Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations

Andris Ambainis*

Abstract

We present two new quantum algorithms. Our first algorithm is a generalization of amplitude amplification to the case when parts of the quantum algorithm that is being amplified stop at different times.

Our second algorithm uses the first algorithm to improve the running time of Harrow et al. algorithm for solving systems of linear equations from $O(\kappa^2 \log N)$ to $O(\kappa \log^3 \kappa \log N)$ where κ is the condition number of the system of equations.

1 Introduction

Solving large systems of linear equations is a very common problem in scientific computing, with many applications. Until recently, it was thought that quantum algorithms cannot achieve a substantial speedup for this problem, because the coefficient matrix A is of size N^2 and it may be necessary to access all or most of coefficients in A to compute x - which requires time $\Omega(N^2)$.

Recently, Harrow, Hassidim and Lloyd [5] discovered a surprising quantum algorithm that allows to solve systems of linear equations in time $O(\log N)$ - in an unconventional sense. Namely, the algorithm of [5] generates the quantum state $|x\rangle = \sum_{i=1}^{N} x_i|i\rangle$ with the coefficients x_i being equal to the values of variables in the solution $x = (x_1, x_2, \ldots, x_N)$ of the system $Ax = b$.

The Harrow-Hassidim-Lloyd algorithm among the most interesting new results in quantum algorithms, because systems of linear equations have many applications in all fields of science. For example, this algorithm has been used to design quantum algorithms for solving differential equations [7, 3].

Besides N, the running time of the algorithms for systems of linear equations (both classical and quantum algorithms) depends on another parameter κ, the condition number of matrix A. The condition number is defined as the ratio

*Faculty of Computing, University of Latvia, Raina bulv. 19, Riga, LV-1586, Latvia, ambainis@lu.lv. Supported by ESF project 1DP/1.1.1.2.0/09/APIA/VIAA/044, FP7 Marie Curie Grant PIRG02-GA-2007-224886 and FP7 FET-Open project QCS.

between the largest and the smallest singular value of A: $\kappa = \max_{i,j} \frac{|\mu_i|}{|\mu_j|}$ where μ_i are the singular values of A.

In the case of sparse classical matrices, the best classical algorithm runs in time $O(\sqrt{\kappa}N)$ [9] while the HHL quantum algorithm runs in time $O(\kappa^2 \log N)$, with an exponentially better dependence on N but worse-than-classical dependence on κ.

In this paper, we present a better quantum algorithm, with the running time $O(\kappa \log^3 \kappa \log N)$. To construct our algorithm, we introduce a new tool, the *variable-time quantum amplitude amplification* which allows to efficiently amplify the success probability of quantum algorithms in which some branches of the computation stop earlier than other branches. The conventional amplitude amplification [4] would wait for all branches to stop - possibly resulting in a substantial inefficiency. Our new algorithm amplifies the success probability in multiple stages and takes advantage of the parts of computation which stop earlier. We expect that this new method will be useful for building other quantum algorithms.

The dependence of our quantum algorithm for solving systems of linear equations on κ is essentially optimal. Harrow et al. [5] show that, unless $BQP = PSPACE$, time of $\Omega(\kappa^{1-o(1)})$ is necessary for generating the state $|x\rangle$ that describes the solution of the system.

2 Overview of main results

2.1 Variable time amplitude amplification

Informally, our result is as follows. Consider a quantum algorithm $\mathcal{A}$ which may stop at one of several times $t_1, \ldots, t_m$. (In the case of systems of linear equations, these times corresponding to m runs of eigenvalue estimation with increasing precision and increasing number of steps.) To indicate the outcome, $\mathcal{A}$ has an extra register O with 3 possible values: 0, 1 and 2. 1 indicates the outcome that should be amplified. 0 indicates that the computation has stopped at this branch but did not the desired outcome 1. 2 indicates that the computation at this branch has not stopped yet.

Let p_i be the probability of the algorithm stopping at time t_i (with either the outcome 0 or outcome 1). The average stopping time of $\mathcal{A}$ (the l_2 average) is

$$T_{av} = \sqrt{\sum_i p_i t_i^2}.$$

T_{max} denotes the maximum running time of the algorithm (which is equal to t_m). Let

$$\alpha_{good}|1\rangle_O|\psi_{good}\rangle + \alpha_{bad}|0\rangle_O|\psi_{bad}\rangle$$

be the algorithm's output state after all branches of the computation have stopped. Our goal is to apply $|\psi_{good}\rangle$ with a high probability. Let $p_{succ} = |\alpha_{good}|^2$ be the probability of obtaining this state via algorithm $\mathcal{A}$.

Our main result is

Theorem 1 *We can construct a quantum algorithm $\mathcal{A}'$ invoking $\mathcal{A}$ several times, for total time*

$$O\left(T_{max}\sqrt{\log T_{max}} + \frac{T_{av}}{\sqrt{p_{succ}}}\log^{1.5} T_{max}\right)$$

that produces a state $\alpha|1\rangle \otimes |\psi_{good}\rangle + \beta|0\rangle \otimes |\psi'\rangle$ with probability $|\alpha|^2 \geq 1/2$ as the output[1]

The algorithm $\mathcal{A}'$ is optimal, up to the factor of $\log T_{max}$. If the algorithm $\mathcal{A}$ has just one stopping time $T = T_{av} = T_{max}$, then amplitude amplification cannot be performed with fewer than $O(\frac{T}{\sqrt{p_{succ}}})$ steps. Thus, the term of $\frac{T_{av}}{\sqrt{p_{succ}}}$ is necessary. The term T_{max} is also necessary because, in some branch of computation, $\mathcal{A}$ can run for T_{max} steps.

By repeating $\mathcal{A}'$ $O(\log \frac{1}{\epsilon})$ times, we can obtain $|\psi_{good}\rangle$ with a probability at least $1 - \epsilon$.

More details are given in section 3. First, in subsection 3.1, we give a precise definition of how a quantum algorithm could stop at different times. Then, in subsections 3.2 and 3.3, we give a proof of Theorem 1.

2.2 Systems of linear equations

We consider solving a system of linear equations $Ax = b$ where $A = (a_{ij})_{i,j\in[N]}$, $x = (x_i)_{i\in[N]}$, $b = (b_i)_{i\in[N]}$. We assume that A is Hermitian. As shown in [5], this assumption is without the loss of generality. Similarly to [5], we also assume that all eigenvalues λ of A satisfy $\frac{1}{\kappa} \leq \lambda \leq 1$ for some known κ. We can then transform the state $|b\rangle = \sum_{i=1}^n b_i|i\rangle$ into $|x\rangle = \sum_{i=1}^n x_i|i\rangle$ as follows:

1. If, in terms of eigenvectors $|v_i\rangle$ of A, we have $|b\rangle = \sum_i c_i|v_i\rangle$, then $|x\rangle = \sum_i \frac{c_i}{\lambda_i}|v_i\rangle$.

2. By eigenvalue estimation, we can create the state $|b'\rangle = \sum_i c_i|v_i\rangle|\tilde{\lambda}_i\rangle$ where $\tilde{\lambda}_i$ are the estimates of the true eigenvalues.

3. We then create the state

$$|b''\rangle = \sum_i c_i|v_i\rangle|\tilde{\lambda}_i\rangle\left(\frac{1}{\kappa\tilde{\lambda}_i}|1\rangle + \sqrt{1 - \frac{1}{\kappa^2\tilde{\lambda}^2}}|0\rangle\right). \quad (1)$$

   Conditional on the last bit being 1, the rest of state is $\sum_i \frac{c_i}{\tilde{\lambda}_i}|v_i\rangle|\tilde{\lambda}_i\rangle$ which can be turned into an approximation of $|x\rangle$ by running eigenvalue estimation in reverse and uncomputing $\tilde{\lambda}_i$.

[1] The first bit of the output state indicates whether we have the desired state $|\psi_{good}\rangle$ or not. Since $|\alpha|^2 \geq 1/2$, we get $|\psi_{good}\rangle$ with probability at least 1/2.

4. We then the part of state which has the last qubit equal to 1 (using amplitude amplification) and obtain a good approximation of $|x\rangle$ with a high probability.

Theorem 2 *[5] Let C be such that the evolution of the Hamiltonian H for time T can be simulated in time $C\min(T,1)$. Then, we can generate $|\psi'\rangle$ satisfying $\|\psi-\psi'\|\leq\epsilon$ in time $(\frac{C\kappa^2}{\epsilon})$.*

The main term in the running time, κ^2 is generated as a product of two κ's. First, for $\|\psi-\psi'\|\leq\epsilon$, we need $|\lambda_i-\tilde{\lambda}_i|=O(\epsilon\tilde{\lambda}_i)$. Since $\lambda_i=\Omega(1/\kappa)$, this means $|\lambda_i-\tilde{\lambda}_i|=O(\frac{\epsilon}{\kappa})$. To estimate λ_i within error $O(\frac{\epsilon}{\kappa})$, we need to run H for time $O(\frac{\kappa}{\epsilon})$. Second, for amplitude amplification, we may need to repeat the algorithm generating $|b''\rangle$ $O(\kappa)$ times - resulting in the total running time $O(\kappa^2/\epsilon)$.

For eigenvalue estimation, the worst case is when all of most of λ_i are small (of order $\Theta(1/\kappa)$). Then, $|\lambda_i-\tilde{\lambda}_i|=\Theta(\frac{\epsilon}{\kappa})$. and eigenvalue estimation with the right precision indeed requires time $\Theta(\frac{\kappa}{\epsilon})$.

For amplitude amplification, the worst case is if most or all of λ_i are large (constant). Then, the coefficients $\frac{1}{\kappa\tilde{\lambda}_i}$ can be of order $\Theta(1/\kappa)$ and $\Theta(\kappa)$ repetitions are required for amplitude amplification.

We now observe that the two $\Theta(\kappa)$'s appear in the opposite cases. One of them appears when λ_i is small ($\lambda_i\approx\kappa$) but the other appears when λ_i is large ($\lambda_i\approx 1$).

If all eigenvalues are of roughly similar magnitude (e.g., $\lambda\in[a,2a]$ for some a), the running time becomes $O(\kappa/\epsilon)$ because we can do eigenvalue estimation in time to error ϵa in $O(1/a\epsilon)$ and, for eigenvalue amplification, it suffices to repeat the generation of $|b''\rangle$ $O(\kappa a)$ times (since the amplitude of 1 in the last qubit of $|b'\rangle$ is at least $\frac{1}{\kappa a}$ for every v_i). Thus, the running time is

$$O\left(\frac{1}{a\epsilon}\right)\cdot O(\kappa a)=O\left(\frac{\kappa}{\epsilon}\right).$$

In the general case (when the eigenvalues λ_i can range from κ to 1), we run the eigenvalue estimation several times. We start by running it for $O(1)$ steps. If we see that the estimate $\tilde{\lambda}_i$ for the eigenvalue is such that the allowed error $O(\epsilon\tilde{\lambda}_i)$ is more than the expected error of the current run of eigenvalue estimation, we stop. Otherwise, we run eigenvalue estimation again, doubling its running time. This doubles the precision achieved by eigenvalue estimation. We continue this until the precision of current estimate becomes better than the allowed error of $O(\epsilon\tilde{\lambda}_i)$.

This results in a quantum algorithm in which some branches of computation (corresponding to eigenvectors with larger eigenvalues λ_i) terminate earlier than others. We apply our variable-time amplitude amplification to this quantum algorithm. This gives us

Theorem 3 *Let C be such that the evolution of the Hamiltonian H for time T can be simulated in time $C\min(T,1)$. Then, we can generate $|\psi'\rangle$ satisfying*

$\|\psi - \psi'\| \leq \epsilon$ *in time*

$$O\left(\frac{C\kappa \log^3 \kappa}{\epsilon^3} \log^2 \frac{1}{\epsilon}\right).$$

3 Variable-time amplitude amplification

3.1 Model

How can a quantum algorithm have different branches of computation stopping at different times? We start by giving a precise definition of that.

We require the state space of $\mathcal{A}$ to be of the form $\mathcal{H} = \mathcal{H}_o \otimes \mathcal{H}_c$ be the Hilbert space of $\mathcal{A}$, consisting of the 0-1-2 valued outcome register $\mathcal{H}_o$ and the rest of the Hilbert space $\mathcal{H}_c$. Let $|\psi_1\rangle, \ldots, |\psi_m\rangle$ be the states of $\mathcal{A}$ at times $t_1, \ldots, t_m$. We insist on the following consistency requirements.

1. For each $i \in \{1, \ldots, m\}$, we can define a subspace $\mathcal{H}_i$ of $\mathcal{H}_o$ in which the computation has stopped. Those subspaces satisfy

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \ldots \subseteq \mathcal{H}_m = \mathcal{H}_o.$$

2. The state $|\psi_i\rangle$ can be expressed as

$$|\psi_i\rangle = \alpha_{i,0}|0\rangle \otimes |\psi_{i,0}\rangle + \alpha_{i,1}|1\rangle \otimes |\psi_{i,1}\rangle + \alpha_{i,2}|2\rangle \otimes |\psi_{i,2}\rangle,$$

   with $|\psi_{i,0}\rangle \in \mathcal{H}_i$, $|\psi_{i,1}\rangle \in \mathcal{H}_i$ and $|\psi_{i,2}\rangle \in \mathcal{H}_o \cap (\mathcal{H}_i)^\perp$. (When $i = m$, we have $|\psi_{m,0}\rangle = |\psi_{bad}\rangle$, $|\psi_{m,1}\rangle = |\psi_{good}\rangle$, $|\psi_{m,2}\rangle = \overrightarrow{0}$.)

3.

$$P_{H_i}|\psi_{i+1,0}\rangle = |\psi_{i,0}\rangle \text{ and } P_{H_i}|\psi_{i+1,1}\rangle = |\psi_{i,1}\rangle.$$

The success probability of $\mathcal{A}$ is $p_{succ} = |\alpha_{m,1}|^2$. We also define $p_{succ,i} = |\alpha_{i,1}|^2$, the probability of $\mathcal{A}$ succeeding before time t_i. The probability of $\mathcal{A}$ stopping at time t_i or earlier is

$$p_{stop,\leq i} = |\alpha_{i,0}|^2 + |\alpha_{i,1}|^2.$$

The probability of $\mathcal{A}$ stopping at exactly time t_i is $p_{stop,1} = p_{stop,\leq 1}$ for $i = 1$ and $p_{stop,i} = p_{stop,\leq i} - p_{stop,\leq i-1}$ for $i > 1$. We will also use the probability of $\mathcal{A}$ stopping later than time t_i, defined as

$$p_{stop,>i} = |\alpha_{i,2}|^2 = 1 - p_{stop,\leq i}.$$

The average stopping time of $\mathcal{A}$ (the l_2 average) is

$$T_{av} = \sqrt{\sum_i p_i t_i^2}.$$

The maximum stopping time of $\mathcal{A}$ is $T_{max} = t_m$. Our goal is to amplify the success probability to $\Omega(1)$, by running $\mathcal{A}$ for time $O\left(T_{max} + \frac{T_{av}}{\sqrt{p_{succ}}} \log T_{max}\right)$.

3.2 Tools

Our variable-time amplitude amplification uses two subroutines. Thr first is a result by Aaronson and Ambainis [1] who gave a tighter analysis of the same algorithm:

Lemma 1 *[1] Let $\mathcal{A}$ be a quantum algorithm that outputs a correct answer and a witness with probability*[2] *$\delta \leq \epsilon$ where ϵ is known. Furthermore, let*

$$m \leq \frac{\pi}{4 \arcsin \sqrt{\epsilon}} - \frac{1}{2}. \tag{2}$$

Then, there is an algorithm $\mathcal{A}'$ which uses $2m+1$ calls to $\mathcal{A}$ and $\mathcal{A}^{-1}$ and outputs a correct answer and a witness with probability

$$\delta_{new} \geq \left(1 - \frac{(2m+1)^2}{3}\delta\right)(2m+1)^2\delta. \tag{3}$$

The distinction between this lemma and the standard amplitude amplification is as follows. The standard amplitude amplification increases the probability from δ to $\Omega(1)$ in $2m+1 = O(\frac{1}{\sqrt{\delta}})$ repetitions. In other words, $2m+1$ repetitions increase the success probability $\Omega((2m+1)^2)$ times. Lemma 1 achieves an increase of almost $(2m+1)^2$ times, without the big-Ω factor. This is useful if we have an algorithm with k levels of amplitude amplification nested one inside another. Then, with the usual amplitude amplification, a big-Ω constant of c would result in a c^k factor in the running time. Using Lemma 1 avoids that.

The second is a version of amplitude estimation from [2].

Theorem 4 *[4, 2] There is a procedure* **Estimate**$(\mathcal{A}, c, p, k)$ *which, given a constant c, $0 < c \leq 1$ and a quantum algorithm $\mathcal{A}$ (with the promise that the probability ϵ that the algorithm $\mathcal{A}$ outputs 1 is either 0 or at least a given value p) outputs an estimate $\tilde{\epsilon}$ of the probability ϵ such that, with probability at least $1 - \frac{1}{2^k}$, we have*

(i) $|\epsilon - \tilde{\epsilon}| < c\tilde{\epsilon}$ *if* $\epsilon \geq p$;

(ii) $\tilde{\epsilon} = 0$ *if* $\epsilon = 0$.

The procedure **Estimate**$(\mathcal{A}, c, p, k)$ *uses the expected number of*

$$\Theta\left(k\left(1 + \log\log\frac{1}{p}\right)\sqrt{\frac{1}{\max(\epsilon, p)}}\right)$$

evaluations of $\mathcal{A}$.

[2] [1] requires the probability to be exactly ϵ but the proof works without changes if the probability is less than the given ϵ.

3.3 The state generation algorithm

We now describe our state generation algorithm. Without the loss of generality, we assume that the stopping times of $\mathcal{A}$ are $t_i = 2^i$ for $i \in \{0, \ldots, m\}$ for some m. We present a sequence of algorithms $\mathcal{A}_i$, with the algorithm $\mathcal{A}_i$ generating an approximation of the state

$$|\psi_i'\rangle = \frac{\alpha_{i,1}}{\sqrt{|\alpha_{i,1}|^2 + |\alpha_{i,2}|^2}}|\psi_{i,1}\rangle + \frac{\alpha_{i,2}}{\sqrt{|\alpha_{i,1}|^2 + |\alpha_{i,2}|^2}}|\psi_{i,2}\rangle,$$

in the following sense: the algorithm $\mathcal{A}_i$ outputs a state

$$|\psi_i''\rangle = \sqrt{r_i}|\psi_i'\rangle + \sqrt{1 - r_i}|0\rangle \otimes |\phi_i\rangle \tag{4}$$

for some $|\phi_i\rangle$ and some r_i satisfying $r_i \geq 1/9m$. (To avoid the problem with nested amplitude amplification described in section 3.2, we only require $r_i \geq 1/9m$ instead of $r_i = \Omega(1)$.)

The algorithm $\mathcal{A}_i$ uses $\mathcal{A}_{i-1}$ as the subroutine. It is defined in two steps. First, we define an auxiliary algorithm $\mathcal{B}_i$.

1. If $i = 0$, B_i runs $\mathcal{A}$ for 1 step and outputs the output state of $\mathcal{A}$.
2. If $i > 0$, B_i runs $\mathcal{A}_{i-1}$ which outputs $|\psi_{i-1}''\rangle$. $\mathcal{B}_i$ then executes $\mathcal{A}$ for time steps from 2^{i-1} to 2^i on the parts of the state $|\psi_{i-1}''\rangle$ where the outcome register is 2 (the computation is not finished).

Algorithm 1: Algorithm $\mathcal{B}_i$

Let $p_i = \textbf{Estimate}(\mathcal{B}_i, c, \frac{1}{\kappa}, \log m + 5)$. Then, $\mathcal{A}_i$ is as follows.

1. If $p > \frac{1}{9m}$, $\mathcal{A}_i = \mathcal{B}_i$.
2. If $p \leq \frac{1}{9m}$, $\mathcal{A}_i = \textbf{Amplify}(B_i, k)$ for the smallest k satisfying $\frac{1}{9m} \leq (2k+1)^2 p \leq \frac{1}{m}$.

Algorithm 2: Algorithm $\mathcal{A}_i$

The overall algorithm $\mathcal{A}'$ is given as Algorithm 3.

We now analyze the running times of algorithms $\mathcal{A}_i$. Let T_i denote the running time of $\mathcal{A}_i$. Let r_i be as defined in equation (4) and let r_i' be a similar quantity for the output state of $\mathcal{B}_i$. Then, we have

Lemma 2

$$T_i \leq \left(1 + \frac{1}{3m-1}\right) \frac{\sqrt{r_i}}{\sqrt{r_i'}} \left(T_{i-1} + 2^{i-1}\right). \tag{5}$$

Proof: The running time of $\mathcal{B}_i$ is $T_{i-1} + 2^{i-1}$. If $\mathcal{A}_i = \mathcal{B}_i$, then the running time of $\mathcal{A}_i$ is the same and, also $r_i = r_i'$ (because the two algorithms output the same state). If $\mathcal{A}_i$ is an amplified version of $\mathcal{B}_i$, then:

1. Run **Estimate** to obtain $p_0 = \mathbf{Estimate}(\mathcal{B}_0, c, \frac{1}{\kappa}, \log m + 5)$.
2. For each $i = 1, 2, \ldots, m$:
   (a) Use p_{i-1} to define $\mathcal{A}_i$ and $\mathcal{B}_i$.
   (b) If $i < m$, run **Estimate** to obtain $p_i = \mathbf{Estimate}(\mathcal{B}_i, c, \frac{1}{\kappa}, \log m + 5)$.

   Amplify $\mathcal{A}_m$ to the success probability at least $1/2$ and output the output state of the amplified $\mathcal{A}_m$.

Algorithm 3: Algorithm $\mathcal{A}'$

1. The running time of $\mathcal{A}_i$ is $(2k+1)(T_{i-1} + 2^{i-1})$.
2. By Lemma 1, we have $r_i \geq (1 - \frac{1}{3m})(2k+1)^2 r'_i$ which implies

$$(2k+1) \leq (1 + \frac{1}{3m-1}) \frac{\sqrt{r_i}}{\sqrt{r'_i}}.$$

Applying (5) recursively, we get

$$T_m \leq \left(1 + \frac{1}{3m-1}\right)^m \sum_{i=1}^{m} \left(\prod_{j=i}^{m} \frac{\sqrt{r_i}}{\sqrt{r'_i}} \right) 2^{i-1}. \tag{6}$$

The first multiplier, $\left(1 + \frac{1}{3m-1}\right)^m$ can be upper-bounded by a constant. We now bound the product $\prod_{j=i}^{m} \frac{\sqrt{r_i}}{\sqrt{r'_i}}$.

Lemma 3

$$\prod_{j=i}^{m} \frac{\sqrt{r_i}}{\sqrt{r'_i}} \leq 3 \left(1 + \sqrt{\frac{p_{stop,>i}}{p_{succ}}}\right).$$

Proof: We consider the quantities

$$o_j = |\langle 1 \otimes \psi_{i,1} | \psi''_i \rangle|^2$$

for $j = i, i+1, \ldots, m$. For $j = i$, we have

$$o_i = r_i |\langle 1 \otimes \psi_{i,1} | \psi'_i \rangle|^2 = r_i \frac{|\alpha_{i,1}|^2}{|\alpha_{i,1}|^2 + |\alpha_{i,2}|^2} = r_i \frac{p_{succ,i}}{p_{succ,i} + p_{stop,>i}}. \tag{7}$$

For $j > i$, we have $o_j = o_{j-1} \frac{r_i}{r'_i}$ because amplification increases the probability of the "good" part of the state (which includes $|1 \otimes \psi_{i,1}\rangle$) $\frac{r_i}{r'_i}$ times. Finally, we have

$$o_m = r_m \frac{p_{succ,i}}{p_{succ}}$$

which follows similarly to (7). Putting all of this together, we have

$$\prod_{j=i}^{m} \frac{r_i}{r'_i} = \frac{o_m}{o_i} = \frac{r_m}{r_i} \cdot \frac{p_{succ,i} + p_{stop,>i}}{p_{succ,i}}.$$

By taking the square roots from both sides and observing that $\frac{r_m}{r_i}$ is at most 9 (because $r_m \leq \frac{1}{m}$ and $r_i \geq \frac{1}{9m}$), we get

$$\prod_{j=i}^{m} \frac{\sqrt{r_i}}{\sqrt{r'_i}} \leq 3\sqrt{1 + \frac{p_{stop,>i}}{p_{succ}}}.$$

The Lemma follows by using $\sqrt{1+x} \leq 1 + \sqrt{x}$. ∎

By applying Lemma 3 to each term in (6), we get

$$T_m \leq C \sum_{i=1}^{m} \left(1 + \sqrt{\frac{p_{stop,>i}}{p_{succ}}}\right) 2^{i-1} = C \sum_{i=1}^{m} 2^{i-1} + C \frac{\sum_{i=1}^{m} 2^{i-1} \sqrt{p_{stop,>i}}}{\sqrt{p_{succ}}}.$$

The first sum is upper bounded by $2^i = O(T_{max})$. For the second sum, in its numerator, we have

$$\sum_{i=1}^{m} 2^{i-1} \sqrt{p_{stop,>i}} = \sum_{i=1}^{m} \sqrt{2^{2i-2} p_{stop,>i}} \leq m T_{av} = T_{av} \log T_{max}.$$

Thus, the algorithm $\mathcal{A}_m$ runs in time

$$O\left(T_{max} + \frac{T_{av}}{\sqrt{p_{succ}}} \log T_{max}\right).$$

The algorithm $\mathcal{A}'$ amplifies $\mathcal{A}_m$ from a success probability of $r_m \geq \frac{1}{9m}$ to a success probability $\Omega(1)$. This increases the running time by a factor of $O(\sqrt{m}) = O(\sqrt{\log T_{max}})$.

4 Faster algorithm for solving systems of linear equations

4.1 Unique-answer eigenvalue estimation

For our algorithm, we need a version of eigenvalue estimation that is guaranteed to output exactly the same estimate with a high probability. The standard version of eigenvalue estimation [6, p. 118] runs $U = e^{-iH}$ up to 2^n times and, if the input is an eigenstate $|\psi\rangle : H|\psi\rangle = \lambda|\psi\rangle$, outputs $x \in \{0, \frac{\pi}{2^n}, \frac{2\pi}{2^n}, \ldots, \frac{(2^n-1)\pi}{2^n}\}$ with probability

$$p(x) = \frac{1}{2^{2n}} \frac{\sin^2 2^n(\lambda - x)}{\sin^2(\lambda - x)} \tag{8}$$

(equation (7.1.30) from [6]). We now consider an algorithm that runs the standard eigenvalue estimation k_{uniq} times and takes the most frequent answer x_{maj}.

Lemma 4 *For $k_{uniq} = O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$, we have*

1. *If $|\lambda - x| \leq \frac{1-\epsilon}{2^{n+1}}$, then $Pr[x_{maj} = x] \geq 1-\epsilon$.*
2. *If $\lambda \in [x + \frac{1-\epsilon}{2^{n+1}}, x + \frac{1+\epsilon}{2^{n+1}}]$, then $Pr[x_{maj} \in \{x, x+1\}] \geq 1-\epsilon$.*

Proof: In the first case, (8) is at least $(1+\epsilon)\frac{4}{\pi^2}$ for the correct x and less than $\frac{4}{\pi^2}$ for any other x. Repeating eigenvalue estimation $O(\frac{1}{\epsilon^2})$ times and taking the majority allows to distinguish the correct x with a fixed probability (say 3/4) and repeating it $O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$ times allows to determine the correct x with a probability at least $1-\epsilon$.

In the second case, the two values x and $x+1$ are output with probability at least $(1-\epsilon)\frac{4}{\pi^2}$ each. In contrast, for any other $y = \frac{m\pi}{2^n}$, $m \in \{0, 1, \ldots, 2^n - 1\}$, we have

$$|y - \lambda| \geq \frac{1-\epsilon}{2^{n+1}}\pi + \frac{1}{2^n}\pi = \frac{3-\epsilon}{2^{n+1}}\pi.$$

This implies

$$p(y) \leq \frac{1}{2^{2n}} \frac{1}{\sin^2 \frac{(3-\epsilon)}{2^{n+1}}\pi} = (1+o(1))\frac{4}{(3-\epsilon)^2\pi^2}.$$

Thus, there is a constant gap between $p(x)$ or $p(x+1)$ and $p(y)$ for any other y. In this case, taking majority of $O(\log\frac{1}{\epsilon})$ runs of eigenvalue estimation is sufficient to produce x or $x+1$ with a probability at least $1-\epsilon$. ∎

We refer to this algorithm as **UniqueEst**$(H, 2^n, \epsilon)$.

When we use unique-answer eigenvalue estimation as a subroutine in algorithm 5, we need the answer to be unique (as in the first case) and not one of two high-probability answers (as in the second case). To deal with that, we will replace H with $H + \frac{\delta\pi}{2^n}I$ for a randomly chosen $\delta \in [0, 1]$. The eigenvalue becomes $\lambda' = \lambda + \frac{\delta\pi}{2^n}$ and, with probability $1-\epsilon$,

$$\lambda' \in \left[\frac{x - \frac{1-\epsilon}{2}}{2^n}\pi, \frac{x + \frac{1-\epsilon}{2}}{2^n}\pi\right]$$

for some integer x. This allows to achieve the first case for all eigenvalues, except a small random fraction of them.

4.2 Main algorithm

We now show that Theorem 1 implies our main result, Theorem 3. We start by describing a variable running time Algorithm 4. This algorithm uses the following registers:

- The input register I which holds the input state $|x\rangle$ (and is also used for the output state);
- The outcome register O, with basis states $|0\rangle, |1\rangle$ and $|2\rangle$ (as described in the setup for variable-time amplitude amplification);

- The step register S, with basis states $|1\rangle$, $|2\rangle$, ..., $|2m\rangle$ (to prevent interference between various branches of computation).
- The estimation register E, which is used for eigenvalue estimation (which is a subroutine for our algorithm).

$\mathcal{H}_I$, $\mathcal{H}_O$, $\mathcal{H}_S$ and $\mathcal{H}_E$ denote the Hilbert spaces of the respective registers.

From now on, we refer to ϵ appearing in Theorem 3 as ϵ_{final}. ϵ without a subscript is an error parameter for subroutines of algorithm 4 (which we will choose at the end of the proof so that the overall error in the output state is at most ϵ_{final}).

Input: parameters $x_1, \ldots, x_m \in [0,1]$, Hamiltonian H.

1. Initialize O to $|2\rangle$, S to $|1\rangle$ and E to $|0\rangle$. Set $j = 1$.
2. Let $m = \lceil \log_2 \frac{\kappa}{\epsilon} \rceil$.
3. Repeat until $j > m$:

   Stage j:

   (a) Let $H' = H + \frac{x_j \pi}{2^j} I$. Using the registers I and S, run bf UniqueEst$(H', 2^j, \epsilon)$. Let λ' be the estimate output by **UniqueEst** and let $\lambda = \lambda' - \frac{x_j \pi}{2^j}$.

   (b) If $\epsilon\lambda > \frac{1}{2^{j+1}}$, perform the transformation

$$|2\rangle_O \otimes |1\rangle_S \to \frac{1}{\kappa\lambda}|1\rangle_O \otimes |2j\rangle_S + \sqrt{1 - \frac{1}{(\kappa\lambda)^2}}|0\rangle_O \otimes |2j\rangle_S. \quad (9)$$

   (c) Run **UniqueEst** in reverse, to erase the intermediate information.

   (d) Check if the register E is in the correct initial state $|0\rangle_E$. If not, apply $|2\rangle_O \otimes |1\rangle_S \to |0\rangle_O \otimes |2j+1\rangle_S$ on the outcome register O.

   (e) If the outcome register O is in the state $|2\rangle$, increase j by 1 and go to step 2.

Algorithm 4: State generation algorithm

Our main algorithm is Algorithm 5 which consists of applying variable-time amplitude amplification

We claim that, conditional on the output register being $|1\rangle_O$, the output state of Algorithm 4 is close to

$$|\psi_{ideal}\rangle = \sum_i \alpha_i |v_i\rangle_I \otimes \left(\frac{1}{\kappa\lambda_i} |1\rangle_O \otimes |2j_i\rangle_S \right). \quad (10)$$

Variable-time amplitude amplification then generates a state that is close to $\frac{|\psi_{ideal}\rangle}{\|\psi_{ideal}\|}$. Fourier transform in the last step of algorithm 5 then effectively erases

Input: Hamiltonian H.

1. Generate uniformly random $x_1, \ldots, x_m \in [0, 1]$.
2. Apply variable-time amplitude amplification to Algorithm 4, with H and $x_1, \ldots, x_m$ as the input.
3. Apply a transformation mapping $|2j\rangle_S \to |j\rangle_S$ to the S register. After that, apply Fourier transform F_m to the S register and measure. If the result is 0, output the state in the I register. Otherwise, stop without outputting a quantum state.

Algorithm 5: Main algorithm

the S register. Conditional on S being in $|0\rangle_S$ after the Fourier transform, the algorithm's output state is close to our desired output state $\frac{|x\rangle}{\|x\|}$, where

$$|x\rangle = \sum_i \alpha_i |v_i\rangle_I.$$

Finally, performing Fourier transform and measuring produces $|0\rangle_S$ with probability $1/m$. Because of that, the success probability of algorithm 5 needs to be amplified. This adds a factor of $O(\sqrt{m})$ to the running time, if we would like to obtain the result state with probability $\Omega(1)$ and a factor of $O(\sqrt{m}\log\frac{1}{\epsilon})$ if we would like to obtain it with probability at least $1-\epsilon$.

Approximation guarantees. We now give a formal proof that the output state of Algorithm 4 is close to the desired output state (10).

Let $|v_i\rangle$ be an eigenvector and λ_i be an eigenvalue. For each j, the unique-value eigenvalue estimation either outputs one estimate $\tilde{\lambda}_{i,j}$ or one of two estimates $\tilde{\lambda}_{i,j}$ and $\tilde{\lambda}_{i,j} - \frac{1}{2^j}$ with a high probability (at least $1-\epsilon$). Let j_i be the smallest j for which the estimate $\tilde{\lambda} = \tilde{\lambda}_{i,j}$ satisfies the condition $\epsilon\tilde{\lambda} \geq \frac{1}{2^{j+1}}$ in step 3b. We call v_i and λ_i *good* if, for $j = j_i$ the unique-value eigenvalue estimation outputs one estimate $\tilde{\lambda}_{i,j}$ with a high probability. Otherwise, we call λ_i *bad*. For both good and bad λ_i, we denote $\tilde{\lambda}_i = \tilde{\lambda}_{i,j_i}$.

Claim 1 *Let $j = j_i$. Then*

$$\frac{1}{\epsilon 2^{j+1}} \leq \tilde{\lambda}_i \leq \left(\frac{1}{\epsilon} + \frac{3}{2}\right)\frac{1}{2^j}.$$

Proof: The first inequality follows immediately. For the second inequality, since $j > j_i - 1$, we have

$$\tilde{\lambda}_{i,j-1} \leq \frac{1}{\epsilon 2^j}.$$

This means that the actual eigenvalue λ satisfies

$$\lambda \leq (1+\epsilon)\frac{1}{\epsilon 2^j} = \frac{1}{\epsilon 2^j} + \frac{1}{2^j}$$

and

$$\tilde{\lambda}_{i,j} \leq (1+\epsilon)\lambda \leq \frac{1}{\epsilon 2^j} + \frac{1}{2^j} + \frac{1}{2^{j+1}}.$$

∎

As a consequence to this lemma, we have

$$\frac{1}{\tilde{\lambda}_i} \geq \left(\frac{2}{2+3\epsilon}\right)\epsilon 2^j.$$

We claim that the part of final state Algorithm 4 that has $|1\rangle$ in the output register O is close to

$$|\psi'\rangle = \sum_i \alpha_i |v_i\rangle_I \otimes \left(\frac{1}{\kappa\tilde{\lambda}_i}|1\rangle_O \otimes |2j_i\rangle_S\right)$$

and $|\psi'\rangle$ is, in turn, close to the state $|\psi_{ideal}\rangle$ defined by equation (10).

The next two lemmas quantify these claims. Let

$$\delta = \sum_{i:\lambda_i \text{ bad}} |\alpha_i|^2 \frac{1}{\lambda_i^2}$$

quantify the size of the part of the state $|\psi'\rangle$ that consists of bad eigenvectors.

Lemma 5 *Let $|\psi\rangle$ be the output state of Algorithm 4 and let be the projection to the subspace where the outcome register O is in the state $|1\rangle$. Then, we have*

$$\|P_1|\psi\rangle - |\psi'\rangle\| \leq ((2m+37)\epsilon + 30\delta)\|\psi'\|.$$

Proof: In section 4.3. ∎

Lemma 6

$$\||\psi'\rangle - |\psi_{ideal}\rangle\| \leq \frac{2\epsilon}{1+2\epsilon}\|\psi_{ideal}\|.$$

Proof: In section 4.3. ∎

When $x_1, \ldots, x_m \in [0,1]$ are chosen uniformly at random, the probability of any given v_i being bad is of order $O(\epsilon)$. Thus, $E[\delta] = O(\epsilon)$ and

$$E\|P_1|\psi\rangle - |\psi_{ideal}\rangle\| = O(m\epsilon\|\psi_{ideal}\|)$$

with the expectation taken over the random choice of $x_1, \ldots, x_m \in [0,1]$.

To achieve an error of at most ϵ_{final}, we choose $\epsilon = \Theta(\epsilon_{final}/m)$.

Running time. We now bound the running time of Algorithm 4. We start with two lemmas bounding the average running time T_{av} and success probability p_{av}.

Lemma 7 *T_{av}, the l_2-average running time of Algorithm 4, is of the order*

$$O\left(\sqrt{\sum_i |\alpha_i|^2 2^{2j_i} k_{uniq}^2}\right). \tag{11}$$

where k_{uniq} is the quantity from Lemma 4.

Proof: In section 4.4. ∎

Lemma 8 *p_{succ}, the success probability of Algorithm 4, is*

$$\Omega\left(\sum_i |\alpha_i|^2 \frac{\epsilon^2 2^{2j_i}}{\kappa^2}\right). \tag{12}$$

Proof: In section 4.4. ∎

By dividing the two expressions above one by another, we get

Corollary 1

$$\frac{T_{av}}{\sqrt{p_{succ}}} = O\left(\frac{\kappa}{\epsilon} k_{uniq}\right).$$

By Theorem 1, the running time of algorithm 5 is

$$O\left(T_{max}\sqrt{\log T_{max}} + \frac{T_{av}}{\sqrt{p_{succ}}} \log^{1.5} T_{max}\right).$$

Since $T_{max} = O(2^m) = O(\frac{\kappa}{\epsilon})$, we have $T_{max} \leq \frac{T_{av}}{\sqrt{p_{succ}}}$ and the running time is

$$O\left(\frac{T_{av}}{\sqrt{p_{succ}}} \log^{1.5} T_{max}\right) = O\left(\frac{\kappa}{\epsilon} k_{uniq} \log^{1.5} \kappa\right) = O\left(\frac{m\kappa}{\epsilon_{final}} k_{uniq} \log^{1.5} \kappa\right),$$

with the 2nd equality following from $\epsilon = \Theta(\epsilon_{final}/m)$. Since algorithm 5 needs to be repeated $O(\sqrt{m} \log \frac{1}{\epsilon_{final}})$ times, the overall running time is

$$O\left(\frac{m^{1.5}\kappa}{\epsilon_{final}} k_{uniq} \log^{1.5} \kappa \log \frac{1}{\epsilon_{final}}\right) = O\left(\frac{\kappa \log^3 \kappa}{\epsilon_{final}^3} \log^2 \frac{1}{\epsilon_{final}}\right),$$

with the equality following from $m = O(\log \kappa)$.

4.3 Proofs of Lemmas about the quality of output state

Proof: [of Lemma 5] Let $|v_i\rangle$ be an eigenstate of $\mathcal{A}$. Then, the eigenvalue estimation leaves $|v_i\rangle$ unchanged (and produces an estimate for the eigenvalue λ_i in the E register). This means that the algorithm above maps $|x\rangle = \sum_i \alpha_i |v_i\rangle$ to

$$\sum_i \alpha_i |v_i\rangle_I \otimes |\phi_i\rangle_{O,S,E}$$

where

$$|\phi_i\rangle_{O,S,E} = |1\rangle_O \otimes |\phi_i'\rangle_{S,E} + |0\rangle_O \otimes |\phi_i''\rangle_{S,E}.$$

We will show:

- If $|v_i\rangle$ is good, then $|\phi'_i\rangle_{S,E}$ is close to $\frac{1}{\kappa\tilde{\lambda}_i}|2j_i\rangle_S \otimes |0\rangle_E$.
- If $|v_i\rangle$ is bad, then $\|\phi'_i\|$ does not become too large (and, therefore, does not make too big contribution to $\|P_1|\psi\rangle - |\psi''\rangle\|$).

These two statements are quantified by two claims below: Claim 2 and Claim 5. The Lemma follows by combining these two claims and the fact that the sum of $|\alpha_i|^2$ over all bad i is equal to δ.

Claim 2 *If $|v_i\rangle$ is good,*

$$\left\| |\phi'_i\rangle - \frac{1}{\kappa\tilde{\lambda}_i}|1\rangle_O \otimes |2j_i\rangle_S \otimes |0\rangle_E \right\|^2 \leq (2m+37)\epsilon C$$

where $C = (\frac{1}{\kappa\tilde{\lambda}_i})^2$.

Proof: We express $|\phi'_i\rangle = \sum_j |2j\rangle_S \otimes |\phi_{i,j}\rangle_E$. Furthermore, we group the terms of $|\phi'_i\rangle$ in a following way:

$$|\phi'_i\rangle = |\phi_<\rangle + |\phi_=\rangle + |\phi_>\rangle$$

where

$$|\phi_<\rangle = \sum_{j<j_i} |2j\rangle_S \otimes |\phi_{i,j}\rangle_E,$$

$$|\phi_=\rangle = \otimes|2j_i\rangle_S \otimes |\phi_{i,j_i}\rangle_E,$$

$$|\phi_>\rangle = \sum_{j>j_i} |2j\rangle_S \otimes |\phi_{i,j}\rangle_E.$$

We have

$$\left\| |\phi'_i\rangle - \frac{1}{\kappa\tilde{\lambda}_i}|2j_i\rangle_S \otimes |0\rangle_E \right\|^2 =$$

$$\|\phi_<\|^2 + \left\| |\phi_=\rangle - \frac{1}{\kappa\tilde{\lambda}_i}|2j_i\rangle_S \otimes |0\rangle_E \right\|^2 + \|\phi_>\|^2.$$

We first show that $\|\phi_<\|$ and $\|\phi_>\|$ are not too large.

For $j < j_i$, the eigenvalue estimation outputs an answer that is more than $\tilde{\lambda}_{i,j}$ with probability at most ϵ. Therefore, the probability of step (3b) being executed is at most ϵ. Moreover, if this step is executed, the estimate λ' for the eigenvalue is at least $\frac{1}{\epsilon 2^j}$. Therefore, the coefficient of $|1\rangle_O$ in (9) is

$$\frac{1}{\kappa\lambda'} \leq \frac{2^{j+1}\epsilon}{\kappa}.$$

By summing over all $j < j_i$, we get

$$\|\phi_<\|^2 = \sum_{j<j_i} \|\phi'_{i,j}\|^2 = \sum_{j<j_i} \left(\frac{2^{j+1}\epsilon}{\kappa}\right)^2 \epsilon \leq \frac{1}{3}\left(\frac{2^{j_i+1}\epsilon}{\kappa}\right)^2 \epsilon,$$

with the inequality following from the formula for the sum of a geometric progression. By using the right hand side of Claim 1, we get

$$\|\phi_<\|^2 \leq \frac{4\epsilon}{3}\left(1+\frac{3\epsilon}{2}\right)^2 C$$

where $C = (\frac{1}{\kappa\tilde{\lambda}_i})^2$. If $\epsilon < 0.1$, we can upper-bound this by $1.6\epsilon C$.

For $j > j_i$, we have $\|\phi_{i,j}\|^2 \leq \epsilon^{j-j_i}$. (We only reach stage j if, in every previous stage k, eigenvalue estimation outputs an estimate that is smaller than $\tilde{\lambda}_i$. For each $k \in \{j_i, j_i+1, \ldots, j-1\}$, this happens with probability at most ϵ.)

Therefore,

$$\|\phi_>\|^2 = \sum_{j>j_i} \|\phi'_{i,j}\|^2 \leq \sum_{j>j_i} \left(\frac{2^{j+1}\epsilon}{\kappa}\right)^2 \epsilon^{j-j_i} \leq$$

$$4\left(1+\frac{3\epsilon}{2}\right)^2 C \sum_{j=1}^{\infty} (4\epsilon)^j = 16\left(1+\frac{3\epsilon}{2}\right)^2 \frac{\epsilon}{1-4\epsilon} C$$

where the 2nd inequality follows from the right hand side of Claim 1 and the last equality follows from the formula for the sum of a geometric progression. If $\epsilon < 0.1$, we can upper bound this by $36\epsilon C$. Thus, both $\|\phi_<\|^2$ and $\|\phi_>\|^2$ are small enough.

For $|\phi_=\rangle$, we first estimate the probability that algorithm reaches stage j_i.

Claim 3 *Algorithm 4 reaches stage j_i with probability at least $1 - 2(m-1)\epsilon$.*

Proof: For each $j < j_i$, the eigenvalue estimation may produce an incorrect answer with probability at most ϵ. This may lead to transformation (9) being executed with probability at most ϵ. Moreover, this causes some disturbance for the next step, when eigenvalue estimation is uncomputed. Let $|\psi\rangle$ be the output of the eigenvalue estimation. We can split $|\psi\rangle = |\psi'\rangle + |\psi''\rangle$ where $|\psi'\rangle$ consists of estimates λ which are smaller than the one in the condition of step 3b and $|\psi''\rangle$ consists of estimates that are greater than or equal to the one in the condition. Then, $\|\psi''\|^2 \leq \epsilon$ and, conditional on outcome register being $|2\rangle$, the estimation register is in the state $|\psi'\rangle$. If the estimation register was in the state $|\psi\rangle$, uncomputing the eigenvalue estimation would lead to the correct initial state $|0\rangle$. If it is in the state $|\psi'\rangle$, then, after uncomputing the eigenvalue estimation, E can be in a basis state different from $|0\rangle$ with probability at most $\|\psi - \psi'\|^2 = \|\psi''\|^2 \leq \epsilon$.

Thus, the probability of the computation terminating for a fixed $j < j_i$ is at most 2ϵ. The probability of that happening for some $j < j_i$ is at most $2(j_i - 1)\epsilon < 2(m-1)\epsilon$. ∎

We now assume that the algorithm is started from stage j_i.

Claim 4 *If Algorithm 4 is started from stage j_i (instead of stage 1), then*

$$\left\| |\phi_{i,j_i}\rangle_E - \frac{1}{\kappa\tilde{\lambda}}|0\rangle_E \right\|^2 \leq \epsilon\left(1+\frac{3\epsilon}{2}\right) C.$$

Proof: Let

$$|\psi\rangle = \sum_{\lambda} \alpha_\lambda |\lambda\rangle$$

be the output of the eigenvalue estimation in stage j_i. Then, $|\alpha_{\tilde{\lambda}_i}|^2 \geq 1-\epsilon$ and $\||\psi\rangle - \alpha_{\tilde{\lambda}_i}|\tilde{\lambda}_i\rangle\|^2 \leq \epsilon$. Conditional on O being mapped to $|1\rangle$, the estimation register E is in the state

$$|\psi'\rangle = \sum_{\lambda} \beta_\lambda |\lambda\rangle$$

where $\beta_\lambda = \frac{\alpha_\lambda}{\kappa\lambda}$ when $\lambda \geq \frac{1}{\epsilon 2^{j+1}}$ and $\beta_\lambda = 0$ otherwise. By Claim 1, we have

$$\frac{1}{\lambda} \in [0, \epsilon 2^{j+1}] \subseteq \left[0, \left(2 + \frac{3\epsilon}{2}\right)\frac{1}{\tilde{\lambda}}\right].$$

When $\lambda \geq \frac{1}{\epsilon 2^{j+1}}$, this implies

$$\left|\beta_\lambda - \frac{\alpha_\lambda}{\kappa\tilde{\lambda}}\right| = \left|\frac{\alpha_\lambda}{\kappa\lambda} - \frac{\alpha_\lambda}{\kappa\tilde{\lambda}}\right| \leq \left(1 + \frac{3\epsilon}{2}\right)\frac{\alpha_\lambda}{\kappa\tilde{\lambda}}.$$

When $\lambda < \frac{1}{\epsilon 2^{j+1}}$, we have $\beta_\lambda = 0$ and

$$\left|\beta_\lambda - \frac{\alpha_\lambda}{\kappa\tilde{\lambda}}\right| = \frac{\alpha_\lambda}{\kappa\tilde{\lambda}}.$$

By summing over all $\lambda \neq \tilde{\lambda}$, we get

$$\left\|\psi' - \frac{1}{\kappa\tilde{\lambda}}\psi\right\|^2 \leq \left(1 + \frac{3\epsilon}{2}\right) C \sum_{\lambda:\lambda\neq\tilde{\lambda}} |\alpha_\lambda|^2 \leq \left(1 + \frac{3\epsilon}{2}\right)\epsilon C.$$

Therefore, (conditional on the outcome register being $|1\rangle$) uncomputing **UniqueEst** leads to a state $|\varphi\rangle_E$ with

$$\left\|\varphi - \frac{1}{\kappa\tilde{\lambda}}|0\rangle\right\|^2 \leq \epsilon\left(1 + \frac{3\epsilon}{2}\right) C.$$

∎

Since the algorithm might not reach stage j_i with probability at most $2(m-1)\epsilon$, we have to combine the error bounds from Claims 3 and 4. This gives us

$$\left\||\phi_{i,j_i}\rangle_E - \frac{1}{\kappa\tilde{\lambda}}|0\rangle_E\right\| \leq \epsilon\left(2m - 1 + \frac{3\epsilon}{2}\right) C.$$

Combining this with bounds of $1.6\epsilon C$ and $36\epsilon C$ on $\|\psi_<\|$ and $\|\psi_>\|$ completes the proof of Claim 2. ∎

Claim 5 *If $|v_i\rangle$ is bad,*

$$\|\phi_i'\|^2 \leq 30C$$

where $C = (\frac{1}{\kappa}\tilde{\lambda}_i)^2$.

Proof: We express

$$|\phi_i'\rangle = |\phi_{\leq}\rangle + |\phi_{>}\rangle$$

where

$$|\phi_{\leq}\rangle = \sum_{j \leq j_i+1} |2j\rangle_S \otimes |\phi_{i,j}\rangle_E,$$

$$|\phi_{>}\rangle = \sum_{j > j_i+1} |2j\rangle_S \otimes |\phi_{i,j}\rangle_E.$$

We have

$$\|\phi_{\leq}\|^2 \leq \left(\frac{1}{\kappa\epsilon 2^{j_i+2}}\right)^2 \leq 16\left(1+\frac{3\epsilon}{2}\right)^2 C. \tag{13}$$

Here, the first inequality follows from the amplitude of $|1\rangle$ in (9) being $\frac{1}{\kappa\lambda}$, $\lambda \geq \frac{1}{\epsilon 2^{j+1}}$ and $j \leq j_i + 1$. The second inequality follows from Claim 1.

Starting from stage $j+1$, the probability of algorithm obtaining $\lambda < \frac{1}{2^{j+1}\epsilon}$ is at most ϵ at each stage. Therefore (similarly to the proof of Claim 2),

$$\|\phi_{>}\|^2 = \sum_{j>j_i+1} \|\phi_{i,j}'\|^2 \leq \sum_{j>j_i+1} \left(\frac{2^{j+1}\epsilon}{\kappa}\right)^2 \epsilon^{j-j_i-1} \leq \left(\frac{2^{j_i+2}\epsilon}{\kappa}\right)^2 \sum_{j=1}^{\infty} (4\epsilon)^j \leq$$

$$16\left(1+\frac{3\epsilon}{2}\right)^2 C \sum_{j=1}^{\infty} (4\epsilon)^j = 16\left(1+\frac{3\epsilon}{2}\right)^2 \frac{4\epsilon}{1-4\epsilon} C. \tag{14}$$

The claim follows by putting equations (13) and (14) together and using $\epsilon < 0.01$. ∎

Proof: [of Lemma 6] We have

$$|\lambda_i - \tilde{\lambda}_i| \leq \frac{1+\epsilon}{2^{j+1}} \leq (1+\epsilon)\epsilon\tilde{\lambda}_i,$$

with the first inequality following from the correctness of the unique-output eigenvalue estimation and the second inequality following from the definition of $\tilde{\lambda}_i$. Let $\delta = (1+\epsilon)\epsilon$.

If $|\lambda_i - \tilde{\lambda}_i| \leq \delta\tilde{\lambda}_i$, then

$$\left|\frac{1}{\lambda_i} - \frac{1}{\tilde{\lambda}_i}\right| \leq \frac{\delta}{1-\delta}\frac{1}{\tilde{\lambda}_i}.$$

Therefore, we have $\||\psi'\rangle - |\psi_{ideal}\rangle\| \leq \frac{\delta}{1-\delta}\||\psi_{ideal}\rangle\|$ and

$$\frac{\delta}{1-\delta} = \frac{(1+\epsilon)\epsilon}{1-(1+\epsilon)\epsilon} < \frac{2\epsilon}{1-2\epsilon}.$$

∎

4.4 Proofs of Lemmas about the running time of Algorithm 4

Proof: [of Lemma 7] We first consider the case when the input state $|x\rangle$ is an eigenstate $|v_i\rangle$ of H. Let $p_{stop,j}$ be the probability that Algorithm 4 stops after stage j. Then, the square of l_2 average running time of Algorithm 4 is of the order

$$O\left(\sum_j p_{stop,j} 2^{2j} k_{uniq}^2\right) \tag{15}$$

since, in first j stages we use amplitude amplification for time

$$k_{uniq}(2 + 2^2 + \ldots + 2^j) = k_{uniq}(2^{j+1} - 2) = O(k_{uniq} 2^j).$$

Let $j \geq j_i + 1$. The probability that, in the j^{th} run of eigenvalue estimation, the algorithm does not stop is at most ϵ. Therefore, $p_{j_i+k} \leq \epsilon^{k-1}$ and the expression in (15) is at most k_{uniq}^2 times

$$2^{2(j_i+1)} + \sum_{j=j_i+2}^{\infty} \epsilon^{j-j_i-1} 2^{2j} < 2^{2(j_i+1)} + 2^{2(j_i+1)} \sum_{j=1}^{\infty} (4\epsilon)^j = O(2^{2j_i}).$$

If $|x\rangle = \sum_i \alpha_i |v_i\rangle$, the square of l_2-average of the number of steps is of the order

$$O\left(\sum_i |\alpha_i|^2 2^{2j_i} k_{uniq}^2\right)$$

because, each subspace of the form $|v_i\rangle \otimes \mathcal{H}_A \otimes \mathcal{H}_S \otimes \mathcal{H}_E$ stays invariant throughout the algorithm and, thus, can be treated separately. Taking square root finishes the proof. ∎

Proof: [of Lemma 8] Again, we can treat each subspace of the form $|v_i\rangle \otimes \mathcal{H}_A \otimes \mathcal{H}_S \otimes \mathcal{H}_E$ separately. As shown in the proof of Claim 2, the probability of the algorithm stopping before stage j_i is at most $2(j_i - 1)\epsilon \leq 2(m-1)\epsilon$. Therefore, the algorithm stops at stage j_i or $j_i + 1$ with a probability that is at least a constant. The probability of algorithm stopping succesfully (i.e., producing $|1\rangle$ in an outcome register) is $\frac{1}{\kappa^2 \lambda^2}$. By Claim 1, we have $\lambda = O(\frac{1}{\epsilon 2^{j_i}})$. This implies that the probability of the algorithm stopping successfully is $O(\frac{\epsilon^2 2^{2j_i}}{\kappa^2})$. ∎

References

[1] S. Aaronson, A. Ambainis, Quantum search of spatial regions. *Theory of Computing*, 1:47-79, 2005. Also quant-ph/0303041.

[2] A. Ambainis. Quantum search with variable times. *Theory of Computing Systems*, 47(3): 786-807, 2010. Earlier versions in STACS'08 and quant-ph/0609188.

[3] D. Berry. Quantum algorithms for solving linear differential equations. arXiv:1010.2745.

[4] G. Brassard, P. Høyer, M. Mosca, A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information Science*, AMS Contemporary Mathematics Series, 305:53-74, 2002. Also quant-ph/0005055.

[5] A. Harrow, A. Hassidim, S. Lloyd, Quantum algorithm for inear systems of equations. *Physical Review Letters*, 15(103):150502, 2009. Also arXiv:0811.3171.

[6] P. Kaye, R. Laflamme, M. Mosca. *An Introduction to Quantum Computing.* Cambridge University Press, 2007.

[7] S. K. Leyton, T. J. Osborne. A quantum algorithm to solve nonlinear differential equations. arXiv:0812.4423.

[8] M. Nielsen, I. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[9] J. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, School of Computer Science, Carnegie Mellon University, 1994.