Adiabatic Quantum State Generation and Statistical Zero Knowledge

Dorit Aharonov* 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 the problem by studying the problem of 'quantum state generation'. We motivate this problem by first showing that the entire class of Statistical Zero Knowledge, which contains many 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 computation. Our first result states that any state which can be efficiently generated by standard quantum computation has an efficient ASG algorithm. We prove two useful lemmas for the design of quantum algorithms in this paradigm, the "sparse Hamiltonian lemma" and the "jagged adiabatic path lemma", and demonstrate how to use them to generate a wide class of Markov-chain related superpositions, including for example the uniform superposition over all perfect matchings in a bipartite graph.

Our results highlight intriguing links between quantum computation and many different areas: the analysis of spectral gaps and groundstates of Hamiltonians in physics, rapidly mixing Markov chains, statistical zero knowledge, and quantum random walks. We hope that these links will bring new insights and methods into quantum algorithms. After the first publication of these results [2], the ideas presented here initiated important progress in two open questions, one regarding the complexity of certain lattice problems [4, 5], and the other the universality of adiabatic computation [3].

*Department of Computer Science and Engineering, Hebrew University, Jerusalem, Israel and mathematical Sciences Research Institute, Berkeley, California, e-mail:doria@cs.huji.ac.il

†Department of Computer Science, Tel Aviv University, Tel-Aviv, Israel, e-mail:amnon@post.tau.ac.il

1 Introduction

Quantum computation carries the hope of solving in quantum polynomial time classically intractable tasks. The most notable success so far is Shor's quantum algorithm for factoring integers and for finding the discrete log [44]. 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 et al [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 walks. The algorithm, however, works in the black box model and solves a fairly contrived problem.

One cannot overstate the importance of developing qualitatively different quantum algorithmic techniques and approaches for the development of the field of quantum computation. In this paper we attempt to make a step in that direction by approaching the issue of quantum algorithms from a different point of view.

It has been folklore knowledge for a few years already that the problem of graph isomorphism, which is considered classically hard [36], has an efficient quantum algorithm as long as a certain state, namely the superposition of all graphs isomorphic to a given graph,

$$|\alpha_G\rangle = \sum_{\sigma \in S_n} |\sigma(G)\rangle \tag{1}$$

can be generated efficiently by a quantum Turing machine (for simplicity, we ignore normalizing constants in the above state and in the rest of the paper). The reason that generating $|\alpha_G\rangle$ suffices is very simple: For two isomorphic graphs, these states are identical, whereas for two non isomorphic graphs they are orthogonal. A simple circuit can distinguish between the case of orthogonal states and that of identical states, where the main idea is that if the states are orthogonal they will prevent the different states of a qubit attached to them to interfere. 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 be easily generated by this argument; However, 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 will mostly be interested in a restricted version of state generation, namely generating states corresponding to classical probability distributions, which we loosely refer to as *quantum generating* a distribution. To be more specific, we consider *the* *probability distribution of a circuit,* D_C , which is the distribution over the outputs of the classical circuit C when its inputs are uniformly distributed. Denote $|C\rangle \stackrel{\text{def}}{=} \sum_{z\in\{0,1\}^m} \sqrt{D_C(z)}\,|z\rangle$. We define the problem of circuit quantum generation:

Definition 1. Circuit Quantum Generation (CQG):

Input: *(ϵ, C) where C is a description of a classical circuit from n to m bits, and $0 \le \epsilon \le \frac{1}{2}$.*

Output: *A description of a quantum circuit Q of size $poly(|C|)$ such that $|Q(|\vec{0}\rangle) - |C\rangle\,| \le \epsilon$.*

We first show that most of the quantum algorithmic problems considered so far can be reduced to CQG. Most problems that were considered good candidates for BQP, such as discrete log (DLOG), quadratic residuosity, approximating closest and shortest vectors in a lattice, graph isomorphism and more, belong to the complexity class *statistical zero knowledge*, or SZK (see section 2 for background.) We prove

Theorem 1. *Any $\mathcal{L} \in SZK$ (Statistical Zero Knowledge) can be reduced to a family of instances of CQG.*

The proof relies on a reduction by Sahai and Vadhan [43] from SZK to a complete problem called statistical difference. Theorem 1 shows that a general solution for CQG implies $SZK \subseteq BQP$. We note that there exists an oracle A relative to which $SZK^A \not\subset BQP^A$ [1], and so such a proof must be non relativizing.

Theorem 1 translates a zero knowledge proof into an instance of CQG. In general, the reduction can be quite involved, building on the reduction in [43]. Specific examples of special interest turn out to be simpler, e.g., for the case of graph isomorphism described above the reduction results in a circuit C_G that gets as an input a uniformly random string and outputs a uniformly random graph isomorphic to G. In section 2 we demonstrate the reduction for three interesting cases: a decision variant of DLOG (based on a zero knowledge proof of Goldreich and Kushilevitz [24]), quadratic residuosity (based on a zero knowledge proof of Goldwasser, Micali and Rackoff [27]) and approximating the closest vector problem in lattices (based on a zero knowledge proof of Goldreich and Goldwasser [25]). The special cases reveal that although quite often one can look at the zero knowledge proof and directly infer the required state generation, sometimes it is not obvious such a transition exists at all. Theorem 1, however, tells us such a reduction is always possible.

The problem of what states can be generated efficiently by a quantum computer is thus of critical importance to the understanding of the computational power of quantum computers. We therefore embark on the task of designing tools for quantum state generation, and studying which states can be generated efficiently. The recently suggested framework of adiabatic quantum computation [23] seems to be tailored exactly for this purpose, since it is stated in terms of quantum state generation; Let us first explain this framework.

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

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle. \tag{2}$$

where $H(t)$ is an operator called the *Hamiltonian* of the system. We consider systems of n qubits where the Hamiltonians $H(t)$ can actually be implemented in the lab. The qualitative statement of the adiabatic theorem is that if the quantum system is initialized in the ground state (the eigenstate with lowest eigenvalue) of $H(0)$, and if the modification of $H(t)$ in time is done slowly enough, namely *adiabatically*, then the final state is the ground state of the final Hamiltonian $H(T)$.

Recently, Farhi, Goldstone, Gutmann and Sipser [23] suggested to use adiabatic evolutions to solve NP-hard languages. It was shown in [23, 18] that such adiabatic evolutions can be simulated efficiently on a quantum circuit, and so designing such a successful process would imply a quantum efficient algorithm for NP problems. Farhi *et. al.*'s idea was to find the minimum of a given function f as follows: $H(0)$ is chosen to be some generic Hamiltonian. $H(T)$ is chosen to be the *problem Hamiltonian*, namely a matrix which has the values of f on its diagonal and zero everywhere else. The system is then initialized in the ground state of $H(0)$ and evolves adiabatically on the convex line $H(t) = (1-\frac{t}{T})H_0 + \frac{t}{T}H_T$. By the adiabatic theorem if the evolution is slow enough, the final state is the groundstate of $H(T)$ which is exactly the sought after minimum of f.

The efficiency of these adiabatic algorithms is determined by how slow the adiabatic evolution needs to be for the adiabatic theorem to hold. It turns out that this depends mainly on the *spectral gaps* of the Hamiltonians $H(t)$. If these spectral gaps are not too small the modification of the Hamiltonians can be done 'fairly fast' and the adiabatic algorithm then becomes efficient. The main problem in analyzing the efficiency of adiabatic algorithms is thus lower bounding the spectral gap; This is a very difficult task in general, and hence not much is known analytically about adiabatic algorithms. [20, 15, 21] analyze numerically the performance of adiabatic algorithms on random instances of NP complete problems. It was proven in [18, 42] that Grover's quadratic speed up [29] can be achieved adiabatically. Lower bounds for special cases were given in [18].

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

In order to provide a framework for the study of state generation using the adiabatic language, we define *adiabatic quantum state generation* as general as we can. Thus, we replace the requirement that the Hamiltonians are on a straight line, with

Hamiltonians on any general path. Second, we only require that the Hamiltonians are *simulatable*, i.e., that the unitary matrix $e^{-itH(s)}$ can be approximated by a quantum circuit to within any polynomial accuracy for any polynomially bounded time t, rather than requiring that they are local as was previously done. We show any adiabatic state generator can be simulated efficiently by a quantum circuit. We give two proofs of this fact. The first proof follows from the adiabatic theorem. The second proof is self contained, and does not require knowledge of the adiabatic theorem. Instead it uses the simple Zeno effect[41], thus providing an alternative point of view of adiabatic algorithms using measurements (such a path was taken also in [14].) Thus, adiabatic state generators can be used as a framework for designing algorithms for quantum state generation.

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

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

We note that this general lemma is useful also in two other contexts: first, in the context of simulating complicated physical systems on a quantum circuit. Second, for continuous quantum walks [16] which use Hamiltonians. For example, in [13] Hamiltonians are used to derive an exponential quantum speed up using quantum walks. Our lemma can be used directly to simplify the Hamiltonian implementation used in [13] and to remove the unnecessary constraints (namely coloring of the nodes) which were assumed for the sake of simulating the Hamiltonian.

The next question that one encounters in designing adiabatic quantum state generation algorithms concerns bounding the spectral gap, which as we mentioned before is a difficult task. We would like to develop tools to find paths in the Hamiltonian space such that the spectral gaps are guaranteed to be non negligible, i.e. larger than $1/poly(n)$. Our next lemma provides a way to do this in certain cases. Denote $\alpha(H)$ to be the ground state of H (if unique.)

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

A naive attempt to prove this lemma is to use the sequence of Hamiltonians H_j as *stepping stones* for the adiabatic computation, connecting H_j to H_{j+1} by a straight line to create the path $H(t)$. However this way the spectral gaps along the way might be small. To overcome this we use two simple but useful ideas. The first idea is to replace each Hamiltonian H_j by the Hamiltonian Π_{H_j} which is the projection on the subspace orthogonal to the ground states of H_j. We show how to implement such a projection using Kitaev's phase estimation algorithm [35]. The second useful idea is to connect by straight lines projections on states with non negligible inner product. We show that the Hamiltonians on such a line are guaranteed to have non negligible spectral gap. These ideas can be put together to show that the jagged adiabatic path connecting the *projections* Π_{H_j} is guaranteed to have sufficiently large spectral gaps.

We use the above tools to show that

Theorem 2. *Any quantum state that can be efficiently generated in the circuit model, can also be efficiently generated by an adiabatic state generation algorithm, and vice versa.*

Thus the question of the complexity of quantum state generation is equivalent (up to polynomial terms) in the circuit model and in the adiabatic state generation model.

In the final part of the paper we demonstrate how our methods for adiabatic quantum state generation work in a particularly interesting domain, namely quantum generating the limiting distribution of a Markov chain. There is an interesting connection between rapidly mixing Markov chains and adiabatic computation. A Markov chain is rapidly mixing if and only if the second eigenvalue gap, namely the difference between the largest and second largest eigenvalue of the Markov matrix M, is non negligible [7]. This clearly bears resemblance to the adiabatic condition of a non negligible spectral gap, and suggests to look at Hamiltonians of the form

$$H_M = I - M. \tag{3}$$

The $I - M$ term appears 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 will be 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 8.) The sparse Hamiltonian lemma has as an immediate corollary that for a special type of Markov chains, which we call *strongly samplable*, the quantum analog of the Markov chain can be implemented:

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

In adiabatic computation one is interested in sequences of Hamiltonians and we therefore consider sequences of strongly samplable Markov chains. This bears resemblance to some approximate counting algorithms [33] where we start with a Markov chain that is easy to count and which is contained in a large set Ω the size of which we want to estimate and we slowly increase the set on which the Markov chain operates so as to finally get to the desired set Ω. This paradigm and modifications of it in which the Markov chains are modified slightly until the desired Markov chain is attained, are a commonly used tool in many algorithms. A notable example is the recent algorithm for approximating the permanent [32]. In the last part of the paper we show how to use our techniques to translate such approximate counting algorithms to quantum generating the limiting distribution of the final Markov chain. We show:

Theorem 3. *(Loosely:) 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 simple Markov chain. Then there is an efficient quantum algorithm Q that quantum generates the final limiting distribution over Ω.*

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

We exploit this paradigm to quantum generate 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 generate all linear extensions of partial orders using Bubley and Dyer algorithm [12], all lattice points in a convex body satisfying certain restrictions using Applegate-Kannan technique [9] and many more states. We note that some of these states (perhaps all) can be generated using standard techniques which exploit the self reducibility of the problem (see [30]). We stress however that our techniques are qualitatively and significantly different from previous techniques for generating quantum states, and in particular do not require self reducibility. This can be important for extending this approach to other quantum states.

In this paper we have set the grounds for the general study of the problem of quantum generation and adiabatic quantum state generation, where we have suggested to use the language of Hamiltonians and ground states for quantum state generation. This direction points at very interesting and intriguing connections between quantum computation and many different areas: the complexity class SZK and its complete problem *statistical difference* [43], the notion of adiabatic evolution [34], the study of rapidly mixing Markov chains using spectral gaps [38], quantum walks [13], and the study of ground states and spectral gaps of Hamiltonians in Physics. Hopefully, these connections will point at various techniques from these areas which can be borrowed to give more tools for adiabatic quantum state generation; notably, the study of spectral gaps of Hamiltonians in physics is a lively area with various recently developed techniques (see [45] and 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. An open question which might help in the task is to present known quantum algorithms, e.g., Shor's DLOG algorithm, or the quadratic residuosity algorithm, in the language of adiabatic computation in an insightful way.

After the publication of the preliminary version of this article [2], the ideas presented in this paper were used to make important progress in two open questions.

First, our results regarding the problem of approximating the closest vector in a lattice up to $\sqrt{n}$ ($GapCVP_{\sqrt{n}}$) were directly used in [4] which showed that the problem lies in quantum NP. This gave the first non trivial quantum complexity upper bound on a lattice problem. Inspired by further ideas from the current paper, [5] improved the results in [4] to give the classical complexity result that the same lattice problem lies in $NP \cap coNP$.

The second place were 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 regarding the universality of ASG were recently improved in [3] to show that the model of adiabatic computation using *local* Hamiltonians is universal for quantum computation.

The rest of the paper is organized as follows. We start with the results related to SZK; We then describe quantum adiabatic computation, define the adiabatic quantum state generation framework and use the adiabatic theorem to prove that an adiabatic state generator implies a state generation algorithm. Next we prove our two main tools: the sparse Hamiltonian lemma, and the jagged adiabatic path lemma. We then use these tools to prove that adiabatic state generation is equivalent to standard quantum state generation. Finally we draw the connection to Markov chains and demonstrate how to use our techniques to generate approximately countable sets. In the appendix we give the second proof of transforming adiabatic state generators to quantum algorithms using the Zeno effect.

2 Quantum Generation and SZK

We start with some background about Statistical Zero Knowledge. For an excellent source on this subject, see Vadhan's thesis [47] or Sahai and Vadhan [43].

2.1 SZK

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, 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.

We say a promise problem Π has an interactive proof with soundness error ϵ_s and completeness error ϵ_c if there exists an interactive protocol between a prover P and a verifier V denoted by (P, V), where V is a probabilistic polynomial time machine, and

- 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.

When an interactive proof system (Π, V) for a promise problem Π is run on an input x, it produces a distribution over "transcripts" that contain 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).

An interactive proof system (Π, V) for a promise problem Π is said to be "honest verifier statistical zero knowledge", and in short HVSZK, 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 ℓ_1 distance defined below) to the distribution on transcripts in the real proof. If the simulated distribution is exactly the correct distribution, we say the proof system is "honest verifier *perfect* zero knowledge", and in short HVPZK.

We stress that the simulator's output is based on the input alone, and the simulator has no access to the prover. Also, note that we only require the simulator to produce a good distribution on inputs in Π_{Yes}, and we do not care about other inputs. This is because for "No" instances there is no correct proof anyway. We refer the interested reader to Vadhan's thesis [47] for rigorous definitions and a discussion of their subtleties.

The definition of HVSZK captures exactly the notion of "zero knowledge"; If the honest verifier can simulate the interaction with the prover by himself in case the input is in Π, then he does not learn anything from the interaction (except for the knowledge that the input is in Π). We denote by HVSZK the class of all promise problems that have an interactive proof which satisfies these restrictions. One can wonder whether cheating verifiers can get information from an honest prover by deviating from the protocol. Indeed, in some interactive proofs this happens. However, a general result says that any HVSZK proof can be simulated by one which does not leak much information even with dishonest verifiers [26]. We thus denote by SZK the class of all promise problems which have interactive proof systems which are statistically zero knowledge against a honest (or equivalently a general) verifier.

It is known that $BPP \subseteq SZK \subseteq AM \cap coAM$ and that SZK is closed under complement. It follows that SZK does not contain any NP–complete language unless the polynomial time hierarchy collapses. For this and other results known about this elegant class we refer the reader, again, to Vadhan's thesis [47].

2.2 The complete problem

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

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

We also define the variation distance to be $||p-q|| = \frac{1}{2}|p-q|_1$ so that it takes values between 0 and 1. The following fact is very useful:

Fact 1. *(See [40])*

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

or equivalently

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

We can now define the complete problem for SZK:

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

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

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

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

Sahai and Vadhan [43, 47] show that for any two constants $0 \leq \beta < \alpha \leq 1$ such that $\alpha^2 > \beta$, $SD_{\alpha,\beta}$ is complete for SZK [1]. A well explained exposition can also be found in [47].

2.3 A Reduction from SZK to CQG.

We need a very simple building block.

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

The proof is a direct calculation. We now proceed to prove Theorem 1.

Proof. Let C_0, C_1 be an input to SD, C_0, C_1 are circuits with m outputs. It is enough to show that given that we can quantum generate the given circuits we can put $SD_{3/4,1/4}$ in BQP. Let us first assume that we can quantum generate both circuits

[1] Sahai and Vadhan also show ([47], Proposition 4.7.1) that any promise problem in HVPZK reduces to $\overline{SD_{1/2,0}}$, where the line above the class denotes complement, i.e., we swap between the yes and no instances.

with $\epsilon = 0$ error. We can therefore generate the superposition $\frac{1}{\sqrt{2}}(|0\rangle|C_0\rangle + |1\rangle|C_1\rangle)$. We then apply a Hadamard gate on the first qubit and measure it. We use Claim 1 with $v = |C_0\rangle$ and $w = |C_1\rangle$. In our case

$$\langle v|w\rangle = \sum_{z\in\{0,1\}^m} \sqrt{D_{C_0}(z)D_{C_1}(z)} = F(D_{C_0}, D_{C_0}) \tag{4}$$

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

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

Setting $\alpha = \frac{3}{4}$ and $\beta = \frac{1}{4}$ we get that if $||D_{C_0} - D_{C_1}|| \geq \alpha$ we measure 0 with probability at most $\frac{1+\sqrt{1-\alpha^2}}{2} \leq 0.831$, while if $||D_{C_0} - D_{C_1}|| \leq \beta$ we measure 0 with probability at least $1-\frac{\beta}{2} \geq \frac{7}{8} = 0.875$. Repeating the experiment $O(\log(\frac{1}{\delta}))$ times, we can decide on the right answer with error probability smaller than δ. If the quantum sampling circuit has a small error (say $\epsilon < \frac{1}{100}$) then the resulting states are close to the correct ones and the error introduced can be swallowed by the gap of the BQP algorithm. □

The above theorem shows that in order to give an efficient quantum algorithm for any problem in SZK, it is sufficient to efficiently quantum generate the corresponding circuits. Often, but not always, one does not need to go through the complicated reduction of Sahai and Vadhan [43], and instead can directly translate a zero knowledge proof for a certain language into a quantum generation problem. We illustrate this with three examples of particular interest for quantum algorithms: discrete log, quadratic residuosity and a gap version of Closest vector in a lattice.

2.4 A promise problem equivalent to Discrete Log

The problem :

Goldreich and Kushilevitz [24] 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: The promise is that $x = \log_g(y)$ is in $[1, cp] \cup [\frac{p}{2}+1, \frac{p}{2}+cp]$,
- Output: Whether $x \in [1, cp]$ or $x \in [\frac{p}{2}+1, \frac{p}{2}+cp]$

[24] proves that DLOG is reducible to DLP_c for every $0 < c < 1/2$. They also prove that DLP_c has a perfect zero knowledge proof if $0 < c \leq 1/6$. We take $c = 1/6$ and show how to solve $DLP_{1/6}$ with CQG.

The reduction :

We assume we can solve the construction problem for the circuit $C_{y,k} = C_{n,g,y,k}$ that computes $C_{y,k}(i) = y \cdot g^i (\text{mod} p)$ for $i \in \{0,1\}^k$. The algorithm gets into the state $\frac{1}{\sqrt{2}}[\ |0\rangle \left|C_{g^{p/2+1}, \lfloor \log(p) \rfloor - 1}\right\rangle + |1\rangle \left|C_{y, \lfloor \log(p) \rfloor - 3}\right\rangle\ ]$ and proceeds as in Claim 1.

Correctness :

We have:

$$\left|C_{g^{p/2+1}, \lfloor \log(p) \rfloor - 1}\right\rangle = \frac{1}{\sqrt{2^t}} \sum_{i=0}^{2^t-1} \left|g^{p/2+i}\right\rangle \tag{5}$$

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

$$\left|C_{y, \lfloor \log(p) \rfloor - 3}\right\rangle = \frac{1}{\sqrt{2^{t'}}} \sum_{i=0}^{2^{t'-1}} \left|g^{x+i}\right\rangle \tag{6}$$

where t' is the largest power of 2 smaller than $p/8$. Now, comparing the powers of g in the support of Equations 5 and 6 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.

2.5 Quadratic residuosity

The problem :

We denote xRn if $x = y^2 (\text{mod} n)$ for some y, and xNn otherwise. The problem QR is to decide on input x, n whether xRn. An efficient algorithm is known for the case of n being 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

Fact 2.

- *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 $n = p_1 p_2 \dots p_k$ then every $z \in Z_n$ that has a square root, has the same number of square roots.*

We show how to reduce the $n = pq$ case to CQG adopting the zero knowledge proof of [27].

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 for every a. On input integers n, x, $(n, x) = 1$, the algorithm gets into the state $\frac{1}{\sqrt{2}}[|0\rangle|C_1\rangle + |1\rangle|C_x\rangle]$ and proceeds as in Claim 1.

Correctness :

We have

$$|C_x\rangle = \sum_z \sqrt{p_z}\,|z\rangle \tag{7}$$

where $p_z = \Pr_r(z = r^2 x)$, and

$$|C_1\rangle = \sum_{z:zRn} \alpha\,|z\rangle \tag{8}$$

for some fixed α independent of z.

- If xRn then $z = r^2 x$ is also a square. Furthermore, as $(x, n) = 1$ we have $p_z = \Pr_r(r \text{ is a square root of } \frac{z}{x})$ and as every square 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 xNn. There are only $p+q-1$ integers $r \in Z_n$ that are not co-prime to n. For every r co-prime with n, $z = xr^2$ must be a non-residue (or else xRn as well). We conclude that $\sum_{z:zRn} p_z \le \frac{p+q}{pq} \approx 0$ and so $\langle C_x|C_1\rangle \approx 0$.

We note that for a general n, different elements might have a different number of solutions (e.g., try $n = 8$) and the number of elements not co-prime to n might be large, so one has to be more careful.

2.6 Approximating CVP

We describe here the reduction to CQG for a gap problem of CVP (closest vector in a lattice), which builds upon the statistical zero knowledge proof of Goldreich and Goldwasser [25]. A lattice of dimension n is represented by a basis, denoted B, which is an $n \times n$ non-singular matrix over $\mathbb{R}$. The lattice $\mathcal{L}(B)$ is the set of points $\mathcal{L}(B) = \{Bc \mid c \in \mathbb{Z}^n\}$, 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 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 problem is NP hard. Furthermore, it is NP hard to approximate the distance to the closest vector in the lattice to within small factors, and it is easy to approximate it to within $2^{\epsilon n}$ factor, for every $\epsilon > 0$. See [25] for a discussion. In [25] an (honest prover) perfect zero knowledge proof for being far away from the lattice is given. We now describe the promise problem.

The problem :

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

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 $\eta \in H_{\frac{g}{2} \cdot d}$. [25] explain how to sample such points with almost the right distribution, i.e. they give a description of an efficient such C_0.

We remark that the points cannot be randomly chosen from the real (continuous) vector space, due to precision issues, but [25] show that taking a fine enough discrete approximation and a large enough cutoff of the lattice results in an exponentially small error. From now on we work in the continuous world, bearing in mind that in fact everything is implemented in a discrete approximation of it.

Now assume we can quantum sample from the circuit C_0. We can then also quantum sample 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 gets into the state $\frac{1}{\sqrt{2}} [\ |0\rangle |C_0\rangle + |1\rangle |C_v\rangle\ ]$ and proceeds as in Claim 1.

Correctness :

If v is far away from the lattice $\mathcal{L}(B)$, then the calculation at [25] 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 [25] 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 $|p - p'|_1 \leq 1 - n^{-2c}$. We see that $\langle C_0 | C_v \rangle = F(p, p') \geq n^{-2c}$. Iterating the above $poly(n)$ times we get an RQP algorithm, namely a polynomial quantum algorithm with one sided error.

3 Physics Background

This section gives background required for our definition of adiabatic state generation. We start with some preliminaries regarding the operator norm and the Trotter formula. We then describe the adiabatic theorem, and the model of adiabatic computation as defined in [23].

3.1 Spectral Norm

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

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

If T is Hermitian or Unitary (in general, if T is normal, namely commutes with its adjoint) than $||T||$ equals the largest absolute value of its eigenvalues. If U is unitary, $||U|| = 1$. Also, $||AB|| \leq ||A|| \cdot ||B||$. Finally, if $A = (a_{i,j})$ is a $D \times D$ matrix, then $||A|| \leq D^2 ||A||_\infty$ where $||A||_\infty = \max_{i,j} |a_{i,j}|$.

Definition 3. *We say a linear transformation T_2 α–approximates a linear transformation T_1 if $||T_1 - T_2|| \leq \alpha$, and if this happens we write $T_2 = T_1 + \alpha$.*

3.2 Trotter Formula

Say $H = \sum H_m$ with each H_m Hermitian. Trotter's formula states that one can approximate e^{-itH} by slowly interleaving executions of e^{-tH_m}. We use the following variant of it:

Lemma 3. *[40] Let H_i be Hermitian, $H = \sum_{m=1}^{M} H_m$. Further assume H and H_i have bounded norm, $||H||, ||H_i|| \leq \Lambda$. Define*

$$U_\delta \quad = \quad [\ e^{-\delta i H_1} \cdot e^{-\delta i H_2} \cdot \ldots \cdot e^{-\delta i H_M}\ ] \cdot [\ e^{-\delta i H_M} \cdot e^{-\delta i H_{M-1}} \cdot \ldots \cdot e^{-\delta i H_1}\ ]$$

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

Using the $|| \cdot ||$ properties stated above we conclude:

Corollary 2. *Let H_i be Hermitian, $H = \sum_{m=1}^{M} H_m$. Assume $||H||, ||H_i|| \leq \Lambda$. Then, for every $t > 0$*

$$||U_\delta^{\frac{t}{2\delta}} - e^{-itH}|| \quad \leq \quad O(\frac{t}{2\delta} \cdot M \cdot (\delta\Lambda)^3) \tag{9}$$

As $||U_\delta - I|| \leq 2M\Lambda\delta$ we also have $||U_\delta^{\lfloor \frac{t}{2\delta} \rfloor} - U_\delta^{\frac{t}{2\delta}}|| \leq 2M\Lambda\delta$ and thus:

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

$$||U_\delta^{\lfloor \frac{t}{2\delta} \rfloor} - e^{-itH}|| \quad \leq \quad O(M\Lambda \cdot \delta + M\Lambda^3 t \cdot \delta^2) \tag{10}$$

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

3.3 Time Dependent Schrodinger Equation

A quantum state $|\psi\rangle$ of a quantum system evolves in time according to Schrodinger's equation:

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle \tag{11}$$

where $H(t)$ is a Hermitian matrix which is called the Hamiltonian of the physical system. The evolution of the state from time 0 to time T can be described by integrating Schrodinger's equation over time. If H is constant and independent of time, one gets

$$|\psi(T)\rangle = U(T)|\psi(0)\rangle = e^{-iHT}|\psi(0)\rangle \tag{12}$$

Since H is Hermitian $U = e^{-iHT}$ is unitary, and so we get the familiar unitary evolution from quantum circuits. The time evolution is unitary regardless of whether $H(t)$ is time dependent or not.

The groundstate 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 $\Delta(H)$.

3.4 The adiabatic Theorem

In the study of *adiabatic evolution* one is interested in the long time behavior (at large times T) of a quantum system initialized in the ground state of H at time 0 when the Hamiltonian of the system $H(t)$ changes very slowly in time, namely *adiabatically*.

The qualitative statement of the adiabatic theorem is that if the quantum system is initialized in the ground state of $H(0)$, the Hamiltonian at time 0, and if the modification of H along the path $H(t)$ in the Hamiltonian space is done infinitely slowly, then the final state will be the ground state of the final Hamiltonian $H(T)$.

To make this statement correct we need to add various conditions and quantifications. Historically, the first and simplest adiabatic theorem was found by Born and Fock in 1928 [11]. In 1958 Kato [34] improved the statement to essentially the statement we use in this paper (which we state shortly), and which is usually referred to as *the adiabatic theorem*. A proof can be found in [39]. For more sophisticated adiabatic theorems see [10] and references therein.

To state the adiabatic theorem, it is convenient and traditional to re-scale time so that it varies between 0 and 1. Thus, as T is the total time, $s = \frac{t}{T}$ is the re-scaled time. The Schrodinger's equation restated in terms of the re-scaled time s then reads

$$i\hbar \frac{d}{ds}|\psi(s)\rangle = T \cdot H(s)|\psi(s)\rangle \tag{13}$$

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

Theorem 4. (The adiabatic theorem, adapted from [39, 23]). *Let $H(\cdot)$ be a function from $[0,1]$ to the vector space of Hamiltonians on n qubits. Assume $H(\cdot)$ is* *continuous, has a unique ground state for every $s \in [0,1]$, and is differentiable in all but possibly finitely many points. Let $\epsilon > 0$ and assume that the following adiabatic condition holds for all points $s \in (0,1)$ in which the derivative is defined:*

$$T\epsilon \geq \frac{\|\frac{d}{ds}H(s)\|}{(\Delta(H(s))^2} \tag{14}$$

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

We will refer to Equation 14 as the *adiabatic condition*.

The proof of the adiabatic theorem is rather involved. One way to get intuition about it is by observing how the Schrodinger equation behaves when eigenstates are considered; If the eigenvalue is λ, the eigenstate evolves by a multiplicative factor $e^{i\lambda t}$, which rotates in time faster the larger the absolute value of the eigenvalue λ is, and so the ground state rotates the least. The fast rotations are essentially responsible to the cancellations of the contributions of the vectors with the higher eigenvalues, due to interference effects.

4 Adiabatic Quantum State Generation

In this section we define our paradigm for quantum *state generation* based on the ideas of adiabatic quantum *computation* and the adiabatic theorem. We would like to allow as much flexibility as possible in the building blocks. We therefore allow any Hamiltonian which can be implemented efficiently by quantum circuits. We also allow using general Hamiltonian paths and not necessarily straight lines. We define:

Definition 4. (Simulatable Hamiltonians). *We say a Hamiltonian H on n qubits is simulatable if for every $t > 0$ and every accuracy $0 < \alpha < 1$ the unitary transformation*

$$U(t) = e^{-iHt} \tag{15}$$

can be approximated to within α accuracy by a quantum circuit of size $poly(n, t, 1/\alpha)$.

If H is simulatable, then by definition so is cH for any $0 \leq c \leq poly(n)$. It therefore follows by Trotter's equation (3) that any convex combination of two simulatable, bounded norm Hamiltonians is simulatable. Also, If H is simulatable and U is a unitary matrix that can be efficiently applied by a quantum circuit, then $UHU^\dagger$ is also simulatable, because $e^{-itUHU^\dagger} = Ue^{-itH}U^\dagger$.

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 allowable path in the Hamiltonian space:

Definition 5. (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 finitely many points,*
- *$\forall s$ $H(s)$ has a unique groundstate, and*
- *$\forall s$, $H(s)$ is simulatable given s.*

Adiabatic quantum state generation is basically the process of implementing the Schrodinger's evolution along an adiabatic path, where we require that the adiabatic condition holds.

Definition 6. (Adiabatic Quantum State Generation). *An adiabatic Quantum State Generator $(H_x(s), T, \epsilon)$ has for every $x \in \{0,1\}^n$ an adiabatic path $H_x(s)$, such that for the given T, ϵ the adiabatic condition is satisfied for all $s \in [0,1]$ where it is defined. We also require that the generator is explicit, i.e., that there exists a polynomial time quantum machine that*

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

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

Remark: We note that in previous papers on adiabatic computation, eg. in [18], a delay schedule $\tau(s)$ which is a function of s was used. We choose to work with one single delay parameter, T, instead, which might seem restrictive; However, working with a single parameter does not restrict the model since more complicated delay schedules can be encoded into the dependence of H on s.

4.1 Circuit simulation of adiabatic state generation

An adiabatic state generator can be simulated efficiently by a quantum circuit:

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

Proof. **(Based on Adiabatic Theorem)** The circuit is built by discretizing time to sufficiently small intervals of length δ, and then applying the unitary matrices $e^{-iH(\delta j)\delta}$. Intuitively this should work, as the adiabatic theorem tells us that a physical system evolving for time T according to Schrodinger's equation with the given adiabatic path will end up in a state close to $\alpha(H_x(1))$. The formal error analysis can be done by exactly the same techniques that were used in [18]. We do not give the details of the proof based on the adiabatic theorem here, since the proof of the adiabatic theorem itself is hard to follow. □

We give a second proof of Claim 2 that does not require knowledge of the adiabatic theorem and instead relies on the Zeno effect[41]. We believe it gives a different, illuminating perspective on the adiabatic evolution from the measurement point of view. We note that another approach toward the connection between adiabatic computation and measurements was taken in [14]. The proof is simple and we sketch it here and give the full proof in Appendix A.

Proof. **(Based on the Zeno effect)** As before, we begin at the state $\alpha(H(0))$, and the circuit is built by discretizing time to sufficiently small intervals of length δ. At each time step j, $j = 1, \ldots, R$, instead of simulating the Hamiltonian as before we apply a measurement determined by $H(s_j)$. Specifically, we measure the state in a basis which includes the groundstate $\alpha(H(s_j))$. If R is sufficiently large, the subsequent Hamiltonians are very close in the spectral norm, and the adiabatic condition guarantees that their groundstates are very close in the Euclidean norm. Given that at time step j the state is the groundstate $\alpha(H(s_j))$, the next measurement results with very high probability in a projection on the new groundstate $\alpha(H(s_{j+1}))$. The Zeno effect guarantees that the error probability behaves like $1/R^2$, i.e. quadratically in R (and not linearly), and so the accumulated error after R steps is still small, which implies that the probability that the final state is the groundstate of $H(1)$ is very high, if R is taken to be large enough. □

5 The Sparse Hamiltonian Lemma

Our first concern is which Hamiltonians can be simulated efficiently. We restate the sparse Hamiltonian lemma:

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

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.

5.1 The reduction to 2×2 combinatorially block diagonal matrices.

Let us define:

Definition 7. (Combinatorial block.) *Let A be a matrix with rows $ROWS(A)$ and columns $COLS(A)$. We say $(R, C) \subseteq ROWS(A) \times COLS(A)$ is a combinatorial* *block if $|R| = |C|$, for every $c \in C$, $r \notin R$, $A(c,r) = 0$, and for every $c \notin C$, $r \in R$, $A(c,r) = 0$.*

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

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

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

Proof. (Of Claim 3) We color all the entries of H with $(D+1)^2 n^6$ colors. For $(i,j) \in [N] \times [N]$ and $i < j$ (i.e., (i,j) is an upper-diagonal entry) we define:

$$col_H(i,j) = (k\ ,\ i \bmod k\ ,\ j \bmod k\ ,\ rindex_H(i,j)\ ,\ cindex_H(i,j)) \tag{16}$$

where

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

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

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

Indeed, fix a color m. Let us order all the upper-triangular, non-zero elements of H_m in a list $NONZERO_m = \{(i,j) \mid H_m(i,j) \neq 0 \ \ and \ \ i \leq j\}$. Say the elements of $NONZERO_m$ are $\{(i_1,j_1), \ldots, (i_B,j_B)\}$. For every element $(i_b,j_b) \in NONZERO_m$ we introduce a block. If $i_b = j_b$ then we set $R_b = C_b = \{i_b\}$ while if $i_b \neq j_b$ then we set $R_b = C_b = \{i_b, j_b\}$.

Say $i_b \neq j_b$ (the $i_b = j_b$ case is similar and simpler) and the elements (i_b,j_b) and (j_b,i_b) are colored by $m = (k, i_b \bmod k, j_b \bmod k, rindex_H(i_b,j_b), cindex_H(i_b,j_b))$. To see that $R_b = C_b = \{i_b, j_b\}$ is a 2×2 combinatorial block it suffices to show that there are no other elements colored by m on the i_b'th and j_b'th 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 element in $NONZERO_m$ from that row or column. Furthermore, all elements (j_b, a) on the j_b'th 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_b'th row is colored by m and the claim follows (see Figure 1). $\square$

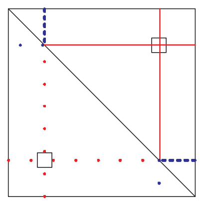

Figure 1: *In the upper diagonal side of the matrix H_m: the row and column of (i_b, j_b) are empty because the color m contains the row-index and column index of (i, j), and the j_b'th row and i_b'th column are empty because m contains k, $i \bmod k$, $j \bmod k$ and $i \bmod k \neq j \bmod k$. The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

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

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

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

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

5.2 2×2 combinatorially block diagonal matrices are simulatable.

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

Proof. Let $t > 0$ and 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 input $|k\rangle$ is captured by a 2×2 unitary transformation U_k. Formally, given k, say $|k\rangle$ belongs to a 2×2 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

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

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 $\alpha^{O(1)}$ accuracy.

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:

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

for $|v\rangle \in Span\{min_k, max_k\}$.

Our algorithm is applying T_1 followed by T_2 and then T_1^{-1} for cleanup.

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

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

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

$\square$

We proved the claim for matrices with 2×2 combinatorial blocks. We remark that the same approach works for matrices with $m \times m$ combinatorial blocks, as long as m is polynomial in n.

5.3 Proving the sparse Hamiltonian lemma

We now prove the sparse Hamiltonian Lemma:

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

We express $H = \sum_{m=1}^{M} H_m$ as in Claim 3, $M \leq (D+1)^2 n^6 \leq poly(n)$. We choose Δ such that $O(Mt\Lambda^3\Delta^2) \leq \frac{\alpha}{2}$. Note that $\frac{1}{\Delta} \leq poly(t,n)$ for some large enough polynomial. By Claim 5 we can simulate in polynomial time each $e^{-i\Delta H_m}$ to within $\frac{\alpha}{2Mt/\Delta}$ accuracy. We then compute $U_{\Delta}^{\frac{t}{2\Delta}}$, using our approximations to $e^{-i\Delta H_m}$, as in Corollary 3. Corollary 3 assures us that our computation is α close to e^{-itH}, as desired (suing the fact that for every m, $||H_m|| \leq ||H|| = \Lambda \leq poly(n)$). The size of the computation is $\frac{t}{2\Delta} \cdot 2M \cdot poly(\Delta, M, n, \alpha) = poly(n,t,\alpha)$ as required. $\square$

6 The Jagged Adiabatic Path Lemma

Next we consider the question of which paths in the Hamiltonian space guarantee non negligible spectral gaps. We restate the jagged adiabatic path lemma.

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

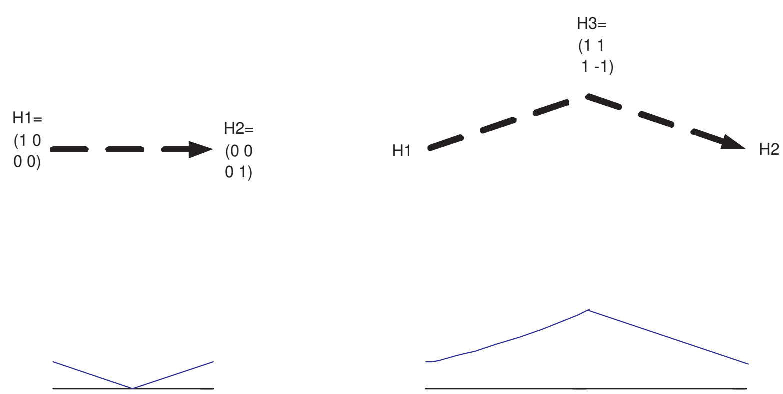

Figure 2: *In the left side of the drawing we see two Hamiltonians H_1 and H_2 connected by a straight line, and the spectral gaps along that line. In the right side of the drawing we see the same two Hamiltonians H_1 and H_2 connected through a jagged line that goes through a third connecting Hamiltonian 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.*

Proof. (of Lemma 2) We replace the sequence $\{H_j\}$ with the sequence of Hamiltonians $\{\Pi_{H_j}\}$ where Π_H is a projection on the space orthogonal to the groundstate of H_j, and we connect two neighboring projections by a line. We prove in claim 6, using Kitaev's phase estimation algorithm, that the fact that H_j is simulatable implies that so is Π_{H_j}. Also, as projections, Π_{H_j} have bounded norms, $||\Pi_{H_j}|| \leq 1$. It follows then, by the results mentioned in Section 3, that all the Hamiltonians on the path connecting these projections are simulatable, as convex combinations of simulatable Hamiltonians.

We now have to show the Hamiltonians on the path have non negligible spectral gap. By definition Π_{H_j} has a spectral gap equal to 1. It remains to show, however, that the Hamiltonians on the line connecting Π_{H_j} and $\Pi_{H_{j+1}}$ have large spectral gaps, which we prove in the simple Claim 7.

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

We now turn to the proofs of claims 6 and 7.

Claim 6. (Hamiltonian-to-projection lemma). *Let H be a Hamiltonian on n qubits such that e^{-iH} can be approximated to within arbitrary polynomial accuracy by a polynomial quantum circuit, and let $\|H\| \leq m = poly(n)$. Let $\Delta(H)$ be non negligible, and larger than $1/n^c$, and further assume that the groundvalue of H is 0. Then the projection Π_H, is simulatable.*

Proof. Kitaev's phase estimation algorithm [35, 40] gives a good approximation of the eigenvalue associated with a given eigenstate. This implies that when the spectral gap of H is non-negligible we can decide with exponentially good confidence whether an eigenstate has the lowest eigenvalue or a larger eigenvalue. We then run the following algorithm:

- Apply Kitaev's phase estimation algorithm to write down one bit of information on an extra qubit: whether an input eigenstate of H is the ground state or orthogonal to it.
- Apply a phase shift of value e^{-it} to this extra qubit, conditioned that it is in the state $|1\rangle$ (if it is $|0\rangle$ we do nothing). This conditional phase shift corresponds to applying for time t a Hamiltonian with two eigenspaces, the ground state and the subspace orthogonal to it, with respective eigenvalues 0 and 1, which is exactly the desired projection Π_H.
- Finally, to erase the extra qubit written down, we reverse the first step and uncalculate the information written on that qubit using Kitaev's phase estimation algorithm again.

□

We will also use the following basic but useful claim regarding the convex combination of two projections. For a vector $|\alpha\rangle$, the Hamiltonian $H_\alpha = I - |\alpha\rangle\langle\alpha|$ is the projection onto the subspace orthogonal to α. We prove:

Claim 7. *Let $|\alpha\rangle, |\beta\rangle$ be two vectors in some subspace, $H_\alpha = I - |\alpha\rangle\langle\alpha|$ and $H_\beta = I - |\beta\rangle\langle\beta|$. For any convex combination $H_\eta = (1-\eta)(I-|\alpha\rangle\langle\alpha|)+\eta(I-|\beta\rangle\langle\beta|, \quad \eta \in [0,1]$, of the two Hamiltonians H_α, H_β, $\Delta(H_\eta) \geq |\langle\alpha|\beta\rangle|$.*

Proof. To prove this, we observe that the problem is two dimensional, write $|\beta\rangle = a|\alpha\rangle + b|\alpha^\perp\rangle$, and write 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

$$\left(\begin{array}{cc} \eta|a|^2 + (1-\eta) & \eta ab^* \\ \eta a^* b & \eta|b|^2 \end{array}\right). \tag{17}$$

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 $|a|$. $\square$

We use the tools we have developed to prove the equivalence of standard and adiabatic state generation complexity, and for generating interesting Markov chain states. We start with the equivalence result.

7 Equivalence of Standard and Adiabatic State Generation

Theorem 2 asserts that any quantum state that can be efficiently generated in the quantum circuit model can also be efficiently generated by an adiabatic state generation algorithm, and vice versa. We already saw the direction from adiabatic state generation to quantum circuits. To complete the proof of Theorem 2 we now show the other direction.

Claim 8. *let $|\phi\rangle$ be the final state of a quantum circuit C with M gates, then there is a quantum adiabatic state generator which outputs this state, of complexity poly(n, M).*

Proof. W.l.o.g. the circuit starts in the state $|0\rangle$. We first modify the circuit so that the state does not change too much between subsequent time steps. The reason we need this will become apparent shortly. To make this modification, let us assume for concreteness that the quantum circuit C uses only Hadamard gates, Toffoli gates and Not gates. This set of gates was recently shown to be universal by Shi [46], and a simplified proof can be found in [6] (our proof works with any universal set with obvious modifications.) We replace each gate g in the circuit by two $\sqrt{g}$ gates. For $\sqrt{g}$ we can choose any of the possible square roots arbitrarily, but for concreteness we notice that Hadamard, Not and Toffoli gates have ± 1 eigenvalues, and we choose $\sqrt{1} = 1$ and $\sqrt{-1} = i$. We call the modified circuit C'. Obviously C and C' compute the same function.

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

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

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

The spectral gaps are large. The Hamiltonians $H_x(j)$ for integer $0 \leq j \leq M'$, have non-negligible spectral gaps, since they are projections. We claim that for any state β and any gate $\sqrt{g}$, $|\langle\beta|\sqrt{g}|\beta\rangle| \geq \frac{1}{\sqrt{2}}$. Indeed, represent β as $a_1v_1 + a_2v_2$ where v_1 belongs to the 1-eigenspace of $\sqrt{g}$ and v_2 belongs to the i-eigenspace of $\sqrt{g}$. We see that $|\langle\beta|\sqrt{g}|\beta\rangle| = ||a_1|^2 + i|a_2|^2|$. As $|a_1|^2 + |a_2|^2 = 1$, a little algebra shows that this quantity is at least $\frac{1}{\sqrt{2}}$. In particular, setting $\beta = \alpha_x(j)$ we see that $|\langle\alpha_x(j)|\alpha_x(j+1)\rangle| \geq \frac{1}{\sqrt{2}}$. It therefore follows by claim 7 that all the Hamiltonians on the line between $H_x(j)$ and $H_x(j+1)$ have spectral gaps larger than $\frac{1}{\sqrt{2}}$.

The Hamiltonians are simulatable. Given a state $|y\rangle$ we can

- Apply the inverse of the first j gates of C',
- If we are in state $|x,0\rangle$ apply a phase shift $e^{-i\delta}$, and
- Apply the first j gates of C'

which clearly implements $e^{-i\delta H_x(j)}$.

Adiabatic Condition is Satisfied. We have $\frac{dH}{ds}(s_0) = lim_{\zeta \to 0} \frac{H(s_0+\zeta) - H(s_0)}{\zeta}$. We ignore the finitely many points $s = \frac{j}{M'}$ where j is an *integer* in $[0, M']$. For all other points s, when ζ goes to 0 both $H(s_0+\zeta)$ and $H(s_0)$ belong to the same interval. Say they belong to the j'th interval, $s_0 = \frac{j+\eta}{M'}$, $0 < \eta < 1$. Then,

$$\begin{aligned} H(s_0) &= (1-\eta)H_x(j) + \eta H_x(j+1) \\ H(s_0+\zeta) = H(\frac{j+\eta+M'\zeta}{M'}) &= (1-\eta-M'\zeta)H_x(j) + (\eta+M'\zeta)H_x(j+1) \end{aligned}$$

It follows that $H(s_0+\zeta) - H(s_0) = M'\zeta H_x(j+1) - M'\zeta H_x(j)$ and $\frac{dH}{ds}(s_0) = M' \cdot [H_x(j+1) - H_x(j)]$. We conclude that $||\frac{dH}{ds}|| \leq 2M'$ and to satisfy Equation (14) we just need to pick $T = O(\frac{M'}{\epsilon})$.

□

8 Quantum State Generation and Markov Chains

Finally, we show how to use our techniques to generate interesting quantum states related to Markov chains.

8.1 Markov chain Background

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

A Markov chain M has eigenvalues between -1 and 1. A Markov chain is said to be *rapidly mixing* if starting from any initial distribution, the distribution after polynomially many time steps is within ϵ total variation distance from the limiting distribution π. [8] shows that a Markov chain is rapidly mixing if and only if its second eigenvalue gap is non negligible, namely bounded from below by $1/poly(n)$. A Markov chain is *reversible* if for the limiting distribution π it holds that $M[i,j] \cdot \pi_i = M[j,i] \cdot \pi_j$. We note that any symmetric Markov chain M is reversible. Also, for an ergodic, reversible Markov chain M, $\pi_i > 0$ for all i. For more background regarding Markov chains, see [38] and references therein; For more background regarding approximate counting algorithms see [33].

8.2 Reversible Markov chains and Hamiltonians

For a reversible M we define

$$H_M = I - Diag(\sqrt{\pi_i}) \cdot M \cdot Diag(\frac{1}{\sqrt{\pi_j}}) \tag{18}$$

A direct calculation shows that M is reversible iff H_M is symmetric. In such a case we call H_M the *Hamiltonian corresponding to M*. The properties of H_M and M are very much related:

Claim 9. *If M is a reversible Markov chain, we have:*

- *H_M is a Hamiltonian with $||H_M|| \le 2$.*
- *The spectral gap of H_M equals the second eigenvalue gap of M.*
- *If π is the limiting distribution of M, then the ground state of H_M is $\alpha(H_M) = |\pi\rangle \stackrel{\text{def}}{=} \sum_i \sqrt{\pi(i)}\, |i\rangle$.*

Proof. If M is reversible, H_M is Hermitian and hence has an eigenvector basis. As $I - H_M = \sqrt{\Pi} M \sqrt{\Pi}^{-1}$ both $I - H_M$ and M have the same spectrum. It follows that if the eigenvalues of H_M are $\{\lambda_r\}$ then the eigenvalues of M are $\{1 - \lambda_r\}$. As a Markov chain M has eigenvalues between -1 and 1 and the first two items of the claim follow.

Further, if v_r is an eigenvector of H_M with eigenvalue λ_r, then $Diag(\sqrt{\pi})v_r$ is the corresponding left eigenvector of M with eigenvalue $1 - \lambda_r$. In particular, $Diag(\sqrt{\pi})\alpha(H_M) = \pi(M)$. It therefore follows that $\alpha(H_M)_i = \sqrt{\pi_i}$, or in short $\alpha(H_M) = |\pi\rangle$. □

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

8.3 Simulating H_M

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

Definition 8. *A reversible Markov chain is strongly samplable if it is:*

- *row computable, and,*
- *Given $i, j \in \Omega$, there is an efficient way to approximate $\frac{\pi_i}{\pi_j}$.*

Row computability holds in most interesting cases but the second requirement is quite restrictive. Still, we note that it holds in many interesting cases such as all Metropolis algorithms (see [28]). It also trivially holds for symmetric M, where the limiting distribution is uniform.

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 row-computable. As H_M has bounded norm, the sparse Hamiltonian lemma implies:

Corollary 1: *If a Markov chain M is strongly samplable then H_M is simulatable.*

8.4 From Markov Chains to Quantum Generation

We now consider sequences of Markov chains and define:

Definition 9. (Slowly Varying Markov Chains). *Let $\{M_t^n\}_{t=1}^{T=T(n)}$ be a sequence of Markov chains on Ω, $|\Omega| = N = 2^n$. Let π_t^n be the limiting distribution of M_t^n. We say the sequence is* slowly varying *if for all $c > 0$, 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 3 precisely.

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

Proof. We already saw the Hamiltonians H_{M_t} are simulatable and have bounded norm. Also, as the Markov chains in the sequence are rapidly mixing, they have non-negligible spectral gaps, say $\frac{1}{n^b}$, and therefore so do the Hamiltonians H_{M_t}. To complete the proof we show that the inner product between the groundstates of subsequent Hamiltonians is non negligible, and then the theorem follows from the jagged path lemma. Indeed, $\langle\alpha(H_{M_t})|\alpha(H_{M_{t+1}})\rangle = \langle\pi_t|\pi_{t+1}\rangle = \sum_i \sqrt{\pi_t(i)\pi_{t+1}(i)} \geq 1 - |\pi_t - \pi_{t+1}| \geq \frac{1}{n^c}$ and therefore is non-negligible. The claim now follows from Lemma 2. □

Essentially all Markov chains that are used in approximate counting that we are aware of meet the criteria of the theorem. The following is a partial list of states we can generate using Theorem 1, where the citations refer to the approximate counting algorithms that we use as the basis for the quantum generation algorithm:

1. Uniform superposition over all perfect matchings of a given bipartite graph [32].
2. All spanning trees of a given graph [12].
3. All lattice points contained in a high dimensional convex body satisfying the conditions of [9].
4. Various Gibbs distribution over rapidly mixing Markov chains using the Metropolis filter [38].
5. Log-concave distributions [9].

We note that most if not all of these states can be generated using other simpler techniques. However our techniques do not rely on self reducibility, and are thus qualitatively different and perhaps extendible in other ways. We illustrate our technique with the example of how to generate all perfect matchings of a given bipartite graph. We also note that if we could relax the second requirement in Definition 8 the techniques in this section could have been used to give a quantum algorithm for graph isomorphism.

8.5 An Example: Generating All Perfect Matchings of a Bipartite Graph

In this subsection we heavily rely on the work of Sinclair, Jerrum and Vigoda [32] who recently showed how to efficiently approximate the permanent of any matrix with non negative entries. It is well known that this can be done if one can efficiently sample a random matching of a given input bi-partite graph.[32] achieve the later using a sequence of Markov chains 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 apply a Metropolis random walk on the set of perfect and near perfect matchings (i.e. perfect matchings minus one edge) of the complete bipartite graph. Since [32] is interested in a given input bipartite graph which is a subgraph of the complete graph, they assign weights to the edges such that edges that do not participate in the input graph are slowly decreasing until the probability they appear in the final distribution practically vanishes. The weights of the edges are updated using data that is collected from running the Markov chain with the previous set of weights, in an adaptive way. The final Markov chain with the final parameters converges to a probability distribution which is essentially concentrated on the perfect and near perfect matchings of the input graph, where the probability of the perfect matchings is $1/n$ times that of the near perfect matching.

Now, we are interested not just in sampling a random matching of a given bipartite graph, but rather in generating the coherent superposition over all these matchings. For that we will first run the classical algorithm for sampling a random matching and (as a side effect!) we will get a sequence $M_1, \ldots, M_T$ of slowly varying Markov chains. We now want to apply Theorem 1. It is easy to check that the Markov chains M_t being used in [32] are all strongly samplable, since they are Metropolis chains. Moreover, the sequence of Markov chains is slowly varying. 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 super-position 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 super-position 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 of the low-dimension.
- 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 to generate the limiting distribution of the final Markov chain. We then measure to see whether the matching is perfect or not, and with non negligible probability we project the state onto the uniform distribution over all perfect matchings of the given graph.

9 Acknowledgements

We wish to thanks Umesh Vazirani, Wim van Dam, Zeph Landau, Oded Regev, Dave Bacon, Manny Knill, Eddie Farhi, Ashwin Nayak and John Watrous for many inspiring discussions. In particular we thank Dave Bacon for an illuminating discussion which led to the proof of claim 8.

References

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

[2] D. Aharonov and A. Ta-Shma, Adiabatic Quantum State Generation and Statistical Zero Knowledge, STOC 2003, a longer version appears on quant-ph/0210077.

[3] D. Aharonov, W. van Dam, Z. Landau, S. Lloyd, J. Kempe and O. Regev, Quantum Computation is Equivalent to Adiabatic computation, in preparation, 2003

[4] D. Aharonov and O. Regev, A lattice problem in Quantum NP, FOCS 2003.

[5] D. Aharonov and O. Regev, in preparation.

[6] D. Aharonov, A simple proof that Toffoli and Hadamard are quantum universal, in preparation, 2002

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

[8] N. Alon and J. Spencer, The Probabilistic Method. John Wiley & Sons, 1991.

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

[10] Y. Avron and A. Elgart, Adiabatic Theorem without a Gap Condition, *Commun. Math. Phys.* 203, pp. 445-463, 1999

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

[12] R. Bubley and M. Dyer, Faster random generation of linear extensions, SODA 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, quant-ph/0209131

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

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

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

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

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

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

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

[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**, 472 (2001), quant-ph/0104129.

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

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

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

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

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

[27] S. Goldwasser, S. Micali and C. Rackoff, The Knowledge Complexity of Interactive Proof Systems, SIAM Journal on Computing, SIAM J Comput, Vol 18 Number 1, pp. 186–208, 1989

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

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

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

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

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

[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 Publishing, Boston, 1996.

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

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

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

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

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

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

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

[41] See A. Peres, Quantum Theory: Concepts and methods, Kluwer Academic Publishers, 1995

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

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

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

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

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

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

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

A Zeno effect approach to simulating adiabatic Generators

Proof. (of Claim 2) We concentrate on a time interval $[s_0, s_1]$, $s_0 < s_1$, where $H(\cdot)$ is continuous on $[s_0, s_1]$ and differentiable on (s_0, s_1). We denote $\eta_{max} = \max_{s \in (s_0,s_1)} ||\frac{dH}{ds}(s)||$

and $\Delta_{min} = \min_{s \in (s_0,s_1)} \Delta(H(s))$. We choose $R \geq \Theta(\frac{\eta_{max}^2}{\Delta_{min}^2}\epsilon)$. Notice that R is polynomially related to the schedule time T in the adiabatic condition.

We divide the interval $[0,1]$ to R time steps. At time step j, $j = 1, \ldots, R$, we measure the state with a projective, orthogonal measurement that has the ground state of $H(\frac{j}{R})$ as one of its answers. We begin at the state $\alpha(H(0))$.

We need to show our procedure can be implemented efficiently, i.e., that if H is simulatable and has a non negligible spectral gap, then such a measurement can be implemented efficiently. We also need to show our procedure is accurate, i.e., that under the condition of the adiabatic theorem, for the R we have chosen, with very high probability the final state is indeed $\alpha(H(1))$.

Accuracy :

We first bound the relative change of $H(s+\delta)$ with respect to $H(s)$. For $s, s+\delta \in [s_0, s_1]$, $H(s+\delta) - H(s) = \int_s^{s+\delta} \frac{dH}{ds}(s)ds$ and so $||H(s+\delta) - H(s)|| = ||\int_s^{s+\delta} \frac{dH}{ds}(s)ds|| \leq \int_s^{s+\delta} ||\frac{dH}{ds}(s)||ds \leq \eta_{max} \cdot \delta$.

Our next step is to claim that two Hamiltonians that are close to each other have close groundstates. This is captured in the following claim, that we prove later.

Claim 10. *Let H, J be two Hamiltonians $\|H - J\| \leq \eta$. Assume H, J have large spectral gaps: $\Delta(H), \Delta(J) \geq \Delta$ Then $|\langle \alpha(H) | \alpha(J) \rangle| \geq 1 - \frac{4\eta^2}{\Delta^2}$.*

Having that, we see that since $\|H(\frac{j+1}{R}) - H(\frac{j}{R})\| \leq \frac{\eta_{max}}{R}$, Claim 10 asserts that the probability for successful projection at the $j'th$ measurement, i.e. the probability that the outcome is indeed the groundstate, is $1 - O(\frac{\eta_{max}^2}{R^2\Delta_{min}^2})$. The probability we err at any of the R steps is therefore at most $O(\frac{\eta_{max}^2}{R\Delta_{min}^2})$ which is at most ϵ by our choice of R.

Efficiency : Kitaev's phase estimation algorithm [35, 40] gives a good approximation of the eigenvalue associated with a given eigenstate. We use it to give, with polynomially good confidence, a polynomially good approximation of the eigenvalue, and we then measure the eigenvalue. As the spectral gap is non-negligible, this in effect does an orthonormal measurement with the groundstate subspace as one possible answer, as desired. The procedure is polynomial because H is simulatable and we can efficiently approximate e^{-iHt} for every polynomial t.

□

We finish with the proof of Claim 10.

Proof. (of Claim 10) W.l.o.g we can assume H and J are positive, otherwise just add $C \cdot I$ to both matrices, for large enough constant C. This does not effect the spectral norm of the difference, the spectral gaps or the inner product between the groundstates.

Let $\{v_i\}$ be the H eigenvectors with eigenvalues $\lambda_1 < \ldots < \lambda_N$, and $\{u_i\}$, $\{\mu_i\}$ for J. Again, w.l.o.g, $0 = \lambda_1 \leq \mu_1$. Notice also that $\mu_1 \leq \lambda_1 + \eta$, because $\mu_1 = \min_{v:||v||=1} |Jv|$, and $|Jv_1| \leq |Hv_1| + |(J-H)v_1| \leq \lambda_1 + \eta$.

So, $|Jv_1| \leq \lambda_1 + \eta$. On the other hand, express $v_1 = au_1 + bu_\perp$, with $u_\perp \perp u_1$. Then, $|Jv_1| = |bJu_\perp + aJu_1| \;\geq\; |b| \cdot \mu_2 - |a| \cdot \mu_1 \;\geq\; |b| \cdot (\mu_1 + \Delta) - |a| \cdot \mu_1 \;\geq\; |b| \cdot (\lambda_1 + \Delta) - |a| \cdot (\lambda_1 + \eta)$. Setting $\lambda_1 = 0$ we get: $\eta \geq |b| \cdot \Delta - |a| \cdot \eta$. Let us denote $c = \frac{\Delta}{\eta}$. We see that $|a| \geq c|b| - 1$.

We now plug in $|a| = \sqrt{1-|b|^2}$, and square both sides of the inequality. We get $1 - |b|^2 \geq 1 - 2c|b| + c^2|b|^2$, i.e., $|b| \leq \frac{2c}{c^2+1} \leq \frac{2c}{c^2} = \frac{2}{c}$. Equivalently, $|\langle \alpha(H_1) | \alpha(H_2) \rangle| = |\langle v_1 | u_1 \rangle| = |a| = \sqrt{1-|b|^2} \geq 1 - \frac{4}{c^2} = 1 - \frac{4\eta^2}{\Delta^2}$ as desired. $\square$