Adiabatic Quantum State Generation and Statistical Zero Knowledge

Dorit Aharonov *
Hebrew University, Jerusalem, Israel
doria@cs.huji.ac.il

Amnon Ta-Shma
Tel Aviv University, Tel-Aviv, Israel
amnon@tau.ac.il

ABSTRACT

The design of new quantum algorithms has proven to be an extremely difficult task. This paper considers a different approach to the problem, by studying the problem of 'quantum state generation'.

We first show that any problem in Statistical Zero Knowledge (including eg. discrete log, quadratic residuosity and gap closest vector in a lattice) can be reduced to an instance of the quantum state generation problem. Having shown the generality of the state generation problem, we set the foundations for a new paradigm for quantum state generation. We define 'Adiabatic State Generation' (ASG) , which is based on Hamiltonians instead of unitary gates. We develop tools for ASG including a very general method for implementing Hamiltonians (The sparse Hamiltonian lemma), and ways to guarantee non negligible spectral gaps (The jagged adiabatic path lemma). We also prove that ASG is equivalent in power to state generation in the standard quantum model. After setting the foundations for ASG, we show how to apply our techniques to generate interesting superpositions related to Markov chains.

The ASG approach to quantum algorithms provides intriguing links between quantum computation and many different areas: the analysis of spectral gaps and groundstates of Hamiltonians in physics, rapidly mixing Markov chains, statistical zero knowledge, and quantum random walks. We hope that these links will bring new insights and methods into quantum algorithms.

Categories and Subject Descriptors

F.2.m [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Miscellaneous*; F.1.3 [**Theory of Computation**]: Computation by abstract devices— *complexity measures and classes*

*D.A. was supported in part by the Institute for Quantum Information through National Science Foundation Grant No. EIA-0086038.

STOC'03, June 9–11, 2003, San Diego, California, USA.

General Terms

Theory, Algorithms

Keywords

Quantum adiabatic computation, quantum sampling, Hamiltonian, spectral gap, state generation, Markov chains, statistical zero knowledge

1. INTRODUCTION

Quantum computation carries the hope of solving in quantum polynomial time classically intractable tasks. The most notable success so far is Shor's quantum algorithm for factoring integers and for finding the discrete log [40]. Following Shor's algorithm, several other algorithms were discovered [27, 44, 13], all heavily relying on the Fourier transform. A new black box algorithm was recently discovered which uses a different approach of quantum random walks, but the problem it solves is somewhat contrived [9]. One cannot overstate the importance of developing qualitatively different quantum algorithmic techniques and approaches for the development of the field of quantum computation. In this paper we attempt to make a step in that direction by approaching the issue of quantum algorithms from a different point of view.

It is folklore knowledge that the problem of graph isomorphism which is considered classically hard [32] has an efficient quantum algorithm as long as the superposition of all graphs isomorphic to a given graph, $|\alpha_G\rangle = \sum_{\sigma \in S_n} |\sigma(G)\rangle$ can be generated efficiently by a quantum Turing machine (for simplicity, we ignore normalizing constants in the above state and in the rest of the paper). To see this notice that for two isomorphic graphs G_1 and G_2, $|\alpha_{G_1}\rangle$ and $|\alpha_{G_1}\rangle$ are identical, whereas for two non isomorphic graphs they are orthogonal. A simple circuit can then distinguish between these two cases. One is tempted to assume that $|\alpha_G\rangle$ is easy to construct since the equivalent classical distribution, namely the uniform distribution over all graphs isomorphic to a certain graph, can be sampled from efficiently. Indeed, the state $|\beta_G\rangle = \sum_{\sigma \in S_n} |\sigma\rangle \otimes |\sigma(G)\rangle$ can be efficiently generated on a quantum computer by this argument; However, so far no one knows how to generate $|\alpha_G\rangle$ efficiently, because we cannot *forget* the value of $|\sigma\rangle$.

In this paper we systematically study the problem of quantum state generation. We will mostly be interested in a restricted version of state generation, namely generating states corresponding to classical probability distributions, which we loosely refer to as *quantum sampling* (or *Qsam-*

pling) from a distribution. To be more specific, we consider *the probability distribution of a circuit,* D_C , which is the distribution over the outputs of the classical circuit C when its inputs are uniformly distributed. Denote $|C\rangle \stackrel{\text{def}}{=} \sum_{z\in\{0,1\}^m} \sqrt{D_C(z)}\,|z\rangle$. We define the problem of circuit quantum sampling:

Definition 1. **Circuit Quantum Sampling (CQS):**
Input: (ϵ, C) where C is a description of a classical circuit from n to m bits, and $0 \leq \epsilon \leq \frac{1}{2}$.
Output: A description of a quantum circuit Q of size $poly(|C|)$ such that $|Q(|\vec{0}\rangle) - |C\rangle\,| \leq \epsilon$.

Connection to Statistical Zero Knowledge. Most problems that were considered good candidates for BQP, such as discrete log (DLOG), quadratic residuosity, gap versions of closest and shortest vectors in a lattice, and graph isomorphism, belong to the complexity class *statistical zero knowledge*, denoted SZK (see section 2 for background.) We prove

Theorem 1. Any $\mathcal{L} \in SZK$ can be reduced to a family of instances of CQS.

The proof relies on a reduction by Sahai and Vadhan [39] from SZK to a complete problem called statistical difference. Theorem 1 shows that a general solution for quantum sampling would imply $SZK \subseteq BQP$. We note that there exists an oracle A relative to which $SZK^A \not\subset BQP^A$ [1], and so such a proof must be non relativizing.

Theorem 1 shows that one can start from a SZK proof for a problem, and derive a description of classical circuits such that efficient Quantum sampling from these circuits would then imply that the problem is in BQP. The derivation of the circuit specification is in general a complicated task, since it builds on the reduction of Sahai and Vadhan [39] from SZK to the complete problem.

We provide one explicit example which is of particular interest as a candidate for BQP: a gap version of closest vector in a lattice (CVP). Unlike the general construction, this reduction from SZK to Qsampling is fairly straightforward. We use the SZK proof of Goldreich and Goldwasser [21] to derive an exact specification of the corresponding circuits which one needs to Qsample from, in order to give a polynomial quantum algorithm for this problem.

Explicit specifications of the circuits to Qsample from can be derived also for the problems of discrete log (DL) and quadratic residuosity (QR), based on the the SZK proofs for these problems by Goldreich and Kushilevitz [20], and by Goldwasser, Micali and Rackoff [23], respectively. Like in the case of CVP, the derivations are fairly straightforward and are similar conceptually to the CVP case. We will not do it in this paper due to lack of space, and since these problems are already known to be in BQP.

The Adiabatic State Generation Paradigm The problem of what states can be generated efficiently by a quantum computer is thus of critical importance to the understanding of the computational power of quantum computers. We therefore embark on the task of designing tools for quantum state generation, and studying which states can be generated efficiently. The recently suggested framework of adiabatic quantum computation [19] seems to be tailored exactly for this purpose, since it is stated in terms of quantum state generation; Let us first explain this framework.

Recall that the time evolution of a quantum system's state $|\psi(t)\rangle$ is described by Schrodinger's equation:

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle. \quad (1)$$

where $H(t)$ is an Hermitian operator called the *Hamiltonian* of the system. Adiabatic evolution concerns the case in which $H(t)$ varies very slowly in time; The qualitative statement of the adiabatic theorem is that if the quantum system is initialized in the ground state (the eigenstate with lowest eigenvalue) of $H(0)$, and if the modification of H in time is done slowly enough (*adiabatically*), then the final state will be the ground state of the final Hamiltonian $H(T)$. Recently, Farhi, Goldstone, Gutmann and Sipser [19] suggested to use adiabatic evolutions to solve NP-hard languages. Farhi *et. al.*'s idea was to find the minimum of a given function f as follows: $H(0)$ is chosen to be some generic Hamiltonian. $H(T)$ is chosen to be the *problem Hamiltonian*, namely a matrix which has the values of f on its diagonal and zero everywhere else. The system is then initialized in the ground state of $H(0)$ and evolves adiabatically on the convex line $H(t) = (1 - \frac{t}{T})H_0 + \frac{t}{T}H_T$. By the adiabatic theorem if the evolution is slow enough, the final state will be the groundstate of $H(T)$ which is exactly the sought after minimum of f.

The efficiency of these adiabatic algorithms is determined by two things. First, H is taken to be *local*, i.e. a sum of operators, each operating on a constant number of qubits out of the n qubits of the system. It was shown in [19, 14] that adiabatic evolutions of such Hamiltonians can be simulated efficiently on a quantum circuit, and so designing such a successful process would imply a quantum efficient algorithm for the problem. Second, the adiabatic evolution has to be slow enough for the adiabatic theorem to hold. It turns out that this depends mainly on the *spectral gaps* of the Hamiltonians $H(t)$. If these spectral gaps are not too small, the modification of the Hamiltonians can be done 'fairly fast', and the adiabatic algorithm then becomes efficient. Not much is known about adiabatic computation, since spectral gaps are hard to analyze in general. [16, 11, 17] study numerically the performance of adiabatic algorithms on random instances of NP complete problems. [14, 38] proved that Grover's quadratic speed up [25] can be achieved adiabatically, and [14, 15] also give lower bounds for specific cases of adiabatic algorithms.

In this paper, we propose to use the language of Adiabatic evolutions, Hamiltonians, ground states and spectral gaps as a theoretical framework for *quantum state generation*. Our goal is not to replace the quantum circuit model, neither to improve on it, but rather to develop a paradigm, or a language, in which quantum state generation can be studied conveniently. The advantage in using the Hamiltonian language is that the task of quantum state generation becomes much more natural, since adiabatic evolution is cast in the language of state generation. Furthermore, as we will see, it seems that this language lends itself more easily than the standard circuit model to developing general tools.

In order to provide a framework for the study of state generation using the adiabatic language, we define *adiabatic state generation* (ASG) as general as we can. Thus, we replace the requirement that the Hamiltonians are on a straight line $H(t) = (1 - \frac{t}{T})H_0 + \frac{t}{T}H_T$, with Hamiltonians on any general path. Second, we replace the requirement

that the Hamiltonians are local, with the requirement that they are *simulatable*, i.e., that the unitary matrix $e^{-itH(s)}$ can be approximated by a quantum circuit to within any polynomial accuracy for any polynomially bounded time t. We thus still use the standard model of quantum circuits in our paradigm to *simulate* the adiabatic process. Our goal is therefore to derive quantum circuits solving the state generation problem, from state generation algorithms cast in the ASG framework.

The fact that any ASG can be simulated efficiently by a quantum circuit, follows from the adiabatic theorem and from a generalization of techniques in [19, 14]. An alternative proof which does not rely on the adiabatic theorem is given in a fuller version of this paper [2], using the much simpler Zeno effect [37]. See [10] for a related approach of measurement based adiabatic computation.

Foundations of ASG. The first question that one encounters is naturally, what kind of Hamiltonians can be used in ASG algorithms. In other words, when is it possible to simulate, or implement, a Hamiltonian efficiently. To this end we prove the *sparse Hamiltonian lemma* which gives a very general condition for a Hamiltonian to be simulatable. A Hamiltonian H on n qubits is row-sparse if the number of non-zero entries at each row is polynomially bounded. H is said to be row-computable if there exists a (quantum or classical) efficient algorithm that given i outputs a list $(j, H_{i,j})$ running over all non zero entries $H_{i,j}$. As a norm for Hamiltonians we use the spectral norm (see section 3.1.1).

Lemma 1. **(The sparse Hamiltonian lemma)**. If H is a row-sparse, row-computable Hamiltonian on n qubits and $||H|| \leq poly(n)$, then H is simulatable.

We note that this lemma is useful also in two other contexts: first, in the context of simulating complicated physical systems on a quantum circuit. Second, for efficient implementation of continuous quantum walks [12] which are based on Hamiltonians. For example, lemma 1 can be used to simplify the Hamiltonian implementation in the recently discovered exponential quantum speed up using quantum walks [9].

The next question that one encounters in designing ASG algorithms concerns bounding the spectral gap. We need tools to find paths in the Hamiltonian space such that the spectral gaps are guaranteed to be non negligible, i.e. larger than $1/poly(n)$. Our next lemma provides a way to do this in certain cases. Denote $\alpha(H)$ to be the ground state of H (if unique.)

Lemma 2. **(The Jagged Adiabatic Path lemma)**. Let $\{H_j\}_{j=1}^{T=poly(n)}$ be a sequence of simulatable Hamiltonians on n qubits, all with polynomially bounded norm, non-negligible spectral gaps and with groundvalues 0, such that the inner product between the unique ground states $\alpha(H_j)$ and $\alpha(H_{j+1})$ is non negligible for all j. Then there is an efficient quantum algorithm that takes $\alpha(H_0)$ to within arbitrarily small distance from $\alpha(H_T)$.

To prove this lemma we develop two simple but very useful tools for manipulating Hamiltonians in the context of ASG: The Hamiltonian-to-Projection Lemma, (Lemma 4) and the Two Dimensional Adiabatic Lemma (Lemma 5).

Finally, we use the above developed tools to show that the question of the complexity of quantum state generation is equivalent (up to polynomial terms) in the circuit model and in the ASG model, and so it is sufficient to study state generation in the ASG paradigm.

Theorem 2. $|\alpha\rangle$ can be efficiently generated in the circuit model iff it can be efficiently generated by ASG.

Using ASG for Markov chain related states In the final part of the paper we demonstrate how our methods for ASG work for Qsampling from the limiting distributions of Markov chains. A Markov chain is rapidly mixing if and only if the second eigenvalue gap, namely the difference between the largest and second largest eigenvalue of the Markov matrix M, is non negligible [4]. This clearly bears resemblance to the adiabatic condition of a non negligible spectral gap, and suggests to look at Hamiltonians of the form $H_M = I - M$. H_M will be a Hamiltonian if M is symmetric; if M is not symmetric but is a reversible Markov chain [34] we can still define the Hamiltonian corresponding to it (see section 4.) The sparse Hamiltonian lemma has as an immediate corollary that for a special type of Markov chains, which we call *strongly samplable*, the quantum analog of the Markov chain can be implemented:

Corroloary 1. If M is a strongly samplable Markov chain, then H_M is simulatable.

To apply the ASG framework for Qsampling from limiting distributions of Markov chains, it is natural to consider *sequences* of Markov chains, where each Markov chain in the sequence is close (in some well defined sense) to the next Markov chain in the sequence. Such sequences appear naturally in the context of a very commonly used paradigm in randomized algorithms, namely approximate counting [29]. We show:

Theorem 3. (Loosely:) Let A be an efficient randomized algorithm to approximately count a set Ω, which uses slowly varying Markov chains starting from a simple Markov chain. Then there is an efficient quantum algorithm Q that Qsamples from the final limiting distribution over Ω.

We stress that it is NOT the case that we are interested in a quantum speed up for sampling from various distributions but rather we are interested in the efficient generation of the quantum state corresponding to the classical distribution.

Essentially all Markov chains that are used in approximate counting that we are aware of meet the criteria of the theorem. The following is a partial list of states we can Qsample from using Theorem 3, where the citations refer to the approximate algorithms that we use as the basis for the quantum sampling algorithm. Uniform superposition over all perfect matchings of a given bipartite graph [28], all spanning trees of a given graph [8], all lattice points contained in a high dimensional convex body satisfying the conditions of [6], various Gibbs distribution over rapidly mixing Markov chains using the Metropolis filter [34], and log-concave distributions [6]. We note that some of these states (perhaps all) can be generated using standard techniques which exploit the self reducibility of the problem (see [26]). We stress however that our techniques are qualitatively and significantly different from previous techniques for generating quantum states, and in particular do not require self reducibility. This can be important for extending this approach to other states.

Connections to other areas and Open Questions In this paper we suggest to use the language of Hamiltonians and spectral gaps. This direction points at very interesting and intriguing connections between quantum computation and many different areas: SZK, adiabatic evolution, rapidly mixing Markov chains and their spectral gap analysis [34], quantum walks [9], and the study of ground states and spectral gaps of Hamiltonians in Physics, which is a lively area of research (see [41] and references therein). Hopefully, these connections will bring techniques and insights from these fields to quantum computation. As an interesting first step, it would be insightful to find alternative algorithms to quadratic residuosity and discrete log by ASG of the states derived by theorem 1 for these problems.

Organization of paper The rest of the paper is divided to three sections which are almost completely independent. Section 2 gives the results related to SZK. Section 3 gives the foundations for the ASG framework: definition, tools, and the equivalence to standard state generation. Section 4 draws the connection to Markov chains and shows how to use ASG to Qsample from approximately countable sets.

2. QSAMPLING AND SZK

2.1 SZK

The complexity class *Statistical Zero Knowledge* (SZK) is defined using interactive proofs systems. Here we omit this definition, since it is not needed for this paper. Instead we will use the characterization of SZK by a complete problem for SZK, called statistical difference (SD), which we describe shortly. SD was recently shown to be complete for SZK by Sahai and Vadhan [39]. The nice thing about SD is that it does not mention interactive proofs in any explicit or implicit way. For excellent sources on SZK see [43, 39]. It is known that $BPP \subseteq SZK \subseteq AM \cap coAM$ and that SZK is closed under complement. It follows (see [43]) that SZK does not contain any NP–complete language unless the polynomial hierarchy collapses.

2.2 Statistical Difference (SD)

We need some facts about distances between distributions to define the complete problem SD. For two classical distributions $\{p(x)\}, \{q(x)\}$ define their ℓ_1 distance and their *fidelity* (this measure is known by many other names as well):

$$\begin{aligned} |p-q|_1 &= \sum_x |p(x)-q(x)| \\ F(p,q) &= \sum_x \sqrt{p(x)q(x)} \end{aligned}$$

We also define the variation distance to be $||p-q|| = \frac{1}{2}|p-q|_1$ so that it is a value between 0 and 1.

Fact 1. (See [36]) $1 - F(p,q) \leq ||p-q|| \leq \sqrt{1-F(p,q)^2}$.

We can now define the complete problem for SZK:

Definition 2. **Statistical Difference ($SD_{\alpha,\beta}$)**

Input : Two classical circuits C_0, C_1 with m output bits.

Promise : $||D_{C_0} - D_{C_1}|| \geq \alpha$ or $||D_{C_0} - D_{C_1}|| \leq \beta$.

Output : Which of the two possibilities occurs? (*yes* for the first case and *no* for the second)

Sahai and Vadhan [39, 43] show that for any two constants $0 \leq \beta < \alpha \leq 1$ where $\alpha^2 > \beta$, $SD_{\alpha,\beta}$ is complete for SZK.

2.3 Reduction from SZK to Qsampling.

To prove Theorem 1 we need a very simple building block which can be proved by direct calculation:

Claim 1. Let $\psi = \frac{1}{\sqrt{2}}(|0,v\rangle + |1,w\rangle)$, apply a Hadamard gate on the first qubit and measure it. The probability of answer 0 is $\frac{1+Real(\langle v|w\rangle)}{2}$.

PROOF. (of Theorem 1) We show that $SD_{1/4,3/4} \subseteq BQP$. Let C_0, C_1 be an input to an $SD_{1/4,3/4}$ problem, and say C_0, C_1 are circuits with m outputs. Let us first assume that we can Qsample from both circuits with $\epsilon = 0$ error. We can therefore generate the superposition $\frac{1}{\sqrt{2}}(|0\rangle|C_0\rangle + |1\rangle|C_1\rangle)$. We then apply a Hadamard gate on the first qubit and measure it. Using claim 1 with $v = |C_0\rangle$, $w = |C_1\rangle$, we have:

$$\langle v|w\rangle = \sum_{z\in\{0,1\}^m} \sqrt{D_{C_0}(z)D_{C_1}(z)} = F(D_{C_0}, D_{C_0}) \qquad (2)$$

We therefore get 0 with probability $\frac{1+F(D_{C_0},D_{C_0})}{2}$. Thus,

- If $||D_{C_0} - D_{C_1}|| \geq \alpha$, then we measure 0 with probability $\frac{1+F(D_{C_0},D_{C_0})}{2} \leq \frac{1+\sqrt{1-||D_{C_0}-D_{C_1}||^2}}{2} \leq \frac{1+\sqrt{1-\alpha^2}}{2} \leq 0.831$, while,
- If $||D_{C_0} - D_{C_1}|| \leq \beta$, then we measure 0 with probability $\frac{1+F(D_{C_0},D_{C_0})}{2} \geq \frac{2-||D_{C_0}-D_{C_1}||}{2} \geq 1 - \frac{\beta}{2} \geq \frac{7}{8} = 0.875$.

Repeating the experiment $O(\log(\frac{1}{\delta}))$ times, we can decide on the right answer with error probability smaller than δ. If the Qsampling circuit has a small error (say $\epsilon < \frac{1}{100}$) then the same argument holds with small corrections. □

2.4 Reduction of gapCVP to Qsampling

A lattice of dimension n is represented by a basis, denoted B, which is an $n \times n$ non-singular matrix over $\mathbb{R}$. The lattice $\mathcal{L}(B)$ is the set of points $\mathcal{L}(B) = \{Bw \mid w \in \mathbb{Z}^n\}$, i.e., all integer linear combinations of the columns of B. The distance $d(v_1, v_2)$ between two points is the Euclidean distance ℓ_2. The distance between a point v and a set $\mathcal{A}$ is $d(v, \mathcal{A}) = \min_{a\in A} d(v,a)$. We also denote $||S||$ the length of the largest vector of the set S. The closest vector problem, CVP, gets as input an n–dimensional lattice B and a target vector $v \in \mathbb{R}^n$. The output should be the point $b \in \mathcal{L}(B)$ closest to v. CVP is NP hard. Furthermore, it is NP hard to approximate the distance to the closest vector in the lattice to within small factors, and it is easy to approximate it to within $2^{\epsilon n}$ factor, for every $\epsilon > 0$. See [21] for a discussion. [21] proves that the following version of CVP is in SZK.

The problem gapCVP.

- Input: An n–dimensional lattice B, a vector $v \in \mathbb{R}^n$ and designated distance d. We set $g = g(n) = \sqrt{\frac{n}{c\log n}}$, for some $c > 0$.
- Promise: Either $d(v, \mathcal{L}(B)) \leq d$ or $d(v, \mathcal{L}(B)) \geq g \cdot d$.

- Output: "Yes" for first case, "No" for second.

The reduction. Given an input (B,v,d) for the gapCVP problem, we describe a classical circuit C_0, an input to the Qsampling problem. We let H_t denote the sphere of all points in $\mathbb{R}^n$ of distance at most t from the origin. The circuit C_0 gets as input a random string, and outputs the vector $r+\eta$, where r is a uniformly random point in $\mathcal{L}(B) \cap H_{2^n \cdot ||B \cup \{v\}||}$ (The lattice points inside a very large sphere), and η is a uniformly random point $\eta \in H_{\frac{d}{2} \cdot d}$. [21] explain how to sample such points with almost the right distribution, i.e. they give a description of an efficient such C_0. We remark that actually, the points cannot be randomly chosen from the real (continuous) vector space, due to precision issues, but [21] shows that taking a fine enough discrete approximation results in an exponentially small error.

Correctness. We need to show that efficient Qsampling from C_0 implies a BQP algorithm for the above gapCVP problem. In fact, we will get an RQP (one sided error) algorithm. The algorithm is defined as follows. First, define another circuit, C_1, to be like C_0 except that the outputs are shifted by the vector v and become $r+\eta+v$. If we can Qsample from C_0, we can also Qsample from C_1 by applying a shift by v at the end. To solve the gap problem the algorithm creates the state $\frac{1}{\sqrt{2}}[\ |0\rangle |C_0\rangle + |1\rangle |C_1\rangle\ ]$ and proceeds as in Claim 1. We show that this indeed gives the correct answer in the two possible cases. If v is far away from the lattice $\mathcal{L}(B)$, then the calculation at [21] shows that the states $|C_0\rangle$ and $|C_1\rangle$ have no overlap and so $\langle C_0|C_1\rangle = 0$. On the other hand, suppose v is close to the lattice, $d(v,\mathcal{L}(B)) \leq d$. Notice that in this case, the spheres of radius $gd/2$ around a lattice point r, and around $r+v$ have a large overlap. Indeed, the argument of [21] shows that the variation distance between the two distributions, $\|D_{C_0} - D_{C_1}\| \leq 1 - n^{-2c}$. By fact 1 we have $\langle C_0|C_1\rangle = F(D_{C_0}, D_{C_1}) \geq n^{-2c}$. Iterating the above $poly(n)$ times we get an RQP algorithm.

3. ADIABATIC STATE GENERATION (ASG)

3.1 Physics Background

3.1.1 Norms

The spectral norm of a linear transformation T, induced by the l_2 norm, is $||T|| = \max_{\psi \neq 0} \frac{|T\psi|}{|\psi|}$. For T Hermitian $||T||$ equals the largest absolute value of its eigenvalues. For U unitary, $||U|| = 1$. Also, $||AB|| \leq ||A|| \cdot ||B||$, and $||A|| \geq \max_{k,\ell} |A_{k,\ell}| \stackrel{\text{def}}{=} A_\infty$. We say a linear transformation T_2 α–approximates a linear transformation T_1 if $||T_1 - T_2|| \leq \alpha$.

3.1.2 Trotter Formula

Say $H = \sum H_m$ with each H_m Hermitian. Trotter's formula states that one can approximate e^{-itH} by slowly interleaving executions of e^{-itH_m} for different m's. We use a variant of it which can be proved using standard techniques from [36]. Define:

$$U_\delta = \ e^{-\delta i H_1} \cdot e^{-\delta i H_2} \cdot \ldots \cdot e^{-\delta i H_M} \tag{3}$$

Lemma 3. Let H_i be Hermitian, $H = \sum_{m=1}^{M} H_m$. Assume $||H||, ||H_i|| \leq \Lambda$. Then, for every $t > 0$

$$||U_\delta^{\lfloor \frac{t}{\delta} \rfloor} - e^{-itH}|| \quad \leq \quad O(M\Lambda \cdot \delta + M^2\Lambda^2 t \cdot \delta) \tag{4}$$

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

3.1.3 Time Dependent Schrodinger Equation

The evolution of the state from time 0 to time T can be described by integrating Schrodinger's equation (1) over time. If H is constant and independent of time, one gets

$$|\psi(T)\rangle = U(T)|\psi(0)\rangle = e^{-iHT}|\psi(0)\rangle \tag{5}$$

Since H is Hermitian, e^{-iHT} is unitary. The result of the integration is unitary also for time dependent Hamiltonians, which gives the familiar unitary evolution from quantum circuits. The groundstate of a Hamiltonian H is the eigenstate with the smallest eigenvalue, and if it is unique we denote it by $\alpha(H)$. The spectral gap of a Hamiltonian H is the difference between the smallest and second to smallest eigenvalues, and we denote it by $\Delta(H)$.

3.1.4 The adiabatic Theorem

In the study of *adiabatic evolution* one is interested in the long time behavior (at large times T) of a quantum system initialized in the ground state of H at time 0 when the Hamiltonian of the system, $H(t)$ changes very slowly in time, namely *adiabatically*. To state the adiabatic theorem [7, 30, 35], it is convenient and traditional to work with a re-scaled time $s = \frac{t}{T}$ where T is the total time. The Schrodinger's equation restated in terms of the re-scaled time s then reads

$$i\hbar \frac{d}{ds}|\psi(s)\rangle = T \cdot H(s)|\psi(s)\rangle \tag{6}$$

where $T = \frac{dt}{ds}$ can be referred to as the *delay schedule*, or the *total time*.

Theorem 4. **(The adiabatic theorem, adapted from [35, 19])**. Let $H(\cdot)$ be a function from $[0,1]$ to the vector space of Hamiltonians on n qubits. Assume $H(\cdot)$ is continuous, has a unique ground state, for every $s \in [0,1]$, and is differentiable in all but possibly finitely many points. Let $\epsilon > 0$ and assume that the following adiabatic condition holds for all points $s \in (0,1)$ in which the derivative is defined:

$$T\epsilon \geq \frac{\|\frac{d}{ds}H(s)\|}{(\Delta(H(s))^2} \tag{7}$$

Then, a quantum system that is initialized at time 0 with the ground state $\alpha(H(0))$, and evolves according to Equation (6) with $H(\cdot)$, ends up at re-scaled time 1 at a state $|\psi(1)\rangle$ that is within ϵ^c distance from $\alpha(H(1))$ for some constant $c > 0$.

We will refer to Equation (7) as the *adiabatic condition*. The proof of the adiabatic theorem is rather involved. For intuition, consider Schrodinger's equation for eigenstates of H; If the eigenvalue is λ, the eigenstate evolves by a multiplicative factor $e^{i\lambda t}$, which rotates in time faster the larger the absolute value of the eigenvalue λ is, and so the groundstate rotates the least. The fast rotations are essentially responsible to the cancellations of the contributions of the vectors with the higher eigenvalues, due to interference.

3.2 Adiabatic Quantum State Generation

In this section we define our paradigm for quantum *state generation*, generalizing the ideas of adiabatic quantum *computation* (and the adiabatic theorem). We define:

Definition 3. **(Simulatable Hamiltonians).** We say a Hamiltonian H on n qubits is *simulatable* if for every $t > 0$ and every accuracy $0 < \alpha < 1$ the unitary transformation $U(t) = e^{-iHt}$ can be approximated to within α accuracy by a quantum circuit of size $poly(n, t, 1/\alpha)$.

If H is simulatable, then by definition so is cH for any $0 \leq c \leq poly(n)$. It therefore follows by Trotter's formula (Lemma 3) that any convex combination of two simulatable, bounded norm Hamiltonians is simulatable. Also, If H is simulatable and U is a unitary matrix that can be efficiently applied by a quantum circuit, then $UHU^\dagger$ is also simulatable, because $e^{-itUHU^\dagger} = Ue^{-itH}U^\dagger$. The interested reader is referred to [36, 9] for a more complete set of rules for simulating Hamiltonians. We now describe an adiabatic path, which is an allowable path in the Hamiltonian space:

Definition 4. **(Adiabatic path).** A function H from $s \in [0, 1]$ to the vector space of Hamiltonians on n qubits, is an *adiabatic path* if $H(s)$ is always continuous, differentiable except for finitely many points, for every s $H(s)$ has a unique groundstate, and for every s $H(s)$ is simulatable given s.

Definition 5. **(Adiabatic Quantum State Generation [ASG]).** An adiabatic Quantum State Generator $(H_x(s), T, \epsilon)$ has for every $x \in \{0, 1\}^n$ an adiabatic path $H_x(s)$, such that for the given T, ϵ the adiabatic condition is satisfied for all $s \in [0, 1]$ where it is defined. We also require that the generator is explicit, i.e., that there exists a polynomial time quantum machine that

- On input $x \in \{0, 1\}^n$ outputs $\alpha(H_x(0))$, the groundstate of $H_x(0)$, and,
- On input $x \in \{0, 1\}^n$, $s \in [0, 1]$ and $\delta > 0$ outputs a circuit $C_x(s)$ approximating $e^{-i\delta H_x(s)}$.

We then say the generator adiabatically generates $\alpha(H_x(1))$.

Remark: We note that in previous papers on adiabatic computation, eg. in [14], a delay schedule $\tau(s)$ which is a function of s was used. We chose to work with one single delay parameter, T, instead, which might seem restrictive; However, working with a single parameter does not restrict the model since more complicated delay schedules can be encoded into the dependence on s.

3.3 Circuit simulation of ASG

An ASG can be simulated efficiently by a quantum circuit:

Claim 2. **(Circuit simulation of ASG).** Let $(H_x(s), T, \epsilon)$ be an ASG. Assume $T \leq poly(n)$. Then, there exists a quantum circuit that on input x generates the state $\alpha(H_x(1))$ to within ϵ accuracy, with size $poly(T, 1/\epsilon, n)$.

PROOF. **(Based on Adiabatic Theorem)** This proof is very similar to the proof given in [14] for the fact that adiabatic evolution can be simulated by quantum circuits efficiently. The circuit is built by discretizing time to sufficiently small intervals of length δ, and then applying the unitary matrices $e^{-iH(\delta j)\delta}$. Intuitively this should work, since the adiabatic theorem tells us that a physical system evolving for time T according to Schrodinger's equation with the given adiabatic path will end up in a state close to $\alpha(H_x(1))$, and the discretization introduces only a small error if δ is small enough. The formal error analysis can be done by exactly the same techniques that were used in [14]. □

We sketch another proof here which does not rely on the adiabatic theorem and can be derived from first principles:

PROOF. **(Based on the Zeno effect)** For the full proof see [2]. As before, we begin at the state $\alpha(H(0))$, and the circuit is built by discretizing time to sufficiently small intervals of length δ. At each time step j, $j = 1, \ldots, R$, instead of simulating the Hamiltonian we measure the state in a basis which includes the groundstate $\alpha(H(s_j))$. This can be done using Lemma 4 below. If R is sufficiently large, the adiabatic condition ensures that subsequent Hamiltonians are very close in the spectral norm. Furthermore, because T is polynomially bounded, the spectral gaps are non negligible. It can be shown that these two facts imply that subsequent groundstates are very close. Given that at time step j the state is the groundstate $\alpha(H(s_j))$, the next measurement results with very high probability in a projection on the new groundstate $\alpha(H(s_{j+1}))$. The Zeno effect guarantees that the error probability behaves like $1/R^2$, i.e. quadratically in R (and not linearly), and so the accumulated error after R steps is still small, which implies that the probability that the final state is the groundstate of $H(1)$ is very high, if R is taken to be large enough. □

3.4 The Sparse Hamiltonian Lemma

The main idea of the proof of Lemma 1 is to explicitly write H as a sum of polynomially many bounded norm Hamiltonians H_m which are all block diagonal (in a combinatorial sense) and such that the size of the blocks in each matrix is at most 2×2. We then show that each Hamiltonian H_m is simulatable and use Trotter's formula to simulate H.

3.4.1 The reduction to 2×2 block matrices.

Definition 6. (Combinatorial block.) Let A be a matrix with rows $R(A)$ and columns $C(A)$. We say $(R, C) \subseteq R(A) \times C(A)$ is a combinatorial block if $|R| = |C|$, for every $c \in C$, $r \notin R$, $A(c, r) = 0$, and for every $c \notin C$, $r \in R$, $A(c, r) = 0$.

A is block diagonal in the combinatorial sense iff there is some renaming of the nodes under which it becomes block diagonal in the usual sense. Equivalently, A is block diagonal in the combinatorial sense iff there is a decomposition of its rows into $R(A) = \bigcup_{b=1}^{B} R_b$, and of its columns $C(A) = \bigcup_{b=1}^{B} C_b$ such that for every b, (R_b, C_b) is a combinatorial block. We say A is 2×2 combinatorially block diagonal, if each combinatorial block (R_b, C_b) is at most 2×2, i.e., for every b either $|R_b| = |C_b| = 1$ or $|R_b| = |C_b| = 2$.

Claim 3. **(Decomposition lemma).** Let H be a row-sparse, row-computable Hamiltonian over n qubits, with at most D non-zero elements in each row. Then there is a way to decompose H into $H = \sum_{m=1}^{(D+1)^2 n^6} H_m$ where:

- Each H_m is a row-sparse, row-computable Hamiltonian over n qubits, and,
- Each H_m is 2×2 combinatorially block diagonal.

PROOF. (Of Claim 3) We color all the entries of H with $(D+1)^2 n^6$ colors. For $(i, j) \in [N] \times [N]$ and $i < j$ (i.e., (i, j) is an upper-diagonal entry) we define the coloring $col_H(i, j)$ to be the tuple $(k,\ i \bmod k,\ j \bmod k,\ rind_H(i, j),\ cind_H(i, j))$ where

- If $i = j$ we set $k = 1$, otherwise we let k be the first integer in the range $[2..n^2]$ such that $i \neq j (\mathrm{mod} k)$, and we know there must be such a k.
- If $H_{i,j} = 0$ we set $rind_H(i,j) = 0$, otherwise we let $rind_H(i,j)$ be the index of $H_{i,j}$ in the list of all non-zero elements in the i'th row of H. $cind_H(i,j)$ is similar, but with regard to the columns of H.

For lower-diagonal entries, $i > j$, we define $col_H(i,j) = col_H(j,i)$. Altogether, we use $(n^2)^3 \cdot (D+1)^2$ colors.

For a color m, we define $H_m[i,j] = H[i,j] \cdot \delta_{col_H(i,j),m}$, i.e., H_m is H on the entries colored by m and zero everywhere else. Clearly, $H = \sum_m H_m$ and each H_m is Hermitian. Also as H is row-sparse and row-computable, there is a simple $poly(n)$-time classical algorithm computing the coloring $col_H(i,j)$, and so each H_m is also row-computable. It is left to show that it is 2×2 combinatorially block-diagonal.

Indeed, fix a color m. Let NZ_m be the set of all upper-triangular, non-zero elements of H_m. Say the elements of NZ_m are $\{(i_1,j_1),\ldots,(i_B,j_B)\}$. For every element $(i_b,j_b) \in NZ_m$ we introduce a block. If $i_b = j_b$ then we set $R_b = C_b = \{i_b\}$ while if $i_b \neq j_b$ then we set $R_b = C_b = \{i_b, j_b\}$.

Say $i_b \neq j_b$ (the $i_b = j_b$ case is similar and simpler). As the color m contains the row-index and column-index of (i_b,j_b), it must be the case that (i_b,j_b) is the only element in NZ_m from that row (or column). Furthermore, as $i_b \bmod k \neq j_b \bmod k$, and both k, $i \bmod k$ and $j \bmod k$ are included in the color m, it must be the case that there are no elements in NZ_m that belong to the j_b row or i_b column (see Figure 1). It follows that (R_b, C_b) is a block. We therefore see that H_m is 2×2 combinatorially block-diagonal as desired. □

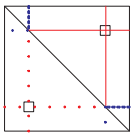

Figure 1: *In the upper diagonal side of the matrix H_m: the row and column of (i_b,j_b) are empty because the color m contains the row-index and column index of (i,j), and the j_b'th row and i_b'th column are empty because m contains k, $i \bmod k$, $j \bmod k$ and $i \bmod k \neq j \bmod k$. The lower diagonal side of H_m is just a reflection of the upper diagonal side. It follows that $\{i_b,j_b\}$ is a 2×2 combinatorial block.*

Claim 4. For every m, $||H_m|| \leq ||H||$.

PROOF. Fix an m. H_m is combinatorially block diagonal and therefore its norm $||H_m||$ is achieved as the norm of one of its blocks. Now, H_m blocks are either

- 1×1, and then the block is $(H_{i,i})$ for some i, and it has norm $|H_{i,i}|$, or,
- 2×2, and then the block is $\begin{pmatrix} 0 & H_{k,\ell} \\ H^*_{k,\ell} & 0 \end{pmatrix}$ for some k,ℓ, and has norm $|H_{k,\ell}|$.

It follows that $\max_m ||H_m|| \leq \max_{k,\ell} |H_{k,\ell}|$. On the other hand $||H|| \geq \max_{k,\ell} |H_{k,\ell}|$. The proof follows. □

3.4.2 2×2 block matrices are simulatable.

Claim 5. Every 2×2 combinatorially block diagonal, row-computable Hamiltonian A is simulatable to within arbitrary polynomial approximation.

PROOF. Let $t > 0$ and $\alpha > 0$ an accuracy parameter.

The simulating quantum circuit. A is 2×2 combinatorially block diagonal. Let $|k\rangle$ be a basis state, and let k belong to the 2×2 block $\{k,\ell\}$ in A. We note that A leaves the subspace spanned by $|k\rangle, |\ell\rangle$ invariant. Set $b_k = 2$ (for a 2×2 block) and $m_k = \min(k,\ell)$, $M_k = \max(k,\ell)$. We then set A_k to be the part of A relevant to this subspace $A_k = \begin{pmatrix} A_{m_k,m_k} & A_{m_k,M_k} \\ A_{M_k,m_k} & A_{M_k,M_k} \end{pmatrix}$ and $U_k = e^{-itA_k}$. If $|k\rangle$ belongs to a 1×1 block we similarly define $b_k = 1$, $m_k = M_k = k$, $A_k = (A_{k,k})$ and $U_k = (e^{-itA_k})$. Our approximation circuit simulates the application of U_k on $|k\rangle$. We need two transformations. We define

$$T_1 : |k,0\rangle \to \left|b_k, m_k, M_k, \widetilde{A_k}, \widetilde{U_k}, k\right\rangle$$

where $\widetilde{A_k}$ is our approximation to the entries of A_k and $\widetilde{U_k}$ is our approximation to $e^{-it\widetilde{A_k}}$, and where both matrices are expressed by their four (or one) entries. We use $\alpha^{O(1)}$ accuracy.

Having $\widetilde{U_k}, m_k, M_k, k$ written down, we can simulate the action of $\widetilde{U_k}$. We can therefore have an efficient unitary transformation T_2:

$$T_2 : \left|\widetilde{U_k}, m_k, M_k\right\rangle |v\rangle = \left|\widetilde{U_k}, m_k, M_k\right\rangle \left|\widetilde{U_k} v\right\rangle$$

for $|v\rangle \in Span\{m_k, M_k\}$. Our algorithm is applying T_1 followed by T_2 and then T_1^{-1} for cleanup.

Correctness. Let us denote $\mathrm{Diff} = e^{-itA} - T_1^{-1} T_2 T_1$. Our goal is to show that $||\mathrm{Diff}|| \leq \alpha$. We notice that Diff is also 2×2 block diagonal, and therefore its norm can be achieved by a vector ψ belonging to one of its dimension one or two subspaces, say to $Span\{m_k, M_k\}$. Let $U_k |\psi\rangle = \alpha |m_k\rangle + \beta |M_k\rangle$ and $\widetilde{U_k} |\psi\rangle = \widetilde{\alpha} |m_k\rangle + \widetilde{\beta} |M_k\rangle$. We see that $|\psi_0\rangle$ is mapped to

$$\begin{aligned} &\xrightarrow{T_1} \quad \left|b_k, m_k, M_k, \widetilde{A_k}, \widetilde{U_k}, \psi\right\rangle \xrightarrow{T_2} \left|b_k, m_k, M_k, \widetilde{A_k}, \widetilde{U_k}, \widetilde{U_k}\psi\right\rangle \\ &= \quad \widetilde{\alpha} \left|b_k, m_k, M_k, \widetilde{A_k}, \widetilde{U_k}, m_k\right\rangle + \widetilde{\beta} \left|b_k, m_k, M_k, \widetilde{A_k}, \widetilde{U_k}, M_k\right\rangle \\ &\xrightarrow{T_1^{-1}} \quad \widetilde{\alpha} |m_k, 0\rangle + \widetilde{\beta} |M_k, 0\rangle \end{aligned}$$

where the first equation holds since it holds for $|m_k\rangle$, $|M_k\rangle$ and by linearity it holds for the whole subspace spanned by them. We conclude that $|\mathrm{Diff}\,|\psi\rangle\,| = |(U_k - \widetilde{U_k})\,|\psi\rangle\,|$ and so $||\mathrm{Diff}|| \leq \max_k ||U_k - \widetilde{U_k}||$. However, by our construction, $||\widetilde{A_k} - A_k||_\infty \leq \alpha^{O(1)}$ and so $||\widetilde{U_k} - U_k|| \leq \alpha$ as desired. □

We proved the claim for matrices with 2×2 combinatorial blocks. We remark that the same approach works for matrices with $m \times m$ combinatorial blocks, for $m = poly(n)$.

3.4.3 Proving the Sparse Hamiltonian Lemma

PROOF. (Of Lemma 1.) Let H be row-sparse with $D \leq poly(n)$ non-zero elements in each row, and $||H|| = \Lambda \leq poly(n)$. Let $t > 0$. Our goal is to efficiently simulate e^{-itH} to within α accuracy.

We express $H = \sum_{m=1}^{M} H_m$ as in Claim 3, $M \leq (D+1)^2 n^6 \leq poly(n)$. We choose δ such that $O(M\Lambda\delta + M^2\Lambda^2 t\delta) \leq \frac{\alpha}{2}$. Note that $\frac{1}{\delta} \leq poly(t,n)$ for some large enough polynomial. We then compute $U_\delta^{\lfloor \frac{t}{\delta} \rfloor}$ to within $\frac{\alpha}{2}$ accuracy, using as in Lemma 3, our approximations to $e^{-i\delta H_m}$ (where

each $e^{-i\delta H_m}$ is computed to within $O(\frac{\alpha}{Mt/\delta})$ accuracy.) By Lemma 3 our computation is $\frac{\alpha}{2}$ close to e^{-itH}, as desired (using the fact that for every m, $||H_m|| \leq ||H|| = \Lambda \leq poly(n)$ by Claim 4). The size of the computation is $\frac{t}{\delta} \cdot M \cdot poly(\delta, M, n, \alpha) = poly(n, t, \alpha)$ as required. □

3.5 The Jagged Adiabatic Path Lemma

PROOF. (of Lemma 2) We consider the sequence of Hamiltonians $\left\{\Pi_{H_j}\right\}$ where Π_H is a projection on the space orthogonal to the groundstate of H_j, and we connect two neighboring projections by a line. We prove in Lemma 4, using Kitaev's phase estimation algorithm, that the fact that H_j is simulatable implies that so is Π_{H_j}. Also, as projections, Π_{H_j} have bounded norms, $||\Pi_{H_j}|| \leq 1$. It follows then, by the results mentioned in Section 3.1, that all the Hamiltonians on the path connecting these projections are simulatable, as convex combinations of simulatable Hamiltonians.

We now have to show the Hamiltonians on the path have non negligible spectral gap. By definition Π_{H_j} has a spectral gap equal to 1. It remains to show, however, that the Hamiltonians on the line connecting Π_{H_j} and $\Pi_{H_{j+1}}$ have large spectral gaps, which we prove in the simple Lemma 5.

We can now apply the adiabatic theorem and get Lemma 2. Indeed, a linear time parameterization suffices to show that this algorithm satisfies the adiabatic condition. □

We note that Lemma 4 is crucial for the above proof, since if we use the Hamiltonians directly and not their projections, the path connecting them might have 0 spectral gap. We now turn to the proofs of Lemmas 4 and 5.

Lemma 4. **(Hamiltonian-to-projection lemma)**. Let H be a Hamiltonian on n qubits such that e^{-iH} can be approximated to within arbitrary polynomial accuracy by a polynomial quantum circuit, and let $\|H\| \leq m = poly(n)$. Let $\Delta(H)$ be non negligible, and larger than $1/n^c$, and further assume that the groundvalue of H is 0. Then the projection Π_H, is simulatable.

PROOF. First apply Kitaev's phase estimation algorithm [31, 36]. As the spectral gap is non-negligible we can decide with exponentially good confidence whether an eigenstate has the lowest eigenvalue or a larger eigenvalue. We can therefore write down one bit of information on an extra qubit: whether an input eigenstate of H is the ground state or orthogonal to it.

Second, apply a phase shift of value e^{-it} to this extra qubit, conditioned that it is in the state $|1\rangle$ (if it is $|0\rangle$ we do nothing). This conditional phase shift corresponds to applying for time t a Hamiltonian with two eigenspaces, the ground state and the subspace orthogonal to it, with respective eigenvalues 0 and 1, which is exactly the desired projection.

Finally, to erase the extra qubit written down, we reverse the first step and uncalculate the information written on that qubit using Kitaev's phase estimation algorithm again. □

For a vector $|\alpha\rangle$, the Hamiltonian $H_\alpha = I - |\alpha\rangle\langle\alpha|$ is the projection onto the subspace orthogonal to α. We prove:

Lemma 5. **The Two Dimensional Adiabatic Lemma** Let $|\alpha\rangle, |\beta\rangle$ be two vectors in some subspace, $H_\alpha = I - |\alpha\rangle\langle\alpha|$ and $H_\beta = I - |\beta\rangle\langle\beta|$. For any convex combination $H_\eta = (1-\eta)(I - |\alpha\rangle\langle\alpha|) + \eta(I - |\beta\rangle\langle\beta|), \quad \eta \in [0,1]$, of the two Hamiltonians H_α, H_β, $\Delta(H_\eta) \geq |\langle\alpha|\beta\rangle|$.

PROOF. Observe that the problem is two dimensional: write $|\beta\rangle = a|\alpha\rangle + b|\alpha^\perp\rangle$, write the matrix H_η in an orthonormal basis which contains $|\alpha\rangle$ and $|\alpha^\perp\rangle$, and diagonalize to find the eigenvalues. □

3.6 Equivalence of Standard and Adiabatic State Generation

The proof of theorem 2 consists of two directions. We already saw one direction in claim 2, and now we give the other direction.

Claim 6. let $|\phi\rangle$ be the final state of a quantum circuit C with M gates, then there is an ASG which outputs this state, of complexity $poly(n, M)$.

PROOF. W.l.o.g. the circuit starts in the state $|0\rangle$. We first modify the circuit so that the state does not change too much between subsequent time steps. The reason we need this will become apparent shortly. To make this modification, let us assume for concreteness that the quantum circuit C uses only Hadamard gates, Toffoli gates and Not gates. This set of gates was shown to be universal by Shi [42, 3]. (Our proof works with any universal set with obvious modifications.) We replace each gate g in the circuit by two $\sqrt{g}$ gates. For $\sqrt{g}$ we can choose any of the possible square roots arbitrarily, but for concreteness we notice that Hadamard, Not and Toffoli gates have ± 1 eigenvalues, and we choose $\sqrt{1} = 1$ and $\sqrt{-1} = i$. We call the modified circuit C'. Obviously C and C' compute the same function.

The path. We let M' be the number of gates in C'. For integer $0 \leq j \leq M'$, we set

$$H_x(\frac{j}{M'}) \quad = \quad I - |\alpha_x(j)\rangle\langle\alpha_x(j)|$$

where $|\alpha_x(j)\rangle$ is the state of the system after applying the first j gates of C' on the input x. For $s = \frac{j+\eta}{M'}$, $\eta \in [0,1)$, define $H_x(s) = (1-\eta)H_x(j) + \eta H_x(j+1)$.

The spectral gaps are large. Clearly all the Hamiltonians $H_x(j)$ for integer $0 \leq j \leq M'$, have non-negligible spectral gaps, since they are projections. We claim that for any state β and any gate $\sqrt{g}$, $|\langle\beta|\sqrt{g}|\beta\rangle| \geq \frac{1}{\sqrt{2}}$. Indeed, represent β as $a_1 v_1 + a_2 v_2$ where v_1 belongs to the 1-eigenspace of $\sqrt{g}$ and v_2 belongs to the i-eigenspace of $\sqrt{g}$. We see that $|\langle\beta|\sqrt{g}|\beta\rangle| = ||a_1|^2 + i|a_2|^2|$. As $|a_1|^2 + |a_2|^2 = 1$, a little algebra shows that this quantity is at least $\frac{1}{\sqrt{2}}$. In particular, setting $\beta = \alpha_x(j)$ we see that $|\langle\alpha_x(j)|\alpha_x(j+1)\rangle| \geq \frac{1}{\sqrt{2}}$. It therefore follows by Lemma 5 that all the Hamiltonians on the line between $H_x(j)$ and $H_x(j+1)$ have spectral gaps larger than $\frac{1}{\sqrt{2}}$.

The Hamiltonians are simulatable. Given a state $|y\rangle$ we can first apply the inverse of the first j gates of C', then if we are in state $|x, 0\rangle$ apply a phase shift $e^{-i\delta}$, finally apply the first j gates of C'. This implements $e^{-i\delta H_x(j)}$.

The Adiabatic Condition is Satisfied. We have $\frac{dH}{ds}(s_0) = lim_{\zeta \to 0} \frac{H(s_0+\zeta) - H(s_0)}{\zeta}$. We ignore the finitely many points

$s = \frac{j}{M'}$ where j is an *integer* in $[0, M']$. For all other points s, when ζ goes to 0 both $H(s_0 + \zeta)$ and $H(s_0)$ belong to the same interval. Say they belong to the j'th interval, $s_0 = \frac{j+\eta}{M'}$, $0 < \eta < 1$. Then, $H(s_0) = (1 - \eta)H_x(j) + \eta H_x(j+1)$ and $H(s_0 + \zeta) = H(\frac{j+\eta+M'\zeta}{M'}) = (1 - \eta - M'\zeta)H_x(j) + (\eta + M'\zeta)H_x(j+1)$. It follows that $H(s_0+\zeta)-H(s_0) = M'\zeta H_x(j+1)-M'\zeta H_x(j)$ and $\frac{dH}{ds}(s_0) = M' \cdot [H_x(j+1)-H_x(j)]$. We conclude that $||\frac{dH}{ds}|| \leq 2M'$ and to satisfy Equation (7) we just need to pick $T = O(\frac{M'}{\epsilon})$. □

4. QSAMPLING AND MARKOV CHAINS

4.1 Markov chain Background

We consider Markov chains with states indexed by n bit strings. If M is an ergodic (i.e. connected, aperiodic) Markov chain, characterized with the matrix M operating on probability distributions over the state space Ω, and p is an initial probability distribution, then $\lim_{t\longmapsto\infty} pM^t = \pi$. The limiting distribution π is independent of p.

A Markov chain M has eigenvalues between -1 and 1. π corresponds to eigenvalue 1. It is convenient to assume that all eigenvalues are non negative (by adding self loops which slow the chain by a factor of 2.) A Markov chain is *rapidly mixing* if starting from any initial distribution, the distribution after $poly(n)$ time steps is within ϵ variation distance from π. [5] shows that a Markov chain is rapidly mixing iff $1 - \lambda_2 \geq 1/poly(n)$, where λ_2 is the second largest eigenvalue. $1 - \lambda_2$ is called the second eigenvalue gap.

A Markov chain is *reversible* if $M[i,j] \cdot \pi_i = M[j,i] \cdot \pi_j$. A symmetric Markov chain M is reversible. Also, for an ergodic, reversible Markov chain M $\pi_i > 0$ for all i.

In approximate counting algorithms one is interested in sequences of rapidly mixing Markov chains, where subsequent Markov chains have close limiting distributions. For more background regarding Markov chains, see [34]. For more background regarding approximate counting see [29].

4.2 Markov chains and Hamiltonians

For a reversible M we define $H_M = I - Diag(\sqrt{\pi_i}) \cdot M \cdot Diag(\frac{1}{\sqrt{\pi_j}})$. A direct calculation shows that H_M is symmetric iff M is reversible. We call H_M the *Hamiltonian corresponding to* M. The properties of H_M and M are very much related, by the following claim (The proof is straight forward and omitted here):

Claim 7. If M is a reversible Markov chain, we have:

- H_M is a Hamiltonian with $||H_M|| \leq 1$.
- The spectral gap of H_M equals the second eigenvalue gap of M.
- If π is the limiting distribution of M, then the ground state of H_M is $\alpha(H_M) = |\pi\rangle \stackrel{\text{def}}{=} \sum_i \sqrt{\pi(i)}\,|i\rangle$.

This claim gives a direct connection between Hamiltonians, spectral gaps and groundstates on one hand, and rapidly mixing reversible Markov chains and limiting distribution on the other hand.

4.3 Simulating H_M

Not every Hamiltonian corresponding to a reversible Markov chain can be easily simulated. However, we will shortly see that the Hamiltonian corresponding to a symmetric Markov chain is simulatable. For general reversible Markov chains we need some more restrictions. We define:

Definition 7. A reversible Markov chain is *strongly samplable* if it is row computable, and, Given $i, j \in \Omega$, there is an efficient way to approximate $\frac{\pi_i}{\pi_j}$.

Row computability holds in most interesting cases but the second requirement is quite restrictive. We note that if we could relax it, the techniques in this section could have been used to give a quantum algorithm for graph isomorphism. Still, we note that the second requirement holds in many interesting cases such as all Metropolis algorithms (see [24]). It also trivially holds for symmetric M, where the limiting distribution is uniform. We can now prove corollary 1:

PROOF. (of corollary 1) Since $H_M[i,j] = \sqrt{\frac{\pi_i}{\pi_j}}M[i,j]$ we see that if M is strongly samplable then H_M is row-computable. H_M has bounded norm and so the sparse Hamiltonian lemma applies. □

4.4 From Markov chains to QSampling

We are interested in strongly samplable rapidly mixing Markov chains, so that the Hamiltonians are simulatable and have non negligible spectral gaps by claim 7. To adapt this setting to adiabatic algorithms, and to the setting of the jagged adiabatic path lemma in particular, we now consider sequences of Markov chains, and define:

Definition 8. **(Slowly Varying Markov Chains)**. Let $\{M_t\}_{t=1}^T$ be a sequence of Markov chains on Ω, $|\Omega| = N = 2^n$. Let π_t be the limiting distribution of M_t. We say the sequence is *slowly varying* if for all $c > 0$, for all large enough n, for all $1 \leq t \leq T$ $\|\pi_t - \pi_{t+1}\| \leq 1 - 1/n^c$.

We prove that we can use a sequence of slowly varying rapidly mixing Markov chains to Qsample from the limiting distribution of the final Markov chain. This is theorem 3, which we can now state precisely.

Theorem 3: *Let $\{M_t\}_{t=1}^T$ be a slowly varying sequence of strongly samplable Markov chains which are all rapidly mixing, and let π_t be their corresponding limiting distributions. Then if there is an efficient Qsampler for $|\pi_0\rangle$, then there is an efficient Qsampler for $|\pi_T\rangle$.*

PROOF. We already saw the Hamiltonians H_{M_t} are simulatable and have bounded norm. Also, as the Markov chains in the sequence are rapidly mixing, they have large spectral gaps, and therefore so do the Hamiltonians H_{M_t}. To complete the proof we show that the inner product between the groundstates of subsequent Hamiltonians is non negligible, and then the theorem follows from the jagged path lemma. Indeed, $\langle\alpha(H_{M_t})|\alpha(H_{M_{t+1}})\rangle = \langle\pi_t|\pi_{t+1}\rangle = \sum_i \sqrt{\pi_t(i)\pi_{t+1}(i)} \geq 1 - \|\pi_t - \pi_{t+1}\|$ and therefore is non-negligible. □

4.5 Qsampling from Perfect Matchings

We illustrate our technique with the example of how to Qsample from all perfect matchings in a given bipartite graph G. In this subsection we heavily rely on the work of Sinclair, Jerrum and Vigoda [28] who recently showed how

to efficiently approximate a permanent of any matrix with non negative entries, using a sequence of Markov chains on the set of Matchings of a bipartite graph. This work is far too involved to explain here fully, and we refer the reader to [28] for more details. In a nutshell, the idea in [28] is to apply a Metropolis random walk on the set of perfect and near perfect matchings (i.e. perfect matchings minus one edge) of the complete bipartite graph. Weights are assigned to the edges such that edges that do not participate in the input graph G are slowly decreasing until the probability they appear in the final distribution practically vanishes. The weights of the edges are updated using data that is collected from running the Markov chain with the previous set of weights, in an adaptive way. The final Markov chain with the final parameters converges to a probability distribution which is essentially concentrated on the perfect and near perfect matchings of the input graph, where the probability of the perfect matchings is $1/n$ times that of the near perfect matching. Hence, if we can Qsample from the final limiting distribution, we can project on the perfect matchings with polynomial success probability

It remains to check that we can apply theorem 3. It is easy to check that the Markov chains being used in [28] are all strongly samplable, since they are Metropolis chains. Moreover, the sequence of Markov chains is slowly varying. It remains to see that we can quantum sample from the limiting distribution of the initial chain that is used in [28]. This limiting distribution is a distribution over all perfect and near perfect matchings in the complete bipartite graph, where the weight of each near perfect matching is n times bigger than that of a perfect matching, where n is the number of nodes of G. It is a simple exercise in quantum computation to Qsample from this distribution efficiently.

5. ACKNOWLEDGMENTS

We wish to thank Umesh Vazirani, Wim van Dam, Zeph Landau, Oded Regev, Dave Bacon, Manny Knill, Eddie Farhi, Ashwin Nayak and John Watrous for many inspiring discussions. In particular we thank Dave Bacon for an illuminating discussion which led to the proof of claim 6.

6. REFERENCES

[1] S. Aaronson, Quantum lower bound for the collision problem. STOC 2002, pp. 635-642

[2] D. Aharonov and A. Ta-Shma. quant-ph/0210077. A longer version of this paper.

[3] D. Aharonov, A simple proof that Toffoli and Hadamard are quantum universal, quant-ph 0301040

[4] N. Alon, Eigenvalues and Expanders. Combinatorica 6(1986), pp. 83-96.

[5] N. Alon and J. Spencer, The Probabilistic Method. 1991.

[6] D. Applegate and R. Kannan, Sampling and integration of near log-concave functions, STOC 1991, pp. 156–163

[7] M. Born, V. Fock and B. des Adiabatensatzes. Z. Phys. 51, pp. 165-169, 1928

[8] R. Bubley and M. Dyer, Faster random generation of linear extensions, SODA 1998, pp. 350-354.

[9] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann and D. A. Spielman, Exponential algorithmic speedup by quantum walk, quant-ph/0209131

[10] A. M. Childs, E. Deotto, E. Farhi, J. Goldstone, S. Gutmann, A. J. Landahl, Quantum search by measurement, Phys. Rev. A 66, 032314 (2002)

[11] A. M. Childs, E. Farhi, J. Goldstone and S. Gutmann, Finding cliques by quantum adiabatic evolution, quant-ph/0012104.

[12] A. M. Childs, E. Farhi and S. Gutmann, An example of the difference between quantum and classical random walks, quant-ph/0103020. Also, E. Farhi and S. Gutmann, Quantum computation and decision Trees, quant-ph/9706062

[13] W. van Dam and S. Hallgren, Efficient quantum algorithms for Shifted Quadratic Character Problems, quant-ph/0011067

[14] W. van Dam, M. Mosca and U. V. Vazirani, How Powerful is Adiabatic Quantum Computation? FOCS 2001, pp 279-287

[15] W. van Dam and U. Vazirani, More on the power of adiabatic computation, unpublished, 2001

[16] E. Farhi, J. Goldstone and S. Gutmann, A numerical study of the performance of a quantum adiabatic evolution algorithm for satisfiability, quant-ph/0007071.

[17] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren and D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem, Science **292**, 472 (2001), quant-ph/0104129.

[18] E. Farhi, J. Goldstone and S. Gutmann, Quantum Adiabatic Evolution Algorithms with Different Paths, quant-ph/0208135

[19] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser, Quantum Computation by Adiabatic Evolution, quant-ph/0001106

[20] O. Goldreich and E. Kushilevitz, A Perfect Zero-Knowledge Proof System for a Problem Equivalent to the Discrete Logarithm, Journal of Cryptology, 6(2), pp. 97–116, 1993

[21] O. Goldreich and S. Goldwasser, On the limits of non-approximability of lattice problems, STOC 1998, pp. 1-9

[22] O. Goldreich, A. Sahai and S. Vadhan, Honest-Verifier Statistical Zero-Knowledge Equals General Statistical Zero-Knowledge, STOC 1998 pp. 399–408,

[23] S. Goldwasser, S. Micali and C. Rackoff, The Knowledge Complexity of Interactive Proof Systems, SIAM J Comput, 18 (1), pp. 186–208, 1989

[24] M. Grotschel and L. Lovasz, Combinatorial Optimization: A Survey, Handbook of Combinatorics, North-Holland, 1993

[25] L. Grover, Quantum Mechanics helps in searching for a needle in a haystack, Phys. Rev. Letters, July 14, 1997

[26] L. Grover and T. Rudolph, Creating superpositions that correspond to efficiently integrable probability distributions, quant-ph/0208112

[27] S. Hallgren, Polynomial-Time Quantum Algorithms for Pell's Equation and the Principal Ideal Problem, STOC 2002

[28] M. Jerrum and A. Sinclair, E. Vigoda A Polynomial-Time Approximation Algorithm for the permanent of a matrix with non-negative entries, STOC 2000

[29] M. Jerrum and A. Sinclair, The Markov chain Monte Carlo method: an approach to approximate counting and integration. in Approximation Algorithms for NP-hard Problems, D.S.Hochbaum ed., 1996.

[30] T. Kato, On the adiabatic theorem of Quantum Mechanics, J. Phys. Soc. Jap. **5**, pp. 435-439 (1951)

[31] A. Yu. Kitaev, Quantum measurements and the Abelian Stabilizer Problem, quant-ph/9511026

[32] J. Kobler, U. Schoning and J. Turan, The Graph Isomorphism Problem. Birkjauser, 1993.

[33] Landau and Lifshitz, *Quantum Mechanics* (Second edition of English Translation), Pergamon press, 1965

[34] L. Lovasz: Random Walks on Graphs: A Survey. Combinatorics, Paul Erdos is Eighty, Vol. 2, ed. D. Miklos, V. T. Sos, T. Szonyi, 1996, pp. 353–398.

[35] Messiah, *Quantum Mechanics*, John Willey & Sons (1958)

[36] M. A. Nielsen and I. Chuang, Quantum Computation and Information 2000

[37] See A. Peres, Quantum Theory: Concepts and methods, 1995

[38] J. Roland and N. Cerf, Quantum Search by Local Adiabatic Evolution Phys. Rev. A 65, 042308 (2002)

[39] A. Sahai and S. P. Vadhan, A Complete Promise Problem for Statistical Zero-Knowledge, FOCS 1997 pp. 448–457

[40] P. W. Shor: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. 26(5) 1997, pp. 1484-1509

[41] W. L. Spitzer and S. Starr, Improved bounds on the spectral gap above frustration free ground states of quantum spin chains, math-ph/0212029

[42] Y. Shi, Both Toffoli and Controlled-NOT need little help to do universal quantum computation, quant-ph/0205115

[43] S. Vadhan, A Study of Statistical Zero Knowledge Proofs, PhD Thesis, M.I.T., 1999

[44] J. Watrous: Quantum algorithms for solvable groups, STOC 2001, pp. 60-67