Classical and Quantum Complexity of the Sturm-Liouville Eigenvalue Problem

A. Papageorgiou[1] and H. Woźniakowski[2]

[1,2]Department of Computer Science, Columbia University, New York, USA

[2]Institute of Applied Mathematics and Mechanics, University of Warsaw, Poland

March 3, 2005

Abstract

We study the approximation of the smallest eigenvalue of a Sturm-Liouville problem in the classical and quantum settings. We consider a univariate Sturm-Liouville eigenvalue problem with a nonnegative function q from the class $C^2([0,1])$ and study the minimal number $n(\varepsilon)$ of function evaluations or queries that are necessary to compute an ε-approximation of the smallest eigenvalue. We prove that $n(\varepsilon) = \Theta(\varepsilon^{-1/2})$ in the (deterministic) worst case setting, and $n(\varepsilon) = \Theta(\varepsilon^{-2/5})$ in the randomized setting. The quantum setting offers a polynomial speedup with *bit* queries and an exponential speedup with *power* queries. Bit queries are similar to the oracle calls used in Grover's algorithm appropriately extended to real valued functions. Power queries are used for a number of problems including phase estimation. They are obtained by considering the propagator of the discretized system at a number of different time moments. They allow us to use powers of the unitary matrix $\exp(\frac{1}{2}\mathrm{i}M)$, where M is an $n \times n$ matrix obtained from the standard discretization of the Sturm-Liouville differential operator. The quantum implementation of power queries by a number of elementary quantum gates that is polylog in n is an open issue.

In particular, we show how to compute an ε-approximation with probability $\frac{3}{4}$ using $n(\varepsilon) = \Theta(\varepsilon^{-1/3})$ bit queries. For power queries, we use the phase estimation algorithm as a basic tool and present the algorithm that solves the problem using $n(\varepsilon) = \Theta(\log \varepsilon^{-1})$ power queries, $\log^2 \varepsilon^{-1}$ quantum operations, and $\frac{3}{2} \log \varepsilon^{-1}$ quantum bits. We also prove that the minimal number of qubits needed for this problem (regardless of the kind of queries used) is at least roughly $\frac{1}{2} \log \varepsilon^{-1}$. The lower bound on the number of quantum queries is proven in [5].

We derive a formula that relates the Sturm-Liouville eigenvalue problem to a weighted integration problem. Many computational problems may be recast as this weighted integration problem, which allows us to solve them with a polylog number of power queries. Examples include Grover's search, the approximation of the Boolean mean, NP-complete problems, and many multivariate integration problems. In this paper we only provide the relationship formula. The implications are covered in [25].

Keywords: Eigenvalue problem, numerical approximation, quantum algorithms

PACS numbers: 03.67.Lx, 02.60.-x

1 Introduction

The study of the potential power of quantum computers has been a major theoretical challenge. There will be an additional incentive to build a quantum computer if we can identify computationally important problems for which quantum computation offers significant speedups over computation on a classical computer.

For discrete problems, the best known quantum algorithms are due to Shor and Grover, see [27, 13]. Shor's algorithm for factorization has an exponential speedup over all *known* algorithms on a classical computer. Still, we can not yet claim that we have an exponential speedup for this problem, since the complexity of factorization on a classical computer is unknown. Grover's algorithm for data search offers a quadratic speedup.

For continuous problems, quantum complexity is known for linear problems such as multivariate integration, path integration and multivariate approximation, see [14, 15, 16, 17, 24, 32]. For these problems we have an exponential speedup over the worst case setting, and a polynomial speedup over the randomized setting. The first quantum study of a nonlinear continuous problem was done in [20] for ordinary differential equations with polynomial speedups over the classical settings.

The purpose of this paper is to present classical and quantum complexity results of another nonlinear continuous problem. This continuous problem is quite natural and computationally important, since it corresponds to the (simplified) univariate Sturm-Liouville eigenvalue problem. The Sturm-Liouville eigenvalue problem is defined in [10] in full generality. Here it is defined as finding the smallest eigenvalue of the differential operator

$$\mathbb{L}_q u\,(x) \;:=\; -u''(x) \,+\, q(x)\,u(x) \qquad \text{for} \;\; x \in (0,1),$$

with the boundary conditions $u(0) = u(1) = 0$. We assume that the function q is nonnegative and belongs to the class $C^2([0,1])$ of twice continuously differentiable functions whose norm $\|q\| := \max_{i=0,1,2} \max_{x\in[0,1]} |q^{(i)}(x)|$ is bounded by 1. The operator $\mathbb{L}_q$ maps $C^2([0,1])$ into $C([0,1])$.

The Sturm-Liouville eigenvalue problem has been extensively studied in the literature. The properties of the eigenvalues and the eigenfunctions are well known and so are numerical algorithms for approximating them on a classical computer, see, e.g. [2, 9, 10, 19, 28]. Nevertheless, the complexity of approximating the smallest eigenvalue in the worst case and randomized settings, as well as in the quantum setting, has not yet been addressed.

In this paper we study classical and quantum algorithms. We prove bounds on the worst case and randomized complexities on a classical computer, and bounds on the query complexity and on the qubit complexity. We prove that the complexity in the classical settings is a polynomial in ε^{-1}.

We study the quantum setting with *bit* queries and prove polynomial speedups over the classical settings. Bit queries correspond to approximate computation of function values, see [14], and are used in all papers dealing with the quantum study of continuous problems.

We also study the quantum setting with *power* queries. Such queries are formally defined in Section 5.2. Here we only mention that they are used in the phase estimation algorithm, which is the core of many quantum algorithms including Shor's and Grover's algorithms.

Power queries are controlled-W^{p_j} queries for some $n \times n$ unitary matrix W and some exponents p_j. For the phase estimation algorithm, we have $p_j = 2^{j-1}$ for $j = 1, 2, \ldots, m$, with m of order $\log \varepsilon^{-1}$. For the factoring problem of a large integer N, Shor's algorithm uses the unitary matrix W such that power queries can be implemented by at most $O(\log^3 N)$ elementary quantum gates.

For the Sturm-Liouville eigenvalue problem, as well for all problems studied in [25], we use power queries with the specific unitary matrix

$$W = \exp\left(\tfrac{1}{2}\,\mathrm{i}\, M_q\right) \qquad \text{with} \qquad \mathrm{i} = \sqrt{-1}, \tag{1}$$

where M_q is an $n \times n$ real symmetric tridiagonal matrix that is a classical approximation of the differential operator $\mathbb{L}_q$, see Section 3.2. The matrix M_q depends on the values of $q(j/(n+1))$ that appear on the diagonal of M_q for $j = 1, 2, \ldots, n$.

Unitary matrices similar to (1) play a key role in quantum mechanics. They give the solution of the Schrödinger equation, they are the propagator of a system evolving with Hamiltonian M_q, and are important in quantum simulation, see [22]. Zalka [34] deals with their implementation. The crucial point about power queries is that we can use W^j of the matrix W given by (1) as one quantum query for some j. Hence, lower bound results for bit queries do not apply to power queries.

We prove that in the quantum setting with power queries, the Sturm-Liouville eigenvalue problem requires only roughly $\log \varepsilon^{-1}$ power queries with the matrix W of (1). As shown in [25], many computational problems can be reduced to the solution of the Sturm-Liouville eigenvalue problem, and they can be also solved in polylog number of power queries. The list of such problems include Grover's search, NP-complete problems, and many continuous problems. This proves that the quantum setting with power queries with the matrix W of (1) is exponentially more powerful than the quantum setting with bit queries.

We stress that, contrary to Shor's algorithm, we do *not* know if power queries with the $n \times n$ matrix W of (1) can be implemented by a number of existing elementary quantum gates that is polylog in n. We asked a number of colleagues and most of them doubt whether this can be achieved. If this is indeed the case, then the positive results on the polylog number of such power queries will be of only theoretical interest. Still, if a future quantum computer is able to perform such power queries in a polylog number of, perhaps, more general elementary quantum gates or by some other quantum devices, the polylog number of power queries will lead to efficient quantum algorithms, and will allow us to solve many computational problems exponentially faster than on a classical computer. From this point of view, we may interpret the positive results on the number of power queries with the matrix W of (1) as the indication that building a quantum computer with such queries would be a very desirable task which would give us a very powerful computational device.

2 Survey of the Results

In this section we explain our results in more technical terms. For a classical computer, we study the worst case and randomized settings in the real number model of computation with oracles, see [23, 30, 31]. That is, we assume that arithmetic operations (addition, subtraction, multiplication, division, and evaluation of elementary functions), as well as comparisons of real numbers, are performed exactly with cost taken as unity. We also assume that the information about functions q is given by sampling q at finitely many points with the cost of one function evaluation taken as $\mathbf{c}$. Typically $\mathbf{c} \gg 1$.

We want to approximate the smallest eigenvalue $\lambda(q)$ of the operator $\mathbb{L}_q$ to within ε. Let $n(\varepsilon)$ be the smallest number of function values of q needed to compute such an ε-approximation in a given setting. The number $n(\varepsilon)$ is called the *information complexity*. The *complexity*, $\mathrm{comp}(\varepsilon)$, is defined as the minimal total cost of computing an ε-approximation in a given setting. Obviously we have

$$\mathbf{c}\, n(\varepsilon) \,\le\, \mathrm{comp}(\varepsilon).$$

We prove that in both classical settings, the complexity of the Sturm-Liouville eigenvalue problem is polynomial in ε^{-1}, or equivalently is exponential in the number $\lfloor \log \varepsilon^{-1} \rfloor$ of correct bits of a computed approximation. More precisely, there exist positive numbers α_i independent of ε such that:

- in the worst case setting,

$$\begin{array}{rcl} \alpha_1\, \varepsilon^{-1/2} \le & n(\varepsilon) & \le \alpha_2\, \varepsilon^{-1/2}, \\ \alpha_1\, \mathbf{c}\, \varepsilon^{-1/2} \le & \mathrm{comp}(\varepsilon) & \le \alpha_2\, \mathbf{c}\, \varepsilon^{-1/2} \,+\, \alpha_3\, \varepsilon^{-1/2} \log \varepsilon^{-1}, \end{array}$$

- in the randomized setting,

$$\begin{array}{rcl} \alpha_4\, \varepsilon^{-2/5} \le & n(\varepsilon) & \le \alpha_5\, \varepsilon^{-2/5}, \\ \alpha_4\, \mathbf{c}\, \varepsilon^{-2/5} \le & \mathrm{comp}(\varepsilon) & \le \alpha_5\, \mathbf{c}\, \varepsilon^{-2/5} \,+\, \alpha_6\, \varepsilon^{-1/2} \log \varepsilon^{-1}. \end{array}$$

The lower bounds on $n(\varepsilon)$, and consequently on $\mathrm{comp}(\varepsilon)$, are obtained by relating the eigenvalue problem to the integration problem for functions from the unit ball of $C^2([0,1])$. It is well known that the minimal number of function values for this integration problem is bounded from below by roughly $\varepsilon^{-1/2}$ in the worst case setting and by $\varepsilon^{-2/5}$ in the randomized setting; see, e.g., [23, 30] and the survey of these results in [31].

The upper bounds on $n(\varepsilon)$ and $\mathrm{comp}(\varepsilon)$ in the worst case setting are obtained by the cost of the classical algorithm that computes an ε-approximation by the bisection algorithm of the Sturm sequence [33, p. 300], see also [11, Ch. 5.3.4], applied to an $n \times n$ matrix which is the classical discretization of the operator $\mathbb{L}_q$ with $n = \Theta(\varepsilon^{-1/2})$. The matrix depends on n function values of q computed at equidistant points of $[0,1]$. Since we need roughly $\log \varepsilon^{-1}$ bisection steps, and the cost of each step is proportional to n, the total cost is of order $(\mathbf{c} + \log \varepsilon^{-1})\varepsilon^{-1/2}$. Hence, modulo the logarithm of ε^{-1}, the worst case complexity is of order $\mathbf{c}\, \varepsilon^{-1/2}$.

The upper bounds on $n(\varepsilon)$ and $\mathrm{comp}(\varepsilon)$ in the randomized setting are obtained by the following algorithm. We first approximate the function q by a natural cubic spline $\bar{q}$ using n deterministic sample points of q at equidistant points of $[0,1]$ with $n = \Theta(\varepsilon^{-2/5})$. The relationship between the smallest eigenvalue and integration problems, see Section 3, states that

$$\lambda(q) \,=\, \lambda(\bar{q}) \,+\, \int_0^1 (q(x) - \bar{q}(x))\, u_{\bar{q}}^2(x)\, dx \,+\, O(n^{-4}). \qquad (2)$$

Here $u_{\bar{q}}$ is the normalized eigenfunction, $\int_0^1 u_{\bar{q}}^2(x)\,dx = 1$, corresponding to the smallest eigenvalue $\lambda(\bar{q})$.

Since we have complete information on the spline $\bar{q}$, we may approximate $\lambda(\bar{q})$ and $u_{\bar{q}}$ with arbitrarily small error. For $\lambda(\bar{q})$, we achieve an error of order ε as in the worst case setting, with cost proportional to $\varepsilon^{-1/2}\log\varepsilon^{-1}$. To obtain an approximation to $u_{\bar{q}}$, we apply one step of the inverse power algorithm with an appropriately chosen initial vector. In this way we obtain a vector, from which we compute $u_{\bar{q}}$ via piecewise interpolation. The total cost of computing $\lambda(\bar{q})$ and $u_{\bar{q}}$ is of order $\varepsilon^{-1/2}\log\varepsilon^{-1}$.

We then approximate the second term in (2) using the Monte Carlo algorithm for the function $(q(x)-\bar{q}(x))u_{\bar{q}}^2(x)$ computed at n randomized points with uniform distribution over $[0,1]$. This leads to an ε-approximation in the randomized setting with cost bounded from above by a quantity proportional to $\mathbf{c}\,\varepsilon^{-2/5}+\varepsilon^{-1/2}\log\varepsilon^{-1}$, where the first term bounds the information cost and the second term bounds the combinatorial cost of the algorithm. Hence, we have a sharp estimate on the randomized information complexity $n(\varepsilon)$. The ratio of the upper to lower bounds of the randomized complexity is roughly at most $\varepsilon^{-1/10}$.

In both classical settings, algorithms for which we obtain upper bounds on complexity require space of order $\varepsilon^{-1/2}$. This follows from the fact that we need to work on $n\times n$ tridiagonal matrices with n of order $\varepsilon^{-1/2}$.

We now turn to the quantum setting. Quantum algorithms are described in Section 4. Here we only mention that quantum algorithms work on $2^\nu\times 2^\nu$ unitary matrices, where ν is the number of qubits. The qubit complexity is defined as the minimal number of qubits needed to solve a problem. Roughly speaking, the qubit complexity corresponds to the space complexity for a classical computer. For the foreseeable future, qubits will be a scarce resource. That is why the qubit complexity is especially important, and computationally important problems with relatively small qubit complexity are of special interest.

We prove that the qubit complexity, $\mathrm{comp}^{\mathrm{qub}}(\varepsilon)$, of the Sturm-Liouville eigenvalue problem is of order $\log\varepsilon^{-1}$, which is relatively modest. In this paper log denotes $\log_2$. More precisely, we prove that

$$\tfrac{1}{2}\,\log\varepsilon^{-1}\,+\,\Omega(1)\,\le\,\mathrm{comp}^{\mathrm{qub}}(\varepsilon)\,\le\,\tfrac{3}{2}\,\log\varepsilon^{-1}\,+\,O(1).$$

These bounds hold regardless of the kind of queries used. Clearly, the qubit complexity yields a lower bound for the cost of any quantum algorithm solving this problem.

We now turn to the quantum setting with bit queries. We show that the bit query complexity is $\Theta(\varepsilon^{-1/3})$. This result is obtained by using:

- equation (2) relating the Sturm-Liouville eigenvalue problem to integration,
- a lower bound on bit queries for integration, and
- a modification of the classical randomized algorithm described above that uses a quantum summation algorithm instead of Monte Carlo to approximate the weighted integral in (2).

We now discuss the quantum setting with power queries. In this setting, the Sturm-Liouville eigenvalue problem can be solved using the well-known phase estimation algorithm as a basic tool, see, e.g., [22, Section 5.2]. This algorithm uses power queries and the quantum inverse Fourier transform as its main ingredients. The power queries have the form controlled-W^{2^j} for $j \in \mathbb{N}$, i.e., they use powers of the matrix $W = \exp\left(\frac{1}{2}\mathrm{i}\, M_q\right)$, with M_q an $n \times n$ real symmetric tridiagonal matrix whose diagonal elements depend on the values of q. The matrix M_q is a well-known discretization of the differential operator $\mathbb{L}_q$, and its size n depends on the necessary accuracy. To obtain an ε-approximation we use n of order $\varepsilon^{-1/2}$.

The phase estimation algorithm uses the exact eigenvector of M_q, equivalently of W, as part of its initial state, see [22, Section 5.2]. Abrams and Lloyd [1] analyzed the case when the exact eigenvector is replaced by an approximate eigenvector and concluded that as long as the approximation is *good enough*, the phase estimation algorithm will still supply a good approximation to the corresponding eigenvalue. Jaksch and Papageorgiou [18] proposed an efficient construction of an approximate eigenvector. Their idea was to solve the problem with low accuracy on a classical computer and obtain a "short" vector which approximates the eigenfunction u_q at few points. Then the amplitudes of this short vector are replicated on a quantum computer by the Hadamard transform, which yields a "long" (vector) state that can be used as the approximate initial state in the phase estimation algorithm.

We show how the construction of Jaksch and Papageorgiou can be used for the Sturm-Liouville eigenvalue problem. In this way, we compute an ε-approximation of the smallest eigenvalue with probability $\frac{3}{4}$ by the phase estimation algorithm using $\log \varepsilon^{-1} + O(1)$ power queries. The algorithm requires an additional number of quantum operations at most of order $\log^2 \varepsilon^{-1}$. This additional cost is for the quantum inverse Fourier transform. Finally, the number of qubits is $\frac{3}{2} \log \varepsilon^{-1} + O(1)$. A lower bound on the number of power queries of order $\log \varepsilon^{-1}$ has been proven in [5].

Comparing these quantum estimates to the classical complexity bounds in the worst case and randomized setting, we see that the quantum setting with power queries yields an exponential speedup between the number of power queries and the number of function values needed for the Sturm-Liouville eigenvalue problem.

Finally, we point out important consequences of our results, which we study in detail in [25]. Knowing that the Sturm-Liouville eigenvalue problem can be solved with polylog power queries, it is natural to study which computational problems can be reduced to this problem. In this respect, we think that the most important result of this paper is the formula that relates this eigenvalue problem to integration. In a particular case, this formula, see (10), states that

$$\lambda(q) \,=\, \pi^2 + \tfrac{1}{2} \,+\, 2\int_0^1 \left(q(x) - \tfrac{1}{2}\right) \sin^2(\pi x)\, dx \,+\, O\left(\|q - \tfrac{1}{2}\|_\infty^2\right). \tag{3}$$

Hence, the problem of computing the smallest eigenvalue is equivalent, modulo the second order term, to the weighted integration problem. Since $\lambda(q)$ can be approximated with polylog power queries, so can the weighted integral of q. It turns out that many computational problems can be formulated as an integration problem. Examples include important discrete problems such as Grover's search, the approximation of the Boolean mean, and NP-complete problems. The approximation of the Boolean mean is used as the primary tool to compute multivariate integrals and path integrals. Hence, all these problems can be solved by reducing them to the Sturm-Liouville eigenvalue problem with a polylog number of power queries in the quantum setting. It is well-known that Grover's search and the approximation of the Boolean mean require a number of bit queries polynomial in the problem size, which in our case is a polynomial in ε^{-1}. This shows that power queries are exponentially more powerful than bit queries, see [25] for details.

3 Problem Definition

We deal with functions from the class

$$\mathbf{Q} = \left\{ q : [0,1] \to [0,1] \;\middle|\; q \in C^2([0,1]) \text{ and } \|q\| := \max_{i=0,1,2} \max_{x\in[0,1]} |q^{(i)}(x)| \le 1 \right\}.$$

For a function $q \in \mathbf{Q}$, we consider the Sturm-Liouville eigenvalue problem $\mathbb{L}_q u = \lambda\, u$ for a non-zero u, or equivalently

$$u''(x) - q(x)u(x) + \lambda u(x) = 0, \quad \text{for } x \in (0,1), \tag{4}$$

with the boundary conditions

$$u(0) = u(1) = 0. \tag{5}$$

Let $\lambda = \lambda(q)$ be the smallest eigenvalue of (4), (5). Multiplying (4) by u and integrating by parts, see [2, 10, 28], we conclude that the smallest eigenvalue satisfies

$$\lambda(q) = \min_{0\neq u \in H_0^1} \frac{\int_0^1 \left[(u'(x))^2 + q(x)u^2(x)\right]\, dx}{\int_0^1 u^2(x)\, dx}, \tag{6}$$

where H_0^1 is the Sobolev space of absolutely continuous[1] functions for which $u' \in L_2([0,1])$ and $u(0) = u(1) = 0$.

Let u_q be a normalized real eigenfunction corresponding to the smallest eigenvalue. It is known that the eigenvalues of $\mathbb{L}_q$ are simple, and the eigenspace corresponding to $\lambda(q)$ is of dimension one. Therefore u_q is uniquely defined up to the sign. In particular, u_q^2 is uniquely defined. Then (6) states that

$$\lambda(q) = \int_0^1 \left(\left(u_q'(x)\right)^2 + q(x)u_q^2(x) \right) dx \qquad \text{and} \qquad \|u_q\|_{L_2} := \left(\int_0^1 u_q^2(x)\, dx \right)^{1/2} = 1. \tag{7}$$

Observe that $q \in \mathbf{Q}$ implies that $u_q \in C^4([0,1])$. Since $\|q\| \le 1$, and $\|u_q\|_{L_2} = 1$ with $u_q(0) = u_q(1) = 0$, then $|u_q^{(i)}(x)|$ are uniformly bounded for all $i \in [0,4]$, $x \in [0,1]$ and $q \in \mathbf{Q}$, see e.g., [10, p. 337].

The smallest eigenvalue $\lambda(q)$ is a non-decreasing function of q, i.e., $q_1(x) \le q_2(x)$ for $x \in [0,1]$ implies $\lambda(q_1) \le \lambda(q_2)$. It is known that for $q \equiv c$ we have

$$\lambda(c) = \pi^2 + c \qquad \text{and} \qquad u_c(x) = \sqrt{2}\, \sin(\pi x).$$

[1] A function f is absolutely continuous if and only if it can be written as $f(x) = f(0) + \int_0^x f'(t)dt$ for all $x \in [0,1]$.

This implies that for $q \in \mathbf{Q}$, we have $\lambda(q) \in [\lambda(0), \lambda(1)] = [\pi^2, \pi^2 + 1]$.

We will need estimates of the smallest eigenvalues and their eigenfunctions for perturbed functions q. This is a classical problem and many such estimates can be found in the literature, not only for the simplified Sturm-Liouville problem that we consider in this paper but also for more general eigenvalue problems. In our case, the problem of perturbed eigenvalues and eigenvectors is well-conditioned, since the differential operator $\mathbb{L}_q$ is symmetric and the eigenvalues of $\mathbb{L}_q$ are well separated. Combining results from [10, 19, 29] one can obtain the following estimates for $q, \bar{q} \in \mathbf{Q}$:

$$
\begin{aligned}
|\lambda(q) - \lambda(\bar{q})| &\leq \|q - \bar{q}\|_\infty := \max_{x \in [0,1]} |q(x) - \bar{q}(x)|, && (8)\\
\|u_q - u_{\bar{q}}\|_\infty &= O\left(\|q - \bar{q}\|_\infty\right), && (9)\\
\lambda(q) &= \lambda(\bar{q}) + \int_0^1 (q(x) - \bar{q}(x))\, u_{\bar{q}}^2(x)\, dx + O\left(\|q - \bar{q}\|_\infty^2\right). && (10)
\end{aligned}
$$

We stress that the factors in the big-O notation are independent of q and $\bar{q}$.

These relations follow by elementary arguments. Indeed, (8) follows from (6) by taking $u = u_{\bar{q}}$, which leads to $\lambda(q) - \lambda(\bar{q}) \leq \|q - \bar{q}\|_\infty$. By replacing the roles of q and $\bar{q}$ we get $\lambda(\bar{q}) - \lambda(q) \leq \|q - \bar{q}\|_\infty$, which implies (8). The next relation (9) can be also proved by a matrix approximation to the operator $\mathbb{L}_q$, which will be done in Section 4. Finally, (10) follows by again taking $u = u_{\bar{q}}$ in (6), which leads to

$$
\begin{aligned}
\lambda(q) &\leq \lambda(\bar{q}) + \int_0^1 (q(x) - \bar{q}(x))\; u_{\bar{q}}^2(x)\, dx \\
&= \lambda(\bar{q}) + \int_0^1 (q(x) - \bar{q}(x))\; u_q^2(x)\, dx + \int_0^1 (q(x) - \bar{q}(x))\; \left(u_{\bar{q}}^2(x) - u_q^2(x)\right)\, dx.
\end{aligned}
$$

By (9), the last term is of order $\|q - \bar{q}\|_\infty^2$. Taking $u = u_q$ in the expression (6) defining $\lambda(\bar{q})$, we obtain

$$
\lambda(\bar{q}) \leq \lambda(q) + \int_0^1 (\bar{q}(x) - q(x))\, u_q^2(x)\, dx.
$$

The last two inequalities imply (10). We shall see later that the formula (10) will be very useful in deriving lower bounds for classical algorithms. Note that if we take $\bar{q} \equiv \frac{1}{2}$, then the formula (10) becomes (3).

4 Classical Algorithms

In this section we consider classical algorithms, i.e., algorithms on a classical (non-quantum) computer. These algorithms can be either deterministic or randomized. They use information about the functions q from $\mathbf{Q}$ by computing $q(t_i)$ for some discretization points $t_i \in [0, 1]$. Here, $i = 1, 2, \ldots, n_q$, for some n_q, and the points t_i can be adaptively chosen, i.e., t_i can be a function

$$
t_i = t_i(t_1, q(t_1), \ldots, t_{i-1}, q(t_{i-1})),
$$

of the previously computed function values and points for $i \geq 2$. The number n_q can also be adaptively chosen, see, e.g., [30] for details.

A classical deterministic algorithm produces an approximation

$$\phi(q) \;=\; \phi(q(t_1), \ldots, q(t_{n_q}))$$

to the smallest eigenvalue $\lambda(q)$ based on finitely many values of q computed at deterministic points. Let $n = \sup_{q\in\mathbf{Q}} n_q$. We assume that $n < \infty$. The worst case error of such a deterministic algorithm ϕ is given by

$$e^{\rm wor}(\phi, n) = \sup_{q\in\mathbf{Q}} |\lambda(q) - \phi(q)|. \tag{11}$$

A classical randomized algorithm produces an approximation to $\lambda(q)$ based on finitely many values of q computed at random points, and is of the form

$$\phi_\omega(q) \;=\; \phi_\omega(q(t_{1,\omega}), \ldots, q(t_{n_{q,\omega},\omega})),$$

where $\phi_\omega, t_{i,\omega}$ and $n_{q,\omega}$ are random variables. We assume that the mappings

$$\begin{aligned}
\omega &\;\mapsto\; t_{i,\omega} \;=\; t_i(t_{1,\omega}, q(t_{1,\omega}), \ldots, t_{i-1,\omega}, q(t_{i-1,\omega})), \\
\omega &\;\mapsto\; \phi_\omega, \\
\omega &\;\mapsto\; n_{q,\omega}
\end{aligned}$$

are measurable. Let $n_q = \mathbb{E}(n_{q,\omega})$ be the expected number of values of the function q with respect to ω . As before, we assume that $n = \sup_{q\in\mathbf{Q}} n_q < \infty$. The randomized error of such a randomized algorithm ϕ is given by

$$e^{\rm ran}(\phi, n) = \sup_{q\in\mathbf{Q}} \left(\mathbb{E}[\lambda(q) - \phi_\omega(q)]^2\right)^{1/2}. \tag{12}$$

For simplicity and brevity we consider the error of randomized algorithms in the L_2 sense. It is straightforward to extend our results for the error of randomized algorithms defined in the L_p-sense with $p \in [1, \infty]$.

We denote the minimal number of function values needed to compute an ε-approximation of the Sturm-Liouville eigenvalue problem in the worst case and randomized settings by

$$\begin{aligned}
n^{\rm wor}(\varepsilon) &\;=\; \min\{\, n: \;\exists\, \phi \text{ such that } e^{\rm wor}(\phi, n) \;\le\; \varepsilon \,\} \;\text{ and} \\
n^{\rm ran}(\varepsilon) &\;=\; \min\{\, n: \;\exists\, \phi \text{ such that } e^{\rm ran}(\phi, n) \;\le\; \varepsilon \,\},
\end{aligned}$$

respectively.

4.1 Lower Bounds

We now prove lower bounds on $n^{\rm wor}(\varepsilon)$ and $n^{\rm ran}(\varepsilon)$.

Theorem 4.1.

$$n^{\rm wor}(\varepsilon) \;=\; \Omega\left(\varepsilon^{-1/2}\right), \qquad n^{\rm ran}(\varepsilon) \;=\; \Omega\left(\varepsilon^{-2/5}\right).$$

Proof. Define

$$F \;=\; \left\{\, f:\; f\in C^2([0,1]),\; \max\left(\|f\|_\infty, \|f'\|_\infty, \|f''\|_\infty\,\right) \;\le\; 1\,\right\}\right), \tag{13}$$

and consider the weighted integration problem

$$I(f) \;=\; \int_0^1 f(x)\sin^2(\pi x)\,dx \qquad \forall\, f\in F.$$

It is well-known that any algorithm using n function values for approximating of this weighted integration problem has worst case error at least proportional to n^{-2} in the worst case setting, and to $n^{-2.5}$ in the randomized setting, see [23, 30][2].

For $c \in (0, \frac{1}{2}]$, consider the class

$$F_c \;=\; F \cap \{\, f\in F \;:\; \|f\|_\infty \;\le\; c\,\}. \tag{14}$$

For n^{-2} much less than c, the proofs for the class F can be used to deduce the same lower bounds on algorithms for approximation of the weighted integration problem for the class F_c.

For $f \in F_c$ define $q = \frac{1}{2} + f$. Then $q \in \mathbf{Q}$. From (3) we have

$$\lambda(q) \;=\; \pi^2 \,+\, \tfrac{1}{2} \,+\, 2\,I(f) \,+\, O(c^2).$$

For any algorithm ϕ using n function values of q for the Sturm-Liouville eigenvalue problem, define the algorithm $\psi(f) = \frac{1}{2}(\phi(q) - \pi^2 - \frac{1}{2})$ for the weighted integration problem. Then ψ uses n function values of f, and

$$\lambda(q) - \phi(q) \;=\; 2\,(I(f) - \psi(f)) \,+\, O(c^2). \tag{15}$$

Let $c = n^{-3/2}$. Then $n^{-2} = o(c)$, and therefore the error of ϕ is lower bounded by $\Omega(n^{-2})$ in the worst case setting, and by $\Omega(n^{-2.5})$ in the randomized setting. Hence, the error of ϕ is at most ε when $n = \Omega(\varepsilon^{-1/2})$ in the worst case setting, and $n = \Omega(\varepsilon^{-2/5})$ in the randomized setting. Since this holds for an arbitrary algorithm ϕ, the proof is complete. □

4.2 Upper Bounds in the Worst Case Setting

We now discuss upper bounds on $n^{\rm wor}(\varepsilon)$, as well as bounds on the complexity in the worst case setting. The worst case cost of an algorithm ϕ using n function values is defined as

$$\mathrm{cost}^{\rm wor}(\phi) \;=\; \sup_{q\in\mathbf{Q}} \left(\mathbf{c}\, n_q + m_q\right),$$

where m_q is the number of arithmetic operations used by the algorithm for a function q from $\mathbf{Q}$. The worst case complexity $\mathrm{comp}^{\rm wor}(\varepsilon)$ is defined as the minimal cost of an algorithm whose worst case error is at most ε,

$$\mathrm{comp}^{\rm wor}(\varepsilon) \;=\; \min\left\{\, \mathrm{cost}^{\rm wor}(\phi) \;:\; \phi \text{ such that } e^{\rm wor}(\phi, n) \;\le\; \varepsilon\,\right\}.$$

[2] Formally, these results are proved for $I(f) = \int_0^1 f(x)\,dx$. However, the same proofs can be applied for the integration problem with the weight $\sin^2(\pi x)$ and the same lower bounds hold.

Obviously, $\mathrm{comp}^{\mathrm{wor}}(\varepsilon) \ge \mathbf{c}\, n^{\mathrm{wor}}(\varepsilon)$.

We now discuss the classical algorithm for the Sturm-Liouville eigenvalue problem, see e.g., [11, 19], and show that it is almost optimal in the worst case setting. This algorithm uses $n = \Theta(\varepsilon^{-1/2})$ function values of q at the equidistant points $i/(n+1)$ for $i = 1, 2, \dots, n$. Then the operator $\mathbb{L}_q$ is approximated by the tridiagonal $n \times n$ matrix M_q of the form

$$M_q = (n+1)^2 \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} + \begin{bmatrix} q(\frac{1}{n+1}) & & & & \\ & q(\frac{2}{n+1}) & & & \\ & & \ddots & & \\ & & & q(\frac{n-1}{n+1}) & \\ & & & & q(\frac{n}{n+1}) \end{bmatrix}.$$

Clearly, M_q is a symmetric and positive definite matrix. Let $\lambda_j = \lambda_j(M_q)$ and $z_j = z_j(M_q)$ be the eigenvalues and eigenvectors of M_q, i.e., $M_q z_j = \lambda_j z_j$ with

$$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n,$$

where the vectors z_j are orthogonal and normalized such that

$$\|z_j\|_{L_2}^2 := \frac{1}{n} \sum_{k=1}^{n} z_{j,k}^2 = 1$$

with $z_{j,k}$ being the kth component of z_j. Note that we use the subscript L_2 in the norm of a vector to stress similarity to the L_2 norm of functions, and to distinguish from the Euclidean second norm. Clearly, $\|z_j\|_{L_2} = \frac{1}{\sqrt{n}}\|z\|_2$.

For $q \equiv c$, it is known, see, e.g., [11], that

$$\lambda_j(M_c) = c + 4(n+1)^2 \sin^2\left(\frac{j\pi}{2(n+1)}\right),$$

and $z_j(M_c) = [z_{j,1}(M_c), z_{j,2}(M_c), \dots, z_{j,n}(M_c)]^T$ with

$$z_{j,k}(M_c) = \left(\frac{2n}{n+1}\right)^{1/2} \sin\left(\frac{jk\pi}{n+1}\right).$$

It is known, see, e.g., [19], that the smallest eigenvalue $\lambda_1(M_q)$ of the matrix M_q approximates the smallest eigenvalue $\lambda(q)$ of the operator $\mathbb{L}_q$ with error of order n^{-2}, i.e.,

$$\lambda(q) - \lambda_1(M_q) = O\left(n^{-2}\right) = O(\varepsilon).$$

Hence, it is enough to approximate $\lambda_1(M_q)$ with error of order ε. This can be achieved by using roughly $\log \varepsilon^{-1}$ bisection steps. Each step consists of computing the n terms of the Sturm sequence, and this can be done in cost proportional to n. The total cost is of order $(\mathbf{c} + \log \varepsilon^{-1})\varepsilon^{-1/2}$. For details, see [11, 33]. Theorem 4.1 and the cost of this algorithm lead to the following bounds for the minimal number of function values and for the worst case complexity.

Theorem 4.2.

$$n^{\text{wor}}(\varepsilon) \;=\; \Theta(\varepsilon^{-1/2}), \qquad \Omega(\mathbf{c}\,\varepsilon^{-1/2}) = \text{comp}^{\text{wor}}(\varepsilon) \;=\; O(\mathbf{c}\,\varepsilon^{-1/2} \;+\; \varepsilon^{-1/2}\log\,\varepsilon^{-1}).$$

Remark 4.1. We now show how (9) can be proven, based on the properties of the matrix M_q. First observe that for $q = 0$, the eigenvalues $\lambda_j(M_0)$ are well separated, since

$$\begin{aligned}\lambda_{j+1}(M_0) - \lambda_j(M_0) \;&=\; 4(n+1)^2 \sin\frac{(2j+1)\pi}{2(n+1)}\,\sin\frac{\pi}{2(n+1)}\\ &\geq\; 4(n+1)^2 \sin\frac{3\pi}{2(n+1)}\,\sin\frac{\pi}{2(n+1)} \;\approx\; 3\pi^2.\end{aligned}$$

For $q \in \mathbf{Q}$, the Hermitian matrix M_q differs from M_0 by the diagonal matrix $\operatorname{diag} q(i/(n+1))$ whose elements satisfy $q(i/(n+1)) \in [0, \|q\|_\infty]$ with $\|q\|_\infty \leq 1$. Using the known estimates on the perturbed eigenvalues of Hermitian matrices, see [33], we have

$$\min_{i=1,2,\ldots,n} |\lambda_j(M_q) - \lambda_i(M_0)| \;\leq\; \|q\|_\infty$$

for all $j = 1, 2, \ldots, n$. Since the intervals $[\lambda_i(M_0) - 1, \lambda_i(M_0) + 1]$ are disjoint, we conclude that

$$|\lambda_j(M_q) - \lambda_j(M_0)| \;\leq\; \|q\|_\infty \;\leq\; 1,$$

and that

$$\lambda_{j+1}(M_q) - \lambda_j(M_q) \;\geq\; \lambda_{j+1}(M_0) - \lambda_j(M_0) - 2 \;\approx\; 3\pi^2 - 2.$$

Define

$$\tilde{u}_{q,n} \;=\; \left[u_q\left(\frac{1}{n+1}\right), \ldots, u_q\left(\frac{n}{n+1}\right)\right]^T,$$

where u_q is the normalized real eigenfunction corresponding to the smallest eigenvalue. Then $\|\tilde{u}_{q,n}\|_{L_2} = 1 + o(1)$. We normalize $\tilde{u}_{q,n}$ and obtain

$$u_{q,n} \;=\; \frac{1}{\|\tilde{u}_{q,n}\|_{L_2}}\,\tilde{u}_{q,n}.$$

As mentioned in Section 3, the eigenfunction u_q is defined uniquely up to its sign. Obviously, the same is true for the eigenvector $z_1(M_q)$. We choose the signs of u_q and $z_1(M_q)$ such that

$$\|u_{q,n} - z_1(M_q)\|_{L_2} \;\leq\; \|u_{q,n} + z_1(M_q)\|_{L_2}.$$

All the components of the vector

$$\eta_n \;:=\; M_q u_{q,n} - \lambda(q) u_{q,n}$$

are of order n^{-2}, and therefore $\|\eta_n\|_{L_2} = O(n^{-2})$. From the a posteriori error estimate, see [33, p. 173], we conclude that

$$\|u_{q,n} - z_1(M_q)\|_{L_2} \;=\; O(n^{-2}) \qquad \forall\, q \in \mathbf{Q}$$

with the factor in the big-O notation independent of q. Note also that

$$M_q u_{\bar{q},n} - \lambda(q) u_{\bar{q},n} \;=\; M_{\bar{q}} u_{\bar{q},n} - \lambda(\bar{q}) u_{\bar{q},n} \;+ r_n,$$

with $\|r_n\|_{L_2} = O(\|q-\bar{q}\|_\infty)$. Hence

$$\|u_{\bar{q},n} - z_1(M_q)\|_{L_2} \;=\; O(\|q-\bar{q}\|_\infty + n^{-2}).$$

Finally, we have

$$\|u_{q,n} - u_{\bar{q},n}\|_{L_2} \;=\; \|u_{q,n} - z_1(M_q) + z_1(M_q) - u_{\bar{q},n}\|_{L_2} \;=\; O(n^{-2} + \|q-\bar{q}\|_\infty).$$

Letting n tend to infinity, we conclude that

$$\|u_q - u_{\bar{q}}\|_{L_2} \;= O(\|q-\bar{q}\|_\infty).$$

Since both u_q and $u_{\bar{q}}$ satisfy (4) for $(q,\lambda(q))$ and $(\bar{q},\lambda(\bar{q}))$, respectively, we have

$$u_q''(x) - u_{\bar{q}}''(x) \;=\; (q(x)-\lambda(q))(u_q(x)-u_{\bar{q}}(x)) \;+\; u_{\bar{q}}(x)\left((q(x)-\bar{q}(x)) - (\lambda(q)-\lambda(\bar{q}))\right).$$

Therefore

$$\|u_q'' - u_{\bar{q}}''\|_{L_2} \;=\; O(\|q-\bar{q}\|_\infty).$$

This and the fact that $u - u_{\bar{q}}$ vanishes at 0 and 1 imply

$$\|u_q - u_{\bar{q}}\|_\infty \;= O(\|q-\bar{q}\|_\infty),$$

as claimed. □

4.3 Upper Bounds in the Randomized Setting

We now turn to the randomized setting. The cost of a randomized algorithm ϕ, using $n = \sup_{q\in\mathbf{Q}} \mathbb{E}(n_{q,\omega}) < \infty$ randomized function values, is now defined as

$$\mathrm{cost}^{\mathrm{ran}}(\phi) \;=\; \sup_{q\in\mathbf{Q}} \left(\mathbb{E}\,(\mathbf{c}\, n_{q,\omega} + m_{q,\omega})^2\right)^{1/2},$$

where $m_{q,\omega}$ is the number of arithmetic operations used by the algorithm for a function q from $\mathbf{Q}$ and a random variable ω. The randomized complexity

$$\mathrm{comp}^{\mathrm{ran}}(\varepsilon) \;=\; \min\left\{\,\mathrm{cost}^{\mathrm{ran}}(\phi)\;:\;\phi\;\text{ such that }\; e^{\mathrm{ran}}(\phi,n)\;\le\;\varepsilon\,\right\},$$

is the minimal cost of an algorithm whose randomized error is at most ε. Obviously, $\mathrm{comp}^{\mathrm{ran}}(\varepsilon) \;\ge\; \mathbf{c}\, n^{\mathrm{ran}}(\varepsilon)$.

We now derive upper bounds on $n^{\mathrm{ran}}(\varepsilon)$ and $\mathrm{comp}^{\mathrm{ran}}(\varepsilon)$ by presenting a randomized algorithm that depends on a number of parameters. Then we find the values of these parameters for which the randomized error is ε. We first compute $m+1$ function values of q at deterministic points i/m, for $i = 0,1,\dots,m$, and construct a cubic natural spline q_{cub} interpolating q at these points, see e.g., [7] for information about cubic splines. It is well known that this can be done with cost proportional to m, and $\|q-q_{\rm cub}\|_\infty = O(m^{-2})$. The function $q_{\rm cub}$ does not have to be non-negative. Since $q \ge 0$ then $\bar{q} = q_{\rm cub} + c \ge 0$ with a constant $c = O(m^{-2})$. We have $\bar{q} \in \mathbf{Q}$ and $\|q - \bar{q}\|_\infty = O(m^{-2})$. We apply the formula (10) for the function $\bar{q}$ and obtain

$$\lambda(q) \,-\, \lambda(\bar{q}) \;=\; \int_0^1 (q(x) - \bar{q}(x))\, u_{\bar{q}}^2(x)\, dx \,+\, O\left(m^{-4}\right). \tag{16}$$

This suggests that we can improve the accuracy of approximating $\lambda(q) - \lambda(\bar{q})$ by using the classical Monte Carlo algorithm applied to the first term of the right hand side of (16). We will need to know, at least approximately, the eigenvalue $\lambda(\bar{q})$ and the eigenvector $u_{\bar{q}}$. Suppose we approximate $\lambda(\bar{q})$ by $\lambda_{\bar{q}}$ with the worst case error

$$\sup_{q \in \mathbf{Q}} |\lambda(\bar{q}) - \lambda_{\bar{q}}| \;\le\; \delta_1, \tag{17}$$

and the eigenfunction $u_{\bar{q}}$ by $z_{\bar{q}}$ with the worst case error

$$\sup_{q \in \mathbf{Q}} \|u_{\bar{q}} - z_{\bar{q}}\|_{L_2} \;\le\; \delta_2. \tag{18}$$

Assume for a moment that $\lambda_{\bar{q}}$ and $z_{\bar{q}}$ have been computed. For a function v, define $f_v(x) = (q(x) - \bar{q}(x))v^2(x)$ and $I(f_v) = \int_0^1 f_v(x)\, dx$.

The randomized algorithm ϕ based on the Monte Carlo with k randomized samples takes the form

$$\phi_\omega(q) \;=\; \lambda_{\bar{q}} \,+\, \frac{1}{k}\sum_{j=1}^{k} \Big(q(x_{j,\omega}) \,-\, \bar{q}(x_{j,\omega})\Big) z_{\bar{q}}^2(x_{j,\omega}),$$

where $x_{j,\omega}$ are independent and uniformly distributed numbers from $[0,1]$. Here ω represents a random element. We have

$$\begin{aligned} |\lambda(q) - \phi_\omega(q)| \;&\le\; |\lambda(\bar{q}) - \lambda_{\bar{q}}| \,+\, |I(f_{u_{\bar{q}}}) - I(f_{z_{\bar{q}}})| \\ &+\; \left| I(f_{z_{\bar{q}}}) - \frac{1}{k}\sum_{j=1}^{k} f_{z_{\bar{q}}}(x_{j,\omega}) \right| \,+\, O(m^{-4}). \end{aligned}$$

Clearly,

$$\big|I(f_{u_{\bar{q}}}) - I(f_{z_{\bar{q}}})\big| \;\le\; \int_0^1 |q(x) - \bar{q}(x)|\, \big|u_{\bar{q}}^2(x) - z_{\bar{q}}^2(x)\big|\, dx \;=\; O(m^{-2}\,\delta_2).$$

Since $\|f_{z_{\bar{q}}}\|_{L_2} = O(m^{-2})$, the well known formula for the randomized error of Monte Carlo yields that

$$\left(\mathbb{E}_\omega \left(I(f_{z_{\bar{q}}}) - \frac{1}{k}\sum_{j=1}^{k} f_{z_{\bar{q}}}(x_{j,\omega}) \right)^2 \right)^{1/2} \;=\; \frac{(I(f_{z_{\bar{q}}}^2) - I^2(f_{z_{\bar{q}}}))^{1/2}}{k^{1/2}} \;=\; O\left(m^{-2}k^{-1/2}\right).$$

We have obtained the bound

$$e^{\rm ran}(\phi, n) \;=\; O\left(\delta_1 + m^{-2}\delta_2 + m^{-2}k^{-1/2} + m^{-4}\right)$$

on the randomized error of ϕ. Hence, to guarantee error at most ε, it is enough to take

$$\delta_1 = \Theta(\varepsilon), \quad m = k = \Theta(\varepsilon^{-2/5}) \;\text{ and }\; \delta_2 = \Theta(\varepsilon^{1/5}).$$

We now explain how to achieve (17) and (18). To get $\lambda_{\bar{q}}$ approximating $\lambda(\bar{q})$ with error of order ε, we approximate the operator $\mathbb{L}_{\bar{q}}$ by the matrix $M_{\bar{q}}$ as in the worst case setting, now with $n = \Theta(\varepsilon^{-1/2})$. Then $\lambda(\bar{q}) - \lambda_1(M_{\bar{q}}) = O(n^{-2}) = O(\varepsilon)$, and we compute $\lambda_{\bar{q}}$ as an ε-approximation of $\lambda_1(M_{\bar{q}})$ as for the worst case setting. This can be done with cost of order $\varepsilon^{-1/2}$ function values of $\bar{q}$, and of order $\varepsilon^{-1/2}\log\varepsilon^{-1}$ arithmetic operations. Since the cost of computing one function value of $\bar{q}$ is of order 1, the total cost of computing $\lambda_{\bar{q}}$ is of order $\varepsilon^{-1/2}\log\varepsilon^{-1}$.

To get $z_{\bar{q}}$ approximating $u_{\bar{q}}$ with error of order $\varepsilon^{1/5}$ we proceed as follows. Consider the eigenvector $z_1(M_{\bar{q}})$ of the matrix $M_{\bar{q}}$, with n not yet specified. By Remark 4.1, we have

$$\|u_{\bar{q},n} - z_1(M_{\bar{q}})\|_{L_2} \;=\; O(n^{-2}). \tag{19}$$

We approximate the smallest eigenvalue $\lambda_1(M_{\bar{q}})$ by $\bar{\lambda}$, with error δ. This can be achieved with cost of order $n\log\delta^{-1}$. Without loss of generality we assume that $\bar{\lambda} \neq \lambda_1(M_{\bar{q}})$. Indeed, we can check this condition by computing $\det(M_{\bar{q}} - \bar{\lambda}I)$ and if this determinant is zero we perturb $\bar{\lambda}$ a little. Then the matrix

$$A \;=\; \left(M_{\bar{q}} - \bar{\lambda}I\right)^{-1}$$

is non-singular and its eigenvalues are $\beta_j = (\lambda_j(M_{\bar{q}}) - \bar{\lambda})^{-1}$. Note that $|\beta_1| \geq \delta^{-1}$ and $\beta_j = O(1)$ for $j \geq 2$. For the jth vector $e_j = [0,\ldots,0,1,0,\ldots,0]^T$ with 1 in the jth position, define

$$x_j \;=\; A\,e_j.$$

We can compute x_j with cost of order n by solving the tridiagonal linear system $(M_{\bar{q}} - \bar{\lambda}I)x_j = e_j$. Then we compute

$$\|x_{j_0}\|_2 \;=\; \max_{j=1,2,\ldots,n} \|x_j\|_2,$$

and

$$z \;=\; \|x_{j_0}\|_2^{-1}\,x_{j_0}.$$

Observe that the cost of computing z is of order n^2.

Since $\{n^{-1/2}z_j(M_{\bar{q}})\}_{j=1}^n$ is orthonormal, we have $\|x_j\|_2^2 = \sum_{\ell=1}^n \beta_\ell^2(e_j, n^{-1/2}\,z_\ell(M_{\bar{q}}))^2$ and $\|n^{-1/2}\,z_1(M_{\bar{q}})\|_2^2 = 1 = \sum_{j=1}^n (e_j, n^{-1/2}\,z_1(M_{\bar{q}}))^2$. Hence, there exists an index j such that

$$(e_j, n^{-1/2}\,z_1(M_{\bar{q}}))^2 \;\geq\; n^{-1},$$

and therefore

$$\|x_{j_0}\|_2 \;\geq\; \|x_j\|_2 \;\geq\; \delta^{-1}n^{-1/2}.$$

We have

$$(M_{\bar{q}} - \lambda_1(M_{\bar{q}})I)z \;=\; (M_{\bar{q}} - \bar{\lambda}I)z \,+\, (\bar{\lambda} - \lambda_1(M_{\bar{q}}))z \;=\; \frac{1}{\|x_{j_0}\|_2}e_{j_0} \,+\, (\bar{\lambda} - \lambda_1(M_{\bar{q}}))z,$$

and therefore

$$\|(M_{\bar q} - \lambda_1(M_{\bar q})I)z\|_2 \;\le\; \delta\sqrt{n} + \delta.$$

From [33, p. 173], we conclude that $\|n^{-1/2}\, z_1(M_{\bar q}) - z\|_2 \;=\; O(\delta\,\sqrt{n})$, and

$$\|z_1(M_{\bar q}) - \sqrt{n}\, z\|_{L_2} \;=\; O(\delta\sqrt{n}). \tag{20}$$

We are finally ready to define $z_{\bar q}$ by piecewise linear interpolation from the successive components of the vector $\sqrt{n}\, z \;=\; [z_1, z_2, \dots, z_n]^T$. More precisely, for $j \;=\; 0, 1, \dots, n$ let $t_j = j/(n+1)$. For $t \in [t_j, t_{j+1}]$, we set

$$z_{\bar q}(t) \;=\; z_j(1 - (n+1)t + j) \;+\; z_{j+1}((n+1)t - j)$$

with $z_0 = z_{n+1} = 0$.

We need to estimate $u_{\bar q} - z_{\bar q}$ in the L_2 norm. Observe that for $t \in [t_j, t_{j+1}]$ we have

$$u_{\bar q}(t) \;=\; u_{\bar q}(t_j)(1 - (n+1)t + j) \;+\; u_{\bar q}(t_{j+1})((n+1)t - j) \;+\; O(n^{-2})$$

since $u_{\bar q} \in \mathbf{Q}$. Therefore

$$|u_{\bar q}(t) - z_{\bar q}(t)| \;\le\; |u_{\bar q}(t_j) - z_{\bar q}(t_j)| \;+\; |u_{\bar q}(t_{j+1}) - z_{\bar q}(t_{j+1})| \;+\; O(n^{-2}).$$

This yields

$$\begin{aligned}
\|u_{\bar q} - z_{\bar q}\|_{L_2}^2 \;&=\; \sum_{j=0}^{n} \int_{t_j}^{t_{j+1}} (u_{\bar q}(t) - z_{\bar q}(t))^2\, dt \\
&=\; O\left(\frac{1}{n+1} \sum_{j=0}^{n} (u_{\bar q}(t_j) - z_{\bar q}(t_j))^2 \;+\; n^{-4} \right).
\end{aligned}$$

Hence,

$$\|u_{\bar q} - z_{\bar q}\|_{L_2} \;=\; O\left(\|u_{\bar q,n} - \sqrt{n}\, z\|_{L_2} \;+ n^{-2}\right)$$

Since $\|u_{\bar q,n} - \sqrt{n}\, z\|_{L_2} \;\le\; \|u_{\bar q,n} - z_1(M_{\bar q})\|_{L_2} \;+\; \|z_1(M_{\bar q}) - \sqrt{n}\, z\|_{L_2}$, we use (19) and (20) to see that

$$\|u_{\bar q} - z_{\bar q}\|_{L_2} \;=\; O(\delta\sqrt{n} + n^{-2}).$$

For $\delta = n^{-5/2}$ we obtain

$$\|u_{\bar q} - z_{\bar q}\|_{L_2} \;=\; O(n^{-2}).$$

Setting $n = \Theta(\varepsilon^{-1/10})$ we obtain (18) with $\delta_2 = \Theta(\varepsilon^{1/5})$. The cost of computing $z_{\bar q}$ is of order $n^2 = \Theta(\varepsilon^{-1/5})$.

Theorem 4.1 and the cost of this randomized algorithm lead to the following bounds on the minimal number of function values and the randomized complexity.

Theorem 4.3.

$$n^{\mathrm{ran}}(\varepsilon) \;=\; \Theta(\varepsilon^{-2/5}), \qquad \Omega(\mathbf{c}\,\varepsilon^{-2/5}) = \mathrm{comp}^{\mathrm{ran}}(\varepsilon) \;=\; O(\mathbf{c}\,\varepsilon^{-2/5} \;+\; \varepsilon^{-1/2}\log \varepsilon^{-1}).$$

5 Quantum Setting

We now turn our attention to the quantum setting. In this setting, we are using *hybrid* algorithms that are combinations of classical algorithms using function values, as explained in the previous sections, and quantum algorithms which we now describe. A quantum algorithm applies a sequence of unitary transformations to an initial state, and the final state is measured, see [3, 8, 14, 22] for the details of the quantum model of computation. We briefly summarize this model to the extent necessary for this paper.

The initial state $|\psi_0\rangle$ 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. Obviously, the dimension of $\mathcal{H}_\nu$ is 2^ν. The number ν denotes the number of qubits used in quantum computation.

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

$$|\psi\rangle := U_T Q_Y U_{T-1} Q_Y \cdots U_1 Q_Y U_0 |\psi_0\rangle. \tag{21}$$

Here, $U_0, U_1, \ldots, U_T$ are unitary matrices that do not depend on the input function q. The unitary matrix Q_Y with $Y = [q(t_1), \ldots, q(t_n)]$ is called a quantum query and depends on n, with $n \leq 2^\nu$, function evaluations of q computed at some non-adaptive points $t_i \in [0, 1]$. The quantum query Q_Y is the only source of information about q. The integer T denotes the number of quantum queries we choose to use.

At the end of the quantum algorithm, a measurement is applied to its final state $|\psi\rangle$. The measurement produces one of M outcomes, where $M \leq 2^\nu$. Outcome $j \in \{0, 1, \ldots, M-1\}$ occurs with probability $p_Y(j)$, which depends on j and the input Y. For example, if $M = 2^\nu$ and the final state is $|\psi\rangle = \sum_{j=0}^{2^\nu-1} c_j |j\rangle$, with $\sum_{j=0}^{2^\nu-1} |c_j|^2 = 1$, then a measurement in the computational orthonormal basis $\{|j\rangle\}$ produces the outcome j with probability $p_Y(j) = |c_j|^2$. Knowing the outcome j, we compute an approximation $\hat{\lambda}_Y(j)$ of the smallest eigenvalue on a classical computer.

In principle, quantum algorithms may have many measurements applied between sequences of unitary transformations of the form presented above. However, any algorithm with many measurements and a total of T quantum queries can be simulated by a quantum algorithm with only one measurement at the end, for details see e.g., [14].

We stress that classical algorithms in floating or fixed point arithmetic can also be written in the form of (21). Indeed, all classical bit operations can be simulated by quantum computations, see e.g., [4]. Classically computed function values will correspond to bit queries which we discuss in Section 5.2.

In our case, we formally use the real number model of computation. Since the Sturm-Liouville eigenvalue problem is well conditioned and properly normalized, we obtain practically the same results in floating or fixed point arithmetic. More precisely, it is enough to use $O(\log \varepsilon^{-1})$ mantissa bits, and the cost of bit operations in floating or fixed point arithmetic is of the same order as the cost in the real number model multiplied by a power of $\log \varepsilon^{-1}$.

Hence, a hybrid algorithm may be viewed as a finite sequence of algorithms of the form (21). It is also known that if we use finitely many algorithms of the form (21) then they can be written as one quantum algorithm of the form (21), see [14, 15].

That is why an arbitrary hybrid algorithm in the quantum setting is of the form (21). This is important when we want to prove lower bounds because it is enough to work with algorithms of the form (21). For upper bounds, it seems to us more natural to distinguish between classical and quantum computations and charge their cost differently. The cost of classical computations is defined as before whereas the cost of quantum computations is defined as the sum of the number of quantum queries multiplied by the cost of one query, and the number of quantum operations besides quantum queries. It will be also important to indicate how many qubits are used by the quantum computations.

We now define the error in the quantum setting. In this setting, we want to approximate the smallest eigenvalue $\lambda(q)$ with a probability $p > \frac{1}{2}$. For simplicity, we take $p = \frac{3}{4}$ for the rest of this section. As it is common for quantum algorithms, we can achieve an ε-approximation with probability arbitrarily close to 1 by repetition of the original quantum algorithm, and by taking the median as the final approximation.

The local error of the quantum algorithm with T queries that computes $\hat{\lambda}_Y(j)$ for the function $q \in \mathbf{Q}$ and the outcome $j \in \{0, 1, \dots, M-1\}$ is defined by

$$e(\hat{\lambda}_Y, T) \;=\; \min\left\{ \alpha : \sum_{j:\; |\lambda(q) - \hat{\lambda}_Y(j)| \le \alpha} p_Y(j) \ge \tfrac{3}{4} \right\}.$$

This can be equivalently rewritten as

$$e(\hat{\lambda}_Y, T) \;=\; \min_{A:\, \mu(A) \ge \frac{3}{4}} \max_{j \in A} \left| \lambda(q) - \hat{\lambda}_Y(j) \right|,$$

where $A \subset \{0, 1, \dots, M-1\}$ and $\mu(A) = \sum_{j \in A} p_Y(j)$.

The *worst probabilistic* error of a quantum algorithm $\hat{\lambda}$ with T queries for the Sturm-Liouville eigenvalue problem is defined by

$$e^{\text{quant}}(\hat{\lambda}, T) \;=\; \sup\left\{ e(\hat{\lambda}_Y, T) \colon \; Y = [q(t_1), \dots, q(t_n)], \quad t_i \in [0,1], \quad \text{for} \;\; q \in \mathbf{Q} \right\}. \tag{22}$$

5.1 Bit Queries

Quantum queries are important in the complexity analysis of quantum algorithms. A quantum query corresponds to a function evaluation in classical computation. By analogy with the complexity analysis of classical algorithms, we analyze the cost of quantum algorithms in terms of the number of quantum queries that are necessary to compute an ε-approximation with probability $\frac{3}{4}$. Clearly, this number is a lower bound on the quantum complexity, which is defined as the minimal total cost of a quantum algorithm that solves the problem.

Different quantum queries have been studied in the literature. Probably the most commonly studied query is the *bit* query. For a Boolean function $f : \{0, 1, \dots, 2^m - 1\} \to \{0, 1\}$, the bit query is defined by

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

Here $\nu = m + 1$, $|j\rangle \in \mathcal{H}_m$, and $|k\rangle \in \mathcal{H}_1$ with $\oplus$ denoting the addition modulo 2. For real functions, such as functions q, the bit query is constructed by taking the most significant bits of the function q evaluated at some points t_j. More precisely, as in [14], the bit query for q has the form

$$Q_q|j\rangle|k\rangle \;=\; |j\rangle|k \oplus \beta(q(\tau(j)))\rangle,$$

where the number of qubits is now $\nu = m'+m''$ and $|j\rangle \in \mathcal{H}_{m'}$, $|k\rangle \in \mathcal{H}_{m''}$ with some functions $\beta : [0,1] \to \{0,1,\ldots,2^{m''}-1\}$ and $\tau : \{0,1,\ldots,2^{m'}-1\} \to [0,1]$. Hence, we compute q at $t_j = \tau(j) \in [0,1]$ and then take the m'' most significant bits of $q(t_j)$ by $\beta(q(t_j))$, for details and a possible use of ancilla qubits see again [14].

Using bit queries, the well known quantum algorithm of Grover [13] requires $\Theta(N^{1/2})$ queries for searching an unordered database of N items. Similarly, the quantum summation algorithm of Brassard et al. [6] computes the mean of a Boolean function defined on the set of N elements with accuracy ε and probability $\frac{3}{4}$ using of order $\min\{N, \varepsilon^{-1}\}$ bit queries. Both algorithms are optimal modulo multiplicative factors in terms of the number of bit queries.

The quantum summation algorithm can be also used for the approximate computation of the mean of a real function $f : [0,1] \to \mathbb{R}$ with $|f(x)| \le M$ for all $x \in [0,1]$, see [14, 23]. More precisely, if we want to approximate

$$\mathrm{S}_N(f) \;:=\; \frac{1}{N}\sum_{j=0}^{N-1} f(x_j)$$

for some $x_j \in [0,1]$ and N, then the quantum summation algorithm $\mathrm{QS}_N(f)$ approximates $\mathrm{S}_N(f)$ such that

$$|\mathrm{S}_N(f) - \mathrm{QS}_N(f)| \;\le\; \varepsilon \qquad \text{with probability } \tfrac{3}{4} \tag{23}$$

using of order $\min(N, M\varepsilon^{-1})$ bit queries, $\min(N, M\varepsilon^{-1}) \log N$ quantum operations, and $\log N$ qubits.

Bit queries have been also used for a number of continuous problems such as multivariate and path integration, multivariate approximation, and ordinary differential equations. Tight bit query complexity bounds are known for a number of such problems, see [14, 15, 16, 17, 20, 24, 32].

In particular, Novak [24] proved that for the integration problem $\int_0^1 f(x)\,dx$ for functions f from the class F given by (13), the bit query complexity is

$$n^{\mathrm{bit-query}}(\varepsilon, \mathrm{INT}_F) \;=\; \Theta(\varepsilon^{-1/3}). \tag{24}$$

Here and elsewhere by the bit query complexity we understand the minimal number of bit queries needed to compute an ε-approximation to a given problem with probability $\frac{3}{4}$. In particular, $n^{\text{bit-query}}(\varepsilon)$ denotes the bit query complexity of the Sturm-Liouville eigenvalue problem.

Based on the result (24) of Novak and the relationship between the Sturm-Liouville eigenvalue problem with integration, we now prove the following theorem.

Theorem 5.1.

$$n^{\text{bit-query}}(\varepsilon) \;=\; \Omega(\varepsilon^{-1/3}).$$

Proof. We first prove that the bit query complexity for the weighted integration problem for the class F_c given by (14) is of the same order as for integration for the class F,

$$n^{\text{bit-query}}(\varepsilon, \mathrm{INT}_{F_c}) = \Theta(\varepsilon^{-1/3}). \tag{25}$$

The upper bound follows from (24). To prove the lower bound, we use the standard proof technique of reducing the integration problem to the mean Boolean summation problem for which a lower bound on bit queries is known.

Assume then that we use an arbitrary quantum algorithm with k bit queries that computes an ε-approximation with probability $\frac{3}{4}$ for the integration problem over the class F_c. Without loss of generality we assume that $k^{-2} \le c$.

Consider the function $h(x) = \alpha x^3(1-x)^3$ for $x \in [0,1]$ and $h(x) = 0$ for $x > 1$. Here, α is a positive number chosen such that $h \in F$ with F given by (13). For $j = 0, 1, \ldots, N-1$, with $N > k$, define $h_j(x) = N^{-2}h(N(x - j/N))$. Clearly, $h_j \in F$ and the support of h_j is $(j/N, (j+1)/N$. Observe that $\|h_j\|_\infty \le N^{-2}$. Hence $h_j \in F_c$. We also have $\int_0^1 h_j(x)\,dx = N^{-3}\int_0^1 h(x)\,dx$. For an arbitrary Boolean function $B : \{0, 1, \ldots, N-1\} \to \{0, 1\}$, define the function

$$f_B(x) = \sum_{j=0}^{N-1} B(j)h_j(x) \quad \forall\, x \in [0, 1].$$

Then $f_B \in F_c$ and

$$\int_0^1 f_B(x)\,dx = \frac{\int_0^1 h(x)\,dx}{N^2}\,\frac{1}{N}\sum_{j=0}^{N-1} B(j).$$

Hence, modulo the factor of order N^{-2}, the computation of the Boolean mean is reduced to the integration problem. Note that $f_B(t) = B(j)h_j(t)$ if $t \in [j/N, (j+1)/N]$, and sampling of f_B is equivalent to sampling of B. From [21] we know that $\Omega(k^{-1})$ is a lower bound for the error of the quantum approximation of the Boolean mean, with k bit queries, and probability $\frac{3}{4}$, where $N \ge \beta k$ for some positive β. Letting $N = \lceil \beta k \rceil$, we conclude that the corresponding lower bound on the integration problem over the class F_c is $\Omega(k^{-3})$. Hence to achieve the error ε we must have $k = \Omega(\varepsilon^{-1/3})$, as claimed in (25).

The same proof techniques allows us to consider the classes $F_{c(\varepsilon)}$ with varying $c(\varepsilon)$, even with $c(\varepsilon)$ tending to zero, although not too fast. We have

$$n^{\text{bit-query}}(\varepsilon, \mathrm{INT}_{F_{c(\varepsilon)}}) = \Theta(\varepsilon^{-1/3}) \qquad \text{if } \lim_{\varepsilon\to 0} c(\varepsilon)\,\varepsilon^{-2/3} = \infty. \tag{26}$$

We now turn to the Sturm-Liouville eigenvalue problem. As in the proof of Theorem 4.1, for $f \in F_c$ with $c \in (0, \frac{1}{2}]$, we define $q = \frac{1}{2} + f$ and consider an arbitrary quantum algorithm ϕ that uses k quantum bit queries and computes an ε-approximation of the smallest eigenvalue with probability $\frac{3}{4}$. Then $\psi(f) = \frac{1}{2}(\phi(q) - \pi^2 - \frac{1}{2})$ is a quantum algorithm for approximating the integration problem over the class F_c. We have

$$|I(f) - \psi(f)| = \left|\tfrac{1}{2}\left(\lambda(q) - \phi(q)\right) + O(c^2)\right| \le \tfrac{1}{2}\,\varepsilon + O(c^2).$$

Take now $c = c(\varepsilon) = \Theta(\varepsilon^{2/3-\delta})$ with $\delta \in (0, \frac{1}{6})$. Then

$$|I(f) - \psi(f)| \le \tfrac{1}{2}\,\varepsilon + O(\varepsilon^{4/3-2\delta}) = \tfrac{1}{2}\,\varepsilon(1 + o(1)) \le \varepsilon \quad \text{for small } \varepsilon.$$

Hence, the quantum error of ψ with probability $\frac{3}{4}$ is ε, and ψ uses k bit queries. Due to (26), we have $k = \Omega(\varepsilon^{-1/3})$ which completes the proof. $\square$

We now derive upper bounds on the bit query complexity $n^{\text{bit-query}}(\varepsilon)$ and on the total quantum complexity $\text{comp}^{\text{bit-quant}}(\varepsilon)$. The total quantum complexity is defined as the minimal cost of a hybrid algorithm that solves the Sturm-Liouville eigenvalue problem with error at most ε and probability $\frac{3}{4}$. The hybrid algorithm may require some classical computations and the use of function values and the cost of them is defined just as before. It may also require some quantum computations and the cost of them is defined as the sum of the number of bit queries multiplied by the cost of one such query plus the number of additional quantum operations. The cost of one bit query is denoted by $\mathbf{c}_{\text{bit}}$.

We present a hybrid algorithm, which will be a combination of the classical algorithm from Section 4 and the quantum summation algorithm QS_N for a properly chosen N. We proceed as in Section 4 and use the same notation. From (16), (17), and (18), we have

$$\lambda(q) \,=\, \lambda_{\bar{q}} + \int_0^1 (q(x) - \bar{q}(x)) z_{\bar{q}}(x)\, dx \,+\, O(\delta_1 + m^{-2}\delta_2 + m^{-4}) \tag{27}$$

with δ_1, δ_2 and m to be specified later. Let

$$f(x) \,=\, (q(x) - \bar{q}(x)) z_{\bar{q}}(x) \qquad x \in [0,1].$$

Observe that $f(x) = O(m^{-2})$, and $f(x)$ depends on $q(x)$, and $q(i/m)$ for $i = 0, 1, \dots, m$, which are used in the construction of $\bar{q}$. Furthermore, we can compute $f(x)$ by computing one function value $q(x)$ and one function value of the already computed functions $\bar{q}$ and $z_{\bar{q}}$ at x. We approximate $\int_0^1 f(x)\, dx$ by

$$\text{S}_N(f) \,=\, \frac{1}{N} \sum_{j=0}^{N-1} f\left(\frac{j+1}{N}\right)$$

with $N = (m+1)k$, where the parameters m and k will be specified later. Since f is twice continuously differentiable and $f''(x)$ is uniformly bounded on the subintervals $(i/m, (i+1)/m)$ for $i = 0, 1, \dots, m-1$, it is easy to see that

$$\int_0^1 f(x)\, dx \,-\, \text{S}_N(f) \,=\, O\left(\frac{1}{N^2}\right).$$

We define N such that N^{-2} is of order ε.

We now apply $\text{QS}_N(f)$ algorithm to compute an $\Theta(\varepsilon)$-approximation with probability $\frac{3}{4}$ to $\text{S}_N(f)$, or, equivalently to $\int_0^1 f(x)\, dx$. To do it, we need to use the bit query Q_f for the function f, although so far we assumed that we can use only bit queries Q_q for the functions q from $\mathbf{Q}$. This problem is resolved in Section 2 of [15] where it is shown that algorithms using the bit query Q_f can be simulated by algorithms using bit queries Q_q at the expense of multiplying the number of bit queries by a factor of 2.

From this and (23) with $M = O(m^{-2})$, we conclude that its is enough to perform of order $\min(\varepsilon^{-1/2}, m^{-2}\varepsilon^{-1})$ bit queries, $\min(\varepsilon^{-1/2}, m^{-2}\varepsilon^{-1}) \log \varepsilon^{-1}$ quantum operations, and using of order $\log \varepsilon^{-1/2}$ qubits.

We finally approximate $\lambda(q)$ by the following algorithm

$$\phi(q) \;=\; \lambda_{\bar{q}} \,+\, \mathrm{QS}_N(f). \tag{28}$$

This algorithm differs from the randomized algorithm of Section 4 since we now apply the QS_N quantum algorithm instead of Monte Carlo to approximate $\int_0^1 f(x)\,dx$. Its error is clearly of the form

$$e^{\text{bit-quant}}(\phi, T) \;=\; O\left(\delta_1 + m^{-2}\delta_2 + m^{-4} + \varepsilon\right).$$

To guarantee that this error is at most ε, we take

$$\delta_1 = \Theta(\varepsilon), \quad m = \Theta(\varepsilon^{-1/3}), \quad k = \Theta(\varepsilon^{-1/6}) \ \text{ and } \ \delta_2 = \Theta(\varepsilon^{1/3}).$$

Using the cost analysis of Section 4 and the results of this section, we conclude the following theorem.

Theorem 5.2. *The Sturm-Liouville eigenvalue problem can be solved in the quantum setting with bit queries by the algorithm ϕ defined by (28). This algorithm approximates the smallest eigenvalue $\lambda(q)$ with error at most ε and probability $\frac{3}{4}$ using of order*

- $\varepsilon^{-1/3}$ *bit queries and function values,*
- $\varepsilon^{-1/3}\log\varepsilon^{-1}$ *quantum operations,*
- $\varepsilon^{-1/2}\log\varepsilon^{-1}$ *classical operations,*
- $\log\varepsilon^{-1}$ *qubits.*

Furthermore,

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

and

$$\Omega(\mathbf{c}_{\text{bit}}\,\varepsilon^{-1/3}) \;=\; \mathrm{comp}^{\text{bit-query}}(\varepsilon) \;=\; O\left((\mathbf{c} + \mathbf{c}_{\text{bit}})\,\varepsilon^{-1/3} \,+\, \varepsilon^{-1/2}\log\varepsilon^{-1}\right).$$

Hence, we have a sharp bound of order $\varepsilon^{-1/3}$ on the number of bit queries whereas the upper bound on the total cost depends, as in the worst case and randomized settings, on $\varepsilon^{-1/2}\log\varepsilon^{-1}$, which is the cost of classical computations.

5.2 Power Queries

In this subsection we study *power* queries. We formally define them as follows. For some problems, a quantum algorithm can be written in the form

$$|\psi\rangle \;:=\; U_m \widetilde{W}_m U_{m-1} \widetilde{W}_{m-1} \cdots U_1 \widetilde{W}_1 U_0 |\psi_0\rangle. \tag{29}$$

Here $U_1, \ldots, U_m$ denote unitary matrices independent of the function q, just as before, whereas the unitary matrices $\widetilde{W}_j$ are of the form controlled-W_j, see [22, p. 178]. That is, $W_j = W^{p_j}$ for an $n \times n$ unitary matrix W that depends on the input of the computational problem, and for some non-negative integers p_j, $j = 1, 2, \ldots, m$. Without loss of generality we assume that n is a power of two. Let $\{|y_k\rangle\}$ be orthonormalized eigenvectors of W, $W|y_k\rangle = \alpha_k|y_k\rangle$ with the corresponding eigenvalue α_k, where $|\alpha_k| = 1$ and $\alpha_k = e^{\mathrm{i}\lambda_k}$ with $\lambda_k \in [0, 2\pi)$ for $k = 1, 2, \dots, n$. For the unit vectors $|x_\ell\rangle = \alpha_\ell|0\rangle + \beta_\ell|1\rangle \in \mathbb{C}^2$, $\ell = 1, 2, \dots, r$, the quantum query $\widetilde{W}_j$ is defined as

$$\widetilde{W}_j\,|x_1\rangle|x_2\rangle\cdots|x_r\rangle|y_k\rangle \;=\; |x_1\rangle|\cdots|x_{j-1}\rangle\bigg(\alpha_j|0\rangle + \beta_j e^{\mathrm{i}\gamma p_j\lambda_k}|1\rangle\bigg)|x_{j+1}\rangle\cdots|x_r\rangle|y_k\rangle. \tag{30}$$

Hence, $\widetilde{W}_j$ is a $2^\nu \times 2^\nu$ unitary matrix with $\nu = r + \log n$. We stress that the exponent p_j only affects the power of the complex number $e^{\mathrm{i}\gamma\lambda_k}$.

We call $\widetilde{W}_j$ a *power* query since they are derived from powers of W. Power queries have been successfully used for a number of problems, see again [22], including the phase estimation problem that will be discussed in the next section. The phase estimation algorithm, see [8, 22], is at the core of many quantum algorithms. It plays a central role in the fast quantum algorithms for factoring and discrete logarithms of Shor [27]. We stress that for Shor's algorithm, power queries can be implemented by a number of elementary quantum gates that is polylog in n. The phase estimation algorithm approximates an eigenvalue of a unitary operator W using the corresponding eigenvector, or its approximation, as part of the initial state. The powers of W are defined by $p_i = 2^{i-1}$. Therefore, phase estimation uses queries with $W_1 = W$, $W_2 = W^2$, $W_3 = W^{2^2}$, $\dots$, $W_m = W^{2^{m-1}}$. It is typically assumed, see [8], that we do not explicitly know W but we are given quantum devices that perform controlled-W, controlled-W^2, controlled-W^{2^2}, and so on.

For the Sturm-Liouville eigenvalue problem, as well as for problems studied in [25], we will use the matrix

$$W \;=\; \exp\left(\mathrm{i}\gamma M_q\right) \quad \text{with } \mathrm{i} = \sqrt{-1} \text{ and a positive } \gamma, \tag{31}$$

where the $n \times n$ matrix M_q was introduced in Section 3.2 as a discretization of the differential operator $\mathbb{L}_q$. The matrix W is unitary since M_q is symmetric.

For the $\widetilde{W}_j$ with the matrix W of (31) we modify the query definition in equation (21) and assume, as in [22, Ch. 5], that for each j the $\widetilde{W}_j$ is one quantum query. Accordingly, for algorithms that can be expressed in the form (29), the number of power queries is m, independently of the powers p_j. By analogy with (22), we denote their error by $e^{\mathrm{quant}}(\hat{\lambda}, m)$.

Allowing quantum algorithms of the form (29) with power queries, we define the power query complexity $n^{\mathrm{power-query}}(\varepsilon)$ to be the minimal number of power queries required to approximate the Sturm-Liouville eigenvalue problem with error ε, i.e.,

$$n^{\mathrm{power-query}}(\varepsilon) = \min\{m : \ \exists\ \hat{\lambda} \ \text{ such that } \ e^{\mathrm{quant}}(\hat{\lambda}, m) \le \varepsilon\}.$$

The cost of one power query is denoted by $\mathbf{c}_{\mathrm{power}}$. The total complexity, $\mathrm{comp}^{\mathrm{power-query}}(\varepsilon)$, is the defined as the minimal cost of a hybrid algorithm in the same way as for bit queries.

We will use the phase estimation algorithm as a basic module for approximating the smallest eigenvalue $\lambda(q)$. As shown by Abrams and Lloyd [1], the phase estimation algorithms can also be used if a good approximation of the eigenvector corresponding to the smallest eigenvalue is known. Such an approximation is obtained by the algorithm of Jaksch and

Papageorgiou [18]. Combining these algorithms, we obtain the quantum algorithm that computes the smallest eigenvalue with error ε and probability $\frac{3}{4}$ using $\Theta(\log \varepsilon^{-1})$ power queries, and $\Theta(\log \varepsilon^{-1})$ qubits.

For the sake of completeness, we review the phase estimation problem and algorithm, the results of Abrams and Lloyd and the results of Jaksch and Papageorgiou in the next subsections.

5.3 Phase Estimation

Consider W defined by (31) with $\gamma = \frac{1}{2}$, i.e.,

$$W = \exp\left(\frac{1}{2}\,\mathrm{i}\,M_q\right).$$

The eigenvalues of W are $e^{\mathrm{i}\lambda_j(M_q)/2}$ with $\lambda_j(M_q)$ being the eigenvalues of the $n \times n$ matrix M_q and n is assumed to be a power of two. These eigenvalues can be written as $e^{2\pi \mathrm{i}\varphi_j}$, where

$$\varphi_j = \varphi_j(M_q) = \frac{1}{4\pi}\,\lambda_j(M_q)$$

are called *phases*. We are interested in estimating the smallest phase $\varphi_1(M_q)$, which belongs to $(0,1)$ since $\lambda_1(M_q) \in [\pi^2, \pi^2+1]$. For convenience, we renumber and normalize the eigenvectors of M_q, and also of W, as

$$|y_j\rangle = \sqrt{n}\,|z_{j+1}(M_q)\rangle,$$

for $j = 0, 1, \ldots, n-1$. We will use $\{|y_j\rangle\}$ as the orthonormal basis of the space.

Phase estimation, see [22, Section 5.2], is a quantum algorithm that approximates the phase $\varphi_1(M_q)$. Note that to compute an ε-approximation of $\lambda_1(M_q)$, it is enough to compute an $\varepsilon/(4\pi)$-approximation of $\varphi_1(M_q)$. The original phase estimation algorithm has been derived for the initial state $|0^{\otimes m}\rangle|y_0\rangle$, where m is related to the accuracy and will be determined later, and $|y_0\rangle = |y_0(M_q)\rangle$ is the eigenvector of the matrix M_q corresponding to the smallest eigenvalue $\lambda_1(M_q)$. Abrams and Lloyd [1] showed that phase estimation can still be used if the eigenvector $|y_0\rangle$ is replaced by a *good* approximation $|\psi_0\rangle$ as the initial state.

More precisely, expanding $|\psi_0\rangle$ in the basis of the eigenvectors $|y_j\rangle$, the initial state takes the form

$$|0\rangle^{\otimes m}|\psi_0\rangle = |0\rangle^{\otimes m}\sum_{j=0}^{n-1} d_j|y_j\rangle.$$

Using m Hadamard gates, we place the first register in an equal superposition, which gives the state

$$|\psi_1\rangle = \frac{1}{\sqrt{2^m}}\sum_{x_1=0}^{1}\sum_{x_2=0}^{1}\cdots\sum_{x_m=0}^{1}|x_1\rangle|x_2\rangle\cdots|x_m\rangle\sum_{j=0}^{n-1} d_j|y_j\rangle.$$

We now apply the controlled quantum gates, see (30), to create the state

$$\begin{aligned}|\psi_2\rangle &= \widetilde{W}_{2^{m-1}}\widetilde{W}_{2^{m-2}}\cdots\widetilde{W}_{2^0}\,|\psi_1\rangle \\ &= \frac{1}{\sqrt{2^m}}\sum_{j=0}^{n-1} d_j|\eta_j\rangle|y_j\rangle\end{aligned}$$

with

$$\begin{aligned}
|\eta_j\rangle &= \Big(|0\rangle + e^{2\pi \mathrm{i}\varphi_j}|1\rangle\Big) \otimes \Big(|0\rangle + e^{2\pi \mathrm{i} 2\varphi_j}|1\rangle\Big) \otimes \cdots \otimes \Big(|0\rangle + e^{2\pi \mathrm{i} 2^{m-1}\varphi_j}|1\rangle\Big) \\
&= \sum_{x_1=0}^{1}\sum_{x_2=0}^{1}\cdots\sum_{x_m=0}^{1} e^{2\pi \mathrm{i}(x_1 2^0 + x_2 2^1 + \cdots x_m 2^{m-1})\varphi_j}|x_1\rangle|x_2\rangle\cdots|x_m\rangle \\
&= \sum_{\ell=0}^{2^m-1} e^{2\pi\,\mathrm{i}\,\ell\,\varphi_j}|\ell\rangle,
\end{aligned}$$

see also [22, p. 222]. Hence,

$$|\psi_2\rangle \;=\; \frac{1}{\sqrt{2^m}}\sum_{j=0}^{n-1} d_j \left(\sum_{\ell=0}^{2^m-1} e^{2\pi\,\mathrm{i}\,\ell\,\varphi_j}|\ell\rangle\right)|y_j\rangle.$$

The inverse Fourier transform performed on the first register creates the state

$$\sum_{j=0}^{n-1} d_j \left(\sum_{\ell=0}^{2^m-1} g(\varphi_j,\ell)|\ell\rangle\right)|y_j\rangle,$$

where

$$g(\varphi_j,\ell) = \begin{cases} \frac{\sin(\pi(2^m\varphi_j-\ell))e^{\pi \mathrm{i}(\varphi_j-\ell 2^{-m})(2^m-1)}}{2^m\sin(\pi(\varphi_j-\ell 2^{-m}))} & \text{if } \;\varphi_j \neq 2^{-m}\ell, \\ 1, & \text{if } \;\varphi_j = 2^{-m}\ell. \end{cases}$$

A measurement of the first register produces the outcome j with probability

$$p_j = \sum_{\ell=0}^{n-1} |d_\ell|^2 |g(\varphi_\ell, j)|^2,$$

and the second register collapses to the state

$$\sum_{\ell=0}^{n-1}\frac{d_\ell g(\varphi_\ell,j)}{\sqrt{p_j}}|y_\ell\rangle.$$

The quantity

$$\Delta(\phi_0,\phi_1) \;=\; \min_{x\in\mathbb{Z}}\{|x+\phi_1-\phi_0|\} \qquad \text{for } \;\phi_0,\phi_1\in\mathbb{R}$$

is defined in [6] and is the fractional part of the distance between two phases ϕ_0 and ϕ_1. It is used to derive the relationship between the approximation error and the success probability. A measurement of the first register produces an outcome from the set

$$\mathcal{G}_k \;=\; \{j : \Delta(j/2^b, \varphi_1(M_q)) \le k/2^m\,\},$$

where $k > 1$, with probability

$$\Pr(\mathcal{G}_k) \;=\; \sum_{j\in\mathcal{G}_k}\sum_{\ell=0}^{n-1}|d_\ell g(\varphi_\ell,j)|^2 \;\ge\; |d|^2\sum_{j\in\mathcal{G}_k}|g(\varphi_1(M_q),j)|^2 \;\ge\; |d|^2 - \frac{|d|^2}{2(k-1)},$$

where $d = \langle y_0|\psi_0\rangle$. For $k = 1$ the probability that

$$\Delta(j/2^m, \varphi_1(M_q)) \le 2^{-m} \quad \text{is bounded from below by } \frac{8}{\pi^2}|d|^2. \tag{32}$$

The proof of the probability bounds can be found in [6, 22]. Using this fact, the authors of [1] conclude that as long as $|d|^2$ is *large enough* or, equivalently, $|\psi_0\rangle$ is *close enough* to $|y_0\rangle$ then phase estimation can be used to approximate the phase $\varphi_1(M_q)$ with probability close to $8/\pi^2 = 0.81\ldots$.

We stress that the phase estimation algorithm uses m power queries. In addition to the cost of the queries there is a quantum operations cost proportional to at most m^2, which is an upper bound on the cost of the quantum inverse Fourier transform, see [22, Section 5.2].

5.4 Eigenvalue and Eigenvector Approximation

The results of Jaksch and Papageorgiou [18] can be applied to efficiently construct a good approximate eigenvector when $W = e^{\frac{\mathrm{i}}{2}M_q}$ as in the previous subsection.

The matrix $M_q = M_q^{(n)}$ has been derived from the discretization of the operator $\mathbb{L}_q$ with mesh size $h_n = (n+1)^{-1}$. Its eigenvectors are also eigenvectors of $W = W^{(n)}$, and we denote them here by $|y_j^{(n)}\rangle$, where $j = 0, 1, \ldots, n-1$. We want to approximate $\lambda_1(M_q^{(n)}) = 4\pi\varphi_1(M_q^{(n)})$ but we do not know the corresponding eigenvector

$$|y^{(n)}\rangle := |y_0^{(n)}\rangle.$$

The expansion of $|y^{(n)}\rangle$ in the computational basis is denoted by

$$|y^{(n)}\rangle = \sum_{j=0}^{n-1} y_j^{(n)}|j\rangle, \tag{33}$$

Recall that u_q is the normalized, $\|u_q\|_{L_2} = \left(\int_0^1 u^2(x)\,dx\right)^{1/2} = 1$, eigenfunction of the differential operator $\mathbb{L}_q$ that corresponds to $\lambda(q)$, and u_q as well as u_q' and u_q'' are uniformly bounded, i.e., $\|u_q\|_\infty$, $\|u_q'\|_\infty$ and $\|u_q''\|_\infty$ are $O(1)$.

Let $|U^{(n)}\rangle = \sum_{j=0}^{n-1} u_q((j+1)h_n)|j\rangle$ be the vector obtained by sampling u_q at the discretization points. Then it is known, see [12, 19] as well as Remark 4.1, that

$$\begin{aligned}\left\| |y^{(n)}\rangle - \frac{|U^{(n)}\rangle}{\||U^{(n)}\rangle\|_2} \right\|_2 &= O(h_n^2) \quad \text{and} \\ |\lambda(q) - \lambda_1(M_q^{(n)})| &= O(h_n^2).\end{aligned} \tag{34}$$

Consider a coarse discretization of $\mathbb{L}_q$ with mesh size $h_{n_0} = (n_0+1)^{-1}$ with n_0 being a power of two. Assume that

$$|\tilde{z}^{(n_0)}\rangle = \sum_{j=0}^{n_0-1} \tilde{z}_j^{(n_0)}|j\rangle, \quad \||\tilde{z}^{(n_0)}\rangle\|_2 = 1,$$

approximates the eigenvector $|y^{(n_0)}\rangle$ that corresponds to the smallest eigenvalue of the matrix $M_q^{(n_0)}$ such that,

$$\| \, |\tilde{z}^{(n_0)}\rangle - |y^{(n_0)}\rangle \, \|_2 \;=\; O(n_0^{-2}). \tag{35}$$

We place the vector $|\tilde{z}^{(n_0)}\rangle$ in a $\log n_0$ qubit register. As explained in Section 4.3, we can compute $|\tilde{z}^{(n_0)}\rangle$ on a classical computer with cost of order n_0^2.

For $n = 2^s n_0$, we construct an approximation $|\tilde{z}^{(n)}\rangle$ of $|y^{(n)}\rangle$ by first appending s qubits, all in the state $|0\rangle$, to $|\tilde{z}^{(n_0)}\rangle$ and then performing the Hadamard transformation on each one of these s qubits, i.e.,

$$|\tilde{z}^{(n)}\rangle = |\tilde{z}^{(n_0)}\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right)^{\otimes s} = \frac{1}{\sqrt{2^s}} \sum_{j=0}^{n-1} \tilde{z}_{g(j)}^{(n_0)} \, |j\rangle, \tag{36}$$

where $\tilde{z}_{g(j)}^{(n_0)}$'s denote the coordinates of $|\tilde{z}^{(n_0)}\rangle$ in the computational basis, and $g(j) = \lfloor j/2^s \rfloor$. The effect of g is to replicate 2^s times the coordinates of $|\tilde{z}^{(n_0)}\rangle$. As in Jaksch and Papageorgiou [18], we use the vector $|\tilde{z}^{(n)}\rangle$ as part of the input to the phase estimation algorithm.

Let $d^{(n)} = \langle y^{(n)} | \tilde{z}^{(n)} \rangle$. We show that $|d^{(n)}|^2$ can be made arbitrarily close to one by choosing a sufficiently large n_0. Hence, we can make the success probability of the phase estimation algorithm at least equal to $\frac{3}{4}$.

Consider two different expansions of $|\tilde{z}^{(n)}\rangle$,

$$|\tilde{z}^{(n)}\rangle \;=\; \sum_{j=0}^{n-1} \tilde{u}_j^{(n)} |j\rangle \tag{37}$$

$$|\tilde{z}^{(n)}\rangle \;=\; \sum_{j=0}^{n-1} d_j^{(n)} |y_j^{(n)}\rangle. \tag{38}$$

The first expansion is in the computational basis $\{|j\rangle\}$ and, by (36),

$$\tilde{u}_j^{(n)} = 2^{-s/2} z_{g(j)}^{(n_0)} \qquad \text{for } j = 0, 1, \ldots, n-1,$$

while the second expansion is with respect to the eigenvectors of $M_q^{(n)}$. Note that $d^{(n)} = d_0^{(n)}$ and clearly $\sum_{j=0}^{n-1} |d_j^{(n)}|^2 = 1$. Equation (38) implies

$$|\tilde{z}^{(n)}\rangle - |y^{(n)}\rangle = (d^{(n)} - 1)|y^{(n)}\rangle + \sum_{j=1}^{n-1} d_j^{(n)} |y_j^{(n)}\rangle. \tag{39}$$

Taking norms on both sides we obtain

$$\big|\big| \, |y^{(n)}\rangle - |\tilde{z}^{(n)}\rangle \, \big|\big|_2^2 \;=\; |d^{(n)} - 1|^2 + \sum_{j=1}^{n-1} |d_j^{(n)}|^2 \;\geq\; \sum_{j=1}^{n-1} |d_j^{(n)}|^2 \;=\; 1 - |d^{(n)}|^2. \tag{40}$$

We now bound the left hand side of (40) from above. Using the expression (33) for $|y^{(n)}\rangle$ and the definition of $|\tilde{z}^{(n)}\rangle$, see (36), (37), we have

$$\begin{aligned}\big\||y^{(n)}\rangle - |\tilde{z}^{(n)}\rangle\big\|_2^2 &= \sum_{j=0}^{n-1} |y_j^{(n)} - 2^{-s/2} z_{g(j)}^{(n_0)}|^2 \\ &= \sum_{j=0}^{n-1} \left| \frac{u_q((j+1)h_n)}{\||U^{(n)}\rangle\|_2} - \frac{u_q((g(j)+1)h_{n_0})}{\sqrt{2^s}\||U^{(n_0)}\rangle\|_2} + \Delta_j^{(n)} - \frac{\Delta_{g(j)}^{(n_0)}}{\sqrt{2^s}} \right|^2,\end{aligned}$$

where, by (34) and (35), we have

$$\sum_{j=0}^{n-1} |\Delta_j^{(n)}|^2 = O(h_n^4) \qquad \text{and} \qquad \sum_{j=0}^{n-1} |\Delta_{g(j)}^{(n_0)}|^2 = 2^s O(h_{n_0}^4).$$

Applying the triangle inequality, we get

$$\big\||y^{(n)}\rangle - |\tilde{z}^{(n)}\rangle\big\|_2 \le \left(\sum_{j=0}^{n-1} \left| \frac{u_q((j+1)h_n)}{\||U^{(n)}\rangle\|_2} - \frac{u_q((g(j)+1)h_{n_0})}{\sqrt{2^s}\||U^{(n_0)}\rangle\|_2} \right|^2 \right)^{1/2} + O(h_{n_0}^2). \tag{41}$$

The definition of $|U^{(n)}\rangle$ and the fact that the derivative of u_q is Lipschitz[3] with the uniform Lipschitz constant imply that $\||U^{(n)}\rangle\|_2 = \sqrt{n}(1 + O(h_n))$. Hence, the square of the term in the parentheses above is equal to

$$\frac{1}{n} \sum_{j=0}^{n-1} |u_q((j+1)h_n)(1 + O(h_n)) - u_q((g(j)+1)h_{n_0})(1 + O(h_{n_0}))|^2. \tag{42}$$

Since u_q is continuous with a bounded first derivative, we have that

$$u_q(x_{2,j}) = u_q(x_{1,j}) + O(|x_{2,j} - x_{1,j}|), \tag{43}$$

where $x_{2,j} = (j+1)h_n$ and $x_{1,j} = (g(j)+1)h_{n_0}$, $j = 0, 1, \ldots, n-1$. Let $\lfloor j/2^s \rfloor = j/2^s - \alpha$ with $\alpha \in (0,1)$. Then

$$\begin{aligned}|x_{2,j} - x_{1,j}| &= \left| \frac{j+1}{2^s n_0 + 1} - \frac{j/2^s + 1 - \alpha}{n_0 + 1} \right| \\ &= j\, \frac{2^s - 1}{(2^s n_0 + 1) 2^s (n_0 + 1)} + O(h_{n_0}) \;=\; O(h_{n_0}).\end{aligned}$$

Using (42), (43) and the triangle inequality, we obtain from (41) that

$$\big\||y^{(n)}\rangle - |\tilde{z}^{(n)}\rangle\big\|_2 \;=\; O(h_{n_0}) \;\le\; \frac{c}{n_0 + 1}$$

for some positive number c independent of n and n_0. Combining this with (40) we finally conclude that

$$|d^{(n)}|^2 \;\ge\; 1 \;-\; \frac{c^2}{(n_0 + 1)^2} \tag{44}$$

and $d^{(n)}$ can be made arbitrarily close to one by taking a sufficiently large n_0.

[3] A function $f : [0,1] \to \mathbb{R}$ is Lipschitz if there is a number $L \ge 0$ such that $|f(x) - f(y)| \le L|x - y|$ for all $x, y \in [0,1]$.

5.5 Quantum Algorithm for the Smallest Eigenvalue

We combine the results of the previous two subsections to derive a quantum algorithm for computing an ε-approximation of the smallest eigenvalue with probability $\frac{3}{4}$.

We choose the parameters for the phase estimation algorithm. Without loss of generality we assume that ε^{-1} is an even power of 2, that is $\varepsilon^{-1} = 2^m$ with an even m. We set $n = \varepsilon^{-1/2} = 2^{m/2}$ and we will be working with the matrix $M_q^{(n)}$. The index $n_0 = 2^{k_0}$ is chosen as the smallest power of two for which

$$\frac{8}{\pi^2}\left(1 - \frac{c^2}{(n_0+1)^2}\right) \;\geq\; \frac{3}{4}, \tag{45}$$

where the number c is from (44). Clearly, $n_0 = O(1)$. Without loss of generality we assume that $\frac{1}{2}m > k_0 = \log n_0$, i.e., we assume that ε is sufficiently small. We finally set $s = \frac{1}{2}m - k_0$. We then compute $|\tilde{z}^{(n_0)}\rangle$ on a classical computer as in Section 4.3 with cost $O(1)$ function values and operations.

We run the phase estimation algorithm for the matrix $W = e^{\frac{\mathrm{i}}{2}M_q^{(n)}}$ with the initial state, see (36),

$$|0\rangle^{\otimes m}|\tilde{z}^{(n)}\rangle \;=\; |0\rangle^{\otimes m}|\tilde{z}^{(n_0)}\rangle\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes s}.$$

Let j be the outcome of the phase estimation algorithm. We finally compute

$$\bar{\lambda}_j \;=\; 4\pi\, j\, 2^{-m}$$

as an approximation of the smallest eigenvalue $\lambda(q)$. We have

$$\begin{aligned}
\bar{\lambda}_j \;-\; \lambda(q) \;&=\; \bar{\lambda}_j \;-\; \lambda_1(M_q^{(n)}) \;+\; \lambda_1(M_q^{(n)}) \;-\; \lambda(q) \\
&=\; 4\pi\left(\frac{j}{2^m} - \varphi_1(M_q^{(n)})\right) \;+\; O(\varepsilon).
\end{aligned}$$

From (32) we know that

$$\left(\frac{j}{2^m} - \varphi_1(M_q^{(n)})\right) \;\leq\; \varepsilon \quad \text{with probability } \frac{8}{\pi^2}\,|d^{(n)}|^2.$$

By (44) and the definition of n_0 we have

$$\frac{8}{\pi^2}|d^{(n)}|^2 \;\geq\; \frac{8}{\pi^2}\left(1 - \frac{c^2}{(n_0+1)^2}\right) \;\geq\; \frac{3}{4}.$$

Hence,

$$|\bar{\lambda}_j \;-\; \lambda(q)| \;=\; O(\varepsilon) \quad \text{with probability at least } \frac{3}{4}.$$

The computation of $\bar{\lambda}_j$ requires

$$m + k_0 + s \;=\; \tfrac{3}{2}\,m \;=\; \tfrac{3}{2}\,\log\varepsilon^{-1}$$

qubits, $m = \log \varepsilon^{-1}$ power queries, plus a number of quantum operations proportional to $m^2 = \log^2 \varepsilon^{-1}$. This yields $n^{\text{power-query}}(\varepsilon) = O(\log \varepsilon^{-1})$. A lower bound on $n^{\text{power-query}}(\varepsilon)$ of the same order is proved in [5]. Hence,

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

We summarize the results of this section in the following theorem.

Theorem 5.3. *The Sturm-Liouville eigenvalue problem can be solved in the quantum setting with power queries by the phase estimation algorithm applied to the discretized matrix of the differential operator $\mathbb{L}_q$ with the initial state given as an approximate eigenvector computed by the Jaksch and Papageorgiou algorithm. This quantum algorithm approximates the smallest eigenvalue $\lambda(q)$ with error ε and probability $\frac{3}{4}$ using*

- $\frac{3}{2} \log \varepsilon^{-1} + O(1)$ *power queries,*
- $O(1)$ *function values and classical operations,*
- $O(\log^2 \varepsilon^{-1})$ *quantum operations besides the power queries, and*
- $\frac{3}{2} \log \varepsilon^{-1} + O(1)$ *qubits.*

Furthermore,

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

and

$$\Omega(\mathbf{c}_{\text{power}} \log \varepsilon^{-1}) \;=\; \text{comp}^{\text{power-query}}(\varepsilon) \;=\; O\left(\mathbf{c}_{\text{power}} \log \varepsilon^{-1} \;+\; \mathbf{c} + \log^2 \varepsilon^{-1}\right).$$

5.6 Qubit Complexity

In this section we address the qubit complexity, $\text{comp}^{\text{qub}}(\varepsilon)$, which is defined as the minimal number of qubits required to approximate the smallest eigenvalue with error ε and probability $\frac{3}{4}$ by quantum algorithms of the form (21). Clearly, $\text{comp}^{\text{qub}}(\varepsilon)$ is upper bounded by $\frac{3}{2} \log \varepsilon^{-1} + O(1)$ since that many qubits are used by the phase estimation algorithm of Section 5.5. Observe that the cost of the classical algorithm computing $|\tilde{z}^{(n_0)}\rangle$ as well as its quantum simulation [22, p. 189-193] is constant since n_0 is bounded by a constant due to (45).

We turn to a lower bound on $\text{comp}^{\text{qub}}(\varepsilon)$. Based on the results obtained in this paper, it is easy to see that the number of qubits necessary to solve our problem must be proportional at least to roughly $\frac{1}{2} \log \varepsilon^{-1}$. Indeed, assume that there is a quantum algorithm of the form (21) that computes $\lambda(q)$ with error ε and probability $\frac{3}{4}$, and uses $k(\varepsilon)$ qubits. This algorithm can use arbitrary quantum queries, assuming that each quantum query is based on at most $2^{k(\varepsilon)}$ function evaluations of q. Note that this holds for bit queries, as well as for the power queries studied in this paper. Then such an algorithm can be simulated by a classical algorithm that uses at most $2^{k(\varepsilon)}$ function evaluations of q. From Theorem 3.2 we know that $2^{k(\varepsilon)} = \Omega(\varepsilon^{-1/2})$ and therefore $k(\varepsilon) \geq \frac{1}{2} \log \varepsilon^{-1} + \Omega(1)$. Hence, the qubit complexity is lower bounded by $\frac{1}{2} \log \varepsilon^{-1} + \Omega(1)$. This proves the following theorem.

Theorem 5.4. *The qubit complexity of the Sturm-Liouville eigenvalue problem in the quantum setting with bit or power queries is bounded by*

$$\tfrac{1}{2}\log\varepsilon^{-1} + O(1) \le \mathrm{comp}^{\mathrm{qub}}(\varepsilon) \le \tfrac{3}{2}\log\varepsilon^{-1} + O(1).$$

Acknowledgments

This research has been supported in part by the National Science Foundation, the Defense Advanced Research Projects Agency (DARPA), and the Air Force Research Laboratory.

We are grateful for valuable comments from Stefan Heinrich, Marek Kwas, Joseph F. Traub and Arthur G. Werschulz.

References

[1] Abrams, D. S. and Lloyd, S. (1999), *Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors,* Phys. Rev. Lett., 83, 5162–5165.

[2] 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.

[3] Beals, R., Buhrman, H., Cleve, M., Mosca, R. and de Wolf, R. (1998), *Quantum lower bounds by polynomials*, Proceedings FOCS'98, 352–361. Also http://arXiv.org/quant-ph/9802049.

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

[5] Bessen, A. J., *A lower bound for eigenvalue estimation on a quantum computer*, in preparation.

[6] 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.

[7] Cheney, W. and Kincaid, D. (2004) Numerical Mathematics and Computing, 5th edition, Brooks/Cole, Pacific Grove, Ca.

[8] Cleve, R., Ekert, A., Macchiavello, C. and Mosca, M. (1996), *Quantum Algorithms Revisited,* Phil. Trans. R. Soc. Lond. A.

[9] Collatz, L. (1960), The Numerical Treatment of Differential Equations, Springer-Verlag, Berlin.

[10] Courant, C. and Hilbert, D. (1989), Methods of Mathematical Physics, Vol. I, Wiley Classics Library, Willey-Interscience, New York.

[11] Demmel, J. W. (1997), Applied Numerical Linear Algebra, SIAM, Philadelphia.

[12] Gary, H. (1965), *Computing Eigenvalues of Ordinary Differential Equations with Finite Differences*, Mathematics of Computation, Vol. 19, No. 91, 365–379.

[13] 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.

[14] Heinrich, S. (2002), *Quantum Summation with an Application to Integration*, J. Complexity, 18(1), 1–50. Also http://arXiv.org/quant-ph/0105116.

[15] Heinrich, S. (2003), *Quantum integration in Sobolev spaces*, J. Complexity, 19, 19–42.

[16] Heinrich, S. (2004), *Quantum Approximation I. Embeddings of Finite Dimensional L_p Spaces,* J. Complexity, 20, 5–26. Also http://arXiv.org/quant-ph/0305030.

[17] Heinrich, S. (2004), *Quantum Approximation II. Sobolev Embeddings,* J. Complexity, 20, 27–45. Also http://arXiv.org/quant-ph/0305031.

[18] 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.

[19] Keller, H. B. (1968), Numerical methods for two-point boundary-value problems, Blaisdell Pub. Co., Waltham, Massachusetts.

[20] Kacewicz, B. Z. (2004), *Randomized and quantum solution of initial value problems*, to appear in J. Complexity.

[21] A. Nayak and F. Wu (1999), *The quantum query complexity of approximating the median and related statistics*, STOC 99, 384–393.Also http://arXiv.org/quant-ph/9804066.

[22] Nielsen, M.A. and Chuang, I.L. (2000), Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, UK.

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

[24] Novak, E. (2001), *Quantum complexity of integration,* J. Complexity, 17, 2–16. Also http://arXiv.org/quant-ph/0008124.

[25] Papageorgiou, A. and Woźniakowski, H., The Sturm-Liouville eigenvalue problem and NP-complete problems in the quantum setting with queries, in preparation.

[26] Simon, D. R. (1997), *On the power of quantum computation,* SIAM J. Comput., 26, 1474–1483.

[27] Shor, P. W. (1997), *Polynomial-time algorithms for prime factorization and discrete logarithm on a quantum computer*, SIAM J. Comput., 26(5), 1484–1509.

[28] Strang, G. and Fix, G. J. (1973), An Analysis of the Finite Element Method, Prentice-Hall, Englewood Cliffs, NJ.

[29] Titchmarsh, E. C. (1958), Eigenfunction Expansions Associated with Second-Order Differential Equations, Part 2, Oxford University Press, Oxford, UK.

[30] Traub, J. F., Wasilkowski, G. W. and Woźniakowski, H. (1988), Information-Based Complexity, Academic Press, New York.

[31] Traub, J. F., and Werschulz, A. G. (1998), Complexity and Information, Cambridge University Press, Cambridge, UK.

[32] 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.

[33] Wilkinson, J. H. (1965), The Algebraic Eigenvalue Problem, Oxford University Press, Oxford, UK.

[34] Zalka, C. (1998), *Simulating quantum systems on a quantum computer*, Proc. R. Soc. Lond. A, 454, 313–322.