

Quantum Algorithms and Complexity for Continuous Problems*

A. Papageorgiou[†] and J. F. Traub[‡]
Department of Computer Science
Columbia University
New York, USA

Article Outline

Glossary
I. Abstract
II. Introduction
III. Overview of Quantum Algorithms
IV. Integration
V. Path Integration
VI. Feynman-Kac Path Integration
VII. Eigenvalue Approximation
VIII. Qubit Complexity
IX. Approximation
X. Partial Differential Equations
XI. Ordinary Differential Equations
XII. Gradient Estimation
XIII. Simulation of Quantum Systems on Quantum Computers
XIV. Future Directions
Bibliography

Glossary

- **Black Box Model.** This model assumes we can collect knowledge about an input f

*To appear in the Springer Encyclopedia of Complexity and Systems Science

[†]ap@cs.columbia.edu

[‡]traub@cs.columbia.edu

thru queries without knowing how the answer to the query is computed. A synonym for black box is oracle.

- **Classical Computer.** A computer which does not use the principles of quantum computing to carry out its computations.
- **Computational Complexity.** In this article, complexity for brevity. The minimal cost of solving a problem by an algorithm. Some authors use the word complexity when cost would be preferable. An upper bound on the complexity is given by the cost of an algorithm. A lower bound is given by a theorem which states there cannot be an algorithm which does better.
- **Continuous Problem.** A problem involving real or complex functions of real or complex variables. Examples of continuous problem are integrals, path integrals, and partial differential equations.
- **Cost of an Algorithm.** The price of executing an algorithm. The cost depends on the model of computation.
- **Discrete Problem.** A problem whose inputs are from a countable set. Examples of discrete problems are integer factorization, traveling salesman and satisfiability.
- **ε -Approximation.** Most real-world continuous problems can only be solved numerically and therefore approximately, that is to within an error threshold ε . The definition of ε -approximation depends on the setting. See worst-case setting, randomized setting, quantum setting.
- **Information-Based Complexity.** The discipline that studies algorithms and complexity of continuous problems.
- **Model of Computation.** The rules stating what is permitted in a computation and how much it costs. The model of computation is an abstraction of a physical computer. Examples of models are Turing machines, real number model, quantum circuit model.
- **Optimal Algorithm.** An algorithm whose cost equals the complexity of the problem.
- **Promise.** A statement of what is known about a problem a priori before any queries are made. An example in quantum computation is the promise that an unknown 1-bit function is constant or balanced. In information-based complexity a promise is also called global information.
- **Quantum Computing Speedup.** The amount by which a quantum computer can solve a problem faster than a classical computer. To compute the speedup one must know the classical complexity and it's desirable to also know the quantum complexity. Grover proved quadratic speedup for search in an unstructured database. Its only conjectured that Shor's algorithm provides exponential speedup for integer factorization since the classical complexity is unknown.

- **Query.** One obtains knowledge about a particular input thru queries. For example, if the problem is numerical approximation of $\int_0^1 f(x) dx$ a query might be the evaluation of f at a point. In information-based complexity the same concept is called an information operation.
- **Quantum Setting.** There are a number of quantum settings. An example is a guarantee of error at most ε with probability greater than $\frac{1}{2}$.
- **Qubit Complexity.** The minimal number of qubits to solve a problem.
- **Query Complexity.** The minimal number of queries required to solve the problem.
- **Randomized Setting.** In this setting the expected error with respect to the probability measure generating the random variables is at most ε . The computation is randomized. An important example of a randomized algorithm is the Monte Carlo method.
- **Worst-Case Setting.** In this setting an error of at most ε is guaranteed for all inputs satisfying the promise. The computation is deterministic.

I Abstract

Most continuous mathematical formulations arising in science and engineering can only be solved numerically and therefore approximately. We shall always assume that we're dealing with a numerical approximation to the solution.

There are two major motivations for studying quantum algorithms and complexity for continuous problems.

1. Are quantum computers more powerful than classical computers for important scientific problems? How much more powerful?
2. Many important scientific and engineering problems have continuous formulations. These problems occur in fields such as physics, chemistry, engineering and finance. The continuous formulations include path integration, partial differential equations (in particular, the Schrödinger equation) and continuous optimization.

To answer the first question we must know the classical computational complexity (for brevity, complexity) of the problem. There have been decades of research on the classical complexity of continuous problems in the field of information-based complexity. The reason we know the complexity of many continuous problems is that we can use adversary arguments to obtain their query complexity. This may be contrasted with the classical complexity of discrete problems where we have only conjectures such as $P \neq NP$. Even the classical complexity of the factorization of large integers is unknown. Knowing the classical complexity of a continuous problem we obtain the quantum computation speedup if we know the quantum complexity. If we know an upper bound on the quantum complexity through the cost of a particular quantum algorithm then we can obtain a lower bound on the quantum speedup.

Regarding the second motivation, in this article we'll report on high-dimensional integration, path integration, Feynman path integration, the smallest eigenvalue of a differential equation, approximation, partial differential equations, ordinary differential equations and gradient estimation. We'll also briefly report on the simulation of quantum systems on a quantum computer.

II Introduction

We provide a summary of the contents of the article.

Section III: Overview of quantum algorithms and complexity.

We define basic concepts and notation including quantum algorithm, continuous problem, query complexity and qubit complexity.

Section IV: Integration

High-dimensional integration, often in hundreds or thousands of variables, is one of the most commonly occurring continuous problems in science. In Section IV.1 we report on complexity results on a classical computer. For illustration we begin with a one-dimensional problem and give a simple example of an adversary argument. We then move on the d -dimensional case and indicate that in the worst case the complexity is exponential in d ; the problem suffers the curse of dimensionality. The curse can be broken by the Monte Carlo algorithm. In Section IV.2 we report on the algorithms and complexity results on a quantum computer. Under certain assumptions on the promise the quantum query complexity enjoys exponential speedup over classical worst case query complexity.

A number of the problems we'll discuss enjoy exponential quantum query speedup. This does not contradict Beals et al. [6] who prove that speedup can only be polynomial. The reason is that [6] deals with problems concerning total Boolean functions.

For many classes of integrands there is quadratic speedup over the classical randomized query complexity. This is the same speedup as enjoyed by Grover's search algorithm of an unstructured database. To obtain the quantum query complexity one has to give matching upper and lower bounds. As usual the upper bound is given by an algorithm, the lower bound by a theorem. The upper bound is given by the amplitude amplification algorithm of Brassard et al. [12]. We outline a method for approximating high-dimensional integrals using this algorithm. The quantum query complexity lower bounds for integration are based on the lower bounds of Nayak and Wu [47] for computing the mean of a Boolean function.

Section V: Path Integration

We define a path integral and provide an example due to Feynman. In Section V.1 we report on complexity results on a classical computer. If the promise is that the class of integrands has finite smoothness, then path integration is intractable in the worst case. If the promise is that the class of integrands consists of entire functions the query complexity is tractable even in the worst case. For smooth functions intractability is broken by Monte Carlo. In Section V.2 we report on the algorithm and complexity on a quantum computer. The quantum query complexity enjoys Grover-type speedup over classical randomized query complexity. We outline the quantum algorithm.

Section VI: Feynman-Kac path integration.

The Feynman-Kac path integral provides the solution to the diffusion equation. In Section VI.1 we report on algorithms and complexity on a classical computer. In the worst case for a d -dimensional Feynman-Kac path integral the problem again suffers the curse of dimensionality which can be broken by Monte Carlo. In Section VI.2 we indicate the algorithm and query complexity on a quantum computer.

Section VII: Eigenvalue approximation.

One of the most important problems in physics and chemistry is approximating the ground state energy governed by a differential equation. Typically, numerical algorithms on a classical computer are well known. Our focus is on the Sturm-Liouville eigenvalue (SLE) problem where the first complexity results were recently obtained. The SLE equation is also called the time-independent Schrödinger equation. In Section VII.1 we present an algorithm on a classical computer. The worst case query complexity suffers the curse of dimensionality. We also state a randomized algorithm. The randomized query complexity is unknown for $d > 2$ and is an important open question. In Section VII.2 we outline an algorithm for a quantum computer. The quantum query complexity is not known when $d > 1$. It has been shown that it is not exponential in d ; the problem is tractable on a quantum computer.

Section VIII: Qubit complexity.

So far we've focused on query complexity. For the foreseeable future the number of qubits will be a crucial computational resource. We give a general lower bound on the qubit complexity of continuous problems. We show that because of this lower bound there's a problem that cannot be solved on a quantum computer but that's easy to solve on a classical computer using Monte Carlo.

A definition of a quantum algorithm is given by (1); the queries are deterministic. Woźniakowski [77] introduced the quantum setting with randomized queries for continuous problems. For path integration there is an exponential reduction in the qubit complexity.

Section IX: Approximation.

Approximating functions of d variables is a fundamental and generally hard problem. The complexity is sensitive to the norm, p , on the class of functions and to several other parameters. For example, if $p = \infty$ approximation suffers the curse of dimensionality in the worst and randomized classical cases. The problem remains intractable in the quantum setting.

Section X: Partial Differential Equations.

Elliptic partial differential equations have many important applications and have been extensively studied. In particular, consider an equation of order $2m$ in d variables. In the classical worst case setting the problem is intractable. The classical randomized and quantum settings were only recently studied. The conclusion is that the quantum may or may not provide a polynomial speedup; it depends on certain parameters.

Section XI: Ordinary Differential Equations.

Consider a system of initial value ordinary equations in d variables. Assume that the right hand sides satisfy a Hölder condition. The problem is tractable even in the worst case with the exponent of ε^{-1} depending on the Hölder class parameters. The complexity

of classical randomized and quantum algorithms have only recently been obtained. The quantum setting yields a polynomial speedup.

Section XII: Gradient estimation.

Jordan [37] showed that approximating the gradient of a function can be done with a single query on a quantum computer although it takes $d+1$ function evaluations on a classical computer. A simplified version of Jordan's algorithm is presented.

Section XIII: Simulation of quantum systems on quantum computers.

There is a large and varied literature on simulation of quantum systems on quantum computers. The focus in these papers is typically on the cost of particular classical and quantum algorithms without complexity analysis and therefore without speedup results. To give the reader a taste of this area we list some sample papers.

Section XIV: Future directions.

We briefly indicate a number of open questions.

III Overview of Quantum Algorithms

A quantum algorithm consists of a sequence of unitary transformations applied to an initial state. The result of the algorithm is obtained by measuring its final state. The quantum model of computation is discussed in detail in [6, 7, 8, 17, 27, 48] and we summarize it here as it applies to continuous problems.

The initial state $|\psi_0\rangle$ of the algorithm is a unit vector of the Hilbert space $\mathcal{H}_\nu = \mathbb{C}^2 \otimes \dots \otimes \mathbb{C}^2$, ν times, for some appropriately chosen integer ν , where \mathbb{C}^2 is the two dimensional space of complex numbers. The dimension of \mathcal{H}_ν is 2^ν . The number ν denotes the number of qubits used by the quantum algorithm.

The final state $|\psi\rangle$ is also a unit vector of \mathcal{H}_ν and is obtained from the initial state $|\psi_0\rangle$ through a sequence of unitary $2^\nu \times 2^\nu$ matrices, i.e.,

$$|\psi\rangle_f := U_T Q_f U_{T-1} Q_f \dots U_1 Q_f U_0 |\psi_0\rangle. \quad (1)$$

The unitary matrix Q_f is called a quantum query and is used to provide information to the algorithm about a function f . Q_f depends on n function evaluations $f(t_1), \dots, f(t_n)$, at deterministically chosen points, $n \leq 2^\nu$. The U_0, U_1, \dots, U_T are unitary matrices that do not depend on f . The integer T denotes the number of quantum queries.

For algorithms solving discrete problems, such as Grover's algorithm for the search of an unordered database [26], the input f is considered to be a Boolean function. The most commonly studied quantum query is the *bit* query. For a Boolean function $f : \{0, \dots, 2^m - 1\} \rightarrow \{0, 1\}$, the bit query is defined by

$$Q_f |j\rangle |k\rangle = |j\rangle |k \oplus f(j)\rangle. \quad (2)$$

Here $\nu = m + 1$, $|j\rangle \in \mathcal{H}_m$, and $|k\rangle \in \mathcal{H}_1$ with \oplus denoting addition modulo 2. For a real function f the query is constructed by taking the most significant bits of the function f evaluated at some points t_j . More precisely, as in [27], the bit query for f has the form

$$Q_f |j\rangle |k\rangle = |j\rangle |k \oplus \beta(f(\tau(j)))\rangle, \quad (3)$$

where the number of qubits is now $\nu = m' + m''$ and $|j\rangle \in \mathcal{H}_{m'}$, $|k\rangle \in \mathcal{H}_{m''}$. The functions β and τ are used to discretize the domain \mathcal{D} and the range \mathcal{R} of f , respectively. Therefore, $\beta : \mathcal{R} \rightarrow \{0, 1, \dots, 2^{m''} - 1\}$ and $\tau : \{0, 1, \dots, 2^{m'} - 1\} \rightarrow \mathcal{D}$. Hence, we compute f at $t_j = \tau(j)$ and then take the m'' most significant bits of $f(t_j)$ by $\beta(f(t_j))$, for the details and the possible use of ancillary qubits see [27].

At the end of the quantum algorithm, the final state $|\psi_f\rangle$ is measured. The measurement produces one of M outcomes, where $M \leq 2^\nu$. Outcome $j \in \{0, 1, \dots, M - 1\}$ occurs with probability $p_f(j)$, which depends on j and the input f . Knowing the outcome j , we classically compute the final result $\phi_f(j)$ of the algorithm.

In principle, quantum algorithms may have measurements applied between sequences of unitary transformations of the form presented above. However, any algorithm with multiple measurements can be simulated by a quantum algorithm with only one measurement [8].

Let S be a linear or nonlinear operator such that

$$S : \mathcal{F} \rightarrow \mathcal{G}. \quad (4)$$

Typically, \mathcal{F} is a linear space of real functions of several variables, and \mathcal{G} is a normed linear space. We wish to approximate $S(f)$ to within ε for $f \in \mathcal{F}$. We approximate $S(f)$ using n function evaluations $f(t_1), \dots, f(t_n)$ at deterministically and a priori chosen sample points. The quantum query Q_f encodes this information, and the quantum algorithm obtains this information from Q_f .

Without loss of generality, we consider algorithms that approximate $S(f)$ with probability $p \geq \frac{3}{4}$. We can boost the success probability of an algorithm to become arbitrarily close to one by repeating the algorithm a number of times. The success probability becomes at least $1 - \delta$ with a number of repetitions proportional to $\log \delta^{-1}$.

The local error of the quantum algorithm (1) that computes the approximation $\phi_f(j)$, for $f \in \mathcal{F}$ and the outcome $j \in \{0, 1, \dots, M - 1\}$, is defined by

$$e(\phi_f, S) = \min \left\{ \alpha : \sum_{j: \|S(f) - \phi_f(j)\| \leq \alpha} p_f(j) \geq \frac{3}{4} \right\}, \quad (5)$$

where $p_f(j)$ denotes the probability of obtaining outcome j for the function f . The worst case error of a quantum algorithm ϕ is defined by

$$e^{\text{quant}}(\phi, S) = \sup_{f \in \mathcal{F}} e(\phi_f, S). \quad (6)$$

The query complexity $\text{comp}^{\text{query}}(\varepsilon, S)$ of the problem S is the minimal number of queries necessary for approximating the solution with accuracy ε , i.e.,

$$\text{comp}^{\text{query}}(\varepsilon) = \min \{ T : \exists \phi \text{ such that } e^{\text{quant}}(\phi, S) \leq \varepsilon \}. \quad (7)$$

Similarly, the qubit complexity of the problem S is the minimal number of qubits necessary for approximating the solution with accuracy ε , i.e.,

$$\text{comp}^{\text{qubit}}(\varepsilon) = \min \{ \nu : \exists \phi \text{ such that } e^{\text{quant}}(\phi, S) \leq \varepsilon \}. \quad (8)$$

IV Integration

Integration is one of the most commonly occurring mathematical problems. One reason is that when one seeks the expectation of a continuous process one has to compute an integral. Often the integrals are over hundreds or thousands of variables. Path integrals are taken over an infinite number of variables. See Section V.

IV.1 Classical Computer

We begin with a one dimensional example to illustrate some basic concepts before moving to the d -dimensional case (Number of dimensions and number of variables are used interchangeably.) Our simple example is to approximate

$$I(f) = \int_0^1 f(x) dx.$$

For most integrands we can't use the fundamental theorem of calculus to compute the integral analytically; we have to approximate numerically (most real world continuous problems have to be approximated numerically). We have to make a promise about f . Assume

$$F_1 = \{f : [0, 1] \rightarrow \mathbb{R} \mid \text{continuous and } |f(x)| \leq 1, x \in [0, 1]\}.$$

Use queries to compute

$$y_f = [f(x_1), \dots, f(x_n)].$$

We show that with this promise one cannot guarantee an ε -approximation on a classical computer. We use a simple adversary argument. Choose arbitrary numbers $x_1, \dots, x_n \in [0, 1]$. The adversary answers all these queries by answering $f(x_i) = 0, i = 1, \dots, n$.

What is f ? It could be $f_1 \equiv 0$ and $\int_0^1 f_1(x) dx = 0$ or it could be the f_2 shown in Figure 1. The value of $\int_0^1 f_2(x) dx$ can be arbitrarily close to 1. Since $f_1(x)$ and $f_2(x)$ are indistinguishable with these query answers and this promise, it is impossible to guarantee an ε -approximation on a classical computer with $\varepsilon < \frac{1}{2}$. We will return to this example in Section VIII.

We move on to the d -dimensional case. Assume that the integration limits are finite. For simplicity we assume we're integrating over the unit cube $[0, 1]^d$. So our problem is

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

If our promise is that $f \in F_d$, where

$$F_d = \{f : [0, 1]^d \rightarrow \mathbb{R} \mid \text{continuous and } |f(x)| \leq 1, x \in [0, 1]^d\},$$

then we can't compute an ε -approximation regardless of the value of d . Our promise has to be stronger. Assume that integrand class has *smoothness* r . There are various ways to define smoothness r for functions of d variables. See, for example, the definition [68, p. 25].

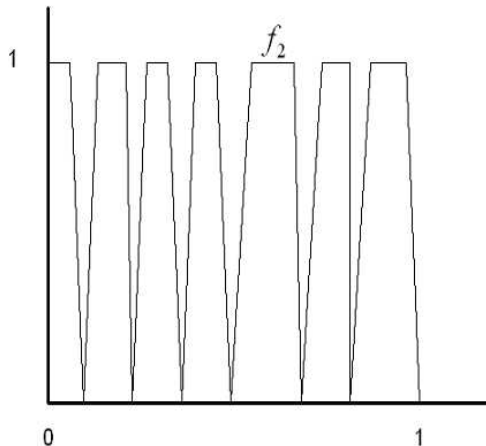


Figure 1: All the function evaluations are equal to zero but the integral is close to one

(For other definitions see [67].) With this definition Bakhvalov [4] showed that the query complexity is

$$\text{comp}_{\text{clas-wor}}^{\text{query}}(\varepsilon) = \Theta(\varepsilon^{-d/r}). \quad (9)$$

This is the worst case query complexity on a classical computer.

What does this formula imply? If $r = 0$ the promise is that the functions are only continuous but have no smoothness. Then the query complexity is ∞ , that is, we cannot guarantee an ε -approximation. If r and ε are fixed the query complexity is an exponential function of d . We say the problem is intractable. Following R. Bellman this is also called the *curse of dimensionality*. In particular, let $r = 1$. Then the promise is that the integrands have one derivative and the query complexity is $\Theta(\varepsilon^{-d})$.

The curse of dimensionality is present for many for continuous problems in the worst case setting. Breaking the curse is one of the central issues of information-based complexity. For high-dimensional integration the curse can be broken for F_d by the Monte Carlo algorithm which is a special case of a randomized algorithm. The Monte Carlo algorithm is defined by

$$\phi^{\text{MC}}(f) = \frac{1}{n} \sum_{i=1}^n f(x_i), \quad (10)$$

where the x_i are chosen independently from a uniform distribution. Then the expected error is

$$e^{\text{MC}}(f) = \frac{\sqrt{\text{var}(f)}}{\sqrt{n}},$$

where

$$\text{var}(f) = \int_{[0,1]^d} f^2(x) dx - \left\{ \int_{[0,1]^d} f(x) dx \right\}^2$$

is the variance of f . For the promise $f \in F_d$ the query complexity is given by

$$\text{comp}_{\text{clas-ran}}^{\text{query}}(\varepsilon) = \Theta(\varepsilon^{-2}). \quad (11)$$

This is the randomized query complexity on a classical computer.

Thus Monte Carlo breaks the curse of dimensionality for the integration problem; the problem is tractable. Why should picking sample points at random be much better than picking them deterministically in the optimal way? This is possible because we've replaced the guarantee of the worst case setting by the stochastic assurance of the randomized setting. There's no free lunch!

The reader will note that (11) is a complexity result even though it is the cost of a particular algorithm, the Monte Carlo algorithm. The reason is that for integrands satisfying this promise Monte Carlo has been proven optimal. It is known that if the integrands are smoother Monte Carlo is not optimal; See [68, p. 32].

In the worst case setting (deterministic) the query complexity is infinite if $f \in F_d$. In the randomized setting the query complexity is independent of d if $f \in F_d$. This will provide us guidance when we introduce Monte Carlo sampling into the model of quantum computation in Section VIII.

Generally pseudo-random numbers are used in the implementation of Monte Carlo on a classical computer. The quality of a pseudo-random number generator is usually tested statistically; see, for example, Knuth [41]. Will (11) still hold if a pseudo-random number generator is used? An affirmative answer is given by [70] provided some care is taken in the choice of the generator and f is Lipschitz.

IV.2 Quantum Computer

We've seen that Monte Carlo breaks the curse of dimensionality for high-dimensional integration on a classical computer and that the query complexity is of order ε^{-2} . Can we do better on a quantum computer?

The short answer is yes. Under certain assumptions on the promises, which will be made precise below, the quantum query complexity enjoys exponential speedup over the classical worst case query complexity and quadratic speedup over the classical randomized query complexity. The latter is the same speedup as enjoyed by Grover's search algorithm of an unstructured database [26].

To show that the quantum query complexity is of order ε^{-1} we have to give matching, or close to matching upper and lower bounds. Usually, the upper bound is given by an algorithm, the lower bound by a theorem. The upper bound is given by the amplitude amplification algorithm of Brassard et al. [12] which we describe briefly.

The amplitude amplification algorithm of Brassard et al. computes the mean

$$\text{SUM}(f) = \frac{1}{N} \sum_{i=0}^{N-1} f(i),$$

of a Boolean function $f : \{0, 1, \dots, N-1\} \rightarrow \{0, 1\}$, where $N = 2^k$, with error ε and probability at least $8/\pi^2$, using a number of queries proportional to $\min\{\varepsilon^{-1}, N\}$. Moreover, Nayak and Wu [47] show that the order of magnitude of the number of queries of this algorithm is optimal. Without loss of generality we can assume that $\varepsilon^{-1} \ll N$.

Perhaps the easiest way to understand the algorithm is to consider the operator

$$G = (I - |\psi\rangle\langle\psi|)O_f,$$

that is used in Grover's search algorithm. Here

$$O_f|x\rangle = (-1)^{f(x)}|x\rangle, \quad x \in \{0, 1\}^k,$$

denotes the query, which is slightly different yet equivalent [48] to the one we defined in (2).

Let

$$|\psi\rangle = \sqrt{\frac{1}{N}} \sum_x |x\rangle$$

be the equally weighted superposition of all the states.

If M denotes the number of assignments for which f has the value 1 then

$$\text{SUM}(f) = \frac{M}{N}.$$

Without loss of generality $1 \leq M \leq N - 1$. Consider the space \mathcal{H} spanned by the states

$$|\psi_0\rangle = \sqrt{\frac{1}{N-M}} \sum_{x:f(x)=0} |x\rangle \quad \text{and} \quad |\psi_1\rangle = \sqrt{\frac{1}{M}} \sum_{x:f(x)=1} |x\rangle.$$

Then

$$|\psi\rangle = \cos(\theta/2)|\psi_0\rangle + \sin(\theta/2)|\psi_1\rangle$$

where $\sin(\theta/2) = \sqrt{M/N}$, and $\theta/2$ is the angle between the states $|\psi\rangle$ and $|\psi_0\rangle$. Thus

$$\sin^2\left(\frac{\theta}{2}\right) = \frac{M}{N}.$$

Now consider the operator G restricted to \mathcal{H} which has the matrix representation

$$G = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

Its eigenvalues are $\lambda_{\pm} = e^{\pm i\theta}$, $i = \sqrt{-1}$, and let $|\xi_{\pm}\rangle$ denote the corresponding eigenvectors.

We can express $|\psi\rangle$ using the $|\xi_{\pm}\rangle$ to get

$$|\psi\rangle = a|\xi_{-}\rangle + b|\xi_{+}\rangle,$$

with $a, b \in \mathbb{C}$, $|a|^2 + |b|^2 = 1$. This implies that phase estimation [48] with G^p , $p = 2^j$, $j = 0, \dots, t-1$, and initial state $|0\rangle^{\otimes t}|\psi\rangle$ can be used to approximate either θ or $2\pi - \theta$ with error proportional to 2^{-t} . Note that the first t qubits of the initial state determine the accuracy of phase estimation. Indeed, let $\tilde{\phi}$ be the result of phase estimation. Since $\sin^2(\theta/2) = \sin^2(\pi - \theta/2)$,

$$\left| \sin^2(\pi\tilde{\phi}) - \frac{N}{M} \right| = O(2^{-t}),$$

with probability at least $8/\pi^2$; see [12, 48] for the details. Setting $t = \Theta(\log \varepsilon^{-1})$ and observing that phase estimation requires a number of applications of G (or queries O_f)

proportional to ε^{-1} yields the result. (The complexity of quantum algorithms for the average case approximation of the Boolean mean has also been studied [33, 52].)

For a real function $f : \{0, 1, \dots, N - 1\} \rightarrow [0, 1]$, we can approximate

$$\text{SUM}(f) = \frac{1}{N} \sum_{i=0}^{N-1} f(i),$$

by reducing the problem to the computation of the Boolean mean. One way to derive this reduction is to truncate $f(i)$ to the desired number of significant bits, typically, polynomial in $\log \varepsilon^{-1}$, and then to use the bit representation of the function values to derive a Boolean function whose mean is equal to the mean of the truncated real function, see, e.g. [77]. The truncation of the function values is formally expressed through the mapping β in (3). Variations of this idea have been used in the literature [3, 27, 50].

Similarly, one discretizes the domain of a function $f : [0, 1]^d \rightarrow [0, 1]$ using the function τ in (3) and then uses the amplitude amplification algorithm to compute the average

$$\text{SUM}(f) = \frac{1}{N} \sum_{i=0}^{N-1} f(x_i), \tag{12}$$

for $x_i \in [0, 1]^d$, $i = 0, \dots, N - 1$.

The quantum query complexity of computing the average (12) is of order ε^{-1} . On the other hand, the classical deterministic worst case query complexity is proportional to N (recall that $\varepsilon^{-1} \ll N$), and the classical randomized query complexity is proportional to ε^{-2} .

We now turn to the approximation of high-dimensional integrals and outline an algorithm for solving this problem. Suppose $f : [0, 1]^d \rightarrow [0, 1]$ is a function for which we are given some promise, for instance, that f has smoothness r . The algorithm integrating f with accuracy ε has two parts. First, using function evaluations $f(x_i)$, $i = 1, \dots, n$, at deterministic points, it approximates f classically, by a function \hat{f} with error ε_1 , i.e.,

$$\|f - \hat{f}\| \leq \varepsilon_1,$$

where $\|\cdot\|$ is the L_∞ norm. The complexity of this problem has been extensively studied and there are numerous results [49, 67, 68] specifying the optimal choice of n and the points x_i that must be used to achieve error ε_1 . Thus

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

where $g = f - \hat{f}$. Since \hat{f} is known and depends linearly on the $f(x_i)$ the algorithm proceeds to integrate it exactly. So it suffices to approximate $\int_{[0,1]^d} g(x) dx$ knowing that $\|g\| \leq \varepsilon_1$.

The second part of the algorithm approximates the integral of g using the amplitude amplification algorithm to compute

$$\text{SUM}(g) = \frac{1}{N} \sum_{i=0}^{N-1} g(y_i),$$

for certain points $y_i \in [0, 1]^d$, with error ε_2 . Once more, there are many results, see [67, 68] for surveys specifying the optimal N and the points y_i , so that $\text{SUM}(g)$ approximates $\int_{[0,1]^d} g(x) dx$ with error ε_2 . Finally, the algorithm approximates the original integral by the sum of the results of its two parts. The overall error of the algorithm is proportional to $\varepsilon = \varepsilon_1 + \varepsilon_2$.

Variations of the integration algorithm we described are known to have optimal query complexity for a number of different promises [27, 29, 35, 50]. The quantum query complexity lower bounds for integration are based on the lower bounds of Nayak and Wu [47] for computing the Boolean mean. The quantum algorithms offer an exponential speedup over classical deterministic algorithms and a polynomial speedup over classical randomized algorithms for the query complexity of high-dimensional integration. The table below summarizes the query complexity (up to polylog factors) of high-dimensional integration in the worst case, randomized and quantum setting for functions belonging to Hölder classes $F_d^{r,\alpha}$ and Sobolev spaces $W_{p,d}^r$. Heinrich obtained most of the quantum query complexity results in a series of papers, which we cited earlier. Heinrich summarized his results in [28] where a corresponding table showing error bounds can be found.

	Worst case	Randomized	Quantum
$F_d^{r,\alpha}$	$\varepsilon^{-d/(r+\alpha)}$	$\varepsilon^{-2d/(2(r+\alpha)+d)}$	$\varepsilon^{-d/(r+\alpha+d)}$
$W_{p,d}^r$ $2 \leq p \leq \infty$	$\varepsilon^{-d/r}$	$\varepsilon^{-2d/(2r+d)}$	$\varepsilon^{-d/(r+d)}$
$W_{p,d}^r$ $1 \leq p \leq 2$	$\varepsilon^{-d/r}$	$\varepsilon^{-pd/(rp+pd-d)}$	$\varepsilon^{-d/(r+d)}$
$W_{1,d}^r$	$\varepsilon^{-d/r}$	$\varepsilon^{-d/r}$	$\varepsilon^{-d/(r+d)}$

Abrams and Williams [3] were the first to apply the amplitude amplification algorithm to high-dimensional integration. Novak [50] was the first to spell out his promises and thus obtained the first complexity results for high-dimensional integration.

V Path Integration

A path integral is defined as

$$I(f) = \int_X f(x) \mu(dx), \tag{13}$$

where μ is a probability measure on an infinite-dimensional space X . It can be viewed as an infinite-dimensional integral. For illustration we give an example due to R. Feynman. In classical mechanics a particle at a certain position at time t_0 has a unique trajectory to its position at time t_1 . Quantum mechanically there are an infinite number of possible trajectories which Feynman called histories, see Figure 2. Feynman summed over the histories. If one goes to the limit one gets a path integral. Setting $t_0 = 0$, $t_1 = 1$ this integral is

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

which is a special case of (13).

Path integration occurs in numerous applications including quantum mechanics, quantum chemistry, statistical mechanics, and mathematical finance.

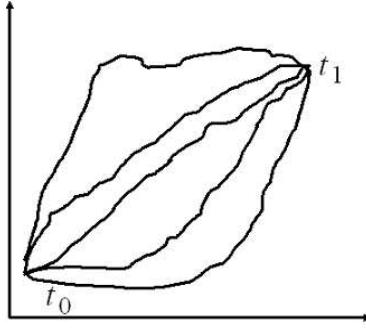


Figure 2: Different trajectories of a particle

V.1 Classical Computer

The first complexity analysis of path integration is due to Wasilkowski and Woźniakowski [72]; see also [68, Ch. 5]. They studied the deterministic worst case and randomized settings and assume that μ is a Gaussian measure; an important special case of a Gaussian measure is a Wiener measure. They make the promise that F , the class of integrands, consists of functions $f : X \rightarrow \mathbb{R}$ whose r -th Fréchet derivatives are continuous and uniformly bounded by unity. If r is finite then path integration is intractable. Curbera [20] showed that the worst case query complexity is of order $\varepsilon^{\varepsilon^{-\beta}}$ where β is a positive number depending on r .

Wasilkowski and Woźniakowski [72] also considered the promise that F consists of entire functions and that μ is the Wiener measure. Then the query complexity is a polynomial in ε^{-1} . More precisely, they provide an algorithm for calculating a worst case ε -approximation with cost of order ε^{-p} and the problem is tractable with this promise. The exponent p depends on the particular Gaussian measure; for the Wiener measure $p = 2/3$.

We return to the promise that the smoothness r is finite. Since this problem is intractable in the worst case, Wasilkowski and Woźniakowski [72] ask whether settling for a stochastic assurance will break the intractability. The obvious approach is to approximate the infinite-dimensional integral by a d -dimensional integral where d may be large (or even huge) since d is polynomial in ε^{-1} . Then Monte Carlo may be used since its speed of convergence does not depend on d . Modulo an assumption that the n -th eigenvalue of the covariance operator of μ does not decrease too fast the randomized query complexity is roughly ε^{-2} . Thus Monte Carlo is optimal.

V.2 Quantum Computer

Just as with finite dimensional integration Monte Carlo makes path integration tractable on a classical computer and the query complexity is of order ε^{-2} . Again we ask whether we can do better on a quantum computer and again the short answer is yes. Under certain assumptions on the promises, which will be made precise below, the quantum query complexity is of order ε^{-1} . Thus quantum query complexity enjoys exponential speedup over the classical worst case query complexity and quadratic speedup over the classical randomized query complexity. Again the latter is the same speedup as enjoyed by Grover's search algorithm of an unstructured database.

The idea for solving path integration on a quantum computer is fairly simple but the analysis is not so easy. So we outline the algorithm without considering the details. We start with a classical deterministic algorithm that uses an average as in (12) to approximate the path integral $I(f)$ with error ε in the worst case. The number of terms N of this average is an exponential function of ε^{-1} . Nevertheless, on a quantum computer we can approximate the average, using the amplitude amplification algorithm as we discussed in Section 4.2, with cost that depends on $\log N$ and is, therefore, a polynomial in ε^{-1} [71, Sec. 6].

A summary of the promises and the results in [71] follows. The measure μ is Gaussian and the eigenvalues of its covariance operator is of order j^{-h} , $h > 1$. For the Wiener measure occurring in many applications we have $h = 2$. The class of integrands consists of functions f whose r -th Fréchet derivatives are continuous and uniformly bounded by unity. In particular assume the integrands are at least Lipschitz. Then

- Path integration on a quantum computer is tractable.
- Query complexity on a quantum computer enjoys exponential speedup over the worst case and quadratic speedup over the classical randomized query complexity. More precisely, the number of quantum queries is at most $4.22\varepsilon^{-1}$.

Results on the qubit complexity of path integration will be given in Section VIII. Details of an algorithm for computing an ε -approximation to a path integral on a quantum computer may be found in [71, Sec. 6].

VI Feynman-Kac Path Integration

An important special case of a path integral is a Feynman-Kac path integral. Assume that X is the space C of continuous functions and that the measure μ is the Wiener measure w . Feynman-Kac path integrals occur in many applications; see [23]. For example consider the diffusion equation

$$\begin{aligned}\frac{\partial z}{\partial t}(u, t) &= \frac{1}{2} \frac{\partial^2 z}{\partial t^2}(u, t) + V(u)z(u, t) \\ z(u, 0) &= v(u),\end{aligned}$$

where $u \in \mathbb{R}$, $t > 0$, V is a potential function, and v is an initial condition function. Under mild conditions on v and V the solution is given by the Feynman-Kac path integral

$$z(u, t) = \int_C v(x(t) + u) e^{\int_0^t V(x(s)+u) ds} w(dx). \quad (14)$$

The problem generalizes to the multivariate case by considering the diffusion equation

$$\begin{aligned}\frac{\partial z}{\partial t}(u, t) &= \frac{1}{2} \Delta z(u, t) + V(u)z(u, t) \\ z(u, 0) &= v(u),\end{aligned}$$

with $u \in \mathbb{R}^d$, $t > 0$, and $V, v : \mathbb{R}^d \rightarrow \mathbb{R}$, the potential and the initial value function, respectively. As usual, Δ denotes the Laplacian. The solution is given by the Feynman-Kac path integral

$$z(u, t) = \int_C v(x(t) + u) e^{\int_0^t V(x(s) + u) ds} w(dx),$$

where C is the set of continuous functions $x : \mathbb{R}_+ \rightarrow \mathbb{R}^d$ such that $x(0) = 0$.

Note that there are two kinds of dimension here. A Feynman-Kac path integral is infinite dimensional since we're integrating over continuous functions. Furthermore u is a function of d variables.

VI.1 Classical Computer

We begin with the case when u is a scalar and then move to the case where u is a multivariate function. There have been a number of papers on the numerical solution of (14); see, for example [14].

The usual attack is to solve the problem with a stochastic assurance using randomization. For simplicity we make the promise that $u = 1$ and V is four times continuously differentiable. Then by Chorin's algorithm [16], the total cost is of order $\varepsilon^{-2.5}$.

The first complexity analysis may be found in Plaskota et al. [58] where a new algorithm is defined which enjoys certain optimality properties. They construct an algorithm which computes an ε -approximation at cost of order $\varepsilon^{-2.5}$ and show that the worst case complexity is of the same order. Hence the exponent of ε^{-1} is an order of magnitude smaller and with a worst case rather than a stochastic guarantee. However, the new algorithm requires a numerically difficult precomputation which may limit its applicability.

We next report on multivariate Feynman-Kac path integration. First consider the worst case setting with the promise that v and V are r times continuously differentiable with r finite. Kwas and Li [43] proved that the query complexity is of order $\varepsilon^{-d/r}$. Therefore in the worst case setting the problem suffers the curse of dimensionality.

The randomized setting was studied by Kwas [42]. He showed that the curse of dimensionality is broken by using Monte Carlo using a number of queries of order ε^{-2} . The number of queries can be further improved to $\varepsilon^{-2/(1+2r/d)}$, which is the optimal number of queries, by a bit more complicated algorithm. The randomized algorithms require extensive precomputing [42, 43].

VI.2 Quantum Computer

Multivariate multivariate Feynman-Kac path integration on a quantum computer was studied in [42]. With the promise as in the worst and randomized case, Kwas presents an algorithm and complexity analysis. He exhibits a quantum algorithm that uses a number of queries of order ε^{-1} that is based on the Monte Carlo algorithm. He shows that the query complexity is of order $\varepsilon^{-1/(1+r/d)}$ and is achieved by a bit more complicated quantum algorithm. Just as in the randomized case the quantum algorithms require extensive precomputing [42, 43].

VII Eigenvalue Approximation

Eigenvalue problems for differential operators arising in physics and engineering have been extensively studied in the literature; see, e.g. [5, 18, 19, 22, 25, 40, 63, 65]. Typically, the mathematical properties of the eigenvalues and the corresponding eigenfunctions are known and so are numerical algorithms approximating them on a classical computer. Nevertheless, the complexity of approximating eigenvalues in the worst, randomized and quantum settings has only recently been addressed for the Sturm-Liouville eigenvalue problem [53, 55] (see also [10] for quantum lower bounds with a different kind of query than the one we discuss in this article). In some cases we have sharp complexity estimates but there are important questions that remain open.

Most of our discussion here concerns the complexity of approximating the smallest eigenvalue of a Sturm-Liouville eigenvalue problem. We will conclude this section by briefly addressing quantum algorithms for other eigenvalue problems.

In the physics literature this problem is called the time-independent Schrödinger equation. The smallest eigenvalue is the energy of the ground state. In the mathematics literature it is called the Sturm-Liouville eigenvalue problem.

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

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

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

$$\mathbb{L}_q u = \lambda u, \quad x \in (0, 1)^d, \tag{15}$$

$$u(x) \equiv 0, \quad x \in \partial I_d. \tag{16}$$

In the variational form, the smallest eigenvalue $\lambda = \lambda(q)$ of (15, 16) is given by

$$\lambda(q) = \min_{0 \neq u \in H_0^1} \frac{\int_{I_d} \sum_{j=1}^d [D_j u(x)]^2 + q(x) u^2(x) dx}{\int_{I_d} u^2(x) dx}, \tag{17}$$

where H_0^1 is the space of all functions vanishing on the boundary of I_d having square integrable first order partial derivatives. We consider the complexity of classical and quantum algorithms approximating $\lambda(q)$ with error ε .

VII.1 Classical Computer

In the worst case we discretize the differential operator on a grid of size $h = \Theta(\varepsilon^{-1})$ and obtain a matrix $M_\varepsilon = -\Delta_\varepsilon + B_\varepsilon$, of size proportional to $\varepsilon^{-d} \times \varepsilon^{-d}$. The matrix Δ_ε is the $(2d + 1)$ -point finite difference discretization of the Laplacian. The matrix B_ε is a diagonal matrix containing evaluations of q at the grid points. The smallest eigenvalue of M_ε approximates $\lambda(q)$ with error $O(\varepsilon)$ [73, 74]. We compute the smallest eigenvalue of M_ε using the bisection method [22, p. 228]. The resulting algorithm uses a number of queries proportional to ε^{-d} .

It turns out that this number of queries is optimal in the worst case, and the problem suffers from the curse of dimensionality.

The query complexity lower bounds are obtained by reducing the eigenvalue problem to high-dimensional integration. For this we use the perturbation formula [53, 55]

$$\lambda(q) = \lambda(\bar{q}) + \int_{I_d} (q(x) - \bar{q}(x)) u_{\bar{q}}^2(x) dx + O(\|q - \bar{q}\|_\infty^2), \quad (18)$$

where $q, \bar{q} \in \mathbf{Q}$ and $u_{\bar{q}}$ is the normalized eigenfunction corresponding to $\lambda(\bar{q})$.

The same formula is used for lower bounds in the randomized setting. Namely, the query complexity is

$$\Omega(\varepsilon^{-2d/(d+2)}).$$

Moreover, we can use (18) to derive a randomized algorithm. First we approximate q by a function \bar{q} and then approximate $\lambda(q)$ by approximating the first two terms on the right hand side of (18); see Papageorgiou and Woźniakowski [55] for $d = 1$, and Papageorgiou [53] for general d . However, this algorithm uses

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

queries. So, it is optimal only when $d \leq 2$. Determining the randomized complexity for $d > 2$ and determining if the randomized complexity is an exponential function of d are important open questions.

VII.2 Quantum Computer

The perturbation formula (18) can be used to show that the quantum query complexity is

$$\Omega(\varepsilon^{-d/(d+1)}).$$

As in the randomized case, we can use (18) to derive an algorithm that uses $O(\varepsilon^{-d/2})$ quantum queries. The difference between the quantum algorithm and the randomized algorithm is that the former uses the amplitude amplification algorithm to approximate the integral on the right hand side of (18) instead of Monte Carlo. The algorithm is optimal only when $d = 1$ [53, 55].

For general d the query complexity is not known exactly, we only have the upper bound $O(\varepsilon^{-p})$, $p \leq 6$. The best quantum algorithm known is based on phase estimation. In particular, we discretize the problem as in the worst case and apply phase estimation to the unitary matrix

$$e^{i\gamma M_\varepsilon},$$

where γ is chosen appropriately so that the resulting phase belongs to $[0, 2\pi)$. We use a splitting formula to approximate the necessary powers of the matrix exponential. The largest eigenvalue of Δ_ε is $O(\varepsilon^{-2})$ and $\|q\|_\infty \leq 1$. This implies that the resulting number of queries does not grow exponentially with d .

Finally, there are a number of papers providing quantum algorithms for eigenvalue approximation without carrying out a complete complexity analysis. Abrams and Lloyd [2]

have written an influential paper on eigenvalue approximation of a quantum mechanical system evolving with a given Hamiltonian. They point out that phase estimation, which requires the corresponding eigenvector as part of its initial state, can also be used with an approximate eigenvector. Jaksch and Papageorgiou [36] give a quantum algorithm which computes a *good* approximation of the eigenvector at low cost. Their method can be generally applied to the solution of continuous Hermitian eigenproblems on a discrete grid. It starts with a classically obtained eigenvector for a problem discretized on a coarse grid and constructs an approximate eigenvector on a fine grid.

We describe this algorithm briefly for the case $d = 1$. Suppose $N = 2^k$ and $N_0 = 2^{k_0}$ are the number of points in the fine and coarse grid, respectively. Given the eigenvector for the coarse grid $|U^{(N_0)}\rangle$, we approximate the eigenvector for the fine grid $|U^{(N)}\rangle$ by

$$|\tilde{U}^{(N)}\rangle = |U^{(N_0)}\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right)^{\otimes(k-k_0)}.$$

The effect of this transformation is to replicate the coordinates of $|U^{(N_0)}\rangle$ 2^{k-k_0} times. The resulting approximation is good enough in the sense that the success probability of phase estimation with initial state $|\tilde{U}_h\rangle$ is greater than $\frac{1}{2}$.

Szkopek et al. [64] use the algorithm of Jaksch and Papageorgiou in the approximation of low order eigenvalues of a differential operator of order $2s$ in d dimensions. Their paper provides an algorithm with cost polynomial in ε^{-1} and generalizes the results of Abrams and Lloyd [2]. However, [64] does not carry out a complexity analysis but only considers known classical algorithms in the worst case for comparison.

VIII Qubit Complexity

For the foreseeable future the number of qubits will be a crucial computational resource. We give a general lower bound on the qubit complexity of continuous problems.

Recall that in (1) we defined a quantum algorithm as

$$|\psi_f\rangle = U_T Q_f U_{T-1} Q_f \dots U_1 Q_f U_0 |\psi_0\rangle.$$

where $|\psi_0\rangle$ and $|\psi_f\rangle$ are the initial and final state vectors, respectively. They are column vectors of length 2^ν . The query Q_f , a $2^\nu \times 2^\nu$ unitary matrix, depends on the values of f at $n \leq 2^\nu$ deterministic points. That is

$$Q_f = Q_f(f(x_1), \dots, f(x_n)).$$

It's important to note that in the standard model the evolution is completely deterministic. The only probabilistic element is in the measurement of the final state.

For the qubit complexity (8) we have the following lower bound

$$\text{comp}_{\text{std}}^{\text{qubit}}(\varepsilon, S) = \Omega(\log \text{comp}_{\text{clas}}^{\text{query}}(2\varepsilon, S)). \quad (19)$$

Here S specifies the problem, see (4), and $\text{comp}_{\text{std}}^{\text{qubit}}(\varepsilon, S)$ is the qubit complexity in the standard quantum setting. On the right hand side $\text{comp}_{\text{clas}}^{\text{query}}(\varepsilon, S)$ is the query complexity on a classical computer in the worst case setting. See [77] for details.

We provide an intuition about (19). Assume 2^ν function evaluations are needed to solve the problem specified by S to within ε . Note that ν qubits are needed to generate the Hilbert space \mathcal{H}_ν of size 2^ν to store the evaluations.

Equation (19) can be interpreted as a certain limitation of the standard quantum setting, which considers queries using function evaluations at deterministic points. We'll show why this is a limitation and show we can do better.

Consider multivariate integration which was discussed in Section IV. We seek to approximate the solution of

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

Assume our promise is $f \in F_0$ where

$$F_d = \{f : [0, 1]^d \rightarrow \mathbb{R} \mid \text{continuous and } |f(x)| \leq 1, x \in [0, 1]^d\}.$$

For the moment let $d = 1$. We showed that with this promise we cannot guarantee an ε -approximation on a classical computer with $\varepsilon < \frac{1}{2}$. That is,

$$\text{comp}_{\text{clas}}^{\text{query}}(\varepsilon, S) = \infty.$$

By (19)

$$\text{comp}_{\text{std}}^{\text{qubit}}(\varepsilon, S) = \infty.$$

If the qubit complexity is infinite even for $d = 1$ its certainly infinite for general d . But we saw that if classical randomization (Monte Carlo) is used

$$\text{comp}_{\text{clas-ran}}^{\text{query}}(\varepsilon, S) = \Theta(\varepsilon^{-2}).$$

Thus we have identified a problem which is easy to solve on a classical computer and is impossible to solve on a quantum computer using the standard formulation of a quantum algorithm (1). Our example motivates extending the notion of quantum algorithm by permitting *randomized queries*. The quantum setting with randomized queries was introduced by Woźniakowski [77]. The idea of using randomized queries is not entirely new. Shor's algorithm [60] uses a special kind of randomized query, namely,

$$Q_x|x\rangle = |jx \bmod N\rangle,$$

with $j = 0, \dots, N - 1$ and a random x from the set $\{2, 3, \dots, N - 1\}$.

In our case, the randomization affects only the selection of sample points and the number of queries. It occurs prior to the implementation of the queries and the execution of the quantum algorithm. In this extended setting, we define a quantum algorithm as

$$|\psi_{f,\omega}\rangle = U_{T_\omega} Q_{f,\omega} U_{T_\omega-1} Q_{f,\omega} \dots U_1 Q_{f,\omega} U_0 |\psi_0\rangle, \quad (20)$$

where ω is a random variable and

$$Q_{f,\omega} = Q_{f,\omega}(f(x_{1,\omega}), \dots, f(x_{n,\omega})),$$

and the $x_{j,\omega}$ are random points. The number of queries T_ω and the points $x_{j,\omega}$ are chosen at random initially and then remain fixed for the remainder of the computation. Note that (20) is identical to (1) except for the introduction of randomization. Randomized queries require that we modify the criterion (6) by which we measure the error of an algorithm. One possibility is to consider the expected error and another possibility is to consider the probabilistic error with respect to the random variable ω . Both cases have been considered in the literature [77] but we will avoid the details here because they are rather technical.

A test of the new setting is whether it buys us anything. We compare the qubit complexity of the standard and randomized settings for integration and path integration.

VIII.1 Integration

We make the same promise as above, namely, $f \in F_d$.

- Quantum setting with deterministic queries. We remind the reader that

$$\begin{aligned}\text{comp}_{\text{std}}^{\text{query}}(\varepsilon) &= \infty \\ \text{comp}_{\text{std}}^{\text{qubit}}(\varepsilon) &= \infty.\end{aligned}$$

- Quantum setting with randomized queries. Then [77]

$$\begin{aligned}\text{comp}_{\text{ran}}^{\text{query}}(\varepsilon) &= \Theta(\varepsilon^{-1}) \\ \text{comp}_{\text{ran}}^{\text{qubit}}(\varepsilon) &= \Theta(\log \varepsilon^{-1}).\end{aligned}$$

Therefore, there is infinite improvement in the randomized quantum setting over the standard quantum setting.

VIII.2 Path integration

- Quantum setting with deterministic queries. With an appropriate promise it was shown [71] that

$$\begin{aligned}\text{comp}_{\text{std}}^{\text{query}}(\varepsilon) &= \Theta(\varepsilon^{-1}) \\ \text{comp}_{\text{std}}^{\text{qubit}}(\varepsilon) &= \Theta(\varepsilon^{-2} \log \varepsilon^{-1}).\end{aligned}$$

Thus, modulo the log factor, the qubit complexity of path integration is a second degree polynomial in ε^{-1} . That seems pretty good but we probably won't have enough qubits for a long time to do new science, especially with error correction.

- Quantum setting with randomized queries. Then [77]

$$\begin{aligned}\text{comp}_{\text{ran}}^{\text{query}}(\varepsilon) &= \Theta(\varepsilon^{-1}) \\ \text{comp}_{\text{ran}}^{\text{qubit}}(\varepsilon) &= \Theta(\log \varepsilon^{-1}).\end{aligned}$$

Thus there is an exponential improvement in the randomized quantum setting over the standard quantum setting.

As the analogue of (19) we have the following lower bound on the randomized qubit complexity [77]

$$\text{comp}_{\text{ran}}^{\text{qubit}}(\varepsilon, S) = \Omega(\log \text{comp}_{\text{clas-ran}}^{\text{query}}(\varepsilon, S)), \quad (21)$$

where $\text{comp}_{\text{clas-ran}}^{\text{query}}(\varepsilon, S)$ is the query complexity on a classical computer in the randomized setting.

IX Approximation

Approximating functions of d variables is a fundamental and generally hard problem. Typically, the literature considers the approximation of functions that belong to the Sobolev space $W_p^r([0, 1]^d)$ in the norm of $L_q([0, 1]^d)$. The condition $r/d > 1/p$ ensures that functions in $W_p^r([0, 1]^d)$ are continuous, which is necessary for function values to be well defined. Thus, when $p = \infty$ the dimension d can be arbitrarily large while the smoothness r can be fixed, which cannot happen when $p < \infty$.

For $p = \infty$ the problem suffers the curse of dimensionality in the worst and the randomized classical cases [49, 67]. Recently, Heinrich [30] showed that quantum computers do not offer any advantage relative to classical computers since the problem remains intractable in the quantum setting.

For different values of the parameters p, q, r, d the classical and quantum complexities are also known [30, 49, 67]. In some cases quantum computers can provide a roughly quadratic speedup over classical computers, but there are also cases where the classical and quantum complexities coincide. The table below summarizes the order of the query complexity (up to polylog factors) of approximation in the worst case, randomized and quantum setting, and is based on a similar table in [30] describing error bounds.

	Worst case	Randomized	Quantum
$1 \leq p < q \leq \infty,$ $r/d \geq 2/p - 2/q$	$\varepsilon^{-dpq/(rpq-d(q-p))}$	$\varepsilon^{-dpq/(rpq-d(q-p))}$	$\varepsilon^{-d/r}$
$1 \leq p < q \leq \infty,$ $r/d < 2/p - 2/q$	$\varepsilon^{-dpq/(rpq-d(q-p))}$	$\varepsilon^{-dpq/(rpq-d(q-p))}$	$\varepsilon^{-dpq/(2rpq-2d(q-p))}$
$1 \leq q \leq p \leq \infty$	$\varepsilon^{-d/r}$	$\varepsilon^{-d/r}$	$\varepsilon^{-d/r}$

X Elliptic Partial Differential Equations

Elliptic partial differential equations have many important applications and have been extensively studied in the literature, see [75] and the references therein. A simple example is the Poisson equation, for which we want to find a function $u : \bar{\Omega} \rightarrow \mathbb{R}$, that satisfies

$$\begin{aligned} -\Delta u(x) &= f(x), & x \in \Omega \\ u(x) &= 0, & x \in \partial\Omega, \end{aligned}$$

where $\Omega \subset \mathbb{R}^d$,

More generally we consider elliptic partial differential equations of order $2m$ on a smooth bounded domain $\Omega \subset \mathbb{R}^d$ with smooth coefficients and homogeneous boundary conditions with the right hand side belonging to $C^r(\Omega)$ and the error measured in the L_∞ norm; see [32] for details.

In the worst case the complexity is proportional to $\varepsilon^{-d/r}$ [75] and the problem is intractable. The randomized complexity of this problem was only recently studied along with the quantum complexity by Heinrich [31, 32]. In particular, the randomized query complexity (up to polylog factors) is proportional to

$$\varepsilon^{-\max\{d/(r+2m), 2d/(2r+d)\}},$$

and the quantum query complexity is proportional to

$$\varepsilon^{-\max\{d/(r+2m), d/(r+d)\}}.$$

Thus the quantum setting may provide a polynomial speedup over the classical randomized setting but not always. Moreover, for fixed m and r and for $d > 4m$ the problem is intractable in all three settings.

XI Ordinary Differential Equations

In this section we consider the solution of a system of ordinary differential equations with initial conditions

$$z'(t) = f(z(t)), \quad t \in [a, b], \quad z(a) = \eta,$$

where $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$, $z : [a, b] \rightarrow \mathbb{R}^d$ and $\eta \in \mathbb{R}^d$ with $f(\eta) \neq 0$. For the right hand side function $f = [f_1, \dots, f_d]$, where $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$, we assume that the f_j belong to the Hölder class $F_d^{r, \alpha}$, $r + \alpha \geq 1$. We seek to compute a bounded function on the interval $[a, b]$ that approximates the solution z .

Kacewicz [38] studied the classical worst case complexity of this problem and found it to be proportional to $\varepsilon^{-1/(r+\alpha)}$. Recently he also studied the classical randomized and quantum complexity of the problem and derived algorithms that yield upper bounds that from the lower bounds by only an arbitrarily small positive parameter in the exponent [39]. The resulting randomized and quantum complexity bounds (up to polylog factors) are

$$O(\varepsilon^{-1/(r+\alpha+1/2-\gamma)})$$

and

$$O(\varepsilon^{-1/(r+\alpha+1-\gamma)}),$$

where $\gamma \in (0, 1)$ is arbitrarily small, respectively. Observe that the randomized and quantum complexities (up to polylog factors) satisfy

$$\Omega(\varepsilon^{-1/(r+\alpha+1/2)})$$

and

$$\Omega(\varepsilon^{-1/(r+\alpha+1)}),$$

respectively. Even more recently, Heinrich and Milla [34] showed that the upper bound for the randomized complexity holds with $\gamma = 0$, thereby establishing tight upper and lower randomized complexity bounds, up to polylog factors.

Once more, the quantum setting provides a polynomial speedup over the classical setting.

XII Gradient Estimation

Approximating the gradient of a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ with accuracy ε requires a minimum of $d + 1$ function evaluations on a classical computer. Jordan [37] shows how this can be done using a single query on a quantum computer.

We present Jordan's algorithm for the special case where the function is a plane passing through the origin, i.e., $f(x_1, \dots, x_d) = \sum_{j=1}^d a_j x_j$, and is uniformly bounded by 1. Then $\nabla f = (a_1, \dots, a_d)^T$. Using a single query and *phase kickback* we obtain the state

$$\frac{1}{\sqrt{N^d}} \sum_{j_1=0}^{N-1} \dots \sum_{j_d=0}^{N-1} e^{2\pi i f(j_1, \dots, j_d)} |j_1\rangle \dots |j_d\rangle,$$

where N is a power of 2. Equivalently, we have

$$\frac{1}{\sqrt{N^d}} \sum_{j_1=0}^{N-1} \dots \sum_{j_d=0}^{N-1} e^{2\pi i (a_1 j_1 + \dots + a_d j_d)} |j_1\rangle \dots |j_d\rangle.$$

This is equal to the state

$$\frac{1}{\sqrt{N}} \sum_{j_1=0}^{N-1} e^{2\pi i a_1 j_1} |j_1\rangle \dots \frac{1}{\sqrt{N}} \sum_{j_d=0}^{N-1} e^{2\pi i a_d j_d} |j_d\rangle.$$

We apply the Fourier transform to each of the d registers and then measure each register in the computational basis to obtain m_1, \dots, m_d . If a_j can be represented with finitely many bits and N is sufficiently large then $m_j/N = a_j$, $j = 1, \dots, d$.

For functions with second order partial derivatives not identically equal to zero the analysis is more complicated and we refer the reader to [37] for the details.

XIII Simulation of Quantum Systems on Quantum Computers

So far this article has been devoted to work on algorithms and complexity of problems where the query and qubit complexities are known or have been studied. In a number of cases, the classical complexity of these problems is also known and we know the quantum computing speedup.

The notion that quantum systems could be simulated more efficiently by quantum computers than by classical computers was first mentioned by Manin [44], see also [45], and discussed thoroughly by Feynman [24].

There is a large and varied literature on simulation of quantum systems on quantum computers. The focus in these papers is typically on the cost of particular quantum and classical algorithms without complexity analysis and therefore without speedup results. To give the reader a taste of this area we list some sample papers:

- Berry et al. [9] present an efficient quantum algorithm for simulating the evolution of a sparse Hamiltonian.
- Dawson, Eisert and Osborne [21] introduce a unified formulation of variational methods for simulating ground state properties of quantum many-body systems.
- Morita and Nishimori [46] derive convergence conditions for the quantum annealing algorithm.
- Brown, Clark, Chuang [13] establish limits of quantum simulation when applied to specific problems.
- Chen, Yezpez and Cory [15] report on the simulation of Burgers equation as a type-II quantum computation.
- Paredes, Verstraete and Cirac [56] present an algorithm that exploits quantum parallelism to simulate randomness.
- Somma et al. [61] discuss what type of physical problems can be efficiently simulated on a quantum computer which cannot be simulated on a Turing machine.
- Yezpez [78] presents an efficient algorithm for the many-body three-dimensional Dirac equation.
- Nielsen and Chuang [48] discuss simulation of a variety of quantum systems.
- Sornborger and Stewart [62] develop higher order methods for simulations.
- Boghosian and Taylor [11] present algorithms for efficiently simulating quantum mechanical systems.
- Zalka [79] shows that the time evolution of the wave function of a quantum mechanical many particle system can be simulated efficiently.
- Abrams and Lloyd [1] provide fast algorithms for simulating many-body Fermi systems.
- Wisner [76] provides two quantum many-body problems whose solution is intractable on a classical computer.

XIV Future Directions

The reason there is so much interest in quantum computers is to solve important problems fast. The belief is that we will be able to solve scientific problems, and in particular quantum mechanical systems, which cannot be solved on a classical computer. That is, that quantum computation will lead to new science.

Research consists of two major parts. The first is identification of important scientific problems with substantial speedup. The second is the construction of machines with sufficient number of qubits and long enough decoherence times to solve the problems identified in the first part. Abrams and Lloyd [2] have argued that with 50 to 100 qubits we can solve interesting classically intractable problems from atomic physics. Of course this does not include qubits needed for fault tolerant computation.

There are numerous important open questions. We will limit ourselves here to some open questions regarding the problems discussed in this article.

1. In Section V we reported big wins for the qubit complexity for integration and path integration. Are there big wins for other problems?
2. Are there problems for which we get big wins for query complexity using randomized queries?
3. Are there tradeoffs between the query complexity and the qubit complexity?
4. What are the classical and quantum complexities of approximating the solution of the Schrödinger equation for a many-particle system? How do they depend on the number of particles? What is the quantum speedup?

Acknowledgements

J. F. Traub is an external professor at the Santa Fe Institute. The research was supported in part by the National Science Foundation.

We are grateful to Erich Novak, University of Jena, and Henryk Woźniakowski, Columbia University and University of Warsaw, for their very helpful comments. We thank Jason Petras, Columbia University, for checking the complexity estimates appearing in the tables.

References

- [1] Abrams, D. S. and Lloyd, S. (1997), Simulation of Many-Body Fermi Systems on a Universal Quantum Computer, *Phys. Rev. Lett.*, 79(13), 2586–2589. Also <http://arXiv.org/quant-ph/9703054>.
- [2] Abrams, D. S. and Lloyd, S. (1999), Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors, *Phys. Rev. Lett.*, 83, 5162–5165.
- [3] Abrams, D. S. and Williams, C. P. (1999), Fast quantum algorithms for numerical integrals and stochastic processes, <http://arXiv.org/quant-ph/9908083>.

- [4] Bakhvalov, N. S. (1977), Numerical Methods, Mir Publishers, Moscow.
- [5] Babuska, I. and Osborn, J. (1991), Eigenvalue Problems, in Handbook of Numerical Analysis, Vol. II, P. G. Ciarlet and J. L. Lions, eds., North-Holland, Amsterdam, 641–787.
- [6] Beals, R., Buhrman, H., Cleve, R., Mosca, R. and de Wolf, R. (1998), Quantum lower bounds by polynomials, Proceedings FOCS'98, 352–361. Also <http://arXiv.org/quant-ph/9802049>.
- [7] Bennett, C. H., Bernstein, E., Brassard, G. and Vazirani, U. (1997) Strengths and weaknesses of quantum computing, *SIAM J. Computing*, 26(5), 1510–1523.
- [8] Bernstein, E., and Vazirani, U. (1997) Quantum complexity theory, *SIAM J. Computing*, 26(5), 1411–1473.
- [9] Berry, D. W., Ahokas, G., Cleve, R., Sanders, B. C. (2007), Efficient quantum algorithms for simulating sparse Hamiltonians, *Communications in Mathematical Physics*, 270(2), 359–371. Also <http://arXiv.org/quant-ph/0508139>
- [10] Bessen, A. J. (2007), On the complexity of classical and quantum algorithms for numerical problems in quantum mechanics, Ph.D. Thesis, Department of Computer Science, Columbia University.
- [11] Boghosian, B. M. and Taylor, W. (1998), Simulating quantum mechanics on a quantum computer, Proceedings of the fourth workshop on Physics and computation, Boston, Massachusetts, 30–42. Also <http://arXiv.org/quant-ph/9701019>
- [12] Brassard, G., Hoyer, P., Mosca, M., and Tapp, A. (2002), Quantum Amplitude Amplification and Estimation, in *Contemporary Mathematics*, Vol. 305, Am. Math. Soc., 53–74. Also <http://arXiv.org/quant-ph/0005055>.
- [13] Brown, K. R., Clark, R. J. and Chuang, I. L. (2006), Limitations of Quantum Simulation Examined by Simulating a Pairing Hamiltonian using Magnetic Resonance, *Phys. Rev. Lett.*, 97(5), 050504. Also <http://arXiv.org/quant-ph/0601021>.
- [14] Cameron, R. H. (1951), A Simpson's rule for the numerical evaluation of Wiener's integrals in function space, *Duke Math. J.*, 8, 111–130.
- [15] Chen, Z., Yezpez, J. and Cory, D. G. (2006), Simulation of the Burgers equation by NMR quantum information processing, *Phys. Rev. A*, 74, 042321. Also <http://arXiv.org/quant-ph/0410198>.
- [16] Chorin, A. J. (1973), Accurate evaluation of Wiener integrals, *Math. Comp.*, 27, 1–15.
- [17] Cleve, R., Ekert, A., Macchiavello, C. and Mosca, M. (1996), Quantum Algorithms Revisited, *Proc. R. Soc. Lond. A.*, 454(1969), 339–354.

- [18] Collatz, L. (1960), *The Numerical Treatment of Differential Equations*, Springer-Verlag, Berlin.
- [19] Courant, C. and Hilbert, D. (1989), *Methods of Mathematical Physics, Vol. I*, Wiley Classics Library, Wiley-Interscience, New York.
- [20] Curbera, F. (2000), Delayed curse of dimension for Gaussian integration, *J. Complexity*, 16(2), 474–506.
- [21] Dawson, C. M., Eisert, J. and Osborne, T. J. (2007), Unifying variational methods for simulating quantum many-body systems, <http://arxiv.org/abs/0705.3456v1>.
- [22] Demmel, J. W. (1997), *Applied Numerical Linear Algebra*, SIAM, Philadelphia.
- [23] Egorov, A. D., Sobolevsky, P. I. and Yanovich, L. A. (1993), *Functional Integrals: Approximate Evaluation and Applications*, Kluwer Academic Publishers, Dordrecht.
- [24] Feynman, R. P. (1982), Simulating physics with computers, *Int. J. Theor. Phys.*, 21:476.
- [25] Forsythe, G. E., and Wasow, W. R. (2004), *Finite-Difference Methods for Partial Differential Equations*, Dover, New York.
- [26] Grover, L. (1997), Quantum mechanics helps in searching for a needle in a haystack, *Phys. Rev. Lett.*, 79(2), 325–328. Also <http://arXiv.org/quant-ph/9706033>.
- [27] Heinrich, S. (2002), Quantum Summation with an Application to Integration, *J. Complexity*, 18(1), 1–50. Also <http://arXiv.org/quant-ph/0105116>.
- [28] Heinrich, S. (2003), From Monte Carlo to Quantum Computation, Proceedings of the 3rd IMACS Seminar on Monte Carlo Methods MCM2001, Salzburg, *Special Issue of Mathematics and Computers in Simulation*, K. Entacher, W. Ch. Schmid, A. Uhl, eds., 62, 219–230.
- [29] Heinrich, S. (2003), Quantum integration in Sobolev spaces, *J. Complexity*, 19, 19–42.
- [30] Heinrich, S. (2004), Quantum Approximation II. Sobolev Embeddings, *J. Complexity*, 20, 27–45. Also <http://arXiv.org/quant-ph/0305031>.
- [31] Heinrich, S. (2006), The randomized complexity of elliptic PDE, *J. Complexity*, 22(2), 220–249.
- [32] Heinrich, S. (2006), The quantum query complexity of elliptic PDE, *J. Complexity*, 22(5), 691–725.
- [33] Heinrich, S., Kwas, M. and Woźniakowski, H. (2004), Quantum Boolean Summation with Repetitions in the Worst-Average Setting, in *Monte Carlo and Quasi-Monte Carlo Methods 2002*, H. Niederreiter, ed., Springer-Verlag, 27–49.

- [34] Heinrich, S. and Milla, B. (2007), The randomized complexity of initial value problems, Talk presented at First Joint International Meeting between the American Mathematical Society and the Polish Mathematical Society, Warsaw, Poland.
- [35] Heinrich, S. and Novak, E. (2002), Optimal summation by deterministic, randomized and quantum algorithms, in Monte Carlo and Quasi-Monte Carlo Methods 2000, K.-T. Fang, F. J. Hickernell and H. Niederreiter eds., Springer-Verlag, Berlin.
- [36] Jaksch, P. and Papageorgiou, A. (2003), Eigenvector approximation leading to exponential speedup of quantum eigenvalue calculation, *Phys. Rev. Lett.*, 91, 257902. Also <http://arXiv.org/quant-ph/0308016>.
- [37] Jordan, S. P. (2005), Fast Quantum Algorithm for Numerical Gradient Estimation, *Phys. Rev. Lett.*, 95, 050501. Also <http://arXiv.org/quant-ph/0405146>.
- [38] Kacewicz, B. Z. (1984), How to increase the order to get minimal-error algorithms for systems of ODEs, *Numer. Math.*, 45, 93-104.
- [39] Kacewicz, B. Z. (2006), Almost optimal solution of initial-value problems by randomized and quantum algorithms. *J. Complexity*, 22(5), 676–690.
- [40] Keller, H. B. (1968), Numerical methods for two-point boundary-value problems, Blaisdell Pub. Co., Waltham, MA.
- [41] Knuth, D. E. (1997), The Art of Computer Programming Volume:2 Seminumerical Algorithms, Addison-Wesley Professional, 3rd edition., Cambridge, MA.
- [42] Kwas, M. (2005), Quantum algorithms and complexity for certain continuous and related discrete problems, Ph.D. thesis, Department of Computer Science, Columbia University.
- [43] Kwas, M. and Li, Y. (2003), Worst case complexity of multivariate Feynman-Kac path integration, *J. Complexity*, 19, 730–743.
- [44] Manin, Y. (1980), Computable and Uncomputable, Sovetskoye Radio, Moscow (in Russian).
- [45] Manin, Y. I. (1999), Classical computing, quantum computing, and Shor’s factoring algorithm, <http://arXiv.org/quant-ph/9903008>.
- [46] Morita, S. and Nishimori, H. (2007), Convergence of Quantum Annealing with Real-Time Schrödinger Dynamics, *Journal of the Physical Society of Japan*, 76(6), 064002. Also <http://arXiv.org/quant-ph/0702252>.
- [47] Nayak, A. and Wu, F. (1999), The quantum query complexity of approximating the median and related statistics, Proc. STOC 1999, 384–393. Also <http://arXiv.org/quant-ph/9804066>.

- [48] Nielsen, M.A. and Chuang, I.L. (2000), *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK.
- [49] Novak, E. (1988), *Deterministic and Stochastic Error Bounds in Numerical Analysis*, Lecture Notes in Mathematics 1349, Springer-Verlag, Berlin.
- [50] Novak, E. (2001), Quantum complexity of integration, *J. Complexity*, 17, 2–16. Also <http://arXiv.org/quant-ph/0008124>.
- [51] Ortiz, G., Gubernatis, J. E., Knill, E. and Laflamme, R. (2001) Quantum algorithms for fermionic simulations, *Phys. Rev. A*, 64(2), 022319. Also <http://arXiv.org/cond-mat/0012334>
- [52] Papageorgiou, A. (2004), Average case quantum lower bounds for computing the boolean mean, *J. Complexity*, 20(5), 713–731.
- [53] Papageorgiou, A. (2007), On the complexity of the multivariate Sturm-Liouville eigenvalue problem, *J. Complexity* (to appear).
- [54] Papageorgiou, A. and Traub, J. F. (2005), Qubit complexity of continuous problems, <http://arXiv.org/quant-ph/0512082>.
- [55] Papageorgiou, A. and Woźniakowski, H. (2005), Classical and Quantum Complexity of the Sturm-Liouville Eigenvalue Problem, *Quantum Information Processing*, 4(2), 87–127. Also <http://arXiv.org/quant-ph/0502054>.
- [56] Paredes, B., Verstraete, F. and Cirac, J. I. (2005), Exploiting Quantum Parallelism to Simulate Quantum Random Many-Body Systems, *Phys. Rev. Lett.*, 95, 140501. Also <http://arXiv.org/cond-mat/0505288>.
- [57] Plaskota, L. (1996), *Noisy Information and Computational Complexity*, Cambridge University Press, Cambridge, UK.
- [58] Plaskota, L., Wasilkowski, G. W. and Woźniakowski, H. (2000), A new algorithm and worst case complexity for Feynman-Kac path integration, *J. Comp. Phys.*, 164(2), 335–353.
- [59] Ritter, K. (2000), *Average-Case Analysis of Numerical Problems*, Lecture Notes in Mathematics, 1733, Springer, Berlin.
- [60] Shor, P. W. (1997), Polynomial-time algorithms for prime factorization and discrete logarithm on a quantum computer, *SIAM J. Comput.*, 26(5), 1484–1509.
- [61] Somma, R., Ortiz, G., Knill, E., Gubernatis (2003), Quantum Simulations of Physics Problems, in Proc. SPIE 2003, Volume 5105, *Quantum Information and Computation*, Andrew R. Pirich, Howard E. Brant eds., 96–103. Also <http://arXiv.org/quant-ph/0304063>.

- [62] Sornborger, A. T. and Stewart, E. D. (1999), Higher Order Methods for Simulations on Quantum Computers, *Phys. Rev. A*, 60(3), 1956–1965. Also <http://arXiv.org/quant-ph/9903055>.
- [63] Strang, G. and Fix, G. J. (1973), *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ.
- [64] Szkopek, T., Roychowdhury, V., Yablonovitch, E. and Abrams, D. S. (2005), Eigenvalue estimation of differential operators with a quantum algorithm, *Phys. Rev. A*, 72, 062318.
- [65] Titchmarsh, E. C. (1958), *Eigenfunction Expansions Associated with Second-Order Differential Equations, Part B*, Oxford University Press, Oxford, UK.
- [66] Traub, J. F. (1999), A continuous model of computation, *Physics Today*, May, 39-43.
- [67] Traub, J. F., Wasilkowski, G. W. and Woźniakowski, H. (1988), *Information-Based Complexity*, Academic Press, New York.
- [68] Traub, J. F., and Werschulz, A. G. (1998), *Complexity and Information*, Cambridge University Press, Cambridge, UK.
- [69] Traub, J. F. and Woźniakowski, H. (1980), *A general theory of optimal algorithms*, ACM Monograph Series, Academic Press, New York.
- [70] Traub, J. F. and Woźniakowski, H. (1992), The Monte Carlo algorithm with a pseudorandom generator, *Math. Comp.*, 58(197), 323–339.
- [71] Traub, J. F. and Woźniakowski, H. (2002), Path integration on a quantum computer, *Quantum Information Processing*, 1(5), 365–388, 2002. Also <http://arXiv.org/quant-ph/0109113>.
- [72] Wasilkowski, G. W. and Woźniakowski, H. (1996), On tractability of path integration, *J. Math. Phys.*, 37(4), 2071–2088.
- [73] Weinberger, H. F. (1956), *Upper and Lower Bounds for Eigenvalues by Finite Difference Methods*, Communications on Pure and Applied Mathematics, IX, 613–623.
- [74] Weinberger, H. F. (1958), *Lower Bounds for Higher Eigenvalues by Finite Difference Methods*, Pacific Journal of Mathematics, 8(2), 339–368.
- [75] Werschulz, A. G. (1991), *The Computational Complexity of Differential and Integral Equations*, Oxford University Press, Oxford.
- [76] Wisner, S. (1996), Simulations of Many-Body Quantum Systems by a Quantum Computer, <http://arXiv.org/quant-ph/96>
- [77] Woźniakowski, H. (2006), The Quantum Setting with Randomized Queries for Continuous Problems, *Quantum Information Processing*, 5(2), 83–130.

- [78] Yezpez, J. (2002), An efficient and accurate quantum algorithm for the Dirac equation, <http://arXiv.org/quant-ph/0210093>.
- [79] Zalka, C. (1998), Simulating quantum systems on a quantum computer, *Proc. Royal Soc. Lond. A*, 454(1969), 313–322. Also <http://arXiv.org/quant-ph/9603026>.