Inverting Well Conditioned Matrices in Quantum Logspace

Amnon Ta-Shma[*]
The Blavatnik School of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel
amnon@tau.ac.il

ABSTRACT

We show that quantum computers improve on the best known classical algorithms for matrix inversion (and singular value decomposition) as far as space is concerned. This adds to the (still short) list of important problems where quantum computers are of help.

Specifically, we show that the inverse of a well conditioned matrix can be *approximated* in quantum logspace with intermediate measurements. This should be compared with the best known classical algorithm for the problem that requires $\Omega(\log^2 n)$ space. We also show how to approximate the spectrum of a normal matrix, or the singular values of an arbitrary matrix, with ϵ additive accuracy, and how to approximate the singular value decomposition (SVD) of a matrix whose singular values are well separated.

The technique builds on ideas from several previous works, including simulating Hamiltonians in small quantum space (building on [2] and [10]), treating a Hermitian matrix as a Hamiltonian and running the quantum phase estimation procedure on it (building on [5]) and making small space probabilistic (and quantum) computation *consistent* through the use of *offline randomness* and the *shift and truncate* method (building on [8]).

Categories and Subject Descriptors

F.1.1 [**Theory of Computation**]: Models of Computation—*Quantum computation theory*; F.2.1 [**Analysis of Algorithms and Problems**]: Computations on matrices

General Terms

Algorithms, Theory

Keywords

Quantum computation, Quantum space complexity, Quantum phase estimation, Quantum state tomography, Matrix inversion, Approximating matrix spectrum

1. INTRODUCTION

In 1996 Peter Shor presented a quantum algorithm for factoring that is exponentially faster than the best currently known classical algorithm for the problem [9]. Since then other quantum algorithms offering time speedup over the best known classical algorithm have been found (see [1]). In this work we present a quantum algorithm for matrix inversion whose *space* complexity is quadratically better than the best currently known classical algorithm for the problem.

Matrix inversion is a fundamental operation in Linear Algebra, equivalent to solving a general system of linear equations. As such, matrix inversion and variants (as sparse matrix inversion) have been heavily studied in various computational models, including sequential and parallel complexity, space bounded complexity and Boolean and arithmetic circuit complexity.

As far as space bounded complexity goes, matrix inversion is complete for the class DET of all functions L–reducible to the determinant of an integer matrix [4, Proposition 5.2], and therefore belongs to the class $\mathrm{DSPACE}(\mathrm{O}(\log^2 \mathrm{n}))$. In particular, a logspace algorithm for matrix inversion would collapse DET to L and since

$$\mathrm{L} \subseteq \mathrm{RL} \subseteq \mathrm{NL} \subseteq \mathrm{DET},$$

this would imply L = RL = NL = DET, which would constitute a major earthquake in our understanding of space bounded computation. Indeed, as of today, no known algorithm solves matrix inversion with $o(\log^2 n)$ space.

Matrix inversion is not *numerically stable.* The problem appears for invertible matrices A that are close to being singular, or, equivalently, matrices A whose smallest singular value $s_n(A)$ is close to zero. The *condition number* of the matrix A, $\kappa(A) = \frac{s_1(A)}{s_n(A)}$, where $s_1(A)$ and $s_n(A)$ are respectively the largest and smallest singular values of A, measures the numerical stability of inverting A.

Using a beautiful technique, Harrow, Hassidim and Lloyd [5] showed that given an $n \times n$ matrix A that is sparse and well conditioned (i.e., its condition number is not too small), and given $b \in \mathbb{C}^n$, one can approximate certain predicates on $x = A^{-1}b$ with a *quantum* machine, exponentially faster than with a classical machine. The set of predicates that can be computed that way is restricted, and consists of pred-

[*]Supported by the FP7 FET-Open project QCS.

STOC'13, June 1Ű4, 2013, Palo Alto, California, USA.

icates that correspond to efficient measurements on an n dimensional Hilbert space.

In this work we develop a space bounded variant of the idea. There are several delicate issues in defining space bounded computation in general, and *quantum* space bounded computation in particular, and we discuss these issues and previous work in Section 2. Roughly speaking, following [10], we consider deterministic logspace computations that also use $O(\log n)$ qubits. The control of the machine (i.e., its δ function) is classical, and may invoke unitary transformations from the set $\{\mathrm{HAD}, \mathrm{CNOT}, \mathrm{T}, \mathrm{M}\}$ on the qubits, where M is a measurement in the standard computational basis, i.e., our model allows *intermediate measurements.* We prove that in this model *approximating* the inverse can be done in *quantum* logspace. Namely:

THEOREM 1.1. *Fix $\epsilon(n)$, $\zeta(n) > 0$. There exists a $\mathrm{BQ \cdot SPACE(O(\log n + \log \epsilon^{-1} + \log \zeta^{-1}))}$ algorithm that given an $n \times n$ matrix A with $1 \geq s_1(A) \geq s_n(A) \geq \zeta$, outputs with probability $1-\epsilon$ a matrix approximating A^{-1} with ϵ accuracy in the ℓ_1 norm.*

We remark that even though we only *approximate* the inverse A^{-1}, we still achieve something that is not known today with classical computation, and as far as we know all current classical algorithms for the task require $\Omega(\log^2 n)$ space complexity.

We also show that if in addition the singular values of A are well separated from each other, then using about the same quantum space complexity, we can decompose A into its SVD, i.e., we can output matrices $\widetilde{U}, \widetilde{V}$ and $\widetilde{D}$ that are close to the correct SVD $A = UDV$, with U, V being unitary and D positive and diagonal. Formally,

THEOREM 1.2. *Fix $\epsilon(n)$, $\zeta(n) > 0$. There exists a $\mathrm{BQ \cdot SPACE(O(\log n + \log \epsilon^{-1} + \log \zeta^{-1}))}$ algorithm that given an $n \times n$ matrix A with singular values s_i such that $|s_i - s_j| \geq \zeta$ for all $i \neq j$, outputs with probability $1-\epsilon$ matrices $\widetilde{U}, \widetilde{V}$ and $\widetilde{D}$ such that $\widetilde{U}$ is ϵ close to a unitary matrix U, $\widetilde{V}$ is ϵ close to a unitary matrix V, $\widetilde{D}$ is ϵ close to a positive, diagonal matrix D and $A = UDV$.*

In particular it follows that given A one can approximate the singular values of A with ϵ additive accuracy using only $O(\log n + \log \epsilon^{-1} + \log \zeta^{-1})$ quantum space. This stands in sharp contrast to the best deterministic or probabilistic algorithms we know today that require $\Omega(\log^2 n)$ space.

We remark that the quadratic gap between the quantum space complexity and the currently best classical space complexity is best possible. From [10] it follows that the classical result $\mathrm{BPL} \subseteq \mathrm{DSPACE(O(\log^2 n))}$ generalizes to $\mathrm{BQL} \subseteq \mathrm{DSPACE(O(\log^2 n))}$, and a similar result was shown before by Watrous [11] for another variant of quantum space–bounded computation.

Finally, we sketch some of the main ingredients of the technique. First, it is a simple observation that w.l.o.g A is Hermitian (see the proofs of Theorem 1.1 in Section 6.3 and Theorem 1.2 in Section 6.2). Assume H is Hermitian with eigenvectors $v_1, \ldots, v_N$ and corresponding eigenvalues $\lambda_1, \ldots, \lambda_N$. Further assume that the eigenvalues are well separated, i.e., there exists some constant c s.t. for all $i \neq j$, $|\lambda_i - \lambda_j| \geq n^{-c}$.

The first observation, building on [5], is that if we can simulate $U = e^{iH}$ space-efficiently, then we can use phase estimation to sample a random eigenvector of H (represented as a pure state in the n-dimensional Hilbert space defined by the $\log(n)$ qubits) together with its eigenvalue. Indeed, in Section 4 we show, building on work from [2] and [10], that $U = e^{iH}$ can be approximated in small quantum space.

Our next step is showing how to approximate *all* the eigenvalues of H. For this purpose we show that the phase estimation process can be made *consistent* in the sense that the output of the quantum machine is almost always the same, depending on some few random bits that are kept fixed throughout the computation (which, following [8], we call *offline* randomness) and the input. The main point is that the output (almost always) does *not* depend on the internal random coins, or the measurements outcome. We heavily use here the *shift and truncate* technique developed by Saks and Zhou [8] in the classical setting, and we "lift" it to superpositions in the quantum setting. We explain this part in Sections 2.1 and 5.2.

Having that we can prepare a fixed mixed state that is uniformly distributed over $\{v_i, \lambda_i\}$, and moreover, each time we get result λ_i we are in the same pure-state v_i. Therefore, we can apply *quantum state tomography* to learn in parallel, and hence in quantum low-space, the eigenvector v_i. In Section 6.1 we show that quantum state tomography can be made to run in small quantum space. Knowing both $\{v_i\}$ and $\{\lambda_i\}$, we can output in BQL (which is the class quantum logspace with bi-sided bounded error) a unitary matrix U and a positive, diagonal matrix D such that $H = UDU^\dagger$. Consequently, we can also output H in BQL. Moreover, when H is well-conditioned we can also output $H^{-1} = UD^{-1}U^\dagger$ in BQL. Doing the general case (where eigenvalues are not necessarily separated) is more involved, and yields the entries of H^{-1} but not the SVD $H = U^\dagger DU$.

The paper is organized as follows. In Section 2 we discuss and define quantum space bounded computation. In Section 2.1 we explain the shift and truncate method of [8] and how to use it to make probabilistic and quantum computations *consistent.* We also explain why consistency allows composition of space bounded machines. In Section 3 we give background on matrix exponentiation and Trotter's formula. In Section 4 (and Appendix A) we prove e^{iH} can be simulated in small quantum space. In Section 5 we explain how to approximate in small space the spectrum of a given matrix. In section 6.1 we prove quantum state tomography is possible in small quantum space. Finally, in Section 6.2 we show how to decompose and invert a well-conditioned matrix whose singular values are well-separated, and in Section 6.3 we do the general case.

2. SPACE BOUNDED COMPUTATION

A *deterministic* space-bounded Turing machine has three semi-infinite tapes: an *input tape* that contains the input and is read-only; a *work tape* that is read-write and an *output tape* that is write only and is unidirectional, i.e., it behaves like a printer and there is no access to previously printed bits. The time complexity is, as usual, the number of transition steps the TM performs on an input, and the space complexity is the number of cells on the work tape, i.e., we don't count the input tape (that never changes) and the output tape (that is accessible only for printing the output) in the space complexity. With that we define $\mathrm{DTimeSpace(t(n), s(n))}$ and $\mathrm{DSPACE(s(n))}$. $\mathrm{L} = \bigcup_c \mathrm{DSPACE(c \log(n))}$.

A *probabilistic* space-bounded Turing machine is similar to the deterministic machine except that it can also toss random coins. One convenient way to formulate this is by adding a fourth semi-infinite tape, the *random-coins tape*, that is read-only, unidirectional (i.e., we do not provide free access to previously tossed coins) and is initialized with perfectly uniform bits. We are only concerned with *bounded error* computation: we say a language is accepted by a probabilistic TM if for every input in the language the acceptance probability is higher than 2/3, and for every input not in the language it is at most 1/3. As usual the acceptance probability can be amplified as long as there is some non-negligible gap between the acceptance probability of yes and no instances. This defines $\mathrm{BP} \cdot \mathrm{TimeSpace}(\mathrm{t(n)}, \mathrm{s(n)})$ and $\mathrm{BP} \cdot \mathrm{SPACE}(\mathrm{s(n)})$.

Another way of defining space-bounded computations is by L–uniform *bounded-depth circuits*. The circuit model is more restricted than the TM model because in the circuit model computation is *oblivious* to the computation history, whereas in the TM model computation at time t depends on the computation history. $\mathrm{NC^k}$ denotes the class of languages accepted by L–uniform polynomial-size circuits of $O(\log^k n)$ depth, where the allowed operations come from a finite universal set of bounded fan-in gates, typically $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}\}$. $\mathrm{AC^k}$ is the same, except that the allowed operations are $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}\}$ with *unbounded* fan-in. It holds that:

$$\mathrm{AC^0} \subsetneq \mathrm{NC^1} \subseteq \mathrm{L} \subseteq \mathrm{BPL} \subseteq \mathrm{NC^2}.$$

In the *quantum* model we introduce another tape, the *quantum work tape*, replacing the random-coins tape. The quantum work tape is semi-infinite and contains quantum bits. A universal basis for quantum computation requires 2-qubit operations. There are several ways to allow this, and following [10] we allow two heads over the quantum work tape. We restrict the set of allowed quantum operations to the universal set $\{\mathrm{HAD}, \mathrm{CNOT}, \mathrm{T}\}$, and the set of allowed measurements to a projection in the standard basis, which we denote M. The transition function is *classical*, and there are four special states that invoke, respectively, HAD, CNOT, T and M, on the one or two qubits that are below the quantum work tape heads.

The main dilemma concerning the definition of quantum computation is whether to allow *intermediate measurements* or not. In quantum time-bounded computation all measurements may be deferred to the end by invoking the *safe storage principle*. This technique, however, may exponentially increase the space complexity of the computation. Currently, it is still an open problem whether allowing intermediate measurements increases the computational power of the model. Thus, one has to decide whether to allow intermediate measurements or not. Earlier work, like [11, 12, 13], does not allow intermediate measurements. Later work notes that currently envisioned quantum systems consist of classical control calling upon quantum operations, and such systems allow intermediate measurements. This is also the model we work with. For a thorough discussion of these issues and previous work see the discussion in [10], and in particular Section 2 therein.

To summarize, in the quantum model we have four semi-infinite tapes: an *input tape*, a *classical work tape*, a *quantum work tape* and an *output tape*. The input, output and classical work tapes are classical, and obey the same rules as in the classical model. The quantum work tape contains qubits and has multiple-heads. The control of the system (i.e., the transition function) is classical and may invoke operations from $\{\mathrm{HAD}, \mathrm{CNOT}, \mathrm{T}, \mathrm{M}\}$, thus allowing intermediate measurements. Only cells in the classical work and quantum work tapes are counted in the space complexity. We only deal with bounded error computations, and we define $\mathrm{BQ} \cdot \mathrm{TimeSpace}(\mathrm{t(n)}, \mathrm{s(n)})$, $\mathrm{BQ} \cdot \mathrm{SPACE}(\mathrm{s(n)})$ and $\mathrm{BQL} = \bigcup_{\mathrm{c}} \mathrm{BQ} \cdot \mathrm{SPACE}(\mathrm{c} \log(\mathrm{n}))$ in the natural way.

Finally, we also have the more restricted model of L–uniform quantum circuits. We say a language A (or a function f) is computed by $\mathrm{DSPACE}(\mathrm{s(n)})$–uniform, quantum circuits of width $w(n)$ if there exist $\mathrm{DSPACE}(\mathrm{s(n)})$–uniform quantum circuits over $w(n)$ qubits, deciding A (or computing f). The set of allowed transformations (or "gates") is $\{\mathrm{HAD}, \mathrm{CNOT}, \mathrm{T}\}$, and projections in the standard basis are allowed only at the end. Clearly, any such language is contained in $\mathrm{BQ} \cdot \mathrm{SPACE}(\mathrm{O(s(n) + w(n))})$. The definitions can obviously be generalized to allow errors.

2.1 Probabilistic machines with a deterministic output

Often we are interested in computing a value (like acceptance probability, an entry in a matrix or the whole matrix) and are only able to approximate it with a probabilistic (or a quantum) machine. For example, assume there exists some value $u = u(x) \in \mathbb{R}$ that is determined by the input $x \in \{0,1\}^n$, and there exists a probabilistic TM $M(x, y)$ such that:

$$\forall_{x \in \{0,1\}^n} \qquad \Pr_y \; [\; |M(x,y) - u(x)| \geq \delta \;] \; \leq \epsilon. \qquad (1)$$

Loosely speaking, we can say M approximates u with δ *accuracy* and ϵ *error*. However, different coin tosses of M may lead to different results. In particular, it is not true that there exists a function u' such that M computes u' and $|u'(x) - u(x)| \leq \epsilon$, because the definition of M computing u' requires that for every input x, M outputs $u'(x)$ with probability $1-\epsilon$, and does not allow u' to depend on y. The above problem is not only syntactic. It also implies that when we compose reductions we cannot assume that the bottom level always returns the same value. This may, or may not, constitute a problem, depending on the properties required by the top level, but it always complicates arguments.

This situation appears in [8]. The argument there starts with the observation that a probabilistic machine can approximate some acceptance probability (and therefore some transition matrix) and it is critical to make sure this approximation almost always depends on the input alone and not on the machine's random coins. To guarantee this, [8] use a *shift and truncate* method. The idea is as follows. Typically, when $M(x,y)$ and $u(x)$ are δ' close, truncating $M(x,y)$ and $u(x)$ to $\delta \gg \delta'$ accuracy gives the same value. To make it concrete, think of $\delta' = 10^{-10}$, $\delta = 10^{-5}$ and $u(x) = 0.123456789123456789$. In fact, this is true for all values $u(x)$ except for those that lie δ' close to a border between two truncation results (e.g., $u = 0.123450000000000000001$ in the example above). The idea is to shift the approximation sections by a random multiple of δ' not exceeding δ. Thus, with a high probability over the shift we are not close to a δ'-boundary, and therefore the probabilistic algorithm almost always outputs the correct truncated result. I.e., once we fix a good shift (and almost all shifts are good) our approximation is good and depends only on the input (and the fixed shift) and not on the machine's random coins. We now make this formal:

LEMMA 2.1. *(based on [8]) Suppose $M(x,y)$ approximates $u(x)$ with δ accuracy and ϵ error using $O(\log(\frac{n}{\epsilon\delta}))$ space, i.e., Equation (1) holds. Let $\zeta > 0$ be arbitrary. Then, there exists another probabilistic algorithm $M'(x,y;s)$ with $|y| = O(\log(\frac{n}{\epsilon\delta\zeta}))$ and $|s| = O(\log(\frac{1}{\zeta}))$, and there exists a function $u'(x,s)$ such that:*

- *For all s: $|u'(x,s) - u(x)| \leq \delta$.*
- $\Pr_s\ [\ \Pr_y\ (\ M'(x,y;s) = u'(x,s)\ )\ < 1 - \epsilon\ ]\ \leq \zeta$.

Proof: We first describe the algorithm. Set $\delta' = \frac{\delta\zeta}{2}$ and let $L = \lfloor \frac{2}{\zeta} \rfloor = \lfloor \frac{\delta}{\delta'} \rfloor$. Pick the shift s randomly from $\{1, \ldots, L\}$ and fix it. Divide $\mathbb{R}$ to the consecutive sections $s\delta' \pm [k\delta, (k+1)\delta)$ for $k \in \mathbb{Z}$ (e.g., $[s\delta' - \delta, s\delta')$ and $[s\delta', s\delta' + \delta)$ are sections). Then run M with δ' accuracy and return the section to which $M(x,y)$ belongs.

For correctness, say a shift s is *good for u* if u is δ' away from a boundary, i.e., there exists a section $[c,d)$ such that $u \in [c+\delta', d-\delta')$. As $L\delta' < \delta$, for every u there exist at most 2 shifts that bring it δ' close to a boundary, and therefore the probability a shift is not good is at most $\frac{2}{L} \leq \zeta$. For a good shift, the probability over the random coins of M that the output is the section to which u belongs is at least $1 - \epsilon$ (because u is δ' away from a boundary), the section number depends only on u and the fixed shift s, and any point within the section (in particular, say, its middle point) is δ close to u. ■

Thus, throughout the computation we have two kinds of random bits: those that we toss and keep (like the random shift) which Saks and Zhou term *offline random bits* and those that we toss and forget (like all other random bits used by M). When we say we fix the shift, we mean that it is offline randomness: we toss the coins and keep it in memory throughout the computation. Subsequently, all offline random bits have to be counted in the algorithm's space complexity. Clearly, the same idea applies without change to quantum machines that have probabilistic, *classical* output. In Section 5.2 we employ a similar idea for quantum machines with a *quantum* output.

We remark that we can compose probabilistic (or quantum) bounded-space reductions together, as long as they almost surely output some fixed classical result that does not depend on the machine's randomness. I.e., assume we have two reductions A_1 and A_2 that given any input x_i (for $i = 1, 2$) use offline randomness s_i, true randomness r_i and with probability $1 - \epsilon_i$ output a fixed value $A_i(x_i, s_i)$ that does not depend on r_i. Then $A_2(A_1(x_1, r_1; s_1), r_2; s_2)$ is with probability $1 - \epsilon_1 - \epsilon_2$ the fixed value $A_2(A_1(x, s_1), s_2)$. Moreover, if A_i runs in space complexity s_i then the composed algorithm $A_2 \circ A_1$ runs in space $O(s_1 + s_2)$. Thus, we can compose a constant number of logarithmic-space reductions, while still maintaining logarithmic space bound.

Finally, we mention that Ambainis [3] also used consistent phase estimation (which he calls unique-answer eigenvalue estimation). However, Ambainis' method is specific for the phase estimation problem, and only works for the classical output of the quantum phase estimation procedure.

3. TROTTER'S FORMULA

3.1 Norms

$\|v\|$ is the ℓ_2 norm, $\|v\| = (\sum_{j=1}^{n} |v_j|^2)^{1/2}$. The *spectral norm* of a linear transformation T is defined by $\|T\| = \max_{\psi \neq 0} \frac{\|T\psi\|}{\|\psi\|}$. For any two matrices, $\|AB\| \leq \|A\| \cdot \|B\|$. Another characterization of the spectral norm is as follows: $\|T\|$ is the largest singular value of the linear operator T. In particular, if T is normal then $\|T\|$ equals the largest absolute value of its eigenvalues. If U is unitary, $\|U\| = 1$. For a general $n \times n$ matrix $A = (a_{i,j})$ we have $\|A\|_\infty \leq \|A\| \leq n^2 \|A\|_\infty$ where $\|A\|_\infty = \max_{i,j} |a_{i,j}|$.

The *trace norm* of an operator T with singular values $s_1(T) \geq \ldots \geq s_n(T) \geq 0$ is

$$\|A\|_{\mathrm{tr}} = \sum_{i=1}^{n} s_i(A).$$

Clearly, $\|A\| \leq \|A\|_{\mathrm{tr}} \leq n\|A\|$. The trace norm satisfies $\|A \otimes B\|_{\mathrm{tr}} = \|A\|_{\mathrm{tr}} \|B\|_{\mathrm{tr}}$. The trace norm of a density matrix is 1, and the if two density matrices ρ_1, ρ_2 are close in the trace norm, i.e., $\|\rho_1 - \rho_2\|_{\mathrm{tr}} \leq \epsilon$, then no physical process can separate them with more than ϵ distinguishing probability (see, e.g., [7, Chapter 9]).

3.2 Matrix exponentiation

If M is Normal it has an orthonormal basis of eigenvectors $\{v_j\}$ with corresponding eigenvalues $\{\lambda_j\}$, and can be expressed as

$$M = \sum_j \lambda_j |v_j\rangle\langle v_j|$$

For a function $f : \mathbb{C} \to \mathbb{C}$, $f(M) = \sum_j f(\lambda_j) |v_j\rangle\langle v_j|$ and shares the same eigenvector basis with M. In particular, this defines e^M. If H is Hermitian then its eigenvalues are real and hence e^{iH} is unitary.

3.3 The formula

Consider the sum $A + B$ of two Hermitian matrices A and B. We are interested in writing the unitary matrix $e^{i(A+B)}$ in terms of e^{iA} and e^{iB}. If A and B commute, this is simple: we have $e^{i(A+B)} = e^{iA} \cdot e^{iB}$. If the two matrices do not commute, Trotter's formula gives a way to do this:

$$\lim_{L\to\infty} (e^{iA/L} e^{iB/L})^L \quad = \quad e^{i(A+B)}$$

In words, it says that if we interleave short executions of A and B, then in the limit we get an execution of $A + B$. For our purpose we need to quantify the error as a function of L, and for that we use Equation (4.104) from [7],

$$\left\| e^{2\delta i(A+B)} - e^{\delta iA} e^{2\delta iB} e^{\delta iA} \right\| \quad \leq \quad O(\delta(\|A\| + \|B\|))^3 \ (2)$$

We also need to deal with matrices $H = \sum_m H_m$ that are sums of more than two Hermitian matrices. The following lemma is taken from [2]:

LEMMA 3.1. *Let H_m be Hermitian, $m = 1, \ldots, M$ and $H = \sum_{m=1}^{M} H_m$. Further assume for every $1 \leq k \leq \ell \leq M$ we have $\left\|\sum_{m=k}^{\ell} H_m\right\| \leq \Lambda$. Let $\delta < \frac{1}{4}$ and define*

$$U_\delta = [e^{\delta iH_1} \cdot e^{\delta iH_2} \cdot \ldots \cdot e^{\delta iH_M}] \cdot [e^{\delta iH_M} \cdot e^{\delta iH_{M-1}} \cdot \ldots \cdot e^{\delta iH_1}] \quad (3)$$

Then $\|U_\delta^{\lfloor \frac{1}{2\delta} \rfloor} - e^{iH}\| \le O(\Lambda\delta + M\delta^2\Lambda^3)$.

Notice that for every fixed M and Λ, the error term goes down to zero with δ. In the application we pick δ in such a way that the above error term is polynomially small.

4. SIMULATING e^{iH} WITH SMALL WIDTH QUANTUM CIRCUITS

In this section we prove:

Theorem 4.1. *There exists a deterministic algorithm that given* $\epsilon > 0$ *and an* $n \times n$ *Hermitian matrix* H *with* $\|H\| \le 1$, *outputs a quantum circuit over* $\lceil \log n \rceil$ *qubits that* ϵ *approximates* e^{iH} *in the spectral norm.*[1] *Moreover, the algorithm runs in* $\mathrm{DSPACE}(\mathrm{O}(\log n + \log \epsilon^{-1}))$.

The unitary transformation $U = e^{iH}$ acts on a dimension n vector space, and we think of it as acting on $\lceil \log n \rceil$ qubits. As the set $\{\mathrm{HAD}, \mathrm{CNOT}, \mathrm{T}\}$ is universal, there exists a quantum circuit approximating U using only such gates. The challenge in proving Theorem 4.1 is producing such a circuit deterministically in *small space*. The standard *existence* proof (see, e.g., [7]) goes through the following steps:

1. First, it is shown that U can be *exactly* expressed as a product $U_k \dots U_1$ of unitaries, where each U_i is a *two-level unitary.* A *two-level* transformation is a transformation that acts non-trivially on a 2×2 subspace $W_{k,\ell} = Span\{|k\rangle, |\ell\rangle\}$ spanned by two vectors in the computational basis, and is identity on the orthogonal complement.

2. Then, it is shown that each such transformation can be computed *exactly* using CNOT and single-qubit operations.

3. Finally, the Solovay-Kitaev Theorem asserts that any single qubit transformation can be approximated using $\{\mathrm{HAD}, \mathrm{T}\}$ gates.

We need space-efficient variants of these three steps. The second step goes without change to the space-efficient scenario (see Appendix A). The third step was recently done by van Melkebeek and Watson ([10], see below). The first step requires more work. The standard proof, that is concerned only with existence, or time-efficient computation, goes through a process that is close in spirit to diagonalizing matrices. It therefore seems unlikely that such a procedure can work in deterministic logspace. Instead, we implement the first step using a different approach. The main idea is expressing the matrix H as the sum of many local Hermitian matrices, and then applying Trotter formula that was stated in Lemma 3.1, using similar machinery to that used in [2]. This is also the approach taken in [5] for the case where H is a sparse, high-dimensional matrix. We now turn to the formal proof:

Proof: (of Theorem 4.1) Let H be as promised. We claim:

Claim 4.1. *There is a way to decompose* H *into* $H = \sum_{m=1}^{n^2/2} H_m$ *where each* H_m *is a two-level Hermitian matrix and* $\|H_m\| \le \|H\|$.

Proof: (of claim) The decomposition is as follows: for $k < \ell$ let $H_{\{k,\ell\}}$ be the matrix that is the same as H at entries (k,ℓ) and (ℓ,k) and zero otherwise. Also, for odd k let $H_{\{k\}}$ be the same as H at entries (k,k) and $(k+1,k+1)$ and zero otherwise. Each matrix in the decomposition is either of the form $\begin{pmatrix} 0 & H_{k,\ell} \\ H^*_{k,\ell} & 0 \end{pmatrix}$ for some k,ℓ and has norm $|H_{k,\ell}|$, or of the form $\begin{pmatrix} H_{k,k} & o \\ 0 & H_{k+1,k+1} \end{pmatrix}$ for some k and has norm $\max\{|H_{k,k}|, |H_{k+1,k+1}|\}$. Thus, $\max_m \|H_m\| = \max_{k,\ell} |H_{k,\ell}| = \|H\|_\infty \le \|H\|$. ■

We approximate e^{iH} with $U_\delta^{\lfloor \frac{1}{2\delta} \rfloor}$, where U_δ is as in Equation (3). This involves repeating $\lfloor \frac{1}{2\delta} \rfloor$ times the operator U_δ which is a product of two-level unitary transformations.

We fix $\Lambda = n^2$ and notice that for every $1 \le k \le \ell \le M$, $\left\|\sum_{m=k}^{\ell} H_m\right\| \le (\ell - k + 1)\|H\| \le M \le n^2 = \Lambda$. We also fix $\delta = \Theta(\epsilon n^{-4})$. By Lemma 3.1, $\|U_\delta^{\lfloor \frac{1}{2\delta} \rfloor} - e^{iH}\| \le O(\Lambda \cdot \delta + M\Lambda^3 \cdot \delta^2) \le \epsilon$.

The space complexity of the algorithm is as follows: we need a counter up to $\lfloor \frac{1}{2\delta} \rfloor$ which consumes $O(\log 1/\delta) = O(\log n + \log 1/\epsilon)$ bits, and we need space for implementing U_δ that we analyze below. This completes the first step reducing to a product of two-level unitary transformations.

We now claim:

Claim 4.2. *For every two-level Hermitian matrix* H *and every* $\epsilon > 0$ *there exists a quantum circuit with* CNOT *and single-qubit gates* ϵ *approximating* e^{iH}. *Moreover, such a circuit can be computed by a deterministic algorithm running in time* $poly(n/\epsilon)$ *and space* $O(\log n + \log \epsilon^{-1})$.

The proof appears in Appendix A. This reduces the problem to a product of CNOT and single-qubit operations. Finally, we use:

Theorem 4.2. *[10] For every unitary* U *acting on* $\mathbb{C}^2$ *and every* $\epsilon > 0$, *there exists a sequence* $U_1, \dots, U_k \in \{\mathrm{HAD}, \mathrm{T}\}$ *with* $k \le poly(\log \epsilon^{-1})$ *and a global phase factor* $e^{i\theta}$ *such that* $\left\|U - e^{i\theta} U_k \cdot \dots \cdot U_1\right\| \le \epsilon$. *Moreover, such a sequence can be computed by a deterministic algorithm running in time* $poly(\log \epsilon^{-1})$ *and space* $O(\log \epsilon^{-1})$, *given as input* ϵ *and a matrix that is at distance at most* $f(\epsilon)$ *from* U, *where* f *is a certain fixed polynomial.*

The space complexity is therefore $O(\log n + \log \epsilon^{-1})$ for counting up to $\lfloor \frac{1}{2\delta} \rfloor$ and reducing e^{iH} to a product of U_δ. Then for each two level unitary in the product (which requires a counter up to $2M$) we reduce the two-level unitary to a sequence of CNOT and single qubit operations, which by Claim 4.2 requires $O(\log n + \log \epsilon^{-1})$ space. Finally, we reduce each single-qubit operation to a product of single-qubit operations from our basis $\{\mathrm{HAD}, \mathrm{T}\}$, which takes $O(\log \epsilon^{-1})$ space by Theorem 4.2. Altogether the algorithm runs in $O(\log n + \log 1/\epsilon)$ space, and produces a $poly(n/\epsilon)$ size quantum circuit over $\log n$ qubits that ϵ approximates e^{iH} as required. ■

[1] In fact, it follows that one can also get an approximation in the ℓ_1 norm, as the two norms are within $poly(n)$ multiplicative factor of each other.

5. APPROXIMATING THE SPECTRUM OF A HERMITIAN MATRIX

5.1 Phase estimation

We use the phase estimation circuit as appearing, e.g., in [7, 6]. The circuit acts upon the input register I (of dimension n) and the estimation register E (of dimension T). T is a function of n, the accuracy parameter δ and the error parameter ϵ, and it is enough to choose $T = \mathrm{poly}(\frac{n}{\epsilon\delta})$. The circuit has oracle calls to $I, U, U^2, U^3, \ldots, U^T$, and we assume that U^k is implemented by k sequential calls to U. The circuit uses $\mathrm{poly}(n, \frac{1}{\epsilon}, \frac{1}{\delta})$ time and $O(\log \frac{n}{\epsilon\delta})$ space. Also the circuit does not produce garbage.

The circuit has the property that if it makes oracle calls to $U = e^{iH}$ and v is an eigenvector of U with eigenvalue $e^{\lambda i}$ then register E contains a good estimate of λ. More formally, denote

$$\mathrm{FAR} = \left\{ j \;:\; 0 \le j \le T-1 \;,\; |(\frac{j}{T}\pi - \lambda)(\mathrm{mod} 2\pi)| \ge \delta \right\},$$

where the modulo operation returns a value in $(-\pi, \pi]$. Then, the circuit maps the state $|v\rangle_I \otimes |\bar{0}\rangle_E$ to a state

$$|v\rangle_I \otimes \sum_{j=0}^{T-1} \beta_j \, |j\rangle_E$$

with the property that

$$\sum_{j \in \mathrm{FAR}} |\beta_j|^2 \le \epsilon. \qquad (4)$$

5.2 Consistent phase estimation

Property (4) allows estimating eigenvalues of H, and is sufficient, e.g., for Theorem 5.1. However, as explained in Sec 2.1, often we need the output to depend on the input only, and not on the measurements the phase estimation process does. The situation is similar to Lemma 2.1 except that now we deal with superpositions and a quantum algorithm. Restating the problem, we would like to map each eigenvector v_i to a *unique* vector $v_i \otimes |s(i)\rangle$, where $s(i)$ is a fixed classical value depending on v_i only, such that from $s(i)$ we can get a good approximation of λ_i. We call this *consistent phase estimation.*

We first describe the algorithm. We follow the same approach as in Lemma 2.1. Given $n, \epsilon, \delta = \epsilon$ and $\zeta = \frac{\epsilon}{n}$ we set $\delta' = \frac{\delta\zeta}{2}$ and let $L = \lfloor \frac{2}{\zeta} \rfloor = \lfloor \frac{\delta}{\delta'} \rfloor$. We pick a shift s randomly from $\{1, \ldots, L\}$ and fix it. Divide $[-1 - s\delta', 1 + \delta - s\delta']$ to consecutive sections of length δ. Then run the phase estimation algorithm with ϵ error and δ' accuracy and write in a new register S the *section* to which $\frac{j}{T}$ belongs. Finally, reverse the phase-estimation algorithm.

For correctness, say a shift s is *good for an eigenvalue* λ if λ is δ' away from a boundary, i.e., there exists a section $[c, d)$ such that $\lambda \in [c + \delta', d - \delta')$. As $L\delta' < \delta$, for every λ there exist at most 2 shifts that bring it δ' close to a boundary, and therefore the probability a shift is not good for λ is at most $\frac{2}{L} \le \zeta$. A shift is *good* if it is good for all eigenvalues of H. We see that the probability s is not good is at most $n\zeta \le \epsilon$.

Now, fix a good shift s, and an eigenvector v with eigenvalue λ. After the phase estimation step, we are in the superposition:

$$\psi \;=\; |v\rangle_I \otimes [\sum_j \beta_j \, |j\rangle_E \otimes |s(j)\rangle_S],$$

where $s(j)$ is the section j belongs to.

As s is a good shift for v, λ is δ' away from a section boundary, and therefore for every $j \notin \mathrm{FAR}$, $s(j) = s(\lambda)$ where $s(\lambda)$ is the section λ belongs to. I.e.,

$$\psi \;=\; |v\rangle \otimes [\sum_{j \notin \mathrm{FAR}} \beta_j \, |j\rangle \otimes |s(\lambda)\rangle + \sum_{j \in \mathrm{FAR}} \beta_j \, |j\rangle \otimes |s(j)\rangle],$$

Thus, ψ has fidelity at least $1 - \epsilon$ with the *ideal state* $\left|\dot{\psi}\right\rangle$:

$$\left|\dot{\psi}\right\rangle \;=\; |v\rangle \otimes [\sum_j \beta_j \, |j\rangle] \otimes |s(\lambda)\rangle\,.$$

In particular, $\left\| |\psi\rangle\langle\psi| - \left|\dot{\psi}\right\rangle\left\langle\dot{\psi}\right| \right\|_{\mathrm{tr}} \le O(\sqrt{\epsilon})$. Reversing the phase-estimation process on $\left|\dot{\psi}\right\rangle$ leads to the state:

$$|v\rangle_I \otimes |\bar{0}\rangle_E \otimes |s(\lambda)\rangle_S\,.$$

We conclude that the procedure described above maps the state $|v, \bar{0}, \bar{0}\rangle_{I \otimes E \otimes S}$, when v is an eigenvector of U, $O(\sqrt{\epsilon})$ close (in the trace distance) to the state $|v, \bar{0}, s(\lambda)\rangle$. Notice that $s(\lambda)$ depends only on v and the fixed shift s, and that $s(\lambda)$ determines a value $\widetilde{\lambda}$ that is δ close to the real eigenvalue λ, as desired. The procedure runs in $\mathrm{BQ} \cdot \mathrm{SPACE}(\mathrm{O}(\log \frac{n}{\epsilon\delta\zeta}))$.

5.3 Approximating the spectrum

We begin this section showing that we can uniformly sample an eigenvalue of H. The main thing to notice here is the way we prepare the input to the phase estimation procedure.

THEOREM 5.1. *There exists a* $\mathrm{DSPACE}(\mathrm{O}(\log \frac{n}{\epsilon\delta}))$*−uniform quantum circuit using* $O(\log \frac{n}{\epsilon\delta})$ *qubits that given a Hermitian matrix* H *with eigenvalues* $1 \ge \lambda_1 \ge \ldots \ge \lambda_n \ge -1$ *outputs a value* θ *according to a distribution that is* ϵ *close to a distribution that samples* $i \in \{1, \ldots, n\}$ *uniformly at random and outputs an element from* $[\lambda_i - \delta, \lambda_i + \delta]$.

Proof: The reduction outputs the phase estimation circuit described above (with or without the random shifts), with error parameter ϵ and accuracy parameter δ. The operator U is taken to be e^{iH}. For the input to the phase estimation problem (i.e. the initial value of register I) we choose $|k\rangle$ with equal probability over $k \in \{1, \ldots, n\}$.

For correctness, let $\{v_k\}$ be the set of eigenvectors of H. Notice that the reduced density matrix of the input register I is $\frac{1}{n}I$ and is the same as when choosing v_k with equal probability. On input v_k with eigenvalue $e^{\lambda_k i}$ the phase estimation procedure outputs, w.h.p., an approximation of $\lambda_k(\mathrm{mod} 2\pi)$, as desired. The space complexity is immediate. ■

As a corollary we see that we can estimate the condition number of well-conditioned matrices, and also prove a matrix is well-conditioned:

COROLLARY 5.1. *There exists a* $\mathrm{BQ} \cdot \mathrm{SPACE}(\mathrm{O}(\log \frac{n}{\epsilon\alpha}))$ *algorithm that given a Hermitian matrix* H *with singular values* $s_1 = 1$ *and* $s_n \ge \alpha$*, outputs a value* θ *such that*

$$\Pr[\; |\theta - s_n| \ge \alpha/2 \;] \le \epsilon.$$

Proof: Run the sampling algorithm n^2 times with $\alpha/2$ accuracy and ϵ/n^2 error, and output the smallest (approximation to) singular value seen. ■

Next we show how to approximate the whole spectrum with additive accuracy. For that we use consistent phase estimation (using random shifts) as described in Section 5.2. We define:

DEFINITION 5.1. *A sequence $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$ α additively-approximates a sequence $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ if for every $i \in [n]$, $|\mu_i - \lambda_i| \leq \alpha$. We denote this by $\overline{\mu} \approx_{+\alpha} \overline{\lambda}$.*

THEOREM 5.2. *There exists a $\mathrm{BQ \cdot SPACE(O(\log \frac{n}{\epsilon}))}$ algorithm that given a Hermitian matrix H with eigenvalues $1 \geq \lambda_1 \geq \ldots \geq \lambda_n \geq -1$, outputs a sequence $\overline{\mu}$ such that*

$$\Pr[\overline{\mu} \not\approx_{+\alpha} \overline{\lambda}(A)] \leq O(\epsilon).$$

Proof: W.l.o.g. $\epsilon \leq \frac{1}{4n}$. Set $T = n^3$, $\delta = \epsilon$, $\zeta = \frac{\epsilon}{n}$ and $\delta' = \frac{\delta\zeta}{2}$. Let $L = \lfloor \frac{2}{\zeta} \rfloor = \lfloor \frac{\delta}{\delta'} \rfloor$. We pick a shift s randomly from $\{1, \ldots, L\}$ and fix it. Except for probability $n\zeta = \epsilon$, s is good for all eigenvalues of H.

Our basic procedure is the consistent phase estimation algorithm described in Subsection 5.2, run with δ' accuracy, ϵ error and the fixed shift s. For each section, we run it T times, each time we measure the section and count the number S of times the approximated eigenvalue belongs to the section. Our estimate for the number of eigenvectors in the section is $\widetilde{k} = [\frac{S}{T} \cdot n]$.

Fix a section with k ($k \geq 0$) eigenvalues. For each eigenvalue, except for probability ϵ, the eigenvalue belongs to the section if and only if the approximated eigenvalue belongs there (because there are no eigenvalues δ' close to the border). Let Y_i be the Boolean random variable that is 1 when the i'th sampled approximated eigenvalue is in the section. Denote $\mu_i = \mathbb{E}(Y_i)$. $Y_1, \ldots, Y_T$ are independent, identically distributed Boolean random variables and $|\mu_i - \frac{k}{n}| \leq \epsilon$. As $|\frac{kT}{n} - \mathbb{E}(\sum Y_i)| \leq T\epsilon \leq \frac{T}{4n}$,

$$\begin{aligned} \Pr[\widetilde{k} \neq k] &\leq \Pr[\,|\sum Y_i - \frac{k}{n}T| \geq \frac{T}{2n}] \\ &\leq \Pr[\,|\sum Y_i - \mathbb{E}(\sum Y_i)| \geq \frac{T}{4n}]. \end{aligned}$$

By the additive version of the Chernoff bound:

$$\Pr[\widetilde{k} \neq k] \;\leq\; 2e^{-\frac{(1/4n)^2}{2}T} \leq 2^{-\Omega(T/n^2)} \leq 2^{-\Omega(n)}.$$

The probability we err for some section is at most $\frac{1}{\epsilon}2^{-\Omega(n)} \leq \epsilon$. Otherwise, we output an ϵ additive approximation of the spectrum. Clearly, the procedure can be made to run in L with oracle calls to the sampling procedure, which itself runs in $\mathrm{BQ \cdot SPACE(O(\log \frac{n}{\epsilon}))}$, completing the proof. ■

We remark that once we fix a good shift s, the output of the procedure is deterministic in the sense that there is a fixed sequence $\overline{\mu}$ (depending only on H and s) that is ϵ close to $\overline{\lambda}$ and such that with probability $1 - \epsilon$ the quantum algorithm outputs $\overline{\mu}$.

6. INVERTING A WELL-CONDITIONED MATRIX

Assume H is Hermitian with eigenvectors $v_1, \ldots, v_N$ and corresponding eigenvalues $\lambda_1, \ldots, \lambda_N$. Further assume that the eigenvalues are well separated, i.e., there exists some constant c s.t. for all $i \neq j$, $|\lambda_i - \lambda_j| \geq n^{-c}$. Then, using techniques similar to those used in Section 5, we can prepare a fixed mixed state that is evenly distributed over $\{v_i, \lambda_i\}$ (for the precise statement see Section 6.2). In such a case, we can use the fact that each time we get result λ_i we are in the same pure-state v_i, and apply quantum state tomography to learn in parallel (and hence in quantum low-space) the eigenvector v_i. Knowing both $\{v_i\}$ and $\{\lambda_i\}$, we can output in BQL a unitary matrix U and a positive, diagonal matrix D such that $H = UDU^\dagger$. Consequently, we can also output H in BQL. Moreover, when H is well-conditioned we can also output $H^{-1} = UD^{-1}U^\dagger$ in BQL.

In the general case, conditioned on getting result λ we are in a uniform mixture over all eigenvectors having that eigenvalue. We can still use quantum state tomography to learn the density matrix, but we do not know how to derive from it in small space an eigenbasis. Instead, we use ideas from [5] to approximate the operator H^{-1}, and then using quantum state tomography we learn the matrix of the operator H^{-1}.

The section is organized as follows. In Subsection 6.1 we state results from quantum state tomography in the bounded-space setting. In Subsection 6.2 we consider the case where the eigenvalues are well-separated, and in Subsection 6.3 the general case.

6.1 Quantum state tomography

THEOREM 6.1. *Suppose we can prepare as many copies as we wish of some mixed-state with density matrix $\rho \in \mathrm{HOM}(\mathbb{C}^n, \mathbb{C}^n)$. Furthermore, assume projections onto the standard basis and two-level unitaries are of no cost. Then:*

- *There exists a probabilistic procedure running in time $\mathrm{poly}(\frac{n}{\epsilon})$ and space $O(\log \frac{n}{\epsilon})$ that except for probability 2^{-n} outputs a fixed matrix (depending only on ρ and the offline randomness) that is $O(\epsilon)$ close in the ℓ_1 norm to the matrix $\rho = (\rho_{i,j})$.*
- *If, moreover, $\| \rho - |\psi\rangle\langle\psi| \|_{\mathrm{tr}} \leq \epsilon$ for some pure state $\psi = \sum \alpha_i |i\rangle$, then the procedure also outputs $(\alpha_1, \ldots, \alpha_n)$ with success probability $1 - 2^{-n}$, $O(n^2\epsilon)$ accuracy in the ℓ_∞ norm and $O(n^3\epsilon)$ accuracy in the ℓ_1 norm.*

Proof: For every $k, \ell \in [n]$ we apply the following four measurements, described by the following four POVMs $\left\{E_1^{(i)}, E_2^{(i)}\right\}$, for $i = 1, 2, 3, 4$:

1. $E_1^{(1)} = |k\rangle\langle k|$, $E_2^{(1)} = I - E_1^{(1)}$, which represents a projection onto k, and its orthogonal complement,
2. $E_1^{(2)} = |\ell\rangle\langle \ell|$, $E_2^{(2)} = I - E_1^{(2)}$, which represents a projection onto ℓ, and its orthogonal complement,
3. $E_1^{(3)} = |+\rangle\langle +|$, $E_2^{(3)} = I - E_1^{(3)}$, where $|+\rangle = \frac{1}{\sqrt{2}}[|k\rangle + |\ell\rangle]$, which represents a projection onto $|+\rangle$, and its orthogonal complement, and,
4. $E_1^{(4)} = |\uparrow\rangle\langle \uparrow|$, $E_2^{(4)} = I - E_1^{(4)}$, where $|\uparrow\rangle = \frac{1}{\sqrt{2}}[|k\rangle + i\,|\ell\rangle]$, which represents a projection onto $|\uparrow\rangle$, and its orthogonal complement.

Notice that all four measurements can be implemented by a simple two-level unitary followed by a projection on the standard basis. Let p_i denote the probability of getting result 1 when applying the i'th measurement (for $i = 1, 2, 3, 4$). By repeating the experiments $\mathrm{poly}(n/\epsilon')$ times, except for probability 2^{-2n} we approximate each p_i with ϵ' accuracy, for $\epsilon' = \Theta(\epsilon/n^2)$. However,

$$\begin{array}{lllll} p_1 & = & \mathrm{Tr}(E_1^{(1)}\rho) & = & \rho_{k,k} \\ p_2 & = & \mathrm{Tr}(E_1^{(2)}\rho) & = & \rho_{\ell,\ell} \\ p_3 & = & \mathrm{Tr}(E_1^{(3)}\rho) & = & \frac{1}{2}[\rho_{k,k} + \rho_{k,\ell} + \rho_{\ell,k} + \rho_{\ell,\ell}] \\ p_4 & = & \mathrm{Tr}(E_1^{(4)}\rho) & = & \frac{1}{2}[\rho_{k,k} - i\rho_{k,\ell} + i\rho_{\ell,k} + \rho_{\ell,\ell}] \end{array}$$

Thus, except for probability 2^{-n} we approximate each the four values $\rho_{k,k}, \rho_{k,\ell}, \rho_{\ell,k}, \rho_{\ell,\ell}$ with $O(\epsilon')$ accuracy. Doing that for all possible k and ℓ we reconstruct ρ with $O(\epsilon')$ accuracy in the ℓ_∞ norm, and therefore also with $O(n^2\epsilon') = O(\epsilon)$ accuracy in the ℓ_1 norm. Moreover, we can do that in $\mathrm{poly}(\frac{n}{\epsilon})$ time and $O(\log \frac{n}{\epsilon})$ space.

For the second item, assume $\| \rho - |\psi\rangle\langle\psi| \|_{\mathrm{tr}} \le \epsilon$. Then the measurements' probabilities are only $O(\epsilon)$ away from those produced by the ideal state $|\psi\rangle\langle\psi|$. Thus, $\rho_{k,k}$ is $O(\epsilon)$ close to $|\alpha_k|^2$, and the complex number $\rho_{k,\ell}$ is $O(\epsilon)$ close to $\alpha_k\alpha_\ell^*$ in the sense that the real (imaginary) part of $\rho_{k,\ell}$ is $O(\epsilon)$ close to the real (imaginary) part of $\alpha_k\alpha_\ell^*$.

Let k be the integer such that $|\alpha_k|$ is the largest. W.l.o.g we can assume α_k is a positive real number. As $\mathrm{Tr}(\rho) = 1$, $\alpha_k \ge 1/n$. By Claim 6.1, we approximate $1/\alpha_k$ with $O(n^2\epsilon)$ accuracy. Also, for every other ℓ, we can approximate $\alpha_k\alpha_\ell^*$ with $O(\epsilon)$ accuracy. Together, we approximate α_ℓ with $O(n^2\epsilon)$ accuracy, thus recovering ψ with $O(n^2\epsilon)$ accuracy in the ℓ_∞ norm. We therefore also recover ψ with $O(n^3\epsilon)$ in the ℓ_1 norm. ■

If ρ has high rank, learning a basis for ρ is equivalent to finding an eigenvector basis to a general positive-definite matrix, and we do not know how to do that in small quantum space.

Finally, we state without a proof the easy claim:

Claim 6.1. *Let $\Delta \ge 2\epsilon > 0$. Let $\theta > \Delta$ and assume $|\theta - \widetilde{\theta}| \le \epsilon$. Than, $|\frac{1}{\theta} - \frac{1}{\widetilde{\theta}}| \le \frac{2\epsilon}{\Delta^2}$.*

6.2 The case of well-separated eigenvalues

Theorem 6.2. *Fix a Hermitian operator H with eigenvectors v_i and eigenvalues λ_i such that $|\lambda_i - \lambda_j| \ge \alpha$ for all $i \ne j$. Then there exists a $\mathrm{BQ \cdot SPACE(O(\log \frac{n}{\epsilon\alpha}))}$ algorithm that with probability $1 - \epsilon$ outputs matrices $\widetilde{U}$ and $\widetilde{D}$ such that $\widetilde{U}$ is ϵ close to a unitary matrix U, $\widetilde{D}$ is ϵ close to a positive, diagonal matrix D and $H = U^\dagger D U$.*

Proof: The algorithm calls the consistent phase estimation protocol, and the approximated spectrum protocol defined in Section 5. We use the same shift for all the different invocations of these protocols, i.e., we first choose one shift, and all subsequent protocols use this fixed shift.

- We call the approximated spectrum algorithm, guaranteed by Theorem 5.2, with accuracy $\alpha' < \alpha$ and error ϵ' (ϵ' and α' will be defined later). The matrix $\widetilde{D}$ is obtained by putting the approximated spectrum on the diagonal. Clearly, except for probability ϵ', $\left\| D - \widetilde{D} \right\| \le n\alpha'$.
- For $\widetilde{U}$, notice that the i'th column of U is v_i. We output the matrix $\widetilde{U}$ column by column. For the i'th column, we first compute and keep the section number $s(\widetilde{\lambda}_i)$ of the i'th approximated eigenvalue, as computed by the approximated spectrum algorithm. We then use quantum-state tomography for pure-states as guaranteed by Theorem 6.1. When the tomography procedure calls the black-box algorithm for constructing the pure state, we do the following: We repeatedly call the consistent phase estimation procedure of Section 5.2 (up to n^2 times) with accuracy $\alpha' < \alpha$ and error ϵ' (to be determined later) and measure the section register S, until we get the section $s(\widetilde{\lambda}_i)$, and then we use the vector in register I.

Notice that once fixing the shift, calling the consistent phase estimation procedure creates a fixed mixed state represented by some fixed density matrix that we shall denote τ. Similarly, the reduced state conditioned on register S being $s(\widetilde{\lambda}_i)$ is also some fixed mixed state represented by some fixed density matrix that we shall denote τ_i. It is this density matrix the quantum state tomography procedure approximates.

It follows from Section 5.2 that for any $1 \le i \le n$:

$$\| \tau_i - |v_i, 0, s(\lambda_i)\rangle\langle v_i, 0, s(\lambda_i)| \|_{\mathrm{tr}} \le O(n\sqrt{\epsilon'}).$$

The second item of Theorem 6.1 now implies that except for probability 2^{-n} the quantum state tomography algorithm approximates the representation of the pure state v_i with $O(n^3\sqrt{\epsilon'})$ accuracy in the ℓ_∞ norm.

Picking $\alpha' = \min\{\alpha/2, \epsilon/n\}$ and $\epsilon' = \epsilon^2/n^{10}$, the algorithm outputs a matrix $\widetilde{U}$ such that $\left\| \widetilde{U} - U \right\|_\infty = O(n^3\sqrt{\epsilon'}) = O(\epsilon/n^2)$, hence $\left\| \widetilde{U} - U \right\| = O(\epsilon)$.

Applying the shift and truncate method on this algorithm, one gets a quantum algorithm that with probability $1 - \epsilon$ outputs one fixed matrix $\widetilde{U}$ such that $\left\| U - \widetilde{U} \right\| \le \epsilon$ as desired.

The running time of the algorithm is clearly polynomial: For each i, we run a quantum state tomography that takes $\mathrm{poly}(n/\epsilon)$ oracle calls, each requiring up to n^2 executions of the consistent phase estimation protocol. Similarly, for the space complexity we need several registers that take $O(\log(n/\epsilon))$ space. They include: running over i, running over the index j of v_i in the quantum state tomography protocol, running the quantum state tomography protocol itself, the n^2 executions of the consistent phase estimation when implementing an oracle call of the quantum state tomography protocol, and the consistent phase estimation protocol itself. Altogether the space complexity is $O(\log(n/\epsilon))$. ■

Finally, it is easy to see that if H is well conditioned then $\widetilde{U}D^{-1}\widetilde{U}^\dagger$ is a good approximation to H^{-1}.

Corollary 6.1. *Assume $H = UDU^\dagger$ as above and furthermore $s_n(H) \ge \epsilon$, $\left\| \widetilde{D} - D \right\| \le \epsilon$ and $\left\| \widetilde{U} - U \right\| \le \epsilon^2$. Then $\left\| H^{-1} - \widetilde{U}\widetilde{D}^{-1}\widetilde{U}^\dagger \right\| \le 3\epsilon$.*

We conclude with the proof of Theorem 1.2:

Proof: Given A define the Hermitian matrix $H = \left(\begin{array}{cc} 0 & A \\ A^\dagger & 0 \end{array} \right)$. Then, assuming $A = UDV$ with U, V unitary and D positive and diagonal, we have

$$H = \left(\begin{array}{cc} U & 0 \\ 0 & V^\dagger \end{array} \right) \cdot \left(\begin{array}{cc} 0 & D \\ D^\dagger & 0 \end{array} \right) \cdot \left(\begin{array}{cc} U^\dagger & 0 \\ 0 & V \end{array} \right).$$

Thus, we can read U, V and D from the decomposition of H. The proof now follows from the Hermitian case and Theorem 6.2. ■

6.3 The general case

We employ techniques from [5] to show:

THEOREM 6.3. *Fix a Hermitian operator H with $1 \geq s_1 \geq s_n \geq \zeta$. Let $\epsilon > 0$. Then there exists a* $\mathrm{BQ} \cdot \mathrm{SPACE}(\mathrm{O}(\log \mathrm{n} + \log \epsilon^{-1} + \log \zeta^{-1}))$ *algorithm that given v, with probability $1 - 2^{-n}$ outputs a value that is ϵ close to $\left\| H^{-1} \left| v \right\rangle \right\|$, and a mixed state that is $O(\epsilon)$ close to the density matrix $\frac{1}{\left\| H^{-1} v \right\|^2} \left| H^{-1} v \right\rangle \left\langle H^{-1} v \right|$.*

Proof: As before we pick a random shift that is used throughout the protocol and fix it.

We first describe a basic protocol that is later used as a building block. We have registers I, E, S as in Section 5, and also an additional one qubit register A. We do the following:

1. We apply the consistent phase estimation procedure from Section 5.2 with ϵ' and α' to be determined later, and get a vector in $I \otimes E \otimes S$.

2. We then implement in low space a transformation over $S \otimes A$ mapping $\left| s, 0 \right\rangle_{S \otimes A}$, where s is the section number and $\widetilde{\lambda}$ the eigenvalue associated with it, and when $\widetilde{\lambda} \geq \widetilde{\zeta} \stackrel{\mathrm{def}}{=} \frac{\zeta}{2}$, ϵ' close to $\left| s \right\rangle \otimes [\frac{\widetilde{\zeta}}{\widetilde{\lambda}} \left| 0 \right\rangle + \sqrt{1 - (\frac{\widetilde{\zeta}}{\widetilde{\lambda}})^2} \left| 1 \right\rangle]$ (if $\widetilde{\lambda} < \widetilde{\zeta}$ we apply the identity transformation). One way of implementing this is as follows:

   (a) Calculate $\widetilde{\lambda}$. Approximate $(\widetilde{\lambda})^{-1}$ with ϵ' accuracy (notice that this is possible by Claim 6.1).

   (b) Approximate the four entries of the 2×2 rotation map rotating $\left| 0 \right\rangle$ to $\frac{\widetilde{\zeta}}{\widetilde{\lambda}} \left| 0 \right\rangle + \sqrt{1 - (\frac{\widetilde{\zeta}}{\widetilde{\lambda}})^2} \left| 1 \right\rangle$.

   (c) Use Theorem 4.2 to simulate the transformation with ϵ' accuracy.

3. Reverse the consistent phase estimation procedure on registers $I \otimes E$.

4. Measure register A.

Repeat the above procedure until the measurement in Step (4) returns 0, and then output register I.

We now analyze the basic protocol: suppose we apply the basic protocol on $v = \sum \alpha_i v_i$. Let us denote by $\rho^{(i)}$, for $i = 1, 2, 3, 4$, the density matrix of the mixture created after the i'th step. Step (1) creates a density matrix $\rho^{(1)}$ that is $O(\sqrt{\epsilon'})$ close to the density matrix $\rho^{(1)}_{ideal}$ of the pure state

$$\sum_i \alpha_i \left| v_i \right\rangle_I \left| s(i) \right\rangle_S \otimes \left| \overline{0} \right\rangle_E \otimes \left| \overline{0} \right\rangle_A ,$$

where $s(i)$ is the section of λ_i. Step (2) creates a density matrix $\rho^{(2)}$ that is $O(\sqrt{\epsilon'})$ close to the density matrix $\rho^{(2)}_{ideal}$ of the pure state:

$$\sum_i \alpha_i \left| v_i \right\rangle_I \left| s(i) \right\rangle_S \otimes [\frac{\widetilde{\zeta}}{\lambda_i} \left| 0 \right\rangle_A + \sqrt{1 - (\frac{\widetilde{\zeta}}{\lambda_i})^2} \left| 1 \right\rangle_A] \otimes \left| \overline{0} \right\rangle_E .$$

Reversing the consistent phase estimation procedure reverses each $\left| v_i \right\rangle_I \left| s(i) \right\rangle_S$ separately close to $\left| v_i \right\rangle_I \otimes \left| \overline{0} \right\rangle_S$, and altogether step (3) creates a density matrix $\rho^{(3)}$ that is $O(\sqrt{\epsilon'})$ close to the density matrix $\rho^{(3)}_{ideal}$ of the pure state

$$\sum_i \alpha_i \left| v_i \right\rangle_I \otimes [\frac{\widetilde{\zeta}}{\lambda_i} \left| 0 \right\rangle_A + \sqrt{1 - (\frac{\widetilde{\zeta}}{\lambda_i})^2} \left| 1 \right\rangle_A] \otimes \left| \overline{0} \right\rangle_E \otimes \left| \overline{0} \right\rangle_S .$$

In the ideal mixture $\rho^{(3)}_{ideal}$, when we measure register A and conditioning on getting 0, we fall into a state proportional to:

$$\phi^{(4)}_{ideal} \;=\; H^{-1} \left| v \right\rangle \;=\; \sum_i \frac{\alpha_i}{\lambda_i} \left| v_i \right\rangle .$$

with probability

$$\Delta_{ideal} \stackrel{\mathrm{def}}{=} \widetilde{\zeta}^2 \sum \frac{\alpha_i^2}{\lambda_i^2} = \widetilde{\zeta}^2 \left\| H^{-1} \left| v \right\rangle \right\|^2 .$$

Notice that Δ_{ideal} is at least $\Omega(\zeta^2)$. Also, the reduced density matrix conditioned on getting a zero is

$$\rho_{ideal} \;=\; \frac{1}{\left\| H^{-1} v \right\|^2} \left| H^{-1} v \right\rangle \left\langle H^{-1} v \right| .$$

Let Δ denote the probability we get 0 in the actual process $\rho^{(3)}$. We saw that $\left| \Delta - \Delta_{ideal} \right| \leq O(\sqrt{\epsilon'})$. Repeating the basic protocol $\mathrm{poly}(\frac{n}{\epsilon'})$ times, we can (w.h.p.) approximate Δ with $\sqrt{\epsilon'}$ accuracy. Therefore, we can also approximate Δ_{ideal} with $O(\sqrt{\epsilon'})$ accuracy. Therefore, we can approximate $\left\| H^{-1} \left| v \right\rangle \right\|^2$ with $O(\frac{\sqrt{\epsilon'}}{\zeta^2})$ accuracy. Taking $\epsilon' = O(\epsilon^2 \zeta^4)$, we approximate $\left\| H^{-1} \left| v \right\rangle \right\|$ with $O(\epsilon)$ accuracy.

Moreover, $\Delta_{ideal} \geq \widetilde{\zeta}^2$ and therefore $\Delta \geq \Delta' - O(\sqrt{\epsilon'}) \geq \Omega(\zeta^2)$. Thus, repeating the basic protocol $O(\frac{n}{\zeta^2})$ times, we almost surely (except for probability 2^{-n}) get 0 in one of the attempts. Moreover, let $\rho^{(4)}$ be the mixed state obtained by applying step (4) on $\rho^{(3)}$ and conditioned on getting 0. Then,

$$\left\| \rho^{(4)} - \rho_{ideal} \right\|_{\mathrm{tr}} \leq O(\frac{\sqrt{\epsilon'}}{\zeta^2}) = O(\epsilon).$$

Thus, $\rho^{(4)}$ is $O(\epsilon)$ close to the state $H^{-1} \left| v \right\rangle$ normalized. ■

Having that, we can apply H^{-1} on e_i to learn the i'th column of H. Using quantum state tomography on the state (which we can repeatedly generate) we learn the i'th column of H, and doing it for all columns we learn the matrix H. Formally,

THEOREM 6.4. *Fix a Hermitian operator H with $1 \geq s_1 \geq s_n \geq \zeta$. Let $\epsilon > 0$. Then there exists a* $\mathrm{BQ} \cdot \mathrm{SPACE}(\mathrm{O}(\log \mathrm{n} + \log \epsilon^{-1} + \log \zeta^{-1}))$ *algorithm that with probability $1 - \epsilon$ outputs one matrix (depending only on the input and offline randomness) that approximates the matrix H^{-1} with ϵ accuracy in the ℓ_1 norm.*

Proof: Let e_i be the all zero vector except in the i'th coordinate. We run the above procedure on $v = e_i$, repeatedly generating (w.h.p.) the mixture $\rho^{(4)}$ that is $O(\epsilon')$ close to the density matrix of the pure state $\frac{H^{-1}|v\rangle}{\|H^{-1}|v\rangle\|}$, for $\epsilon' = \Theta(\epsilon/n^4)$. Running quantum state tomography on $\rho^{(4)}$ we approximate the representation of $\frac{H^{-1}|v\rangle}{\|H^{-1}|v\rangle\|}$ with $O(n^2\epsilon') = O(\epsilon/n^2)$ accuracy in the ℓ_∞ norm, as guaranteed by the second item of Theorem 6.1.

Also, we can estimate $\|H^{-1}|v\rangle\|$ with ϵ' accuracy, and therefore we can also approximate the entries of the vector $H^{-1}|v\rangle$ itself with $O(\epsilon/n^2)$ accuracy in the ℓ_∞ norm. The theorem now follows by noticing that $H^{-1}e_i$ is the vector corresponding to the i'th column of H^{-1}. Running the procedure for $i = 1, \ldots, n$ we output approximations of the columns of the matrix H^{-1} one by one with ϵ accuracy in the ℓ_1 norm as required.

Finally, using the shift and truncate method, we get an algorithm that with probability $1 - \epsilon$ outputs one matrix (depending only on the input and offline randomness) that is ϵ close to H^{-1} in the ℓ_1 norm. ■

The proof of Theorem 1.1 is based on Theorem 6.4 in a similar way to the proof of Theorem 1.2.

Acknowledgment

I would like to thank Klim Efremenko and Dean Doron for useful and interesting conversations.

7. REFERENCES

[1] Quantum algorithm zoo. Available at http://math.nist.gov/quantum/zoo.

[2] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation. *SIAM Journal on Computing*, 37(1):47, 2007.

[3] A. Ambainis. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations. *arXiv preprint arXiv:1010.4458*, 2010.

[4] S. Cook. A taxonomy of problems with fast parallel algorithms. *Information and control*, 64(1):2–22, 1985.

[5] A. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[6] A. Kitaev, A. Shen, and M. Vyalyi. *Classical and quantum computation*, volume 47. Amer Mathematical Society, 2002.

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

[8] M. Saks and S. Zhou. BPHSPACE(S) $\subseteq$ DSPACE($S^{3/2}$). *Journal of Computer and System Sciences*, 58(2):376–403, 1999.

[9] P. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. IEEE, 1994.

[10] D. van Melkebeek and T. Watson. Time-space efficient simulations of quantum computations. *Theory of Computing*, 8:1–51, 2012.

[11] J. Watrous. Space-bounded quantum complexity. *J. Comput. Syst. Sci.*, 59(2):281–326, 1999.

[12] J. Watrous. On the complexity of simulating space-bounded quantum computations. *Computational Complexity*, 12(1):48–84, 2003.

[13] A. Yakaryılmaz and A. Say. Unbounded-error quantum computation with small space bounds. *Information and Computation*, 209(6):873–892, 2011.

APPENDIX

A. APPROXIMATING TWO-LEVEL TRANSFORMATIONS

Proof: (Of Claim 4.2) The two-level unitary $U = e^{iH}$ acts non-trivially on a 2×2 subspace $W_{k,\ell} = Span\{|k\rangle, |\ell\rangle\}$ spanned by two vectors in the computational basis $\{|j\rangle\}_{j=1}^n$. Our goal is to approximate U using only CNOT and single-qubit operations. Now,

- Following [7, Section 4.5.2] the problem can be reduced to implementing controlled-NOT and controlled-$\dot{U}$ operations, where the control is over *multiple* qubits, and $\dot{U}$ is the 2×2 unitary transformation defined by U on $W_{k,\ell}$.
- An explicit way of implementing a CNOT with multiple control bits is given in [6, sec 8.1.3].
- To implement the controlled-$\dot{U}$ with multiple control bits, we again follow, e.g, [7, Section 4.3], and find three single-qubit unitary operators A, B and C such that $ABC = I$ and $AXBXC = \dot{U}$. Then the problem of implementing controlled-$\dot{U}$ with multiple control bits is reduced to implementing controlled-X (i.e., CNOT) with multiple control bits, that we already solved.

To follow this recipe we first need to approximate the 2×2 matrix representing $\dot{U}$. We can do that by finding the eigenvalues and eigenvectors of H on $W_{k,\ell}$ with ϵ accuracy, then exponentiating the eigenvalues and computing the unitary 2×2 matrix.

Given a matrix $\begin{pmatrix} v_{1,1} & v_{1,2} \\ v_{2,1} & v_{2,2} \end{pmatrix}$ representing the 2×2 unitary $\dot{U}$, our goal is to find the required operators A, B and C. This can be done by finding approximations to real numbers α, β, γ and δ such that

$$\dot{U} = \begin{pmatrix} e^{i(\alpha-\beta/2-\delta/2)} \cos\frac{\gamma}{2} & -e^{i(\alpha-\beta/2+\delta/2)} \sin\frac{\gamma}{2} \\ e^{i(\alpha+\beta/2-\delta/2)} \sin\frac{\gamma}{2} & -e^{i(\alpha+\beta/2+\delta/2)} \cos\frac{\gamma}{2} \end{pmatrix},$$

and then following the formula given in the proof of [7, Corollary 4.2]. By computing

$$\begin{aligned} det(\dot{U}) &= v_{1,1}v_{2,2} - v_{1,2}v_{2,1} \\ &= -e^{2i\alpha}\cos(\gamma), \end{aligned}$$

we approximate γ up to sign (because $|\cos(\gamma)| = |det(\dot{(U)})|$. For each of the two solutions for γ, we find α and then solve the remaining equations to approximate β and δ.

The above procedure can be worked out with ϵ accuracy in space $O(\log n + \log \epsilon^{-1})$, using the fact that the basic arithmetic operations are in L and the explicit nature of the Taylor sums of the Trigonometric functions. ■