

# On the complexity of the multivariate Sturm-Liouville eigenvalue problem

A. Papageorgiou

Department of Computer Science, Columbia University, New York, USA

June 13, 2007

## Abstract

We study the complexity of approximating the smallest eigenvalue of  $-\Delta + q$  with Dirichlet boundary conditions on the  $d$ -dimensional unit cube. Here  $\Delta$  is the Laplacian, and the function  $q$  is nonnegative and has continuous first order partial derivatives. We consider deterministic and randomized classical algorithms, as well as quantum algorithms using quantum queries of two types: bit queries and power queries. We seek algorithms that solve the problem with accuracy  $\varepsilon$ . We exhibit lower and upper bounds for the problem complexity. The upper bounds follow from the cost of particular algorithms. The classical deterministic algorithm is optimal. Optimality is understood modulo constant factors that depend on  $d$ . The randomized algorithm uses an optimal number of function evaluations of  $q$  when  $d \leq 2$ . The classical algorithms have cost exponential in  $d$  since they need to solve an eigenvalue problem involving a matrix with size exponential in  $d$ . We show that the cost of quantum algorithms is not exponential in  $d$ , regardless of the type of queries they use. Power queries enjoy a clear advantage over bit queries and lead to an optimal complexity algorithm.

## 1 Introduction

In a recent paper with H. Woźniakowski [21] we studied the classical and quantum complexity of the Sturm-Liouville eigenvalue problem. This paper extends those results to the multidimensional case. By analogy with the Sturm-Liouville eigenvalue problem [11] in one dimension, we consider the eigenvalue problem  $-\Delta u + qu = \lambda u$  defined on the  $d$  dimensional unit cube with Dirichlet boundary condition. Here  $\Delta$  is the  $d$ -dimensional Laplacian, and  $q$  is a non-negative function of  $d$  variables whose first order partial derivatives exist and are continuous. Then we study the complexity of approximating the smallest eigenvalue  $\lambda(q)$  with accuracy  $\varepsilon$ .

We assume that  $q$  is not explicitly known but we can sample it at any point of the unit cube. Any algorithm solving this problem will need to compute a number of evaluations of  $q$  and to combine them to obtain an approximation of the eigenvalue of interest.

Classical algorithms may be deterministic or randomized. The former evaluate  $q$  at deterministically chosen points, while the latter can sample  $q$  at randomly chosen points.

Moreover, randomized algorithms may also combine the evaluations of  $q$  randomly. We obtain the worst case error of classical deterministic algorithms, and the worst expected error of randomized algorithms.

We address the information cost of classical algorithms, i.e., the number of function evaluations the algorithms use, as well as their total cost by taking into account the additional cost of the operations that are used for combining the function evaluations. Accordingly, the minimal information cost of any algorithm solving the problem with accuracy  $\varepsilon$  is the information complexity of the problem, while the minimal total cost of any algorithm with error at most  $\varepsilon$  is the problem complexity. Clearly, the information complexity provides a lower bound for the problem complexity.

Quantum algorithms use quantum queries to evaluate  $q$  at deterministically chosen points. (Recently, quantum algorithms with randomized queries have been considered [30] but we do not deal with them in this paper.) The query information is combined using a number of quantum operations. Quantum algorithms succeed in producing an  $\varepsilon$ -approximation with probability, say,  $\frac{3}{4}$ . The minimal number of queries of any algorithm solving the problem with accuracy  $\varepsilon$  is the query complexity. The total cost of a quantum algorithm takes into account the additional quantum operations, excluding the ones used for queries, required to solve the problem with accuracy  $\varepsilon$ . We will distinguish between quantum algorithms using two types of queries, bit queries and power queries. Bit queries are oracle calls similar to those in Grover's search algorithm [14]. Power queries are obtained by considering the propagator of the system at different time steps, as in phase estimation [19].

In some cases, quantum algorithms may be used to solve parts of the problem while other parts may be solved classically. In such a case we need to consider the cost of the classical and the quantum parts.

The definition of the error of algorithms and the details of the model of computation in the different settings can be found in [21] but we will include them in this paper for the convenience of the reader.

Turning to the eigenvalue problem we show a perturbation formula relating the eigenvalues  $\lambda(q)$  and  $\lambda(\bar{q})$  for two functions  $q$  and  $\bar{q}$ , as in [21]. In particular, we show that

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

where  $u_{\bar{q}}$  is the eigenfunction that corresponds to  $\lambda(\bar{q})$ . Using this equation we reduce the eigenvalue problem to the integration problem.

For deterministic and randomized classical algorithms we use known lower bounds [24] for the information complexity of integration to obtain lower bounds for the information complexity of the eigenvalue problem. For upper bounds we study the cost of particular algorithms that approximate  $\lambda(q)$  with error  $\varepsilon$ . We show that by discretizing the continuous problem and solving the resulting matrix eigenvalue problem we obtain an optimal deterministic algorithm. Optimality is understood modulo multiplicative constants that depend on  $d$ . We derive a randomized algorithm using the perturbation formula above. Roughly speaking, the idea is to first approximate  $q$  by a function  $\bar{q}$ , and then to approximate the first two terms in the right hand side of the perturbation formula. Using a matrix discretization we approximate  $\lambda(\bar{q})$  and using Monte Carlo we approximate the weighted integral. We

derive the cost of the algorithm and show that it has optimal information complexity only when  $d \leq 2$ . Proving the optimality of this algorithm for  $d > 2$  is an open question at this time.

In summary, denoting by  $n(\varepsilon)$  and  $\text{comp}(\varepsilon)$  the information complexity and the problem complexity, for deterministic algorithms we have

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

while for randomized algorithms we have

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

where the asymptotic constants depend on  $d$ , and  $\mathbf{c}$  denotes the cost of one function evaluation. It is worth pointing out that even if one is able to obtain matching information complexity bounds for any  $d$ , the combinatorial cost (i.e., the number of operations excluding function evaluations) of the randomized algorithm is still exponential in  $d$ , because we have to solve a matrix eigenvalue problem and the size of the matrix is exponential in  $d$ .

For quantum algorithms, we treat algorithms using bit queries and power queries separately. For quantum algorithms with bit queries we use the perturbation formula above to reduce the problem to integration. We obtain lower bounds for the query complexity of integration, which yield lower bounds for the query complexity of the eigenvalue problem. We see that we can modify the classical randomized algorithm we discussed above, to obtain a hybrid algorithm, i.e., an algorithm with classical and quantum parts. The only difference with the randomized algorithm is that, instead of using Monte Carlo, we approximate the weighted integral in the perturbation formula by a quantum algorithm. The quantum algorithm that approximates the integral is due to Novak [20].

We show that the number of queries plus the number of classical function evaluations of the hybrid algorithm matches the query complexity lower bound only when  $d = 1$ . Then  $q \in C^1([0, 1])$ , while for  $q \in C^2([0, 1])$  the same result has been shown in [21]. When  $d > 1$  we only show that the algorithm has information cost and uses a number of classical function evaluations that is exponential in  $d$ . The cost of approximating  $q$  by  $\bar{q}$  with error  $\varepsilon$  is dominant in the worst case. As we already indicated, even if we are able to show matching upper and lower bounds for the query complexity that are also proportional to the classical information cost when  $d > 1$ , the number of classical operations required by the algorithm is still exponential in  $d$ , due to the cost of the matrix eigenvalue problem.

However, there is a different quantum algorithm (without any classical parts) that uses bit queries whose cost is not exponential in  $d$ . Indeed, we can use phase estimation to solve the problem. Phase estimation typically uses power queries [1, 19] but we can approximate the power queries using a number of bit queries that is polynomial in  $\varepsilon^{-1}$ , where the degree of the polynomial is independent of  $d$ . Denoting the query complexity by  $n^{\text{query}}(\varepsilon)$  we show that for bit queries

$$\Omega(\varepsilon^{-d/(d+1)}) = n^{\text{query}}(\varepsilon) = O(\varepsilon^{-6} \log^2 \varepsilon^{-1}),$$

where the asymptotic constants depend on  $d$ . Moreover the algorithm uses a number of quantum operations, excluding the queries, that is proportional to

$$d\varepsilon^{-6} \log^4 \varepsilon^{-1},$$

a number of qubits proportional to

$$d \log \varepsilon^{-1},$$

and the algorithm succeeds with probability at least  $\frac{3}{4}$ . We remark that due to the results of [30] the number of qubits is optimal modulo multiplicative constants.

Phase estimation with power queries has a considerable advantage since

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

where the asymptotic constant is an absolute constant, and the lower bound follows using the results of [5, 6]. The number of quantum operations, excluding queries, is proportional to

$$\log^2 \varepsilon^{-1},$$

the number of qubits is proportional to

$$d \log \varepsilon^{-1}$$

and thereby optimal, while the algorithm succeeds with probability at least  $\frac{3}{4}$ .

## 2 Problem Definition

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

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

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

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

$$u(x) \equiv 0, \quad x \in \partial I_d. \tag{2}$$

In the variational form, the smallest eigenvalue  $\lambda = \lambda(q)$  of (1, 2) is given by

$$\lambda(q) = \min_{0 \neq u \in H_0^1} \frac{\int_{I_d} \sum_{j=1}^d [D_j u(x)]^2 + q(x) u^2(x) dx}{\int_{I_d} u^2(x) dx}. \tag{3}$$

We will study the complexity of classical and quantum algorithms approximating  $\lambda(q)$  with error  $\varepsilon$ . We will show asymptotic bounds for the error of the algorithms and the problem complexity, assuming that  $d$  is fixed. Henceforth, all asymptotic constants in the error estimates, the complexity estimates and the cost of algorithms are either absolute constants or depend on  $d$ . Often we will be addressing these constants. In some cases their nature will be evident from the properties of the algorithm under consideration, but in all cases, especially when the constants are omitted from the discussion, the reader may assume they depend only on  $d$  for simplicity.

## 2.1 Preliminary Analysis

The properties of the eigenvalues and eigenvectors of problems such as (1, 2) (defined on a rectangular domain) are discussed extensively in [23] where it is shown that the eigenfunctions are continuous and they have continuous partial derivatives of the first order, including the boundary of  $I_d$ .

The operator  $\mathbb{L}_q$  is symmetric and its eigenvalues and eigenvectors are real. The eigenvalues are positive, they can be indexed in nondecreasing order

$$0 < \lambda_1(q) \leq \lambda_2(q) \leq \dots \leq \lambda_k(q) \leq \dots,$$

and the sequence of eigenvalues tends to infinity. We denote the corresponding eigenvectors by  $u_{q,k}$ ,  $k = 1, 2, \dots$ .

The smallest eigenvalue  $\lambda(q) \equiv \lambda_1(q)$  is simple, the corresponding eigenspace has dimension one, and the eigenvector,  $u_q \equiv u_{q,1}$ , is uniquely determined up to the sign. It is convenient to assume that the  $u_{q,k}$  are normalized, i.e.,

$$\|u_{q,k}\|_{L_2} := \left( \int_{I_d} u_{q,k}^2(x) dx \right)^{1/2} = 1, \quad k = 1, 2, \dots$$

Thus they form a complete orthonormal system in  $L_2(I_d)$ . Then (3) becomes

$$\begin{aligned} \lambda(q) &= \min_{u \in H_0^1, \|u\|_{L_2}=1} \int_{I_d} \sum_{j=1}^d (D_j u)^2(x) + q(x) u^2(x) dx \\ &= \int_{I_d} \sum_{j=1}^d (D_j u_q)^2(x) + q(x) u_q^2(x) dx. \end{aligned} \quad (4)$$

For additional details concerning the properties of eigenvalues and eigenfunctions of elliptic operators as well as numerical methods approximating them, see [2, 11, 12, 13] and the references therein.

For a constant function  $q \equiv c$  we know that

$$\lambda(c) = d\pi^2 + c, \quad \text{and} \quad u_c(x_1, \dots, x_d) = 2^{d/2} \prod_{j=1}^d \sin(\pi x_j).$$

It is also known that the eigenvalues of  $\mathbb{L}_q$  are nondecreasing functions of  $q$  [11, 23], i.e.,  $q(x) \leq \bar{q}(x)$ , for all  $x \in [0, 1]^d$ , implies that  $\lambda_k(q) \leq \lambda_k(\bar{q})$ , for all  $k = 1, 2, \dots$ . Thus, using (4) for the class  $\mathbf{Q}$  we get

$$d\pi^2 = \lambda(0) \leq \lambda(q) \leq d\pi^2 + 1, \quad q \in \mathbf{Q}.$$

For  $d > 1$ , the eigenvalues of  $\mathbb{L}_q$  are, generally, not all simple. However, as in the case  $d = 1$ , the smallest eigenvalue  $\lambda(q)$  is simple and is well separated from the remaining eigenvalues. This is because of the nondecreasing property of the eigenvalues of  $\mathbb{L}_q$  with respect to  $q$ , and

the fact that the second smallest eigenvalue of  $\mathbb{L}_0$  is equal to  $\lambda_2(0) = (d+3)\pi^2$ . Therefore, using  $\lambda(q) \leq d\pi^2 + 1$ , we obtain

$$\lambda_k(q) - \lambda(q) \geq \lambda_2(q) - \lambda(q) \geq 3\pi^2 - 1, \quad k \geq 3, \quad q \in \mathbf{Q}. \quad (5)$$

We will use this fact to establish an estimate for the smallest eigenvalue by considering a perturbation of  $q$ .

For any two functions  $q, \bar{q} \in \mathbf{Q}$  we have

$$|\lambda(q) - \lambda(\bar{q})| \leq \|q - \bar{q}\|_\infty \quad (6)$$

$$\|u_q - u_{\bar{q}}\|_{L_2} \leq O(\|q - \bar{q}\|_\infty), \quad (7)$$

$$\lambda(q) = \lambda(\bar{q}) + \int_{I_d} (q(x) - \bar{q}(x)) u_{\bar{q}}^2(x) dx + O(\|q - \bar{q}\|_\infty^2). \quad (8)$$

Equations (6) and (8) are derived as in [21]. They follow from elementary arguments and (7). For the convenience of the reader we point out that it is easy to show that

$$\lambda(q) \leq \lambda(\bar{q}) + \int_{I_d} (q(x) - \bar{q}(x)) u_{\bar{q}}^2(x) dx + \int_{I_d} (q(x) - \bar{q}(x))(u_{\bar{q}}^2(x) - u_q^2(x)) dx,$$

and similarly

$$\lambda(\bar{q}) \leq \lambda(q) + \int_{I_d} (\bar{q}(x) - q(x)) u_q^2(x) dx.$$

These inequalities imply (6), and using them with (7) we obtain (8). Moreover,

$$\int_{I_d} (q(x) - \bar{q}(x))(u_{\bar{q}}^2(x) - u_q^2(x)) dx \geq 0.$$

We prove equation (7) using an approach similar to that in [29], which is based on the separation between  $\lambda_2(q)$  and  $\lambda(q)$ . It is a different proof from the one used in [21].

Indeed, let  $q$  and  $\bar{q}$  be two functions from the class  $\mathbf{Q}$  and consider  $\mathbb{L}_q$  and  $\mathbb{L}_{\bar{q}}$ . Let  $\lambda_k(q)$ ,  $u_{q,k}$  and  $\lambda_k(\bar{q})$ ,  $u_{\bar{q},k}$ , be the eigenvalues and the normalized eigenvectors,  $k = 1, 2, \dots$ , of  $\mathbb{L}_q$  and  $\mathbb{L}_{\bar{q}}$ , respectively. Then

$$\mathbb{L}_{\bar{q}} u_q - \lambda(q) u_q = \mathbb{L}_q u_q + (\bar{q} - q) u_q - \lambda(q) u_q,$$

which implies that

$$\|\mathbb{L}_{\bar{q}} u_q - \lambda(q) u_q\|_{L_2} = \|(\bar{q} - q) u_q\|_{L_2} \leq \|\bar{q} - q\|_\infty.$$

Since the eigenvectors of  $\mathbb{L}_{\bar{q}}$  form a complete orthonormal system in  $L_2(I_d)$  we have

$$\begin{aligned} u_q &= \sum_{k=1}^{\infty} a_k u_{\bar{q},k}, \quad \text{with} \quad \|u_q\|_{L_2}^2 = \sum_{k=1}^{\infty} a_k^2 = 1, \quad a_k \in \mathbb{R}, \quad \text{and} \\ \mathbb{L}_{\bar{q}} u_q &= \sum_{k=1}^{\infty} a_k \lambda_k(\bar{q}) u_{\bar{q},k}. \end{aligned}$$

Thus

$$\begin{aligned} \|q - \bar{q}\|_\infty^2 &\geq \left\| \sum_{k=1}^{\infty} a_k [\lambda_k(\bar{q}) - \lambda(q)] u_{\bar{q},k} \right\|_{L_2}^2 = \sum_{k=1}^{\infty} a_k^2 |\lambda_k(\bar{q}) - \lambda(q)|^2 \\ &\geq \sum_{k=2}^{\infty} a_k^2 |\lambda_k(\bar{q}) - \lambda(q)|^2 \geq (3\pi^2 - 1)^2 \sum_{k=2}^{\infty} a_k^2 = (3\pi^2 - 1)^2 (1 - a_1^2), \end{aligned}$$

where the last inequality is due to the lower bound (5). Thus

$$a_1^2 \geq 1 - (3\pi^2 - 1)^{-2} \|q - \bar{q}\|_\infty^2. \quad (9)$$

Observe that the inequality above implies that  $a_1^2 \geq 0.99$ . Without loss of generality we assume that the sign of  $u_q$  has been chosen such that  $a_1 > 0$  and then

$$a_1 \geq \sqrt{1 - (3\pi^2 - 1)^{-2} \|q - \bar{q}\|_\infty^2}.$$

Also

$$\|u_q - u_{\bar{q}}\|_{L_2}^2 = (1 - a_1)^2 + \sum_{k=2}^{\infty} a_k^2 \quad (10)$$

$$\begin{aligned} &= (1 - a_1)^2 + 1 - a_1^2 = 2(1 - a_1) \\ &\leq 2 \left[ 1 - \sqrt{1 - (3\pi^2 - 1)^{-2} \|q - \bar{q}\|_\infty^2} \right] \\ &\leq (3\pi^2 - 1)^{-2} \|q - \bar{q}\|_\infty^2, \end{aligned} \quad (11)$$

where the last inequality is due to the fact that  $(3\pi^2 - 1)^{-2} \|q - \bar{q}\|_\infty^2 \in (0, 1)$ , and this proves (7).

As a final remark, we observe that the same analysis that led to (9) can be used to establish that  $\lambda(q)$  is indeed a simple eigenvalue for any  $q \in \mathbf{Q}$ . Clearly  $\lambda(0)$  is simple. Using (9) with  $\bar{q} = 0$ , we obtain that the square of the projection of  $u_0$  onto  $u_q$  is bounded from below, i.e., that  $\left\{ \int_{I_d} u_0(x) u_q(x) dx \right\}^2 > 1/2$ . If  $\lambda(q)$  were not simple and the eigenspace corresponding to it had dimension greater than one, there would be at least two orthogonal eigenfunctions  $u_{q,1}$  and  $u_{q,2}$  (both corresponding to  $\lambda(q)$ ). Then each of the projections of  $u_{q,1}$  and  $u_{q,2}$  on  $u_0$  satisfies the preceding inequality (since  $\lambda(0)$  is simple.) Thus, expanding  $u_0$  using the eigenfunctions of  $\mathbb{L}_q$  would lead us to conclude that  $\|u_0\|_{L_2} > 1$ , a contradiction since we have assumed  $u_0$  is a normalized eigenfunction..

### 3 Classical Algorithms

Let us now discuss the type of classical algorithms we consider and define how we measure their error and cost. 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  $t_i \in [0, 1]^d$ . Here,  $i = 1, 2, \dots, n_q$ , for some  $n_q$ , and the points  $t_i$  can be adaptively chosen, i.e.,  $t_i$  can be a function

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

of the previously computed function values and points for  $i \geq 2$ . The number  $n_q$  can also be adaptively chosen, see, e.g., [24] for details.

A classical deterministic algorithm produces an approximation

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

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  $\phi$  is given by

$$e^{\text{wor}}(\phi, n) = \sup_{q \in \mathbf{Q}} |\lambda(q) - \phi(q)|. \quad (12)$$

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

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

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

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

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

$$e^{\text{ran}}(\phi, n) = \sup_{q \in \mathbf{Q}} (\mathbb{E}[\lambda(q) - \phi_\omega(q)]^2)^{1/2}. \quad (13)$$

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

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

respectively. We refer to  $n^{\text{wor}}(\varepsilon)$  and  $n^{\text{ran}}(\varepsilon)$  as the worst case and the randomized case information complexity, respectively.

We also consider the cost of combining the function evaluations. For a function  $q \in \mathbf{Q}$ , let  $m_q$  be the number of arithmetic operations used by an algorithm in order to combine  $n_q$  function values and obtain the final result. Then the worst case cost of an algorithm  $\phi$  is defined as

$$\text{cost}^{\text{wor}}(\phi) = \sup_{q \in \mathbf{Q}} (\mathbf{c} n_q + m_q),$$

where  $\mathbf{c}$  denotes the cost of an evaluation of  $q$ . The worst case complexity  $\text{comp}^{\text{wor}}(\varepsilon)$  is defined as the minimal cost of an algorithm whose worst case error is at most  $\varepsilon$ ,

$$\text{comp}^{\text{wor}}(\varepsilon) = \min\{\text{cost}^{\text{wor}}(\phi) : \phi \text{ such that } e^{\text{wor}}(\phi, n) \leq \varepsilon\}.$$



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

The cost of a randomized algorithm  $\phi$  using  $n = \sup_{q \in \mathbf{Q}} \mathbb{E}(n_{q,\omega}) < \infty$  randomized function evaluations is defined as

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

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  $\omega$ . The randomized complexity

$$\text{comp}^{\text{ran}}(\varepsilon) = \min \{ \text{cost}^{\text{ran}}(\phi) : \phi \text{ such that } e^{\text{ran}}(\phi, n) \leq \varepsilon \},$$

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

### 3.1 Deterministic Algorithms

In this section we derive lower and upper bounds for the error and the complexity of deterministic algorithms in the worst case. We begin with the lower bounds. Our derivation is based on the proof in [21] which deals with the case  $d = 1$ . Let  $\bar{q} = 1/2$  and consider  $q \in \mathbf{Q}$  such that  $\|q - 1/2\|_\infty \leq c$ . Then  $u_{1/2}$  is known and (8) becomes

$$\lambda(q) = d\pi^2 + \frac{1}{2} + 2^d \int_{I_d} \left( q(x_1, \dots, x_d) - \frac{1}{2} \right) \prod_{j=1}^d \sin^2(\pi x_j) dx_1 \dots dx_d + \delta, \quad (14)$$

where  $|\delta| = O(c^2)$ . Recall that  $\delta \leq 0$  because the first three terms in the right hand side of the equation overestimate  $\lambda(q)$  due to (4). Functions that differ by a constant satisfy the above equation with the same value of  $\delta$ . Assume that  $c > 0$  is sufficiently small so that  $c + |\delta| < 1/2$ .

We will reduce the eigenvalue problem to the multivariate integration problem and use the well known [24] lower bounds for integration to establish a lower bound for the eigenvalue problem.

Consider the class of functions

$$F_c = \{ f : I_d \rightarrow \mathbb{R} \mid f, D_j f \in C(I_d), \|D_j f\|_\infty \leq 1, j = 1, \dots, d, \|f\|_\infty \leq c \}. \quad (15)$$

and the approximation of weighted integrals of the form

$$S(f) = \int_{I_d} f(x_1, \dots, x_d) \prod_{j=1}^d \sin^2(\pi x_j) dx_1 \dots dx_d. \quad (16)$$

The worst case error of any deterministic algorithm approximating such integrals using  $n$  points in  $I_d$  is  $\Omega(n^{-1/d})$ , where the asymptotic constant depends on  $d$ . Here we assume that  $n$  is large enough so that  $c \gg n^{-1/d}$ . This lower bound is known [24] for integration without weights but the same proofs carry over to this case.

Take an  $f \in F_c$  and set  $q = f + 1/2$ . Then  $q$  belongs to  $\mathbf{Q}$ . The functions  $q \pm \delta$  also belong to the class  $\mathbf{Q}$  because  $c + |\delta| < 1/2$ . Let  $\tilde{q} = q - \delta$ . Then

$$\lambda(\tilde{q}) = d\pi^2 + \frac{1}{2} + 2^d S(f).$$

Let  $\hat{\lambda}(\tilde{q})$  be an algorithm approximating  $\lambda(\tilde{q})$  using  $n$  function evaluations of  $\tilde{q}$  at deterministic points. Then

$$\phi(f) = 2^{-d} \left[ \hat{\lambda}(\tilde{q}) - d\pi^2 - \frac{1}{2} \right], \quad (17)$$

is an algorithm approximating the weighted integral  $S(f)$  with error

$$|S(f) - \phi(f)| = 2^{-d} |\hat{\lambda}(\tilde{q}) - \lambda(\tilde{q})|.$$

In the worst case with respect to  $f$  this quantity is  $\Omega(n^{-1/d})$ .

Hence, the error of any deterministic algorithm  $\hat{\lambda}$  that approximates  $\lambda(q)$ , for  $q \in \mathbf{Q}$ , using  $n$  evaluations of  $q$  is bounded from below as follows

$$e^{\text{wor}}(\hat{\lambda}, n) = \sup_{q \in \mathbf{Q}} |\hat{\lambda}(q) - \lambda(q)| = \Omega(n^{-1/d}),$$

where the asymptotic constant depends on  $d$ . Therefore, the worst case information complexity  $n^{\text{wor}}(\varepsilon)$  is bounded from below by a quantity proportional to  $\varepsilon^{-d}$ .

Let us now consider upper bounds for the problem complexity. We discretize  $\mathbb{L}_q$  at the points  $(i_1 h, \dots, i_d h)$ ,  $i_j = 1, \dots, m$ ,  $j = 1, \dots, d$ , where  $h = (m+1)^{-1}$ , and we obtain an  $m^d \times m^d$  matrix  $M_h(q) = -\Delta_h + B_h(q)$ , where  $-\Delta_h$  is the  $m^d \times m^d$  matrix resulting from the  $(2d+1)$ -point finite difference discretization of the Laplacian [12, 13]. The matrix  $B_h(q)$  is diagonal containing evaluations of  $q$  at all the discretization points. The matrix  $M_h(q)$  is sparse, symmetric positive definite, and its smallest eigenvalue approximates the smallest eigenvalue of  $\mathbb{L}_q$  with error  $O(h)$  [26, 27], i.e.,

$$|\lambda(\bar{q}) - \lambda(M_h(\bar{q}))| = O(h).$$

For example when  $d = 2$  we have

$$-\Delta_h = h^{-2} \begin{pmatrix} T_h & -I & & & & \\ -I & T_h & -I & & & \\ & & \ddots & \ddots & \ddots & \\ & & & -I & T_h & -I \\ & & & & -I & T_h \end{pmatrix} \quad \text{and} \quad B_h(\bar{q}) = \begin{pmatrix} b_{11} & & & & & \\ & \ddots & & & & \\ & & b_{ij} & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & b_{mm} \end{pmatrix},$$

where  $I$  is the  $m \times m$  identity matrix,  $b_{ij} = q(ih, jh)$ ,  $i, j = 1, \dots, m$ , and  $T_h$  is the  $m \times m$  matrix given by

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

see [12, p. 270] for more details.

The matrix  $-\Delta_h$  has been extensively studied in the literature; see [12, 13] and the references therein. Its eigenvalues and eigenvectors are known. The smallest eigenvalue of  $M_h(0) = -\Delta_h$  is  $\lambda(M_h(0)) = 4dh^{-2} \sin^2(\pi h/2) = d\pi^2(1 + O(h^2))$ . Moreover, the eigenvectors of  $M_h(0)$  are tensor products of the eigenvectors of the corresponding matrix in the one-dimensional case  $d = 1$ . This is also trivially true for the eigenvectors of  $M_h(c)$ , where  $c$  is any constant. Using results concerning the eigenvalues of perturbed symmetric matrices [29] we have that the smallest eigenvalue  $\lambda(M_h(q))$  of  $M_h(q)$  satisfies

$$|\lambda(M_h(0)) - \lambda(M_h(q))| \leq 1.$$

Moreover, the eigenvalues  $\lambda_k(M_h(q))$ ,  $k = 1 \dots, m^d$ , (indexed in non-decreasing order) satisfy an equation similar to (5), namely,

$$\lambda_k(M_h(q)) - \lambda(M_h(q)) \geq \lambda_2(M_h(q)) - \lambda(M_h(q)) \geq 3\pi^2 - 2, \quad k \geq 3, \quad q \in \mathbf{Q}. \quad (18)$$

The inequalities follow from results concerning the eigenvalues of the sum of two symmetric matrices [29, p. 101], and the separation of the eigenvalues of the matrix  $M_h(0) = -\Delta_h$ .

We can approximate the smallest eigenvalue of  $M_h(q)$  with error  $h$  using the bisection method [12, p. 228] in  $O(\log m)$  steps. Each step takes a number of arithmetic operations proportional to the number of non-zero elements in  $M_h(q)$ , which is  $O(m^d)$ , with the asymptotic constant depending on  $d$ . Hence, the total cost of approximating  $\lambda(M_m(q))$  is  $O(m^d \log m)$ , and the asymptotic constant depends on  $d$ .

Setting  $m + 1 = \varepsilon^{-1}$ , we obtain an algorithm that approximates  $\lambda(q)$  by the smallest eigenvalue of the matrix  $\lambda(M_\varepsilon(q))$ . This algorithm has error  $O(\varepsilon)$  and uses  $\varepsilon^{-d}$  evaluations of  $q$ , and  $O(\varepsilon^{-d} \log \varepsilon^{-1})$  arithmetic operations.

Combining the lower bound for  $n^{\text{wor}}(\varepsilon)$  from the first part of this section with the cost of the algorithm above we obtain the following theorem.

**Theorem 3.1.**

$$n^{\text{wor}}(\varepsilon) = \Theta(\varepsilon^{-d}), \quad \Omega(\mathbf{c} \varepsilon^{-d}) = \text{comp}^{\text{wor}}(\varepsilon) = O(\mathbf{c} \varepsilon^{-d} + \varepsilon^{-d} \log \varepsilon^{-1}),$$

where the asymptotic constants depend on  $d$ .

We conclude this section by remarking that we can extend these results about  $n^{\text{wor}}(\varepsilon)$  to the case where  $q$  has continuous and bounded partial derivatives up to order  $r$ . The same approach yields that  $n^{\text{wor}}(\varepsilon) = \Theta(\varepsilon^{-d/r})$ . So we have a *delayed curse of dimension*.

### 3.2 Randomized Algorithms

We first prove lower bounds for  $n^{\text{ran}}(\varepsilon)$  just as we proved lower bounds for  $n^{\text{wor}}(\varepsilon)$ . We reduce the problem to multivariate integration and use the known randomized information complexity lower bounds for integration. Recall the perturbation formula (14), the definition (15) of the class  $F_c$  and the weighted integration problem (16).

Assuming that  $n$  is sufficiently large so that  $c \gg n^{-(d+2)/(2d)}$ , we know [24] that the error of any randomized algorithm that approximates the weighted integral  $S(f)$  using  $n$  function

evaluations at randomly chosen points is bounded from below by a quantity proportional to  $n^{-(d+2)/(2d)}$ . (As we already mentioned, this is known for integrals without weights but the same proofs carry over to this case.)

For  $f \in F_c$ , set  $q = f + 1/2 \in \mathbf{Q}$  and  $\tilde{q} = q - \delta \in \mathbf{Q}$ ; see (14). Let  $\hat{\lambda}_\omega$  be any randomized algorithm that uses  $n$  function evaluations to approximate  $\lambda(\tilde{q})$ . Then  $\phi_\omega(f)$ , defined by replacing  $\hat{\lambda}$  with  $\hat{\lambda}_\omega$  in (17), is a randomized algorithm approximating  $S(f)$ , and its error is

$$\{\mathbb{E}[S(f) - \phi_\omega(f)]^2\}^{1/2} = 2^{-d} \{\mathbb{E}[\lambda_\omega(q) - \lambda(q)]^2\}^{1/2}.$$

Taking the worst case with respect to  $f$  we see that this quantity is  $\Omega(n^{-(d+2)/(2d)})$ .

Therefore for any randomized algorithm  $\hat{\lambda}$  that approximates  $\lambda(q)$ , for  $q \in \mathbf{Q}$ , using  $n$  function evaluations of  $q$  at randomly chosen points, we have

$$e^{\text{ran}}(\hat{\lambda}, n) = \Omega(n^{-(d+2)/(2d)}),$$

which implies that  $n^{\text{ran}}(\varepsilon) = \Omega(\varepsilon^{-2d/(d+2)})$ , where the asymptotic constant depends on  $d$ .

We now derive upper bounds for  $\text{comp}^{\text{ran}}$  by constructing an algorithm. First we take  $(n+1)^d$  samples of  $q$  on a grid of equally spaced points  $(i_1/n, \dots, i_d/n)$ ,  $i_j = 0, \dots, n$ ,  $j = 1, \dots, d$ . Using these points we construct a piecewise polynomial  $\tilde{q}$  by interpolation. For instance,  $\tilde{q}$  can be a natural spline. Then  $\|\tilde{q} - q\|_\infty = O(n^{-1})$ . Setting  $\bar{q} = \tilde{q} + O(n^{-1})$  we have that  $\bar{q} \geq 0$  and  $\|\bar{q} - q\|_\infty = O(n^{-1})$ . Clearly, given the evaluations of  $q$ ,  $\bar{q}$  can be constructed with  $O(n^d)$  arithmetic operations.

The perturbation formula (8) for  $q$  and  $\bar{q}$  becomes

$$\lambda(q) = \lambda(\bar{q}) + \int_{I_d} (q(x) - \bar{q}(x)) u_{\bar{q}}^2(x) dx + O(n^{-2}). \quad (19)$$

We will approximate  $\lambda(q)$  by an algorithm that

1. computes  $\hat{\lambda}(\bar{q})$  that approximates  $\lambda(\bar{q})$  by discretizing  $\mathbb{L}_{\bar{q}}$  and solving a matrix eigenvalue problem,
2. replaces  $u_{\bar{q}}$  in the integral above by an approximate eigenfunction  $\hat{u}_{\bar{q}}$ , and
3. approximates the resulting integral by Monte Carlo.

Therefore, (19) becomes

$$\lambda(q) = \lambda(\bar{q}) + \int_{I_d} (q(x) - \bar{q}(x)) \hat{u}_{\bar{q}}^2(x) dx + \int_{I_d} (q(x) - \bar{q}(x)) (u_{\bar{q}}^2(x) - \hat{u}_{\bar{q}}^2(x)) dx + O(n^{-2}), \quad (20)$$

and the algorithm approximates the first two terms in the right hand side of this expression. In particular, the algorithm is given by

$$\tilde{\lambda}(q) := \hat{\lambda}(\bar{q}) + \frac{1}{k} \sum_{i=1}^k (q(t_{i,\omega}) - \bar{q}(t_{i,\omega})) \hat{u}_{\bar{q}}^2(t_{i,\omega}), \quad (21)$$

where  $t_{1,\omega}, \dots, t_{k,\omega}$  are independent random numbers that follow the uniform distribution in  $I_d$ . Then the expected error of this algorithm satisfies

$$\begin{aligned} & \left\{ \mathbb{E}[\lambda(q) - \tilde{\lambda}(q)]^2 \right\}^{1/2} \leq |\lambda(\bar{q}) - \hat{\lambda}(\bar{q})| \\ & + \left\{ \mathbb{E} \left[ \int_{I_d} (q(x) - \bar{q}(x)) \hat{u}_{\bar{q}}^2(x) dx - \frac{1}{k} \sum_{i=1}^k (q(t_{i,\omega}) - \bar{q}(t_{i,\omega})) \hat{u}_{\bar{q}}^2(t_{i,\omega}) \right]^2 \right\}^{1/2} \\ & + \|u_{\bar{q}} - \hat{u}_{\bar{q}}\|_{L_2} O(n^{-1}) + O(n^{-2}) \end{aligned} \quad (22)$$

Let us now discuss the individual steps of the algorithm and the resulting errors. We discretize the operator  $\mathbb{L}_{\bar{q}}$  on a grid with mesh size  $h = (m+1)^{-1}$ , exactly as we did in the previous section. The smallest eigenvalue  $\lambda(M_h(\bar{q}))$  of the resulting matrix approximates  $\lambda(q)$  with error

$$|\lambda(\bar{q}) - \lambda(M_h(\bar{q}))| = O(h), \quad (23)$$

see [27]. We approximate  $\lambda(M_h(\bar{q}))$  by  $\hat{\lambda}(M_h(\bar{q}))$  with error

$$|\hat{\lambda}(M_h(\bar{q})) - \lambda(M_h(\bar{q}))| \leq h, \quad (24)$$

which we obtain using the bisection method with cost proportional to  $m^d \log m$  times a constant that depends on  $d$ . In (21), we set  $\hat{\lambda}(\bar{q}) := \hat{\lambda}(M_h(\bar{q}))$ .

We now show how to construct the approximate eigenfunction  $\hat{u}_{\bar{q}}$  required for the second step of our algorithm. Let  $z = z(M_h(\bar{q}))$  be the eigenvector of  $M_h(\bar{q})$  that corresponds to  $\lambda(M_h(\bar{q}))$ . We assume that  $z$  is normalized so that

$$\|z\|_2 := \left( \sum_{k=1}^{m^d} z_k^2 \right)^{1/2} = 1.$$

Given  $\hat{\lambda}(M_h(\bar{q}))$ , we compute an approximation of  $z$  using an inverse iteration with the matrix

$$M_h(\bar{q}) - \hat{\lambda}(M_h(\bar{q})) I.$$

We can compute the determinant of this matrix with cost proportional to  $m^d$ . If the matrix is singular, we can perturb  $\hat{\lambda}(M_h(\bar{q}))$  by  $h$  to obtain a nonsingular matrix. The initial vector in inverse iteration is  $z_0$ , the eigenvector of  $M_h(0)$  that corresponds to its smallest eigenvalue. Observe that the separation of eigenvalues of  $M_h(\bar{q})$  as expressed by (18) and arguments similar to those that led to (9) and which can be found in [29, p. 172], yield that  $(z_0^T z)^2 \geq 1 - \|\bar{q}\|_\infty^2 / (3\pi^2 - 2)^2$ . Since the projection of the initial vector onto the eigenvector of interest is sufficiently large, with  $O(\log m)$  inverse iteration steps we obtain an approximate eigenvector  $\hat{z}$ , with  $\|\hat{z}\|_2 = 1$ , such that

$$\|\hat{z} - z\|_2 = O(h).$$

The total cost to obtain  $\hat{z}$  is  $O(m^d \log m)$ .

The Rayleigh quotient

$$\mu_h = \frac{\hat{z}^T M_h(\bar{q}) \hat{z}}{\|\hat{z}\|_2^2},$$

also approximates  $\lambda(M_h(\bar{q}))$  with error  $O(h)$ . Using  $\hat{z}$  we construct the approximate eigenfunction  $\hat{u}_{\bar{q}}$  of  $\mathbb{L}_{\bar{q}}$  by a method suggested by Courant [10] and used in [27]. In particular, we subdivide  $I_d$  into simplices whose vertices are the grid points. Then we construct a piecewise linear function on each simplex that is zero on the boundary of  $I_d$  and interpolates the values of  $\hat{z}$  at the grid points; see [27] for the details. We denote the interpolating function by  $\tilde{u}_{\bar{q}}$ . The cost for constructing  $\tilde{u}_{\bar{q}}$  is  $O(m^d)$ .

Consider now the Rayleigh quotient

$$\mu = \frac{\int_{I_d} \sum_{j=1}^d [D_j \tilde{u}_{\bar{q}}(x)]^2 + \bar{q}(x) \tilde{u}_{\bar{q}}^2(x) dx}{\|\tilde{u}_{\bar{q}}\|_{L_2}^2} \quad (25)$$

for the function  $\tilde{u}_{\bar{q}}$ . From [27] we know that

$$\lambda(\bar{q}) \leq \mu \leq \mu_h + O(h).$$

Since  $|\mu_h - \lambda(M_h(\bar{q}))| = O(h)$ , the equation above and (23) imply that

$$|\mu - \lambda(\bar{q})| = O(h). \quad (26)$$

We set  $\hat{u}_{\bar{q}} := \tilde{u}_{\bar{q}} / \|\tilde{u}_{\bar{q}}\|_{L_2}$  with cost  $O(m^d)$ . Let us now estimate  $\|u_{\bar{q}} - \hat{u}_{\bar{q}}\|_{L_2}$ . Consider the eigenvalues  $\lambda_k(\bar{q})$  and eigenvectors  $u_{\bar{q},k}$ ,  $k = 1, \dots$ , of  $\mathbb{L}_{\bar{q}}$ . Then we have  $\hat{u}_{\bar{q}} = \sum_{k=1}^{\infty} a_k u_{\bar{q},k}$ , where  $\sum_{k=1}^{\infty} a_k^2 = 1$ . Thus from (25) we obtain

$$\mu = \sum_{k=1}^{\infty} \lambda_k(\bar{q}) a_k^2.$$

Equivalently,

$$0 = \sum_{k=1}^{\infty} a_k^2 [\lambda_k(\bar{q}) - \mu] = \sum_{k=2}^{\infty} a_k^2 [\lambda_k(\bar{q}) - \mu] - a_1^2 [\mu - \lambda_1(\bar{q})].$$

Using (5) and (26) and the fact that  $\lambda(\bar{q}) = \lambda_1(\bar{q})$  we obtain

$$a_1^2 [\mu - \lambda(\bar{q})] \geq (3\pi^2 - 2) \sum_{k=2}^{\infty} a_k^2 = (3\pi^2 - 2)(1 - a_1^2),$$

and using (26) again, we find that

$$1 - a_1^2 = O(h).$$

Hence,

$$\|u_{\bar{q}} - \hat{u}_{\bar{q}}\|_{L_2}^2 = O(h). \quad (27)$$

The proof of the last equation is the same as the proof we used to derive (11) from equation (9).

Recall that the algorithm (21) uses Monte Carlo to approximate the first integral in (20). It is well known that Monte Carlo (MC) with  $k$  function evaluations has error bounded from above by the  $L_2$  norm of the integrand times  $k^{-1/2}$ , i.e., the MC error does not exceed

$$n^{-1}k^{-1/2} \quad (28)$$

Combining (22) with (23), (24), (27), (28) we obtain that the expected error of the algorithm  $\tilde{\lambda}(q)$ , described in (21), is bounded from above by a quantity proportional to

$$m^{-1} + n^{-1}k^{-1/2} + n^{-1}m^{-1/2} + n^{-2}. \quad (29)$$

The cost of this algorithm is equal to  $n^d$  evaluations of  $q$  at deterministic points, plus  $k$  evaluations involving  $q$  (i.e., evaluations of  $(q - \bar{q})\hat{u}_{\bar{q}}^2$ ), plus a number of arithmetic operations proportional to  $n^d + m^d \log m + k$  times a constant that depends on  $d$ .

Taking  $m^{-1} = \varepsilon$  and observing that we can take  $k = n^d$  without changing the order of magnitude of the cost of the algorithm, expression (29) becomes

$$\varepsilon + n^{-(d+2)/2} + n^{-1}\varepsilon^{1/2} + n^{-2}. \quad (30)$$

The number of evaluations of  $q$  is proportional to  $n^d$  and the number of arithmetic operations is proportional to  $n^d + \varepsilon^{-d} \log \varepsilon^{-1}$  times a constant that depends on  $d$ .

The cost of approximating  $q$  by  $\bar{q}$  is proportional to  $n^d$ . It is worth noting that this is the dominant part of the algorithm cost. Indeed, even though we can approximate the first integral of (21) with high accuracy using Monte Carlo with  $O(n^d)$  function evaluations, the advantages of this approximation are lost when  $n^{-(d+2)/2} = O(n^{-2})$  since the eigenvalue error depends on  $O(n^{-2})$  as seen in (30).

Therefore when  $d \leq 2$ , we get error of order  $\varepsilon$  with  $\varepsilon^{-2d/(d+2)}$  function evaluations, while for  $d > 2$  we get error of order  $\varepsilon$  with  $\varepsilon^{-d/2}$  function evaluations. In both cases the number of arithmetic operations is proportional to  $\varepsilon^{-d} \log \varepsilon^{-1}$  times a constant that depends on  $d$ .

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

**Theorem 3.2.**

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

where the asymptotic constants depend on  $d$ .

When  $d > 2$  we do not have matching upper and lower bounds for  $n^{\text{ran}}(\varepsilon)$  and improving the upper bound is an open problem at this time. One possibility would be to use a perturbation formula of higher order of accuracy. On the other hand, we see that if we consider functions that have continuous and bounded mixed partial derivatives up to order  $r$  then our approach yields that

$$\Omega(\varepsilon^{-2d/(2r+d)}) = n^{\text{ran}}(\varepsilon) = O(\varepsilon^{-\max(2d/(2r+d), d/(2r))}),$$

which extends the range of values of  $d$  to  $1 \leq d \leq 2r$  for which we do have matching upper and lower bounds.

## 4 Quantum Algorithms

A quantum algorithm applies a sequence of unitary transformations to an initial state, and the final state is measured. See [3, 8, 15, 19] 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  $\nu$ -fold tensor product Hilbert space  $\mathcal{H}_\nu = \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$ , for some appropriately chosen integer  $\nu$ , where  $\mathbb{C}^2$  is the two dimensional space of complex numbers. The dimension of  $\mathcal{H}_\nu$  is  $2^\nu$ . The number  $\nu$  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.,

$$|\psi\rangle := U_T Q_Y U_{T-1} Q_Y \cdots U_1 Q_Y U_0 |\psi_0\rangle. \quad (31)$$

Here,  $U_0, U_1, \dots, U_T$  are unitary matrices that do not depend on the input function  $q$ . The unitary matrix  $Q_Y$  with  $Y = [q(t_1), \dots, q(t_n)]$  is called a quantum query and depends on  $n$  (with  $n \leq 2^\nu$ ), function evaluations of  $q$  computed at some non-adaptive points  $t_i \in I_d$ . 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  $j \in \{0, 1, \dots, M-1\}$  occurs with probability  $p_Y(j)$ , which depends on  $j$  and the input  $Y$ . Knowing the outcome  $j$ , we compute an approximation  $\hat{\lambda}_Y(j)$  of the smallest eigenvalue on a classical computer.

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  $p = \frac{3}{4}$  in the rest of this section. As is common for quantum algorithms, we can achieve an  $\varepsilon$ -approximation with probability arbitrarily close to 1 by repeating 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

$$e(\hat{\lambda}_Y, T) = \min \left\{ \alpha : \sum_{j: |\lambda(q) - \hat{\lambda}_Y(j)| \leq \alpha} p_Y(j) \geq \frac{3}{4} \right\}.$$

This can be equivalently rewritten as

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

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

$$e^{\text{quant}}(\hat{\lambda}, T) = \sup \left\{ e(\hat{\lambda}_Y, T) : Y = [q(t_1), \dots, q(t_n)], \quad t_i \in [0, 1]^d, \quad \text{for } q \in \mathbf{Q} \right\}. \quad (32)$$



We define the query complexity  $n^{\text{query}}(\varepsilon)$  of a quantum algorithm by

$$n^{\text{query}}(\varepsilon) = \min\{T : \exists \hat{\lambda} \text{ such that } e^{\text{quant}}(\hat{\lambda}, T) \leq \varepsilon\}. \quad (33)$$

Moreover, since we will be dealing with two types of queries, bit queries and power queries, we will be using the notation  $n^{\text{bit-query}}(\varepsilon)$  and  $n^{\text{power-query}}(\varepsilon)$ , respectively, to label the query complexity by the type of queries used.

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., [15].

Classical algorithms in floating or fixed point arithmetic can also be written in the form of (31). 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 the next section.

We formally use the real number model of computation [25]. Since our 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  $\log \varepsilon^{-1}$ .

*Hybrid* algorithms, which are combinations of classical and quantum algorithms, can be viewed as finite sequences of algorithms of the form (31) and can be expressed as one quantum algorithm of the form (31), see [15, 16]. Consequently, when proving lower bounds it suffices to consider only algorithms of the form (31). For upper bounds it is sometimes convenient to distinguish between classical and quantum computations and charge their costs differently. The cost of classical computations is defined in the previous section. The cost of quantum computations is defined as the sum of the number of quantum queries multiplied by the cost of a query, plus the number of quantum operations other than queries. It is also important to indicate how many qubits are used by the quantum algorithm.

## 4.1 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  $\varepsilon$ -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 as used in Grover's search algorithm [14]. For a Boolean function  $f : \{0, 1, \dots, 2^m - 1\} \rightarrow \{0, 1\}$ , the bit query is defined by

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

Here  $\nu = m + 1$ ,  $|j\rangle \in \mathcal{H}_m$ , and  $|k\rangle \in \mathcal{H}_1$  with  $\oplus$  denoting addition modulo 2. For real functions  $g$  the bit query is constructed by taking the most significant bits of the function

evaluated at some points  $t_j$ . More precisely, as in [15], the bit query for the function  $q$  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  $\beta : [0, 1] \rightarrow \{0, 1, \dots, 2^{m''} - 1\}$  and  $\tau : \{0, 1, \dots, 2^{m'} - 1\} \rightarrow I_d$ , and  $\oplus$  denotes addition modulo  $2^{m''}$ . Hence, we compute  $q$  at  $t_j = \tau(j) \in I_d$  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 [15].

The quantum amplitude amplification algorithm of Brassard et al. [7] computes the mean of a Boolean function defined on the set of  $N$  elements with accuracy  $\varepsilon$  and probability  $\frac{3}{4}$  using of order  $\min\{N, \varepsilon^{-1}\}$  bit queries. Modulo multiplicative factors, it is an optimal algorithm, in terms of the number of bit queries.

This algorithm can be also used to approximate the mean of a real function  $f : I_d \rightarrow \mathbb{R}$  with  $|f(x)| \leq M$ ,  $x \in I_d$ , see [15, 20]. More precisely, if we want to approximate

$$S_N(f) := \frac{1}{N} \sum_{j=0}^{N-1} f(x_j)$$

for some  $x_j \in I_d$  and  $N$ , then the amplitude amplification algorithm  $QS_N(f)$  approximates  $S_N(f)$  such that

$$|S_N(f) - QS_N(f)| \leq \varepsilon \quad \text{with probability } \frac{3}{4} \quad (34)$$

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

We begin by showing a lower bound for the query complexity,  $n^{\text{bit-query}}(\varepsilon)$ , of the eigenvalue problem. We do this by first estimating the bit query complexity,  $n^{\text{bit-query}}(\varepsilon, \text{INT}_{F_c})$ , of the weighted integration problem (16) in the class  $F_c$ , as defined in (15), and then reducing the eigenvalue problem to the integration problem.

From [20] we have

$$n^{\text{bit-query}}(\varepsilon, \text{INT}_{F_c}) = O(\varepsilon^{-d/(d+1)}).$$

Consider now any quantum algorithm that solves the integration problem with error  $\varepsilon$  and probability at least  $\frac{3}{4}$ , using  $k$  bit queries.

Let  $h(x_1, \dots, x_d) = \alpha \prod_{j=1}^d h_j(x_j)$  for  $(x_1, \dots, x_d) \in I_d$ , where  $h_j(x) = x^2(1-x)^2$ ,  $x \in [0, 1]$ , and  $h(x_1, \dots, x_d) = 0$  for  $(x_1, \dots, x_d) \in \mathbb{R}^d \setminus I_d$ . Here,  $\alpha$  is a constant such that  $h \in F_1$ , where  $F_1$  is defined by (15) with  $c = 1$ . For each  $j = 1, \dots, d$  and  $i = 0, \dots, n-1$ , let  $h_{i,j}(x) = h_j(n(x - i/n))$ . Then the support of  $h_{i,j}$  is  $[i/n, (i+1)/n]$ .

We obtain  $n^d$  functions on  $I_d$ . Each function is defined by

$$h_{i_1, \dots, i_d}(x_1, \dots, x_d) = \frac{\alpha}{n} \prod_{j=1}^d h_{i_j, j}(x_j),$$

and its support is the cube  $\prod_{j=1}^d [i_j/n, (i_j+1)/n]$ ,  $i_j = 0, \dots, n-1$ .

For notational convenience we re-index these functions, in any desirable way, and denote them by  $g_\ell$ ,  $\ell = 0, \dots, n^d - 1$  (i.e.,  $g_\ell = h_{i_1, \dots, i_d}$ ). Thus  $\|g_\ell\|_\infty \leq n^{-1}$  and assuming that  $c \gg n^{-1}$  we have  $g_\ell \in F_c$ . Then

$$\int_{I_d} g_\ell(x) dx = n^{-d-1} \int_{I_d} h(x) dx, \quad \ell = 0, \dots, n^d - 1.$$

Consider now any boolean function  $B : \{0, 1, \dots, n^d - 1\} \rightarrow \{0, 1\}$  and define the function

$$f_B(x) = \sum_{\ell=0}^{n^d-1} B(\ell)g_\ell(x), \quad x \in I_d.$$

Then  $f_B \in F_c$  and

$$\int_{I_d} f_B(x) dx = \frac{\int_{I_d} h(x) dx}{n} \frac{1}{n^d} \sum_{\ell=0}^{n^d-1} B(\ell).$$

Thus, computing the Boolean mean is reduced to computing the integral of  $f_B$ . From [18] we know that  $k < n^d$  bit queries yield error  $\Omega(k^{-1})$  in the approximation of the Boolean mean. Therefore, by setting  $k = \beta n^d$ ,  $\beta \in (0, 1)$ , we obtain that the error in approximating the integral of  $f_B$  is  $\Omega(n^{-(d+1)})$ . Hence, for error  $\varepsilon$  we need  $k = \Omega(\varepsilon^{-d/(d+1)})$  bit queries. Using the upper bound of [20] we obtain

$$n^{\text{bit-query}}(\varepsilon, \text{INT}_{F_c}) = \Theta(\varepsilon^{-d/(d+1)}).$$

This complexity bound remains valid if  $c$  depends on  $\varepsilon$  and  $c(\varepsilon) \rightarrow 0$ , as  $\varepsilon \rightarrow 0$ , but not very fast. Therefore, when  $c(\varepsilon)\varepsilon^{-1/(d+1)} \rightarrow \infty$  as  $\varepsilon \rightarrow 0$ , the bit query complexity for integration in the class  $F_{c(\varepsilon)}$  is  $\Theta(\varepsilon^{-d/(d+1)})$ .

Now that we have the bit query complexity for integration, we reduce the eigenvalue problem to integration and obtain a lower bound for the bit query complexity of the eigenvalue problem. This is done in exactly the same way as for classical deterministic and randomized algorithms. In particular, using equation (14) we see that any algorithm approximating  $\lambda(\tilde{q})$  can be used to derive an algorithm that solves the integration problem  $S(f)$  defined in (16), and  $f = \tilde{q} + \delta - 1/2$  belongs to the class  $F_c$  (15). We omit the details since we have already presented this argument twice. Therefore, solving the eigenvalue problem with error  $\varepsilon$  and probability at least  $\frac{3}{4}$  implies that we can solve the integration problem with error  $O(\varepsilon)$  and probability at least  $\frac{3}{4}$ . Consequently the bit query complexity  $n^{\text{bit-query}}(\varepsilon)$  of the eigenvalue problem is at least as large as the bit query complexity of the integration problem. We have proved the following theorem.

**Theorem 4.1.**

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

To derive a quantum algorithm for the eigenvalue problem we can slightly modify the randomized algorithm we presented previously. The third and last step of the randomized algorithm approximates a weighted integral using Monte Carlo. The quantum algorithm will

approximate that integral using the amplitude amplification algorithm [7]. In particular the quantum algorithm approximates the first two terms on the right hand side of equation (20) by

$$\tilde{\lambda}(q) := \hat{\lambda}(\bar{q}) + \phi((q - \bar{q})\hat{u}_{\bar{q}}^2), \quad (35)$$

where, just as before,  $\bar{q}$  approximates  $q$  with error  $O(n^{-1})$ ,  $\hat{\lambda}(\bar{q}) := \hat{\lambda}(M_h(\bar{q}))$ ,  $h = (m+1)^{-1}$ , while  $\phi((q - \bar{q})\hat{u}_{\bar{q}}^2)$  is the result of the amplitude amplification algorithm with  $T$  bit queries as applied in [20] for the approximation of the integral of  $(q - \bar{q})\hat{u}_{\bar{q}}^2$  in (20).

Since  $\|(q - \bar{q})\hat{u}_{\bar{q}}^2\|_\infty = O(n^{-1})$ , with probability  $\frac{3}{4}$  the error of (35) is bounded from above by

$$|\lambda(\bar{q}) - \hat{\lambda}(\bar{q})| + O((nT)^{-1}) + \|u_{\bar{q}} - \hat{u}_{\bar{q}}\|_{L_2} O(n^{-1}) + O(n^{-2}),$$

where the second term is the error of the quantum algorithm  $\phi$ ; see also (34). We have seen that  $|\lambda(\bar{q}) - \hat{\lambda}(\bar{q})| = O(m^{-1})$  and  $\|u_{\bar{q}} - \hat{u}_{\bar{q}}\|_{L_2} = O(m^{-1/2})$ . This yields an error proportional to

$$m^{-1} + (nT)^{-1} + n^{-1}m^{-1/2} + n^{-2}.$$

The algorithm uses  $n^d$  evaluations of  $q$  at deterministic points, plus a number of classical operations proportional to  $n^d + m^d \log m$  times a constant that depends on  $d$ . The algorithm also uses  $T$  bit queries involving  $q$ , plus of order  $\log^2 T + d \log m$  quantum operations, excluding the cost of queries, for the details see [7, 19]. Note that  $\log^2 T$  operations are sufficient for the quantum implementation of the Fourier transform used in the amplitude amplification algorithm. The number of qubits is of order  $\log T + d \log m$ .

Setting  $m^{-1} = \varepsilon^2$  and  $T = O(n^d)$ , we get that the error of our algorithm is bounded from above by a quantity proportional to

$$\varepsilon^2 + n^{-(d+1)} + n^{-1}\varepsilon + n^{-2}.$$

Note that when  $d \geq 2$  we do not necessarily have to take as many as  $O(n^d)$  queries, since reducing the integration error does not reduce the upper bound of the algorithm error which still depends on  $n^{-2}$ . However, taking  $T = O(n^d)$  does not change the order of magnitude of the cost of the algorithm. The dominant component of the cost of the algorithm is the  $n^d$  classical function evaluations required for the approximation of  $q$  by  $\bar{q}$ . Finally, setting  $n = \varepsilon^{-1/2}$  yields error  $O(\varepsilon)$ .

**Theorem 4.2.** *The eigenvalue problem can be solved with probability  $\frac{3}{4}$  and error  $O(\varepsilon)$  by the hybrid algorithm (35). This algorithm uses*

- $\varepsilon^{-d/2}$  classical function evaluations,
- $\varepsilon^{-2d} \log \varepsilon^{-1}$  (times a constant that depends on  $d$ ) classical arithmetic operations,
- $\varepsilon^{-d/2}$  bit queries,
- $d^2 \log^2 \varepsilon^{-1}$  (times a constant independent of  $d$ ) quantum operations, excluding queries, (mostly used for the quantum implementation of the Fourier transform,)
- and a number of qubits proportional to  $d \log \varepsilon^{-1}$

We see that the number of bit queries used by this algorithm matches the bit query complexity only when  $d = 1$ . Perhaps, as in the case of the randomized algorithm, we can improve this situation and obtain matching upper and lower bounds for the number of bit queries using a perturbation formula with higher order terms. This is an open question at this time. Nevertheless, even if this question has a positive answer, the number of arithmetic operations will remain exponential in  $d$ , and this is also true for the deterministic and randomized algorithms we have seen, because all of them solve a matrix eigenvalue problem and the size of the matrix is exponential in  $d$ .

We can solve the eigenvalue problem with cost (number of queries plus other operations) that is not exponential in  $d$  using a quantum algorithm without any classical components. The details of the algorithm will become apparent after we discuss, in the next section, a quantum algorithm that solves the eigenvalue problem using a different type of queries, called power queries. This algorithm is based on phase estimation [19], a quantum algorithm approximating an eigenvalue of a Hermitian matrix, which solves the problem with  $O(\log \varepsilon^{-1})$  power queries. Each of the power queries can be approximated by bit queries using the Trotter formula [19] and phase kick-back [8]. The number of bit queries required for the approximation of each power query is a polynomial in  $\varepsilon^{-1}$  and its degree is independent of  $d$ . In particular, the degree of this polynomial only depends on the norm of the matrix whose eigenvalue is sought, which is independent of  $d$ , and on the accuracy demand  $\varepsilon$ . We have the following theorem whose proof we postpone to the next section.

**Theorem 4.3.** *Phase estimation applied for the approximation of the smallest eigenvalue of  $M_\varepsilon(q)$  achieves error  $O(\varepsilon)$  with probability at least  $\frac{3}{4}$  using a number of bit queries proportional to  $\varepsilon^{-6} \log^2 \varepsilon^{-1}$ . The initial state for phase estimation is the eigenvector of  $M_\varepsilon(0) = -\Delta_\varepsilon$  that corresponds to its smallest eigenvalue. The algorithm uses a number of quantum operations, excluding bit queries, proportional to  $d\varepsilon^{-6} \log^4 \varepsilon^{-1}$  and a number of qubits proportional to  $d \log \varepsilon^{-1}$ . Consequently,*

$$n^{\text{bit-query}}(\varepsilon) = O(\varepsilon^{-6} \log^2 \varepsilon^{-1}).$$

## 4.2 Power Queries

In this section, we consider *power* queries as they have been described in [21]. For some problems, a quantum algorithm can be written in the form

$$|\psi\rangle := U_T \widetilde{W}_T U_{T-1} \widetilde{W}_{T-1} \cdots U_1 \widetilde{W}_1 U_0 |\psi_0\rangle. \quad (36)$$

Here  $U_1, \dots, U_T$  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 [19, p. 178]. Then  $W_j = W^{p_j}$  for an  $n \times n$  unitary matrix  $W$  that depends on the input of the computational problem, and for some non-negative integers  $p_1, \dots, p_T$ . Without loss of generality we assume that  $n$  is a power of two. Let  $\{|y_k\rangle\}$  be orthonormalized eigenvectors of  $W$ , so that  $W|y_k\rangle = \alpha_k|y_k\rangle$  with the corresponding eigenvalue  $\alpha_k$ , where  $|\alpha_k| = 1$  and  $\alpha_k = e^{i\lambda_k}$  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

$$\widetilde{W}_j |x_1\rangle|x_2\rangle \cdots |x_r\rangle|y_k\rangle = |x_1\rangle \cdots |x_{j-1}\rangle \left( \alpha_j|0\rangle + \beta_j e^{ip_j \lambda_k} |1\rangle \right) |x_{j+1}\rangle \cdots |x_r\rangle|y_k\rangle. \quad (37)$$

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  $e^{i\lambda_k}$ .

$\widetilde{W}_j$  is called a *power query* since it is derived from powers of  $W$ . Power queries have been successfully used for a number of problems including the phase estimation problem, see [8, 19]. The phase estimation algorithm approximates an eigenvalue of a unitary operator  $W$  using a good approximation [1] of the corresponding eigenvector 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 our eigenvalue problem, we discretize the operator  $\mathbb{L}_q$  on a grid with mesh size  $h$ , as we did when we were discussing deterministic algorithms. We obtain an  $m^d \times m^d$  matrix  $M_h(q)$ , with  $h = (m + 1)^{-1}$ , that is symmetric positive definite. Then we define the matrix

$$W = \exp(i\gamma M_h(q)) \quad \text{with } i = \sqrt{-1} \text{ and a positive } \gamma, \quad (38)$$

which is unitary since  $M_h(q)$  is symmetric.

Using the powers of  $W$  we obtain the matrices  $\widetilde{W}_j$  used in (36). Accordingly, we modify the query definition in equation (31) by assuming, as in [19, Ch. 5], that for each  $j$  the  $\widetilde{W}_j$  is one quantum query. Hence for algorithms that can be expressed in the form (36), the number of power queries is  $T$ , independently of the powers  $p_j$ .

With the understanding that the number of queries  $T$  is defined differently in this section than before the error  $e^{\text{quant}}(\hat{\lambda}, T)$  of the algorithm (36) is given by (32). Similarly, the power query complexity  $n^{\text{power-query}}(\varepsilon)$  is defined by (33).

We now exhibit a quantum algorithm with power queries that approximates  $\lambda(q)$  with error  $O(\varepsilon)$ . Consider  $W$  defined by (38) with  $\gamma = 1/(2d)$ , i.e.,

$$W = \exp\left(\frac{1}{2d} i M_h(q)\right). \quad (39)$$

The eigenvalues of  $W$  are  $e^{i\lambda_j(M_h(q))/(2d)}$ , with  $\lambda_j(M_h(q))$  being the eigenvalues of the  $m^d \times m^d$  matrix  $M_h(q)$ . Without loss of generality we assume that  $m$  is a power of two. These eigenvalues can be written as  $e^{2\pi i \varphi_j}$ , where

$$\varphi_j = \varphi_j(M_h(q)) = \frac{1}{4d\pi} \lambda_j(M_h(q))$$

are called *phases*. We are interested in estimating the smallest phase  $\varphi_1(M_q)$ , which belongs to  $(0, 1)$  since  $\lambda_1(M_h(q)) \in [d\pi^2, d\pi^2 + 1]$ . We denote the eigenvector of  $M_h(q)$  and  $W$  that corresponds to  $\lambda_j(M_h(q))$  by  $z_j(M_h(q))$ , with  $\|z_j(M_h(q))\|_2 = 1$ ,  $j = 1, \dots, m^d$ , indexed in non-decreasing order of eigenvalues.

Phase estimation, see [19, Section 5.2], is a quantum algorithm that approximates the phase  $\varphi_1(M_q)$ . Clearly, to compute an  $\varepsilon$ -approximation of  $\lambda_1(M_h(q))$ , it is enough to compute an  $\varepsilon/(4d\pi)$ -approximation of  $\varphi_1(M_h(q))$ . The initial state of phase estimation algorithm is  $|0\rangle^{\otimes b} |z_1\rangle$ , where  $b$  is related to the accuracy of the algorithm and will be determined later, while  $|z_1\rangle = |z_1(M_h(q))\rangle$ . It is helpful to think of two registers holding the initial state. The

top register is  $b$  qubits long and holds  $|0\rangle^{\otimes b}$ , while the bottom register holds the eigenvector  $|z_1\rangle$ .

Abrams and Lloyd [1] showed that phase estimation can still be used even if the eigenvector  $|z_1\rangle$  is replaced by a good approximation  $|\psi\rangle$ . More precisely, expanding  $|\psi\rangle$  in the basis of the eigenvectors  $|z_j\rangle$ , the initial state takes the form

$$|0\rangle^{\otimes b}|\psi\rangle = |0\rangle^{\otimes b} \sum_{k=0}^{m^d-1} d_k |z_k\rangle.$$

The success probability of the algorithm depends on  $|d_1|^2$ , the square of the projection of  $|\psi\rangle$  onto  $|z_1\rangle$ .

Omitting the details, which are not important in the analysis here and can be found in [1, 21], a measurement of the top register of the final state of phase estimation, with probability at least

$$\frac{8}{\pi^2}|d_1|^2,$$

will produce an index  $j \in [0, 2^b - 1]$  such that

$$\left| \frac{j}{2^b} - \varphi_1(M_h(q)) \right| \leq \frac{1}{2^b}.$$

The cost of phase estimation is equal to  $b$  power queries, plus a number of operations proportional to  $b^2 + d \log m$ , plus the cost for preparing the initial state  $|\psi\rangle$ . The number of qubits used is  $b + d \log m$ . We remark that the  $O(b^2)$  operations are for the quantum implementation of the (inverse) Fourier transform used in phase estimation [19].

Taking into account that the matrix eigenvalue approximates  $\lambda(q)$  with error  $O(h)$ , where  $h = (m + 1)^{-1}$ , we obtain that

$$\left| \lambda(q) - \frac{4dj\pi}{2^b} \right| \leq \frac{4d\pi}{2^b} + O(h).$$

Therefore, it suffices to set  $h = \varepsilon$  and  $b = \log \varepsilon^{-1}$  to obtain error  $O(\varepsilon)$  in the approximation of  $\lambda(q)$ . Under these conditions, the cost of the algorithm is equal to  $\log \varepsilon^{-1}$  power queries, plus a number of operations proportional to  $\log^2 \varepsilon^{-1}$ , plus the cost of preparing the initial state. Recall that we want to implement a good approximation  $|\psi\rangle$  of  $|z_1\rangle$  leading to success probability at least  $\frac{3}{4}$ .

Consider the eigenvector corresponding to the smallest eigenvalue when  $q = 0$ . Denote this eigenvector by  $|z_1(M_\varepsilon(0))\rangle$ ,  $M_\varepsilon(0) = -\Delta_\varepsilon$ . Then  $|z_1(M_\varepsilon(0))\rangle = |z_1^{(1)}\rangle^{\otimes d}$ , i.e.,  $|z_1(M_\varepsilon(0))\rangle$  is the tensor product of the eigenvectors of the  $\varepsilon^{-1} \times \varepsilon^{-1}$  matrix of the corresponding one-dimensional problem (i.e., when  $d = 1$ ) [12]. Each  $|z_1^{(1)}\rangle$  can be implemented using the Fourier transform with a number of operations proportional to  $\log^2 \varepsilon^{-1}$ , see [19, p. 209] and [9, 28] for more details. Therefore, we can implement  $|z_1(M_\varepsilon(0))\rangle$  with cost proportional to  $d \log^2 \varepsilon^{-1}$ .

Now consider any  $q \in \mathbf{Q}$ . Since we know that the eigenvalues of  $M_\varepsilon(q)$  are well separated (18), using  $|z_1(M_\varepsilon(0))\rangle$  as an approximate eigenvector we find that the square of its projection

onto  $z_1(M_\varepsilon(q))$  satisfies [29, p. 173]

$$|\langle z_1(M_\varepsilon(q)) | z_1(M_\varepsilon(0)) \rangle|^2 \geq 1 - \frac{1}{(3\pi^2 - 2)^2}.$$

Define the initial state of phase estimation using

$$|\psi\rangle := |z_1(M_\varepsilon(0))\rangle$$

to obtain that  $|d_1|^2 \geq 1 - (3\pi^2 - 2)^{-2}$  which leads to a success probability

$$\frac{8}{\pi^2} |d_1|^2 \geq \frac{8}{\pi^2} \left( 1 - \frac{1}{(3\pi^2 - 2)^2} \right) \geq \frac{3}{4}.$$

We have proved the following theorem.

**Theorem 4.4.** *The eigenvalue problem can be solved with error  $O(\varepsilon)$ , and probability at least  $\frac{3}{4}$ , by discretizing  $\mathbb{L}_q$  and then approximating the smallest eigenvalue of the resulting matrix  $M_\varepsilon(q)$  by phase estimation that uses power queries. The initial state of phase estimation uses the eigenvector of  $M_\varepsilon(0) = -\Delta_\varepsilon$  that corresponds to its smallest eigenvalue. The cost of the algorithm is proportional to*

- $\log(\varepsilon^{-1})$  power queries,
- $\log^2 \varepsilon^{-1} + d \log \varepsilon^{-1}$  quantum operations,
- $d \log \varepsilon^{-1}$  qubits.

Let us now turn to the query complexity  $n^{\text{power-query}}(\varepsilon)$ . The previous theorem implies that  $n^{\text{power-query}}(\varepsilon) = O(\log \varepsilon^{-1})$ . Consider a function  $q \in \mathbf{Q}$  such that  $q(x_1, \dots, x_d) = \sum_{j=1}^d g(x_j)$ , where  $g \in C^1([0, 1])$  is non-negative and  $\|g\|_\infty \leq 1$  and  $\|g'\|_\infty \leq 1$ . Then [23, p. 113] the eigenvalue problem (1), (2) has a separable solution which is obtained by solving the Sturm-Liouville eigenvalue problem

$$\begin{aligned} -y''(x) + g(x)y(x) &= \mu y(x), & x \in (0, 1) \\ y(0) &= y(1) = 0. \end{aligned}$$

Denoting the smallest eigenvalue of this problem by  $\mu(g)$  we have

$$\lambda(q) = d \mu(g).$$

Any algorithm that approximates  $\lambda(q)$  with error  $O(\varepsilon)$  also approximates  $\mu(g)$  with error  $O(\varepsilon)$ . Using the power query lower bound for the Sturm-Liouville eigenvalue problem [5, 6], we conclude any quantum algorithm with power queries that approximates  $\lambda(q)$  with error  $O(\varepsilon)$  must use  $\Omega(\log \varepsilon^{-1})$  queries.

Combining the lower bound with the previous theorem leads to tight power query complexity bounds.



**Theorem 4.5.**

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

We are now ready to prove the upper bound for the bit-query complexity of Theorem 4.3. **Proof of Theorem 4.3.** We use phase estimation as in the proof of Theorem 4.4 but instead of power queries we will use bit queries to approximate them. Recall equation (39), with  $h = \varepsilon$ . The matrix  $M_\varepsilon(q)$  has size  $m^d \times m^d$  with  $(m+1)^{-1} = \varepsilon$ . Its largest eigenvalue does not exceed  $4d\varepsilon^{-2} + 1$  [12, p. 268]. Therefore, we have  $\|(2d)^{-1}M_\varepsilon(q)\|_2 \leq (4d\varepsilon^{-2} + 1)/(2d)$ . For  $\beta = 4d\varepsilon^{-2} + 1$  we have  $\|(2d\beta)^{-1}M_\varepsilon(q)\|_2 \leq 1$ . Recall that  $(2d\beta)^{-1}M_\varepsilon(q) = -(2d\beta)^{-1}\Delta_\varepsilon + (2d\beta)^{-1}B_\varepsilon(q)$ . For notational convenience define  $A_1 = -(2d\beta)^{-1}\Delta_\varepsilon$  and  $A_2 = (2d\beta)^{-1}B_\varepsilon(q)$ . Then  $\|A_1\|_2 \leq 1$  and  $\|A_2\|_2 \leq 1$ .

Using the Trotter formula [19, p. 208] we have

$$\left\| e^{i(A_1+A_2)/k} - e^{iA_1/k}e^{iA_2/k} \right\|_2 \leq ck^{-2},$$

where  $c$  is a constant, (see also [17, 22] and the references therein). From (39) we have

$$W^L = e^{i(A_1+A_2)\beta L} \quad \text{for any } L \in \mathbb{N}$$

and therefore

$$\left\| W^L - \left( e^{iA_1/k}e^{iA_2/k} \right)^{k\beta L} \right\|_2 \leq c \frac{\beta L}{k}. \quad (40)$$

In phase estimation we require the maximum power of  $W$  to be of order  $\varepsilon^{-1}$ . Setting  $L = O(\varepsilon^{-1})$  in the equation above, we have that  $\beta L$  is of order  $\varepsilon^{-3}$ . Thus for  $k$  proportional to  $\varepsilon^{-3} \log^2 \varepsilon^{-1}$ , the error in the approximation of the matrix exponential (40) is  $O(\log^{-2} \varepsilon^{-1})$ .

From [8] we know that using bit queries and phase kick-back we can obtain  $e^{iA_2/k}$ . Hence, to approximate the  $O(\log \varepsilon^{-1})$  power queries of phase estimation the algorithm we need a total number of bit queries proportional to  $\varepsilon^{-6} \log^2 \varepsilon^{-1}$ .

Since the eigenvalues and eigenvectors of  $-\Delta_\varepsilon$  are known, each of the  $e^{iA_1/k}$  can be implemented using the quantum Fourier transform with a number of quantum operations proportional to  $d \log^2 \varepsilon^{-1}$ . Thus the total number of quantum operations, excluding bit queries, required to approximate all the power queries is proportional to  $d \varepsilon^{-6} \log^4 \varepsilon^{-1}$ .

Using (40) to approximate the power queries only changes the success probability of phase estimation [19, p. 195]. Since phase estimation uses of order  $\log \varepsilon^{-1}$  power queries and each is approximated with error  $O(\log^{-2} \varepsilon^{-1})$  the success probability may be reduced by a quantity proportional to  $\log^{-1} \varepsilon^{-1}$ . Therefore for  $\varepsilon$  sufficiently small, the probability remains greater than or equal to  $\frac{3}{4}$ . ■

We conclude by addressing the qubit complexity of our problem. By qubit complexity we mean the minimum number of qubits required for a quantum algorithm to achieve error  $\varepsilon$ . We denote the qubit complexity by  $n^{\text{qubit}}(\varepsilon)$ . The qubit complexity is related to the classical information complexity  $n^{\text{wor}}(\varepsilon)$  by

$$n^{\text{qubit}}(\varepsilon) = \Omega(\log n^{\text{wor}}(\varepsilon)).$$

This is shown in [30] and it holds regardless of the type of queries used. Since,  $n^{\text{wor}}(\varepsilon) = \Omega(\varepsilon^{-d})$  we get

$$n^{\text{qubit}}(\varepsilon) = \Omega(\log \varepsilon^{-1}).$$

On the other hand, phase estimation solves the problem with error  $O(\varepsilon)$  using a number of qubits proportional to  $d \log \varepsilon^{-1}$ . We have proved the following theorem.

**Theorem 4.6.**

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

## 5 Acknowledgements

I am very grateful to A. Bessen for the extensive discussions we had and his insightful remarks that significantly improved this paper. I thank J. H. Lai, J. F. Traub and A. G. Werschulz for their comments and suggestions.

## 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, R., Mosca, R. and de Wolf, R. (1998), *Quantum lower bounds by polynomials*, Proceedings FOCS'98, 352–361. Also <http://arXiv.org/quant-ph/9802049>.
- [4] Bernstein, E., and Vazirani, U. (1997), *Quantum complexity theory*, SIAM J. Computing, 26(5), 1411–1473.
- [5] Bessen, A. J. (2005), *A lower bound for phase estimation*, Physical Review A, 71(4), 042313. Also <http://arXiv.org/quant-ph/0412008>.
- [6] Bessen, A. J. (2006), *A lower bound for the Sturm-Liouville eigenvalue problem on a quantum computer*, Journal of Complexity, 22(5), 660–675. Also <http://arXiv.org/quant-ph/04512109>
- [7] 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>.
- [8] Cleve, R., Ekert, A., Macchiavello, C. and Mosca, M. (1998), *Quantum Algorithms Revisited*, Proc. R. Soc. Lond. A, 454, 339–354.

- [9] Klappenecker, A. and Rötteler, M. (2001), *Discrete Cosine Transforms on Quantum Computers*, <http://arXiv.org/quant-ph/0111038>.
- [10] Courant, R. (1943), *Variational Methods for the Solution of Problems of Equilibrium and Variations*, Bulletin American Mathematical Society, 49, 1–23.
- [11] Courant, C. and Hilbert, D. (1989), *Methods of Mathematical Physics, Vol. I*, Wiley Classics Library, Wiley-Interscience, New York.
- [12] Demmel, J. W. (1997), *Applied Numerical Linear Algebra*, SIAM, Philadelphia.
- [13] Forsythe, G. E., and Wasow, W. R. (2004), *Finite-Difference Methods for Partial Differential Equations*, Dover, New York.
- [14] 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>.
- [15] Heinrich, S. (2002), *Quantum Summation with an Application to Integration*, J. Complexity, 18(1), 1–50. Also <http://arXiv.org/quant-ph/0105116>.
- [16] Heinrich, S. (2003), *Quantum integration in Sobolev spaces*, J. Complexity, 19, 19–42.
- [17] Jahnke, T. and Lubich, C. (2000), Error bounds for exponential operator splitting, BIT, 40(4), 735–744.
- [18] A. Nayak and F. Wu (1999), The quantum query complexity of approximating the median and related statistics, Proceedings of the 31st Annual ACM Symposium on the Theory of Computing (STOC), 384–393. LANL preprint [quant-ph/9804066](http://arXiv.org/quant-ph/9804066).
- [19] Nielsen, M.A. and Chuang, I.L. (2000), *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK.
- [20] Novak, E. (2001), *Quantum complexity of integration*, J. Complexity, 17, 2–16. Also <http://arXiv.org/quant-ph/0008124>.
- [21] Papageorgiou, A. and Woźniakowski, H. (2005), *Classical and Quantum Complexity of the Sturm-Liouville Eigenvalue Problem*, Quantum Information Processing, 4, 87–127. Also <http://arXiv.org/quant-ph/0502054>.
- [22] Suzuki, M. (1992), *General theory of higher-order decomposition of exponential operators and symplectic integrators*, Physics Letters A, 165, 387–395.
- [23] Titchmarsh, E. C. (1958), *Eigenfunction Expansions Associated with Second-Order Differential Equations, Part B*, Oxford University Press, Oxford, UK.
- [24] Traub, J. F., Wasilkowski, G. W. and Woźniakowski, H. (1988), *Information-Based Complexity*, Academic Press, New York.
- [25] Traub, J. F. (1999), *A continuous model of computation*, Physics Today, May, 39–43.

- [26] Weinberger, H. F. (1956), *Upper and Lower Bounds for Eigenvalues by Finite Difference Methods*, Communications on Pure and Applied Mathematics, IX, 613–623.
- [27] Weinberger, H. F. (1958), *Lower Bounds for Higher Eigenvalues by Finite Difference Methods*, Pacific Journal of Mathematics, 8(2), 339–368.
- [28] Wickerhauser, M. V. (1994), *Adapted Wavelet Analysis from Theory to Software*, A. K. Peters, Wellesley.
- [29] Wilkinson, J. H. (1965), *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK.
- [30] Woźniakowski, H. (2006), *The Quantum Setting with Randomized Queries for Continuous Problems*, Quantum Information Processing, 5(2), 83–130. Also <http://arXiv.org/quant-ph/060196>.