Quantum algorithms for continuous problems and their applications *

A. Papageorgiou and J. F. Traub
Department of Computer Science
Columbia University
New York, NY 10027

December 15, 2011

Abstract

Many problems in science and engineering are formulated using continuous mathematical models. Usually they can only be solved numerically and therefore approximately. Since they are often difficult to solve on a classical computer it's interesting to investigate whether they can be solved faster on a quantum computer.

After a brief introduction to quantum algorithms we report on a wide range of applications including high dimensional integration, path integration, Hamiltonian simulation, and ground state energy estimation. We provide a rather extensive bibliography.
PACS numbers: 03.67.Ac, 03.67.Lx, 02.60.-x

1 Introduction

Problems in science and engineering are frequently formulated using continuous models. In most cases they can only be solved numerically and, therefore, approximately to within a given accuracy ε. Examples include

*This work has been supported in part by the National Science Foundation.

multivariate integration, path integration, function approximation, the solution of ordinary and partial differential equations, optimization and eigenvalue problems. There are numerous applications requiring the solution of these problems ranging from physics and chemistry to economics and finance.

Typically, the solution of a problem depends on an underlying function, often of many variables. The algorithm solving the problem must obtain information about the function, for example, by sampling it at a number of points, and then it must combine the information to produce the result. The computational complexity (for brevity, the complexity) of a problem is the least number of resources required to solve a problem with accuracy ε. In this paper the resources will be the information operations, the combinatorial operations and the space, e.g., the number of qubits.

There have been decades of research on the classical complexity of continuous problems, e.g., see the monographs [62, 74, 76, 89, 90, 88, 98, 65, 66]. Over the last decade there has been a significant amount of work on algorithms and complexity of continuous problems in the quantum setting. This research was motivated by the results of Shor [78] for integer factorization, and Grover [35] for searching an unstructured database.

The challenges of quantum computing are:

- To find quantum algorithms that are better than any known classical algorithm for solving certain continuous problems.

- To determine for which continuous problems quantum computers are provably more powerful than classical computers.

Similar challenges apply to discrete problems, however, we do not deal with them in this paper. An extensive review of quantum algorithms for discrete problems can be found in [59]. The first challenge, which is weaker than the second, allows us to consider problems for which we do not know optimal classical algorithms or we do not have sharp bounds for the classical complexity. Important problems, such as eigenvalue estimation for multiparticle systems, fall in this category. On the other hand, there are problems, such as path integration, where we know the quantum complexity and that the optimal quantum algorithm is faster than any classical algorithm that has the same accuracy.

There are certain ideas and techniques that are broadly applicable and have led to a significant number of results for continuous problems in the quantum setting. We discuss them briefly.

The amplitude amplification and estimation algorithm of Brassard et al. [13] has been applied in the study of integration, path integration, the solution of ordinary differential equations and other problems. Using it we obtain a quantum algorithm that approximates the mean of a Boolean function. This can be extended to an algorithm approximating the weighted average of N numbers. Hence, it can be used to approximate integrals. As a result, using the amplitude amplification and estimation algorithm we can convert classical algorithms for integration, as well as modules of algorithms for other problems that need to compute integrals, to quantum algorithms more or less directly.

The quantum lower bounds of Nayak and Wu [60] establishing the optimality of the algorithm in [13] for computing the Boolean mean also lead to lower bounds for the cost of quantum algorithms and the complexity not only of integration but other continuous problems as well.

For the simulation of quantum systems, splitting formulas [83, 84] have been used to derive quantum algorithms [61, 32, 58]. They are used to implement efficiently approximations of matrix exponentials that are required in a number of quantum algorithms. An example is the algorithm solving linear systems in [37]. Phase estimation [61] has had an impact. It is used in eigenvalue estimation problems such as those of differential operators. The solution of the time-independent Schrödinger equation is an example.

2 The model of computation

We begin by summarizing the model of computation that is used for classical algorithms solving continuous problems before we discuss the quantum model of computation. This will motivate the approach taken in the analysis of quantum algorithms allowing one to draw an analogy between the classical and quantum model of computation.

In the study of the classical complexity of continuous problems the *real number model* with *oracles* is often used. In this model one can compute function evaluations or linear functionals as information operations, see e.g. [89]. The information operations are represented as *black box* or *oracle* calls. One can also perform arithmetic operations, comparisons and evaluate elementary functions. This model of computation is an abstraction of fixed precision floating point arithmetic used in science and engineering. It has also been used in the study of the complexity of algebraic problems such as

matrix multiplication [81]. A comparison of the real number model and the Turing machine model of computation can be found in [87].

Continuous problems are typically defined for classes of functions of $d \geq 1$ variables and are to be solved with accuracy ε. As a result the cost of the algorithms and the problem complexity are studied with respect to the parameters d and ε.

For some continuous problems, such as certain zero finding problems, convex optimization and the solution of linear systems with well conditioned matrices the complexity depends logarithmically on ε^{-1}. However, for the majority of continuous problems the complexity grows much faster. A continuous problem is considered to be tractable if its complexity is proportional to

$$d^{p_1}\varepsilon^{-p_2} \quad \text{for some } p_1, p_2 \in \mathbb{R}. \tag{1}$$

We stress that a problem's complexity depends on the setting in which it is studied. Indeed, for multivariate integration of smooth functions the cost of any classical algorithm with worst case accuracy ε can grow at least as $\varepsilon^{-\alpha d}$, for some $\alpha > 0$. Then the problem suffers from the curse of dimensionality and is intractable in the worst case. In some cases, classical randomized algorithms can break intractability because they exhibit a polynomial dependence on ε^{-1} and, as we will see, quantum algorithms provide an additional speedup.

2.1 Quantum queries

Inputs to quantum algorithms are often given using quantum queries. They correspond to black box or oracle calls returning evaluations of some function f. The model of Beals et al. [6] has been used in the study of discrete problems and, with a slight change in the definition of the queries, it has also been used in the study of continuous problems.

In the case of a Boolean function $f : \{0, \ldots, 2^m - 1\} \to \{0, 1\}$ the quantum query providing information about f is defined by the unitary operator

$$Q_f|j\rangle|k\rangle = |j\rangle|k \oplus f(i)\rangle, \tag{2}$$

where $|j\rangle$ is an m qubit computational basis state, $|k\rangle$ is single qubit computational basis state and $\oplus$ denotes addition modulo 2. This type of query is used in Grover's search algorithm.

In the case of real valued bounded functions different quantum queries have been studied in the literature. Without loss of generality assume that $f : \{0, \ldots, 2^m - 1\} \to [0, 1]$. Abrams and Williams [3] in their study of integration used the query

$$\begin{aligned} Q_f|j\rangle|0\rangle &= \sqrt{1 - f(j)^2}|j\rangle|0\rangle + f(j)|j\rangle|1\rangle \\ Q_f|j\rangle|1\rangle &= -f(j)|j\rangle|0\rangle + \sqrt{1 - f(j)^2}|j\rangle|1\rangle. \end{aligned} \tag{3}$$

We point out that this query is defined using the real number $f(j)$. Therefore, depending on f it may be very hard to implement the query exactly using elementary quantum gates. A truncation of the value $f(j)$ to a finite number of most significant bits can be used to overcome this difficulty. Often, truncations of the function evaluations to a number of significant bits proportional to ε^{-1} can be used without loss of generality.

Novak [63] in his paper studying the complexity of integration on Hölder[1] classes used the query

$$\begin{aligned} Q_f|j\rangle|0\rangle &= \sqrt{f(j)}|j\rangle|0\rangle + \sqrt{1 - f(j)}|j\rangle|1\rangle \\ Q_f|j\rangle|1\rangle &= -\sqrt{1 - f(j)}|j\rangle|0\rangle + \sqrt{f(j)}|j\rangle|1\rangle. \end{aligned} \tag{4}$$

The above considerations about truncating the function evaluations apply to this case too.

A third kind of query was introduced by Heinrich in [38]. Namely

$$Q_f|j\rangle|k\rangle = |j\rangle|k \oplus \hat{f}(j)\rangle, \tag{5}$$

where $|j\rangle$ and $|k\rangle$ are m and ν qubit computational basis states respectively, $\hat{f}(j)$ is obtained from $f(j)$ using a ν bit discretization of the range of f and $\oplus$ denotes addition modulo 2^ν. Moreover, in the case of real functions of real variables $g : [0, 1]^d \to [0, 1]$, $d \geq 1$, it suffices to consider a discretization $\tau : \{0, 1, \ldots, 2^m - 1\} \to [0, 1]^d$ of the domain of g and to use the query definition above with the function $f(j) = g(\tau(j))$, $j = 0, \ldots, 2^m - 1$.

The three queries are not equivalent, in general. Query (5) can be used to efficiently simulate queries (3) and (4) but the converse is not true [11].

[1] A function belongs to a Hölder class or a Sobolev space if it satisfies certain smoothness express by conditions on its partial derivatives. A reader not familiar with these concepts may think of a function with bounded partial derivatives up to a given order.

2.2 Quantum algorithms

Consider a problem defined using a linear or nonlinear operator S such that

$$S : \mathcal{F} \to \mathcal{G}. \tag{6}$$

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 $f(t_1), \ldots, f(t_n)$ at deterministically and a priori chosen sample points. The quantum query Q_f encodes this information and provides it to the algorithm.

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, 24, 38, 61]. 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 $\mathcal{H}_\nu = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$, ν times, for some appropriately chosen integer ν, where $\mathbb{C}^2$ is the two dimensional space of complex numbers. The dimension of $\mathcal{H}_\nu$ 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 $\mathcal{H}_\nu$ and is obtained from the initial state $|\psi_0\rangle$ through a sequence of unitary $2^\nu \times 2^\nu$ matrices, i.e.,

$$|\psi\rangle_f := U_T Q_f U_{T-1} Q_f \cdots U_1 Q_f U_0 |\psi_0\rangle. \tag{7}$$

The unitary matrix Q_f is a quantum query and as we already mentioned it is used to provide information about a function f. Q_f depends on n function evaluations $f(t_1), \ldots, f(t_n)$, at deterministically chosen points, $n \leq 2^\nu$. The selection of the query among the types (2), (3), (4) or (5) is often a matter of convenience.

The matrices $U_0, U_1, \ldots, U_T$ are unitary and do not depend on f. The integer T denotes the number of quantum queries.

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 $j \in \{0, 1, \ldots, M-1\}$ 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].

We consider algorithms that approximate $S(f)$ with probability $p \geq \frac{2}{3}$. 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 (7) that computes the approximation $\phi_f(j)$, for $f \in \mathcal{F}$ and the outcome $j \in \{0, 1, \ldots, M-1\}$, is defined by

$$e(\phi_f, S) \;=\; \min\left\{ \alpha : \sum_{j:\ \|S(f)-\phi_f(j)\| \leq \alpha} p_f(j) \geq \tfrac{2}{3} \right\},$$

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

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

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

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

The query complexity gives a sense of the depth of the quantum circuit realizing the algorithm and provides a complexity lower bound. It allows one to study algorithms and obtain complexity results for classes of functions in a way that is unobscured by the cost of a query, which varies with f. This is how the cost is measured in Grover's search algorithm [35, 61].

The cost for combining the queries to produce the result, i.e., the implementation cost of the unitary operators $U_0, \ldots, U_t$ must also be taken into account. The complexity of the problem is the minimal cost, including the queries and other quantum operations, of an algorithm solving the problem with accuracy ε. An algorithm with cost equal to the complexity, modulo an absolute constant, is considered to be optimal.

Some papers in the literature consider only the query complexity. In some cases, this is a simplification. In other cases, the query complexity, modulo polylog factors, reflects the total cost of the optimal algorithm as well. On the other hand, not all papers we review consider the query complexity alone.

A number of them give estimates of the query complexity as well as the cost of other quantum operations.

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

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

3 Applications

3.1 Integration

There are numerous applications where one seeks the expected value of a quantity and, therefore, is confronted with an integral. Often these are integrals of multivariate functions. The number of variables can be huge, say, in the hundredths or thousandths. In most cases the integrals cannot be computed analytically and their values are approximated numerically.

Classical algorithms for integration have been extensively studied in the literature and optimal algorithms are known for numerous classes of functions, e.g., see [89] and the references therein.

In many cases, the optimal algorithms are linear. Thus the optimal algorithm approximating

$$S(f) = \int_{I_d} f(x)\, dx, \tag{9}$$

where $I_d \subset \mathbb{R}^d$, has the form

$$A(f) = \sum_{j=0}^{N-1} a_j f(x_j), \tag{10}$$

where a_j, $j = 0, \ldots, N-1$, are independent of f and x_j, $j = 0, \ldots, N-1$, are deterministic or random sample points. An example of particular interest is the algorithm

$$A(f) = \frac{1}{N} \sum_{j=0}^{N-1} f(x_j). \tag{11}$$

For $I_d = [0,1]^d$ this is the midpoint rule in d dimensions that samples f on a uniform grid. The Monte Carlo algorithm that samples f at random points also has this form.

Quantum algorithms can compute an approximation to $A(f)$ fast. This gives them an advantage over classical algorithms for integration as we will see below. This idea was used by Abrams and Williams [3] who were the first to derive a quantum algorithm for integration. They use query (3) to provide the necessary function evaluations to their algorithm. Novak [63] was the first to study the quantum complexity of integration in Hölder classes of functions. His algorithm uses query (4). Soon after, Heinrich studied the quantum complexity of integration in Sobolev spaces using the query (5). He has obtained a large number of results [38, 39, 40, 45, 44].

We will use [63] to illustrate the key ideas in the derivation of quantum algorithms for integration. We will start with an algorithm approximating the average of N real numbers and then use it to obtain an algorithm for integration in Hölder classes. Consider a function $f : \{0, \ldots, N-1\} \to [0, 1]$ and $N = 2^n$ and the average of equation (11). To approximate the more general sums (10) it suffices to express them as an average using a suitable transformation.

The approximation of the Boolean mean is a special simple case. For a function $f : \{0, \ldots, N-1\} \to \{0, 1\}$, the amplitude amplification and estimation algorithm of Brassard et al. [13] computes an approximation of the mean (11) with error ε using $O(\varepsilon^{-1})$ queries. Moreover, the lower bounds of Nayak and Wu [60] show that this algorithm is optimal modulo constants.

Considering the mean (11) of a real valued function, using the query (4) we have

$$\begin{aligned} Q_f(H^{\otimes n} \otimes I)|0\rangle^{\otimes n}|0\rangle &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \left[\sqrt{f(j)}|j\rangle|0\rangle + \sqrt{1-f(j)}|j\rangle|1\rangle \right] \\ &= a_0|\psi_0\rangle + a_1|\psi_1\rangle, \end{aligned}$$

where H is the Hadamard gate, $N = 2^n$, $a_0^2 = \frac{1}{N}\sum_{j=0}^{N-1} f(j)$, $a_0^2 + a_1^2 = 1$, $\langle\psi_0|\psi_1\rangle = 0$ and

$$\begin{aligned} |\psi_0\rangle &= \frac{1}{a_0\sqrt{N}} \sum_{j=0}^{N-1} \sqrt{f(j)}|j\rangle|0\rangle \\ |\psi_1\rangle &= \frac{1}{a_1\sqrt{N}} \sum_{j=0}^{N-1} \sqrt{1-f(j)}|j\rangle|1\rangle, \end{aligned}$$

have unit length.

Then $A(f) = a_0^2$ and the modification of the amplitude amplification and estimation algorithm that uses T queries of the form (4) (instead of queries of the form (2)) approximates $A(f) = \alpha_0^2$ with error [13, Th. 12]

$$2\pi k \frac{\sqrt{a(1-a)}}{T} + k^2 \frac{\pi^2}{T^2},$$

and its success probability is at least

$$\begin{array}{ll} 1 - \frac{1}{2(k-1)} & k > 1 \\ \frac{8}{\pi^2} & k = 1. \end{array}$$

For functions of real variables, the smoothness of the functions determines lower bounds on the cost of the algorithms and the complexity of integration. There are different ways to define classes of smooth functions using a condition on their (partial) derivatives. Recall that in the Hölder classes of functions the growth of all partial derivatives up to a given order is restricted is a certain way, as we see in the definition below. These classes have been considered extensively in the study of classical algorithms for integration. Similarly, Sobolev spaces are also classes of smooth functions and we discuss them briefly at the end of this section. We define Hölder classes before stating Novak's integration results.

Definition 1. *The Hölder class $F_d^{k,\alpha}$ is the class of all k-times continuously differentiable functions $f : [0,1]^d \to \mathbb{R}$ with $\|f\|_\infty \leq 1$ that satisfy*

$$|\partial^l f(x) - \partial^l f(y)| \leq \|x - y\|^\alpha$$

for all partial derivatives $\partial^l = \partial_1^{l_1} \cdots \partial_d^{l_d}$ of order $|l| = l_1 + \cdots + l_d = k$.

The integration algorithm of Novak [63] proceeds as follows. A function $f \in F_d^{k,\alpha}$ is approximated using a piecewise polynomial $P_n(f)$ that interpolates f at n points. This way an evaluation of $f - P_n(f)$ has cost independent of n, although the cost may depend on d and k. Also $|f - P_n(f)| = \theta(n^{-\gamma})$, $\gamma = (k + \alpha)/d$.

Then

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

Since the integral of $P_n(f)$ can be computed exactly it suffices to approximate the third integral above. Using the notation of equation (9) we approximate

$I(f - P_n(f))$ by the average $A(f - P_n(f))$ as in equation (11). All the steps are classical except the approximation of $A(f - P_n(f))$ which is done by the quantum algorithm above. Appropriately selecting the values of n and N in a way that the algorithm approximates the integral with accuracy ε and with probability at least $\frac{3}{4}$, Novak shows that, modulo polylog factors, the query complexity of integration in Hölder classes satisfies

$$\mathrm{comp}^{\mathrm{query}}(\varepsilon) \asymp \varepsilon^{-1/(1+\gamma)} \quad \gamma = (k+\alpha)/d.$$

For other classes of functions, such as Sobolev spaces $W^r_{p,d}$, variations of this approach lead to algorithms with optimal query complexity. Table 1 summarizes the query complexity results (up to polylog factors) for multivariate integration in the worst case, randomized and quantum setting for functions belonging to Hölder classes $F_d^{k,\alpha}$ and Sobolev spaces $W^r_{p,d}$. An example of a classical randomized algorithm for integration is the well known Monte Carlo algorithm. Observe that the integration problem suffers from the curse of dimensionality in the classical worst case, see e.g. [89]. Quantum algorithms offer an exponential speedup over classical algorithms in the worst case and a polynomial speedup over classical randomized algorithms. Heinrich who obtained most of the quantum query complexity results in a series of papers, which we cited earlier, summarized his results in [39] where a corresponding table showing error bounds can be found.

	Worst case	Randomized	Quantum
$F_d^{k,\alpha}$	$\varepsilon^{-d/(k+\alpha)}$	$\varepsilon^{-2d/(2(k+\alpha)+d)}$	$\varepsilon^{-d/(k+\alpha+d)}$
$W^r_{p,d}, \quad 2 \le p \le \infty$	$\varepsilon^{-d/r}$	$\varepsilon^{-2d/(2r+d)}$	$\varepsilon^{-d/(r+d)}$
$W^r_{p,d}, \quad 1 \le p \le 2$	$\varepsilon^{-d/r}$	$\varepsilon^{-pd/(rp+pd-d)}$	$\varepsilon^{-d/(r+d)}$
$W^r_{1,d}$	$\varepsilon^{-d/r}$	$\varepsilon^{-d/r}$	$\varepsilon^{-d/(r+d)}$

Table 1: Complexity of integration in Sobolev spaces and Hölder classes

Quantum algorithms for integration have been used to derive optimal quantum algorithms for other continuous problems, such as path integration, certain approximation problems, and the solution of ordinary differential equations.

3.2 Path integration

Traub and Wońiakowski [91] study quantum algorithms and the complexity of path integration. Path integrals can be viewed as infinite dimensional

integrals. They are defined by

$$I(f) = \int_X f(x)\,\mu(dx),$$

where μ is a probability measure on X, an infinite dimensional space in general, $f : X \to \mathbb{R}$ belongs to a class F, of μ-integrable functions. In particular, they consider a Gaussian measure μ where the eigenvalues of its covariance operator are of order j^{-k}, $k > 1$. The Wiener measure is an example with $k = 2$. They also assume that F is the class of functions with rth Frechet derivative ($r < \infty$) continuous and uniformly bounded by one.

We describe the idea leading to the algorithm in [91]. First approximate $I(f)$, with error ε, by a d-dimensional integral

$$I_d(f) = \int_{\mathbb{R}^d} f_d(t)\,\mu_d(df),$$

where μ_d is a zero mean Gaussian measure. This implies that $d = d(\varepsilon)$ is a polynomial in ε^{-1} and its degree depends on k. Then approximate $I_d(f)$, with error ε, using an algorithm $A(f)$ of the form (10). The number of terms in the sum is $N = m^d$ and m is a polynomial in ε^{-1}. Thus, N is an exponential function of ε^{-1}. Finally, use a quantum algorithm for integration to approximate the value $A(f)$ with error ε.

The resulting algorithm has error proportional to ε. It uses a number of queries proportional to ε^{-1}. The number of additional quantum operations is polynomial in ε^{-1}. The number of qubits is also polynomial in ε^{-1}.

On the other hand, the classical complexity of path integration has been considered in [97, 25]. In the worst case it is of order $\varepsilon^{-\varepsilon^{-\beta}}$, where β is a positive number that depends on r. Hence, the problem is intractable in the worst case. Approximating the finite dimensional integral $I_d(f)$ using Monte Carlo leads to a classical randomized algorithm with cost proportional to ε^{-2}, which is an optimal classical algorithm.

In summary we have:

- Path integration on a quantum computer is tractable.
- Path integration on a quantum computer can be solved roughly ε^{-1} times faster than on a classical computer using randomization, and exponentially faster than on a classical computer with a worst case assurance.

The Feynman-Kac path integral is a special case of a path integral and occurs in many applications [29]. In this case $X = C$ the space of continuous functions and the measure is the Wiener measure $\mu = w$. For example 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 $V, v : \mathbb{R}^d \to \mathbb{R}$, are the potential and the initial value functions, respectively, and Δ denotes the Laplacian. The solution is given by the Feynman-Kac path integral

$$z(u,t) = \int_C v(x(t)+u) e^{\int_0^t V(x(s)+u)\,ds}\, w(dx), \tag{12}$$

where C is the set of continuous functions $x : \mathbb{R}_+ \to \mathbb{R}^d$ 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 and u is a function of d variables.

Kwas [53] following an approach similar to that of Traub and Woźniakowski derived a quantum algorithm for Feynman-Kac path integration that uses a number of queries proportional to ε^{-1}. He also showed that a slightly more complicated algorithm that uses a number of queries proportional

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

is optimal.

For comparison, we briefly discuss the classical complexity of Feynman-Kac path integration. For $d = 1$ when u is a scalar there is an number of papers dealing with the solution of (12); see e.g. [17]. In particular, for $u = 1$ and V four times continuously differentiable, Chorin's well known algorithm has cost proportional to $\varepsilon^{-2.5}$. Plaskota et al. [75] where the first to study the complexity in the worst case. They construct and algorithm with cost $\varepsilon^{-0.25}$ and show it is optimal. We remark that the algorithm depends on a numerically difficult precomputation. Multivariate Feynman-Kac integration is studied in [54] in the worst case and in [53]. In the worst case the complexity is $\varepsilon^{-d/r}$, for v and V that are $r < \infty$ times continuously differentiable. In the randomized case the curse of dimensionality is broken. An algorithm based on Monte Carlo has cost of order ε^{-2}. We remark that the quantum

algorithm of Kwas uses a quantum algorithm for integration as a module instead of Monte Carlo. Finally, as in the quantum case, a more complicated randomized algorithm with cost proportional to $\varepsilon^{-2/(1+2r/d)}$ is optimal.

3.3 Approximation

Classically, approximation of functions of d variables has been studied for functions in Sobolev spaces $W^r_{p,d}$ with error measured using the norm of L_q. Its complexity depends on the values of the parameters [62, 88] and the accuracy ε. Here r is a smoothness parameter, d is the number of variables and $1 \leq p \leq \infty$ indicates the norm is the L_p norm. For $p = \infty$ the problem suffers the curse of dimensionality in the classical worst and randomized case, i.e the cost of any classical deterministic or randomized algorithm grows exponentially with the number of variables d. Recently, Heinrich [41] showed that quantum algorithms do not provide an advantage.

Table 2 summarizes the classical and quantum complexity of approximation (modulo polylog factors) in Sobolev spaces for the various values of the parameters p, q, r, d.

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

Table 2: Complexity of approximation in Sobolev spaces

On the other hand, there are approximation problems for which quantum algorithms have an advantage over classical algorithms. Novak et al. [64] study such problem. They consider a space of functions of d variables, where certain variables are more important than others. Weights are used to define the relative importance of the variables. They show a quantum algorithm that is exponentially faster than any classical algorithm in the worst case, and is roughly $\varepsilon^{-(1+r)}$ times faster than any classical randomized algorithm.

The parameter r depends on the weights and can be large. Moreover, the quantum algorithm uses about $d + \log \varepsilon^{-1}$ qubits.

3.4 Ordinary differential equations

Quantum algorithms for initial value problems for systems of first order equations and scalar equations of higher order have been studied in the literature. In both cases the algorithms are derived from classical algorithms by taking the modules that compute integrals classically and replacing them by a quantum algorithm for integration.

Kacewicz [50] studied the problem

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

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

He derived a quantum algorithm with cost that differs from the lower bound by only an arbitrarily small parameter in the exponent. In particular, its cost (modulo polylog factors) is

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

where $\gamma \in (0, 1)$ is arbitrarily small, while the quantum complexity satisfies

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

The complexity of classical randomized algorithms is also studied by Kacewicz in the same paper. He derived a randomized algorithm with cost

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

modulo polylog factors, where $\gamma \in (0, 1)$ is arbitrarily small, and showed the complexity lower bound

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

Much earlier he had studied the classical worst case complexity of ordinary differential equations [49].

Goćwin and Szczesny [34] considered quantum and classical randomized algorithms for the solution of

$$\begin{cases} u^{(k)}(x) = g(x, u(x), u'(x), \ldots, u^{(q)}), & x \in [a, b], \\ u^{(j)}(a) = u_a^j, & j = 0, 1, \ldots, k-1, \end{cases}$$

where $0 \leq q < k$, $g : [a.b] \times \mathbb{R}^{q+1} \to \mathbb{R}$, $u : [a, b] \to \mathbb{R}$ $(a < b)$. They showed the same complexity upper and lower bounds as the ones above hold for any k.

Finally we mention that Heinrich and Milla [46] recently showed that the randomized complexity lower bound of Kacewicz holds with $\gamma = 0$.

3.5 Partial differential equations

The numerical solution of partial differential equations is a vast subject. Here we confine ourselves to elliptic equations. They have many applications and classical algorithms for solving them have been extensively studied in the literature, see [98] and the references within. A simple example is the Poisson equation, for which we want to find a function $u : \bar{\Omega} \to \mathbb{R}$, that satisfies

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

where Δ denotes the Laplacian and $\Omega \subset \mathbb{R}^d$.

Heinrich [42] studied the quantum query complexity of 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 function belonging to $C^r(\Omega)$ and the error measured in the L_∞ norm, and (modulo polylog factors) found it proportional to

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

We note that classical randomized algorithms have cost at least proportional to

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

and this lower bound is sharp [43]. In the worst case the problem has complexity proportional to $\varepsilon^{-d/r}$ and is intractable.

Hence, quantum algorithms may have a polynomial advantage over classical algorithms, but not always. For fixed m and r and for $d > 4m$ the problem is intractable in the quantum and classical settings.

3.6 Optimization

The query complexity of finding the maximum of a multivariate function belonging to the Hölder class $F_d^{d\alpha}$ is studied in [33]. Lower bounds are derived using the results of [60]. An optimal quantum algorithm is also shown. The main idea is to discretize the function and then use an algorithm that finds the maximum of a finite sequence. The latter algorithm is based on that in [28]. In particular, the query complexity is

$$\mathrm{comp}^{\mathrm{query}}(\varepsilon) = \Theta(\varepsilon^{-d/(2(r+\alpha))}).$$

The classical worst case and randomized complexity of this problem is well known [62]. In both cases it is $\Theta(\varepsilon^{-d(k+\alpha)})$. Thus quantum algorithms provide a quadratic speedup relative to classical algorithms.

3.7 Gradient estimation

Approximating the gradient of a function $f : \mathbb{R}^d \to \mathbb{R}$ with accuracy ε requires a minimum of $d + 1$ function evaluations on a classical computer. Jordan [48] 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., $f(x_1, \ldots, x_d) = \sum_{j=1}^{d} a_j x_j$, and is uniformly bounded by 1. Then $\nabla f = (a_1, \ldots, a_d)^T$. Using a single query and *phase kickback* we obtain the state

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

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

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

This is equal to the state

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

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

For functions with second order partial derivatives not identically equal to zero the analysis is more complicated. We refer the reader to [48] for the details.

3.8 Simulation

In 1982 Richard Feynman [32] observed that simulating quantum systems would be difficult or impossible on a classical computer. The number of parameters describing the quantum states grows exponentially with the system size and so does the computational cost of the best classical deterministic algorithms known. In some cases classical randomized algorithms have been used to overcome these difficulties, however, randomized algorithms also have limitations. As an alternative to simulation with a classical computer Feynman proposed simulation with a quantum computer. He conjectured that quantum computers might be able to carry the simulation more efficiently than classical computers. For an overview of quantum simulation see, e.g., [32, 58, 51, 16].

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

Lloyd [58] showed that local Hamiltonians can be simulated efficiently on a quantum computer. About the same time, Zalka [105, 104] showed that many-particle systems can be simulated efficiently on a quantum computer. Later, Aharonov and Ta-Shma [4] generalized Lloyd's results to sparse Hamiltonians. We note that Hamiltonian simulation is also related to adiabatic evolution and quantum walks [31, 21, 30, 19, 20].

Berry et al. [10] extended the complexity results of [4] for sparse Hamiltonians. They assume that the Hamiltonian H is given by an *oracle* (a *black-box*) and that H can be decomposed efficiently, by a quantum algorithm using oracle calls, into a sum of Hamiltonians H_j, $j = 1, \ldots, m$, that individually can be simulated efficiently. They approximate e^{-iHt} with error

ε by a sequence of N unitary operators of the form $e^{-iH_{j_\ell}t_{j_\ell}}$, $\ell = 1, \ldots, N$. The cost of the simulation is the total number of oracle calls. All the unitary operators in the sequence have to be considered in the simulation, one after the other. The algorithm has to make oracle calls to each Hamiltonian appearing in the sequence and to simulate it. Each oracle call to any H_j is simulated by making oracle calls to H; see [10, Sec. 5] for details. Thus the total number of oracle calls is proportional to N, although it is not equivalent since there can be overhead in implementing each $e^{-iH_{j_\ell}t_{j_\ell}}$, $\ell = 1, \ldots, N$.

In particular, let $H = \sum_{j=1}^{m} H_j$, where $e^{-iH_j t}$, $t \in \mathbb{R}$, can be implemented efficiently, and the H_j do not commute, $j = 1, \ldots, m$. Consider algorithms approximating e^{-iHt}, $t \in \mathbb{R}$, that are obtained using Suzuki's high order splitting formulas [83, 84]. These algorithms have the form

$$\prod_{l=1}^{N} e^{-iH_{j_l}t_{j_l}}, \tag{13}$$

for suitable $t_{j_l} \in \mathbb{R}$, where $j_l \in \{1, \ldots, m\}$. The cost of the simulation of H is proportional to the number of exponentials, N, so that

$$\left\| e^{-iHt} - \prod_{l=1}^{N} e^{-iH_{j_l}t_{j_l}} \right\| \leq \varepsilon.$$

Berry et al. [10] show that

$$N \leq N_{\text{prev}} := m5^{2k}(m\|H_1\|t)^{1+\frac{1}{2k}}\varepsilon^{-1/(2k)}, \tag{14}$$

where the splitting formula is of order $2k+1$ and $\|H_1\| \geq \|H_1\| \geq \cdots \geq \|H_m\|$. They also derive the value of k that minimizes the upper bound. Papageorgiou and Zhang [72] improve this estimate for N by showing

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

From this they also derive an improved estimate for the k that minimizes the upper bound.

There are many applications of these estimates. In [10] they are used along with the decomposition cost of H and the simulation cost of the individual H_j, $j = 1, \ldots, m$, to derive the overall simulation cost. Recently,

Childs and Kothari [22] used the estimates in [10] in the simulation of sparse Hamiltonians with star decompositions.

A more general Hamiltonian simulation problem is studied by Wiebe et al. [100] who derive an estimate similar to (14).

Besides the papers mentioned above there is a large and varied literature on Hamiltonian simulation. Many papers deal with particular algorithms. There are no tight complexity bounds. The following list of papers, which is by no means complete, is of interest. Abrams and Lloyd [1] show algorithms for the simulation of many-body Fermi systems. Brown et al. [15] establish limits of quantum simulation. Boghosian and Taylor [12] present efficient algorithms simulating quantum mechanical systems. Bravyi et al. [14] show an efficient algorithm for the simulation of weakly interacting quantum spin systems. Buluta and Nori [16] provide an overview of quantum simulators. Chen et al. [18] study the simulation of the Burgers equation. Kassal et al. [52, 51] deal with quantum simulation in chemistry. Ortiz et al. [68] study algorithms for fermionic simulations. Paredes et al. [73] present an algorithm that exploits quantum parallelism to simulate randomness. Somma et al. [79, 80] study the quantum simulation of physics problems. Whitfield et al. [99] study the quantum simulation of electronic Hamiltonians. Wiesner [101] studies the quantum simulation of many-body systems. Wu et al. [102] study the simulation of pairing models on a quantum computer. Yepez [103] presents an efficient algorithm for the many-body three-dimensional Dirac equation.

3.9 Eigenvalue estimation

The estimation of the ground state eigenvalue of a time-independent Hamiltonian corresponding to a multiparticle system is an important problem in physics and chemistry. Decades of calculating ground state eigenvalues of systems with a large number of particles have suggested that such problems are hard on a classical computer. That is why researchers have been experimenting with quantum computers to solve eigenvalue problems in quantum chemistry with very encouraging results [27, 55].

In fact, there has been a fair amount of work dealing with eigenvalue problems see, e.g. [14, 57, 36, 67, 86, 92, 94, 96]. See also [51, 52] and the references therein.

Abrams and Lloyd [2] were the first to observe that the ground state eigenvalue of the Born-Oppenheimer electronic Hamiltonian [85, p. 43] can

be approximated on a quantum computer using the phase estimation algorithm [61, Fig. 5.2]. Phase estimation is not limited to this particular Hamiltonian eigenvalue problem but is broadly applicable, provided its requirements are met with reasonable cost for the problem at hand.

One requirement is that the second register of its initial state should contain an approximation of the eigenvector corresponding to the eigenvalue of interest. For instance, for the estimation of the ground state eigenvalue one needs an approximate ground state eigenvector. This approximation does not need to be very precise. It suffices that the magnitude of its projection on the actual eigenvector is not exponentially small. The success probability of the algorithm depends on the quality of the approximate eigenvector. In some cases such approximations can be computed efficiently by quantum algorithms [47], in other cases quantum algorithms designed to prepare general quantum states [94] are used to prepare the approximate eigenvector, or it is empirically or randomly chosen [2, 57].

The second requirement is the implementation of powers of the unitary matrix U that phase estimation uses. In the case of the time-independent Schrödinger equation $U = e^{i\gamma H}$, where H denotes the system Hamiltonian and γ is a suitable constant that aims to ensure the phase corresponding to the eigenvalue of interest belongs to $[0, 1)$. Then one needs to derive the cost simulating the U^{2^t}, $t = 0, \ldots, b-1$, so that the algorithm has accuracy $2^{-b} \leq \varepsilon$, with high probability. We remark that the powers of U do not have to be simulated very accurately, because the simulation error only affects the success probability of phase estimation. The algorithms for Hamiltonian simulation we discussed previously are used to approximate the powers of U. Thus the total cost of phase estimation includes the simulation cost.

For the approximation of the ground state eigenvalue (ground state energy) of the time-independent Schrödinger equation the form of the Hamiltonian H used in phase estimation depends on the way the eigenvalue problem is approached. One possibility is to obtain H by spatially discretizing the time-independent Schrödinger equation. An advantage of this is that one solves the problem for a class of potentials. Another possibility is to use the Born-Oppenheimer electronic Hamiltonian in the second quantized form [85, p. 89] in phase estimation. We discuss both alternatives below.

Papageorgiou et al. [71] using a spatial discretization of the Schrödinger equation provide rigorous estimates of the cost and the success probability of the phase estimation. Their algorithm prepares the initial state, and simulates all the U^{2^t}, $t = 0, \ldots, b-1$. We remark that there are some

similarities between their approach and that already used by Lidar and Wang [57] for the calculation of the thermal rate constant.

In particular, Papageorgiou et al. consider the approximation of the smallest eigenvalue E_1 of the equation

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

where ∂I_d denotes the boundary of the unit cube, x is the position variable, and Ψ_1 is a normalized eigenfunction. For simplicity, they assume that all masses and the normalized Planck constant are one. The boundary conditions are for particles in a box. Multiparticle systems on bounded domains with the wave function equal to zero on the boundary have been studied in the literature; see e.g. [23, p. 621].

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 p, e.g., $d = 3p$. For many applications the number of particles p and hence d is huge. Moreover, it is assumed that V and its first order partial derivatives $\partial V/\partial x_j$, $j = 1, \ldots, d$, are continuous and uniformly bounded by 1.

To approximate E_1 with relative error proportional to ε observe that the finite difference discretization of the operator $-\frac{1}{2}\Delta + V$ on a regular grid with mesh size $h = \varepsilon$ yields a matrix $H = -\frac{1}{2}\Delta_h + V_h$, whose smallest eigenvalue $E_{h,1}$ approximates E_1 with relative error $O(\varepsilon)$. The matrix size is $\varepsilon^{-d} \times \varepsilon^{-d}$.

Phase estimation approximates $E_{h,1}$ using:

1. The state
$$|\psi_1\rangle^{\otimes d}$$
in the second register of the initial state. This is an estimate of the eigenvector corresponding to $E_{h,1}$, where $|\psi_1\rangle^{\otimes d}$ is the ground state eigenvector of $-\Delta_h$, which are implemented efficiently using the quantum Fourier transform with a number of quantum operations proportional to $d(\log \varepsilon^{-1})^2$.

2. Suzuki's [83, 84] high order splitting formulas to simulate the unitaries U^{2^t}, $t = 0, \ldots, b-1$, where $U = e^{iH/(2d)}$. These splitting formulas use exponentials involving $-\frac{1}{2}\Delta_h$ and V_h, respectively. The former are implemented using the quantum Fourier transform with cost proportional to $d(\log \varepsilon^{-1})^2$. The latter, involving the evaluations of the potential, are implemented using quantum queries.

The overall simulation error is at most 1/20 using

$$O\left(\varepsilon^{-3} e^{\sqrt{\ln \frac{1}{d\varepsilon}}}\right) \quad \text{as } d\varepsilon \to 0.$$

matrix exponentials.

The errors due to the approximation of the ground state eigenvector and the simulation of the exponentials affect the success probability of phase estimation. However, it remains at least 2/3. Its total cost, including the number of queries and the number of all other quantum operations, is

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

where $\delta > 0$ is arbitrarily small and C is a constant. The number of qubits is

$$cd \log \varepsilon^{-1}.$$

where c is a constant.

Tight quantum complexity bounds for the ground state eigenvalue problem are not known. On the other hand the cost of any classical algorithm in the worst case with respect to V grows exponentially with d. Indeed, consider a potential function V and let $\bar{V}$ be a perturbation of V. Then the eigenvalue $E_1(V)$ corresponding to V and the eigenvalue $E_1(\bar{V})$ corresponding to $\bar{V}$ are related according to the formula

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

where $\Psi_1(\cdot; \bar{V})$ denotes the eigenfunction corresponding to $E_1(\bar{V})$. This implies that approximating E_1 is at least as hard as approximating the multivariate integral involving V in the worst case. As a result, any classical deterministic algorithm for the eigenvalue problem with accuracy ε must use a number of function evaluations of V that grows as ε^{-d}; see [70] for details.

As we indicated, for multiparticle systems many papers consider the Born-Oppenheimer electronic Hamiltonian in the second quantized form, see e.g., [5, 52, 69, 93, 95, 99]. It is given by

$$H = \sum_{p,q} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s,$$

where h_{pq} and h_{pqrs} are one- and two-electron integrals, respectively, in a molecular spin orbital basis, $p, q, r, s = 1 \ldots, M$; see [99, Eq. 3 and Eq. 4], and M is the number of basis functions. So we have $M^2 + M^4$ integrals. The values of these integrals are considered known since they are computed classically once the basis functions are chosen. The a_j, $a_j^\dagger$ are fermionic annihilation and creation operators, respectively, $j = 1, \ldots, M$. We have the anti-commutation relations

$$a_j a_i + a_i a_j = 0, \quad \text{and} \quad a_j a_i^\dagger + a_i^\dagger a_j = \delta_{ij} I, \quad i, j = 1, \ldots, M, \tag{15}$$

where δ_{ij} is the Kronecker delta and I is the identity operator.

Whitfield et al. [99], Ovrum and Hjorth-Jensen [69] and Veis and Pittner [93] take similar approaches for estimating the ground state eigenvalue of H. Our discussion is based on [99]. The Jordan-Wigner transformation is used to map the creation and annihilation operators $a_j^\dagger$ and a_j to products of Pauli matrices. This transformation is defined by

$$\begin{aligned} a_j &\to \sigma_j^+ \left(\prod_{k=j+1}^{M} \sigma_k^z \right) \\ a_j^\dagger &\to \sigma_j^- \left(\prod_{k=j+1}^{M} \sigma_k^z \right), \end{aligned} \tag{16}$$

where $\sigma_j^s = 1 \otimes \cdots \otimes \sigma^s \otimes 1 \cdots \otimes 1$, σ^s is applied to the jth qubit, $s \in \{z, +, -\}$, and

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \sigma^+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \sigma^- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

See [99, Eq. 5a and Eq. 5b] for details.

This yields the Pauli representation of the Hamiltonian

$$H = \sum_{j=1}^{K} H_j,$$

with $K = O(M^4)$. For simplicity we assume here that each of the H_j corresponds to a term of the original Hamiltonian. Phases estimation is used to approximate the ground state eigenvalue of this Hamiltonian. The exponentials $e^{-iH_j t}$, $t \in \mathbb{R}$, $j = 1, \ldots, K$, can be implemented using $O(M)$ elementary quantum gates. We remark that in [99] extra care is taken to group the terms

in a way that each of the resulting ones can be implemented using quantum circuit primitives with cost $O(M)$, and this is an important feature of their algorithm. In either case, the cost of implementing the exponentials of all the terms in H is $O(M^5)$.

In phase estimation the Trotter formula is used to simulate the exponential of the Hamiltonian as follows

$$e^{i\gamma H 2^t} \approx \left(\prod_{j=1}^{K} e^{i\gamma H_j 2^t / N_t} \right)^{N_t} \qquad t = 0, \ldots, b-1,$$

where b is the number of qubits in the first register of phase estimation upon which the accuracy depends. The error of this approximation is bounded by $c\gamma 2^t \sum_{j=1}^{K} \|H_j\| N_t^{-1}$,, see [82, Th. 3], where c is constant independent of K and the H_j, $j = 1, \ldots, K$. Often, the values of the N_t, $t = 0, \ldots, b-1$, are chosen empirically in practice. Similarly, the value of b is determined by considerations, such as reasonable chemical accuracy, and is relatively small.

3.10 Linear systems

Many applications require the solution of systems of linear equations. There is an extensive literature about classical algorithms for this problem; see [26, 77] and the references therein. Recently Harrow et al. [37] derived a quantum algorithm for this problem. We sketch this algorithm.

Consider the linear system $Ax = b$, where A is an $N \times N$ Hermitian matrix. Harrow et al. derive an quantum algorithm that computes the solution of $A|x\rangle = |b\rangle$. They assume that the singular values of A belong to $[\kappa^{-1}, 1]$ (so the condition number $K(A)$ of A satisfies $K(A) \leq \kappa$) and that b is a unit vector that has been quantum mechanically implemented and is given as state $|b\rangle$. The algorithm does not output $|x\rangle$ classically. The solution is available as a quantum state so that one can compute functionals involving it. For instance, an expectation $\langle x|M|x\rangle$, for a given M.

Let λ_j be the eigenvalues and $|u_j\rangle$, $j = 1, \ldots, N$ be the normalized eigenvectors of A. Then $|b\rangle = \sum_{j=1}^{N} \beta_j |u_j\rangle$. Consider phase estimation as in [61, Fig. 5.2] with the state $|b\rangle$ in the second register, and the conditional Hamiltonian evolution

$$\sum_{k=0}^{T-1} |k\rangle\langle k| \otimes e^{iAkt_0/T},$$

where $t_0 = O(\kappa\varepsilon^{-1})$, T is a sufficiently large number, and ε is the desired accuracy in the solution of the system. After the inverse Fourier transform is applied at the top register the state is

$$\sum_{j=1}^{N} \beta_j \sum_{k=0}^{T-1} \alpha(j,k)|k\rangle|u_j\rangle.$$

Then $|\alpha(j,k)|$ is large for the indices $k_j \in \{0,\ldots,T-1\}$ that lead to good approximations $\lambda_j \approx \frac{2\pi k_j}{t_0} =: \tilde{\lambda}_j$, and are small for the remaining indices, $j = 1,\ldots,N$.

Neglecting the terms that do not lead to good approximations of the eigenvalues of A we have

$$\sum_{j=1}^{N} \beta_j \alpha(j,k_j)|k_j\rangle|u_j\rangle.$$

Adding a qubit and performing a conditional rotation we get

$$\sum_{j=1}^{N} \beta_j \alpha(j,k_j)|k_j\rangle|u_j\rangle \left(\sqrt{1 - C^2/\tilde{\lambda}_j^2}|0\rangle + C/\tilde{\lambda}_j|1\rangle\right),$$

where $C = O(\kappa^{-1})$. At this point we do not need the $|k_j\rangle$ any longer and we undo phase estimation. To further simplify the analysis we assume that $\alpha(j,k_j) = 1$, $j = 1,\ldots,N$. So we have the state

$$\sum_{j=1}^{N} \beta_j |u_j\rangle \left(\sqrt{1 - C^2/\tilde{\lambda}_j^2}|0\rangle + C/\tilde{\lambda}_j|1\rangle\right),$$

If we measure the last qubit and the outcome is 1 the system collapses to the state

$$\gamma \sum_{j=1}^{N} \frac{\beta_j}{\tilde{\lambda}_j}|u_j\rangle,$$

which is an approximation to the solution $|x\rangle$ of the linear system, modulo the factor $\gamma = (\sum_{j=1}^{N} C^2|\beta_j|^2/|\tilde{\lambda}_j|^2)^{-1/2}$, which is the reciprocal of the square root of the probability to obtain outcome 1. This probability is $\Omega(\kappa^{-2})$ because $C = O(\kappa^{-1})$. Thus $O(\kappa)$ steps of amplitude amplification [13] are sufficient to boost this probability.

Assuming that A is s-sparse and taking into account the cost of simulating e^{iAt}, as well as the error of phase estimation the total cost of the quantum algorithm is proportional to

$$s \log N \, \kappa^2 \varepsilon^{-1}.$$

The best general purpose classical algorithm for the solution of linear systems is the conjugate gradient algorithm [77] and has cost proportional to

$$sN\sqrt{\kappa} \log \varepsilon^{-1}$$

for a positive definite matrix A, and

$$sN\kappa \log \varepsilon^{-1}$$

otherwise.

This quantum algorithm is very efficient for systems involving matrices of huge size with condition number polynomial in $\log N$, assuming ε is not arbitrarily small. However, in many applications the matrix size N depends on the desired accuracy ε, and grows as $\varepsilon \to 0$. This can be true for the condition number $K(A)$ as well. Thus the dependence of N and $K(A)$ on ε determines if there is an advantage of the quantum algorithm.

Two recent papers apply this quantum algorithm to the solution of first order differential equations [56, 9]. Since neither addresses the relationship between N and ε for solving the differential equations with error ε it is hard to draw a conclusion about the efficiency of the quantum algorithm in these cases.

However, a detailed analysis of the performance of the quantum algorithm for the linear systems involved in the solution of differential equations with error ε may reveal a significant advantage. Consider, for example, the systems obtained from the discretization of second order elliptic partial differential equations in d dimensions. Then we can have $N = \varepsilon^{-d}$ and $K(A) = \Theta(\varepsilon^{-2})$. Observe that classical algorithm that solves the system must have cost at least proportional to the matrix size $N = \varepsilon^{-d}$, i.e., exponential in d. On the other hand, the quantum algorithm whose cost depends on the logarithm of N, and the condition number $K(A)$, is exponentially faster than the classical algorithm when d is large.

Acknowledgements

We thank Iasonas Petras for his comments and suggestions.

References

[1] D. S. Abrams and S. Lloyd. Simulation of many-body fermi systems on a universal quantum computer. *Phys. Rev. Lett.*, 79(13):2586–2589, 1997. Also http://arxiv.org/abs/quant-ph/9703054.

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

[3] D. S. Abrams and C. P. Williams. Fast quantum algorithms for numerical integrals and stochastic processes. *Report NASA Jet Propulsion Laboratory*, 1999. Also http://arXiv.org/abs/quant-ph/9908083.

[4] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. *Proc. 35th Annual ACM Symp. on Theory of Computing*, page 359, 2003.

[5] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707, 2005.

[6] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Proc. FOCS 1998*, pages 352–361, 1998. Also http://arxiv.org/abs/quant-ph/9802049.

[7] C. H. Bennet, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Computing*, 26(5):1510–1523, 1997.

[8] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM J. Computing*, 26(5):1411–1473, 1997.

[9] D. W. Berry. Quantum algorithms for solving linear differential equations, 2010. http://arxiv.org/abs/1010.2745.

[10] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders. Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications in Mathematical Physics*, 270:359, 2007.

[11] A. Bessen. The power of various real-valued queries. *J. Complexity*, 20(5):699–712, 2004.

[12] B. M. Boghosianm and W. Taylor. Simulating quantum mechanics on a quantum computer. *Proc. fourth workshop on Physics and computation, Boston, MA*, pages 30–42, 1998. Also http://arXiv.org/abs/quant-ph/9701019.

[13] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp. *Quantum Amplitude Amplification and estimation*, volume 305, page 53. in Contemporary Mathematics, Quantum Computation and Information, Samuel J. Lomonaco Jr. and Howard E. Brandt, Editors, AMS, Providence, RI, 2002. Also http://arXiv.org/abs//quant-ph/0005055.

[14] S. Bravyi, D. DiVincenzo, and D. Loss. Polynomial-time algorithm for simulation of weakly interacting quantum spin systems. *Comm. Math. Phys.*, 284:481–507, 2008.

[15] K. R. Brown, R. J. Clark, and I. L. Chuang. Limitations of quantum simulation examined by simulating a pairing hamiltonian using magnetic resonance. *Phys. Rev. Lett.*, 97(5):050504, 2006. Also http://arXiv.org/abs/quant-ph/0601021.

[16] I. Buluta and F. Nori. Quantum simulators. *Science*, 326:108, 2009.

[17] R. M. Cameron. A Simpson's rule for the numerical evaluation of Wiener's integrals in function space. *Duke Math. J.*, 8:111–130, 1951.

[18] Z. Chen, J. Yepez, and D. G. Cory. Simulation of the Burgers equation by NMR quantum information processing. *Phys. Rev. A*, 74:042321, 2006. Also http://arXiv.org/abs/quant-ph/0410198.

[19] A. Childs. Universal computation by quantum walk. *Phys. Rev. Lett.*, 102:180501, 2009.

[20] A. Childs. Continuous- and discrete-time quantum walk. *Communications in Mathematical Physics*, 294:581–603, 2010.

[21] A. Childs, E. Farhi, and S. Gutmann. An example of the difference between quantum and classical random walks. *J. Quant. Inf. Proc.*, 1:35–43, 2002.

[22] A. Childs and R. Kothari. Simulating sparse hamiltonians with star decompositions. In *Proceedings of the 5th conference on Theory of quantum computation, communication, and cryptography*, TQC'10, pages 94–103, Berlin, Heidelberg, 2011. Springer-Verlag. Also http://arXiv.org/abs/1003.3683.

[23] P. G. Ciarlet and C. Le Bris. *Handbook of Numerical Analysis, Special Volume Computational Chemistry*, volume X. North Holland, Amsterdam, 2003.

[24] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proc. R. Soc. Lond. A.*, 454:339–354, 1996.

[25] F. Curbera. Delayed curse of dimension for gaussian integration. *J. Complexity*, 16(2):474–506, 2000.

[26] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[27] J. Du, N. Xu, X. Peng, P. Wang, S. Wu, and D. Lu. Nmr implementation of a molecular hydrogen quantum simulation with adiabatic state preparation. *Phys. Rev. Lett.*, 104:030502, 2010.

[28] C. Dür and P. Hoyer. A quantum algorithm for finding the minimum. *Proc. of the 30th Ann. ACM Symposium on Theory of Computing*, pages 1516–1524, 1998.

[29] A. D. Egorov, P. I. Sobolevsky, and L. A. Yanovich. *Functional Integrals: Approximate Evaluation and Applications*. Kluwer Academic Publishers, Dordrecht, 1993.

[30] E. Farhi, J. Goldstone, and S. Gutmann. A quantum algorithm for the hamiltonian nand tree, 2007. http://arXiv.org/abs/quant-ph/0702144.

[31] E. Farhi, J. Goldstone, and M. Sipser. Quantum computation by adiabatic evolution, 2000. http://arXiv.org/abs/quant-ph/0001106.

[32] R. P. Feynman. Simulating physics with computers. *Int. J. Theoret. Phys.*, 21:467–488, 1982.

[33] M. Goćwin. On the complexity of searching for a maximum of a function on a quantum computer. *Quantum Information Processing*, 5(1):31–41, 2006.

[34] M. Goćwin and M. Szczesny. Randomized and quantum algorithms for solving initial-value problems in ordinary differential equations of order k. *Opuscula Mathematica*, 28(3):247–277, 2008.

[35] L. Grover. A fast quantum algorithm for database search. *Proceedings 28 Annual Symposium on the Theory of Computing, ACM Press, New York*, pages 212–219, 1996.

[36] A. Hams and H. DeRaedt. Fast algorithm for finding the eigenvalue distribution of very large matrices. *Phys. Rev. E*, 62(3):4365–4377, Sep 2000.

[37] A. W. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, 2009.

[38] S. Heinrich. Quantum summation with an application to integration. *j. Complexity*, 18(1):1–50, 2002. Also http://arXiv.org/abs/quant-ph/0105116.

[39] S. Heinrich. 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, 2003.

[40] S. Heinrich. Quantum integration in Sobolev spaces. *J. Complexity*, 19:19–42, 2003.

[41] S. Heinrich. Quantum approximation II. Sobolev embeddings. *J. Complexity*, 20:27–45, 2004.

[42] S. Heinrich. The quantum query complexity of elliptic PDE. *J. Complexity*, 22(5):691–725, 2006.

[43] S. Heinrich. The randomized complexity of elliptic PDE. *J. Complexity*, 22(2):220–249, 2006.

[44] S. Heinrich, M. Kwas, and H. Woźniakowski. Quantum boolean summation with repetitions in the worst-average setting. *in Monte Carlo and Quasi-Monte Carlo Methods 2002, H. Niederreiter, ed.*, pages 27–49, 2004.

[45] S. Heinrich and E. Novak. Optimal summation by deterministic, randomized and quantum algorithms. *in Monte Carlo and Quasi-Monte Carlo Methods 2000, K.-T. Fang, F. J. Hickernell and H. Niederreiter eds., Springer-Verlag, Berlin*, 2002.

[46] S. Heirich and B. Milla. The randomized complexity of initial value problems. *J. Complexity*, 24(2):77–88, 2008.

[47] P. Jaksch and A. Papageorgiou. Eigenvector approximation leading to exponential speedup of quantum eigenvalue estimation. *Phys. Rev. Lett.*, 91:257902, 2003.

[48] S. P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Phys. Rev. Lett.*, 95:050501, 2005.

[49] B. Z. Kacewicz. How to increase the order to get minimal-error algorithms for systems of odes. *Numer. Math.*, 45:93–104, 1984.

[50] B. Z. Kacewicz. Almost optimal solution of initial-value problems by randomized and quantum algorithms. *J. Complexity*, 22(5):676–690, 2006.

[51] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik. Polynomial-time quantum algorithm for the simulation of chemical dynamics. *Proc. Nat. Acad. Sci.*, 105:18681, 2008.

[52] I. Kassal, J. D. Whitfield, A. Perdomo-Ortiz, M.-H. Yung, and A. Aspuru-Guzik. Simulating chemistry using quantum computers. *Ann. Rev. Phys. Chem.*, 62:185–207, 2011.

[53] M. Kwas. Quantum algorithms and complexity for certain continuous and related discrete problems, 2005. Ph.D. thesis, Department of Computer Science, Columbia University.

[54] M. Kwas and Y. Li. Worst case complexity of multivariate Feynman-Kac path integration. *J. Complexity*, 19:730–743, 2003.

[55] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White. Towards quantum chemistry on a quantum computer. *Nature Chemistry*, 2:106–111, 2010.

[56] S. K. Leyton and T. Osborne. A quantum algorithm to solve nonlinear differential equations, 2008. http://arxiv.org/abs/0812.4423.

[57] D. A. Lidar and H. Wang. Calculating the thermal rate constant with exponential speedup on a quantum computer. *Phys. Rev. E*, 59:2429–2438, 1999.

[58] S. Lloyd. Universal quantum simulators. *Science*, 273:1073–1078, 1996.

[59] M. Mosca. *Quantum algorithms*, page 7088. in Encyclopedia of Complexity and Systems Science, Vol. 8, Springer, New York, 2009.

[60] A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. *Proc. 31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 384–393, 1999.

[61] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.

[62] E. Novak. *Deterministic and Stochastic Error Bounds in Numerical Analysis*. Lecture Notes in Mathematics 1349, Springer-Verlag, Berlin, 1988.

[63] E. Novak. Quantum complexity of integration. *J. Complexity*, 19(1):19–42, 2001.

[64] E. Novak, I. H. Sloan, and H. Woźniakowski. Tractability of approximation for weighted korobov spaces on classical and quantum computers. *Journal of Foundations of Computational Mathematics*, 4:121–156, 2004.

[65] N. Novak and H. Woźniakowski. *Tractability of Multivariate Problems, Volume I: Linear Information*. European Mathematical Society, Zurich, 2008.

[66] N. Novak and H. Woźniakowski. *Tractability of Multivariate Problems, Volume II: Standard Information for Functionals*. European Mathematical Society, Zurich, 2010.

[67] S. Oh. Quantum computational method of finding the ground-state energy and expectation values. *Phys. Rev. A*, 77(1):012326, Jan 2008.

[68] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme. Quantum algorithms for fermionic simulations. *Phys. Rev. A*, 64(2):022319, 2001. Also http://arXiv.org/abs/cond-mat/0012334.

[69] E. Ovrum and M. Hjorth-Jensen. Quantum computation algorithm for many-body studies. *arXiv:quant-ph/0705.1928*, 2007.

[70] A. Papageorgiou. On the complexity of the multivariate Sturm-Liouville eigenvalue problem. *J. Complexity*, 23(4-6):802–827, 2007.

[71] A. Papageorgiou, I. Petras, J. F. Traub, and C. Zhang. A fast algorithm for approximating the ground state energy on a quantum computer, 2010. http://xxx.lanl.gov/abs/1008.4294.

[72] A. Papageorgiou and C. Zhang. On the efficiency of quantum algorithms for Hamiltonian simulation. *Quantum Information Processing*, (to appear), 2011. Also http://arxiv.org/abs/1005.1318.

[73] B. Paredes, F. Verstraete, and J. I. Cirac. Exploiting quantum parallelism to simulate quantum random many-body systems. *Phys. Rev. Lett.*, 95:140501, 2005. Also http://arXiv.org/abs/cond-mat/0505288.

[74] L. Plaskota. *Noisy Information and Computational Complexity*. Cambridge University Press, Cambridge, UK, 1996.

[75] L. Plaskota, G. W. Wasilkowski, and H. Woźniakowski. A new algorithm and worst case complexity for Feynman-Kac path integration. *J. Comp. Phys.*, 164(2):335–353, 2000.

[76] K. Ritter. *Average-Case Analysis of Numerical Problems*. Lecture Notes in Mathematics, 1733, Springer, Berlin, 2000.

[77] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, PA, 2003.

[78] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA*, pages 124–134, 1994. arXiv.org/abs/quant-ph/9508027.

[79] R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme. Simulating physical phenomena by quantum networks. *Phys. Rev. A*, 65:042323, 2002.

[80] R. Somma, G. Ortiz, E. Knill, and J. E. Gubernatis. Quantum simulations of physics problems. *Proc. SPIE 2003 Quantum Information and Computation, Andrew R. Pirich, Howard E. Brant eds.*, 5105:96–103, 2003. Also http://arXiv.org/abs/quant-ph/0304063.

[81] V. Strassen. Gaussian elimination is not optimal. *Num. Math.*, 13:354–356, 1969.

[82] M. Suzuki. Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to man y-body problems. *Comm. Math. Phys.*, 51(2):183–190, 1976.

[83] M. Suzuki. Fractal decomposition of exponential operators with applications to many body theories and Monte Carlo simulations. *Phys. Lett. A*, 146:319–323, 1990.

[84] M. Suzuki. General theory of fractal path integrals with application to many-body theories and statistical physics. *J. Math. Phys.*, 32:400–407, 1991.

[85] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry.* Dover Publications Inc., New York, 1996.

[86] T. Szkopek, V. Roychowdhury, E. Yablonovitch, and D. S. Abrams. Eigenvalue estimation of differential operators with a quantum algorithm. *Phys. Rev. A*, 72(6):062318, Dec 2005.

[87] J. F. Traub. A continuous model of computation. *Physics Today*, May 1999:39–44, 1999.

[88] J. F. Traub, G. Wasilkowski, and H. Woźniakowski. *Information-Based Complexity.* Academic Press, New York, 1988.

[89] J. F. Traub and A. G. Werschulz. *Complexity and Information*. Cambridge University Press, Cambridge, UK, 1998.

[90] J. F. Traub and H. Woźniakowski. *A general theory of optimal algorithms*. ACM Monograph Series, Academic Press, New York, 1980.

[91] J. F. Traub and W. Woźniakowski. Path integration on a quantum computer. *Quantum Information Processing*, 1(5):365–388, 2002.

[92] P. Varga and B. Apagyi. Phase estimation procedure to solve quantum-mechanical eigenvalue problems. *Phys. Rev. A*, 78(2):022337, Aug 2008.

[93] L Veis and J. Pittner. Quantum computing applied to calculations of molecular energies: Ch2 benchmark. *J. Chem. Phys.*, 133:194106, 2010.

[94] H. Wang, S. Ashhab, and F. Nori. Efficient quantum algorithm for preparing molecular-system-like states on a quantum computer. *Phys. Rev. A*, 79:042335, 2009.

[95] H. Wang, S. Kais, A. Aspuru-Guzik, and M. Hoffmann. Quantum algorithms for obtaining the energy spectrum of molecular systems. *Phys. Chem. Chem. Phys.*, 10:5388–5393, 2008.

[96] H. Wang, S. Kais, A. Aspuru-Guzik, and M. R. Hoffmann. Quantum algorithm for obtaining the energy spectrum of molecular systems. *Phys. Chem. Chem. Phys.*, 10:5388–5393, 2008.

[97] G. W. Wasilkowski and H. Woźniakowski. On tractability of path integration. *J. Math. Phys.*, 37(4):2071–2088, 1996.

[98] A. G. Werschulz. *The Computational Complexity of Differential and Integral Equations*. Oxford University Press, Oxford, UK, 1991.

[99] J. Whitfield, J. Biamonte, and A. Aspuru-Guzik. Simulation of electronic structure Hamiltonians using quantum computers. *Molecular Physics*, 109(5):735–750, 2011.

[100] N. Wiebe, D. Berry, P. Hoyer, and B. C. Sanders. Higher order decompositions of ordered operator exponentials. *J. Phys. A: Math. Theor.*, 43:065203, 2010.

[101] S. Wiesner. Simulations of many-body quantum systems by a quantum computer, 1996. http://arXiv.org/abs/quant-ph/9603028.

[102] L.-A. Wu, M. S. Byrd, and D. A. Lidar. Polynomial-time simulation of pairing models on a quantum computer. *Phys. Rev Lett.*, 89(5):057904, 2002.

[103] J. Yepez. An efficient and accurate quantum algorithm for the Dirac equation, 2002. http://arXiv.org/abs/quant-ph/0210093.

[104] C. Zalka. Efficient simulation of quantum systems by quantum computers. *Fortschritte der Physik*, 46:877, 1998.

[105] C. Zalka. Simulating quantum systems on a quantum computer. *Proc. R. Soc. Lond. A*, 454:313, 1998.