SIAM J. Comput.
Vol. 37, No. 1, pp. 47–82

© 2007 Society for Industrial and Applied Mathematics

ADIABATIC QUANTUM STATE GENERATION*

DORIT AHARONOV[†] AND AMNON TA-SHMA[‡]

Abstract. The design of new quantum algorithms has proven to be an extremely difficult task. This paper considers a different approach to this task by studying the problem of *quantum state generation*. We motivate this problem by showing that the entire class of statistical zero knowledge, which contains natural candidates for efficient quantum algorithms such as graph isomorphism and lattice problems, can be reduced to the problem of quantum state generation. To study quantum state generation, we define a paradigm which we call *adiabatic state generation* (ASG) and which is based on adiabatic quantum computation. The ASG paradigm is not meant to replace the standard quantum circuit model or to improve on it in terms of computational complexity. Rather, our goal is to provide a natural theoretical framework, in which quantum state generation algorithms could be designed. The new paradigm seems interesting due to its intriguing links to a variety of different areas: the analysis of spectral gaps and ground-states of Hamiltonians in physics, rapidly mixing Markov chains, adiabatic computation, and approximate counting. To initiate the study of ASG, we prove several general lemmas that can serve as tools when using this paradigm. We demonstrate the application of the paradigm by using it to turn a variety of (classical) approximate counting algorithms into efficient quantum state generators of nontrivial quantum states, including, for example, the uniform superposition over all perfect matchings in a bipartite graph.

Key words. quantum computation, quantum algorithm, adiabatic theorem, Hamiltonians, Markov chains, quantum sampling, state generation, statistical zero knowledge, Zeno effect, spectral gap

AMS subject classification. 81P68

DOI. 10.1137/060648829

1. Introduction.

Quantum computation carries the hope of solving classically intractable tasks in quantum polynomial time. The most notable success so far is Shor's quantum algorithm for factoring integers and for finding the discrete log [45]. Following Shor's algorithm several other algorithms were discovered, such as Hallgren's algorithm for solving Pell's equation [31], Watrous's algorithms for the group black box model [48], and the Legendre symbol algorithm by van Dam and Hallgren [17]. Except for [17] all of these algorithms fall into the framework of the hidden subgroup problem and in fact use exactly the same quantum circuitry; the exception, [17], is a different algorithm but also heavily uses Fourier transforms and exploits the special algebraic structure of the problem. Recently, a beautiful new algorithm by Childs et al. [13] was found which gives an exponential speed-up over classical algorithms using an entirely different approach, namely quantum random walks. The algorithm, however, works in the black box model and solves a fairly contrived problem.

In order to develop new quantum algorithms, it is crucial that we have a larger

[*] Received by the editors January 2, 2006; accepted for publication (in revised form) October 6, 2006; published electronically March 30, 2007. A preliminary version of this paper appeared in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, San Diego, CA, 2003, pp. 20–29.

http://www.siam.org/journals/sicomp/37-1/64882.html

[†] Department of Computer Science and Engineering, Hebrew University, 91904 Jerusalem, Israel (doria@cs.huji.ac.il). This author's research was supported by an Alon fellowship (The Council of High Education in Israel) under grant 033-7233, and by the Israel Science Foundation under grants 032-9738 and 039-7549.

[‡] Department of Computer Science, Tel Aviv University, 69978 Tel-Aviv, Israel (amnon@post.tau.ac.il). This author's research was supported by the Binational Science Foundation, by the EU Integrated Project QAP, and by the Israel Science Foundation.

variety of quantum algorithmic techniques and approaches. In this paper we attempt to make a step in that direction by studying the problem of quantum algorithm design from a different point of view, that of "quantum state generation."

It has been folklore knowledge for almost a decade already that the problem of graph isomorphism, which is considered hard classically [37], has an efficient quantum algorithm as long as a certain state, namely the superposition of all graphs isomorphic to a given graph,

$$|\alpha_G\rangle = \sum_{\sigma \in S_n} |\sigma(G)\rangle, \tag{1.1}$$

can be generated efficiently by a quantum Turing machine (here and in the rest of the paper we ignore normalizing constants for the sake of brevity). The reason that generating $|\alpha_G\rangle$ suffices is very simple: for two isomorphic graphs G_1 and G_2, the states $|\alpha_{G_1}\rangle$ and $|\alpha_{G_2}\rangle$ are identical, whereas for two nonisomorphic graphs they are orthogonal. Using a simple quantum circuit known as the SWAP test (see section 3.2), one can approximate the inner product between two given states and thus can distinguish between the two cases of orthogonal and parallel $|\alpha_G\rangle$'s.

One is tempted to assume that such a state, $|\alpha_G\rangle$, 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 easily be generated by this reasoning. However, $|\beta_G\rangle$ is inadequate for our needs, as $|\beta_{G_1}\rangle$ and $|\beta_{G_2}\rangle$ are always orthogonal. It is a curious (and disturbing) fact of quantum mechanics that though $|\beta_G\rangle$ is an easy state to generate, 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 are interested in a restricted version of quantum state generation, that of generating states corresponding to *efficiently samplable* classical probability distributions. To be specific, let C be a classical circuit with n inputs and m outputs. We define the probability distribution D_C to be the distribution over the outputs of the classical circuit C when its inputs are uniformly distributed, i.e., $D_C(z) = \Pr_{x \in \{0,1\}^n}[C(x) = z]$. We denote

$$|C\rangle \overset{\text{def}}{=} \sum_{z \in \{0,1\}^m} \sqrt{D_C(z)}\, |z\rangle$$

and define the quantum sampling (QS) problem.

DEFINITION 1.1 (quantum sampling (QS_δ)).

Input: *A description of a classical circuit C and a constant $\delta \geq 0$.*

Output: *A description of a quantum circuit Q of size poly($|C|$), with a marked set of output qubits, such that on input $|0\rangle$ the final state ρ of the output qubits of the circuit Q is close to $|\phi\rangle = |C\rangle$, namely,*

$$\| \rho - |\phi\rangle \langle\phi| \|_{tr} \leq \delta.$$

The norm above is the trace norm (see section 2). We say Q quantum samples (or Qsamples) the output distribution of the circuit C. If δ is not specified, we take δ to be some fixed small constant, say 10^{-5}.

The problem of generating the graph isomorphism state from (1.1) is an instance of QS, that of Qsampling the uniform distribution over all isomorphic graphs. We proceed with the study of quantum state generation as follows:

- In section 3 we prove that any problem in the complexity class statistical zero knowledge (SZK) can be reduced to an instance of QS. SZK contains most problems which are considered good candidates for an efficient quantum algorithm, or for which such an algorithm already exists. Hence, this provides a strong motivation for the study of the QS problem. Additional results related to SZK and to the QS problem are given.
- In section 4 we define a new paradigm for quantum state generation, called adiabatic state generation (ASG). We show that the existence of ASG implies the existence of a standard quantum algorithm to generate the same state, that of polynomially related complexity. Thus, in order to design a quantum state generator, it is sufficient to design ASG for the same state. The ASG paradigm is strongly related in spirit to the framework of adiabatic quantum computation and the physical terminology used therein, such as Schrödinger's equation and the adiabatic theorem. Nevertheless, our definition and proofs do not require any knowledge of those notions and can be understood from first principles.
- Section 5 shows that a fairly general class of classical approximate counting algorithms (that use rapidly mixing Markov chains) can be transformed into ASG algorithms that Qsample from the final distributions of the Markov chains. This solves the QS problem for various interesting cases, such as the uniform distribution over all perfect matchings of a given graph. This section draws intriguing links between the ASG paradigm and Markov chains and spectral gap analysis.
- Section 6 collects lemmas that were used in previous sections and which might be useful when applying the ASG paradigm in other cases. These include the Hamiltonian-to-projection and the Hamiltonian-to-measurement lemmas, the jagged adiabatic path lemma, and the sparse-Hamiltonian lemma, and we explain their meaning below.

The problem of QS was also considered by Grover and Rudolph [30], without a name. They show how to apply standard techniques to construct the state $\sum_i \sqrt{p_i}\,|i\rangle$ for a probability distribution $\{p_i\}$ that is "integrable," i.e., for which $\sum_{i=k}^{\ell} p_i$ can be efficiently computed (approximated) given k and ℓ. One can apply these techniques to construct the states that we construct in section 5. This is done by exploiting the self-reducibility of the problems corresponding to these states. We stress, however, that the techniques we develop in this paper 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 our approach to other quantum states in which self-reducibility cannot be used.

In the remainder of the introduction we provide overviews of each of the different parts of the paper. We note that each one of these sections can be read in an almost self-contained way.

1.1. QS and SZK (section 3).

Our first observation is an interesting connection between the QS problem and the complexity class SZK (see section 3 for the definition and background on this class).

Theorem 1.1. *If QS $\in$ BQP, then SZK $\subseteq$ BQP.*

The proof of Theorem 1.1 relies on a result of Sahai and Vadhan [44]. They defined a problem, called statistical difference, and proved it is SZK-complete. We provide a quantum algorithm for the statistical difference problem given a quantum algorithm for QS.

Theorem 1.1 shows that a general quantum algorithm for the problem of QS implies SZK $\subseteq$ BQP.[1] We note that most problems that were shown to be in BQP or are considered good candidates for BQP, such as discrete log, quadratic residuosity, approximating closest and shortest vectors in a lattice, graph isomorphism, and more, belong to SZK. Theorem 1.1 thus connects the problem of QS to all these algorithmic problems. This motivates our definition and study of the QS problem.

A possibly easier task than solving the general QS problem is to solve specific instances of the problem. To this end, one can apply the proof of Theorem 1.1 to a specific problem in SZK. This would lead to the discovery of the relevant QS instance to which the problem can be reduced. In the general case, this might be quite complicated to do, since the proof of Theorem 1.1 uses the nontrivial completeness result of [44]. In some cases, however, the specification of the relevant QS instance is much easier. Three such cases are discrete log, quadratic residuosity, and a certain lattice related problem. We provide in Appendix A explicit specifications of the QS instances (namely, the quantum superpositions) to which each one of these problems can be reduced. Note that we already know efficient quantum algorithms for the first two problems. The case of solving the Qsampling instance associated with the lattice problem is wide open.

It is interesting to ask whether Theorem 1.1 also holds in the other direction. In other words, is solving QS equivalent to solving the SZK problem, or is the QS problem harder? We show that at least for some cases, equivalence holds. It is easy to see that the QS instance corresponding to discrete log can be solved using the quantum algorithm for discrete log. We prove that the same is also true for the graph isomorphism problem; namely, by trying to solve the QS problem for graph isomorphism, we are not making the problem harder.

Finally, we also study the case of perfect Qsampling. One might hope that if $\mathrm{QS}_{\delta=0}$ can be solved in quantum polynomial time, this would imply that SZK lies in quantum polynomial time with the one-sided error, RQP. We do not know how to prove this, but we provide a slightly weaker result.

1.2. The adiabatic quantum state generation paradigm (section 4). In the past few years, a paradigm called *adiabatic quantum computation* which was defined in [22] attracted considerable attention. Adiabatic quantum computation is a framework for quantum algorithms which uses, instead of the unitary gates used in the standard quantum circuit model, the more physical language of Hamiltonians, spectral gaps, and ground-states, which we will soon explain.

Inspired by adiabatic quantum computation, we define a paradigm for designing quantum state generating algorithms (sometimes called quantum state generators) in the standard quantum circuit model. We call this paradigm ASG. Our goal in the definition of ASG is not to replace the quantum circuit model, or to improve on it, but rather to develop a paradigm, or a language, in which the problem of quantum state generation, and QS in particular, can be studied conveniently. The advantage in using the language of the adiabatic computation model is that the task of quantum state generation becomes more natural, since adiabatic evolution is cast into the language of quantum 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.

Our definition of ASG and results regarding this paradigm do not rely on knowledge of the physical terminology on which adiabatic computation is based, such as

[1] Note that there exists an oracle A relative to which $\mathrm{SZK}^A \not\subseteq \mathrm{BQP}^A$ [1].

Schrödinger's equation and the adiabatic theorem. Nevertheless, since these notions, and the adiabatic computation model in particular, provide so much of the intuition behind our definitions and proofs, we now provide some background regarding these notions to motivate our discussion. We refer the reader to [40, 22, 8] for more information regarding physical background, adiabatic computation, and the adiabatic theorem, respectively.

1.2.1. Adiabatic computation: The physical motivation for ASG. In the standard model of quantum computation, the state of n qubits evolves in *discrete* time steps by unitary operations. In contrast, the underlying physical description of this evolution is *continuous*. This evolution is described by Schrödinger's equation: $\frac{d}{dt}\left|\psi(t)\right\rangle = iH(t)\left|\psi(t)\right\rangle$, where $\left|\psi(t)\right\rangle$ is the state of the n qubits at time t, and $H(t)$ is a Hermitian $2^n \times 2^n$ matrix operating on the space of n qubits. This matrix is called the *Hamiltonian*. The term $\frac{d}{dt}\left|\psi(t)\right\rangle$ stands for $\lim_{\zeta\to 0}\frac{\left|\psi(t+\zeta)\right\rangle-\left|\psi(t)\right\rangle}{\zeta}$ and is a vector measuring the *direction* in which $\left|\psi(t)\right\rangle$ evolves at a given time t. Loosely speaking, the integration of Schrödinger's equation over time from time 0 to a later time t gives the discrete time evolution of the quantum state from time 0 to t; the fact that the Hamiltonian is Hermitian can be shown to be equivalent to the familiar fact that the discrete time evolution is unitary. When the Hamiltonian is independent of time, the solution of Schrödinger's equation is easy: one can verify that Schrödinger's equation is satisfied by

$$\left|\psi(t)\right\rangle = e^{iHt}\left|\psi(0)\right\rangle \tag{1.2}$$

(see section 2 for exponentiation of matrices). Moreover, the fact that H is Hermitian implies that the matrix e^{iHt} is unitary.

From the physicist's point of view, not every Hamiltonian can be used in the above equation, since not every Hamiltonian can be applied on a physical system. The physically realistic Hamiltonians are those that are *local*, namely, involve only interactions between a small number of particles. More formally, such a Hamiltonian H can be written as the sum $H(t) = \sum_m H_m(t)$, where m is small and each $H_m(t)$ is a tensor product of some Hermitian matrix on a small number of qubits with identity on the rest.

An important question in physics is the following. We are given a system which is in some initial state $\left|\psi(0)\right\rangle$ at time $t = 0$, and we let the system evolve according to a time-dependent Hamiltonian $H(t)$ from time $t = 0$ to $t = T$. This means that we set $\left|\psi(0)\right\rangle$ as the initial conditions for Schrödinger's equation and set the Hamiltonian to be $H(t)$. Our goal is to solve the equation and find out the state of the system at time T.

Adiabatic evolution is a special case of the above question in which an elegant solution exists. In adiabatic evolution, one considers a parameterized path in the Hamiltonian domain, $H(s)$ for $s \in [0, 1]$, which starts at some Hamiltonian $H(0) = H_{init}$ and ends at another Hamiltonian $H(1) = H_{final}$. We require that the ground-state (the eigenstate of lowest eigenvalue) of the Hamiltonian $H(s)$ is unique for all $s \in [0, 1]$. To specify the adiabatic evolution, one picks the duration of the process, namely T. The system is initialized at time $t = 0$ in the ground-state of $H(0)$. The system then evolves by Schrödinger's equation from time 0 to time T, under the Hamiltonian $\tilde{H}(t) = H(t/T)$. The term adiabatic means that the Hamiltonian is modified infinitely slowly along the path; in other words, we take $T \mapsto \infty$. In this limit we are guaranteed by the celebrated *adiabatic theorem* [34, 39] that the final state will be equal to the ground-state of the final Hamiltonian H_{final}.

Here we are interested in finite processes. Taking T to be finite introduces an error in the final state: it is no longer exactly the ground-state of $H(T)$ but only close to it in Euclidean distance. How large should T be in order for the error to be small? Two parameters turn out to be important. Denote by $\Delta(H(s))$ the spectral gap of $H(s)$, namely the difference between the Hamiltonian's lowest eigenvalue and the next one. The first parameter is Δ, the minimal spectral gap of the time-dependent Hamiltonian $H(s)$, along the path from $H(0)$ to $H(1)$. The other parameter is related to how fast the Hamiltonian changes in time; we set η to be the maximal norm of the first derivative of H with respect to s: $\eta = \max_s \|\frac{dH}{ds}(s)\|$. It turns out that for the final state to be within ϵ Euclidean distance from the final ground-state, T should be

$$T = poly\left(\frac{\eta}{\Delta\epsilon}\right). \tag{1.3}$$

Different versions of the theorem derive different (small degree) polynomials in the above parameters [43, 8, 10]. In [8] the second derivative of the Hamiltonian with respect to s also plays a role. We learn from (1.3) that if we would like to consider processes with polynomially bounded T, we need η to be polynomially bounded, and Δ to be nonnegligible, namely, bounded from below by a function which is inverse polynomial in the size of the system.

The proof of the adiabatic theorem [39] is rather nontrivial and is beyond the scope of this paper. We refer the reader to [8] for an elementary proof of the theorem and for further references. A very rough intuition about the proof is derived by considering a fixed Hamiltonian H, and observing how the solution to Schrödinger's equation behaves when the initial state is an eigenstate of the Hamiltonian H, with eigenvalue Λ. In this case, the matrix e^{iHt} applied on the eigenstate simply multiplies it by a scalar $e^{i(\Lambda t \bmod 2\pi)}$. This complex number, of absolute value 1, can be viewed as a vector in the complex plane, which rotates in time faster when Λ is larger and slower when Λ is smaller. Hence, for the ground-state it rotates the least. The fast rotations essentially cancel the contributions of the vectors with the higher eigenvalues due to interference effects.

Farhi et al. considered the possibility of using adiabatic quantum evolutions to solve NP-hard optimization problems. The idea of Farhi et al. was to find the (unique) minimum of a given function $f : \{0,1\}^n \to \{0,1\}$ as follows: H_{init} is chosen to be some generic Hamiltonian. H_{final} is chosen to be the *problem Hamiltonian*, namely a $2^n \times 2^n$ 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_{init} and evolves adiabatically on the convex line $H(s) = (1-s)H_{init} + sH_{final}$. By the adiabatic theorem, if the minimal spectral gap is lower bounded by some inverse polynomial, then T can be taken to be polynomially bounded, and the final state would be sufficiently close to the ground-state of H_{final} which is exactly the sought after minimum of f. Despite initial optimistic numerical results [20, 15, 21], there is now strong evidence that the spectral gap along the Hamiltonian path for the NP-hard problems considered is exponentially small [18, 19, 43].

The general model of adiabatic computation (see, e.g., [4]) does not require the final Hamiltonian to be diagonal as above, but it does require the Hamiltonians to be local. We now know that this model is in fact equivalent in computational power to the standard quantum circuit model. The fact that adiabatic computations with local Hamiltonians can be simulated efficiently by the standard model was shown in [22, 18]. The other direction, showing that any standard quantum computation can be simulated efficiently in the adiabatic model with local Hamiltonians, was recently shown by Aharonov et al. [4].[2] Adiabatic computation is thus a quantum algorithmic framework equivalent to the standard quantum computation model, in which computation is thought of as the process of generating (ground-) states. It is thus natural to draw intuition from it when attempting to design frameworks for quantum state generation.

1.2.2. An adiabatic framework for quantum state generation. We would now like to define a paradigm for quantum state generation that is based, in spirit, on similar ideas used in adiabatic computation. Our goal is to develop a tool that can be used when designing quantum algorithms that generate complicated quantum states. We therefore generalize adiabatic computation as much as we can, while maintaining its basic structure. First, we allow the path in the Hamiltonian domain to be a general path (with mild conditions such as smoothness). This is different from the choice often made in adiabatic computation literature, that the path be a straight line. Second, and very importantly, we relax the requirement that the Hamiltonians are local and require only that the Hamiltonians are *simulatable*. This means that the time evolution of the system governed by the Hamiltonian, namely the unitary matrix e^{iHt}, can be approximated by a *quantum circuit* to within any polynomial accuracy (for a rigorous definition see Definition 4.1). An *adiabatic state generator* is thus a specification of such a nicely behaved path of simulatable Hamiltonians in the Hamiltonians domain. The running time of the adiabatic state generator is taken to be exactly the time required for the adiabatic evolution to succeed, roughly given by (1.3) (again, for the exact condition see section 4).

We need to show that the existence of an adiabatic state generator implies the existence of a corresponding quantum state generator of the final state of the adiabatic state generator.

THEOREM 1.2 (informal). *Let A be an adiabatic state generator, initiated by a quantum state $|\psi(0)\rangle$, with a final state $|\psi(T)\rangle$, with a polynomially bounded time T. Then, if there exists an efficient quantum algorithm that generates $|\psi(0)\rangle$, then there exists an efficient quantum algorithm that generates $|\psi(T)\rangle$.*

This result means that one can use the framework of ASG as a quantum *reduction* from a (presumably, difficult to generate) quantum state to another quantum state (which is presumably easier to generate).

To prove the theorem we need to show how to simulate the adiabatic state generator efficiently using a quantum circuit, which is not too difficult, and we also need to show that the final state is indeed close to the ground-state of the final Hamiltonian. The second claim follows immediately from the adiabatic theorem. This provides a natural easy proof of Theorem 1.2, based on the adiabatic theorem. In this paper we prefer to avoid the use of the adiabatic theorem and instead provide an elementary proof which uses only the much simpler Zeno effect [42]. We now sketch the idea.

The Zeno effect considers the following situation. We start at some vector v_0 and apply a sequence of M projective measurements, each in a very close basis to the previous one. More precisely, if the jth measurement is in a basis which includes some vector v_j, we require the vectors v_j to be *slowly varying*; i.e., v_j is close to v_{j-1}. In this case, the Zeno effect states that with very high probability, after M measurements, we end up very close to v_M (even though v_0 and v_M might be very far away from each other). The Zeno effect resembles adiabatic computation in its

[2] In fact, the seeds for the result of [4] were planted in the preliminary version of the current paper.

slowly varying nature, with the important difference that in adiabatic computation we have a sequence of Hamiltonians, while in the Zeno effect we have a sequence of *measurements*. Our proof uses the useful Hamiltonian-to-projection lemma (Lemma 1.2), which we prove in section 6.

We thank Manny Knill [36] for pointing out to us the similarity between adiabatic evolution and the Zeno effect, which lead to this proof. A similar connection was used independently in [14].

1.3. ASG and Markov chains (section 5). We now proceed to show how the ASG paradigm can be used to Qsample from the limiting distributions of various Markov chains. This is done by converting classical approximate counting algorithms, based on rapidly mixing Markov chains, to adiabatic state generators (for a background on Markov chains see section 5).

It is well known that a Markov chain is rapidly mixing iff the second eigenvalue gap, namely the difference between the largest and second largest eigenvalue (in absolute values) of the transition matrix M, is nonnegligible [7]. This clearly bears resemblance to the adiabatic condition of a nonnegligible spectral gap and suggests looking at Hamiltonians of the form

$$H_M = I - M.$$

We use $I - M$ so that the spectrum is reversed; i.e., the largest eigenvector of M becomes the smallest of $H_M = I - H$, and the second eigenvalue gap of M turns into the spectral gap of H_M. H_M defined this way is a Hamiltonian if M is symmetric; if M is not symmetric but is a reversible Markov chain [38], we can still define the Hamiltonian corresponding to it (see section 5).

The first question is whether the resulting Hamiltonian can be used in our ASG framework. In other words, when is the Hamiltonian arising from a Markov chain simulatable? To this end we prove in section 6 a general lemma, called *the sparse Hamiltonian lemma*, which provides a very general condition for a Hamiltonian to be simulatable. Essentially, the condition is that the Hamiltonian be a sparse matrix. Based on this lemma, we show that for a (very natural) class of Markov chains, which we call *strongly samplable*, the Hamiltonian arising from the Markov chain is simulatable and can be used in the ASG paradigm.

In ASG one is interested not in a single Hamiltonian but in a path in the Hamiltonian domain. We recall that many approximate counting algorithms [33] use a sequence of Markov chains. Usually one starts with a simple Markov chain and slowly varies it until it gets close to a desired Markov chain. A notable example is the recent algorithm for approximating the permanent [32]. We show that such approximate counting algorithms naturally translate to adiabatic state generators. More precisely, but still informally, we have the following theorem.

Theorem 1.3 (informal). *Let A be an efficient randomized algorithm to approximately count a set Ω, possibly with weights. Suppose A uses slowly varying Markov chains starting from a Markov chain with a simple limiting distribution. Then there is an efficient quantum algorithm Q that Qsamples the final limiting distribution over Ω.*

We summarize the correspondence between Markov chains and adiabatic computation in Figure 1.1. We stress that it is *not* the case that we are interested in a quantum speed-up for sampling various distributions. Rather, we are interested in the *coherent* quantum state generation of the classical distribution, namely, in the solution for the QS problem.

A Markov chain	⇔	A Hamiltonian
A strongly samplable Markov chain	⇔	A simulatable Hamiltonian
Slowly varying strongly samplable Markov chains	⇔	ASG

FIG. 1.1. *The correspondence between Markov chains and adiabatic computation.*

The proof of Theorem 1.3 uses another general tool, which we call the *jagged adiabatic path lemma*. This lemma shows how we can connect the sequence of Hamiltonians resulting from the sequence of Markov chains, into a continuous path, such that if two subsequent Hamiltonians in the sequence are not too far, and all Hamiltonians in the sequence have nonnegligible spectral gaps, then all Hamiltonians along the path have nonnegligible spectral gaps. We state and prove this theorem in section 6.

We exploit this paradigm to Qsample the set of all perfect matchings of a bipartite graph using the recent algorithm by Jerrum, Sinclair, and Vigoda [32]. Using the same ideas we can also Qsample the set of all linear extensions of partial orders using an algorithm by Bubley and Dyer [12], all lattice points in a convex body satisfying certain restrictions using the Applegate–Kannan technique [9], and many more states.

1.4. Basic tools (section 6). In this section we collect several claims and lemmas that are used in the proofs inside the paper. We separate them from the rest of the paper, since these results are of a general flavor, and we believe they might be useful in other work related to adiabatic state generators, adiabatic computation, and computation with Hamiltonians in general.

We denote by $\alpha(H)$ the unique ground-state of a Hamiltonian H. The first claim shows that two close Hamiltonians have close ground-states, as long as their spectral gaps are big enough.

CLAIM 1.1. *Let A, B be two Hamiltonians of equal dimensions such that $\|A - B\| \leq \eta$. Moreover, assume that A, B have spectral gaps bounded from below: $\Delta(A), \Delta(B) \geq \Delta$. Then $|\langle\alpha(A)|\alpha(B)\rangle| \geq 1 - \frac{4\eta^2}{\Delta^2}$.*

The norm we use is the spectral norm, also called the operator norm (see section 2). The next basic but useful claim provides a lower bound on the spectral gap of a convex combination of two projections. For a vector $|\alpha\rangle$, the Hamiltonian $\Pi_\alpha = I - |\alpha\rangle\langle\alpha|$ is the projection onto the subspace orthogonal to α.

CLAIM 1.2. *Let $|\alpha\rangle, |\beta\rangle$ be two vectors in some Hilbert space. For any convex combination $H_\eta = (1-\eta)\Pi_\alpha + \eta\Pi_\beta$, $\eta \in [0, 1]$, we have $\Delta(H_\eta) \geq |\langle\alpha|\beta\rangle|$.*

Both proofs use simple algebra.

Next, we prove the Hamiltonian-to-measurement lemma, which does the following. We are given a simulatable Hamiltonian with nonnegligible spectral gap. We design an efficient quantum circuit which essentially simulates a measurement in a basis which contains the ground-state of the given Hamiltonian.

LEMMA 1.1 (Hamiltonian-to-measurement lemma). *Assume H is a simulatable Hamiltonian on n qubits. For any constant d, there exists a poly$(n, \frac{1}{\Delta(H)})$-size circuit O_H which takes $|\alpha(H)\rangle$ to $|\alpha(H)\rangle \otimes |\gamma\rangle$, and for any eigenstate of H $|\alpha^\perp\rangle$ orthogonal to the ground-state, $O_H|\alpha^\perp\rangle = |\alpha^\perp\rangle \otimes |\beta(\alpha^\perp)\rangle$, where $|\langle\gamma|\beta(\alpha^\perp)\rangle| \leq O(n^{-d})$.*

The next lemma achieves a related task. It shows that if H is a simulatable Hamiltonian with nonnegligible spectral gap, then the Hamiltonian $\Pi_{\alpha(H)}$, which is the projection on the subspace orthogonal to the ground-state of H, is also simulatable.

LEMMA 1.2 (Hamiltonian-to-projection lemma). *Assume H is a simulatable Hamiltonian on n qubits, with nonnegligible spectral gap $\Delta(H) \geq 1/n^c$ for some constant $c > 0$ and with a known ground-value. Then the Hamiltonian $\Pi_{\alpha(H)}$ is simulatable.*

The proof of both lemmas is a simple application of Kitaev's phase estimation algorithm [35].

The next lemma we prove allows connecting a sequence of Hamiltonians with not too far away ground-states into one adiabatic path.

LEMMA 1.3 (the jagged adiabatic path lemma). *Let $\{H_j\}_{j=1}^{T=poly(n)}$ be a sequence of bounded norm, simulatable Hamiltonians on n qubits, with nonnegligible spectral gaps, $\Delta(H_j) \geq n^{-c}$, and with known ground-values, such that the inner product between the unique ground-states $\alpha(H_j), \alpha(H_{j+1})$ is at least n^{-c} for all j. Then there exists an adiabatic state generator with $\alpha(H_0)$ as its initial state and $\alpha(H_T)$ as its final state. In particular there exists an efficient quantum algorithm that takes $\alpha(H_0)$ to within arbitrarily small distance from $\alpha(H_T)$.*

The proof of this lemma is fairly simple, with one trick required. Our first attempt would be to consider the (jagged) path in the Hamiltonian domain that connects one Hamiltonian in the sequence to the next by a straight line. The main point is to show that the spectral gap along the lines is not too small. In fact, this does not hold in the general case (see section 6.4), but if instead of connecting the Hamiltonians we actually connect the projections $\Pi_{\alpha(H)}$'s, we can then use Claim 1.2 to prove that the convex combination of these projections has a nonnegligible spectral gap.

Finally, we ask which Hamiltonians can be used in the ASG framework, namely, which Hamiltonians are simulatable. We provide a very general condition under which we can simulate the Hamiltonian using a quantum circuit. We say that H on n qubits is *sparse* if it has at most polynomially many nonzero elements in each row (and column, as it is Hermitian). We say it is *explicit* if there exists an efficient classical algorithm that given an index of a row, j, outputs an *approximation* of *all* nonzero elements in the jth row of the Hamiltonian.

DEFINITION 1.2 (an explicit matrix). *We say an $N \times N$ matrix A is explicit if for every $d > 0$ there exists an algorithm that on input $j \in N$ outputs an approximation of all nonzero elements in the jth row of A to within n^{-d} accuracy and whose running time is polynomial in* $\log(N)$.

LEMMA 1.4 (the sparse Hamiltonian lemma). *If H is an explicit and sparse Hamiltonian on n qubits and $\|H\| \leq poly(n)$, then H is simulatable.*

We note that a local Hamiltonian is in particular sparse and explicit, but sparse and explicit Hamiltonians are not necessarily local.

The main idea of the proof is to 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. This is done using some combinatorial and number theoretical tricks. We then show that each Hamiltonian H_m is simulatable. To simulate the sum of the Hamiltonians we use standard techniques (namely, Trotter's formula—see section 4.1.1).

1.5. Conclusions. This paper sets the grounds for the general study of quantum state generation, using the paradigm of ASG. This direction points at interesting and intriguing connections between quantum computation and many different areas: the complexity class SZK and its complete problem SD [44], the notion of adiabatic evolution [34], the study of rapidly mixing Markov chains using spectral gaps [38], quantum random walks [13], and the study of ground-states and spectral gaps of Hamiltonians in physics. Hopefully, techniques from these areas can be borrowed to give more tools for ASG. Notably, the study of spectral gaps of Hamiltonians in physics is a lively area with various recently developed techniques (see [46] and the references therein).

It seems that a much deeper understanding of the adiabatic paradigm is required in order to solve the most interesting open question, namely to design interesting new quantum algorithms. As an intermediate task, it would be interesting to present known quantum algorithms, e.g., Shor's discrete log algorithm, or the quadratic residuosity algorithm, in the ASG paradigm in an insightful way.

1.6. Related work. The definition of ASG uses adiabatic evolutions along general paths in the Hamiltonian domain, and not just straight lines. Such adiabatic evolutions were also studied in [14].

The connection between adiabatic evolution, the Zeno effect, and measurements, which we use in our work, was observed before. We thank Manny Knill for pointing this out to us [36]. These connections were also considered, in a recent independent work, in [14].

We believe that the sparse Hamiltonian lemma might have other interesting implications, e.g., in the context of Hamiltonian based quantum random walks on graphs [16, 23, 13]. For example, Childs et al. [13] use quantum random walks to provide an exponential algorithmic speed-up over any possible classical algorithm for a certain graph reachability task. To do this, they define certain Hamiltonians and use a method of coloring to show that these Hamiltonians can be simulated efficiently by a quantum circuit. The sparse Hamiltonian lemma immediately implies that the Hamiltonians used in [13] are simulatable.

After the publication of the preliminary version of this article [3], the ideas presented in it were used to make progress in two different directions.

The first direction is the characterization of the computational complexity of the problem of approximating the shortest vector in a lattice up to $\sqrt{n}$ (GapSVP$_{\sqrt{n}}$). Our reduction of this problem to a QS problem (section 3) was used in [5] to show that the problem lies in quantum NP. This gave the first nontrivial quantum complexity upper bound on a lattice problem. A following paper [6] improved this result and proved that GapSVP$_{\sqrt{n}}$ lies in NP $\cap$ *co*NP. Interestingly, this result initiated from an attempt to design an ASG algorithm for the relevant QS problem.

The second place where these results inspired further progress is in the study of adiabatic computation, where an important open question was the clarification of the computational power of the model. Our results raised the question of how powerful quantum adiabatic algorithms are and gave tools to prove some preliminary results about their universality [2]. These results were recently improved in [4] to show that the model of adiabatic computation using *local* Hamiltonians is equivalent to standard quantum computation.

1.7. Paper organization. The paper is organized as follows. We give some notation and general mathematical preliminaries in section 2. Background related to particular parts of the paper is given at the beginning of each section.

In section 3 we show that the QS problem is sufficient for solving all the languages in SZK, and we also discuss whether it is equivalent to solving SZK. The specific examples of the Qsampling instances associated with discrete log, quadratic residuosity, and a lattice problem are given in Appendix A. We define adiabatic quantum state generation in section 4. We also show (using measurements and the Zeno effect) that adiabatic state generators can be simulated by quantum circuits. In section 5 we show the connection to Markov chains, and prove that a host of approximating counting algorithms can be translated into adiabatic state generators, generating many interesting coherent states. Finally, in section 6, we prove several lemmas that serve as basic building blocks for our previous results, including the sparse Hamiltonian lemma, the jagged adiabatic path lemma, and the Hamiltonian-to-projection and Hamiltonian-to-measurement lemmas.

2. Preliminaries.

We assume the reader is familiar with the basic terminology of quantum computation: qubits, pure states, Hilbert space, density matrix, the class BQP, etc. For background on these notions, please consult [40]. We now give some preliminaries relevant for the entire paper. More specific preliminaries are given at the beginning of each section.

2.1. Distances between distributions: Fidelity and variational distance.

For two classical distributions $\{p(x)\}, \{q(x)\}$ we define their *variational distance* and their *fidelity* (this measure is known by many other names as well) to be, respectively,

$$|p-q| = \frac{1}{2}\sum_x |p(x)-q(x)|,$$

$$F(p,q) = \sum_x \sqrt{p(x)q(x)}.$$

The following fact is very useful.

Fact 2.1 (see [40]).

$$1 - F(p,q) \leq |p-q| \leq \sqrt{1-F(p,q)^2}$$

or, equivalently,

$$1 - |p-q| \leq F(p,q) \leq \sqrt{1-|p-q|^2}.$$

A distribution D is flat if for every z_1 and Z_2 for which $D(z_1), D(z_2) > 0$ we have $D(z_1) = D(z_2)$; i.e., D is uniform over all elements in its support.

2.2. Norms on matrices: Trace norm and operator norm.

The trace norm of a Hermitian matrix H with eigenvalues $\lambda_1, \ldots, \lambda_n$ is $\|H\|_{tr} = \sum |\lambda_i|$. Note that the trace norm of a density matrix is 1. The trace norm satisfies that $\|A \otimes B\|_{tr} = \|A\|_{tr}\|B\|_{tr}$.

The operator norm of a linear transformation T induced by the l_2 norm is called the *spectral norm* and is defined by

$$\|T\| = \max_{\psi \neq 0} \frac{|T\psi|}{|\psi|}.$$

The operator norm satisfies that for any two matrices, $\|AB\| \leq \|A\| \cdot \|B\|$.

If T is Hermitian or unitary (in general, if T is normal, namely, commutes with its adjoint), then $\|T\|$ equals the largest absolute value of its eigenvalues. Hence, if U is unitary, $\|U\| = 1$.

For any two unitary matrices A and B and any integer k, $\|A^k - B^k\| \leq k\|A-B\|$. This follows from the fact that $\|AB - CD\| \leq \|AB - CB\| + \|CB - CD\|$, which for unitary matrices is $\leq \|A-C\| + \|B-D\|$.

Finally, for a general $N \times N$ matrix $A = (a_{i,j})$ we have $\|A\|_\infty \leq \|A\| \leq N^2\|A\|_\infty$, where $\|A\|_\infty = \max_{i,j} |a_{i,j}|$.

2.3. Distances between density matrices. The variational distance and the fidelity can be generalized to density matrices, and Fact 2.1 also holds for density matrices (see [40]).

The generalization of the variational distance is the trace norm of the difference between the two matrices. It is a well-known fact that the two output distributions resulting from applying the same quantum measurement on two different density matrices, ρ_1 andρ_2, can have variational distance at most $\frac{1}{2}\|\rho_1 - \rho_2\|_{tr}$. For more details we refer the reader to [40, section 9.2].

In this paper we need only define fidelity for pure states. For two vectors ϕ_1, ϕ_2 in some Hilbert space, the fidelity is simply the absolute value of their inner product: $F(\phi_1, \phi_2) = |\langle\phi_1|\phi_2\rangle|$.

2.4. Power of a matrix. If M is a Hermitian matrix, then it has an orthonormal basis of eigenvectors $\{v_i\}$ with real eigenvalues $\{\lambda_i\}$. For a function $f : \mathbb{C} \to \mathbb{C}$, $f(M)$ is the linear transformation that has $\{v_i\}$ as an orthonormal basis of eigenvectors with eigenvalues $\{f(\lambda_i)\}$. In particular, this defines e^M.

2.5. Hamiltonian terminology. The set of Hamiltonians is the set of Hermitian matrices. The ground-state of a Hamiltonian H is the eigenstate with the smallest eigenvalue, and 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 $\Delta(H)$. If H is Hermitian, then its eigenvalues are real, and hence e^{-iH} is unitary.

3. Quantum state generation and SZK.

In this section we connect the QS problem to the class SZK. We start with some background about SZK. We refer the interested reader to Vadhan's thesis [47] and to Sahai and Vadhan [44] for rigorous definitions, a discussion of their subtleties, and other results known about this elegant class. We then proceed to prove Theorem 1.1, and in Appendix A we provide explicit examples of interesting QS instances. We also prove that the task of QS for graph isomorphism is not harder than solving the graph isomorphism problem itself, and that if QS can be done with no error, then the graph isomorphism problem is in RQP$\bigcap$*co*RQP.

3.1. Background on SZK.

3.1.1. Interactive proofs. A pair $\Pi = (\Pi_{Yes}, \Pi_{No})$ is a promise problem if $\Pi_{Yes} \subseteq \{0,1\}^*$, $\Pi_{No} \subseteq \{0,1\}^*$, and $\Pi_{Yes} \cap \Pi_{No} = \emptyset$. We look at Π_{Yes} as the set of all *yes* instances and Π_{No} as the set of all *no* instances, and we do not care about all other inputs. If every $x \in \{0,1\}^*$ is in $\Pi_{Yes} \cup \Pi_{No}$, we call Π a language.

An interactive proof is a protocol in which a prover P tries to convince a verifier V of some fact through an exchange of messages. Formally, the prover and the verifier are described by probabilistic Turing machines which act on their private working spaces plus some interaction domain. The verifier is required to be polynomial time, and the prover is assumed to be all powerful. The interactive proof is denoted by (P, V).

We say that a promise problem Π has an interactive proof with soundness error ϵ_s and completeness error ϵ_c if there exist V, P such that we have the following:

- If $x \in \Pi_{Yes}$, V accepts with probability at least $1 - \epsilon_c$.
- If $x \in \Pi_{No}$, then *for every* prover P^*, V accepts with probability at most ϵ_s.

The class NP consists of one-message interactive proofs with $\epsilon_c, \epsilon_s = 0$.

When an interactive proof system (P, V) for a promise problem Π is run on an input x, it produces a distribution over *transcripts* that contains the conversation between the prover and the verifier; i.e., each possible transcript appears with some probability (depending on the random coin tosses of the prover and the verifier).

3.1.2. SZK. The class SZK consists of promise problems for which there are interactive proofs which exhibit the following remarkable property: for $x \in \Pi_{yes}$, the verifier learns (almost) nothing from the interaction with the prover P, other than the fact that x is a *yes* instance. It is remarkable that such proof systems in fact exist. This is captured mathematically by the concept of *simulation* as follows.

An interactive proof system (P, V) for a promise problem Π is said to be an honest verifier SZK, if there exists a probabilistic polynomial time *simulator* S that for every $x \in \Pi_{Yes}$ produces a distribution on transcripts that is close (in the variational distance sense; see section 2.1) to the distribution on transcripts that V and P would produce in their interaction. Note that the simulator has no access to the prover, and that we require only the simulator to produce a good distribution on inputs in Π_{Yes}, since for *no* instances there is no proof to learn anyway.

One might wonder whether it is possible for the verifier to deviate from the protocol (namely, to *cheat*) and by this to get information from an honest prover. Indeed, there are honest verifier SZK proofs which are not secure against a cheating verifier. However, it was shown in [27] that whenever there exists an honest verifier SZK proof, then there is also an interactive proof that is also secure against dishonest verifiers. By this we mean that a simulator also exists for verifiers that deviate from the protocol.

We denote by SZK the class of all promise problems which have interactive proof systems which are statistically zero knowledge against an honest (or, equivalently, a general) verifier. It is known that BPP $\subseteq$ SZK $\subseteq$ AM $\cap$ *co*AM [24, 11, 47] and that SZK is closed under complement [41, 47]. It follows that SZK does not contain any NP-complete language unless the polynomial time hierarchy collapses.

3.1.3. A complete problem for SZK. Sahai and Vadhan [44] found a natural complete problem for SZK. One nice thing about the problem is that it does not mention interactive proofs in any explicit or implicit way. We define the complete problem for SZK.

DEFINITION 3.1 (statistical difference ($\mathrm{SD}_{\alpha,\beta}$)).

Input: *Two classical circuits C_0, C_1 with m Boolean outputs.*

Promise:

- *Yes:* $|D_{C_0} - D_{C_1}| \geq \alpha$.
- *No:* $|D_{C_0} - D_{C_1}| \leq \beta$.

Sahai and Vadhan [44] and Vadhan [47] show that for any two constants $0 < \beta < \alpha < 1$ such that $\alpha^2 > \beta$, $\mathrm{SD}_{\alpha,\beta}$ is complete for SZK.

3.2. A reduction from SZK to QS. We are now ready to prove Theorem 1.1. We first describe a very simple, standard building block in quantum computation, called the SWAP test.

DEFINITION 3.2 (the SWAP test). *The algorithm operates on three quantum registers: A is a one qubit register, and B and C are two registers with the same number of qubits. The algorithm applies a Hadamard on the first qubit, then conditioned on the first control qubit swaps between the second and third registers, and, finally, applies a Hadamard on the control qubit and measures it.*

By a direct calculation, we have the following claim.

CLAIM 3.1. *Let v_1, v_2 be two vectors in the same Hilbert space. If the SWAP test is applied on $|0, v_1, v_2\rangle$, then the outcome of the SWAP test is* 0 *with probability* $\frac{1+|\langle v_1|v_2\rangle|^2}{2}$ *and* 1 *with probability* $\frac{1-|\langle v_1|v_2\rangle|^2}{2}$.

We now proceed to prove Theorem 1.1.

Proof of Theorem 1.1. We assume that QS is in BQP. It is enough to show that $\mathrm{SD}_{0.9,0.1}$, which is an SZK-complete problem, is in BQP.

Indeed, let C_0, C_1 be an input to $\mathrm{SD}_{0.9,0.1}$. By our assumption there is an efficient quantum algorithm that can generate states ρ_0, ρ_1 such that $\|\rho_i - |C_i\rangle\langle C_i|\|_{tr} \le \delta$ for $i = 0, 1$ and $\delta = 10^{-5}$. We can therefore apply the SWAP test on the two states ρ_0, ρ_1 efficiently. We now claim the test results in the outcome 1 with probability greater than 0.4 in case $|D_{C_0} - D_{C_1}| \ge 0.9$ and with probability smaller than 0.1 in case $|D_{C_0} - D_{C_1}| \le 0.1$.

The BQP algorithm follows from this claim easily: To achieve error ϵ, simply repeat the SWAP test $O(\log(\frac{1}{\epsilon}))$ times, generating the states each time from scratch. Then count the number of outcomes 1. If it is more than 0.25 of the tests, accept (the distributions are far); otherwise, reject (the distributions are close).

To prove the claim, we first write down the probability for 1 in the ideal case, in which the BQP algorithm outputs $|C_i\rangle$ exactly. We have

$$\langle C_0|C_1\rangle = \sum_{z\in\{0,1\}^m} \sqrt{D_{C_0}(z)D_{C_1}(z)} \;=\; F(D_{C_0}, D_{C_1}).$$

Claim 3.1 implies, therefore, that the SWAP test on the state $|0, C_0, C_1\rangle$ results in 1 with probability $\frac{1-F(D_{C_0},D_{C_1})^2}{2}$.

In fact, the state ρ_i is within δ trace distance from $|C_i\rangle$. This implies that the actual state on which we apply the swap test is $\rho_1 \otimes \rho_2$, which is 2δ-close in the trace norm to that of the pure state $|0, C_0, C_1\rangle$ (see section 2.2). By section 2.3, the variational distance between the distributions resulting from applying the SWAP test in the two cases is δ. This implies that the probability for 1 in the actual SWAP test is $\frac{1-F(D_{C_0},D_{C_1})^2}{2} \pm \delta$.

Using Fact 2.1, we have the following:

- If $|D_{C_0} - D_{C_1}| \ge \alpha$, we measure 1 with probability $\frac{1-F(D_{C_0},D_{C_1})^2}{2} \pm \delta \ge \frac{|D_{C_0}-D_{C_1}|^2}{2} - \delta \ge \frac{\alpha^2-2\delta}{2}$.
- If $|D_{C_0} - D_{C_1}| \le \beta$, we measure 1 with probability $\frac{1-F(D_{C_0},D_{C_1})^2}{2} \pm \delta \le \frac{2|D_{C_0}-D_{C_1}|-|D_{C_0}-D_{C_1}|^2}{2} + \delta \le \frac{2\beta-\beta^2+2\delta}{2}$.

Setting $\alpha = 0.9$ and $\beta = 0.1$, we get the desired results. □

3.3. Perfect QS and one-sided-error quantum algorithms. One might hope that if one could perfectly solve QS (i.e., $\mathrm{QS}_{\delta=0} \in$ BQP), then SZK $\subseteq$ RQP, where RQP is the one-sided variant of BQP. This, however, does not follow, because $\mathrm{SD}_{\alpha,\beta}$ is known to be SZK complete only when $\beta > 0$ and $\alpha < 1$. Instead, we can prove a weaker version of this general result, concerning the class honest verifier *perfect* zero knowledge (HVPZK), where the simulator can *exactly* simulate the transcripts distribution. This class contains the graph isomorphism and the graph nonisomorphism problems.

LEMMA 3.1. *If $QS_{\delta=0} \in BQP$, then coHVPZK $\subseteq$ RQP.*

Proof. The proof uses the fact that $\mathrm{SD}_{0.5,0}$ is complete for coHVPZK [47]. It is enough to show that $\mathrm{SD}_{0.5,0}$ is in RQP. Indeed, let C_0, C_1 be an input to $\mathrm{SD}_{0.5,0}$. By our assumption we can generate the superpositions $|C_i\rangle$ for $i = 0, 1$. The quantum algorithm proceeds as in the proof of Theorem 1.1 and accepts iff the result of one of the measurements is 1. For yes instances, $|D_{C_0} - D_{C_1}| \geq \alpha = 0.5$, and so we measure 1 with probability at least 0.12. For no instances, we never measure 1. Hence we get an RQP algorithm. □

As both graph isomorphism and graph nonisomorphism are in HVPZK, we get the following corollary.

Corollary 3.1. *If* $QS_{\delta=0} \in BQP$, *then* $GI \in RQP \bigcap coRQP$.

3.4. Specific examples. We saw that every problem L in SZK reduces to a pair of circuits $C_{L,0}, C_{L,1}$ such that if we can Qsample $|C_{L,i}\rangle$, we can solve L in quantum polynomial time. Unfortunately, we do not know how to solve the QS problem in general. We would like to specify explicitly interesting instances of the QS problem, associated with specific problems in SZK.

In theory, such an instance can be derived from the SZK proof of the promise problem in the following way. For every problem L in SZK, one can follow the reduction from L to $\mathrm{SD}_{0.9,0.1}$ (guaranteed by the SZK-completeness of $\mathrm{SD}_{0.9,0.1}$ [44]) and find two specific circuits $C_{L,i}$ corresponding to L. Qsampling from these circuits would be sufficient for solving L in quantum polynomial time. In practice, however, specifying the circuits is often not easy, as the reduction to $\mathrm{SD}_{0.9,0.1}$ is quite involved.

However, it is often possible to infer two such circuits $C_{L,i}$ directly from the zero-knowledge proof of L. We already saw in the introduction such a specific example for the graph isomorphism problem. In Appendix A we give three more examples of particular interest for quantum algorithms: discrete log, quadratic residuosity, and a gap version of closest vector in a lattice.

3.5. Is solving QS equivalent to solving SZK? We saw that QS $\in$ BQP implies that SZK $\subseteq$ BQP. A natural question is whether the QS problem is *equivalent* to solving SZK or *strictly harder*.

We start with the simplest case. Say L is a (promise) problem such that

- for any x, (L, x) can be efficiently reduced to solving the instance $|C_x\rangle$ of QS,
- C_x is one-to-one on its inputs, and
- there exists a procedure in BQP that using L as an oracle can invert C_x; i.e., given z, it computes a y such that $C_x(y) = z$.

For example, the discrete log problem gives rise to such a situation (see the problem DLP and the circuit C given in Appendix A). We claim the following.

Claim 3.2. *If* L *and* C *are as above, then* $L \in BQP$ *iff* C *is Qsamplable.*

Proof. We already know that (L, x) can be reduced to solving the instance $|C_x\rangle$ of QS. We show the other direction. Assume $L \in$ BQP. Fix some input x. Then, given $|y\rangle$, we can compute $|y, C_x(y)\rangle$ (because the circuit C_x is given to us), and, given $|C_x(y)\rangle$, we can compute $|C_x(y), y\rangle$ (because $L \in$ BQP and we assume we can invert C_x using L).[3] It then follows that there exists an efficient procedure that *replaces* $|y\rangle$ with $|C_x(y)\rangle$ (by undoing the computation). In particular we can build the superposition $\sum_y |y\rangle$ and transform it into the superposition $|C_x\rangle = \sum |C_x(y)\rangle$. □

Next, we consider the case where C_x is not one-to-one but rather *regular*; i.e., the distribution $D_C = D_{C_x}$ it induces is flat. Let us further assume that we have an

[3] In fact, we approximate only the state, since we run a BQP algorithm for the inversion procedure, and this algorithm may err.

efficient way to complete C_x to a one-to-one function. Formally, say L is a (promise) problem such that

- for any x, (L, x) can be efficiently reduced to solving the instance $|C_x\rangle$ of QS, and C_x is a circuit computing a function $C_x : \{0,1\}^n \rightarrow \Lambda_C$ for some domain Λ_C,
- there exists an efficient function $f_x : \{0,1\}^n \rightarrow \Lambda_f$, for some domain Λ_f, such that $C_x \otimes f_x : \{0,1\}^n \rightarrow \Lambda_C \times \Lambda_f$ (defined by $(C_x \otimes f_x)(y) = (C_x(y), f_x(y))$) is *one-to-one* and *onto*, and
- there exists an efficient procedure that using L as an oracle can invert $C_x \otimes f_x$; i.e., given z, it computes a y such that $(C_x \otimes f_x)(y) = z$.

We claim the following.

CLAIM 3.3. *If L and C are as above, then $L \in BQP$ iff C is Qsamplable.*

Proof. As before, we can create the state $\phi = \sum |C_x(y), f_x(y)\rangle$. As $C_x \otimes f_x$ is one-to-one and onto $\Lambda_C \times \Lambda_f$, we have that $\phi = \sum_{z \in \Lambda_C} |z\rangle \otimes \sum_{v \in \Lambda_f} |v\rangle$. Hence ϕ is in fact a product state, and we get the state $|C_x\rangle$ by just ignoring the second register. □

Graph isomorphism is an example to such a situation, as we now show. A key fact that we use is that there exists a deterministic search-to-decision reduction for graph isomorphism (see, e.g., [37, section 1.2]). Given any two isomorphic graphs G and G', the reduction R gives a permutation $\pi = R(G, G') \in \mathcal{S}_n$ such that $\pi(G) = G'$, where n is the number of vertices in G, and $\mathcal{S}_n$ is the set of all permutations on n elements.

Then the circuit $C_G : \mathcal{S}_n \rightarrow \mathcal{S}_n(G)$ gets $\pi \in \mathcal{S}_n$ as an input and outputs the permuted graph $\pi(G)$, $C_G(\pi) = \pi(G)$. The function $f_G : \mathcal{S}_n \rightarrow Aut(G)$ is defined by $f(\pi) = (R(G, \pi(G)))^{-1} \cdot \pi$ (where the product is in $\mathcal{S}_n$). We leave it to the reader to show that $f(\pi) \in Aut(G)$, that $C \otimes f$ is one-to-one and onto, and that the above three conditions are satisfied. Then we have the following lemma.

LEMMA 3.2. *GI $\in$ BQP iff $|\alpha_G\rangle = \sum_{\sigma \in S_n} |\sigma(G)\rangle$ can be generated in BQP.*

It is tempting to try extending the above approach in order to prove Lemma 3.2 for the SZK-complete problem $\mathrm{SD}_{\alpha,\beta}$. However, we face the following problems:

- C_x might not be regular; i.e., different elements $C_x(y)$ might have a different number of preimages.
- Even worse, even if we assume C_x is regular, in fact even if C_x is a permutation, it might be possible that C_x is hard to invert (and then C_x is a one-way function), and it is possible that it is hard to invert even given access to an oracle solving L.

  Thus, for this approach to work, it must be true that if $L = \mathrm{SD}$ (and therefore also the whole of SZK) is easy (classically or quantumly), then there are no one-way functions in the quantum model. We note that the question of whether it is possible that SZK = BPP but yet one-way functions exist (in the classical model) is a major open problem (see [47, Open problem 4.8.10]).

We therefore do not know if, in general, solving QS in BQP is equivalent to solving SZK in BQP, and we leave it as an open problem.

4. The ASG paradigm. In this section we define the paradigm of ASG. At the end of the section we formally state and prove Theorem 1.2, which states that any adiabatic state generator can be simulated efficiently by a quantum circuit. This is done using the Zeno effect. As mentioned before, our proof does not rely on the adiabatic theorem. We start with some background on the Trotter formula, which we need for our proofs in this section.

4.1. Preliminaries.

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

$$\lim_{n\to\infty} (e^{iAt/n} e^{iBt/n})^n = e^{i(A+B)t}.$$

In other 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 n, and for that we use the following variant from ([40, eq. 4.104]):

$$||e^{2\delta i(A+B)} - e^{\delta iA} e^{2\delta iB} e^{\delta iA}|| \leq O((\max\{||A||, ||B||\} \cdot \delta)^3). \tag{4.1}$$

We also need to deal with Hamiltonians of the form $H = \sum_m H_m$ that are sums of $m > 2$ Hamiltonians. We prove the following lemma (a very similar statement appears in [40, Exercise 4.50]).

Lemma 4.1. *Let H_m be Hermitian, $m = 1, \ldots, M$, and let $H = \sum_{m=1}^M H_m$. Further, assume that for every $1 \leq k \leq \ell \leq M$ we have $\| \sum_{i=k}^{\ell} H_i \| \leq \Lambda$. Define*

$$U_\delta = [\ e^{\delta i H_1} \cdot e^{\delta i H_2} \cdot \cdots \cdot e^{\delta i H_M}\ ] \cdot [\ e^{\delta i H_M} \cdot e^{\delta i H_{M-1}} \cdot \cdots \cdot e^{\delta i H_1}\ ]. \tag{4.2}$$

Then $\|U_\delta - e^{2\delta i H}\| \leq O(M \cdot (\delta\Lambda)^3)$.

Proof. We prove by induction on M. The case $M = 2$ is (4.1). For the induction step, we notice that by (4.1)

$$||e^{2\delta i \sum_{i=1}^M H_i} - e^{\delta i H_1} e^{2\delta i \sum_{i=2}^M H_i} e^{\delta i H_1}|| \leq O((\delta\Lambda)^3).$$

Also, $U_\delta = e^{\delta i H_1}[e^{\delta i H_2} \cdot \cdots \cdot e^{\delta i H_M}\ ] \cdot [\ e^{\delta i H_M} \cdot \cdots \cdot e^{\delta i H_2}\ ] e^{\delta i H_1}$. Thus,

$$\begin{aligned} &||U_\delta - e^{2\delta i \sum_{i=1}^M H_i}|| \\ &\quad \leq ||[e^{\delta i H_2} \cdot \cdots \cdot e^{\delta i H_M}\ ] \cdot [\ e^{\delta i H_M} \cdot \cdots \cdot e^{\delta i H_2}\ ] - e^{2\delta i \sum_{i=2}^M H_i}|| + O((\delta\Lambda)^3), \end{aligned}$$

and by induction this is bounded by $O(M(\delta\Lambda)^3)$. □

Corollary 4.1. *Let H, H_m satisfy the conditions of Lemma 4.1. Then, for every $t > 4\delta$,*

$$\left\| U_\delta^{\lfloor \frac{t}{2\delta} \rfloor} - e^{-itH} \right\| \leq O(\Lambda \cdot \delta + M\Lambda^3 t \cdot \delta^2).$$

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

Proof.

$$\begin{aligned} \left\| U_\delta^{\lfloor \frac{t}{2\delta} \rfloor} - e^{-itH} \right\| &\leq \left\lfloor \frac{t}{2\delta} \right\rfloor \cdot \left\| U_\delta - e^{\frac{-it}{\lfloor \frac{t}{2\delta} \rfloor} H} \right\| \\ &\leq \frac{t}{2\delta} \cdot \left[\left\| U_\delta - e^{\frac{-it}{\frac{t}{2\delta}} H} \right\| + \left\| e^{\frac{-it}{\frac{t}{2\delta}} H} - e^{\frac{-it}{\lfloor \frac{t}{2\delta} \rfloor} H} \right\| \right]. \end{aligned}$$

The first term $\|U_\delta - e^{\frac{-it}{\frac{t}{2\delta}}H}\| = \|U_\delta - e^{-i2\delta H}\| \leq O(M \cdot (\delta\Lambda)^3)$, by Lemma 4.1.

For the second term $\|e^{\frac{-it}{\frac{t}{2\delta}}H} - e^{\frac{-it}{\lfloor\frac{t}{2\delta}\rfloor}H}\|$, we notice that both matrices (and therefore also their difference) have the same eigenvector basis (that of H). As the norm is maximized at some eigenvector, $\|e^{\frac{-it}{\frac{t}{2\delta}}H} - e^{\frac{-it}{\lfloor\frac{t}{2\delta}\rfloor}H}\| = |e^{\frac{-it}{\frac{t}{2\delta}}\lambda} - e^{\frac{-it}{\lfloor\frac{t}{2\delta}\rfloor}\lambda}|$ for some eigenvalue λ with $|\lambda| \leq \Lambda$ (because H has bounded norm). We now use the identities $|e^{-\theta i} - e^{-\theta' i}| = 2|\sin(\frac{\theta-\theta'}{2})| \leq |\theta - \theta'|$. We see that the second term is bounded by $\frac{8\delta^2\lambda}{t}$.

Altogether,

$$\begin{aligned}\left\|U_\delta^{\lfloor\frac{t}{2\delta}\rfloor} - e^{-itH}\right\| &\leq O\left(\frac{t}{\delta}\right) \cdot \left[(M \cdot (\delta\Lambda)^3) + \frac{\Lambda\delta^2}{t}\right] \\ &= O(M\Lambda^3 t\delta^2) + O(\Lambda\delta). \qquad \Box\end{aligned}$$

4.2. Adiabatic quantum state generation. We now define our paradigm for *quantum state generation* inspired by the adiabatic theorem. As explained in the introduction, we would like to allow as much flexibility as possible and therefore allow any Hamiltonian which can be implemented efficiently by quantum circuits. We define the following.

Definition 4.1 (simulatable Hamiltonians). *We say that a Hamiltonian H on n qubits is simulatable if for every real value $t > 0$ and every accuracy $0 < \epsilon < 1$ the unitary transformation*

$$U(t) = e^{-iHt}$$

can be approximated by a quantum circuit of size $poly(n, t, 1/\epsilon)$ to within ϵ accuracy in the operator norm.

Corollary 4.1 implies that a local Hamiltonian is simulatable (but the other direction is not true). 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 that any convex combination of two simulatable, polynomially 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 $e^{-itUHU^\dagger} = Ue^{-itH}U^\dagger$. We note that these rules cannot be applied unboundedly many times in a recursive way, because the simulation will then blow up. The interested reader is referred to [40, 13] for a more complete set of rules for simulating Hamiltonians.

We now describe an adiabatic path, which is an allowed path in the Hamiltonian space.

Definition 4.2 (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 continuous,*
- *$H(s)$ is differentiable, except for polynomially many points,*
- *for all s, $H(s)$ has a unique ground-state, and*
- *for all s, $H(s)$ is simulatable given s.*

Adiabatic quantum state generation is supposed to mimic the process of implementing Schrödinger's evolution along an adiabatic path, where the adiabatic condition holds.

In our case, we use simulatable Hamiltonians rather than local Hamiltonians. The time associated with ASG is defined using similar parameters to those used in the adiabatic theorem (as explained in the introduction). For an adiabatic path $H(s)$ we define

$$\eta(H(\cdot)) = \max_{s\in[0,1]\setminus\mathbf{D}} \left\| \frac{dH}{ds}(s) \right\| \text{ and}$$
$$\Delta(H(\cdot)) = \min_{s\in[0,1]} \Delta(H(s)),$$

where in the above, D is the set of at most polynomially many points where the derivative is not defined.

Definition 4.3 (adiabatic quantum state generation). *An adiabatic quantum state generator $H_x(s)$ is a function from $x \in \{0,1\}^n$ to adiabatic paths $\{H_x(s)\}_{s\in[0,1]}$. We require that the generator is* explicit*, i.e., that there is a quantum machine running in time polynomial in its input and output length, such that*

- *on input $x \in \{0,1\}^n$ outputs $\alpha(H_x(0))$, the ground-state of $H_x(0)$, and*
- *on input $x \in \{0,1\}^n$, $s \in [0,1]$, $t > 0$, and ϵ outputs a $poly(n, t, \frac{1}{\epsilon})$-size circuit $C_x(s)$ approximating $e^{-itH_x(s)}$ to within ϵ accuracy.*

We define $T(x,\epsilon) = \frac{\eta^2(H_x(\cdot))}{\epsilon\cdot\Delta^2(H_x(\cdot))}$. For $\epsilon > 0$ we let $T_\epsilon = \max_x\{T(x,\epsilon)\}$, and we say the adiabatic quantum state generator $H(\cdot)$ takes time T_ϵ (for the given ϵ).

4.3. Circuit simulation of adiabatic quantum state generation. We now prove that an adiabatic quantum state generator can be simulated efficiently by a quantum circuit.

Theorem 1.2 (formal). *Let $\epsilon > 0$. Let $H_x(s)$ be an adiabatic state generator taking time T_ϵ. There exists a quantum circuit of size $poly(T_\epsilon, \frac{1}{\epsilon}, n)$ such that for every input x, it generates $\alpha(H_x(1))$ to within ϵ accuracy.*

Proof. We start by an overview of the proof. The circuit is built by discretizing time to sufficiently small intervals of length $\delta = \frac{1}{R}$ for some large enough $R = poly(T_\epsilon, \frac{1}{\epsilon}, n)$. At each time step j, $j = 1, \ldots, R$, we apply a measurement in a basis which includes the ground-state $\alpha(H(s_j))$. In other words, we attempt to project $\alpha(H(s_{j-1}))$ onto $\alpha(H(s_j))$. This is done using the Hamiltonian-to-measurement lemma (Lemma 1.1). If R is sufficiently large, the subsequent Hamiltonians are very close in the spectral norm, and their ground-states are very close in the Euclidean norm (by Claim 1.1). Given that at time step j the state is the ground-state $\alpha(H(s_j))$, the next measurement results with very high probability in a projection onto the new ground-state $\alpha(H(s_{j+1}))$. The Zeno effect [42] 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 ground-state of $H(1)$ is very high, if R is taken to be large enough. We now give a formal treatment.

The description of the quantum circuit. For a given input x, the adiabatic state generator specifies an adiabatic path $H_x(s)$. Recall that $[0,1]$ can be decomposed into $m = poly(n)$ time intervals of the form $[s_j, s_{j+1}]$ where $H(\cdot)$ is continuous on $[s_j, s_{j+1}]$ and differentiable on (s_j, s_{j+1}). Let $\eta = \eta(H_x(\cdot)), \Delta = \Delta(H_x(\cdot))$. We divide each interval into R equal intervals, where we choose $R \geq \Theta(\frac{\eta^2}{\Delta^2}\frac{m}{\epsilon})$, and we set $t_{j,k} = s_j + (s_{j+1} - s_j)\frac{k}{R}$. For each interval, we apply the following R steps. At the kth step, $k = 1, \ldots, R$, we apply the operation $O_{H(t_{j,k})}$ defined in the statement of the Hamiltonian-to-measurement lemma (Lemma 1.1). Each of these applications of O_H takes time which is $poly(n, 1/\Delta)$, by Lemma 1.1. The complexity of the algorithm is therefore $O(\frac{\eta^2}{\Delta^2}\frac{m^2}{\epsilon})$ times the complexity of applying the measurement from Lemma 1.1. This is indeed $poly(T_\epsilon, n, 1/\epsilon)$.

Error analysis in the case that O_H is perfect. We first show the algorithm works when we assume that the O_H's are perfect; i.e., in Lemma 1.1, $\langle\gamma|\beta(\alpha^\perp)\rangle = 0$. We show that starting with the state $\alpha(H(s_j))$, the state after the jth interval is, with high probability, $\alpha(H(s_{j+1}))$. We first bound the relative change of $H(s+\delta)$ with respect to $H(s)$. For $s, s+\delta \in [s_j, s_{j+1}]$,

$$\|H(s+\delta) - H(s)\| = \left\| \int_s^{s+\delta} \frac{dH}{ds}(s)ds \right\| \\ \leq \int_s^{s+\delta} \left\| \frac{dH}{ds}(s) \right\| ds \ \leq \ \eta \cdot \delta.$$

Hence, $\|H(t_{j,k+1}) - H(t_{j,k})\| \leq \frac{\eta}{R}$. Claim 1.1 implies that

$$|\langle \alpha(H(t_{j,k+1})) \mid \alpha(H(t_{j,k})) \ \rangle| \ \geq \ 1 - 4\frac{\eta^2}{R^2\Delta^2}.$$

Hence the probability for successful projection at the $k'th$ measurement, i.e., the probability that the outcome is indeed the ground-state, is $(1 - \frac{4\eta^2}{R^2\Delta^2})^2 \geq 1 - \frac{8\eta^2}{R^2\Delta^2}$. The probability that we err at any of the R steps in the jth interval is therefore at most $O(\frac{\eta^2}{R\Delta^2})$. And the probability that we err at any of the intervals is therefore at most m times that. This is at most ϵ by our choice of R.

Including nonperfect O_H's. We now have to correct the fact that we can apply the measurements with only some exponentially good accuracy but not exactly. The above discussion showed that in the perfect measurement case, the output is within ϵ trace distance from the desired density matrix of the final ground-state. To analyze the nonperfect case, we keep track of the entire system (recall that O_H adds ancilla qubits to operate on). We now compare the overall state of the system after the application of O_H to the overall state after the application of an ideal O_H which simulates a perfect measurement. By Lemma 1.1, the Euclidean distance between the states is arbitrarily small. Summing up all these errors over polynomially many O_H's still results in an arbitrarily small distance from the state in the case of perfect measurements. When considering the reduced state to the original subspace, this results in a state which is arbitrarily close (in trace distance) to the state in the case of perfect measurements. □

5. ASG for Markov chain states.

Finally, we show how to use our techniques to generate interesting quantum states related to Markov chains and approximate counting algorithms. We give some Markov chain background below; for more background, see [38] and the references therein. For background regarding approximate counting, see [33].

5.1. Markov chain background.

We consider a Markov chain on a graph, with nodes indexed by n bit strings. The bits strings are called *states* (not to be confused with quantum states). The Markov chain is characterized by a matrix M operating over the state space. The matrix M has eigenvalues between -1 and 1. Under mild conditions on M (namely, M is connected and aperiodic), for any p an initial probability distribution over the state space, the limit $\lim_{t\to\infty} pM^t = \pi$ exists, π is called the limiting distribution, it is independent of p, and it is a left eigenvector of M with eigenvalue 1. Also, $\pi_i > 0$ for all i.

A Markov chain is *reversible* if for the limiting distribution π, for every i and j, it holds that $M[i,j] \cdot \pi_i = M[j,i] \cdot \pi_j$, i.e., if every directed edge has the same weight under the stationary distribution. We note that any symmetric Markov chain M is reversible, and its limiting distribution must be the uniform distribution.

A Markov chain is said to be *rapidly mixing* if starting from any initial distribution, the distribution after $poly(n, \frac{1}{\epsilon})$ steps is within ϵ total variation distance from the limiting distribution π. A reversible Markov chain is rapidly mixing iff its second eigenvalue gap, namely, the difference between the first and second largest (in absolute value) eigenvalues, is nonnegligible, namely, bounded from below by $1/poly(n)$.

For the sake of simplicity, we restrict our attention in this paper to Markov chains with nonnegative eigenvalues. This is standardly done, by adding self-loops with probability 1/2, and makes sure that no absolute values are needed in the definition of the eigenvalue gap.

5.2. Reversible Markov chains and Hamiltonians.

For a reversible Markov chain M with a limiting distribution π, we define

$$H_M = I - \mathrm{Diag}(\sqrt{\pi}) \cdot M \cdot \mathrm{Diag}\left(\frac{1}{\sqrt{\pi}}\right),$$

where $\mathrm{Diag}(\sqrt{\pi})$ is the diagonal matrix with $\sqrt{\pi_i}$ in its diagonal ith entry. Similarly, $\mathrm{Diag}(\frac{1}{\sqrt{\pi}})$ has $\frac{1}{\sqrt{\pi_j}}$ over its diagonal.

A direct calculation shows that M is reversible iff H_M is symmetric. Thus, for a reversible Markov chain, we denote by H_M the *Hamiltonian corresponding to M*. The properties of H_M and M are very much related.

Claim 5.1. *Suppose M is a reversible Markov chain with limiting distribution π. Then*

- *H_M is a Hamiltonian with $\|H_M\| \leq 2$;*
- *the spectral gap of H_M equals the second eigenvalue gap of M.*

Let us define $|\pi\rangle \stackrel{\text{def}}{=} \sum_i \sqrt{\pi_i}\,|i\rangle$. Then

- *the ground-state $\alpha(H_M)$ of H_M is $|\pi\rangle$ with ground-value* 0.

Proof. If M is reversible, H_M is Hermitian and hence has an eigenvector basis. It is easy to see that v is an eigenvector of H_M with eigenvalue λ iff $\mathrm{Diag}(\sqrt{\pi})^{-1}v$ is an eigenvector of M with eigenvalue $1-\lambda$. Also, $v^t\mathrm{Diag}(\sqrt{\pi})$ is a left eigenvector of M with the same eigenvalue. It follows that if the eigenvalues of H_M are $\{\lambda_r\}$, then the eigenvalues of M are $\{1-\lambda_r\}$. M is a reversible Markov chain and therefore has eigenvalues between -1 and 1, and the first two items of the claim follow. If we denote $v = (\pi_1, \ldots, \pi_n)$ to be the (unique) left eigenvector of M with eigenvalue 1, then $\mathrm{Diag}(\sqrt{\pi})^{-1}v$ is the (unique) eigenvector of H_M with eigenvalue 0. All other eigenvectors of M have eigenvalues strictly smaller than 1, and so all other eigenvectors of H_M have eigenvalues strictly larger than 0. It follows that $|\pi\rangle = \mathrm{Diag}(\sqrt{\pi})^{-1}v$ is the unique ground-state of H_M with ground-value 0. □

This gives a direct connection between Hamiltonians, spectral gaps, and ground-states on one hand and rapidly mixing reversible Markov chains and limiting distributions on the other hand.

5.3. Simulating H_M.

Even if M is sparse and explicit, its corresponding Hamiltonian H_M might not be explicit, because for approximating $H_M[i,j] = -\sqrt{\frac{\pi_i}{\pi_j}}M[i,j]$ we need to be able to approximate $\frac{\pi_i}{\pi_j}$. Special cases are easier. For example, it is easy to compute $\frac{\pi_i}{\pi_j}$ when M is symmetric. For general reversible Markov chains we define the following.

Definition 5.1 (strongly samplable). *A reversible Markov chain on the state space Ω with limiting distribution π is called strongly samplable if it is*

- *sparse and explicit, and,*
- *given $i, j \in \Omega$, there is an efficient way to approximate $\frac{\pi_i}{\pi_j}$.*

Sparseness and explicitness hold in most interesting Markov chains. The second requirement is more restrictive. Still, we note that it holds in many interesting cases such as all Metropolis algorithms (see [29]). As $H_M[i,j] = -\sqrt{\frac{\pi_i}{\pi_j}} M[i,j]$ for $i \neq j$, we see that if M is strongly samplable, then H_M is sparse and explicit. As H_M has bounded norm, we can use the sparse Hamiltonian lemma (Lemma 1.4). This implies the following corollary.

COROLLARY 5.1. *If a Markov chain M is strongly samplable, then H_M is simulatable.*

5.4. From Markov chains to quantum state generation. We now consider sequences of Markov chains and define the following.

DEFINITION 5.2 (slowly varying Markov chains). *Let $\{M_t^n\}_{t=1}^{T=T(n)}$ be a sequence of Markov chains on Ω_n, $|\Omega_n| = N = 2^n$. Let π_t^n be the limiting distribution of M_t^n. We say that the sequence is* slowly varying *if there exists some $c > 0$ such that for all large enough n, for all $1 \leq t \leq T(n)$ $|\pi_t^n - \pi_{t+1}^n| \leq 1 - 1/n^c$.*

We can now state Theorem 1.3 precisely.

THEOREM 1.3 (formal). *Let $\{M_t^n\}_{t=1}^{T}$ be a slowly varying sequence of strongly samplable Markov chains which are all rapidly mixing, and let π_t^n be their corresponding limiting distributions. Then if there exists an efficient quantum state generator for $|\pi_0^n\rangle$, then there exists an efficient quantum state generator for $|\pi_{T(n)}^n\rangle$.*

Proof. By Corollary 5.1, the Hamiltonians $H_{M_t^n}$ are simulatable. Also, Claim 5.1 implies that $\|H_{M_t^n}\| \leq 2$ and that the ground-values of all these Hamiltonians are 0. Also, the Markov chains in the sequence are rapidly mixing, which means that they have nonnegligible spectral gaps, say $\geq \frac{1}{n^b}$, for some $b > 0$. This means that $\Delta(H_{M_t^n}) \geq \frac{1}{n^b}$. To complete the proof, we show that the inner product between the ground-states of subsequent Hamiltonians is nonnegligible. Indeed,

$$\begin{aligned}\langle \alpha(H_{M_t}) | \alpha(H_{M_{t+1}}) \rangle &= \langle \pi_t | \pi_{t+1} \rangle \\ &= \sum_i \sqrt{\pi_t(i)\pi_{t+1}(i)} \geq 1 - |\pi_t - \pi_{t+1}| \geq \frac{1}{n^c},\end{aligned}$$

where the second to last inequality follows from Fact 2.1. The theorem then follows from the jagged adiabatic path lemma (Lemma 1.3). □

Essentially all Markov chains that are used in approximate counting algorithms that we are aware of meet the criteria of the theorem. The following is a partial list of states we can generate by applying this theorem. The citations refer to the approximate counting algorithms that we use as the basis for the quantum state generation algorithm:

1. uniform superposition over all perfect matchings of a given bipartite graph [32],
2. all lattice points contained in a high-dimensional convex body satisfying the conditions of [9],
3. various Gibbs distributions over rapidly mixing Markov chains using the Metropolis filter [38],
4. log-concave distributions [9],
5. all linear extensions of a given partial order [12].

Remark 5.1. We note that the second requirement in the definition of strongly samplable Markov chains (Definition 5.1) is crucial. If this requirement can be relaxed, and one can prove Theorem 1.3 without it, then these techniques could have been used to solve the QS problem related to the graph isomorphism problem and thus to give a quantum algorithm for graph isomorphism.

We illustrate our technique with the example of how to Qsample all perfect matchings of a given bipartite graph.

5.5. An example: All perfect matchings of a bipartite graph. In this section we rely heavily on the work of Sinclair, Jerrum, and Vigoda [32], who recently showed how to efficiently approximate the permanent of any matrix with nonnegative entries. It is well known that this can be done if one can efficiently sample a random perfect matching of a given input bipartite graph. Sinclair et al. achieve the latter using a sequence of Markov chains $M_1, \ldots, M_T$ on the set of matchings of a bipartite graph. The details of this work are far too involved to fully explain here, and we refer the interested reader to [32] for further details.

In a nutshell, the idea in [32] is to start with M_1, which is 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. Since [32] is interested in a given input bipartite graph which is a subgraph of the complete bipartite graph, they assign weights to the edges such that weights on the edges that do not participate in the input graph are slowly decreasing as we move from M_1 to M_T until their weights in the final Markov chain M_T practically vanish. The weights of the edges are updated from Markov chain M_t to the next M_{t+1} using data that is collected from running the Markov chain M_t with the current set of weights.

The final Markov chain M_T with the final set of 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.

We would like to apply Theorem 1.3 in order to solve the quantum sampling problem for the limiting distribution of M_T, namely, to generate the coherent superposition over perfect and near perfect matchings with the correct weights.

We need to check that the conditions of the theorem hold. It is easy to check that the Markov chains M_t being used in [32] are all strongly samplable, since they are Metropolis chains, and so the corresponding Hamiltonians are simulatable by Corollary 5.1. Moreover, the sequence of Markov chains is slowly varying. To apply Theorem 1.3, it remains to show that we can Qsample the limiting distribution of M_1, the first Markov chain in the sequence.

The limiting distribution of the initial Markov chain M_1 is a distribution over all perfect and near perfect matchings in the complete bipartite graph, with each near perfect matching having weight n times that of a perfect matching, where n is the number of nodes of the given graph. To generate this superposition we do the following:

- We generate in the first register $\sum_{\pi \in S_n} |m_\pi\rangle$, where m_π is the matching on the complete bipartite graph induced by $\pi \in S_n$. We can efficiently generate this state because we can generate a superposition over all permutations in S_n, and there is an easy computation from a permutation to a perfect matching in a complete bipartite graph, and vice versa.
- In the second register, we generate the state $|0\rangle + \sqrt{n} \sum_{i=1}^{n} |i\rangle$ normalized on a $\log(n)$-dimensional register. This can be done efficiently because any unitary transformation on $\log(n)$ qubits can be performed efficiently.
- We apply a transformation that maps $|m, i\rangle$ to $|0, m\rangle$ when $i = 0$, and to

$|0, m - \{e_i\}\rangle$ for $i > 0$, where $m - \{e_i\}$ is the matching m minus the $i'th$ edge in the matching. There is an easy computation from $m - \{e_i\}$ to m, i, and vice versa, and so this transformation can be done efficiently. We are now at the desired state.

Thus we can apply Theorem 1.3 to generate the limiting distribution of the final Markov chain. We then measure to see whether or not the matching is perfect, and with nonnegligible probability we project the state onto the uniform distribution over all perfect matchings of the given graph.

6. The basic tools.

6.1. A lemma about ground-states of close Hamiltonians.

CLAIM 6.1 (Claim 1.1 repeated). *Let A, B be two Hamiltonians of equal dimensions such that $\|A - B\| \leq \eta$. Moreover, assume that A, B have spectral gaps bounded from below: $\Delta(A), \Delta(B) \geq \Delta$. Then $|\langle\alpha(A)|\alpha(B)\rangle| \geq 1 - \frac{4\eta^2}{\Delta^2}$.*

Proof. Adding $g \cdot I$ to both matrices, for any constant g, does not affect the spectral norm of the difference, the spectral gaps, or the inner product between the ground-states. We can therefore assume without loss of generality that A and B are positive, A's ground-value is 0, and B's ground-value, denoted by λ_B, is larger than 0.

Since $\lambda_B = \min_{v:|v|=1} |Bv|$, we have $\lambda_B \leq |B\alpha(A)|$. Also,

$$|B\alpha(A)| \leq |A\alpha(A)| + |(B - A)\alpha(A)| \ \leq \ \eta, \tag{6.1}$$

and so $\lambda_B \leq \eta$. We now express $\alpha(A) = c\alpha(B) + c^\perp\alpha(B)^\perp$, with $\alpha(B)^\perp \perp \alpha(B)$. Then

$$\begin{aligned} |B\alpha(A)| &= |cB\alpha(B) + c^\perp B\alpha(B)^\perp| \\ &\geq |c^\perp| \cdot \Delta - |c| \cdot \lambda_B \\ &\geq |c^\perp| \cdot \Delta - |c| \cdot \eta. \end{aligned} \tag{6.2}$$

Equations (6.1) and (6.2) together imply that $\eta \geq |c^\perp| \cdot \Delta - |c| \cdot \eta$. We see that $|c| \geq \frac{\Delta}{\eta}|c^\perp| - 1$. Equivalently, $\sqrt{1 - |c^\perp|^2} \geq \frac{\Delta}{\eta}|c^\perp| - 1$.

Denoting $r = \frac{\Delta}{\eta} > 0$, we see that if the right-hand side is negative, then $|c^\perp| \leq \frac{1}{r}$; otherwise, solving the inequality we get

$$|c^\perp| \leq \frac{2r}{r^2 + 1} \leq \frac{2r}{r^2} = \frac{2}{r}.$$

We get $|\langle\alpha(A)|\alpha(B)\rangle| = |c| = \sqrt{1 - |c^\perp|^2} \geq 1 - \frac{4}{r^2} = 1 - \frac{4\eta^2}{\Delta^2}$ as desired. □

6.2. The spectral gap of a convex combination of projections. We now prove the basic but useful Claim 1.2 regarding the convex combination of two projections. Recall that for a vector $|\alpha\rangle$, the Hamiltonian $\Pi_\alpha = I - |\alpha\rangle\langle\alpha|$ is the projection onto the subspace orthogonal to α.

CLAIM 6.2 (Claim 1.2 repeated). *Let $|\alpha\rangle, |\beta\rangle$ be two vectors in some Hilbert space. For any convex combination $H_\eta = (1 - \eta)\Pi_\alpha + \eta\Pi_\beta$, $\eta \in [0, 1]$, we have $\Delta(H_\eta) \geq |\langle\alpha|\beta\rangle|$.*

Proof. We observe that the problem is two-dimensional. We write $|\beta\rangle = a|\alpha\rangle + b|\alpha^\perp\rangle$. We now express the matrix H_η in a basis which contains $|\alpha\rangle$ and $|\alpha^\perp\rangle$. The eigenvalues of this matrix are all 1, except for a two-dimensional subspace, where the matrix is exactly

$$\begin{pmatrix} \eta|a|^2 + (1 - \eta) & \eta ab^* \\ \eta a^*b & \eta|b|^2 \end{pmatrix}.$$

Diagonalizing this matrix we find that the spectral gap is exactly $\sqrt{1-4(1-\eta)\eta|b|^2}$, which is minimized for $\eta = 1/2$ where it is exactly $|a| = |\langle\alpha|\beta\rangle|$. □

6.3. The Hamiltonian-to-measurement and projection lemmas. Consider a simulatable Hamiltonian H with ground-state $|\alpha\rangle$. It is sometimes desirable to apply a measurement in the basis of eigenstates of the Hamiltonian. The Hamiltonian-to-measurement lemma provides an *approximation* of this procedure in the case where the spectral gap of the Hamiltonian is nonnegligible. The lemma is based on Kitaev's phase estimation procedure, which we now recall.

LEMMA 6.1 (phase estimation, adapted from [35]; see also [46, section 5.2]). *Let U be a unitary transformation on n qubits, and assume U can be implemented by a poly(n)-size circuit. Let $\epsilon > 0$. Then there exists a quantum procedure Q running in time poly$(n, \frac{1}{\epsilon})$ which on input v, that is an eigenvector of U with eigenvalue $e^{i\lambda}$, outputs $Q\,|v,0\rangle = |v,\psi\rangle$ such that the following conditions hold. $|\psi\rangle$ is exponentially close in fidelity to another vector $|\psi'\rangle$, $F(|\psi\rangle, |\psi'\rangle) \geq 1 - 2^{-\Omega(n)}$, where $|\psi'\rangle = |\lambda'\rangle \otimes |0\rangle$, and λ' is a real number such that $|\lambda' - \lambda| \leq \epsilon$. The right register in $|\psi'\rangle$ consists of the ancilla bits of the algorithm.*

We can now proceed to proving Lemma 1.1.

LEMMA 6.2 (Lemma 1.1 repeated). *Assume H is a simulatable Hamiltonian on n qubits. For any constant d, there exists a poly$(n, \frac{1}{\Delta(H)})$-size circuit O_H which takes $|\alpha(H)\rangle$ to $|\alpha(H)\rangle \otimes |\gamma\rangle$, and for any eigenstate of H $|\alpha^\perp\rangle$ orthogonal to the ground-state, $O_H|\alpha^\perp\rangle = |\alpha^\perp\rangle \otimes |\beta(\alpha^\perp)\rangle$, where $|\langle\gamma|\beta(\alpha^\perp)\rangle| \leq O(n^{-d})$.*

Proof. We would like to apply the phase estimation algorithm for the unitary matrix e^{iH}, with $\epsilon = \Delta(H)/2$. Suppose we could exactly simulate H; namely, we could apply e^{iH} exactly. In this case, by the above lemma, Kitaev's phase estimation algorithm does the following. For an input state which is a ground-state of H, the output state is exponentially close to a vector of the form $|\lambda'\rangle \otimes |0\rangle$ for $|\lambda' - \lambda_0| < \Delta(H)/2$, where λ_0 is the ground-value. For an input state which is any other eigenvector, the output state is exponentially close to a vector of the form $|\lambda''\rangle \otimes |0\rangle$ for $|\lambda'' - \lambda_j| < \Delta(H)/2$. For a superposition of eigenvectors not including the ground-state, the output is a superposition of such vectors. The two output vectors are therefore exponentially close to being orthogonal, as required.

We now need to show how to apply the phase estimation algorithm. To do that, we recall that H is simulatable, and so we can ζ-approximate e^{-iH} in time polynomial in n and $\frac{1}{\zeta}$. We pick ζ to be small enough, but still inverse polynomial in n and $\frac{1}{\epsilon}$, as follows. Let m be the number of times that e^{iH} is applied in the phase estimation algorithm, with the above ϵ. We set ζ to be n^{-d}/m. We then apply the above phase estimation algorithm, where every time we need to apply e^{iH}, we simulate it with this accuracy. The accumulated error due to the inaccuracies in the simulation of H is at most n^{-d}. The output states are therefore within n^{-d} Euclidean distance from the correct ones. This means that the inner product discussed above (between the output state for an input which is a ground-state, and the output state in case the input is orthogonal) is also $O(n^{-d})$. □

We now prove the Hamiltonian-to-projection lemma (Lemma 1.2). Consider a simulatable Hamiltonian H whose ground-state is α. This time we want to simulate the Hamiltonian $\Pi_{\alpha(H)}$, rather than the original Hamiltonian H. The Hamiltonian-to-projection lemma provides a way to do this, provided the spectral gap of H is nonnegligible.

LEMMA 6.3 (Lemma 1.2 repeated). *Assume H is a simulatable Hamiltonian on n qubits, with nonnegligible spectral gap $\Delta(H) \geq 1/n^c$ for some constant $c > 0$ and* *with a known ground-value. Then the Hamiltonian* $\Pi_{\alpha(H)}$ *is simulatable.*

Proof. As before, let us start with the assumption that we can apply e^{iH} efficiently and perfectly. We do the following:

- Apply Kitaev's phase estimation algorithm for the unitary matrix e^{iH}, using $\epsilon = \Delta(H)/2$. Now, given the output λ', we can write down one bit of information on an extra qubit: whether an input eigenstate of H is the ground-state or orthogonal to it (it is here that we use the fact that we know the ground-value).
- Apply a phase shift of value e^{-it} to this extra qubit, conditioned that it is in the state $|1\rangle$ (if it is $|0\rangle$, 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 $\Pi_{\alpha(H)}$.
- Finally, to erase the extra data written down, we reverse the first step and uncalculate the information written on the qubits again using Kitaev's phase estimation algorithm.

This procedure computes the desired transformation $e^{i\Pi_{\alpha(H)}t}$ on any vector, with $1 - 2^{-O(n)}$ fidelity.

As in the proof of Lemma 1.1, we now need to take into account the fact that we cannot apply e^{iH} exactly, but we can simulate H only approximately. We approximate each of the applications of e^{iH} to within ζ. To determine ζ, we note that the number of times m we apply e^{iH} in the above procedure is $poly(n, \frac{1}{\Delta(H)})$. To get an overall error of size ζ', we simply fix ζ to be $\zeta'/2m$. The overall procedure is then polynomial in $\zeta', n, \frac{1}{\Delta(H)}$. □

Remark 6.1. We remark that in the two previous lemmas there is nothing special about the ground-state of the Hamiltonian. The same techniques work for measuring or projecting onto any eigenstate with a *known* eigenvalue, that is, separated from all other eigenvalues.

6.4. The jagged adiabatic path lemma. Next, we consider the question of which paths in the Hamiltonian space guarantee nonnegligible spectral gaps. Figure 6.1 gives an example of two Hamiltonians H_1, H_2 with nonnegligible spectral gaps that can be connected through a jagged line but not through a direct line.

The jagged adiabatic path lemma (Lemma 1.3) provides a way, in a more specific case, to connect Hamiltonians such that the spectral gaps along the path are always nonnegligible. The additional condition in Lemma 1.3 is that the ground-state of H_j is close to the ground-state of H_{j+1} (this condition is not fulfilled in the example of Figure 6.1). It may seem natural that in this case the way to prove the jagged adiabatic path lemma is to connect each pair of Hamiltonians H_j and H_{j+1} by a straight line. However, again this does not work. To see this consider

$$H_1 = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad H_2 = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

where λ_1 and λ_2 are the eigenvalues of

$$\begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

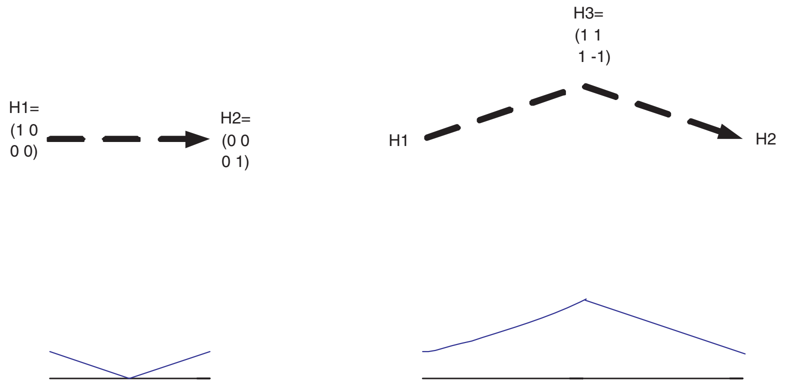

FIG. 6.1. *In the left side of the drawing, we see two Hamiltonians H_1 and H_2 connected by a straight line and the spectral gaps along that line. In the right side of the drawing, we see the same two Hamiltonians H_1 and H_2 connected through a jagged line that goes through a third connecting Hamiltonian H_3 in the middle and the spectral gaps along that jagged path. Note that on the left the spectral gap becomes zero in the middle, while on the right it is always larger than one.*

(they are about 0.3 and 1.7). Let us say the matrices are represented in the orthonormal basis $\{v_1, \ldots, v_4\}$. Then H_1 has $|v_4\rangle$ as a unique ground-state with ground-value 0, and H_2 has $\frac{1}{\sqrt{2}}[|v_3\rangle - |v_4\rangle]$ as a unique ground-state with ground-value 0. It is now easy to check that in $\frac{1}{2}[H_1 + H_2]$ all eigenspaces have dimension two, and no eigenvalue is separated from the others.

The problem with connecting H_1 and H_2 by a line stems from the fact that H_i may behave arbitrarily outside the subspace containing their ground-states. This also hints at the solution. As we are interested only in the ground-states, let us project each H_i onto the subspace orthogonal to its ground-state; i.e., if H is a Hamiltonian, let us define Π_H to be the projection onto the space orthogonal to the ground-state of H. We then replace the sequence $\{H_j\}$ with the sequence of Hamiltonians $\{\Pi_{H_j}\}$, and we connect two neighboring *projections* by a line.

LEMMA 6.4 (Lemma 1.3 repeated). *Let $\{H_j\}_{j=1}^{T=poly(n)}$ be a sequence of bounded norm, simulatable Hamiltonians on n qubits, with nonnegligible spectral gaps, $\Delta(H_j) \geq n^{-c}$, and with known ground-values, such that the inner product between the unique ground-states $\alpha(H_j), \alpha(H_{j+1})$ is at least n^{-c} for all j. Then there exists an adiabatic state generator with $\alpha(H_0)$ as its initial state and $\alpha(H_T)$ as its final state. In particular there exists an efficient quantum algorithm that takes $\alpha(H_0)$ to within arbitrarily small distance from $\alpha(H_T)$.*

Proof. We replace the sequence $\{H_j\}$ with the sequence of Hamiltonians $\{\Pi_{H_j}\}$, and we connect two neighboring projections by a line. We claim the following:

- As projections, Π_{H_j} have bounded norms, $\|\Pi_{H_j}\| \leq 1$. As the Hamiltonians on the path connecting these projections are convex combinations of the projections, they also have bounded norm.
- We proved in Lemma 1.2 that the fact that H_j is simulatable and has a known ground-value implies that Π_{H_j} is also simulatable. It follows that the Hamiltonians on the path connecting these projections, which are convex combinations of the projections and are of polynomially bounded norms, are also simulatable (see section 4). This means that all the Hamiltonians on the path are simulatable.

- The projections Π_{H_i} have the same ground-states as the Hamiltonians H_i, and as a result each two neighboring projections have nonnegligible inner product between their ground-states. In Claim 1.2 we showed that this implies the Hamiltonians on the line connecting Π_{H_j} and $\Pi_{H_{j+1}}$ have nonnegligible large spectral gaps. Notice that this step is possible only when working with the projections Π_H but not with the Hamiltonians H_i themselves.

Thus, we can define an adiabatic state generator in which the initial state is $\alpha(H_0)$ and the final state is $\alpha(H_T)$. The interval $[0,1]$ of the rescaled time s is divided equally between the different steps of the jagged path; when we move from one projection to the next, s increases by $1/T$ (recall that T is the number of Hamiltonians we connect). Note that this implies that the maximal norm of the first derivative of the Hamiltonians in the adiabatic state generator, which is denoted by η in the definition of ASG, is $O(T)$. The minimal spectral gap in the adiabatic state generator, which we denoted by Δ, is bounded from below by an inverse polynomial in n, by the arguments above. This gives us an adiabatic state generator which takes time $T_\epsilon = \frac{\eta^2}{\epsilon\Delta^2} = poly(T, n, \frac{1}{\epsilon})$. We can now apply Theorem 1.2 and get an efficient quantum algorithm that takes $\alpha(H_0) = \alpha(\Pi_{H_0})$ to within arbitrarily small distance from $\alpha(H_T) = \alpha(\Pi_{H_T})$. □

6.5. The sparse Hamiltonian lemma. The sparse Hamiltonian lemma (Lemma 1.4) gives fairly general conditions for a Hamiltonian to be simulatable: the Hamiltonian need only be sparse and explicit.

The main idea of the proof 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.

6.5.1. Decomposition of *H* as a sum of block diagonal matrices with 2 × 2 blocks.

Definition 6.1 (combinatorial block). *We say that A is combinatorially block diagonal if we can decompose the set of rows of A by $ROWS(A) = \bigcup_{b=1}^{B} R_b$, where we require that $A(i,j) \neq 0$ implies that there exists b such that both $i \in R_b$ and $j \in R_b$.*

We say that A is 2×2 combinatorially block diagonal if, for every b, either $|R_b| = 1$ or $|R_b| = 2$.

Claim 6.3 (decomposition lemma). *Let H be a sparse explicit Hamiltonian over n qubits, with at most D nonzero 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 sparse explicit Hamiltonian over n qubits, and*
- *each H_m is 2×2 combinatorially block diagonal.*

Proof. We color all the entries of H with $(D+1)^2 n^6$ colors as follows. For $(i,j) \in [N] \times [N]$ and $i \leq j$ (i.e., (i,j) is an upper-diagonal entry), we define

$$col_H(i,j) = (k\ ,\ i \bmod k\ ,\ j \bmod k\ ,\ rindex_H(i,j)\ ,\ cindex_H(i,j)),$$

where we have the following:

- If $i = j$, we set $k = 1$; otherwise, we let k be the first integer in the range $[2 \dots n^2]$ such that $i \neq j \pmod{k}$. We know there must be such a k (for the product of all primes smaller than n^2 is larger than 2^n, and by the Chinese remainder theorem two numbers that have the same modula are equal).

- $rindex_H(i,j) = 0$ if $H_{i,j} = 0$, and otherwise it is the index of $H_{i,j}$ in the list of all nonzero elements in the ith row of H. We define $cindex_H(i,j)$ similarly, using the columns.

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

The Hamiltonian's entries are decomposed by their 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 explicit and sparse, there is a simple $poly(n)$-time classical algorithm computing the coloring $col_H(i,j)$, and so each H_m is also explicit and sparse. It is left to show that it is 2×2 combinatorially block diagonal.

Indeed, fix a color m and consider H_m. For every nonzero element (i_b, j_b) of H_m, we define a block. If $i_b = j_b$, then we set $R_b = \{i_b\}$, while if $i_b \neq j_b$, then we set $R_b = \{i_b, j_b\}$. Say $i_b \neq j_b$ (the $i_b = j_b$ case is similar and simpler). We know that the elements (i_b, j_b) and (j_b, i_b) are colored by the same color m, and suppose $m = (k, i_b \bmod k, j_b \bmod k, rindex_H(i_b, j_b), cindex_H(i_b, j_b))$. To see that $R_b = \{i_b, j_b\}$ is a 2×2 block, we need to show that there are no other elements colored by m on the i_bth and j_bth rows and columns. As the color m contains the row-index and column-index of (i_b, j_b), it must be that (i_b, j_b) is the only nonzero element in H_m from that row or column. Furthermore, all elements (j_b, a) on the j_bth row have $j_b \bmod k \neq i_b \bmod k$ and therefore are not colored by m. A similar argument shows no element on the i_bth row is colored by m, and the claim follows (see Figure 6.2). □

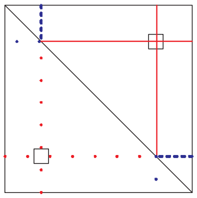

FIG. 6.2. *In the upper diagonal side of the matrix H_m, the row and column of (i_b, j_b) are empty because the color m contains the row-index and column-index of (i,j), and the j_bth row and i_bth column are empty because m contains k, $i \bmod k$, $j \bmod k$, and $i \bmod k \neq j \bmod k$. The lower 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 6.4. *For every m, $\|H_m\| \leq \|H\|$.*

Proof. Fix an m. H_m is 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 combinatorial block is

$$\begin{pmatrix} 0 & A_{k,\ell} \\ A^*_{k,\ell} & 0 \end{pmatrix}$$

for some k, ℓ, and it has norm $|A_{k,\ell}|$.

It follows that $\max_m \|H_m\| = \max_{k,\ell} |H_{k,\ell}|$. On the other hand, $\|H\| \geq \max_{k,\ell} |H_{k,\ell}|$ (see section 2.2). The proof follows. □

6.5.2. 2×2 combinatorially block diagonal matrices are simulatable.

CLAIM 6.5. *Every 2×2 block diagonal, explicit Hamiltonian A is simulatable to within arbitrary polynomial approximation.*

Proof. The proof is standard, but we include it for completeness.

Let $t > 0$, and let $\alpha > 0$ be an accuracy parameter.

The circuit: A is 2×2 combinatorially block diagonal. It therefore follows that A's action on a given basis state $|k\rangle$ is captured by a 2×2 unitary transformation U_k. Formally, given k, say $|k\rangle$ belongs to a 2×2 combinatorial block $\{k, \ell\}$ in A. We set $b_k = 2$ (for a 2×2 block) and $\min_k = \min(k, \ell)$, $\max_k = \max(k, \ell)$ (for the subspace to which k belongs). We then set A_k to be the part of A relevant to this subspace,

$$A_k = \begin{pmatrix} A_{min_k, min_k} & A_{min_k, max_k} \\ A_{max_k, min_k} & A_{max_k, max_k} \end{pmatrix},$$

and $U_k = e^{-itA_k}$. If $|k\rangle$ belongs to a 1×1 block, we similarly define $b_k = 1$, $\min_k = \max_k = k$, $A_k = (A_{k,k})$, and $U_k = (e^{-itA_k})$.

Our approximated circuit simulates this behavior. We need two transformations. We define

$$\widetilde{T_1} : |k, 0\rangle \rightarrow \left| b_k, \min_k, \max_k, \widetilde{A_k}, \widetilde{U_k}, k \right\rangle,$$

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 $\Theta(\alpha)$ accuracy such that $\|U_k - \widetilde{U_k}\| \leq 4\|U_k - \widetilde{U_k}\|_\infty \leq \alpha$. Having $\widetilde{U_k}, \min_k, \max_k, k$ written down, we can simulate the action of $\widetilde{U_k}$. We can therefore have an efficient unitary transformation T_2:

$$\widetilde{T_2} : \left| \widetilde{U_k}, \min_k, \max_k \right\rangle |k\rangle = \left| \widetilde{U_k}, \min_k, \max_k \right\rangle \left| \widetilde{U_k} k \right\rangle$$

for $|\widetilde{U_k} k\rangle \in Span\{\min_k, \max_k\}$. We can similarly define T_2, which applies U_k on its input, $T_2 = |A, \min, \max, k\rangle = |A, \min, \max, U_k k\rangle$.

Our algorithm applies $\widetilde{T_1}$, followed by $\widetilde{T_2}$ and then $\widetilde{T_1}^{-1}$ for cleanup.

Correctness: Let us denote $X = e^{-itA} - T_1^{-1} T_2 T_1$. Our goal is to show that $\|X\| \leq \alpha$. We notice that X is also 2×2 combinatorially 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 $Span\{|\min_k\rangle, |\max_k\rangle\}$. On this subspace, we have a 2×2 operator $X = e^{-itA_k} - \widetilde{T_1}^{-1} \widetilde{T_2} \widetilde{T_1}$. Also, $e^{-itA_k} = \widetilde{T_1} T_2 \widetilde{T_1}^{-1}$. It follows that $\|X\| = \|T_2 - \widetilde{T_2}\| = \|U_k - \widetilde{U_k}\| \leq \alpha$. □

Remark 6.2. We proved the claim for matrices with 2×2 combinatorial blocks. A similar claim applies for matrices with $m \times m$ combinatorial blocks, with the same proof technique, as long as m is polynomial in n.

6.5.3. Proving the sparse Hamiltonian lemma.

We now prove the sparse Hamiltonian lemma.

LEMMA 6.5 (Lemma 1.4 repeated). *If H is an explicit and sparse Hamiltonian on n qubits and $\|H\| \leq poly(n)$, then H is simulatable.*

Proof. Let H be row-sparse with $D \leq poly(n)$ nonzero elements in each row, and let $\|H\| = \Lambda \leq poly(n)$. 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 6.3, $M \leq (D+1)^2 n^6 \leq poly(n)$. By Claim 6.5, for every $\delta > 0$ we can simulate $e^{-i\delta H_m}$ to within $\frac{\alpha}{2Mt/\delta}$ accuracy in time

$poly(n, t, M, \frac{1}{\delta}, \frac{1}{\alpha})$. It follows that we can approximate U_δ (see section 4.1.1) to within $\frac{\alpha}{t/\delta}$ accuracy and $U_\delta^{\lfloor \frac{t}{2\delta} \rfloor}$ to within $\frac{\alpha}{2}$ accuracy.

Also, Corollary 4.1 ensures us that $U_\delta^{\lfloor \frac{t}{2\delta} \rfloor}$ is $O(M\widetilde{\Lambda}\delta + M\widetilde{\Lambda}^3 t\delta^2)$ close to e^{-itH}, where $\widetilde{\Lambda} = \max_{k \le \ell} \| \sum_{i=k}^{\ell} H_i \| \le M\|H\|$ (because we saw that for every m, $\|H_m\| \le \|H\|$). Picking δ small enough (inverse polynomial in M, Λ, t), we see that $U_\delta^{\lfloor \frac{t}{2\delta} \rfloor}$ is $\frac{\alpha}{2}$ close to e^{-itH}. Altogether, our approximation is α close to e^{-itH}. It follows that our approximation has circuit size bounded by $poly(n, t, M, \frac{1}{\delta}, \frac{1}{\alpha}) = poly(n, t, \frac{1}{\alpha})$. □

Appendix A. QS instances: Specific problems in SZK.

We saw in the introduction an example of a QS problem, the solution of which implies an efficient quantum algorithm for graph isomorphism. Here we give three more examples of problems in SZK which are of particular relevance for quantum algorithms: discrete log, quadratic residuosity, and a gap version of closest vector in a lattice. An efficient QS algorithm for the lattice problem is a major open problem.

We also remind the reader that the existence of the reduction of the problems we consider below to certain instances of QS problems already follows from Theorem 1.1. Here we do a direct reduction and get simple QS instances sufficient for solving the above problems.

A.1. A promise problem equivalent to discrete log.

The problem: Goldreich and Kushilevitz [25] define the promise problem DLP$_c$ as

Input: A prime p, a generator g of Z_p^*, and an input $y \in Z_p^*$.

Promise:

- Yes: $x = \log_g(y)$ is in $[1, cp]$.
- No: $x = \log_g(y)$ is in $[\frac{p}{2} + 1, \frac{p}{2} + cp]$.

Goldreich and Kushilevitz [25] prove that the discrete-log problem is reducible to DLP$_c$ for every $0 < c < 1/2$. They also prove that DLP$_c$ has a perfect zero knowledge proof if $0 < c \le 1/6$. We take $c = 1/6$ and show how to solve DLP$_{1/6}$, given a certain QS algorithm.

The reduction: We assume we can solve the QS problem for the circuit $C_{y,k} = C_{p,g,y,k}$ that computes $C_{y,k}(i) = y \cdot g^i \pmod{p}$ for $0 \le i < 2^k$. The algorithm generates the states $\left|C_{g^{p/2+1}, \lfloor \log(p) \rfloor - 1}\right\rangle, \left|C_{y, \lfloor \log(p) \rfloor - 3}\right\rangle$ and proceeds as in Theorem 1.1.

Correctness:

We have

$$\left|C_{g^{p/2+1}, \lfloor \log(p) \rfloor - 1}\right\rangle = \frac{1}{\sqrt{t}} \sum_{i=0}^{t-1} \left|g^{p/2+i}\right\rangle, \tag{A.1}$$

where t is the largest power of 2 smaller than $p/2$. Also, as $y = g^x$ we have

$$\left|C_{y, \lfloor \log(p) \rfloor - 3}\right\rangle = \frac{1}{\sqrt{t'}} \sum_{i=0}^{t'-1} \left|g^{x+i}\right\rangle, \tag{A.2}$$

where t' is the largest power of 2 smaller than $p/8$. Now, comparing the powers of g in the support of (A.1) and (A.2), we see that

- if $x \in [1, cp]$, then $\left|C_{g^{p/2+1}, \lfloor \log(p) \rfloor - 1}\right\rangle$ and $\left|C_{y, \lfloor \log(p) \rfloor - 3}\right\rangle$ have disjoint supports, and therefore $\langle C_{y, \lfloor \log(p) \rfloor - 3} | C_{g^{p/2+1}, \lfloor \log(p) \rfloor - 1} \rangle = 0$, while

- if $x \in [\frac{p}{2}+1, \frac{p}{2}+cp]$, then the overlap is large and

$$|\langle C_{y,\lfloor \log(p) \rfloor -3} | C_{g^{p/2+1}, \lfloor \log(p) \rfloor -1} \rangle|$$

is a constant.

A.2. Quadratic residuosity.

The problem: The (total) language QR is to decide on input x, n whether or not x is a square modulo n. Without loss of generality we can assume the input x to the problem belongs to Z_n^*. Let us denote xRn if x is a square, i.e., if $x = y^2 \pmod{n}$ for some y. An efficient algorithm is known for the case where n is a prime, and the problem is believed to be hard for $n = pq$, where p, q are chosen at random among large primes p and q. A basic fact that follows directly from the Chinese remainder theorem is the following.

Fact A.1.

- *If the prime factorization of n is $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$, then for every x*

$$xRn \iff \forall_{1 \le i \le k}\ xRp_i.$$

- *If the prime factorization of n is $n = p_1 p_2 \dots p_k$ with different primes p_i, then every $z \in Z_n^*$ that has a square root has exactly 2^k square roots.*

Using this fact, we show how to reduce the $n = pq$ case to QS adopting the zero knowledge proof of [28].

The reduction: We use the circuit $C_a(r)$ that on input $r \in Z_n^*$ outputs $z = r^2 a \pmod{n}$. Suppose we know how to generate $|C_a\rangle$ for every a. On input integers n, x, $(n, x) = 1$, the algorithm proceeds as in the proof of Theorem 1.1 with the states $|C_1\rangle, |C_x\rangle$.

Correctness: We have

$$|C_x\rangle = \sum_{z \in Z_n^*} \sqrt{p_z}\, |z\rangle ,$$

where $p_z = \Pr_{r \in Z_n^*}(z = r^2 x)$, and

$$|C_1\rangle = \alpha \sum_{z : zRn} |z\rangle$$

for $\alpha = \frac{4}{(p-1)(q-1)}$ independent of z.

- If xRn, then $z = r^2 x$ is also a square. Furthermore, $p_z = \Pr_{r \in Z_n^*}(z = r^2 x) = \Pr_r(r \text{ is a square root of } \frac{z}{x})$, and as every square in Z_n^* has the same number of square roots, we conclude that $|C_x\rangle = |C_1\rangle$ and $\langle C_x | C_1 \rangle = 1$.
- Suppose x is not a square. For every $r \in Z_n^*$, $z = xr^2$ must be a nonresidue (or else xRn as well). We conclude that C_x has support only on nonresidues, and so $\langle C_x | C_1 \rangle = 0$.

We note that for a general n, different elements might have a different number of solutions (e.g., try $n = 8$), and so the given construction does not work.

A.3. Approximating CVP. Here we describe the reduction to QS from a gap problem of CVP (closest vector in a lattice), which builds upon the SZK proof of Goldreich and Goldwasser [26]. A lattice of dimension n is represented by a basis, denoted B, which is an $n \times n$ nonsingular matrix over $\mathbb{R}$. The lattice $\mathcal{L}(B)$ is the set of points $\mathcal{L}(B) = \{Bc \mid c \in \mathbb{Z}^n\}$, 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 represented by a basis B and a target vector $v \in \mathbb{R}^n$. The output should be the point $b \in \mathcal{L}(B)$ closest to v.

The CVP problem is NP-hard. Here we are interested in the variant of the problem in which the distance to the lattice is approximated to within a factor g. The approximation problem is known to be NP-hard when g is small, and on the other hand, it is known to be easy when g is exponential (see [6] for exact parameters and references). Here we are interested in the intermediate case, when g is about $\sqrt{\frac{n}{\log(n)}}$. In this range, the problem is not known to be in BPP, but on the other hand, it is known to be in SZK by [26] and therefore is not likely to be NP-hard. We use the SZK proof of [26] to give a reduction to the QS problem. We first describe the promise problem.

The problem:

Input: An n-dimensional lattice given by a basis B, a vector $v \in \mathbb{R}^n$, and a designated distance d. We set $g = g(n) = \sqrt{\frac{n}{c \log n}}$ for any fixed $c > 0$.

Promise:

- Yes: Instances where $d(v, \mathcal{L}(B)) \leq d$.
- No: Instances where $d(v, \mathcal{L}(B)) \geq g \cdot d$.

We let H_t denote the sphere of all points in $\mathbb{R}^n$ of distance at most t from the origin.

The reduction: The circuit C_0 gets as input a random string, and outputs the vector $r + \eta$, where r is a uniformly random point in $H_{2^n \|B \cup \{v\}\|} \cap \mathcal{L}(B)$ and η is a uniformly random point $\eta \in H_{\frac{g}{2} \cdot d}$. Reference [26] explains 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 the points cannot be randomly chosen from the real (continuous) vector space, due to precision issues, but [26] shows that taking a fine enough discrete approximation results in an exponentially small error. From now on, we work in the continuous world, bearing in mind that in fact everything is implemented by its discrete approximation. Now assume we can Qsample from the circuit C_0. We can then also Qsample from the circuit C_v, which we define to be the same circuit, except that the outputs are shifted by the vector v and become $r + \eta + v$. To solve the gap problem, the algorithm proceeds as in the proof of Theorem 1.1 with the states $|C_0\rangle, |C_v\rangle$.

Correctness: In a No instance, v is far away from the lattice $\mathcal{L}(B)$, namely, $d(v, \mathcal{L}(B)) \geq g \cdot d$. The calculation in [26] shows that the states $|C_0\rangle$ and $|C_v\rangle$ have no overlap, and so $\langle C_0 | C_v \rangle = 0$. On the other hand, suppose v is close to the lattice, $d(v, \mathcal{L}(B)) \leq d$. Notice that the noise η has magnitude about gd, and so the spheres around any lattice point r and around $r + v$ have a large overlap. Indeed, the argument of [26] shows that if we express $|C_0\rangle = \sum_z p_z |z\rangle$ and $|C_v\rangle = \sum_z p'_z |z\rangle$, then $\|p - p'\| \leq 1 - n^{-2c}$. We see that $\langle C_0 | C_v \rangle = F(p, p') \geq n^{-2c}$. Hence, if we could generate these states, we could iterate the above $poly(n)$ times and get a BQP algorithm for the problem.

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. D.A. is particularly grateful to Erik Winfree for an insightful conversation that initiated this work.

REFERENCES

[1] S. Aaronson, *Quantum lower bound for the collision problem*, in Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, Montreal, Canada, 2002, pp. 635–642.

[2] D. Aharonov, *Adiabatic quantum computation: Universality and tools*, talk at Mathematical Sciences Research Institute, 2002.

[3] D. Aharonov and A. Ta-Shma, *Adiabatic quantum state generation and statistical zero knowledge*, in Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, San Diego, CA, 2003, pp. 20–29. A longer version is available online from http://arxiv.org/abs/quant-ph/0210077.

[4] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, *Adiabatic quantum computation is equivalent to standard quantum computation*, in Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, Rome, Italy, 2004, pp. 42–51.

[5] D. Aharonov and O. Regev, *A lattice problem in quantum NP*, in Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, Cambridge, MA, 2003, pp. 210–219.

[6] D. Aharonov and O. Regev, *Lattice problems in $NP \cap coNP$*, in Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, Rome, Italy, 2004, pp. 362–371.

[7] N. Alon, *Eigenvalues and expanders*, Combinatorica, 6 (1986), pp. 83–96.

[8] A. Ambainis and O. Regev, *An elementary proof of the quantum adiabatic theorem*, http://arxiv.org/abs/quant-ph/0411152 (2006).

[9] D. Applegate and R. Kannan, *Sampling and integration of near log-concave functions*, in Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing, New Orleans, LA, 1991, pp. 156–163.

[10] Y. Avron and A. Elgart, *Adiabatic theorem without a gap condition*, Comm. Math. Phys., 203 (1999), pp. 445–463.

[11] R. B. Boppana, J. Hastad, and S. Zachos, *Does coNP have short interactive proofs?*, Inform. Process. Lett., 25 (1987), pp. 127–132.

[12] R. Bubley and M. Dyer, *Faster random generation of linear extensions*, in Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, 1998, pp. 350–354.

[13] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, *Exponential algorithmic speedup by quantum walk*, in Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, San Diego, CA, 2003, pp. 59–68. Also available online from http://arxiv.org/abs/quant-ph/0209131.

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

[15] A. M. Childs, E. Farhi, J. Goldstone, and S. Gutmann, *Finding cliques by quantum adiabatic evolution*, http://arxiv.org/abs/quant-ph/0012104 (2000).

[16] A. M. Childs, E. Farhi, and S. Gutmann, *An example of the difference between quantum and classical random walks*, http://arxiv.org/abs/quant-ph/0103020 (2001).

[17] W. van Dam and S. Hallgren, *Efficient quantum algorithms for shifted quadratic character problems*, http://arxiv.org/abs/quant-ph/0011067 (2001).

[18] W. van Dam, M. Mosca, and U. V. Vazirani, *How powerful is adiabatic quantum computation?*, in Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, Las Vegas, NV, 2001, pp. 279–287.

[19] W. van Dam and U. Vazirani, *More on the Power of Adiabatic Computation*, manuscript, 2001.

[20] E. Farhi, J. Goldstone, and S. Gutmann, *A numerical study of the performance of a quantum adiabatic evolution algorithm for satisfiability*, http://arxiv.org/abs/quant-ph/0007071 (2000).

[21] 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 (2001), pp. 472–476.

[22] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, *Quantum computation by adiabatic*

evolution, http://arxiv.org/abs/quant-ph/0001106 (2000).

[23] E. Farhi and S. Gutmann, *Quantum computation and decision trees*, http://arxiv.org/abs/quant-ph/9706062 (1998).

[24] L. Fortnow, *The complexity of perfect zero knowledge*, in Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, New York, NY, 1987, pp. 204–209.

[25] O. Goldreich and E. Kushilevitz, *A perfect zero-knowledge proof system for a problem equivalent to the discrete logarithm*, J. Cryptology, 6 (1993), pp. 97–116.

[26] O. Goldreich and S. Goldwasser, *On the limits of nonapproximability of lattice problems*, in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, Dallas, TX, 1998, pp. 1–9.

[27] O. Goldreich, A. Sahai, and S. Vadhan, *Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge*, in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, Dallas, TX, 1998, pp. 399–408.

[28] S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof systems*, SIAM J. Comput., 18 (1989), pp. 186–208.

[29] M. Grötschel and L. Lovász, *Combinatorial optimization*, in Handbook of Combinatorics, Vol. 1, 2, Elsevier, Amsterdam, 1995, pp. 1541–1597.

[30] L. Grover and T. Rudolph, *Creating superpositions that correspond to efficiently integrable probability distributions*, http://arxiv.org/abs/quant-ph/0208112 (2002).

[31] S. Hallgren, *Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem*, in Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, Montreal, Canada, 2002, pp. 653–658.

[32] M. Jerrum, A. Sinclair, and E. Vigoda, *A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries*, in Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, Heraklion, Greece, 2001, pp. 712–721.

[33] 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., PWS, Boston, MA, 1996.

[34] T. Kato, *On the adiabatic theorem of quantum mechanics*, J. Phys. Soc. Japan, 5 (1951), pp. 435–439.

[35] A. Yu. Kitaev, *Quantum measurements and the Abelian stabilizer problem*, http://arxiv.org/abs/quant-ph/9511026 (1995).

[36] E. Knill, *private communication.*

[37] J. Kobler, U. Schoning, and J. Turan, *The Graph Isomorphism Problem*, Birkhäuser Boston, Boston, MA, 1993.

[38] L. Lovász, *Random walks on graphs: A survey*, in Combinatorics, Paul Erdös is Eighty, Vol. 2, D. Miklos, V. T. Sos, and T. Szonyi, eds., Janos Bolyai Mathematical Society, Budapest, Hungary, 1996, pp. 353–398.

[39] A. Messiah, *Quantum Mechanics*, John Wiley and Sons, New York, 1958.

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

[41] T. Okamoto, *On relationships between statistical zero knowledge proofs*, in Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, 1996, pp. 649–658.

[42] A. Peres, *Quantum Theory: Concepts and Methods*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[43] B. Reichardt, *The quantum adiabatic optimization algorithm and local minima*, in Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, Chicago, IL, 2004, pp. 502–510.

[44] A. Sahai and S. P. Vadhan, *A complete promise problem for statistical zero-knowledge*, in Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, Miami Beach, FL, 1997, pp. 448–457.

[45] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput., 26 (1997), pp. 1484–1509.

[46] W. L. Spitzer and S. Starr, *Improved bounds on the spectral gap above frustration free ground states of quantum spin chains*, http://arxiv.org/abs/math-ph/0212029 (2002).

[47] S. Vadhan, *A Study of Statistical Zero Knowledge Proofs*, Ph.D. thesis, MIT, Cambridge, MA, 1999.

[48] J. Watrous, *Quantum algorithms for solvable groups*, in Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, Heraklion, Greece, 2001, pp. 60–67.