Eigenpath Traversal by Poisson-Distributed Phase Randomisation

Joseph Cunningham ✉ 🆔
Centre for Quantum Information and Communication (QuIC), Ecole polytechnique de Bruxelles, Université libre de Bruxelles, Belgium

Jérémie Roland ✉ 🆔
Centre for Quantum Information and Communication (QuIC), Ecole polytechnique de Bruxelles, Université libre de Bruxelles, Belgium

―――― **Abstract** ――――

We present a framework for quantum computation, similar to Adiabatic Quantum Computation (AQC), that is based on the quantum Zeno effect. By performing randomised dephasing operations at intervals determined by a Poisson process, we are able to track the eigenspace associated to a particular eigenvalue.

We derive a simple differential equation for the fidelity, leading to general theorems bounding the time complexity of a whole class of algorithms. We also use eigenstate filtering to optimise the scaling of the complexity in the error tolerance ϵ.

In many cases the bounds given by our general theorems are optimal, giving a time complexity of $O(1/\Delta_m)$ with Δ_m the minimum of the gap. This allows us to prove optimal results using very general features of problems, minimising the problem-specific insight necessary.

As two applications of our framework, we obtain optimal scaling for the Grover problem (i.e. $O(\sqrt{N})$ where N is the database size) and the Quantum Linear System Problem (i.e. $O(\kappa \log(1/\epsilon))$ where κ is the condition number and ϵ the error tolerance) by direct applications of our theorems.

1 Introduction

It has long been appreciated that the ability to prepare a ground state of a given Hamiltonian is useful for a large number computational tasks. Many NP-hard problems, including various types of partitioning, covering, and satisfiability problems, can be solved by finding the ground state of an Ising system [14]. There are also many applications in the fields of quantum chemistry, where finding the ground state of molecules is a common task, and physics, where knowledge of the ground state helps to understand low-temperature phenomena such as superconductivity and superfluidity.

For computational problems we have the following strategy: (1) find a physical system such that the ground state encodes useful information for solving the problem, (2) prepare the ground state using some physical process and (3) use the information contained in the ground state to solve the problem. This paper is about performing the second step of this strategy.

The most famous way to perform the second step is known as Adiabatic Quantum Computation (AQC) [8]. Suppose H_P is the Hamiltonian whose ground state is of interest. This procedure requires a second Hamiltonian, H_0, with an easily preparable ground state.

19th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2024).
Editors: Frédéric Magniez and Alex Bredariol Grilo; Article No. 7; pp. 7:1–7:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Now consider the following interpolated Hamiltonian: $H(s) = (1 - s)H_0 + sH_P$ and pick a large time T. We start with the system in the ground state of H_0 and evolve according to the time-dependent Hamiltonian $H(t/T)$, for time $t \in [0, T]$. The adiabatic theorem says that if T is large enough, then the resulting state will be close to the ground state of $H(1) = H_P$. See [11] for results detailing how large T has to be. Clearly we want to take T as small as possible, since a larger T means our computation takes longer.

While AQC is polynomially equivalent to the quantum circuit model [1], it suffers from a few drawbacks. The most significant one being that it requires the system to evolve under a very specific time-dependent Hamiltonian. It is typically very hard to physically implement a system that evolves under exactly this Hamiltonian. Often the complicated time-dependent dynamics are approximated by a sequence of simpler evolutions, which introduces discretisation error. In particular this is necessary when implementing AQC on a conventional quantum computer. Bounding the discretisation error analytically is typically hard to do. In contrast, our method only requires the evolution under a finite number of time-independent Hamiltonians for finite time and thus has no discretisation cost.

There exist alternatives to AQC that are also based on an interpolation $H(s)$ between a Hamiltonian whose ground state is easy to prepare and one whose ground state is difficult to prepare. These approaches use alternate ways to transform the ground state of $H(0)$ into that of $H(1)$, or, more generally some eigenstate of $H(0)$ into the corresponding eigenstate of $H(1)$. They often make use of a variant of the quantum Zeno effect. For instance [5] uses measurement and [4] simulates the quantum Zeno effect by applying Hamiltonian evolutions for random amounts of time in a procedure known as the Randomisation Method (RM).

Our framework builds on the RM of [4] in the following way: instead of performing a fixed sequence of phase randomising steps, we stochastically choose when to perform phase randomisation, based on a Poisson process with rate $\lambda(s)$, see algorithm 1. This has a number of advantages. Firstly it yields a simple differential equation for the state evolution, which fits in the general framework of non-unitary adiabatic theorems of [3], and greatly simplifies the analysis. It allows us to obtain general theorems, see in particular theorems 4 and 5, that in many cases yield optimal results with minimal extra work or problem-specific insight. Also, we only need very minimal technical assumptions on $H(s)$: it only needs to be twice continuously differentiable and we need to know some estimate of the gap between the eigenvalue of interest and the rest of the spectrum. We do not assume precise knowledge of the spectrum or gap. We allow the eigenspace of interest to be degenerate.

Our theorem 4 deals with the case where the rate of the Poisson process λ is taken to be constant. The result we obtain is better than the corresponding result for AQC with a constant-speed linear interpolation. In theorem 5 we describe a variable $\lambda(s)$ that can significantly improve the time complexity, up to $O(1/\Delta_m)$ in the minimum gap Δ_m. Finally theorem 7 improves the dependence of the time complexity on the error tolerance. Typically algorithms based on AQC and the RM have a complexity that scales as $O(1/\epsilon)$ in the error tolerance ϵ. Eigenstate filtering, introduced in [13], can be used to reduce this to $O(\log(1/\epsilon))$. This has been used before to RM-inspired algorithms that use the circuit model, see [13] and [12], but we provide a version native to our cost model.

From theorem 5, we see that the following property is very useful to obtain fast algorithms: $\int_0^1 \frac{1}{\Delta(s)^p} \, ds = O(\Delta_m^{1-p})$, where Δ is the gap, $\Delta_m = \inf_{s \in [0,1]} \Delta$ and $p > 1$. This property seems to be quite generic, in particular it holds for both the Grover search problem and the Quantum Linear System Problem (QLSP).

In the Grover search problem, [9], the goal is to prepare a specific state in an N-dimensional space with the help of an oracle. It is well-known that this can be done in $O(\sqrt{N})$ queries to the oracle. When AQC was first used to tackle this problem, a complexity of $O(N)$

was obtained [8]. The trick to achieving a complexity of $O(\sqrt{N})$ was to use an adapted schedule [17, 20]. In our framework, the algorithm using constant λ already achieves a scaling of $O(\sqrt{N}\log(N))$ by theorem 4, which is significantly better than the corresponding case for AQC. Using a variable $\lambda(s)$, we recover the scaling $O(\sqrt{N})$ by theorem 5. It is interesting to note that, by the generality of our theorems, we actually obtain a whole family of schedules, parametrised by some value $0 < q < 1$, that solve the problem optimally. This is analogous to the range of adiabatic schedules considered in [2] and [7]. The original schedule of [17] actually corresponds to a choice of $q = 1$. It seems like the RM can be considered as the $q = 1$ case of a family of methods, at least in the case of linear interpolation. This falls outside the range of our theorem, but it turns out that $q = 1$ is in fact good enough to give optimality for the Grover problem, which explains why the RM was already known to be able to perform Grover search with optimal complexity $O(\sqrt{N})$, [4].

In general, for other problems, $q = 1$ does not give optimal scaling. The Quantum Linear System Problems (QSLP) is an example of a problem where $q = 1$ does not work, neither in our framework, nor in AQC [2]. In QLSP, [10], the goal is to prepare a quantum state $|x\rangle$ that is proportional to the solution of a system of linear equations $Ax = b$. In [19] the randomisation method was used to construct an algorithm with complexity $O(\kappa \log(\kappa)/\epsilon)$, where κ is the condition number and ϵ the error tolerance. This is improved to $O(\kappa \log(\kappa/\epsilon))$ in [12]. In [7] an algorithm based on a discrete adiabatic theorem was proposed which scales as $O(\kappa \log(1/\epsilon))$. We are able to match this in our framework.

There has actually been some discussion recently on the merits of these two approaches to QLSP, i.e. the approach based on the RM of [19] and the approach based on the discrete adiabatic theorem [7]. The approach based on the discrete adiabatic theorem has the better asymptotic scaling, but it turns out that the proven complexity for reasonable values of κ is very large. The paper [12] presents an algorithm that is based on the RM and has a better proven complexity for reasonable values of κ, but is asymptotically suboptimal. Finally [6] uses numerical methods to determine the actual performance of the algorithm based on the discrete adiabatic theorem. They claim that it works much better than the proven bound and in fact better than the algorithm based on the RM. We can contribute to this discussion by noting that our framework gives an algorithm that is based on the RM and has optimal asymptotic scaling. In addition, since the RM seems to correspond to $q = 1$, which we know to be suboptimal, it is likely that the algorithm of [12] can be made asymptotically optimal by changing the scheduling.

1.1 General setup

We assume we have a physical system and a set of (time-independent) Hamiltonians, i.e. self-adjoint operators, such that we can evolve the system under e^{-itH} at a cost of t for any Hamiltonian H in this set.[1] We call the Hamiltonians in this set admissible. Which Hamiltonians are admissible will depend on the device or setup, but typically they will be bounded in norm.

This is not the cost model used by references [7] and [12], which use a query complexity rather than a time complexity. We discuss a translation of our results to this model using optimal Hamiltonian simulation in appendix B. The asymptotic complexities are mostly unaffected, but there are different constants involved.

For a given instance of a problem, we assume that we have a continuous, twice differentiable

---

[1] We set $\hbar = 1$.

path of admissible Hamiltonians $H(s)$, where $s \in [0, 1]$. We also assume that we can prepare the ground state of $H(0)$.

We are interested in the asymptotic scaling of the time complexity, as measured by the total length of time we apply unitaries of the form e^{-itH}. We produce theorems that give bounds on the complexity in terms of the spectral gap and the derivatives $\|H'\|$ and $\|H''\|$.

Our main tool will be the randomised application of unitaries of the form e^{-itH}. Since we are using classical randomness, it will be useful to use the density matrix formalism. Using this formalism, we can derive differential equations for these new procedures that share essential features with the Liouville–von Neumann equation in the adiabatic limit. This allows us to use many of the same mathematical tricks to study these procedures and we can derive "generalised" adiabatic theorems in the sense of [3].

1.1.1   Cost and error model

It is clear what the cost of one run of the algorithm is: it is just the total time spent evolving the system under some Hamiltonian. In order to state the time complexity we have the additional problem that the running time of the algorithm is not deterministic. That is, even for a fixed input, multiple runs of the algorithm will take different amounts of time. Our time complexity uses the expected run time of each input. Thus we say our algorithm has time complexity T if, for all relevant inputs I, the expected time taken by the algorithm with input I is less than T. In other words, we may consider this a worst-case expected-time complexity.

In order to guarantee that the algorithm does not take too long, we could abort if the chosen amount of time was too long. This would yield an additional error, which can be bounded by Markov's inequality.

Our algorithms are also not guaranteed to give the correct answer, rather we aim to produce the target state with at least a certain target fidelity.

1.1.2   Technical assumptions on the spectrum

We assume the existence of the following objects: a number $\Delta_m > 0$ and functions $\omega_0 : [0, 1] \to \mathbb{R}$ and $\Delta : [0, 1] \to [0, 1]$ such that

- ω_0 continuous;
- $\omega_0(s)$ is an eigenvalue of $H(s)$ for all $s \in [0, 1]$;
- $\Delta(s) \geq \Delta_m$ for all $s \in [0, 1]$;
- the intersection of $[\omega_0(s) - \Delta(s), \omega_0(s) + \Delta(s)]$ with the spectrum of $H(s)$ is exactly $\{\omega_0(s)\}$.

Let $P(s)$ be the projector on the eigenspace associated to the eigenvalue $\omega(s)$. We also set $Q(s) = \mathrm{id} - P(s)$.

In order to perform our algorithm, we assume knowledge of Δ, which bounds the gap. We do not assume more detailed knowledge of the gap, ω_0, or any other part of the spectrum.

2   Poisson-distributed phase randomisation

Our algorithms are built using a finite number of steps, where at each step a Zeno-like dephasing operation is performed. This dephasing operation is given by the following proposition:

▶ **Proposition 1** (Phase randomisation [4]). *Let H be a Hamiltonian and ω_0, Δ, P and Q as above. Assume we can simulate e^{-itH} for any positive or negative time t at a cost of $|t|$. Then we can construct a stochastic variable τ such that for all states ρ,*

$$\langle e^{-i\tau H} \rho e^{i\tau H} \rangle = P\rho P + Q\langle e^{-i\tau H} \rho e^{i\tau H} \rangle Q, \tag{1}$$

with cost $\langle |\tau| \rangle = t_0/\Delta$, where $t_0 = 2.32132$.

The angled brackets mean taking the average over τ. The result is originally from [4]. The value for t_0 was obtained in theorem 2 of [12].

The algorithm is now simple to state:

▪ **Algorithm 1** Poisson-distributed phase randomisation.

---
1 Pick a Poisson process $N : [0,1] \times (\Omega, \mathcal{A}, P) \to \mathbb{N}$ with rate $\lambda(s)$;
2 At each jump point s of the Poisson process, pick an instance t of the random variable T as defined in proposition 1 and evolve the system under the Hamiltonian evolution $e^{-itH(s)}$;

The density matrix describing the system is a random variable that satisfies the stochastic differential equation $\mathrm{d}\rho = \left(e^{-i\tau(s)H(s)} \rho e^{i\tau(s)H(s)} - \rho \right) \mathrm{d}N$. Averaging over realisations, we get

$$\mathrm{d}\langle \rho \rangle = \left(P\langle \rho \rangle P + Q\langle e^{-i\tau H} \rho e^{i\tau H} \rangle Q - \langle \rho \rangle \right) \lambda \, \mathrm{d}s. \tag{2}$$

Note that this should properly be thought of as a "marginalised" density matrix, rather than an "average" density matrix. This is entirely analoguous to the situation for classical probability distributions, where integrating out a variable gives the marginal distribution. In this case we are marginalising over the choice of Poisson process N. In the rest of the paper, we will use ρ to refer to the marginalised distribution $\langle \rho \rangle$. This corresponds to the density matrix you would observe if you were not told which Poisson process N was chosen.

The total time taken by one run of the algorithm is a random variable T satisfying

$$\mathrm{d}T = \tau \, \mathrm{d}N. \tag{3}$$

In order to find the time complexity, we take the average. This gives $\mathrm{d}T = t_0 \Delta^{-1} \lambda \, \mathrm{d}s$, so $T = \int_0^1 \frac{t_0 \lambda}{\Delta} \, \mathrm{d}s$.

2.1 Analysis

▶ **Lemma 2.** *Under the assumptions in 1.1.2, the algorithm 1 with rate $\lambda(s)$ produces a state with an infidelity that is bounded by*

$$\epsilon \leq \left\| \lambda(0)^{-1} P'(0) \right\| + \left\| \lambda(1)^{-1} P'(1) \right\| + \int_0^1 \left(\left\| \frac{P''}{\lambda} \right\| + \left| \left(\frac{1}{\lambda} \right)' \right| \|P'\| \right) \mathrm{d}s. \tag{4}$$

Proof. The infidelity is given by $\epsilon = 1 - \mathrm{Tr}\left(P(1)\rho(1) \right) = \mathrm{Tr}\left(P(0)\rho(0) \right) - \mathrm{Tr}\left(P(1)\rho(1) \right) = \left| \mathrm{Tr}(P\rho)\big|_0^1 \right|$, so it makes sense to track how the fidelity $\mathrm{Tr}\left(P(s)\rho(s) \right)$ changes in time.

We construct a differential equation for $\mathrm{Tr}(P\rho)$ by taking the derivative with respect to s, $\mathrm{Tr}(P\rho)' = \mathrm{Tr}(P'\rho) + \mathrm{Tr}(P\rho')$. This can be simplified using the fact that $PP'P = 0$ and $QP'Q = 0$.[2] Indeed, we have

$$\mathrm{Tr}(P\rho') = \lambda \, \mathrm{Tr}\left(P(P\rho P + Q\langle e^{-i\tau H} \rho e^{i\tau H} \rangle Q - \rho) \right) = \mathrm{Tr}(P\rho P) - \mathrm{Tr}(P\rho) = 0 \tag{5}$$

---

[2] We have $P' = (PP)' = P'P + PP'$, so $PP'P = 2PP'P$ and $QP'Q = 0$.

and

$$\text{Tr}(P'\rho) = \text{Tr}\left(P'(P\rho P + Q\langle e^{-i\tau H}\rho e^{i\tau H}\rangle Q - \lambda^{-1}\rho')\right) \tag{6}$$

$$= \text{Tr}(PP'P\rho) + \text{Tr}\left((QP'Q)\langle e^{-i\tau H}\rho e^{i\tau H}\rangle\right) - \text{Tr}(\lambda^{-1}P'\rho') \tag{7}$$

$$= -\text{Tr}(\lambda^{-1}P'\rho'), \tag{8}$$

so $\text{Tr}(P\rho)' = -\text{Tr}(\lambda^{-1}P'\rho')$. Integrating gives

$$\text{Tr}(P\rho)\big|_0^1 = -\int_0^1 \text{Tr}(\lambda^{-1}P'\rho')\,\mathrm{d}s \tag{9}$$

$$= -\lambda^{-1}\text{Tr}(P'\rho)\big|_0^1 + \int_0^1 \left(\text{Tr}\left(\frac{P''}{\lambda}\rho\right) + \text{Tr}\left(\left(\tfrac{1}{\lambda}\right)'P'\rho\right)\right)\mathrm{d}s, \tag{10}$$

which we can bound by

$$\epsilon = \left|\text{Tr}(P\rho)\big|_0^1\right| \leq \left|\text{Tr}(\lambda^{-1}P'\rho)\big|_0^1\right| + \int_0^1 \left(\text{Tr}\left(|\frac{P''}{\lambda}\rho|\right) + \left|\left(\tfrac{1}{\lambda}\right)'\right|\text{Tr}\left(|P'\rho|\right)\right)\mathrm{d}s \tag{11}$$

$$\leq \left(\left\|\lambda(0)^{-1}P'(0)\right\| + \left\|\lambda(1)^{-1}P'(1)\right\|\right)\text{Tr}(\rho)$$

$$+ \int_0^1 \left(\left\|\frac{P''}{\lambda}\right\| + \left|\left(\tfrac{1}{\lambda}\right)'\right|\left\|P'\right\|\right)\text{Tr}(\rho)\,\mathrm{d}s \tag{12}$$

$$\leq \left\|\lambda(0)^{-1}P'(0)\right\| + \left\|\lambda(1)^{-1}P'(1)\right\|$$

$$+ \int_0^1 \left(\left\|\frac{P''}{\lambda}\right\| + \left|\left(\tfrac{1}{\lambda}\right)'\right|\left\|P'\right\|\right)\mathrm{d}s. \tag{13}$$

◄

The next step is to bound $\|P'\|$ and $\|P''\|$ by more useful quantities. We make use of the following lemma:

► **Lemma 3.** *Under the assumptions stated in 1.1.2, we have*

1. $\|P'\| \leq 2\frac{\|H'\|}{\Delta}$;
2. $\|P''\| \leq 8\frac{\|H'\|^2}{\Delta^2} + 2\frac{\|H''\|}{\Delta}$.

This is fairly standard. See for example [16]. A proof is provided in appendix A. We are now ready to use lemma 2 in two distinct contexts, leading to theorems 4 and 5.

2.1.1   Constant λ

We first derive a theorem under the assumption that λ is constant. In this case we obtain the following result:

► **Theorem 4.** *Under the assumptions in 1.1.2, the algorithm 1 produces the target state with fidelity of at least $1 - \epsilon$ if λ is constant and*

$$\epsilon^{-1}2\left(\frac{\|H'(0)\|}{\Delta(0)} + \frac{\|H'(1)\|}{\Delta(1)} + \int_0^1 4\frac{\|H'\|^2}{\Delta^2} + \frac{\|H''\|}{\Delta}\,\mathrm{d}s\right) \leq \lambda. \tag{14}$$

In this case the time complexity of the procedure is given by $T = \lambda t_0 \int_0^1 \frac{1}{\Delta}\,\mathrm{d}s$.

Proof. Let ϵ_0 be the actual error of the algorithm. We need $\epsilon_0 \leq \epsilon$. We can use lemma 3 to rewrite the inequality in lemma 2 as

$$\epsilon_0 \leq \lambda^{-1} \big(\|P'(0)\| + \|P'(1)\| \big) + \lambda^{-1} \int_0^1 \|P''\| \, \mathrm{d}s \tag{15}$$

$$\leq \lambda^{-1} \Big(2 \frac{\|H'(0)\|}{\Delta(0)} + 2 \frac{\|H'(1)\|}{\Delta(1)} + \int_0^1 8 \frac{\|H'\|^2}{\Delta^2} + 2 \frac{\|H''\|}{\Delta} \, \mathrm{d}s \Big). \tag{16}$$

Set $B := 2 \Big(\frac{\|H'(0)\|}{\Delta(0)} + \frac{\|H'(1)\|}{\Delta(1)} + \int_0^1 4 \frac{\|H'\|^2}{\Delta^2} + \frac{\|H''\|}{\Delta} \, \mathrm{d}s \Big)$. Then we have

$$\epsilon_0 \leq \lambda^{-1} B \leq \epsilon B^{-1} B = \epsilon, \tag{17}$$

so the algorithm works. The time complexity is simply given by $T = \int_0^1 \frac{t_0 \lambda}{\Delta} \, \mathrm{d}s = \lambda t_0 \int_0^1 \frac{1}{\Delta} \, \mathrm{d}s$.
◀

This result can be compared to theorem 15 in the circuit model.

2.1.2 Scaling λ with the gap

We know from [17] and [20] that the performance of AQC can be improved with an adapted schedule, taking more time when the gap is small. Similarly we expect it to be possible to improve the performance of our procedure by varying λ. Indeed this is the case.

▶ **Theorem 5.** *Under the assumptions in 1.1.2, we additionally assume that there exists $0 \leq q \leq 1$ and B_1, B_2 such that $\int_0^1 \frac{1}{\Delta^{1+q}} \, \mathrm{d}s \leq B_1 \Delta_m^{-q}$ and $\int_0^1 \frac{1}{\Delta^{2-q}} \, \mathrm{d}s \leq B_2 \Delta_m^{q-1}$ for all instances of the problem. Then algorithm 1 produces the target state with a fidelity of at least $1 - \epsilon$ if*

$$\lambda = \epsilon^{-1} \frac{C}{\Delta^q \Delta_m^{1-q}}, \tag{18}$$

where $C := 2 \sup_{s \in [0,1]} \Big(2 \|H'(s)\| + 4 \|H'(s)\|^2 B_2 + \|H''(s)\| + q |\Delta'(s)| \|H'(s)\| B_2 \Big)$.
In this case the time complexity of the procedure is given by

$$T \leq \epsilon^{-1} \frac{t_0 B_1 C}{\Delta_m}.$$

▶ **Corollary 6.** *If $\int_0^1 \frac{1}{\Delta^p} \, \mathrm{d}s = O(\Delta_m^{1-p})$ holds for all $p > 1$, $|\Delta'| = O(1)$, $\|H'\| = O(1)$ and $\|H''\| = O(1)$, then algorithm 1 with the rate defined in theorem 5 produces the target state with fidelity $1 - \epsilon$ and a time complexity of $O(\Delta_m^{-1})$ for all $0 < q < 1$.*

Proof of Theorem 5. Let ϵ_0 be the actual error of the algorithm, we need $\epsilon_0 \leq \epsilon$. In this case the inequality in lemma 2 becomes

$$\epsilon_0 \leq \epsilon C^{-1} \Delta_m^{1-q} \big(\Delta(0)^q \|P'(0)\| + \Delta(1)^q \|P'(1)\| \big) + \epsilon C^{-1} \int_0^1 \Delta^q \Delta_m^{1-q} \|P''\| + \Big| \big(\Delta^q \Delta_m^{1-q} \big)' \Big| \|P'\| \, \mathrm{d}s. \tag{19}$$

We bound the terms separately, using lemma 3. For the first, we have

$$\Delta_m^{1-q} \Delta^q \|P'\| \leq 2 \Delta_m^{1-q} \Delta^q \frac{\|H'\|}{\Delta} = 2 \Delta_m^{1-q} \frac{\|H'\|}{\Delta^{1-q}} \leq 2 \Delta_m^{1-q} \frac{\|H'\|}{\Delta_m^{1-q}} = 2 \|H'\| \leq 2 \sup_{s \in [0,1]} \|H'\| \tag{20}$$

at both $s = 0$ and $s = 1$, so we bound the sum by $4 \sup_{s \in [0,1]} \|H'\|$.

The second term splits into two, since we bound $\|P''\|$ by $8 \frac{\|H'\|^2}{\Delta^2} + 2 \frac{\|H''\|}{\Delta}$. For the first part we have

$$8 \int_0^1 \Delta^q \Delta_m^{1-q} \frac{\|H'\|^2}{\Delta^2} \, \mathrm{d}s \le 8 \sup_{s \in [0,1]} \|H'(s)\|^2 \Delta_m^{1-q} \int_0^1 \frac{1}{\Delta^{2-q}} \, \mathrm{d}s \tag{21}$$

$$\le 8 \sup_{s \in [0,1]} \|H'(s)\|^2 B_2 \Delta_m^{1-q} \Delta_m^{q-1} = 8 \sup_{s \in [0,1]} \|H'(s)\|^2 B_2. \tag{22}$$

The second part gives

$$2 \int_0^1 \Delta^q \Delta_m^{1-q} \frac{\|H''\|}{\Delta} \, \mathrm{d}s \le 2 \sup_{s \in [0,1]} \|H''(s)\| \Delta_m^{1-q} \int_0^1 \frac{1}{\Delta^{1-q}} \, \mathrm{d}s \tag{23}$$

$$\le 2 \sup_{s \in [0,1]} \|H''(s)\| \Delta_m^{1-q} \Delta_m^{q-1} = 2 \sup_{s \in [0,1]} \|H''(s)\|. \tag{24}$$

Finally, for the third term,

$$\int_0^1 \left| \left(\Delta^q \Delta_m^{1-q} \right)' \right| \|P'\| \, \mathrm{d}s = \int_0^1 q \Delta^{q-1} \Delta_m^{1-q} |\Delta'| \, \|P'\| \, \mathrm{d}s \tag{25}$$

$$\le 2q \Delta_m^{1-q} \left(\sup_{s \in [0,1]} |\Delta'(s)| \, \|H'(s)\| \right) \int_0^1 \frac{\Delta^{q-1}}{\Delta} \, \mathrm{d}s \tag{26}$$

$$= 2q \Delta_m^{1-q} \left(\sup_{s \in [0,1]} |\Delta'(s)| \, \|H'(s)\| \right) \int_0^1 \frac{1}{\Delta^{2-q}} \, \mathrm{d}s \tag{27}$$

$$\le 2q \left(\sup_{s \in [0,1]} |\Delta'(s)| \, \|H'(s)\| \right) \Delta_m^{1-q} B_2 \Delta_m^{q-1} \tag{28}$$

$$= 2q B_2 \left(\sup_{s \in [0,1]} |\Delta'(s)| \, \|H'(s)\| \right). \tag{29}$$

Plugging everything back into equation (19), gives

$$\epsilon_0 \le \epsilon C^{-1} \sup_{s \in [0,1]} \left(4 \|H'(s)\| + 8 \|H'(s)\|^2 B_2 + 2 \|H''(s)\| + 2q |\Delta'(s)| \, \|H'(s)\| B_2 \right) \tag{30}$$

$$= \epsilon C^{-1} C = \epsilon, \tag{31}$$

so the procedure works. We can then calculate the time complexity

$$T = \int_0^1 \frac{t_0 \lambda}{\Delta} \, \mathrm{d}s = \epsilon^{-1} t_0 \int_0^1 \frac{C}{\Delta^q \Delta_m^{1-q} \Delta} \, \mathrm{d}s = \epsilon^{-1} t_0 C \Delta_m^{q-1} \int_0^1 \frac{1}{\Delta^{q+1}} \, \mathrm{d}s$$

$$\le \epsilon^{-1} t_0 C \Delta_m^{q-1} B_1 \Delta_m^{-q} = \epsilon^{-1} t_0 C B_1 \Delta_m^{-1}. \tag{32}$$

◀

These results can be compared to theorem 16 and corollary 17 in the circuit model.

3   Improving the scaling in the error with eigenstate filtering

The use of eigenstate filtering was introduced in [13] to improve scaling in the error tolerance for algorithms based on adiabatic principles and the quantum Zeno effect, in particular with application to QLSP.

A similar technique was used in [7] to achieve optimal scaling, but using Linear Combinations of Unitaries (LCU) instead of Quantum Signal Processing (QSP). We adapt the technique of [7] to the present situation.

▶ **Theorem 7.** *Let H be a Hamiltonian with $\|H\| \leq 1$ and 0 in the spectrum of H, $\sigma(H)$. Suppose*

- $\Delta \geq 0$ *is such that* $[-\Delta, \Delta] \cap \sigma(H) = \{0\}$;
- P *is the orthogonal projector on the eigenspace associated to the eigenvalue 0, we set* $Q := \mathbb{1} - P$;
- ρ *is a density matrix of the form* $P\rho_0 P + Q\rho_1 Q$ *with* $\mathrm{Tr}(P\rho_0) > 1/2$, *that we can prepare at cost T_0;*
- $\epsilon > 0$.

Further, suppose

- *we can adjoin two ancilla qubits to ρ;*
- *we can can measure and reprepare the ancilla qubits;*
- *we can evolve the system under $H \otimes R$ and $\mathbb{1} \otimes R$ for time t for all Hermitian operators R on $\mathbb{C}^{2\times 2}$ with $\|R\| \leq 1$ at a cost of t.*

Then we can prepare a state ρ_2 such that $\mathrm{Tr}(P\rho_2) \geq 1 - \epsilon$ at a cost of $T = O(T_0 + \Delta^{-1} \log(1/\epsilon))$.

The idea of the procedure is relatively simple. With these assumptions, we can apply controlled versions of the unitary e^{-itH}, i.e. $e^{itH\otimes\Pi}$ for some projector Π on $\mathbb{C}^{2\times 2}$. This means that we can apply linear combinations of $e^{itH\otimes\Pi}$ using the technique of linear combinations of unitaries, see lemmas 8 and 9. In particular we can apply a polynomial that has a large peak at 0 and is very small everywhere else. We use this to filter out the part of the state that we do not want.

▶ **Lemma 8** (LCU with arbitrarily large ancilla register). *Let $f(x) = \sum_{k=-n}^{n} a_k x^k$ be a rational polynomial with complex coefficients such that $\sum_{k=-n}^{n} |a_k|^2 = 1$. Let H be a Hamiltonian and ρ the state of the system. Assume we have access to an ancilla register with orthonormal basis $\{|k\rangle \mid k \in \mathbb{Z}\}$. Then, at a cost of $O(nt)$, we can do an operation which either*

- *succeeds and applies $\sum_{k=-n}^{n} |a_k|^2 e^{-itkH}$ to the system,*
- *or fails, with a probability of $1 - \mathrm{Tr}\left(\left(\sum_{k=-n}^{n} |a_k|^2 e^{-itkH}\right)\rho\left(\sum_{k=-n}^{n} |a_k|^2 e^{itkH}\right)\right)$. We can see when this has happened thanks to the measured contents of the ancilla register.*

Proof. The procedure is as follows: we first prepare the ancilla in the state $|f\rangle := \sum_{k=-n}^{n} a_k |k\rangle$, then apply $\sum_{k=-n}^{n} kH \otimes |k\rangle\langle k|$ for time t and finally measure the state $|f\rangle$. If we measure any other state than $|f\rangle$, the procedure fails.

The result then follows from the following identity:

$$\sum_{k,l=-n}^{n} \left(\mathbb{1} \otimes a_k\langle k|\right) e^{-it\sum_{m=-n}^{n} mH\otimes|m\rangle\langle m|} \left(\mathbb{1} \otimes \overline{a_l}|l\rangle\right) = \sum_{m} |a_m|^2 e^{-itmH}. \tag{33}$$

Defining

$$\Pi_m^0 = \mathbb{1} - \sum_{k=0}^{m} |k\rangle\langle k| \qquad \text{and} \qquad \Pi_m^1 = \mathbb{1} - \sum_{k=-m}^{0} |k\rangle\langle k|, \tag{34}$$

we can write $e^{-it\sum_{m=-n}^{n} mH\otimes|m\rangle\langle m|} = \prod_{m=0}^{n-1} e^{-itH\otimes\Pi_m^0} e^{itH\otimes\Pi_m^1}$, which we can clearly apply at a cost of $2nt$.

The cost of ancilla preparation depends on the admissible operations on the ancilla register, but in a worst-case scenario, each a_k needs to be set separately[3] which means that the cost is $O(n)$. The total cost is then still $O(nt)$. ◀

---

[3] This is the case for the procedure used in lemma 9.

▶ **Lemma 9** (LCU with two ancilla qubits). *We can achieve the results of 8 only using two ancilla qubits at a time.*

The construction is identical to the one in [7].

Proof of theorem 7. Let $Q := \mathbb{1} - P$ and write $Q = \sum_j Q_j$, where each Q_j is an eigenprojector of H associated to the eigenvalue ω_j. Now we observe

$$\Big(\sum_{k=-n}^{n} |a_k|^2 e^{-ikH} \Big) Q_j = \Big(\sum_{k=-n}^{n} |a_k|^2 e^{-ik\omega_j} \Big) Q_j = A(\omega_j) Q_j, \tag{35}$$

where $A(\omega)$ is the Fourier transform of the sequence $|a_k|^2$. Thus

$$\Big(\sum_{k=-n}^{n} |a_k|^2 e^{-ikH} \Big) Q \rho Q \Big(\sum_{k=-n}^{n} |a_k|^2 e^{ikH} \Big) = \sum_{j,l} \Big(\sum_{k=-n}^{n} |a_k|^2 e^{-ikH} \Big) Q_j \rho Q_l \Big(\sum_{k=-n}^{n} |a_k|^2 e^{ikH} \Big) \tag{36}$$

$$= \sum_{j,l} A(\omega_j) A(-\omega_l) Q_j \rho Q_l. \tag{37}$$

Taking the trace gives $\mathrm{Tr}\Big(\sum_{j,l} A(\omega_j) A(-\omega_l) Q_j \rho Q_l \Big) \leq \max_{\omega \notin [-\Delta, \Delta]} A(\omega)^2 \, \mathrm{Tr}(Q \rho Q) \leq \max_{\omega \notin [-\Delta, \Delta]} A(\omega)^2$. The goal then becomes to find a sequence and its Fourier transform such that $A(\omega_0) = 1$, $\max_{\omega \notin [-\Delta, \Delta]} A(\omega)^2 \leq \epsilon$ and whose window n is as small as possible. The answer to this optimisation problem is well-known and is given by the Dolph-Chebyshev window [15]. In this case we need a window of[4]

$$n = \frac{\cosh^{-1}(1/\sqrt{\epsilon})}{\cosh^{-1}\big(\sec(\Delta) \big)} \leq \frac{1}{2\Delta} \log\Big(\frac{4}{\epsilon} \Big). \tag{38}$$

By lemma 8, we can implement this at a cost of $O(n)$. Note that this procedure terminates succesfully with a probability of at least $\mathrm{Tr}(P\rho_0)$ (which is bounded below) and we can check to see whether the procedure failed. If it failed, we repeat. On average we need to repeat fewer than $\mathrm{Tr}(P\rho_0)^{-1}$ times, which is $O(1)$. ◀

4 Applications

4.1 Grover search

For the Grover problem, we have an N-dimensional vector space we want to find an element of an M-dimensional subspace $\mathcal{M}$. In order to help us, we assume we have access to an oracle Hamiltonian $H_1 = \mathbb{1} - P_{\mathcal{M}}$, where $P_{\mathcal{M}}$ is the orthogonal projector on $\mathcal{M}$. In other words, we assume H_1 is admissible. We also assume $H_0 = \mathbb{1} - |u\rangle\langle u|$ is admissible, where $|u\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i\rangle$ is the uniform superposition. The aim is now to use the interpolation $H(s) = (1-s)H_0 + sH_1$ to prepare as state in $\mathcal{M}$. For more details see [8] and [18].

We see that $H(s)$ has four eigenvalues:

$$\lambda_{1,2} = \frac{1}{2}\left(1 \pm \sqrt{1 - 4(1 - \frac{M}{N})s(1-s)} \right) \qquad \text{with multiplicity } 1 \tag{39}$$

$$\lambda_3 = 1 - s \qquad \text{with multiplicity } M - 1 \tag{40}$$

$$\lambda_4 = 1 \qquad \text{with multiplicity } N - M - 1. \tag{41}$$

---

[4] We note that we improve the scaling by a factor of two compared to [7]. This is because we are able to start from a state where $P\rho Q = 0 = Q\rho P$.

The eigenvectors corresponding to λ_3 are the eigenvectors in $\mathcal{M}$ with zero overlap with $|u\rangle$. The eigenvectors corresponding to λ_4 are the eigenvectors in $\mathcal{M}^\perp$ with zero overlap with $|u\rangle$. Since the initial state has zero overlap with any of these vectors and they are eigenvectors of each $H(s)$, none of them are prepared by the procedure and everything happens in the two-dimensional space spanned by the eigenvectors associated to λ_1 and λ_2.

We have explicitly computed the gap, so we can use this as the bound Δ:

$$\Delta(s) = \sqrt{1 - 4(1 - \frac{M}{N})s(1-s)}. \tag{42}$$

We can set $\Delta_m = \min_{s \in [0,1]} \Delta(s) = \sqrt{M/N}$. In order to give bounds on the time-complexity, we use the following result:

▶ **Lemma 10.** *For all $p > 1$ and Δ given by (42), we have*

$$\int_0^1 \frac{1}{\Delta(s)^p}\, \mathrm{d}s = O\big(\sqrt{N/M}^{\,p-1}\big) = O\big(\Delta_m^{1-p}\big), \tag{43}$$

and, for $p = 1$,

$$\int_0^1 \frac{1}{\Delta(s)}\, \mathrm{d}s = O\big(\log(N/M)\big). \tag{44}$$

We provide a proof in appendix C.1. For constant λ, we apply theorem 4 and use the lemma 10 to get a time complexity $O\big(\sqrt{N/M}\log(N/M)\big)$.

We are able to take the q in corollary 6 to be anywhere in the range $0 < q < 1$, since for any such q both $1 + q$ and $2 - q$ are strictly greater than 1. This is related to the range of schedules described in [2]. The time complexity of the algorithm for any such q is $O\big(\sqrt{N/M}\big)$, since it is easy to check that to other conditions hold: $\|H'\| = \|H_1 - H_0\|$, $\|H''\| = 0$ and

$$|\Delta'| = \left| \frac{4(1 - \frac{M}{N})(\frac{1}{2} - s)}{\Delta} \right| \tag{45}$$

$$\leq \frac{2\sqrt{4(1 - \frac{M}{N})(\frac{1}{2} - s)^2}}{\Delta} \tag{46}$$

$$\leq \frac{2\sqrt{\frac{M}{N} + 4(1 - \frac{M}{N})(\frac{1}{2} - s)^2}}{\Delta} = 2\frac{\Delta}{\Delta} = 2. \tag{47}$$

4.2 Solving linear systems of equations

The Quantum Linear Systems Problem (QLSP) was introduced in [10]. Suppose A is an invertible $N \times N$ matrix $b \in \mathbb{C}^N$ a vector. The goal is to prepare the quantum state $\frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}$. We express the time complexity of our algorithm in terms of the condition number $\kappa = \|A\|\,\|A^{-1}\|$.

We may restrict ourselves to Hermitian matrices because we can use the following trick from [10]: If A is not Hermitian, we consider the matrix $\begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix}$, which has the same condition number, and solve the equation $\begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix}|y\rangle = \begin{pmatrix} |b\rangle \\ 0 \end{pmatrix}$.

First we rescale the matrix A to $\frac{A}{\|A\|}$. We do this because typically admissible matrices need to be uniformly bounded. This has the effect of shifting the lowest singular value from $\frac{1}{\|A^{-1}\|}$ to $\frac{1}{\|A\|\|A^{-1}\|} = \kappa^{-1}$. Now we consider a path of Hamiltonians that was introduced in

[19]. Define $A(s) := (1-s)\sigma_z \otimes \mathbb{1} + s\sigma_x \otimes A$, $Q_{b,+} := \mathbb{1} - (|+\rangle|b\rangle)(\langle+|\langle b|)$ and $\sigma_\pm := \frac{1}{2}(\sigma_x \pm i\sigma_y)$.
Set $H(s) = \sigma_+ \otimes (A(s)Q_{b,+}) + \sigma_- \otimes (Q_{b,+}A(s))$. This can be written as a linear interpolation
$H(s) = (1-s)H_0 + sH_1$, where

$$H_0 := \sigma_+ \otimes ((\sigma_z \otimes \mathbb{1})Q_{b,+}) + \sigma_- \otimes (Q_{b,+}(\sigma_z \otimes \mathbb{1})) \tag{48}$$

$$H_1 := \sigma_+ \otimes ((\sigma_x \otimes A)Q_{b,+}) + \sigma_- \otimes (Q_{b,+}(\sigma_x \otimes A)). \tag{49}$$

Following the analysis of [19], we see that $H(s)$ has 0 as an eigenvalue for all $s \in [0,1]$. The
corresponding eigenspace is spanned by $\{|0\rangle \otimes |x(s)\rangle, |1\rangle \otimes |+\rangle|b\rangle\}$, where $|x(s)\rangle := \frac{A(s)^{-1}|b\rangle}{\|A(s)^{-1}|b\rangle\|}$.
Since $H(s)$ does not allow transition between these states, we are sure to not prepare
$|1\rangle \otimes |+\rangle|b\rangle$, so long as we start with $|0\rangle \otimes |x(0)\rangle$.

In [19] it was also shown that the eigenvalue zero is separated from the rest of the
spectrum by a gap that is at least

$$\Delta(s) = \sqrt{(1-s)^2 + \left(\frac{s}{\kappa}\right)^2}. \tag{50}$$

If κ is large enough, then we can take $\Delta_m := \frac{1}{2\kappa} \leq \sqrt{\frac{1}{\kappa^2+1}} = \min_{s \in [0,1]} \Delta(s)$.

In order to give bounds on the time-complexity, we use the following result:

▶ **Lemma 11.** *For all $p > 1$, we have*

$$\int_0^1 \frac{1}{\Delta(s)^p} \, ds = O(\kappa^{p-1}) = O(\Delta_m^{1-p}), \tag{51}$$

and, for $p = 1$,

$$\int_0^1 \frac{1}{\Delta(s)} \, ds = O(\log(\kappa)). \tag{52}$$

We provide a proof in appendix C.2.

For constant λ, we apply theorem 4 and use the lemma 11 to get a time complexity
$O(\kappa \log(\kappa))$. This is also the complexity that was obtained in [19].

As before, we have a full order reduction for $p > 1$ and thus we are able to take the q in
corollary 6 to be anywhere in the range $0 < q < 1$, since for any such q both $1 + q$ and $2 - q$
are strictly greater than 1. If $q = 0$ or $q = 1$, the complexity gains a factor of $\log(\kappa)$. This
exactly mirrors the situation in [2] and is the reason why the algorithms for QLSP based on
the RM have an extra factor of $\log(\kappa)$ in the asymptotic complexity, see [19] and [12].

We can apply 6 since $\|H'\| = \|H_1 - H_0\|$, $\|H''\| = 0$ and

$$|\Delta'| = \left| \frac{s - 1 + s/\kappa^2}{\Delta} \right| \tag{53}$$

$$= \frac{\sqrt{(s - 1 + s/\kappa^2)^2}}{\Delta} \tag{54}$$

$$= \frac{\sqrt{(1 + 1/\kappa^2)^2 s^2 - (1 + 1/\kappa^2)2s + 1}}{\Delta} \tag{55}$$

$$\leq \frac{\sqrt{(1 + 1/\kappa^2)^2 s^2 - (1 + 1/\kappa^2)2s + (1 + 1/\kappa^2)}}{\Delta} \tag{56}$$

$$= \sqrt{1 + 1/\kappa^2} \frac{\Delta}{\Delta} = \sqrt{1 + 1/\kappa^2} = O(1). \tag{57}$$

This yields a time complexity of $O(\kappa)$ for fixed error tolerance. The scaling on both condition number and error tolerance is $O(\epsilon^{-1}\kappa)$. By a straightforward application of theorem 7 at $s = 1$, we get a scaling of $O\big(\log(\epsilon^{-1})\kappa\big)$. This is possible, since we know the eigenvalue of interest is 0.

This result is optimal and matches the complexity reported in [7], where it was achieved using a very different method.

References

1 Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 42–51, 2004. `doi:10.1109/FOCS.2004.8`.

2 Dong An and Lin Lin. Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm. *ACM Transactions on Quantum Computing*, 3(2):1–28, March 2022. `doi:10.1145/3498331`.

3 Joseph E. Avron, Martin Fraas, Gian M. Graf, and Philip Grech. Adiabatic theorems for generators of contracting evolutions. *Communications in Mathematical Physics*, 314(1):163–191, May 2012. `doi:10.1007/s00220-012-1504-1`.

4 Sergio Boixo, Emanuel Knill, and Rolando Somma. Eigenpath traversal by phase randomization. *Quantum Information and Computation*, 9(9&10):833–855, September 2009. `doi:10.26421/QIC9.9-10-7`.

5 Andrew M. Childs, Enrico Deotto, Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Andrew J. Landahl. Quantum search by measurement. *Physical Review A*, 66(3), September 2002. `doi:10.1103/physreva.66.032314`.

6 Pedro C. S. Costa, Dong An, Ryan Babbush, and Dominic Berry. The discrete adiabatic quantum linear system solver has lower constant factors than the randomized adiabatic solver, 2023. `arXiv:2312.07690`.

7 Pedro C.S. Costa, Dong An, Yuval R. Sanders, Yuan Su, Ryan Babbush, and Dominic W. Berry. Optimal scaling quantum linear-systems solver via discrete adiabatic theorem. *PRX Quantum*, 3:040303, October 2022. `doi:10.1103/PRXQuantum.3.040303`.

8 Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000. `arXiv:quant-ph/0001106`.

9 Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. Association for Computing Machinery. `doi:10.1145/237814.237866`.

10 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), October 2009. `doi:10.1103/physrevlett.103.150502`.

11 Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler. Bounds for the adiabatic approximation with applications to quantum computation. *Journal of Mathematical Physics*, 48(10), October 2007. `doi:10.1063/1.2798382`.

12 David Jennings, Matteo Lostaglio, Sam Pallister, Andrew T Sornborger, and Yiğit Subaşı. Efficient quantum linear solver algorithm with detailed running costs, 2023. `arXiv:2305.11352`.

13 Lin Lin and Yu Tong. Optimal polynomial based quantum eigenstate filtering with application to solving quantum linear systems. *Quantum*, 4:361, November 2020. `doi:10.22331/q-2020-11-11-361`.

14 Andrew Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2, 2014. `doi:10.3389/fphy.2014.00005`.

15 Peter Lynch. The dolph–chebyshev window: A simple optimal filter. *Monthly Weather Review*, 125(4):655–660, 1997. `doi:10.1175/1520-0493(1997)125<0655:TDCWAS>2.0.CO;2`.

16   Ben W. Reichardt. The quantum adiabatic optimization algorithm and local minima. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 502–510, New York, NY, USA, 2004. Association for Computing Machinery. `doi:10.1145/1007352.1007428`.

17   Jérémie Roland and Nicolas J. Cerf. Quantum search by local adiabatic evolution. *Physical Review A*, 65(4), March 2002. `doi:10.1103/physreva.65.042308`.

18   Jérémie Roland and Nicolas J. Cerf. Quantum-circuit model of hamiltonian search algorithms. *Physical Review A*, 68(6), December 2003. `doi:10.1103/physreva.68.062311`.

19   Yiğit Subaşı, Rolando D. Somma, and Davide Orsucci. Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing. *Physical Review Letters*, 122(6), February 2019. `doi:10.1103/physrevlett.122.060504`.

20   Wim van Dam, Michele Mosca, and Umesh Vazirani. How powerful is adiabatic quantum computation? In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. IEEE, 2001. `doi:10.1109/sfcs.2001.959902`.

A   Bounds on derivatives of projectors

We provide a proof of lemma 3

▶ **Lemma.** *Under the assumptions stated in 1.1.2, we have*
1. $\|P'\| \leq 2\frac{\|H'\|}{\Delta}$;
2. $\|P''\| \leq 8\frac{\|H'\|^2}{\Delta^2} + 2\frac{\|H''\|}{\Delta}$.

Proof. Let Γ be a circle in the complex plane, centred at the ground energy with radius $\Delta/2$. Then we have the Riesz form of the projector

$$P = \frac{1}{2\pi i} \oint_\Gamma R_H(z) \, dz,$$

where $R_H(z) = \left(z\,\mathrm{id} - H\right)^{-1}$ is the resolvent of H at z. Then $R_H(z)' = R_H(z)H'R_H(z)$ (the derivative is with respect to s, not z). As H is a normal operator, the norm $\|R_H(z)\|$ is equal to the inverse of the distance from z to the spectrum $\sigma(H)$. On the circle Γ this is equal to $(\Delta/2)^{-1}$ everywhere. We can then approximate

$$\begin{aligned}
\|P'\| &= \left\| \frac{1}{2\pi i} \oint_\Gamma R_H(z)' \, dz \right\| \\
&\leq \frac{1}{2\pi} \oint_\Gamma \|R_H(z)'\| dz \\
&\leq \frac{1}{2\pi} \oint_\Gamma \|R_H(z)\| \cdot \|H'\| \cdot \|R_H(z)\| \, dz \\
&= \frac{1}{2\pi} \left(\frac{2}{\Delta}\right)^2 \|H'\| \oint_\Gamma dz \\
&= \frac{1}{2\pi} \left(\frac{2}{\Delta}\right)^2 2\pi \frac{\Delta}{2} \|H'\| \\
&= 2\frac{\|H'\|}{\Delta}.
\end{aligned}$$

Similarly, we can write

$$P'' = \frac{1}{2\pi i} \oint_\Gamma 2R_H(z)H'R_H(z)H'R_H(z) + R_H(z)H''R_H(z) \, dz.$$

Estimating this in the same way as before yields

$$\|P''\| \leq 8\frac{\|H'\|^2}{\Delta^2} + 2\frac{\|H''\|}{\Delta}. \qquad \qquad \blacktriangleleft$$

B Comparison with the circuit model

So far we have assumed access to a device that can evolve a system under a given Hamiltonian in real time, i.e. it takes time t to apply e^{-itH}. This is the typical setting of AQC and is also the setting of [4] and [19].

Many papers use a slightly different setup. In [7] and [12] the setting is a standard quantum computer which is given access to block encodings of $H(s)$ for all $s \in [0,1]$. In this case the complexity is given by the number of times such a block-encoded Hamiltonian is used. In other words, the complexity is a query complexity rather than a time complexity.

Given access to only a block encoding of a Hamiltonian H, it is generally not possible to simulate e^{-itH} exactly. Instead, we can use proposition 12, which is taken from [12].

▶ **Proposition 12** (Theorem 4 from [12])**.** *Given access to an* $(\alpha, m, 0)$*–block-encoding* U_H *of a Hermitian operator* H *with* $\|H\| \leq 1$*, we can realise a* $(1, m+2, \delta)$*–block-encoding of* e^{-itH} *for* $t \in \mathbb{R}$ *with*

$$3\left\lceil \frac{e}{2}\alpha|t| + \log\left(\frac{2c}{\delta}\right) \right\rceil$$

calls to U_H, U_H^* *with* $c = 4(\sqrt{2\pi}e^{\frac{1}{13}})^{-1} \approx 1.47762$.

In this proposition δ gives the error of the block encoding, i.e. if U is the block encoding, then $\left\|U - e^{-itH}\right\| \leq \delta$.

This motivates replacing the algorithm 1 by the algorithm 2, which now depends on both a rate $\lambda(s)$ and an allowable simulation error $\delta(s)$.

■ **Algorithm 2** Poisson-distributed phase randomisation with imperfect time evolution.

---

1 Pick a Poisson process $N : [0,1] \times (\Omega, \mathcal{A}, P) \to \mathbb{N}$ with rate $\lambda(s)$;

2 At each jump point s of the Poisson process, pick an instance t of the random variable T as defined in proposition 1 and use proposition 12 to simulate the evolution under the time-independent Hamiltonian $H(s)$ for a time t with error at most $\delta(s)$;

In this case the number of queries is bounded by a quantity Q is a random variable with stochastic differential equation $dQ = 3\left(\frac{e}{2}\alpha|\tau| + \log\left(\frac{2c}{\delta}\right) + 1\right)dN$. Taking the average yields $Q = 3\int_0^1 \left(\frac{e\alpha t_0}{2\Delta} + \log\left(\frac{2c}{\delta}\right) + 1\right)\lambda\,ds$.

To analyse algorithm 2, we prove lemma 13, which is analogous to lemma 2.

▶ **Lemma 13.** *Given the assumptions in 1.1.2 and that for all* $s \in [0,1]$ *and* $t \in [0, \infty[$*, we can apply an operation* $A(s,t)$ *such that* $\left\|e^{-itH(s)} - A(s,t)\right\| \leq \delta(s)$*, the algorithm 1 with rate* $\lambda(s)$ *has an error that is bounded by*

$$\epsilon \leq \int_0^1 (2\delta + \delta^2)\left(2\frac{\|H'\|}{\Delta} + \lambda(s)\right)ds + \left\|\lambda(0)^{-1}P'(0)\right\| + \left\|\lambda(1)^{-1}P'(1)\right\| + \int_0^1 \left(\left\|\frac{P''}{\lambda}\right\| + \left|\left(\frac{1}{\lambda}\right)'\right|\|P'\|\right)ds. \tag{58}$$

Proof. The differential equation (2) is then

$$\rho' = \lambda\left(P\rho P + Q\langle e^{-i\tau H}\rho e^{i\tau H}\rangle Q - \rho\right) + \lambda\left(A(s,t)\rho A(s,t)^* - e^{-i\tau H}\rho e^{i\tau H}\right). \tag{59}$$

We set $E := A(s,t)\rho A(s,t)^* - e^{-i\tau H}\rho e^{i\tau H}$. Then equations (5) and (8) become $\mathrm{Tr}(P\rho') = \lambda\,\mathrm{Tr}(E)$ and $\mathrm{Tr}(P'\rho) = \mathrm{Tr}(P'E) - \mathrm{Tr}(\lambda^{-1}P'\rho')$, so

$$\mathrm{Tr}(P\rho)' = \mathrm{Tr}(P\rho') + \mathrm{Tr}(P'\rho) = \lambda\,\mathrm{Tr}(E) + \mathrm{Tr}(P'E) - \mathrm{Tr}(\lambda^{-1}P'\rho'). \tag{60}$$

The integral of $-\operatorname{Tr}(\lambda^{-1}P'\rho')$ was bounded in the proof of lemma 2.

Using lemma 14, we bound $\operatorname{Tr}(|E|) \le 2\delta + \delta^2$. Together with the bound $\|P'\| \le 2\frac{\|H'\|}{\Delta}$, this yields the result. ◀

In this proof we have made use of the following lemma:

▶ **Lemma 14** (Lemma 9 from [12]). *Suppose A, B are operators such that $\|A - B\| \le \delta$. Then, for each density operators ρ, we have*

$$\operatorname{Tr}(|A\rho A^* - B\rho B^*|) \le 2\delta \|A\| + \delta^2. \tag{61}$$

▶ **Theorem 15.** *Under the assumptions in 1.1.2, the algorithm 2 produces the target state with fidelity of at least $1 - \epsilon$ if λ is constant, $\delta = \frac{4\epsilon}{27\lambda}$ and*

$$\epsilon^{-1}4\Big(\frac{\|H'(0)\|}{\Delta(0)} + \frac{\|H'(1)\|}{\Delta(1)} + \int_0^1 4\frac{\|H'\|^2}{\Delta^2} + \frac{\|H''\|}{\Delta}\,\mathrm{d}s\Big) \le \lambda. \tag{62}$$

Using the Hamiltonian simulation of proposition 12, this gives a query complexity of

$$Q = \lambda\Big(e\alpha t_0\frac{3}{2}\int_0^1 \frac{1}{\Delta}\,\mathrm{d}s + 3\log\big(\frac{27c}{2\epsilon}\big) + \log(\lambda) + 1\Big).$$

Proof. Let ϵ_0 be the actual error of the algorithm. We need $\epsilon_0 \le \epsilon$. As in the proof of 4, rewrite the inequality in lemma 13 as

$$\epsilon_0 \le (2\delta+\delta^2)\Big(\int_0^1 2\frac{\|H'\|}{\Delta}\,\mathrm{d}s + \lambda\Big) + \lambda^{-1}\Big(2\frac{\|H'(0)\|}{\Delta(0)} + 2\frac{\|H'(1)\|}{\Delta(1)} + \int_0^1 8\frac{\|H'\|^2}{\Delta^2} + 2\frac{\|H''\|}{\Delta}\,\mathrm{d}s\Big). \tag{63}$$

As in the proof of 4, the second term is bounded by $\epsilon/2$. Since all the terms of equation (62) are positive, we have $\int_0^1 2\frac{\|H'\|}{\Delta}\,\mathrm{d}s \le \frac{\epsilon\lambda}{8}$. Then we bound

$$(2\delta + \delta^2)\Big(\int_0^1 2\frac{\|H'\|}{\Delta}\,\mathrm{d}s + \lambda\Big) \le 3\delta\big(\frac{\epsilon\lambda}{8} + \lambda\big) \tag{64}$$

$$\le \frac{27}{8}\delta\lambda(\epsilon + 1) \le \frac{27}{8}\delta\lambda \le \frac{\epsilon}{2}. \tag{65}$$

Finally we consider the query complexity and calculate

$$Q = 3\int_0^1 \Big(\frac{e\alpha t_0}{2\Delta} + \log\big(\frac{2c}{\delta}\big) + 1\Big)\lambda\,\mathrm{d}s \tag{66}$$

$$\le \lambda\Big(e\alpha t_0\frac{3}{2}\int_0^1 \frac{1}{\Delta}\,\mathrm{d}s + 3\log\big(\frac{27c}{2\epsilon}\big) + \log(\lambda) + 1\Big). \tag{67}$$

◀

▶ **Theorem 16.** *Under the assumptions in 1.1.2, we additionally assume that there exists $0 \le q \le 1$ and B_1, B_2 such that $\int_0^1 \frac{1}{\Delta^{1+q}}\,\mathrm{d}s \le B_1\Delta_m^{-q}$ and $\int_0^1 \frac{1}{\Delta^{2-q}}\,\mathrm{d}s \le B_2\Delta_m^{q-1}$ for all instances of the problem. Then algorithm 2 produces the target state with a fidelity of at least $1 - \epsilon$ if*

$$\lambda = \epsilon^{-1}\frac{2C}{\Delta^q\Delta_m^{1-q}} \tag{68}$$

$$\delta = \frac{2\epsilon}{15\lambda} \tag{69}$$

where $C := 2\sup_{s\in[0,1]}\Big(2\|H'(s)\| + 4\|H'(s)\|^2 B_2 + \|H''(s)\| + q|\Delta'(s)|\,\|H'(s)\| B_2\Big)$.

If, in addition, there exists a constant B_3 such that $\int_0^1 \frac{1}{\Delta^{2q}}\,ds \leq B_3 \Delta_m^{1-2q}$ and the Hamiltonian simulation of 12 is used, this gives a query complexity of

$$Q \leq \frac{1}{\epsilon\Delta_m}\Big(12C\log(\epsilon^{-1}) + 3e\alpha t_0 C B_1 + 6\log(15c)C + 12C^2 B_3\Big).$$

Proof. As before, we need to bound the inequality in lemma 13. Everything except the first term has already been bounded in the proof of theorem 5 to be less than $\epsilon/2$. (Notice that we are taking λ to be twice the rate specified in theorem 5.)

We now need to show that the first term in the inequality in lemma 13 can be bounded by $\epsilon/2$. Indeed, we calculate

$$\int_0^1 (2\delta + \delta^2)\Big(2\frac{\|H'\|}{\Delta} + \lambda(s)\Big)\,ds \leq \int_0^1 3\delta\Big(\frac{C}{2\Delta} + \lambda\Big)\,ds \tag{70}$$

$$= \int_0^1 \frac{2\epsilon}{5\lambda}\Big(\frac{C}{2\Delta} + \lambda\Big)\,ds \tag{71}$$

$$= \frac{2\epsilon}{5}\Big(1 + \int_0^1 \epsilon\frac{\Delta_m^{q-1}}{4\Delta^{q-1}}\,ds\Big) \tag{72}$$

$$\leq \frac{2\epsilon}{5}\Big(1 + \frac{\epsilon}{4}\Big) \leq \frac{\epsilon}{2}. \tag{73}$$

Finally we consider the query complexity and calculate, using the fact that $\log(x)+1 \leq x$ for all positive x,

$$Q = 3\int_0^1 \Big(\frac{e\alpha t_0}{2\Delta} + \log\Big(\frac{2c}{\delta}\Big) + 1\Big)\lambda\,ds \tag{74}$$

$$= 3\int_0^1 \Big(\frac{e\alpha t_0}{2\Delta} + \log\Big(\frac{15c}{\epsilon}\Big) + \log(\epsilon\lambda) + 1\Big)\lambda\,ds \tag{75}$$

$$\leq 3\int_0^1 \Big(\frac{e\alpha t_0}{2\Delta}\lambda + \log\Big(\frac{15c}{\epsilon^2}\Big)\lambda + \epsilon\lambda^2\Big)\,ds. \tag{76}$$

We bound each term separately. First

$$3\int_0^1 \frac{e\alpha t_0}{2\Delta}\lambda\,ds = \frac{3e\alpha t_0 C}{\epsilon\Delta_m^{1-q}}\int_0^1 \frac{1}{\Delta^{q+1}}\lambda\,ds \tag{77}$$

$$\leq \frac{3e\alpha t_0 C}{\epsilon\Delta_m^{1-q}}B_1\Delta_m^{-q} = \frac{3e\alpha t_0 C B_1}{\epsilon\Delta_m}. \tag{78}$$

Next

$$3\int_0^1 \log\Big(\frac{15c}{\epsilon^2}\Big)\lambda\,ds = 3\log\Big(\frac{15c}{\epsilon^2}\Big)\epsilon^{-1}\int_0^1 \frac{2C}{\Delta^q\Delta_m^{1-q}}\,ds \tag{79}$$

$$\leq \log\Big(\frac{15c}{\epsilon^2}\Big)\epsilon^{-1}\frac{6C}{\Delta_m}. \tag{80}$$

Finally

$$3\int_0^1 \epsilon\lambda^2\,ds = \frac{12C^2}{\epsilon\Delta_m^{2-2q}}\int_0^1 \frac{1}{\Delta^{2q}}\,ds \tag{81}$$

$$\leq \frac{12C^2 B_3}{\epsilon\Delta_m^{2-2q}}\Delta_m^{1-2q} = \frac{12C^2 B_3}{\epsilon\Delta_m}. \tag{82}$$

Putting everything together yields the query complexity. ◀

▶ **Corollary 17.** *If $\int_0^1 \frac{1}{\Delta^p}\,\mathrm{d}s = O(\Delta_m^{1-p})$ holds for all $p > 1$, $|\Delta'| = O(1)$, $\|H'\| = O(1)$ and $\|H''\| = O(1)$, then algorithm 2 with the Hamiltonian simulation of 12 and the parameters of 16 for some $1/2 < q < 1$, produces a state with fidelity larger than $1 - \epsilon$ using a number of queries that scales as $O\big(\Delta_m^{-1}\epsilon^{-1}\log(\epsilon^{-1})\big)$.*

The asymptotic scaling in the error ϵ is slightly worse here, since there is an extra logarithmic factor, but this is not an issue if we want to apply eigenstate filtering. With eigenstate filtering the scaling in the error is still $O(\log(1/\epsilon))$.

C   Gap properties

C.1   The gap in the Grover problem

For the Grover problem we have the following gap:

$$\Delta(s) = \sqrt{1 - 4(1 - \frac{M}{N})s(1 - s)}. \tag{83}$$

We can set $\Delta_m = \min_{s \in [0,1]} \Delta(s) = \sqrt{M/N}$. We provide a proof of lemma 10.

▶ **Lemma.** *For all $p > 1$ and Δ given by (83), we have*

$$\int_0^1 \frac{1}{\Delta(s)^p}\,\mathrm{d}s = O\big(\sqrt{N/M}^{p-1}\big) = O\big(\Delta_m^{1-p}\big), \tag{84}$$

and, for $p = 1$,

$$\int_0^1 \frac{1}{\Delta(s)}\,\mathrm{d}s = O\big(\log(N/M)\big). \tag{85}$$

Proof. We note that $\Delta(s)$ is symmetric about $s = 1/2$. It is also strictly decreasing on $[0, 1/2]$, going from 1 to a minimum of $\sqrt{M/N}$. So we can write

$$\int_0^1 \frac{1}{\Delta(s)^p}\,\mathrm{d}s = 2\int_0^{1/2} \frac{1}{\Delta(s)^p}\,\mathrm{d}s \tag{86}$$

$$= 2\Big(\int_0^{1/2 - \sqrt{M/N}} \frac{1}{\Delta(s)^p}\,\mathrm{d}s + \int_{1/2 - \sqrt{M/N}}^{1/2} \frac{1}{\Delta(s)^p}\,\mathrm{d}s \Big). \tag{87}$$

Since Δ has a minimum of $\sqrt{M/N}$, we can bound the second integral by

$$\int_{1/2 - \sqrt{M/N}}^{1/2} \frac{1}{\Delta(s)^p}\,\mathrm{d}s \leq \sqrt{\frac{M}{N}} \Big(\frac{1}{\min_{s \in [0,1]} \Delta(s)} \Big)^p = \frac{\sqrt{M/N}}{\sqrt{M/N}^p} = \sqrt{N/M}^{p-1}.$$

For the first integral, we write

$$\int_0^{1/2 - \sqrt{M/N}} \frac{1}{\Delta(s)^p}\,\mathrm{d}s = \int_1^{\Delta\left(1/2 - \sqrt{M/N}\right)} \frac{1}{\Delta^p}\frac{\mathrm{d}s}{\mathrm{d}\Delta}\,\mathrm{d}\Delta \tag{88}$$

$$= \int_{\Delta\left(1/2 - \sqrt{M/N}\right)}^1 \frac{1}{\Delta^p}\Big(-\frac{\mathrm{d}s}{\mathrm{d}\Delta} \Big)\,\mathrm{d}\Delta. \tag{89}$$

We can invert (42) to obtain $s = \frac{1}{2} - \frac{1}{2}\sqrt{1 - \frac{1 - \Delta^2}{1 - N/M}}$. Then we have

$$-\tfrac{\mathrm{d}s}{\mathrm{d}\Delta} = \frac{\Delta}{2\sqrt{(1 - M/N)(\Delta^2 - M/N)}}. \tag{90}$$

We now calculate

$$\Delta\Big(\frac{1}{2} - \sqrt{\frac{M}{N}}\Big) = \sqrt{\frac{M}{N}}\sqrt{5 - 4\frac{M}{N}} \geq 2\sqrt{\frac{M}{N}},$$

assuming $M/N \leq 1/4$. So

$$\int_0^{1/2-\sqrt{M/N}} \frac{1}{\Delta^p}\,\mathrm{d}s \leq \int_{2\sqrt{\frac{M}{N}}}^1 \frac{1}{\Delta^p}\Big(-\frac{\mathrm{d}s}{\mathrm{d}\Delta}\Big)\,\mathrm{d}\Delta \tag{91}$$

$$= \int_{2\sqrt{\frac{M}{N}}}^1 \frac{1}{\Delta^p}\frac{\Delta}{2\sqrt{(1-M/N)(\Delta^2 - M/N)}}\,\mathrm{d}\Delta \tag{92}$$

$$\leq \int_{2\sqrt{\frac{M}{N}}}^1 \frac{1}{\Delta^p}\frac{\Delta}{2\sqrt{(1-M/N)(\Delta^2 - \Delta^2/4)}}\,\mathrm{d}\Delta \tag{93}$$

$$= \frac{1}{\sqrt{3(1-M/N)}} \int_{2\sqrt{\frac{M}{N}}}^1 \frac{1}{\Delta^p}\,\mathrm{d}\Delta. \tag{94}$$

Now $\frac{1}{\sqrt{3(1-M/N)}}$ is $O(1)$ and $\int_{2\sqrt{\frac{M}{N}}}^1 \frac{1}{\Delta^p}\,\mathrm{d}\Delta = \Big[\frac{1}{(p-1)\Delta^{p-1}}\Big]_{2\sqrt{M/N}}^1$ is $O\big(\sqrt{N/M}^{p-1}\big)$, if $p > 1$. If $p = 1$, then it is $O\big(\log\sqrt{N/M}\big)$. ◀

C.2 The gap in QLSP

For the quantum linear system problem we have the following bound on the gap:

$$\Delta(s) = \sqrt{(1-s)^2 + \Big(\frac{s}{\kappa}\Big)^2}. \tag{95}$$

If κ is large enough, then we can take $\Delta_m \coloneqq \frac{1}{2\kappa} \leq \sqrt{\frac{1}{\kappa^2+1}} = \min_{s\in[0,1]}\Delta(s)$. We provide a proof of lemma 11.

▶ **Lemma.** *For all $p > 1$, we have*

$$\int_0^1 \frac{1}{\Delta(s)^p}\,\mathrm{d}s = O\big(\kappa^{p-1}\big) = O\big(\Delta_m^{1-p}\big), \tag{96}$$

and, for $p = 1$,

$$\int_0^1 \frac{1}{\Delta(s)}\,\mathrm{d}s = O\big(\log(\kappa)\big). \tag{97}$$

Proof. We note that $\Delta(s)$ is strictly decreasing on $\Big[0, 1 - \frac{1}{\kappa^2+1}\Big]$, going from 1 to a minimum of $\sqrt{\frac{1}{\kappa^2+1}}$. So we can write

$$\int_0^1 \frac{1}{\Delta(s)^p}\,\mathrm{d}s = \int_0^{1-\frac{1}{\kappa^2+1}} \frac{1}{\Delta(s)^p}\,\mathrm{d}s + \int_{1-\frac{1}{\kappa^2+1}}^1 \frac{1}{\Delta(s)^p}\,\mathrm{d}s.$$

Since Δ has a minimum of $\sqrt{\frac{1}{\kappa^2+1}}$, we can bound the second integral by

$$\int_{1-\frac{1}{\kappa^2+1}}^1 \frac{1}{\Delta(s)^p}\,\mathrm{d}s \leq \frac{1}{\kappa^2+1}\Big(\frac{1}{\min_{s\in[0,1]}\Delta(s)}\Big)^p = \frac{1}{\kappa^2+1}\big(\kappa^2+1\big)^{p/2} = \big(\kappa^2+1\big)^{p/2-1}.$$

For the first integral, we write

$$\int_0^{1-\frac{1}{\kappa^2+1}} \frac{1}{\Delta^p} \, \mathrm{d}s = \int_1^{\Delta\left(1-\frac{1}{\kappa^2+1}\right)} \frac{1}{\Delta^p} \frac{\mathrm{d}s}{\mathrm{d}\Delta} \, \mathrm{d}\Delta \tag{98}$$

$$= \int_{\Delta\left(1-\frac{1}{\kappa^2+1}\right)}^1 \frac{1}{\Delta^p} \left(-\frac{\mathrm{d}s}{\mathrm{d}\Delta}\right) \mathrm{d}\Delta \tag{99}$$

$$= \int_{\sqrt{\frac{1}{\kappa^2+1}}}^1 \frac{1}{\Delta^p} \left(-\frac{\mathrm{d}s}{\mathrm{d}\Delta}\right) \mathrm{d}\Delta. \tag{100}$$

We can invert (95) on $\left[0, 1-\frac{1}{\kappa^2+1}\right]$ to obtain $s = \frac{\kappa^2}{\kappa^2+1}(1-\Delta)$. Then we have

$$-\frac{\mathrm{d}s}{\mathrm{d}\Delta} = \frac{\kappa^2}{\kappa^2+1}, \tag{101}$$

so

$$\int_0^{1-\frac{1}{\kappa^2+1}} \frac{1}{\Delta^p} \, \mathrm{d}s = \int_{\sqrt{\frac{1}{\kappa^2+1}}}^1 \frac{1}{\Delta^p} \frac{\kappa^2}{\kappa^2+1} \, \mathrm{d}\Delta \tag{102}$$

$$= \frac{\kappa^2}{\kappa^2+1} \left(\frac{1}{(p-1)\Delta^{p-1}}\right)\Big|_{\Delta=1}^{\Delta=\sqrt{\frac{1}{\kappa^2+1}}} \tag{103}$$

$$= O(\kappa^{p-1}). \tag{104}$$

If $p = 1$, then the integral is $O\big(\log(\kappa)\big)$.   ◄