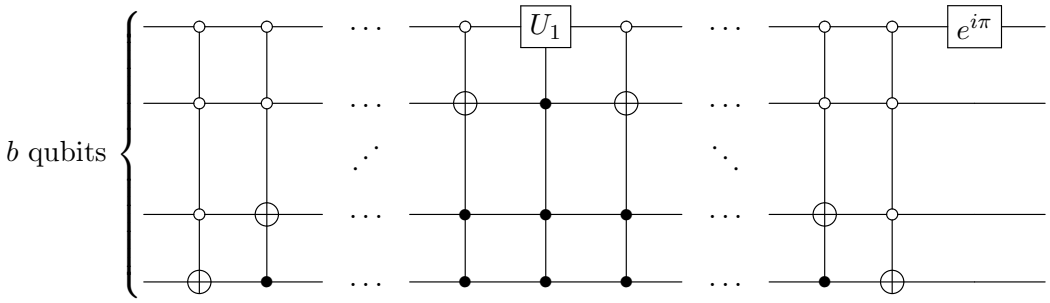


6.4.4.2 Diffusion on r'

The diffusion operator is the $2^b \times 2^b$ matrix

$$R_{|u'\rangle} = -I \begin{pmatrix} \frac{2}{k^{L/2}} - 1 & & -\frac{2}{k^{L/4}}\sqrt{1-\frac{1}{k^{L/2}}} \\ & I & \\ -\frac{2}{k^{L/4}}\sqrt{1-\frac{1}{k^{L/2}}} & & 1-\frac{2}{k^{L/2}} \end{pmatrix}$$

The operator $R_{|u'\rangle}$ can be implemented by the circuit



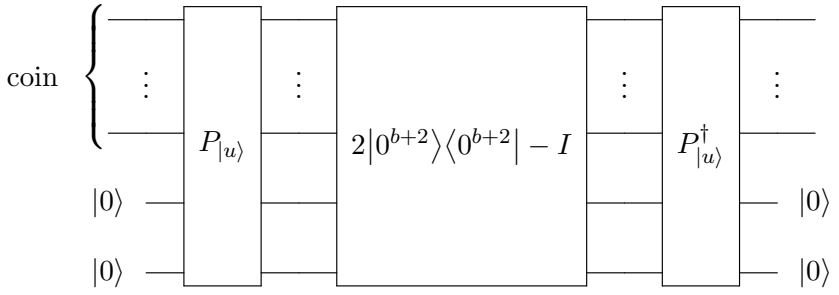
using Gray codes, where

$$U_1 = \begin{pmatrix} \frac{2}{k^{L/2}} - 1 & -\frac{2}{k^{L/4}}\sqrt{1-\frac{1}{k^{L/2}}} \\ -\frac{2}{k^{L/4}}\sqrt{1-\frac{1}{k^{L/2}}} & 1-\frac{2}{k^{L/2}} \end{pmatrix}$$

is a single qubit gate. U_1 can be expressed in terms of single qubit rotations and possibly a global phase [47, Thm. 4.1]. In Section 6.5, we express U_1 in terms of Clifford and T gates, using the Solovay Kitaev algorithm [18].

6.4.4.3 Diffusion on internal vertices

The diffusion on the internal vertices is a reflection around the state that denotes the equal superposition of all the possible coin outcomes, namely $|u\rangle = \frac{1}{\sqrt{k+1}} \sum_{j \in \mathcal{C}} |j\rangle$. It can be implemented as



using $P_{|u\rangle}$, a circuit to prepare $|u\rangle$ using two auxiliary qubits, presented below.

The unitary $P_{|u\rangle}$ can be implemented as follows. Starting with $\left|0^b\right\rangle|0\rangle$ we create the equal superposition state $\frac{1}{2^{b/2}}\sum_{j=0}^{2^b-1}|j\rangle|0\rangle$. We next apply the quantum bit query O_C that represents the characteristic function for the set $\mathcal{C}$, namely $O_C|j\rangle|0\rangle = |j\rangle|\chi_{\mathcal{C}}(j)\rangle$, to get $|\psi\rangle = \frac{1}{2^{b/2}}\sum_{j=0}^{2^b-1}|j\rangle|\chi_{\mathcal{C}}(j)\rangle$, which can be written as

$$\frac{\sqrt{k+1}}{2^{b/2}}\left(\frac{1}{\sqrt{k+1}}\sum_{j\in\mathcal{C}}|j\rangle|1\rangle\right) + \frac{\sqrt{2^b-(k+1)}}{2^{b/2}}\left(\frac{1}{\sqrt{2^b-(k+1)}}\sum_{j\notin\mathcal{C}}|j\rangle|0\rangle\right)$$

Let

$$|\text{good}\rangle = |u\rangle = \frac{1}{\sqrt{k+1}}\sum_{j\in\mathcal{C}}|j\rangle$$

and

$$|\text{bad}\rangle = \frac{1}{\sqrt{2^b-(k+1)}}\sum_{j\notin\mathcal{C}}|j\rangle.$$

As a result

$$|\psi\rangle = \frac{\sqrt{k+1}}{2^{b/2}}|\text{good}\rangle|1\rangle + \frac{\sqrt{2^b-(k+1)}}{2^{b/2}}|\text{bad}\rangle|0\rangle$$

Note that the states $|\text{good}\rangle$ and $|\text{bad}\rangle$ are orthonormal. We construct the state $|\text{good}\rangle$ applying amplitude amplification with known amplitudes[2] [28, Sec. 2.1] to turn the state $|\psi\rangle$ to $|\text{good}\rangle$. More specifically, the preparation of $|u\rangle$ works as follows

1. Append a qubit initialized to $|0\rangle$

2. Perform the single qubit rotation

$$R_1(k) = \begin{pmatrix} \sqrt{1-\frac{\overline{\alpha}}{\alpha}} & -\sqrt{\frac{\overline{\alpha}}{\alpha}} \\ \sqrt{\frac{\overline{\alpha}}{\alpha}} & \sqrt{1-\frac{\overline{\alpha}}{\alpha}} \end{pmatrix}$$

   on the appended qubit, where $\alpha = (k+1)/2^b$, $\overline{\alpha} = \sin^2(\pi/(4\overline{m}+2))$ with $\overline{m} = \lceil \frac{\pi}{4\theta_\alpha} - \frac{1}{2} \rceil$ and $\theta_\alpha = \arcsin(\sqrt{\alpha})$. The rotation $R_1(k)$ is a single qubit $R_y(\theta_k)$ rotation, where $\theta_k = 2\arcsin\sqrt{\frac{\overline{\alpha}}{\alpha}}$. Let $|\psi_{\overline{\alpha}}\rangle$ denote the resulting state.

3. Perform a conditional phase shift, with every state whose last two qubits are $|1\rangle|1\rangle$ receiving a phase factor -1.

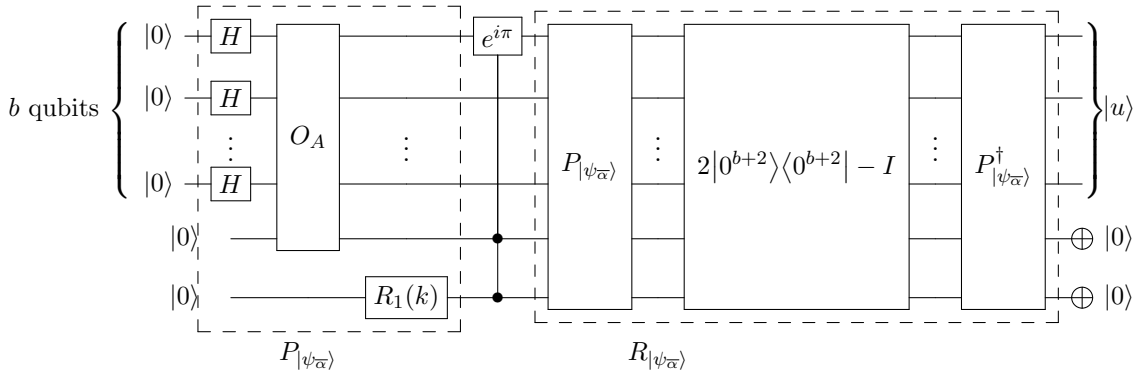
[2] k (and, as a result, b) are known in advance.

4. Rotate the resulting state according to the unitary matrix

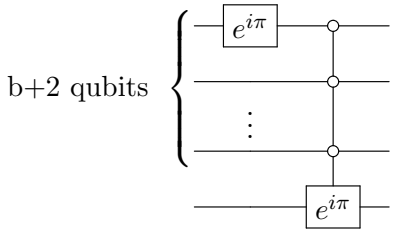
$$R_{\psi_{\overline{\alpha}}} = 2(|\psi_{\overline{\alpha}}\rangle\langle\psi_{\overline{\alpha}}|) - I$$

$\overline{m}$ times, where $\overline{m} = 0$ if $k = 2^b - 1$ and $\overline{m} = 1$ otherwise.

Hence, the circuit implementing $P_{|u\rangle}$ is

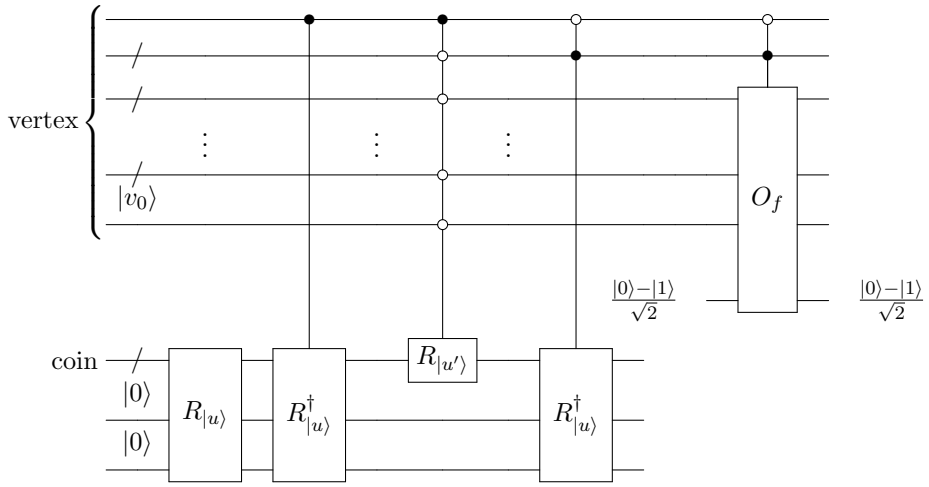


The reflection operator $2|0^{b+2}\rangle\langle 0^{b+2}| - I$ retains the phase for $|0^{b+2}\rangle$, while it introduces a factor of -1 on the amplitude of every other basis state. It can be implemented using an auxiliary qubit as follows



6.4.4.4 The circuit implementing the diffusion

Combining the diffusion circuits together, the circuit implementing the diffusion is



6.4.5 Initial state preparation

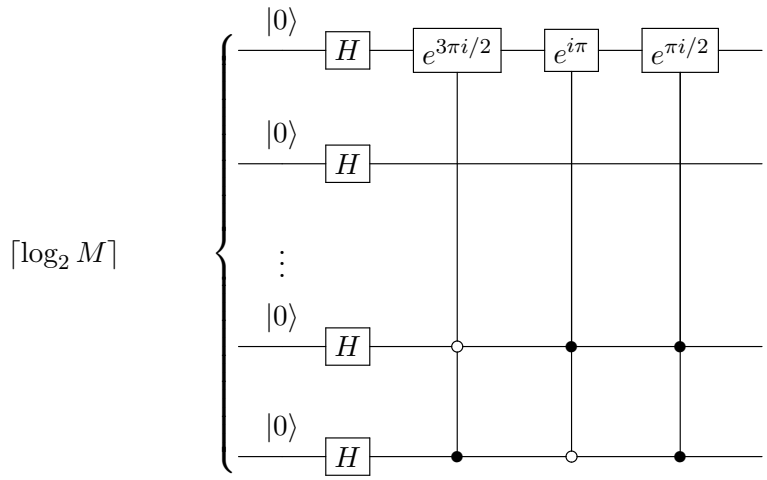
The initial state is

$$\left(\frac{1}{\sqrt{M}}\sum_{t=0}^{M-1}(-i)^t|t\rangle\right)|\mathrm{r}''\rangle|\mathrm{right}\rangle = \left(\frac{1}{\sqrt{M}}\sum_{t=0}^{M-1}(-i)^t|t\rangle\right)\left|110^{(L+1)b-1}\right\rangle\left|0^b\right\rangle,$$

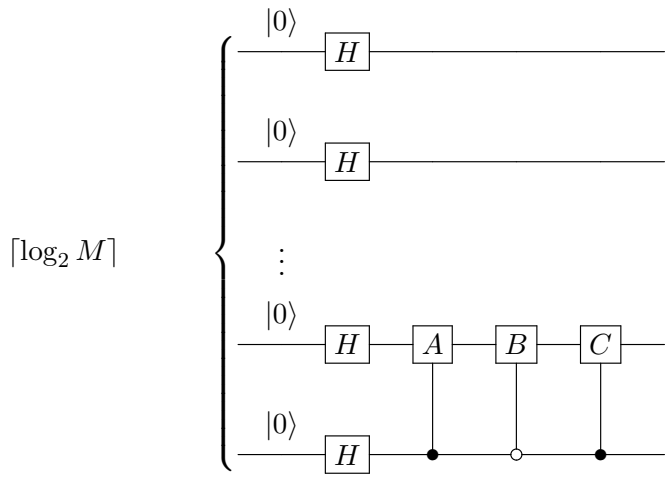
where $M = O(\sqrt{N})$. The only register whose initialization is of interest is the first one, namely $\frac{1}{\sqrt{M}}\sum_{t=0}^{M-1}(-i)^t|t\rangle$. Observe that

$$\frac{1}{\sqrt{M}}\sum_{t=0}^{M-1}(-i)^t|t\rangle = \frac{1}{\sqrt{M}}\sum_{k=0}^{(M/4)-1}(|4k\rangle + (-i)|4k+1\rangle + (-1)|4k+2\rangle + i|4k+3\rangle).$$

Thus the first register is initialized using controlled phase shifts



The circuit above is equivalent to



where

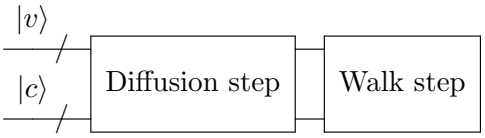
$$A = \begin{bmatrix} e^{3\pi i/2} & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -i & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

and

$$C = \begin{bmatrix} 1 & 0 \\ 0 & e^{\pi i/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = S.$$

6.4.6 Circuit for the algorithm

The circuit implementing the one step of the coin walk V is



where $|v\rangle$ and $|c\rangle$ denote the vertex and coin states respectively.

The algorithm we implement is phase estimation with $\lceil \log_2 M \rceil = O(\log_2 \sqrt{N})$ power queries on V and initial state

$$\left(\frac{1}{\sqrt{M}} \sum_{t=0}^{M-1} (-i)^t |t\rangle \right) |\mathrm{r}''\rangle |\mathrm{right}\rangle.$$

As a result, the circuit implementing the algorithm is

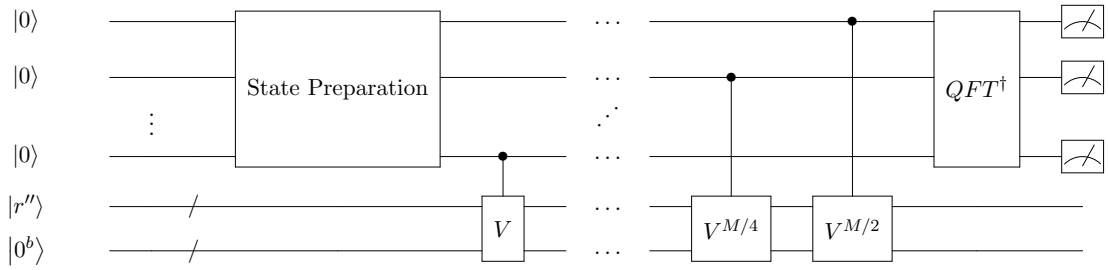


Figure 6.4: The quantum circuit implementing the algorithm of Section 6.3.

6.4.6.1 Circuit cost

Theorem 7. *There exists a quantum circuit of single qubit, controlled versions of single qubit gates, CNOT gates and Toffoli gates that implements the algorithm in Section 6.3 using* $O(\sqrt{N} \cdot \log N)$ *gates.*

Proof. We count separately the type and number of gates used at each part of the circuit constructed in the previous sections.

The state preparation circuit is implemented using

- $O(\log \sqrt{N})$ Hadamard gates,
- 1 single-controlled $\begin{bmatrix} -i & 0 \\ 0 & 1 \end{bmatrix}$ gate,
- 1 single-controlled $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ gate,
- 1 single-controlled $\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ gate.

One step of the walk can be implemented using

- multiply controlled Hadamard gates,
- multiply controlled single qubit gates such as $R_1(k)$ gate, U_1 gate, universal phase gate $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$, other phase shift gates such as $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $\begin{bmatrix} -i & 0 \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$, that are also used in state preparation.

- multiply controlled CNOT gates,
- 9 oracle calls on multiply controlled $O_{\mathcal{C}}$ and 1 oracle call on the multiply controlled input oracle O_f.

and the conjugate transpose versions of those gates.

Any multiply controlled on n qubits single qubit gate (denoted by C_n-U) can be implemented using $2n-2$ Toffoli gates, a single controlled U gate and $n-1$ auxilliary qubits [47, pp. 183–184]. Hence we can represent the circuit in terms of single controlled single qubit gates and Toffoli gates. The total number of such elementary gates required to implement one step of the walk is $C_V = O(L) = O(\log N)$.

The controlled applications of V in the phase estimation routine are implemented using $\sum_{j=0}^{\lceil \log_2 M \rceil - 1} 2^j \cdot C_V \leq MC_V = O(\sqrt{N} \cdot \log N)$ elementary gates.

Furthermore, the inverse Fourier transform requires $O(\log^2 M) = O(\log^2 N)$ gates. It consists of Hadamard gates and gates of the form C_1-R_j with

$$R_j = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^j} \end{bmatrix}, \quad \text{for j} = 2, 3, \ldots, \lceil \log_2 \text{M} \rceil.$$

Altogether, the circuit requires $O(\sqrt{N} \cdot \log N)$ elementary gates and $O(\sqrt{N})$ oracle calls on $O_{\mathcal{C}}$ and O_f. □

6.5 Implementation using Clifford and T gates

Consider the Clifford and T gates. This set of gates is universal, in the sense that it is possible, using gates solely from this set, to approximately implement any unitary matrix. Recall that it consists of the Hadamard gate

$$H = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix},$$

the $\pi/8$ gate

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix},$$

the phase gate

$$S = T^2 = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix},$$

the conjugate transpose versions $T^\dagger$ and $S^\dagger$ and finally the CNOT gate. Let $\mathcal{G}$ denote the set of these gates.

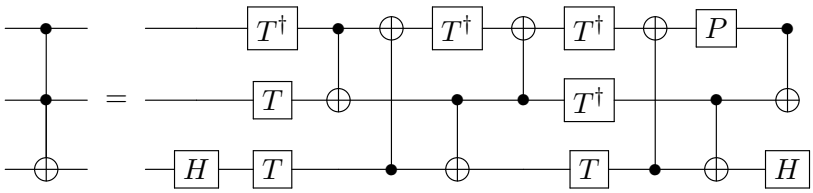
We represent each of the gates used in Section 6.4 using gates from $\mathcal{G}$. Some of them, such as the controlled H or the controlled T gates can be implemented exactly, while others are approximated using the Solovay-Kitaev theorem [47], in a constructive way [18]. The resulting circuit approximates the exact circuit of section 6.4 with error $O(\varepsilon)$.

We remind that any multiply controlled single qubit gate C_n-U (where U acts on 1 qubit and C_n is controlled on n qubits) can be implemented using $2n-2$ Toffoli gates, a single controlled U gate and $n-1$ auxilliary qubits [47, pp. 183–184]. Since Toffoli gates can be implemented exactly using gates from $\mathcal{G}$ (see Section 6.5.1), we only deal with the implementation of the C_1-U gates that appear in Section 6.4.

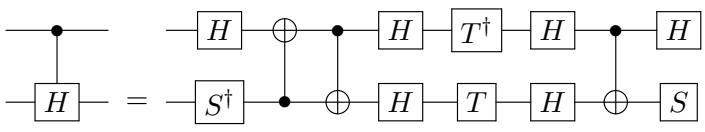
6.5.1 Exact gate implementation

The following gates from the circuit in Section 6.4 can be implemented exactly, using Clifford and T gates:

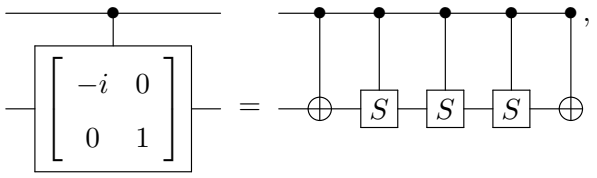
- The Toffoli gate [5]



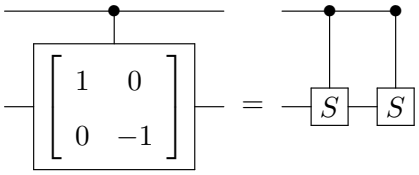
- The C_1-H gate [5]



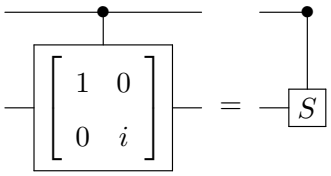
- The C_1-$\begin{bmatrix} -i & 0 \\ 0 & 1 \end{bmatrix}$ gate



the C_1-$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ gate



and C_1-$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ gate



used in initial state preparation.

6.5.2 Approximate implementation of the circuit

The remaining gates, namely C_1-$R_1(k)$, C_1-U_1 and C_1-R_j for $j = 4, 5, \ldots \lceil \log_2 M \rceil$, are likely not exactly implementable by the Clifford group and T gates [42]. For that reason, we employ the Solovay-Kitaev theorem [47]. For any single qubit gate we can construct a circuit [18] of gates from the set $\mathcal{G}$ that approximates the gate within error ε_0 using $O(\log^c \varepsilon_0^{-1})$ gates. Note that if a single qubit gate U is approximated by a circuit within ε_0, the gate C_1-U is approximated within ε_0 as well.

Theorem 8. *There exists a quantum circuit of Clifford and T gates that implements algorithm in Section 6.3 with $O(\varepsilon)$ error, using $O(\sqrt{N} \cdot \log N \cdot \log^c(N/\varepsilon))$ gates.*

Proof. We initially apply Solovay-Kitaev to build quantum circuits comprising of gates in $\mathcal{G}$ that approximate the single qubit gates $R_1(k)$ and U_1 within error $\Theta(\varepsilon/\sqrt{N})$. There is a

constant number of such gates per application of the walk V. Hence the total error for one application of the quantum walk is $\Theta(\varepsilon/\sqrt{N})$. Phase estimation requires $\sum_{j=0}^{\lceil \log_2 M \rceil - 1} 2^j = O(M) = O(\sqrt{N})$ applications of the walk V. Hence the total error before the inverse quantum Fourier transform is $O(\varepsilon)$.

Using the Solovay-Kitaev algorithm we further approximate the inverse Fourier transform. If the inverse Fourier transformation is applied on at least four qubits, it cannot be implemented exactly using our allowed gate set [42]. We approximate each R_j, for $j = 4, 5, \ldots, \lceil \log_2 M \rceil$ gate with error $\Theta(\varepsilon/\log^2 N))$. Note that C_1-$R_2 = C_1$-S and C_1-$R_3 = C_1$-T, which can be implemented exactly as shown later. The total error for the inverse Fourier transform is $O(\varepsilon)$, since it requires $O(\log^2 \sqrt{N})$ gates.

As a result, the final error of the circuit is $O(\varepsilon)$. The probability of success is reduced by at most twice that amount [47, pg. 194], namely $O(\varepsilon)$.

Finally, we calculate the total number of Clifford and T gates used. Recall that the Solovay-Kitaev algorithm requires $O(\log^c(1/\varepsilon_0))$ gates for error ε_0. Hence one step of the quantum walk requires $O(\log^c(\sqrt{N}/\varepsilon))$ gates for approximation of each gate and a total of $O(\log N \cdot \log^c(N/\varepsilon))$ gates. Consequently, the circuit prior to the application of the inverse Fourier transform requires

$$O(\sqrt{N} \cdot \log N \cdot \log^c(N/\varepsilon))$$

gates. The inverse Fourier transform requires

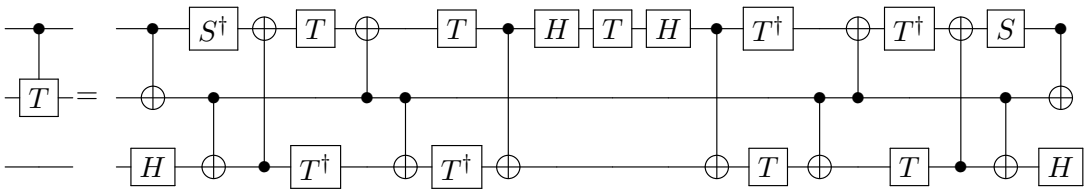
$$O\left(\log^2 N \cdot \log^c\left(\frac{\log^2 N}{\varepsilon}\right)\right)$$

gates. In conclusion, the total number of gates in the set $\mathcal{G}$ that are required for error $O(\varepsilon)$ is

$$O(\sqrt{N} \cdot \log N \cdot \log^c(N/\varepsilon)). \qquad \square$$

Note that to implement the approximation of the controlled gates we have to represent the controlled versions of the H, CNOT, T and S gates using only gates from $\mathcal{G}$. This has already been accomplished for H and CNOT gates. C_1-T is implemented using an auxiliary

qubit [5] as



where the upper qubit is the control, the middle qubit is the target and the bottom qubit is the ancilla. Since $S = T^2$, we implement C_1-S trivially, using the circuit for C_1-T.

Part III

Conclusions

Chapter 7

Conclusions and future work

In this thesis we study algorithms for multivariate problems.

In Chapters 2 and 3 we study linear tensor product problems in the worst and average case setting respectively. We provide necessary and sufficient conditions for weak tractability under the absolute error criterion. In addition, we provide necessary and sufficient conditions for such problems to be weak but not polynomially tractable in the average case setting.

There are other tractability criteria that one may consider; see Section 1.2. For example, it is an open problem to derive necessary and sufficient conditions for linear tensor product problems in the average case setting to be $\ln^{\kappa}$-weakly tractable, for $\kappa > 1$. Our techniques in Chapter 3 could be useful for that purpose. Some preliminary results in the worst case setting have already been presented [52], extending techniques we developed in [53] (also presented in Chapter 2).

In Chapters 4 and 5 we studied the time-independent Schrödinger equation and developed quantum algorithms approximating the smallest eigenvalue of the Hamiltonian for certain classes of potentials. Our methods are based on phase estimation and in the case of smooth potentials uniformly bounded by a relatively small constant the algorithm vanquishes the curse of dimensionality classical deterministic algorithms suffer from. In the case of convex smooth potentials (Chapter 5) we are also able to approximate the ground state eigenvector of the discretized Hamiltonian within arbitrary distance.

There are still open questions related to eigenvalue estimation of Hamiltonians corresponding to multiparticle systems. For instance, it seems possible to derive algorithms with

better cost, since there is still gap between upper and lower bounds for the problems we study. Furthermore, the question of whether randomized algorithms can estimate the ground state energy for smooth uniformly bounded potentials with cost polynomial in d and ε^{-1} is still open. If the lower bound is proven to be exponential in d, we would have the first non trivial case of a problem where quantum computers perform significantly better than any randomized and by extension any classical algorithm. One can also design quantum algorithms dealing with other interesting classes of potentials. Finally, estimating the ground state energy for the time-independent Schrödinger equation for particles of different masses (1.5) is an open problem.

In Chapter 6 we present results in [57] regarding the implementation of a modification of the NAND evaluation algorithm in [4, Fig. 2] on complete k-ary trees. We present two quantum circuits. The first circuit consists of arbitrary single qubit gates and their controlled variants. It requires $O(\sqrt{N} \cdot \log N)$ gates and is asymptotically optimal. The other approximates the previous circuit and consists of gates only from the Clifford and T gate set. It requires $O(\sqrt{N} \cdot \log N \cdot \log^c(N/\varepsilon))$ gates for error ε, where c is the constant implied in [18].

It is still an open problem to derive estimates (up to constants) on the number of quantum gates required to implement the algorithm for certain values of k, e.g. $k = 2$.

Bibliography

[1] D. S. Abrams and S. Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.*, 83:5162–5165, 1999.

[2] D. S. Abrams and C. P. Williams. Fast quantum algorithms for numerical analysis and stochastic processes. *Report NASA Jet Propulsion Laboratory*, 1999.

[3] A. Ambainis. Quantum walk algorithm for element distinctness. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 22–31, Washington, DC, USA, 2004. IEEE Computer Society.

[4] A. Ambainis, A. M. Childs, B. W. Reichardt, R. Špalek, and S. Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. *SIAM Journal on Computing*, 39(6):2513–2530, 2010.

[5] M. Amy, D. Maslov, M. Mosca, and M. Roetteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *arXiv:1206.0758v2 [quant-ph]*, 2012.

[6] B. Andrews and J. Clutterbuck. Proof of the fundamental gap conjecture. *J. Amer. Math. Soc.*, 24:899–916, 2011.

[7] M. S. Ashbaugh. The fundamental gap. available at http://www.aimath.org/WWN/loweigenvalues/, 2006.

[8] N. S. Bakhvalov. On approximate computation of integrals (in russian). *Vestnik MGV, Ser. Math. Astr. Phys. Chem.*, 1959.

[9] H. Barnum and M. Saks. A lower bound on the quantum query complexity of read-once functions. *Journal of Computer and System Sciences*, 69(2):244–258, 2004.

[10] R. E. Bellman. *Adaptive Control Processes*. Princeton University Press, Princeton, 1961.

[11] A. J. Bessen. A lower bound for the Sturm-Liouville eigenvalue problem on a quantum computer. *Journal of Complexity*, 22(5):660 – 675, 2006. Special Issue: Information-Based Complexity Workshops FoCM Conference Santander, Spain, July 2005.

[12] G. Brassard, P. Høyer, and A. Tapp. Quantum Counting. In *ICALP*, pages 820–831, 1998.

[13] C.-F. Chiang, D. Nagaj, and P. Wocjan. Efficient circuits for quantum walks. *Quantum Info. Comput.*, 10:420–434, May 2010.

[14] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, STOC '03, pages 59–68, New York, NY, USA, 2003. ACM.

[15] A. M. Childs, R. Cleve, S. P. Jordan, and D. Yonge-Mallo. Discrete-query quantum algorithm for NAND trees. *Theory of Computing*, 5(1):119–123, 2009.

[16] A. M. Childs, E. Farhi, and S. Gutmann. An example of the difference between quantum and classical random walks. *Quantum Information Processing*, 1:35–43, April 2002.

[17] C. R. Clark, T. S. Metodi, S. D. Gasster, and K. R. Brown. Resource requirements for fault-tolerant quantum simulation: The ground state of the transverse ising model. *Phys. Rev. A*, 79:062314, 2009.

[18] C. M. Dawson and M. A. Nielsen. The Solovay-Kitaev algorithm. *Quantum Information and Computation*, 6(1):81–95, January 2006.

[19] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[20] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[21] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc. Royal Soc. London A*, 439:553–558, October 1992.

[22] D. P. DiVincenzo. Quantum computation. *Science*, 270(5234):255–261, October 1995.

[23] D. P. DiVincenzo. Two-bit gates are universal for quantum computation. *Phys. Rev. A*, 51:1015–1022, Feb 1995.

[24] B. L. Douglas and J. B. Wang. Efficient quantum circuit implementation of quantum walks. *Phys. Rev. A*, 79:052335, May 2009.

[25] E. Farhi, J. Goldstone, and S. Gutmann. A quantum algorithm for the Hamiltonian NAND tree. *Theory of Computing*, 4(1):169–190, 2008.

[26] E. Farhi and S. Gutmann. Analog analogue of a digital quantum computation. *Phys. Rev. A*, 57:2403–2406, Apr 1998.

[27] R. P. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, 21:467–488, 1982.

[28] M. Mosca G. Brassard, P. Hoyer and A. Tapp. Quantum amplitude amplification and estimation. *Quantum Computation and Information, Contemporary Mathematics Series, AMS*, 2002.

[29] M. Gnewuch and H. Woźniakowski. Quasi-polynomial tractability. *J. Complex.*, 27(3-4):312–330, June 2011.

[30] D. G. Griffiths. *Introduction to Quantum Mechanics*. Pearson Prentice Hall, 2003.

[31] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, Jul 1997.

[32] H. Häffner, C. F. Roos, and R. Blatt. Quantum computing with trapped ions. *Phys. Reports*, 469:155, 2008.

[33] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

[34] S. Heinrich. Optimal summation and integration by deterministic, randomized, and quantum algorithms. In *In 4th international conference on Monte Carlo and Quasi-Monte Carlo Methods*, Hong Kong, 2000.

[35] S. Heinrich. Quantum summation with an application to integration. *J. Complex.*, 18:1–50, March 2002.

[36] S. Heinrich. From Monte Carlo to quantum computation. *Math. Comput. Simul.*, 62:219–230, March 2003.

[37] P. Jaksch and A. Papageorgiou. Eigenvector approximation leading to exponential speedup of quantum eigenvalue estimation. *Phys. Rev. Lett.*, 91:257902, 2003.

[38] S. P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Phys. Rev. Lett.*, 95:050501, Jul 2005.

[39] P. Kaye, R. Laflamme, and M. Mosca. *An Introduction to Quantum Computing*. Oxford University Press, New York, 2007.

[40] J. Kempe, A. Kitaev, and O. Regev. The complexity of the Local Hamiltonian Problem. *SIAM J. Computing*, 35:1070–1097, 2006.

[41] Andreas Klappenecker and Martin Rötteler. Discrete cosine transforms on quantum computers. In *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis*, pages 464–468, 2001.

[42] V. Kliuchnikov, D. Maslov, and M. Mosca. Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates. *arXiv:1206.5236v3 [quant-ph]*, 2012.

[43] P. Kritzer, F. Pillichshammer, and H. Wozniakowski. Multivariate integration of infinitely many times differentiable functions in weighted Korobov spaces. *Mathematics of Computation*, 2012, to appear.

[44] F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via quantum walk. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 575–584, New York, NY, USA, 2007. ACM.

[45] F. Magniez, M. Santha, and M. Szegedy. Quantum algorithms for the triangle problem. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 1109–1117, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[46] A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, STOC '99, pages 384–393, New York, NY, USA, 1999. ACM.

[47] M. A. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[48] E. Novak. Quantum Complexity of Integration. *Journal of Complexity*, 17(1):2 – 16, 2001.

[49] E. Novak and H. Woźniakowski. *Tractability of Multivariate Problems, Linear Information*, volume 1. European Mathematical Society, 2008.

[50] A. Papageorgiou. On the complexity of the multivariate Sturm-Liouville eigenvalue problem. *J. Complex.*, 23:802–827, August 2007.

[51] A. Papageorgiou and I. Petras. Estimating the ground state energy of the Schrödinger equation for convex potentials. in preparation.

[52] A. Papageorgiou and I. Petras. Linear multivariate and linear tensor product problems under a new criterion of tractability. in preparation.

[53] A. Papageorgiou and I. Petras. On the tractability of linear tensor product problems in the worst case. *Journal of Complexity*, 25(5):415 – 419, 2009.

[54] A. Papageorgiou, I. Petras, J.F. Traub, and C. Zhang. A fast algorithm for approximating the ground state energy on a quantum computer. *Mathematics of Computation*, 2012. to appear.

[55] A. Papageorgiou and H. Woźniakowski. Classical and quantum complexity of the Sturm-Liouville eigenvalue problem. *Quantum Information Processing*, 4:87–127, June 2005.

[56] A. Papageorgiou and C. Zhang. On the efficiency of quantum algorithms for Hamiltonian simulation. *Quantum Information Processing*, 11:541–561, 2012. 10.1007/s11128-011-0263-9.

[57] I. Petras and M. Rötteler. Circuit design for the NAND evaluation algorithm. in preparation.

[58] M. Saks and A. Wigderson. Probabilistic Boolean decision trees and the complexity of evaluating game trees. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 29–38, Washington, DC, USA, 1986. IEEE Computer Society.

[59] M. Santha. On the Monte Carlo boolean decision tree complexity of read-once formulae. *Random Structures & Algorithms*, 6(1):75–87, 1995.

[60] R. Shankar. *Principles of Quantum Mechanics*. Plenum Press, 2nd edition edition, 1994.

[61] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26:1484–1509, October 1997.

[62] M. Snir. Lower bounds on probabilistic linear decision trees. *Theoretical Computer Science*, 38(0):69–82, 1985.

[63] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Physics Letters A*, 146(6):319 – 323, 1990.

[64] M. Suzuki. General theory of fractal path integrals with application to many-body theories and statistical physics. *J. Math. Phys.*, 32:400–407, 1991.

[65] M. Suzuki. Fractal path integrals with applications to quantum many-body systems. *Physica A: Statistical Mechanics and its Applications*, 191(1-4):501 – 515, 1992.

[66] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski. *Information-Based Complexity*. Academic Press, 1988.

[67] J. F. Traub and A. G. Werschulz. *Complexity and Information*. Cambridge University Press, 1999.

[68] J.F. Traub and H. Woźniakowski. Path integration on a quantum computer. *Quantum Information Processing*, 1:365–388, 2002. 10.1023/A:1023417813916.

[69] H. F. Weinberger. Upper and lower bounds for eigenvalues by finite difference methods. *Communications on Pure and Applied Mathematics*, 9(3):613–623, 1956.

[70] H. F. Weinberger. Lower bounds for higher eigenvalues by finite difference methods. *Pacific J. Math.*, 8:339–368, 1958.

[71] M. V. Wickerhauser. *Adapted wavelet analysis from theory to software.* A.K. Peters, Wellesley, MA, 1994.

[72] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, Oxford, UK, 1965.

[73] H. Woźniakowski. Tractability and strong tractability of linear multivariate problems. *Journal of Complexity*, 10(1):96–128, 1994.

[74] A. Chih-Chih Yao. Quantum circuit complexity. *FOCS*, pages 352–361, 2007.

[75] S.-T. Yau. Gap of the first two eigenvalues of the Schrödinger operator with nonconvex potentials. *Mat. Contemp.*, 35:267–285, 2008.

Part IV

Appendices

Appendix A

Appendix on repeated phase estimation

In this Appendix, we derive useful Lemmas for use in the repeated phase estimation algorithm of Chapter 5.

Lemma 3. *For* $0 \leq \varepsilon_H < \frac{\sqrt{8}}{\pi^2}$, $|c_0|^2 \geq \frac{\pi^2}{16}$ *we have*

$$\frac{|\alpha(m,\phi_0) - \varepsilon_H|}{\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m,\phi_j)|^2} + \varepsilon_H} > \frac{|\alpha(m,\phi_0)|}{\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m,\phi_j)|^2}} - 7\varepsilon_H,$$

where c_0 *and* $a(m,\phi_0)$ *are defined in* (5.6) *and* (5.7)*, respectively.*

Proof. We first show $\frac{|\alpha(m,\phi_0)|}{\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m,\phi_j)|^2} + \varepsilon_H} > \frac{|\alpha(m,\phi_0)|}{\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m,\phi_j)|^2}} - \gamma\varepsilon_H$ for $\gamma > 2$. Indeed

$$\begin{aligned} \frac{|\alpha(m,\phi_0)|}{\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m,\phi_j)|^2} + \varepsilon_H} &> \frac{|\alpha(m,\phi_0)|}{\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m,\phi_j)|^2}} - \gamma\varepsilon_H \\ \Leftrightarrow \gamma &> \frac{|\alpha(m,\phi_0)|}{\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m,\phi_j)|^2}(\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m,\phi_j)|^2} + \varepsilon_H)} \end{aligned}$$

It suffices to take $\gamma > 4$. Indeed $\sqrt{\sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m,\phi_j)|^2} \geq |\alpha(m,\phi_0)||c_0| \geq \frac{1}{2}$, since $\frac{\sqrt{4}}{\pi} \leq |\alpha(m,\phi_0)| \leq 1$ due to the fact that $\left|\frac{m}{2^b} - \phi_0\right| \leq 2^{-(b+1)}$ [28], and $|c_0| \geq \pi/4$.

Hence

$$\begin{aligned}\frac{|\alpha(m,\phi_0)-\varepsilon_H|}{\sqrt{\sum_{j=1}^{n^d-1}|c_j|^2|\alpha(m,\phi_j)|^2}+\varepsilon_H} &> \frac{|\alpha(m,\phi_0)|}{\sqrt{\sum_{j=1}^{n^d-1}|c_j|^2|\alpha(m,\phi_j)|^2}+\varepsilon_H}-2\varepsilon_H \\ &> \frac{|\alpha(m,\phi_0)|}{\sqrt{\sum_{j=1}^{n^d-1}|c_j|^2|\alpha(m,\phi_j)|^2}}-(\gamma+2)\varepsilon_H,\end{aligned}$$

for $\gamma > 4$. Take $\gamma = 5$ to complete the proof. □

Lemma 4. *Consider $|\psi_{2,m}\rangle$ as defined in* (5.8)*. Then $\||\psi_{2,m}\rangle\| \le \varepsilon_H$*

Proof. We have

$$\begin{aligned}\|\psi_{2,m}\| &= \|\frac{1}{2^b}\sum_{j=0}^{n^d-1}c_j\sum_{k=0}^{2^b-1}e^{-2\pi imk/2^b}|m\rangle|x_{j,k}\rangle\| \\ &\le \frac{1}{2^b}\sum_{k=0}^{2^b-1}\left|e^{-2\pi imk/2^b}\right|\||m\rangle\|\cdot\|\sum_{j=0}^{n^d-1}c_j|x_{j,k}\rangle\| \\ &= \frac{1}{2^b}\sum_{k=0}^{2^b-1}\|D_k\sum_{j=0}^{n^d-1}c_j|u_j\rangle\| \\ &\le \frac{1}{2^b}\sum_{k=0}^{2^b-1}\|D_k\|\cdot\|\sum_{j=0}^{n^d-1}c_j|u_j\rangle\| \\ &= \frac{1}{2^b}\sum_{k=0}^{2^b-1}\|D_k\| \le \varepsilon_H,\end{aligned}$$

since $\|D_k\| \le \varepsilon_H$, and $\|\sum_{j=0}^{n^d-1}c_j|u_j\rangle\| = 1$. □

Lemma 5. *Under the conditions of Theorem 4, i.e. m' is the result of phase estimation for which $\left|\frac{m'}{2^{t_0+b}}-\phi_0\right| \le \frac{1}{2^b}$, $|\phi_j-\phi_0| > \frac{5}{2^b}$ for all $j = 1,2,\ldots,n^d-1$ and $|c_0|^2 \ge \pi^2/16$, we have*

$$\frac{|\alpha(m',\phi_j)|^2}{|\alpha(m',\phi_0)|^2} \le \frac{\pi^2}{32}$$

with probability $p(t_0) \ge 1-\left(\frac{5\pi^2}{2^5}+\frac{1-\frac{\pi^2}{16}}{2^5}\right)\cdot\frac{1}{2^{t_0}}$.

Proof. Let $M_2 = 2^{t_0+b}$. For $j \ge 1$ we have [28]

$$|\alpha(m',\phi_j)|^2 \le \frac{1}{4(M_2\Delta_j)^2} \le \frac{1}{2^{2t_0+6}}, \tag{A.1}$$

where $\Delta_j = \left|\frac{m'}{M_2} - \phi_j\right| = \phi_j - \frac{m'}{M_2} \geq \frac{4}{2^b}$, since $\phi_j > m'/M_2$. Additionally,

$$\begin{aligned} \left|\alpha(m', \phi_0)\right|^2 &= M_2^{-2} \cdot \frac{\sin^2(M_2\Delta_0\pi)}{\sin^2(\Delta_0\pi)} \geq \frac{\sin^2(M_2\Delta_0\pi)}{(M_2\Delta_0\pi)^2} \\ &= \operatorname{sinc}^2(M_2\Delta_0\pi) = \frac{\sin^2(M_2\phi_0\pi)}{M_2^2(\Delta_0\pi)^2} \end{aligned} \tag{A.2}$$

since $\Delta_0 < 2^{t_0}/M_2$ is small enough. Hence,

$$\frac{|\alpha(m', \phi_j)|^2}{|\alpha(m', \phi_0)|^2} \leq \frac{\pi^2}{4}\left(\frac{\Delta_0}{\Delta_j}\right)^2 \frac{1}{\sin^2(M_2\phi_0\pi)}.$$

Note that $\Delta_j > 2^{t_0+2}/M_2$. As a result

$$\frac{|\alpha(m', \phi_j)|^2}{|\alpha(m', \phi_0)|^2} < \frac{\pi^2}{2^6} \cdot \frac{1}{\sin^2(M_2\phi_0\pi)} \tag{A.3}$$

Note that the upper bound in (A.3) depends on how close $M_2\phi_0$ is to an integer or not, or equivalently, what is the fractional part of $m' - M_2\phi_0$ for $m' \in \mathcal{G}$. Assume, without loss of generality, that the closest result m_0 to $M_2\phi_0$ is such that $m_0 - M_2\phi_0 = Y \cdot 2^{-q} < 1/2$, namely $m_0 > M_2\phi_0$. We denote by m_ℓ, for $\ell = -2^{t_0}, -2^{t_0}+2, \ldots, 2^{t_0}-1$, the measurement result such that $m_\ell - M_2\phi_0 = \ell + Y \cdot 2^{-q}$. These are all the elements of $\mathcal{G}$.

Case 1: Let $1/2 > Y \cdot 2^{-q} \geq 1/4$. Then

$$\sin^2(M_2\phi_0\pi) = \sin^2\left(M_2\left(m_\ell - \ell - Y \cdot 2^{-q}\right)\pi\right) = \sin^2(Y \cdot 2^{-q}\pi) \geq \sin^2(\pi/4) \geq 1/2,$$

which according to equation (A.3) implies

$$\frac{|\alpha(m_\ell, \phi_j)|^2}{|\alpha(m_\ell, \phi_0)|^2} \leq \frac{\pi^2}{32}, \tag{A.4}$$

for all $m_\ell \in \mathcal{G}$.

Case 2: We now examine the case where $Y \cdot 2^{-q} < 1/4$, i.e. $q \geq 2$. In this case we deal with results m_ℓ in the set $\mathcal{G}$ for which the bound of interest $\frac{|\alpha(m_\ell, \phi_j)|^2}{|\alpha(m', \phi_0)|^2}$ may become greater than $\frac{\pi^2}{32}$. We show that these results occur with probability at most $\left(\frac{5\pi^2}{2^5} + \frac{1-\frac{\pi^2}{16}}{2^5}\right)2^{-t_0}$.

Initially we consider m_0. Equation (A.2) becomes

$$|\alpha(m_0, \phi_0)|^2 \geq \frac{\sin^2(Y \cdot 2^{-q}\pi)}{(Y \cdot 2^{-q}\pi)^2} = \operatorname{sinc}^2(Y \cdot 2^{-q}\pi).$$

Note that $Y \cdot 2^{-q} < 1/4$, hence $\operatorname{sinc}(\cdot)$ is decreasing. As a result

$$|\alpha(m_0, \phi_0)|^2 \geq \operatorname{sinc}^2(2^{-q}\pi) \geq \left(\frac{2^{-q}\pi - \frac{(2^{-q}\pi)^3}{6}}{2^{-q}\pi}\right)^2 = \left(1 - \frac{\pi^2}{6 \cdot 2^{2q}}\right)^2,$$

since $\sin(x) \geq x - \frac{x^3}{3!}$, for $x < 1$. Furthermore, since $q \geq 2$ and from equation (A.1) we get

$$\frac{|\alpha(m_0, \phi_j)|^2}{|\alpha(m_0, \phi_0)|^2} \leq \frac{1}{4 \cdot 2^{2(t_0+2)}} \cdot \left(1 - \frac{\pi^2}{6 \cdot 2^{2q}}\right)^{-2} < \frac{1}{28} \cdot \left(1 - \frac{\pi^2}{6 \cdot 2^4}\right) < \frac{\pi^2}{32}, \tag{A.5}$$

since $t_0 \geq 1$ and $q \geq 2$.

We now examine the remaining results m_ℓ for $\ell = \pm 1, \pm 2, \ldots, \pm(2^{t_0} - 1), -2^{t_0}$. From equation (A.2) we have

$$|\alpha(m_\ell, \phi_0)|^2 \geq \text{sinc}^2((\ell + Y \cdot 2^{-q})\pi) = \frac{\sin^2(Y \cdot 2^{-q}\pi)}{((\ell + Y \cdot 2^{-q})\pi)^2} \geq \frac{2^{-2(q+1)} \cdot 8}{(\ell + Y \cdot 2^{-q})^2 \pi^2},$$

since $1/4 > Y \cdot 2^{-q} \geq 2^{-(q+1)}$ and $\sin x \geq \frac{2\sqrt{2}}{\pi} x$, for $x < \pi/4$. From equation (A.1) we have

$$\frac{|\alpha(m_\ell, \phi_j)|^2}{|\alpha(m_l, \phi_0)|^2} \leq \frac{\pi^2}{8} \cdot \frac{(\ell + Y \cdot 2^{-q})^2 \cdot 2^{2(q+1)}}{2^{2t_0+6}}$$

which for $\ell \geq 1$ implies

$$\frac{|\alpha(m_\ell, \phi_j)|^2}{|\alpha(m_l, \phi_0)|^2} \leq \frac{\pi^2}{2^9} \cdot \frac{(\ell+1)^2}{2^{2t_0}} \cdot 2^{2(q+1)} := \beta(\ell, q, t_0) \tag{A.6}$$

and for $\ell \leq -1$ implies

$$\frac{|\alpha(m_\ell, \phi_j)|^2}{|\alpha(m_l, \phi_0)|^2} \leq \frac{\pi^2}{2^9} \cdot \frac{\ell^2}{2^{2t_0}} \cdot 2^{2(q+1)} := \beta(\ell, q, t_0) \tag{A.7}$$

If q is large enough (namely $M_2\phi_0$ is very close to m_0), we might have $\beta(\ell, q, t_0) > \pi^2/16$, for some results m_l. Let $\mathcal{B} = \{\ell \in \{-2^{t_0} + 1, -2^{t_0} + 2, \ldots, 2^{t_0} - 1\} : \beta(\ell, q, t_0) > \pi^2/32\}$ the set of the indices of those results, with $\mathcal{B}_- = \{\ell \in \mathcal{B} : \ell < 0\}$ and $\mathcal{B}_+ = \{\ell \in \mathcal{B} : \ell > 0\}$. In addition ℓ_1 is minimum element of the set $\mathcal{B}_+$ and ℓ_2 is the maximum element of $\mathcal{B}_-$.

For any $\ell \in \mathcal{B}_+$ we have

$$\frac{\pi^2}{2^9} \cdot \frac{(\ell+1)^2}{2^{2t_0}} \cdot 2^{2(q+1)} > \pi^2/32 \Rightarrow (\ell+1)^2 2^{2q} > \frac{2^7 \cdot 2^{2t_0}\pi^2}{2^5\pi^2} = 4 \cdot 2^{2t_0}. \tag{A.8}$$

Similarily, for any $\ell \in \mathcal{B}_-$ we have

$$\frac{\pi^2}{2^9} \cdot \frac{\ell^2}{2^{2t_0}} \cdot 2^{2(q+1)} > \pi^2/32 \Rightarrow \ell^2 2^{2q} > \frac{2^7 \cdot 2^{2t_0}\pi^2}{2^5\pi^2} = 4 \cdot 2^{2t_0}. \tag{A.9}$$

From [28, Thm. 11] and for $\ell \in \mathcal{B}_+$ we have

$$|\alpha(m_\ell, \phi_0)|^2 = M_2^{-2} \cdot \frac{\sin^2((\ell + Y \cdot 2^{-p})\pi)}{\sin^2\left(\frac{\ell + Y \cdot 2^{-q}}{M_2} \cdot \pi\right)} \leq \frac{(Y \cdot 2^{-q} \cdot \pi)^2}{\left(\frac{2\sqrt{2}}{\pi}(\ell + Y \cdot 2^{-q})\pi\right)^2} \leq \frac{\pi^2}{2^{2q} \cdot \ell^2 \cdot 2^3} \tag{A.10}$$

since $Y \cdot 2^{-q} < 1/4$. Similarly for $\ell \in \mathcal{B}_-$ we have

$$|\alpha(m_\ell, \phi_0)|^2 \leq \frac{(Y \cdot 2^{-q} \cdot \pi)^2}{\left(\frac{2\sqrt{2}}{\pi}(\ell + Y \cdot 2^{-q})\pi\right)^2} \leq \frac{\pi^2}{2^{2q} \cdot (\ell + 1/4)^2 \cdot 2^3} \tag{A.11}$$

Let $P_1(\mathcal{B})$ the probability of getting a result $m_\ell \in \mathcal{B}$. We have

$$P_1(\mathcal{B}) = \sum_{\ell \in \mathcal{B}} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 = \sum_{\ell \in \mathcal{B}_-} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 + \sum_{\ell \in \mathcal{B}_+} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2$$

We can write

$$\sum_{\ell \in \mathcal{B}_-} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 = \sum_{\ell \in \mathcal{B}_-} \left(|c_0|^2 |\alpha(m_\ell, \phi_0)|^2 + \sum_{j=1}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 \right)$$

and

$$\sum_{\ell \in \mathcal{B}_+} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 = \sum_{\ell \in \mathcal{B}_+} \left(|c_0|^2 |\alpha(m_\ell, \phi_0)|^2 + \sum_{j=1}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 \right)$$

Note that $\sum_{j=1}^{n^d-1} |c_j|^2 = 1 - |c_0|^2 \leq 1 - \frac{\pi^2}{16}$ according to the Lemma's assumptions, and $|\alpha(m_\ell, \phi_j)|^2 \leq 2^{-(2t_0+6)}$ from (A.1). Using the bound from (A.11) we have

$$\begin{aligned}
\sum_{\ell \in \mathcal{B}_-} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 &\leq \sum_{\ell \in \mathcal{B}_-} \left(|\alpha(m_\ell, \phi_0)|^2 + \frac{1 - \pi^2/16}{2^{2t_0+6}} \right) \\
&\leq \frac{\pi^2}{2^{2q+3}} \sum_{\ell=-2^{t_0}}^{\ell_2} \frac{1}{(\ell + 1/4)^2} + (\ell_2 + 2^{t_0}) \left(1 - \frac{\pi^2}{16}\right) \frac{1}{2^{2t_0+6}} \\
&\leq \frac{\pi^2}{2^{2q+3}} \sum_{\ell=-2^{t_0}}^{\ell_2} \frac{1}{(\ell + 1/4)^2} + \left(1 - \frac{\pi^2}{16}\right) \frac{1}{2^{t_0+6}}.
\end{aligned}$$

We now take cases in order to calculate $\sum_{\ell=-2^{t_0}}^{\ell_2} \frac{1}{(\ell+1/4)^2}$ depending on the value of ℓ_2.

Case 2.1 Let $\ell_2 = -1$. Then

$$\sum_{\ell=-2^{t_0}}^{\ell_2} \frac{1}{(\ell + 1/4)^2} = \frac{2^4}{3^2} + \int_{-2^{t_0}}^{-1} \frac{1}{(x + 1/4)^2} dx \leq \frac{2^4}{3^2} + \frac{4}{3} = \frac{28}{9}.$$

As a result,

$$\sum_{\ell \in \mathcal{B}_-} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 \leq \frac{7}{18} \cdot \frac{\pi^2}{2^{2q}} + \left(1 - \frac{\pi^2}{16}\right) \frac{1}{2^{t_0+6}}. \tag{A.12}$$

Case 2.2 Let $\ell_2 < -1$. Then

$$\sum_{\ell=-2^{t_0}}^{\ell_2} \frac{1}{(\ell+1/4)^2} \leq \int_{-2^{t_0}}^{\ell_2+1} \frac{1}{(x+1/4)^2} dx \leq \frac{1}{-\ell_2 - 5/4}.$$

As a result,

$$\sum_{\ell \in \mathcal{B}_-} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 \leq \frac{\pi^2}{2^{2q+3}} \cdot \frac{1}{-\ell_2 - 5/4} + \left(1 - \frac{\pi^2}{16}\right) \frac{1}{2^{t_0+6}}. \tag{A.13}$$

Similarly we examine the probability of the results m_ℓ for $\ell \in \mathcal{B}_+$. Using the bound from (A.10) we have

$$\begin{aligned}
\sum_{\ell \in \mathcal{B}_+} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 &\leq \sum_{\ell \in \mathcal{B}_+} \left(|\alpha(m_\ell, \phi_0)|^2 + \frac{1 - \pi^2/16}{2^{2t_0+6}} \right) \\
&\leq \frac{\pi^2}{2^{2q+3}} \sum_{\ell=\ell_1}^{2^{t_0}-1} \frac{1}{\ell^2} + (2^{t_0} - \ell_1 + 1) \left(1 - \frac{\pi^2}{16}\right) \frac{1}{2^{2t_0+6}} \\
&\leq \frac{\pi^2}{2^{2q+3}} \sum_{\ell=\ell_1}^{2^{t_0}-1} \frac{1}{\ell^2} + \left(1 - \frac{\pi^2}{16}\right) \frac{1}{2^{t_0+6}}.
\end{aligned}$$

We now consider different values of ℓ_1, to calculate $\sum_{\ell=\ell_1}^{2^{t_0}-1} \frac{1}{\ell^2}$.

Case 2.3 Let $\ell_1 = 1$. Then

$$\sum_{\ell=\ell_1}^{2^{t_0}-1} \frac{1}{\ell^2} = 1 + \int_1^{2^{t_0}-1} \frac{1}{x^2} dx \leq 2.$$

As a result,

$$\sum_{\ell \in \mathcal{B}_+} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 \leq \frac{\pi^2}{2^{2q+2}} + \left(1 - \frac{\pi^2}{16}\right) \frac{1}{2^{t_0+6}}. \tag{A.14}$$

Case 2.4 Let $\ell_1 > 1$. Then

$$\sum_{\ell=\ell_1}^{2^{t_0}-1} \frac{1}{\ell^2} \leq \int_{\ell_1-1}^{2^{t_0}-1} \frac{1}{x^2} dx \leq \frac{1}{\ell_1 - 1}.$$

As a result,

$$\sum_{\ell \in \mathcal{B}_-} \sum_{j=0}^{n^d-1} |c_j|^2 |\alpha(m_\ell, \phi_j)|^2 \leq \frac{\pi^2}{2^{2q+3}} \cdot \frac{1}{\ell_1 - 1} + \left(1 - \frac{\pi^2}{16}\right) \frac{1}{2^{t_0+6}}. \tag{A.15}$$

Let $\ell_1 = 1, \ell_2 = -1$. Then from (A.12),(A.14)

$$P_1(\mathcal{B}) \leq \left(\frac{7}{18} + \frac{1}{4}\right)\frac{\pi^2}{2^{2q}} + \frac{1-\pi^2/16}{2^5}\cdot\frac{1}{2^{t_0}}.$$

From (A.8),(A.9) we have $2^{2q} > 2^{2t_0+2}$ and $2^{2q} > 2^{2t_0}$. Hence

$$P_1(\mathcal{B}) \leq \left(\frac{7}{18} + \frac{1}{4}\right)\frac{\pi^2}{2^{2t_0+2}} + \frac{1-\pi^2/16}{2^5}\cdot\frac{1}{2^{t_0}} \leq \frac{1-\pi^2/16}{2^4}2^{-t_0}, \tag{A.16}$$

for t_0 sufficiently large.

Let $\ell_1 = 1, \ell_2 < -1$. From (A.13),(A.14)

$$P_1(\mathcal{B}) \leq \frac{\pi^2}{2^{2q+3}}\cdot\frac{1}{-\ell_2 - 5/4} + \frac{\pi^2}{2^{2q+2}} + \frac{1-\pi^2/16}{2^5}\cdot\frac{1}{2^{t_0}}.$$

From (A.8),(A.9) we have $2^{2q} > 2^{2t_0}$ and $2^{2q} > \frac{2^{2t_0+2}}{\ell_2^2}$. Hence

$$\begin{aligned} P_1(\mathcal{B}) &\leq \frac{\pi^2}{2^3}\cdot\frac{\ell_2^2}{(-\ell_2-5/4)^2}\cdot 2^{-(2t_0+2)} + \frac{1-\pi^2/16}{2^5}\cdot\frac{1}{2^{t_0}} + \frac{\pi^2}{2^{2t_0+2}} \\ &\leq \left(\frac{\pi^2}{2^3} + \frac{1-\pi^2/16}{2^5}\right)2^{-t_0}, \end{aligned} \tag{A.17}$$

since $\frac{\ell_2^2}{-\ell_2-5/4} \leq 2^{t_0+1}$ and for t_0 sufficiently large.

Let $\ell_1 > 1, \ell_2 = -1$. From (A.12),(A.15)

$$P_1(\mathcal{B}) \leq \frac{7}{18}\cdot\frac{\pi^2}{2^{2q}} + \frac{\pi^2}{2^{2q+3}}\cdot\frac{1}{\ell_1 - 1} + \frac{1-\pi^2/16}{2^5}\cdot\frac{1}{2^{t_0}}.$$

From (A.8),(A.9) we have $2^{2q} > 2^{2t_0+2}$ and $2^{2q} > \frac{2^{2t_0+2}}{(\ell_1^2+1)^2}$. Hence

$$\begin{aligned} P_1(\mathcal{B}) &\leq \frac{7}{18}\cdot\frac{\pi^2}{2^{2t_0+2}} + \frac{1-\pi^2/16}{2^5}\cdot\frac{1}{2^{t_0}} + \frac{\pi^2}{2^{2t_0+2}} + \frac{\pi^2}{2^{2t_0+5}}\cdot\frac{(\ell_1+1)^2}{\ell_1-1} \\ &\leq \left(\frac{\pi^2}{2^3} + \frac{1-\pi^2/16}{2^5}\right)2^{-t_0}, \end{aligned} \tag{A.18}$$

since $\frac{(\ell_1+1)^2}{\ell_1-1} \leq 3\cdot 2^{t_0}$ and for t_0 sufficiently large.

Now consider $\ell_1 > 1, \ell_2 < -1$. From (A.13),(A.15)

$$P_1(\mathcal{B}) \leq +\frac{\pi^2}{2^{2q+3}}\cdot\frac{1}{-\ell_2 - 5/4} + \frac{1-\pi^2/16}{2^5}\cdot\frac{1}{2^{t_0}} + \frac{\pi^2}{2^{2q+3}}\cdot\frac{1}{\ell_1 - 1}.$$

From (A.8),(A.9) we have $2^{2q} > \frac{2^{2t_0+2}}{(\ell_1+1)^2}$ and $2^{2q} > \frac{2^{2t_0+2}}{\ell_2^2}$. Hence

$$\begin{aligned} P_1(\mathcal{B}) &\leq \frac{\pi^2}{2^{2t_0+5}}\cdot\frac{(\ell_1+1)^2}{\ell_1-1} + \frac{1-\pi^2/16}{2^5}\cdot\frac{1}{2^{t_0}} + \frac{\pi^2}{2^{2t_0+5}}\cdot\frac{\ell^2}{-\ell_2-5/4} \\ &\leq \left(\frac{5\pi^2}{2^5} + \frac{1-\pi^2/16}{2^5}\right)2^{-t_0}. \end{aligned} \tag{A.19}$$

Finally, combining the results from (A.16) ,(A.17),(A.18) and (A.19) we have

$$P_1(\mathcal{B}) \leq \left(\frac{5\pi^2}{2^5} + \frac{1 - \pi^2/16}{2^5} \right) 2^{-t_0}.$$

□