universität
wien

DISSERTATION

Titel der Dissertation

On the Computational Power of Quantum Computers

verfasst von

Dipl.-Ing. Martin Schwarz

angestrebter akademischer Grad

Doktor der Naturwissenschaften (Dr. rer. nat.)

Wien, 2013

Studienkennzahl laut Studienblatt:	A 091 411
Dissertationsgebiet laut Studienblatt:	Physik
Betreuer:	Univ.-Prof. Dr. Frank Verstraete

Contents

Zusammenfassung **5**

Abstract **7**

Acknowledgements **9**

Introduction **11**

1 Preparing Projected Entangled-Pair States on a Quantum Computer **17**

1.1 Introduction . . . 18

1.2 Definitions and Result . . . 19

1.3 The Algorithm . . . 20

1.4 Analysis . . . 23

1.4.1 Bounding the transition probabilities . . . 23

1.4.2 Bounding the convergence rate . . . 24

1.4.3 Proof of Theorem 2 . . . 25

1.5 Contracting tensor networks on a quantum computer . . . 26

1.6 Conclusion . . . 27

2 Preparing Topological Projected Entangled-Pair States on a Quantum Computer **29**

2.1 Introduction . . . 30

2.2 Algorithm . . . 33

2.3 Analysis . . . 37

2.4 Conclusion . . . 39

3 An Information-Theoretic Proof of the Constructive Commutative Quantum Lovász Local Lemma **41**

3.1 Introduction . . . 42

3.2 The Algorithm . . . 44

3.2.1 Moser's classical algorithm . . . 44

3.2.2 The quantum algorithm . . . 46

3.3 Analysis . . . 48

3.4 Conclusions . . . 50
3.5 Acknowledgements . . . 51
3.6 Appendix . . . 51
3.6.1 Detailed algorithm and proof . . . 51
3.6.2 Proof of Theorem 11 . . . 56
3.6.3 Upper bound on N . . . 61

4 Simulating Quantum Circuits with Sparse Output Distributions 65
4.1 Introduction . . . 66
4.2 Main results: statements and discussion . . . 68
4.3 Proof outline and organization of the paper . . . 74
4.4 Approximate sparseness . . . 75
4.4.1 Basic definitions . . . 75
4.4.2 Basic properties . . . 76
4.4.3 Sparse distributions have large coefficients . . . 77
4.5 Additively approximable probability distributions . . . 78
4.5.1 Definition and basic properties . . . 78
4.5.2 Estimating large coefficients . . . 80
4.6 Algorithm for additively approximable, approximately sparse distributions . . . 81
4.7 Classical simulation of CT states . . . 84
4.8 Proofs of main results . . . 84
4.8.1 Proof of Theorem 20 . . . 84
4.8.2 Proof of Theorem 21 . . . 86
4.8.3 Proof of Theorem 22 and Theorem 23 . . . 86
4.8.4 Proof of Theorem 24 . . . 88
4.9 Further research . . . 89
4.10 Appendix . . . 89
4.10.1 Proof of lemma 27 . . . 89
4.10.2 Proofs of lemmas 32 and 33 . . . 89

Conclusions and Outlook 91

Bibliography 93

Lebenslauf 101

Zusammenfassung

Diese Dissertation beschäftigt sich mit der Frage, wie viel Rechenkraft Quantencomputer bieten können. Diese Frage geht in ihrer allgemeinsten Form weit über die aktuellen Möglichkeiten und Methoden der Komplexitätstheorie hinaus. Dennoch liefert die Komplexitätstheorie Beispiele von Problemen, die sogar für Quantencomputer als nicht effizient lösbar gelten. Andererseits können Quantencomputer vermutlich nicht effizient von klassischen Computern simuliert werden. Wir wollen mit dieser Arbeit zum Verständnis des dazwischen liegenden Bereichs von Problemen, die effizient auf Quantencomputern, aber nicht effizient auf klassischen Computern berechnet werden können, beitragen. Um besseren Einblick in die Fragestellung zu erlagen, entwickeln wir in dieser Dissertation klassische und quantenmechanische Algorithmen zur Lösung von Problemen, die *unter einschränkenden Annahmen* a priori als nicht effizient lösbar geltende Probleme dennoch effizient lösen. Wir erforschen in diesem Sinne also die Grenzen des auf Quantencomputern und klassischen Computern Machbaren, in dem wir Annahmen identifizieren, die dazu geeignet sind grundsätzlich als schwer geltende Probleme in den Bereich des Machbaren zu bringen. Im Speziellen entwickeln wir einen Quantenalgorithmus, der sogenannte *Projective Entangled-Pair States* (PEPS) effizient herstellen kann, sobald *Injektivität* und *Wohlkonditierung* des PEPS gegeben ist. In einem zweiten Schritt erweitern wir diesen Algorithmus auf sogenannte *G-injektive PEPS*, einer wesentlich allgemeineren Klasse von Quantenzuständen, die auch die exotische Eigenschaft der *topologischen Ordnung* besitzen können. Weiters entwickeln wir einen Quantenalgorithmus, der das im Allgemeinen für Quantencomputer als schwer geltende Problem den Grundzustand von *lokalen Hamiltonoperatoren* zu erzeugen löst, sofern diese die Voraussetzungen des (nicht-konstruktiven) *Quanten-Lovász-Local-Lemma* erfüllen und die Operatoren kommutieren. Unser Quantenalgorithmus kann in diesem Fall auch als unabhängiger, konstruktiver Beweis des Quanten-Lovász-Local-Lemmas betrachtet werden. Um die Grenze zu klassischen Computern zu erforschen, entwickeln wir einen klassischen Algorithmus, der das sehr allgemeine Problem Quantenschaltkreise zu simulieren unter gewissen Annahmen effizient lösen kann. Wäre es in voller Allgemeinheit lösbar, würden Quantencomputer keinen Vorteil bieten. Unser Algorithmus kann Quantenschaltkreise simulieren, die in ihrer Struktur jener des Quantenalgorithmus von Shor zur Faktorisierung ganzer Zahlen, der als exponentiell schneller als der schnellste bekannte klassische Faktorisierungsalgorithmus gilt, sehr ähnlich sind. Weiters treffen wir die Annahme, dass die erzeugte Wahrscheinlichkeitsverteilung *annähernd dünn-besetzt* ist. Unser Ergebnis impliziert daher, dass die von Algorithmen dieser Struktur erzeugte Wahrscheinlichkeitsverteilung notwendigerweise dicht-besetzt sein muss, um einen exponentiellen Geschwindigkeitsvorteil erzielen zu können.

Abstract

In this thesis we make progress in our understanding of the computational power made available by quantum computers. While answering this question in full generality in the framework of computational complexity theory appears beyond the reach of current methods, research in this direction has identified certain problems that are conjectured to be hard even for quantum computers, such as the Local Hamiltonian problem, or more general quantum state preparation problems. On the other hand, the generic problem of simulating quantum computers is conjectured to be hard for classical computers. We make progress in understanding the delineations among the classical and quantum models of computation by designing algorithms that explore these borders by identifying critical assumptions that allow us to efficiently solve certain problems which otherwise would be considered intractable. More specifically, towards exploring the power of quantum computers, we present a quantum algorithm that efficiently prepares quantum states specified by so-called projected entangled-pair states (or PEPS). While this problem is known to be hard in general even for quantum computers, assuming the PEPS is *injective* and *well-conditioned* allows us to prepare it efficiently. In a second step, we generalize this algorithm to efficiently prepare quantum states with *topological order* as specified by well-conditioned G-injective PEPS. A third quantum algorithm presented in this thesis prepares the zero-energy ground state of a certain class of Local Hamiltonians characterized by the non-constructive Quantum Lovász Local Lemma. While the lemma guarantees the existence of such a state, we present a quantum algorithm to efficiently prepare it assuming commutativity of the local projector terms. Finally, we explore the opposite direction towards the power of classical computers. We present a classical algorithm to simulate quantum circuits of a specific structure that is similar to Shor's integer factorization quantum algorithm (which is exponentially faster than any known classical factoring algorithm). Assuming approximate sparseness of the output distribution produced by quantum circuits of this structure allows us to simulate them classically. Thus we have discovered a region where quantum computers cannot retain their presumed computational advantage. Moreover, this result implies that no *exact* quantum algorithm of this structure can offer exponential speed-up, and that output distributions of such quantum circuits must necessarily have super-polynomially large support and must have additional structure (e.g. group structure) to allow for an exponential speed-up.

Acknowledgements

I would like to thank my supervisor Prof. Frank Verstraete for his support and encouragement during my graduate studies. Due to his enthusiasm for quantum physics and other fields like information theory or complexity theory I have had the unique opportunity to work in an inspiring interdisciplinary environment and learn many interesting things outside the main scope of my research.

I would like to thank my collaborators Toby S. Cubitt, David Pérez-García, Maarten van den Nest, and Kristan Temme for the countless hours of intense joint work during our research projects right up to paper submissions deadlines. I would also like to thank Toby and David for their hospitality in Madrid, as well as Maarten for his hospitality in Garching.

I would also like to thank Or Sattath for finding errors in several of our attempts to proof the general case of the constructive Quantum Lovász Local Lemma and for interesting discussions on this and related topics.

I would like to thank Daniel Nagaj for the many inspiring discussions on our shared topics of interest in quantum physics and computer science, and for proofreading many of my drafts.

I would also like to thank all my colleagues from the University of Vienna for the pleasant time we spent together during the last several years.

I would also like to thank the University of Vienna in general and the Faculty of Physics in particular for providing such an inspiring environment for research in theoretical as well as experimental quantum information science, featuring weekly talks and visits of leading researchers of the field. May there only be more permanent positions in the future!

I would especially like to thank Olivia, my parents, and my friends for their encouragement throughout the time of my graduate studies.

I gratefully acknowledge the generous support by the Austrian AMS Bildungskarenz program and Austrian FWF SFB project FoQuS (F4014).

Introduction

The informal yet fundamental question that motivates this work is:

> *How powerful is a quantum computer?*

In particular,

1. *What can a quantum computer compute more efficiently than a classical computer?*
2. *For what kinds of problems should we expect quantum computers to have an advantage?*

One approach towards these questions is computational complexity theory [Arora and Barak, 2009]. Complexity theory classifies (yes/no) decision problems into complexity classes, some of which are known to be tractable whereas others are *conjectured* to be intractable. For the sake of discussion, let us introduce informally some well-known classical complexity classes

- P – problems decidable deterministically in time polynomial in the input size (things we can do on an ordinary computer)
- BPP – problems decidable probabilistically with bounded error in time polynomial in the input size (things we can do on an ordinary computer with the help of randomness)
- NP – problems with solutions that are *verifiable* by a deterministic computer in time polynomial in the input size given a bit string as a proof (solutions are easy to check)
- PP – for an NP-complete problem, *count* whether more than one half or at most one half of all possible variable assignments are indeed solutions (not only if there is a solution)

While it is standard in computer science to assume P=BPP (e.g. [Impagliazzo and Wigderson, 1997]), a proof (or disproof) of the common conjecture about P $\neq$ NP is worth \$1.000.000 [Clay Mathematics Institute, 2000] and considered well beyond the reach of current methods (as are proofs of many other conjectured complexity class separations). Notably, constraint satisfaction problems (CSPs) over integer domains are contained in NP (e.g. finding a ground state of classical Ising spin glasses in three dimensions is NP-complete [Barahona, 1982]).

Throughout the past two decades, computational complexity theory has been extended to *quantum* computational complexity theory [Bernstein and Vazirani, 1993, Watrous, 2008], and quantum analogues of the relevant complexity classes have been defined:

- BQP – problems decidable by a quantum computer with bounded error in time polynomial in the input size (contains BPP, since quantum computers can easily simulate reversible classical computers)
- QMA – problems with solutions that are *verifiable* by a quantum computer in time polynomial in the input size given a quantum state as proof (e.g. given a ground state, measure its energy)

A formal complexity theoretic version of our fundamental question thus reads

$$\mathsf{BPP} \overset{?}{\neq} \mathsf{BQP}$$

As with many complexity class separation conjectures, a formal proof of this conjecture is considered beyond the reach of current methods. Nevertheless, formal evidence exists supporting the conjecture: if one assumes that the input is only available in the form of a black box that answers yes/no questions (*"queries"*), it is indeed possible to show an exponential advantage in terms of the number of queries we need to ask between quantum and classical computers [Simon, 1997]. Let us discuss further relations among these classes: Since it is known that quantum computers can efficiently simulate (deterministic and probabilistic) classical computers [Bernstein and Vazirani, 1993], it is clear that $\mathsf{P} \subseteq \mathsf{BPP} \subseteq \mathsf{BQP}$ and $\mathsf{NP} \subseteq \mathsf{QMA}$. Similar to the classical case, one conjectures $\mathsf{BQP} \neq \mathsf{QMA}$. Furthermore, it is known that $\mathsf{QMA} \subseteq \mathsf{PP}$ [Marriott and Watrous, 2005].

A complete problem for QMA is the Local Hamiltonian problem [Aharonov and Naveh, 2002, Kitaev et al., 2002], which asks to decide whether the ground state energy of a quantum Hamiltonian on a spin system specified by local interaction terms is below some low-energy bound or above some high-energy bound. Given a low-energy eigenstate of the Hamiltonian as witness, a quantum computer can indeed measure and thus *verify* the energy (e.g. using phase estimation [Nielsen and Chuang, 2000]). On the other hand, *preparing* such a ground state is considered hard even for a quantum computer, unless BPQ=QMA.[1] Since preparing ground states of quantum Hamiltonians is considered hard, this already demonstrates a "ceiling" for the type of problems for which a quantum computer may be expected to be useful. On the other hand, designing quantum algorithms that are nevertheless able to prepare such ground states (under certain additional assumptions) appears to us as a worthwhile endeavor exploring the upper limits of what a quantum computer can do efficiently.

While the complexity-theoretic approach tries to answer the fundamental question in full generality, another approach to gain insight – *which we will pursue in this thesis* – is to explicitly design efficient quantum algorithms solving problems for which no efficient classical counterpart is known. Since only rather few (classes of) quantum algorithms outperforming

[1] The statement is conjectured to be true even for *classical* Local Hamiltonians, unless $\mathsf{NP} \subseteq \mathsf{BQP}$, even though quantum computers have a quadratic advantage over classical computers in this case due to Grover's algorithm [Grover, 1996].

classical ones are known [Jordan, 2013], any fundamentally new algorithm that is not obviously related to the known ones may allow us to gain further intuition about the types of problems where quantum computers may excel.

Historically the field of quantum computing has thrived because of two important examples of quantum algorithms (and their generalizations) that outperform their best known classical counterparts: Shor's algorithm for integer factorization [Shor, 1999], and Grover's algorithm for unstructured search problems [Grover, 1996]. Ultimately, the investment required to construct real large-scale quantum computers will only be justified by the applications they enable (if these are expected to remain beyond the reach of classical techniques for the foreseeable future.) Clearly, to identify novel applications, novel algorithms are required.

Many quantum algorithms known today simply yield faster *quantum* algorithms for solving *classical* problems. But then, there are distinctly *quantum* problems that almost by definition only a quantum computer may solve, e.g. producing a particular quantum state as output. Quantum computers can receive quantum states as input and produce quantum states as output. Since a classical computer can neither receive nor produce a (non-classical) quantum state in principle (and simulating quantum computers in general appears to be hard), we consider this a particularly interesting non-classical feature of the quantum model of computation, which leaves behind the paradigm of classical decision problems, and thus may have the potential to shed new light on our fundamental question.

Of course, a reductionist might argue that any quantum state that a quantum algorithm may prepare, must be eventually measured producing a classical outcome in order to be useful. Therefore, quantum computers do not really leave the paradigm of classical input/classical output behind. In contrast, we think that by *not* considering quantum states as meaningful output in their own right (i.e. without specifying any particular measurement *a priori*, but leaving the doubly exponential number of efficient measurement choices open to the user of the algorithm), such a reduction misses out on a potentially important opportunity to demonstrate the utility of a quantum computer. Immediate further questions arise: what are interesting quantum states to prepare? How should they be specified? Are there complexity-theoretic limitations to the class of states we may expect a quantum computer to prepare? Certainly, Richard Feynman's original idea [Feynman, 1982] of using quantum computers to efficiently simulate quantum mechanics – the very reason he *invented* the concept of a quantum computer! – suggests that the capability to prepare physically relevant quantum states in order to study their properties by measuring arbitrary observables is certainly useful to, e.g. chemists, material scientists, or the pharmaceutical industry.

While the discussion above focuses on the utility of quantum computers, considering our fundamental question it is equally important to investigate the opposite direction: are quantum computers always hard to simulate or can we identify situations where quantum computers lose their conceived advantage? In what situations can classical computers efficiently simulate quantum computers? This question has been addressed by a series of works that identify restrictions on quantum circuits that render them classically simulable (e.g. [Gottesman, 1999, Valiant, 2002, Van den Nest, 2010, 2011, 2012] and several others). Of special interest are simulation methods capable of simulating circuits with structures like those found in quantum algorithms with expected exponential advantage over classical ones (e.g. Shor's). By trying to simulate these specific quantum algorithms we hope to get closer to discovering the root cause of the exponential advantage of quantum computers.

Outline and Summary of the results

We address our fundamental question by presenting three novel quantum algorithms preparing classes of quantum states which *a priori* might have been considered hard to prepare for a quantum computer, thus demonstrating the power of quantum computers. Furthermore, we present a classical simulation algorithm which is able to simulate a certain large class of quantum circuits under an additional assumption on the output state produced by the algorithm, yielding a surprising demonstration of the power of *classical* computation. On the other hand this means the discovery of a region where quantum computers cannot retain their presumable advantage. Figure 1 illustrates these results.

Chapter 1: In this chapter we present a quantum algorithm to prepare injective Projected Entangled-Pair States, or PEPS, on a quantum computer. Open tensor networks representing quantum states such as PEPS [Verstraete and Cirac, 2004] have been proposed as a class of quantum states especially suited to describe the ground states of local Hamiltonians in quantum many-body physics. PEPS are a higher-dimensional generalization of the one-dimensional Matrix Product States [Rommer and Östlund, 1997], or MPS, for which local expectations values can be efficiently calculated classically, and which can be efficiently generated on a quantum computer. For PEPS, on the other hand, much less is known. [Verstraete et al., 2006] ask whether such states could be even created on a quantum computer. Since an algorithm that would allow to prepare any PEPS would allow for the solution of PP-complete problems [Schuch et al., 2007], it is unlikely that this is possible without any additional restrictions. We resolve this question in the positive for the special case of injective, well-conditioned PEPS projectors. In particular, the run-time of our quantum algorithm scales polynomially with the inverse of the minimum condition number of the PEPS projectors and, essentially, with the inverse of the spectral gap of the PEPS' parent Hamiltonian.

Chapter 2: In this chapter we extend the results of Chapter 1 to include states with *topological order*. While injective PEPS are inherently unable to represent topologically ordered states, we consider the more general class of G-injective PEPS [Schuch et al., 2010] to overcome this limitation. 'G-injectivity' is a substantially weaker requirement than injectivity, which explicitly allows for topological order. We show that any G-injective PEPS can be prepared on a quantum computer in polynomial time, when the spectral gap of the associated parent Hamiltonian scales at most inverse-polynomially in the system size. A compelling example of the significance of this result is the very recently proven fact that the resonating valence bond (RVB) state in the Kagome lattice (conjectured to be a topological spin liquid), is a $\mathbb{Z}_2$-injective PEPS, with numerical evidence that the gap assumption is also verified [Poilblanc et al., 2012]. Our result therefore gives one way in which the RVB state (and other topological states) can be prepared efficiently on a general quantum simulator.

Chapter 3: In this chapter we present a quantum algorithm to prepare the zero-energy ground state of a certain class of Local Hamiltonians characterized by the Quantum Lovász Local Lemma (QLLL) [Ambainis et al., 2012], a quantum generalization of the well-known Lovász Local Lemma. It states that, if a collection of subspace constraints are "weakly dependent", there necessarily exists a state satisfying all of the constraints. It implies e.g. that certain instances of the k-QSAT problem are necessarily satisfiable, or that many-body systems with "not too many" interactions are never frustrated. However, the QLLL only asserts existence; it says nothing about how to find the quantum state that satisfies the constraints. Inspired by Moser's breakthrough classical result [Moser, 2009], we present a constructive version of the QLLL in the setting of commuting constraints, proving that a simple quantum algorithm efficiently prepares the sought quantum state.

Chapter 4: In this chapter we show that several quantum circuit families can be simulated efficiently classically if it is promised that their output distribution is approximately sparse i.e. the distribution is close to one where only a polynomially small, a priori unknown subset of the measurement probabilities are nonzero. Classical simulations are thereby obtained for quantum circuits which — *without the additional sparsity promise* — are considered hard to simulate. Our results apply in particular to a family of Fourier sampling circuits (which have structural similarities to Shor's factoring algorithm [Shor, 1999]) but also to several other circuit families, such as IQP circuits [Shepherd and Bremner, 2009], with the assumption of sparsity. Our results provide examples of quantum circuits that cannot achieve exponential speed-ups due to the presence of too much destructive interference i.e. too many cancelations of amplitudes. The crux of our classical simulation is an efficient algorithm for approximating the significant Fourier coefficients of a class of states called computationally tractable states [Van den Nest, 2011]. The latter result may have applications beyond the scope of this work. In the proof we employ and extend sparse approximation techniques, in particular the Kushilevitz-Mansour algorithm [Kushilevitz and Mansour, 1991], in combination with probabilistic simulation methods for quantum circuits.

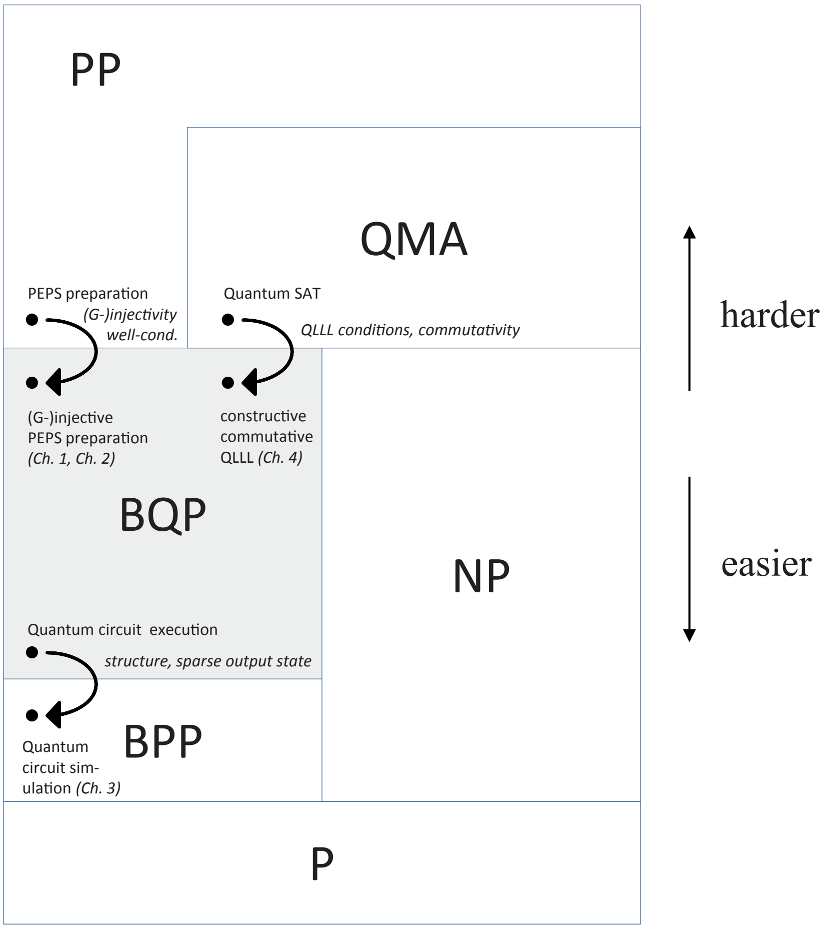

Figure 1: A graphical guide to this thesis. We depict the complexity classes introduced in this section. Classes indicated by boxes higher up contain the complexity classes further below and are conjectured to be strictly harder than those below them. We are especially interested in the power of BQP. We give examples of algorithms *right at the borders* of BQP in the following sense. In Chapter 1 we introduce a quantum algorithm for preparing PEPS. While the general problem of preparing PEPS is known to be PP-complete, we show that the (quite generic) assumptions of *injectivity* and *well-conditioning* suffice to put the problem of preparing PEPS into BQP. In Chapter 2 we extend the previous result and show that the even weaker assumption of *G-injectivity* suffices (together with *well-conditioning*) to prepare the larger class of topologically ordered G-injective PEPS. In Chapter 3 we consider the border to QMA and find that the quantum satisfiability problem (Quantum SAT), a special case of the Local Hamiltonian problem, can be solved in BQP, if the local terms commute and do not overlap too much (i.e. they satisfy the so-called QLLL conditions). Finally, in Chapter 4 we move on to examine the power of BPP and find that quantum circuits with a structure similar to Shor's algorithm can indeed be simulated classically, if one makes a crucial approximate sparseness assumption about the output distribution.

Chapter 1

Preparing Projected Entangled-Pair States on a Quantum Computer

Synopsis:

We present a quantum algorithm to prepare injective PEPS on a quantum computer, a class of open tensor networks representing quantum states. The run-time of our algorithm scales polynomially with the inverse of the minimum condition number of the PEPS projectors and, essentially, with the inverse of the spectral gap of the PEPS' parent Hamiltonian.

Based on:
M. Schwarz, K. Temme, and F. Verstraete,
Phys. Rev. Lett. **108**, 110502 (2012)

Changes compared to published version:
Section 1.5 has not been published due to the journal's length restrictions.

1.1 Introduction

Projected Entangled Pair States, or PEPS [Verstraete and Cirac, 2004], have been proposed as a class of quantum states especially suited to describe the ground states of local Hamiltonians in quantum many-body physics. PEPS are a higher-dimensional generalization of the one-dimensional Matrix Product States [Rommer and Östlund, 1997], or MPS, for which many interesting properties have been proven: For example, MPS provably approximate the ground state of 1D local Hamiltonians with constant spectral gap [Hastings, 2007, Verstraete and Cirac, 2006], exhibit an area law [Hastings, 2007] as well as an exponential decay of two-point correlation functions. Furthermore, for each MPS with the *injectivity* property [Perez-Garcia et al., 2008], a parent Hamiltonian can be constructed with this MPS as its unique ground state. MPS can also be prepared efficiently on a quantum computer [Schön et al., 2005]. PEPS however form a much richer class of states, and can e.g. represent critical systems and systems with topological quantum order [Verstraete et al., 2006]. It is conjectured that all ground states of gapped local Hamiltonians in higher dimensions can be represented faithfully as PEPS, and although there are strong indications for this fact, this has not been proven. What is clear, however, is the fact that one can also construct parent Hamiltonians for them [Perez-Garcia et al., 2008], and the PEPS will be the unique ground states of those Hamiltonians if the PEPS obeys the so-called injectivity condition [Perez-Garcia et al., 2008]. Many physically relevant classes of PEPS on lattices are known to be almost always injective, including e.g. the 2D AKLT state [Perez-Garcia et al., 2008]. A particularly interesting subclass of PEPS is the one that consists of all those states whose parent Hamiltonian have a gap that scales at most as an inverse polynomial as a function of the system size: in that case, a local observable (i.e. the local Hamiltonian) allows to distinguish the state from all other ones, as the ground state always has energy zero by construction. It was an open problem [Verstraete et al., 2006] whether such states could however be even created on a quantum computer, as an algorithm that would allow to prepare any PEPS would allow for the solution of PP-complete problems [Schuch et al., 2007].

In this article we show how well-conditioned injective PEPS can be prepared on a quantum computer efficiently. The key idea of our approach is to grow the PEPS step by step. We demand that not only our final PEPS is the unique ground state of its parent local Hamiltonian, but also that there exists a sequence of partial sums of the local terms of the parent Hamiltonian, such that each partial sum has a unique ground state of its own. Based on this assumption, the algorithm starts with a physical realization of the valence bond pairs as its initial state and iteratively performs entangling measurements on the virtual particles to map virtual degrees of freedom to physical ones, just as in the definition of the PEPS. The PEPS is called injective, iff this map is (left) invertible which can only be the case if the dimension of the physical space is actually at least as large as the dimension of the virtual space at each vertex. Preparing a PEPS by measurements may seem to require post-selection to project onto the right measurement outcome. To overcome this issue we use the Marriott-Watrous trick [Marriott and Watrous, 2005, Nagaj et al., 2009] of undoing a measurement based on Jordan's lemma [Jordan, 1875] and combine it with the uniqueness property of injective PEPS [Perez-Garcia et al., 2008] to prepare the required eigenstates. A key element that contributes to the success of this algorithm is the fact that the measurements are not done locally, such as in the framework of dissipative quantum state engineering [Verstraete et al., 2009], but globally by running a phase estimation algorithm that singles out the ground subspace; a similar approach was used in the context of the quantum Metropolis sampling algorithm [Temme et al., 2011]. Alternatively, methods for eigenpath traversal [Aharonov and Ta-Shma, 2007, Boixo et al., 2010] can also be applied [Boixo, 2011].

1.2 Definitions and Result

Before stating the result, we review the definition of PEPS and their essential properties. Recall [Perez-Garcia et al., 2008, Verstraete and Cirac, 2004] that PEPS are quantum states defined over an arbitrary graph $G = (V, E)$ such that quantum systems of local dimension d are assigned to each vertex. We construct the PEPS by assigning to each edge $e \in E$ a maximally entangled state $\sum_{i=1}^{D} |ii\rangle$. In this way, a vertex $v \in V$ with degree k gets associated with k *virtual* D-dimensional systems. Finally, a map $A^{(v)} : \mathbb{C}^D \otimes \mathbb{C}^D \otimes \cdots \mathbb{C}^D \mapsto \mathbb{C}^d$ is applied to each vertex, taking the k *virtual* D-dimensional systems to a single *physical* d-dimensional system. The linear map $A^{(v)}$ is usually called the PEPS "projector" and is parameterized by tensors $A_i^{(v)}$ as follows:

$$A^{(v)} = \sum_{i=1}^{d} \sum_{j_1,\ldots,j_k=1}^{D} A_{i,j}^{(v)} |i\rangle \langle j_1, \ldots, j_k| \tag{1.1}$$

where $A_i^{(v)}$ is a tensor with k indices. The PEPS can now be written as

$$|\psi\rangle = \sum_{i_1,\ldots,i_n=1}^{d} \mathcal{C}[\{A_{i_v}^{(v)}\}_v] |i_1, \ldots, i_n\rangle \tag{1.2}$$

where $\mathcal{C}$ means the contraction of all tensors $A_i^{(v)}$ according to the edges of the graph. In the most general case the virtual index dimension D as well as the physical index dimension d may also depend on the edges e and vertices v of the interaction graph, but we suppress this detail in favor of simplicity. Note, that w.l.o.g. $A^{(v)} \geq 0$ may be assumed, since for arbitrary $\tilde{A}^{(v)}$ we can choose a local basis by performing a polar decomposition, i.e.

$$\tilde{A}^{(v)} = U^{(v)} A^{(v)} \tag{1.3}$$

with $U^{(v)}$ unitary and $A^{(v)} \geq 0$.

A PEPS $|\psi\rangle$ is called *injective* [Perez-Garcia et al., 2008], if each PEPS projector $A^{(v)}$ has a left inverse. For some PEPS this may only be true, after some local contractions of a constant number of PEPS tensors $A^{(v)}$ according to the interaction graph of the PEPS forming new projectors $\hat{A}^{(v)}$ for which the condition above holds. Since this blocking can be performed efficiently for constant degree graphs, we may assume for the remainder of this paper, that it has already been performed, such that each individual $A^{(v)}$ in our input is already injective by itself. Note, that the existence of a left inverse allows us to strengthen the assumption $A^{(v)} \geq 0$ w.l.o.g. to $A^{(v)} > 0$ for all v.

For injective PEPS, there is a simple construction [Perez-Garcia et al., 2008] of a 2-local parent Hamiltonian, such that the injective PEPS is its unique, zero-energy ground state. This construction gives a parent Hamiltonian for a quantum system consisting of n particles with d-dimensional Hilbert spaces.

Definition 1. *Let H be a Hermitian matrix with $\lambda_0 < \lambda_1$ its smallest and second smallest eigenvalues. Then we call $\Delta(H) = \lambda_1 - \lambda_0$ the* spectral gap *of H. For any matrix A, the* condition number *$\kappa(A)$ is defined as $\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$, where $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ are the largest and smallest singular values of A, respectively.*

We are now in a position to state the performance of our algorithm as our main theorem:

Theorem 2. *Let $G = (V, E)$ be an interaction graph with bounded degree and some total order defined on V. Let $\{A^{(v)}\}_{v \in V_{[t]}}$ be a set of injective PEPS projectors of dimension $d \times D^k$ associated with each v in V up to vertex t (according to the total vertex order) describing a sequence of PEPS $|\psi_t\rangle$, and let $\kappa = \max\limits_{v \in V} \kappa(A^{(v)})$ be the largest condition number of all PEPS projectors. Let $\Delta = \min_t \Delta(H_t)$, where $\Delta(H_t)$ is the spectral gap of the parent Hamiltonian H_t of the PEPS $|\psi_t\rangle$. Then there exists a quantum algorithm generating the final PEPS $|\psi_{|V|}\rangle$ with probability at least $1 - \varepsilon$ in time $\tilde{O}(\frac{|V|^2|E|^2\kappa^2}{\varepsilon\Delta} + |V|kd^6)$.*

1.3 The Algorithm

Conceptually, PEPS are constructed by first preparing entangled pair states $|\psi\rangle = \sum_i |ii\rangle$ for each edge of the interaction graph describing the PEPS, and then projecting the k *virtual* indices associated with each vertex to a single *physical* index. While this construction is usually considered only a theoretical device, the proposed algorithm is indeed simulating the above construction for the case of injective PEPS with gapped Hamiltonians. This entails making the virtual indices physical as well.

Figure 1.1 presents our algorithm in pseudo-code. We proceed by explaining each step in detail. PEPS construction starts in step 2 by distributing maximally entangled states of the desired bond dimension according to the interaction graph $G = (E, V)$. The resulting system is the zero-energy ground state of a simple Hamiltonian H_0 consisting purely of terms projecting onto $H_e = \mathbb{1} - \frac{1}{d} \sum_{i,j=1}^{d} |ii\rangle\langle jj|$ for each edge of the interaction graph (step 3). Note, that this simple Hamiltonian is gapped.

Input: Interaction graph $G = (V, E)$ with degree bound k and total vertex order. For each $v \in V$, $d \times D^k$-matrices $A^{(v)}$ as PEPS projectors. Acceptable probability of failure ε.
Output: PEPS $|\psi\rangle$ with probability at least $1 - \varepsilon$.

```
1. $t \leftarrow 0$
2. $|\psi_t\rangle \leftarrow$ entangled pair for each edge $e \in E$.
3. $H_t = \sum_{e \in E} H_e$
4. For each $v \in V$ according to total order:
   (a) $H_{t+1} \leftarrow H_t$
   (b) For each neighbor $v' \in V$ of $v$:
       • remove term $H_t^{(v,v')}$ from $H_{t+1}$
       • compute parent Hamiltonian term $H_{t+1}^{(v,v')}$ using $A^{(v)}$
       • add term $H_{t+1}^{(v,v')}$ to $H_{t+1}$
   (c) Add $H_{phy}^{(v)} = c(\mathbb{1} - P_{phy}^{(v)})$ to $H_{t+1}$
   (d) $|\psi_{t+1}^{(\perp)}\rangle \leftarrow$ measure $H_{t+1}$ on $|\psi_t\rangle$
   (e) While measured energy nonzero:
       i.  $|\psi_t^{(\perp)}\rangle \leftarrow$ measure $H_t$ on $|\psi_{t+1}^{(\perp)}\rangle$
       ii. $|\psi_{t+1}^{(\perp)}\rangle \leftarrow$ measure $H_{t+1}$ on $|\psi_t^{(\perp)}\rangle$
   (f) $t \leftarrow t + 1$
```

Figure 1.1: Algorithm constructing injective PEPS

We now describe the main iteration of the algorithm (step 2), which is illustrated in figure 1.2. In steps 4a-4c, after having selected the next vertex v of the interaction graph according to the total vertex order, we construct a new Hamiltonian H_{t+1} from H_t: First, we select a d-dimensional "physical" subspace from the D^k dimensional space at each vertex v. This subspace is represented by projector $P_{phy}^{(v)}$. Then we remove for each neighboring vertex v' of v the term $H_t^{(v,v')}$. These are either trivial H_e terms or temporary boundary terms (see below). Next we compute the new parent Hamiltonian terms $H_{t+1}^{(v,v')}$ according to [Perez-Garcia et al., 2008] and add them to H_{t+1} reflecting the application of $A^{(v)}$. Restricted to the "physical" subspace $P_{phy}^{(v)}$, each $H^{(v,v')}$ is simply a sum of 2-local terms over all edges e from v to vertices v'. Note, that parent Hamiltonian terms $H^{(v,v')}$ towards any open "virtual index" v' are only temporary boundary terms which are computed in exactly the same way just as those for any other vertex by assuming the identity as the applicable PEPS map. Since the identity is trivially invertible, each intermediate PEPS is also injective and thus the unique ground state of the

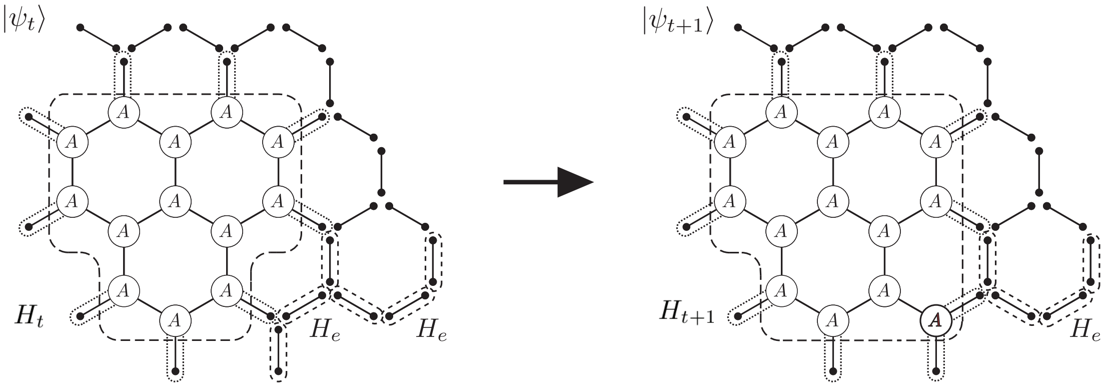

Figure 1.2: In each step, the algorithm processes one vertex v: H_t is grown into H_{t+1} by removing all existing terms referring to v, before the $2k$-local parent Hamiltonian terms are added implementing PEPS projector $A^{(v)}$ at v. Note, that terms around v connecting to an open "virtual" index v' (bonds with dotted border) are only temporary and are removed in later steps of the algorithm. All terms constraining a vertex v only restrict the physical subspace $P^{(v)}_{phy}$, while degrees of freedom in the orthogonal subspace $(\mathbb{1} - P^{(v)}_{phy})$ are eliminated with an additional penalty term $H^{(v)}_{phy}$ that is added per vertex.

intermediate Hamiltonian H_{t+1}. Since the "physical" d-dimensional space is just a subspace of the D^k "virtual" space *that is in fact also implemented physically in this algorithm*, $H^{(v)}$ is actually a sum of $2k$-local projectors. In order to ensure we produce a state with a single d-dimensional local space associated to each vertex v in the final PEPS, we add an extra term $H^{(v)}_{phy} = c(\mathbb{1} - P^{(v)}_{phy})$ in this step. This term penalizes the orthogonal complement of the chosen subspace with some energy $c \gg \Delta$.

Note, that prior to the execution of step 3, the system is in the ground state $|\psi_t\rangle$ of H_t by construction. This ground state is unique by the injectivity assumption we make for each intermediate PEPS $|\psi_t\rangle$ prepared in each iteration. In order to transition to the ground state $|\psi_{t+1}\rangle$ of H_{t+1}, we run the phase estimation [Nielsen and Chuang, 2000] algorithm for Hamiltonian H_{t+1}, perform a binary measurement to project $|\psi_t\rangle$ onto the zero/non-zero energy subspaces of H_{t+1}, and uncompute the phase estimation (step 3). This step requires an inverse eigenvalue gap Δ^{-1} between these two subspaces that scales with $O(poly(|V|))$ for the phase estimation to be efficient and precise enough [Berry et al., 2007]. We assume that such a gap exists for each intermediate parent Hamiltonian H_t that we construct according to the total vertex order defined on the interaction graph.

If the measurement results in the projection onto the zero-energy subspace of H_{t+1} we proceed to the next iteration (step 4). By Lemma 3, this event occurs with probability at least $\kappa(A^{(v)})^{-2}$, where $\kappa(A^{(v)})$ is the condition number of PEPS projector $A^{(v)}$ associated with vertex v. Note, that the injectivity property of the PEPS assures, that each $\kappa(A^{(v)})$ is a positive constant. If the measurement projects onto the excited subspace of H_{t+1}, we *undo* the measurement by measuring H_t again (step 5). If this second measurement results in a projection on the ground state, we have exactly undone the (unsuccessful) measurement of H_{t+1}, otherwise the system is in the excited subspace of H_t. In both cases the projection onto the ground state of H_{t+1} can now be attempted again, with success probabilities $\kappa(A^{(v)})^{-2}$ and $1 - \kappa(A^{(v)})^{-2}$, respectively (step 6). By Lemma 4, the inner loop will succeed in projecting onto the ground state of H_{t+1} with probability at least $1 - \frac{1}{2es}$ after at most $\kappa^2 s$ attempts, with s chosen as $s = \frac{|V|}{2e\varepsilon}$. Once all $|V|$ vertices have been covered, the outer loop terminates with the PEPS $|\psi\rangle$ in its output register with probability at least $1 - \varepsilon$, as shown in Theorem 2.

1.4 Analysis

1.4.1 Bounding the transition probabilities

As a first step in our analysis, we need a lower bound on the transition probability from $|\psi_t\rangle$ to $|\psi_{t+1}\rangle$. To this end we proof the following lemma.

Lemma 3. *Let $|\psi_t\rangle = \frac{1}{\sqrt{Z_t}} |A_t\rangle$ be the normalized PEPS $|A_t\rangle$, where $|A_t\rangle$ is the unnormalized partial PEPS resulting from the contraction of PEPS projectors $A^{(v)}$ for all vertices v processed in the algorithm up to step t and let $Z_t = \langle A_t|A_t\rangle$. Let $|A_{t+1}\rangle = A_{t+1}|A_t\rangle$ where A_{t+1} is the PEPS projector of time step $t+1$. Then $|\langle\psi_{t+1}|\psi_t\rangle|^2 \geq \frac{1}{\kappa(A_{t+1})^2} > 0$.*

Proof. A simple calculation shows

$$\begin{aligned}
\langle\psi_{t+1}|\psi_t\rangle &= \frac{1}{\sqrt{Z_t}}\frac{1}{\sqrt{Z_{t+1}}}\langle A_t| A_{t+1}^\dagger |A_t\rangle && (1.4)\\
&\geq \frac{1}{\sqrt{Z_t}}\frac{1}{\sqrt{Z_{t+1}}}\frac{\langle A_t| A_{t+1}^\dagger A_{t+1} |A_t\rangle}{\sigma_{\max}(A_{t+1})} && (1.5)\\
&= \frac{1}{\sqrt{Z_t}}\frac{1}{\sqrt{Z_{t+1}}}\frac{Z_{t+1}}{\sigma_{\max}(A_{t+1})} && (1.6)\\
&= \frac{1}{\sigma_{\max}(A_{t+1})}\left(\frac{Z_{t+1}}{Z_t}\right)^{\frac{1}{2}} && (1.7)
\end{aligned}$$

where the inequality follows from the operator inequalities $A_{t+1} \geq 0$ and $\frac{A_{t+1}}{\sigma_{\max}(A_{t+1})} \leq \mathbb{1}$. This implies

$$|\langle\psi_{t+1}|\psi_t\rangle|^2 \geq \frac{1}{\sigma_{\max}(A_{t+1})^2}\frac{Z_{t+1}}{Z_t} \tag{1.8}$$

But

$$\begin{aligned}
Z_{t+1} = \langle A_t| A_{t+1}^2 |A_t\rangle &\geq \sigma_{\min}(A_{t+1})^2\langle A_t|A_t\rangle && (1.9)\\
&= \sigma_{\min}(A_{t+1})^2 Z_t. && (1.10)
\end{aligned}$$

Thus Eq. 1.8 and Eq. 1.10 yield the claim

$$p = |\langle\psi_{t+1}|\psi_t\rangle|^2 \geq \left(\frac{\sigma_{\min}(A_{t+1})}{\sigma_{\max}(A_{t+1})}\right)^2 = \frac{1}{\kappa(A_{t+1})^2} \tag{1.11}$$

Finally, the injectivity assumption of PEPS $|\psi_{t+1}\rangle$ implies left invertibility of A_{t+1} for each v, thus $\kappa(A_{t+1})$ is finite, therefore $p > 0$. □

1.4.2 Bounding the convergence rate

In this section we analyze the termination probability of the loop at step 4.

Lemma 4. *Let H_t, H_{t+1} be Hamiltonians with unique zero-energy ground states $|\psi_t\rangle$ and $|\psi_{t+1}\rangle$, respectively. Let s be a positive integer. If the system is in state $|\psi_t\rangle$ initially, then the measurement process alternatingly measuring H_{t+1} and H_t and stopping once $|\psi_{t+1}\rangle$ is reached, takes the system to state $|\psi_{t+1}\rangle$ with probability at least $1 - \frac{1}{2es}$ after at most s/p alternations, where $p = |\langle\psi_{t+1}|\psi_t\rangle|^2$.*

Proof. Let P, Q be the ground state projectors of H_t and H_{t+1}, respectively, and let $P^\perp = \mathbb{1} - P$, $Q^\perp = \mathbb{1} - Q$. By Jordan's Lemma, there exists an orthonormal basis in which the Hilbert space decomposes into (1) two-dimensional subspaces S_i invariant under both, P and Q, and (2) one-dimensional subspaces T_j on which PQ is either an identity- or zero-projector [Nagaj et al., 2009].

Since we know that $|\psi_t\rangle$ and $|\psi_{t+1}\rangle$ are the unique 1-eigenstates of P and Q with overlap $\sqrt{p}$, exactly one S_i is relevant to our analysis. This two-dimensional subspace is spanned by both, $|\psi_t\rangle$ and some $|\psi_t^\perp\rangle$, as well as by $|\psi_{t+1}\rangle$ and some $|\psi_{t+1}^\perp\rangle$. Among these four vectors, we have got the following relationships [Marriott and Watrous, 2005]:

$$|\psi_t\rangle = -\sqrt{p}\,|\psi_{t+1}\rangle + \sqrt{1-p}\,|\psi_{t+1}^\perp\rangle \tag{1.12}$$

$$|\psi_t^\perp\rangle = \sqrt{1-p}\,|\psi_{t+1}\rangle + \sqrt{p}\,|\psi_{t+1}^\perp\rangle \tag{1.13}$$

$$|\psi_{t+1}\rangle = -\sqrt{p}\,|\psi_t\rangle + \sqrt{1-p}\,|\psi_t^\perp\rangle \tag{1.14}$$

$$|\psi_{t+1}^\perp\rangle = \sqrt{1-p}\,|\psi_t\rangle + \sqrt{p}\,|\psi_t^\perp\rangle \tag{1.15}$$

Considering these symmetrical relations, we see that alternating measurements of H_t and H_{t+1} generate a Markov process among these four states. Since the process terminates whenever it hits $|\psi_{t+1}\rangle$, the only histories which can keep the process from terminating are those with an initial transition $|\psi_t\rangle \to |\psi_{t+1}^\perp\rangle$ and which then keep repeating either one of the following two pairs of transitions

$$|\psi_{t+1}^\perp\rangle \to |\psi_t\rangle \to |\psi_{t+1}^\perp\rangle \tag{1.16}$$

$$|\psi_{t+1}^\perp\rangle \to |\psi_t^\perp\rangle \to |\psi_{t+1}^\perp\rangle, \tag{1.17}$$

which occur with probabilities $(1-p)^2$ and p^2, respectively. Thus the process terminates after at most $2m+1$ measurements with probability

$$p_{\text{term}}(p, m) = 1 - (1-p)(p^2 + (1-p)^2)^m. \tag{1.18}$$

To lower-bound this probability we upper-bound $p_{\text{fail}}(p, m) = 1 - p_{\text{term}}(p, m)$ as

$$p_{\text{fail}}(p, m) \le (1-p)exp(-2mp(1-p)) \tag{1.19}$$

which follows from $(1-q)^m \le e^{-qm}$, for $0 \le q \le 1$ and $m \ge 0$. Finally we choose m as a multiple of $\frac{1}{p}$ and find $p_{\text{fail}}(p, s/p) \le \frac{1}{2es}$, which can be seen by straightforward calculus. □

1.4.3 Proof of Theorem 2

Proof of Theorem 2. We complete the proof of Theorem 2 by using Lemma 4 for bounding the failure probability p_{fail} of the inner loop to derive a lower bound on the success probability of the outer loop over all vertices in V. That is, we have to show that

$$(1 - p_{\text{fail}})^{|V|} \geq 1 - \varepsilon. \tag{1.20}$$

Since

$$(1 - p_{\text{fail}})^{|V|} \geq 1 - |V| p_{\text{fail}} \tag{1.21}$$

by truncating higher-order terms from the binomial series and assuming $|V| > 1$ it suffices to show

$$|V| p_{\text{fail}} \leq \varepsilon. \tag{1.22}$$

Using Lemma 4, we find the first inequality of

$$|V| p_{\text{fail}} \leq \frac{|V|}{2es} \leq \varepsilon, \tag{1.23}$$

while the second inequality is satisfied by choosing

$$s \geq \frac{|V|}{2e\varepsilon}. \tag{1.24}$$

Thus, for the algorithm to succeed with at least probability $1 - \varepsilon$ we have to choose $m \geq \frac{s}{p} \geq \frac{|V|}{2pe\varepsilon}$. Since we know from Lemma 3 that $p \geq \frac{1}{\kappa^2}$, choosing

$$m \geq \frac{\kappa^2 |V|}{2e\varepsilon} \geq \frac{|V|}{2pe\varepsilon} \tag{1.25}$$

suffices.Thus the inner loop performs at most $2m + 1 \leq \frac{\kappa^2 |V|}{e\varepsilon}$ measurements. The outer loop iterates over $|V|$ vertices, thus the total number of measurements is less than $\frac{\kappa^2 |V|^2}{e\varepsilon} + |V|$. Bookkeeping of the active Hamiltonian terms in the outer loop requires a total time of $O(|V|k)$ using simple arrays as data structures, and $O(|V|kd^6)$ to compute all parent Hamiltonian terms, both of which are dominated by the $O(|V|^2)$ time of the inner loop for small d. Finally, since each phase estimation step requires $\tilde{O}(|E|^2/\Delta)$ [Berry et al., 2007, Nagaj et al., 2009, Nielsen and Chuang, 2000], where $\tilde{O}(\cdot)$ suppresses more slowly growing factors such as $exp(\sqrt{\ln(|E|/\Delta)})$ [Harrow et al., 2009], we find a total runtime of

$$\tilde{O}(\frac{|V|^2 |E|^2 \kappa^2}{\varepsilon \Delta} + |V|kd^6). \tag{1.26}$$

This completes the proof of Theorem 2. □

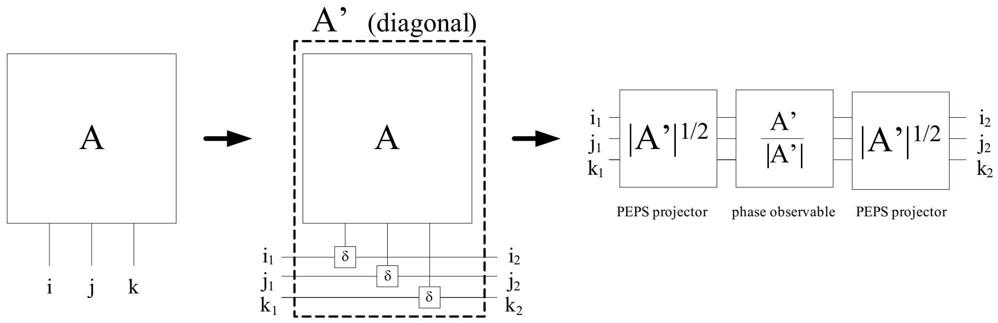

Figure 1.3: A generic tensor A can be split into a PEPS projector A' and an associated diagonal observable consisting exclusively of phases $e^{i\phi}$.

1.5 Contracting tensor networks on a quantum computer

Once an injective PEPS has been prepared, any local observable might be estimated. This is considered the primary application of our algorithm.

More generally, PEPS are a special class of open tensor networks. In [Arad and Landau, 2010] a quantum algorithm has been developed to compute an additive approximation of the contraction value of a (closed) tensor network under certain conditions. One might wonder, whether a general tensor network could be related to a PEPS and whether measuring an observable on such a PEPS could be used to approximate the contraction value of the tensor network as well. Indeed, under certain conditions such an approach seems feasible as we will sketch below.

Consider a generic tensor network T consisting of tensors $\{A\}$. The contraction value $\mathcal{C}(T)$ of tensor network T can now be related to the expected value of an observable on a PEPS by doubling each index of each A as shown in Figure 1.3 for three indices i, j, k, thus mapping $T = \{A\} \mapsto \{A'\} = T'$. Note, that each A' is diagonal by construction. We take the square-root of the absolute value of A' as our PEPS projector and $\hat{O} = \frac{A'}{|A|}$ yields a diagonal observable consisting entirely of phases $e^{i\phi}$. Let $|\psi\rangle$ be a state consisting of entangled pairs joining up the indices of tensor network T and let $\tilde{A}'$ be the tensor product of all A' of T', then an estimation of

$$\langle \hat{O} \rangle = \frac{\langle\psi|\,|\tilde{A}'|^{\frac{1}{2}}\,\hat{O}\,|\tilde{A}'|^{\frac{1}{2}}\,|\psi\rangle}{\langle\psi|\,|\tilde{A}'|\,|\psi\rangle}$$

approximates $\mathcal{C}(T) = \langle\hat{O}\rangle\,\langle\psi|\,|\tilde{A}'|\,|\psi\rangle$. The normalization factor $\langle\psi|\,|\tilde{A}'|\,|\psi\rangle$ is an unknown quantity, which can be approximated by Monte Carlo methods avoiding the sign problem since $|\tilde{A}'| > 0$. In this way, this approach splits the problem into a quantum part to be performed on a quantum computer, and a tractable classical part. The procedure yields an efficient algorithm to approximate $\mathcal{C}(T)$, if the ratio is bounded by a polynomial in the system size. While this will not be the case in general, the method may yield polynomial scaling in interesting instances, similar to the situation of Metropolis sampling. We leave a more detailed analysis and comparison to [Arad and Landau, 2010] to future work.

1.6 Conclusion

In this Letter we have shown how to construct quantum states described by injective PEPS in polynomial time by first reducing the problem to the generation of a sequence of unique ground states of certain Hamiltonians and then preparing that sequence. In future work we will focus on extending the class of preparable PEPS and possible performance improvements following from the results of [Boixo and Somma, 2010, Boixo et al., 2010, Somma and Boixo, 2013].

This work has been supported by Austrian SFB project FoQuS F4014.

Chapter 2

Preparing Topological Projected Entangled-Pair States on a Quantum Computer

Synopsis:

Simulating exotic phases of matter that are not amenable to classical techniques is one of the most important potential applications of quantum information processing. We present an efficient algorithm for preparing a large class of topological quantum states – the G-injective Projected Entangled Pair States (PEPS) – on a quantum computer. Important examples include the resonant valence bond (RVB) states, conjectured to be topological spin liquids. The runtime of the algorithm scales polynomially with the condition number of the PEPS projectors, and inverse-polynomially in the spectral gap of the PEPS parent Hamiltonian.

Based on:
M. Schwarz, K. Temme, F. Verstraete, D. Pérez-García, T. S. Cubitt,
Phys. Rev. A **88**, 032321 (2013)

Changes compared to published version: minor corrections.

2.1 Introduction

Creating and studying exotic phases of matter is one of the most challenging goals in contemporary physics. The increasingly sophisticated simulation abilities of systems such as cold atoms in optical lattices, trapped ions or superconducting qubits make this accessible by means of Feynman's original idea of using highly controllable quantum systems to simulate other quantum systems. Among those exotic phases, non-abelian topologically ordered states and topological spin liquids – such as resonating valence bond (RVB) states in frustrated lattices – are probably the holy grails of this area of quantum state engineering. Progress on the creation of such exotic phases in various experimental systems has accelerated rapidly in recent years, including cold atoms [Bloch et al., 2012], ion traps [Blatt and Roos, 2012], photonic devices [Aspuru-Guzik and Walther, 2012] and superconducting devices [Houck et al., 2012].

Recently [Schwarz et al., 2012], a very general way of constructing quantum states on a quantum computer was proposed. The wide applicability of the method lies in the fact that there is a variational class of quantum states, called Projective Entangled Pair States (PEPS) [Verstraete and Cirac, 2004], which has a simple local description but is nonetheless complex enough to approximate the low-energy sector of local Hamiltonians. (A review of the analytical and numerical evidence for this can be found in [Schuch et al., 2010] and the references therein.) However, a crucial technical assumption in the main result of [Schwarz et al., 2012], called 'injectivity', excludes any possibility of constructing quantum states with topological order.

The main aim of this article is to significantly extend these results to include exotic topological quantum phases, by proving:

Main result Any G-injective PEPS can be prepared on a quantum computer in polynomial time, when the spectral gap of the associated parent Hamiltonian scales at most inverse-polynomially in the system size.

'G-injectivity', introduced only recently in [Schuch et al., 2010] (and explained more fully below), is a substantially weaker requirement than injectivity, which explicitly allows for topological order. A compelling example of the significance of this result is the very recently proven fact that the resonating valence bond (RVB) state in the Kagome lattice (conjectured to be a topological spin liquid), is a $\mathbb{Z}_2$-injective PEPS, with numerical evidence that the gap assumption is also verified [Poilblanc et al., 2012]. Our result therefore gives one way in which the RVB state (and other topological states) can be prepared efficiently on a general quantum simulator. A large class of topological states is captured by quantum-double models [Kitaev, 2003]. These are equivalent to G-isometric PEPS [Schuch et al., 2011] and easy to prepare [Aguado and Vidal, 2008], which is related to the fact that the terms of the respective parent Hamiltonians always commute. G-injective PEPS generalize G-isometric PEPS and capture an even larger class of topological quantum states which are ground states of parent Hamiltonians with *non-commuting* terms for which no efficient preparation procedure has been known before. Engineering exotic quantum states by quantum simulation complements research aimed at finding materials that directly exhibit topological behavior, and is already beginning to bear fruit experimentally [Aspuru-Guzik and Walther, 2012, Blatt and Roos, 2012, Bloch et al., 2012, Houck et al., 2012].

In the following section, we summarize basic notions of PEPS required in this work, and introduce the class of G-injective PEPS which includes many of the important topological quantum states. We then briefly review the algorithm of Ref. [Schwarz et al., 2012] for preparing injective (non-topological) PEPS, before proceeding to show how this algorithm can be extended to the much larger class of G-injective PEPS, thereby allowing efficient preparation of many exotic topological quantum states. Finally, we close with some concluding remarks and open questions.

Projected Entangled Pair States For simplicity, we will focus on PEPS defined on a square lattice, but the results can be generalized to other lattices. An (unnormalized) PEPS can be described as follows. Place maximally entangled states of dimension D along all edges of the lattice. To each vertex ν, apply a linear map $A^\nu : (\mathbb{C}^D)^{\otimes 4} \to \mathbb{C}^d$ to the four D-dimensional systems labeled l, t, r, b (for 'left', 'top', 'right' and 'bottom'), where $A^\nu = \sum_{i;l,t,r,b} A^\nu_{i;ltrb} |i\rangle\langle ltrb|$. The resulting vector is the unnormalized PEPS. Since local unitaries do not change the complexity of preparing a state, for the purposes of this work we can assume without loss of generality that A is positive-semidefinite, by taking its polar decomposition. When A is left-invertible, we call the PEPS *injective* [Schuch et al., 2010].

A particularly interesting class of PEPS is the class of G-isometric PEPS, defined for any finite group G as follows. Take a semi-regular representation of G [Schuch et al., 2010] – that is, a representation $U_g = \oplus_\alpha V^\alpha_g \otimes \mathbb{1}_{r_\alpha}$ having at least one copy of each irrep α. Note that the regular representation is exactly the one for which r_α is the dimension d_α of the irrep V^α_g for all α. We can define the re-weighting map

$$\Delta = \oplus_\alpha \left(\frac{d_\alpha}{r_\alpha}\right)^{\frac{1}{4}} \mathbb{1}_{d_\alpha} \otimes \mathbb{1}_{r_\alpha} \tag{2.1}$$

which is real, diagonal, commutes with U_g and satisfies $\operatorname{Tr} \Delta^4 U_g = |G|\delta_{g,e}$. (For the regular representation $\Delta = \mathbb{1}$.) The PEPS is then defined by taking, for all ν:

$$A^\nu = \frac{1}{|G|} \sum_{g \in G} \Delta \bar{U}_g \otimes \Delta \bar{U}_g \otimes \Delta U_g \otimes \Delta U_g. \tag{2.2}$$

G-isometric PEPS were originally defined in [Schuch et al., 2010] only for the regular representation, and shown to be exactly the quantum-double models of Kitaev [Kitaev, 2003]. We use the argument described in [Schuch et al., 2010] for the Toric Code and RVB states, generalized here to arbitrary G-isometric PEPS, to see that the G-isometric PEPS for any semi-regular representation is equivalent to the one for the regular representation. Let us start with a semi-regular representation U_g of a group G, and let

$$B = \frac{1}{|G|} \sum_g \Delta\bar{U}_g \otimes \Delta\bar{U}_g \otimes \Delta U_g \otimes \Delta U_g \,. \tag{2.3}$$

We will show how B can indeed be seen as the G-isometric PEPS corresponding to the regular representation – possibly composed with an isometry which embeds the initial Hilbert space into a sufficiently large one. The latter can be prepared efficiently on a quantum computer by other means [Aguado and Vidal, 2008], which will be discussed in more detail in Sec. 3.2.

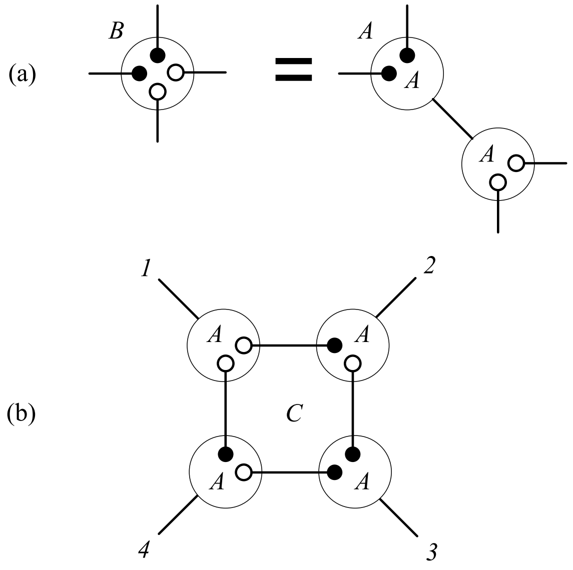

Figure 2.1: (a) illustrates the decomposition of the original tensor in the tensors A. We mark in white the bonds in which we have U_g and in black those in which we have $\bar{U}_g$. (b) illustrates the new way of grouping the tensors to get a G-isometric PEPS, called C. The bonds of this new tensor are numbered clockwise as in the figure.

To show this, we decompose the tensor B into two tensors of the form $A = (\sqrt{|G|})^{-1} \sum_g \Delta U_g \otimes \Delta U_g \otimes |g\rangle$ (where U_g and U_g are interchanged as needed, as shown in Fig. 2.1(a)). By regrouping these new tensors, we obtain a new PEPS decomposition of the same state, where now the bond dimension is $|G|$ (Fig. 2.1(b)). The resulting tensor C (Fig. 2.1(c)), as a map from the virtual to the physical indices, is given by

$$\begin{aligned} &C : |g_1 g_2 g_3 g_4\rangle \mapsto \\ &\quad \frac{1}{|G|^2} \Delta^2 U_{g_1 g_2^{-1}} \otimes \Delta^2 U_{g_2 g_3^{-1}} \otimes \Delta^2 U_{g_4 g_3^{-1}} \otimes \Delta^2 U_{g_1 g_4^{-1}} \,. \end{aligned} \tag{2.4}$$

By calling $g = g_1^{-1} g_1'$ and using Eq. 2.1 it is straightforward to see that

$$\langle g_1' g_2' g_3' g_4' | \, C^\dagger C \, | g_1 g_2 g_3 g_4 \rangle \\ = \frac{1}{|G|^4} \prod_{r=1}^{2} \mathrm{Tr}(\Delta^4 U_{g_r g_{r+1}^{-1} g_{r+1}' g_r'^{-1}}) \prod_{r=3}^{4} \mathrm{Tr}(\Delta^4 U_{g_{r+1} g_r^{-1} g_r' g_{r+1}'^{-1}}) \quad (2.5)$$

equals 1 if and only if there exist g such that $g_i g = g_i'$ for all i. Otherwise, the expression is identically zero.

Therefore $C^\dagger C = (|G|)^{-1} \sum_g R_g^{\otimes 4}$ for the regular representation R_g, hence the new PEPS C is the G-isometric PEPS corresponding to the regular representation.

If, on top of a G-isometric PEPS, we apply a further invertible (and w.l.o.g. positive-definite) linear map $A^\nu : \mathbb{C}^d \to \mathbb{C}^d$, we obtain a 'G-injective' PEPS [Schuch et al., 2010]. (Here, d is the dimension of the symmetric subspace associated with the group.) The parallel with plain injective PEPS is clear. Both are defined by invertible maps on top of a G-isometric PEPS. In the case of injective PEPS, the group is the trivial one and the representation is simply $\mathbb{1}_d$ (d copies of the left-regular representation of the trivial group).

G-isometric PEPS have very nice properties, coming from their topological character, which are inherited by the more general G-injective PEPS. For instance, for each G-isometric PEPS $|\psi\rangle$ there exists a local frustration-free Hamiltonian (called the PEPS "parent Hamiltonian" [Schuch et al., 2010]), consisting of commuting projectors and having as ground space the subspace (over-)spanned by $\{|\psi; K\rangle : K = (g, h), [g, h] = 0\}$. (Here, $|\psi; K\rangle$ is the PEPS obtained by the same maps A, except that we first apply an additional $U_g^{\otimes V}$ to exactly one vertical strip V and $U_h^{\otimes H}$ to exactly one horizontal strip H in the initial collection of maximally entangled states [Schuch et al., 2010]). This generalizes to G-injective PEPS, except that the local Hamiltonian terms are no longer necessarily commuting projectors.

We will denote by $|A^1 \cdots A^t\rangle$ the G-injective PEPS defined by applying the map A^j to vertex j for $j = 1, \ldots, t$ (and identity to the rest of the vertices) on top of the G-isometric PEPS, and define the states $|A^1 \cdots A^t; K\rangle$ analogously to above, which again (over-)span the ground space of a frustration-free local parent Hamiltonian H_t.

2.2 Algorithm

Preparing injective PEPS We first briefly review the algorithm of [Schwarz et al., 2012] for preparing injective PEPS on a quantum computer. Let H_t be the parent Hamiltonian of the partially constructed state $|A^1 \cdots A^t\rangle$. The algorithm starts at $t = 0$ with maximally entangled states between all pairs of adjacent sites in the lattice, and proceeds by successively projecting onto the ground states of H_t for $t = 1 \ldots N$ until the final state $|A^1 \cdots A^N\rangle$ is reached.

Since the ground state P_t of H_t is a complex, many-body quantum state, it is not immediately clear (i) how to efficiently perform the projective measurement $\{P_t, P_t^\perp\}$ onto the ground state. Furthermore, measurement in quantum mechanics is probabilistic, so even if this measurement can be performed, it is not at all clear (ii) how to guarantee the desired outcome P_t.

The answer to (i) is to run the coherent quantum phase estimation algorithm [Knill et al., 2007, Nielsen and Chuang, 2000] for the unitary generated by time-evolution under H_t. (Time-evolution under the local Hamiltonian H_t can be simulated efficiently by standard Hamiltonian simulation techniques [Berry et al., 2007].) If $\sum_k \alpha_k |\psi_k\rangle$ is the initial state expanded in the eigenbasis of H_t, then the phase estimation entangles this register with an output register containing an estimate of the corresponding eigenvalue: $\sum_k \alpha_k |\psi_k\rangle |E_k\rangle$. Performing a partial measurement on the output register to determine if its value is less than Δ_t (the spectral gap of H_t) completes the implementation of the measurement $\{P_t, P_t^\perp\}$. (See [Schwarz et al., 2012] for full details.)

The solution to (ii) is more subtle, and makes use of Camille Jordan's lemma of 1875 on the simultaneous block diagonalization of two projectors, which we first recall:

Lemma 5 (Jordan [Jordan, 1875]). *Let R and Q be projectors with rank s_r and s_q respectively. Then both projectors can be decomposed simultaneously in the form*

$$R = \bigoplus_{k=1}^{s_r} R_k \qquad \text{and} \qquad Q = \bigoplus_{k=1}^{s_q} Q_k, \tag{2.6}$$

where R_k, Q_k denote rank-1 projectors acting on one- or two-dimensional subspaces. The eigenvectors $|r_k\rangle, |r_k^\perp\rangle$ and $|q_k\rangle, |q_k^\perp\rangle$ of the 2×2 projectors R_k and Q_k are related by

$$\begin{aligned}
|r_k\rangle &= \sqrt{d_k}\,|q_k\rangle + \sqrt{1-d_k}\,|q_k^\perp\rangle \\
|r_k^\perp\rangle &= -\sqrt{1-d_k}\,|q_k\rangle + \sqrt{d_k}\,|q_k^\perp\rangle \\
|q_k\rangle &= \sqrt{d_k}\,|r_k\rangle - \sqrt{1-d_k}\,|r_k^\perp\rangle \\
|q_k^\perp\rangle &= \sqrt{1-d_k}\,|r_k\rangle + \sqrt{d_k}\,|r_k^\perp\rangle .
\end{aligned}$$

Ref. [Schwarz et al., 2012] shows that if the current state is in the 2×2 block containing the ground state of H_t, then the PEPS structure guarantees the probability of a successful projection onto P_{t+1} is at least $\kappa(A^{t+1})^{-2}$, where $\kappa(A^{t+1})$ is the condition number of the matrix A^{t+1}. Assume for induction that we have already successfully prepared the (unique) ground state of H_t. We first attempt to project from this state onto the unique ground state of H_{t+1} by measuring $\{P_{t+1}, P_{t+1}^\perp\}$. If this fails, we attempt to project back to the state we started from by measuring $\{P_t, P_t^\perp\}$ (a technique introduced by Marriott and Watrous [Marriott and Watrous, 2005] in the context of QMA-amplification). If this "rewind" measurement succeeds, then we're back where we started and can try again. What if the "rewind" measurement fails? By Lemma 5, we must be in the excited state from the same 2×2 block, so we can still try to project "forwards" with the same lower bound on the success probability. Thus iterating forwards and backwards measurements until success generates a Markov chain with successful

Algorithm 1 Preparing a G-injective PEPS. H_t ($t = 1 \dots N$) is the parent Hamiltonian for the G-injective PEPS $|A^1 \dots A^t\rangle$, P_t the projector onto its ground state subspace. Note that by specifying A^v, we are implicitly selecting a particular semi-regular representation of G.

Input: G-injective A^v defined on an N-vertex lattice; $\epsilon > 0$.

Output: $|\psi\rangle \in \operatorname{span}|A^1, \dots, A^t; K\rangle$ with probability $\geq 1 - \epsilon$.

```
Prepare corresponding G-isometric PEPS
for $t = 1$ to $N$ do
    Measure $\{P_{t+1}, P_{t+1}^{\perp}\}$ on $|\psi\rangle$.
    while measurement outcome is $P_{t+1}^{\perp}$ do
        Measure $\{P_t, P_t^{\perp}\}$.
        Measure $\{P_{t+1}, P_{t+1}^{\perp}\}$.
    end while
end for
```

projection onto the ground state of H_{t+1} as the unique absorbing state. Moreover, the success probability in each step is bounded away from zero, so this process converges in polynomial time to the ground state of H_{t+1}.

Preparing G-injective PEPS Consider the algorithm of the preceding section from the perspective of G-injective PEPS. An injective PEPS can always be viewed as a G-injective PEPS for the representation $\mathbb{1}$ of the trivial group. The algorithm starts from the state consisting of maximally-entangled pairs between each site, and transforms this into the desired state by projecting onto the ground states of a sequence of injective parent Hamiltonians. But the initial state is none other than the G-*isometric* PEPS corresponding to the representation $\mathbb{1}$ of the trivial group. This hints at a generalization of the algorithm to G-injective PEPS for arbitrary groups G: start by preparing the corresponding G-*isometric* PEPS, and successively transform this into the desired G-injective PEPS by projecting onto the ground states of the sequence of G-injective parent Hamiltonians (see Fig. 1).

There are, however, two obstacles to implementing this approach. (i) The initial G-isometric PEPS can be a substantially more complicated many-body quantum state than the trivial product of maximally-entangled pairs we must prepare in the injective case. (ii) Since G-injective parent Hamiltonians are topological, they have degenerate ground state subspaces. But the Marriott-Watrous "rewinding trick" [Marriott and Watrous, 2005] relies on the measurement projectors being 1-dimensional; it breaks down in general for higher-dimensional projectors.

There is a direct solution to (i). Ref. [Schuch et al., 2010] proves that, for any group G, the parent Hamiltonian of the G-isometric PEPS for the regular representation corresponds precisely to a quantum-double model [Kitaev, 2003, Schuch et al., 2011]. But Ref. [Aguado and Vidal, 2008] shows that ground states of quantum-double models can be generated exactly by a polynomial-size quantum circuit. We can therefore use this circuit to efficiently prepare the G-isometric PEPS for the regular representation of G. Even though the argument above only applies to regular representations of G, we have shown in Sec. 2.1, that a G-isometric PEPS for any semi-regular representation is equivalent to the one for the regular representation (up to local isometries), by simply regrouping the tensors. Thus the argument generalizes straightforwardly.

The second obstacle (ii) is more delicate. As described above, the Marriott-Watrous "rewinding" used in the injective case [Schwarz et al., 2012] works because the Hamiltonians H_t and H_{t+1} at each step have unique ground states, so that the back-and-forth measurement process is confined to a single 2×2 block. However, in the G-injective case, the Hamiltonians no longer have unique ground states, and there are multiple 2×2 blocks corresponding to different ground states. Thus when we "rewind" a failed measurement, the backwards measurement could project us back into any superposition of states from any of the blocks corresponding to the ground state subspace. Now, A^{t+1} is only invertible on the G-symmetric subspace, so it necessarily has some zero eigenvalues. Hence $\kappa(A^{t+1}) = \infty$ and the lower bound $\kappa(A^{t+1})^{-2} = 0$ on the probability of a successful forward measurement is useless. Although there may still exist some ground state $|\psi_t^1\rangle$ of H_t which has positive probability of successful forward transition to a ground state of H_{t+1}, this does not rule out existence of another ground state $|\psi_t^2\rangle$ of H_t for which the probability of a successful forward transition is 0. In the worst case, if a forward measurement fails and we end up in a state $|\varphi_{t+1}^{\perp}\rangle$, the rewinding step could have probability 1 of transitioning back to $|\psi_t^2\rangle$, so that we remain stuck forever bouncing back and forth between $|\psi_t^2\rangle$ and $|\varphi_{t+1}^{\perp}\rangle$.

To overcome this, we must show that *if we start from the G-isometric state*, then the structure of G-injective PEPS ensures that this situation can never occur. To prove this, we need the following technical lemma, which generalizes Lemma 2 of [Schwarz et al., 2012].

Lemma 6. *Let P_t and P_{t+1} denote two projectors on the ground state subspace of the partial PEPS parent Hamiltonians H_t and H_{t+1} for $|A^1 \dots A^t\rangle$ and $|A^1 \dots A^t, A^{t+1}\rangle$. The overlap d_k between P_t and P_{t+1} (cf. Lemma 5) is lower-bounded by $d_{\min} \geq \kappa(A^{t+1}|_{S_G})^{-2}$, where $\kappa(A^{t+1}|_{S_G}) := \sigma_{\max}(A_{t+1}|_{S_G})/\sigma_{\min}(A_{t+1}|_{S_G})$ is the condition number restricted to the G-symmetric subspace S_G.*

Proof. The minimum overlap $d_{\min}$ between projectors P_t and P_{t+1} is given by

$$d_{\min} = \min_{|\psi_t\rangle} \max_{|\psi_{t+1}\rangle} |\langle\psi_t|\psi_{t+1}\rangle|^2 , \tag{2.7}$$

where $|\psi_t\rangle$ and $|\psi_{t+1}\rangle$ are states in the respective ground state subspaces $\ker H_t$ and $\ker H_{t+1}$. Now, $\ker H_t$ is spanned by the partially constructed PEPS $|A^1, \dots, A^t; K\rangle$, with different boundary conditions K giving different ground states. Thus we can decompose any $|\psi_t\rangle \in \ker H_t$ as a linear combination of partial PEPS: $|\psi_t\rangle = \sum c_k |A^1, \dots, A^t; K^k\rangle$.

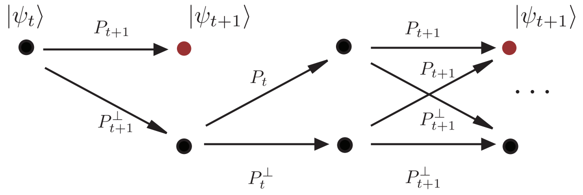

Figure 2.2: The sequence of outcomes of the binary measurements $\{P_t, P_t^{\perp}\}$ and $\{P_{t+1}, P_{t+1}^{\perp}\}$. We start in an eigenstate $|\psi_t\rangle$ of P_t, and want to transition to a state in the subspace P_{t+1}. With non-zero probability, the first measurement succeeds with outcome P_{t+1}. If it fails, we have prepared a state in the $P_{t+1}^{\perp}$ subspace. We "unwind" the measurement by measuring $\{P_t, P_t^{\perp}\}$ again. Upon repeating the $\{P_{t+1}, P_{t+1}^{\perp}\}$ measurement, we again have a non-zero probability of successfully obtaining the P_{t+1} outcome. This is repeated until success.

H_{t+1} is obtained from H_t by replacing all the G-isometric local Hamiltonian terms at one vertex with the G-injective terms. So applying A_{t+1} to *any* $|A^1, \dots, A^t; K\rangle$ takes us to the next ground state subspace. Therefore, the state

$$|\varphi_{t+1}\rangle = \frac{A_{t+1}|\psi_t\rangle}{\sqrt{\langle\psi_t| A_{t+1}^{\dagger} A_{t+1} |\psi_t\rangle}} \tag{2.8}$$

is contained in $\ker H_{t+1}$. Choosing $|\psi_{t+1}\rangle = |\varphi_{t+1}\rangle$ in Eq. (2.7), we obtain the lower bound

$$d_{\min} \geq \min_{|\psi_t\rangle} |\langle\psi_t|\varphi_{t+1}\rangle|^2 \geq \min_{|\psi_t\rangle} \frac{|\langle\psi_t| A_{t+1} |\psi_t\rangle|^2}{\langle\psi_t| A_{t+1}^{\dagger} A_{t+1} |\psi_t\rangle}. \tag{2.9}$$

It is immediate from the definition of G-injective PEPS that the ground states of H_t are symmetric, so that the projector P_t is supported on the symmetric subspace S_G. Thus the minimization is over symmetric states and, recalling that w.l.o.g. A_t is positive-semidefinite, we obtain the claimed bound

$$d_{\min} \geq \min_{|\psi_t\rangle} \frac{\langle\psi_t| A_{t+1}|_{S_G} |\psi_t\rangle^2}{\langle\psi_t| A_{t+1}^2|_{S_G} |\psi_t\rangle} \geq \frac{\sigma_{\min}\left(A_{t+1}|_{S_G}\right)^2}{\sigma_{\max}\left(A_{t+1}|_{S_G}\right)^2}, \tag{2.10}$$

by the variational characterization of eigenvalues. □

2.3 Analysis

Runtime We are now in a position to establish the runtime of the algorithm given in Fig. 1. We start by bounding the failure probability of growing the partial PEPS by a single site. The proof of the following lemma is closely analogous to Lemma 3 in [Schwarz et al., 2012] and reproduced below.

Lemma 7. *The measurement sequence depicted in Fig. 2.2 with the two projective measurements $\{P_t, P_t^{\perp}\}$ and $\{P_{t+1}, P_{t+1}^{\perp}\}$ has a failure probability bounded by*

$$p_{fail}(m) < \frac{1}{2\, d_{\min} m} \tag{2.11}$$

after m-subsequent measurement steps, where $d_{\min} = \min_k d_k$ is the minimal overlap between the eigenstates of P_t and P_{t+1}.

Proof. Let $Q_1 = P_{t+1}$, $Q_0 = P_{t+1}^{\perp}$ and $R_1 = P_t$, $R_0 = P_t^{\perp}$, in accordance with the notation in Lemma 5. Hence Q_1 projects on to the new ground state subspace, whereas R_1 is the projector on to the old ground state subspace. If we start in some state $|\psi\rangle = R_1 |\psi\rangle$, the probability of failure of the measurement sequence depicted in Fig. 2.2 after m steps is $p_{\text{fail}}(m) = \sum_{s_1,\dots,s_m} \operatorname{Tr}(Q_0 R_{s_m} Q_0 \dots R_{s_1} Q_0 |\psi\rangle\langle\psi| Q_0 R_{s_1} \dots Q_0 R_{s_m} Q_0)$. Note that $[Q_0 R_s Q_0, Q_0 R_p Q_0] = 0$ for all s, p. We can therefore rearrange this to express $p_{\text{fail}}(m)$ as the sum

$$\sum_{k=0}^{m} \binom{m}{k} \langle\psi| \left(Q_0 R_0 Q_0\right)^{2k} \left(Q_0 R_1 Q_0\right)^{2(m-k)} |\psi\rangle$$
$$= \langle\psi| \left(\left(Q_0 R_0 Q_0\right)^2 + \left(Q_0 R_1 Q_0\right)^2\right)^m |\psi\rangle .$$

If we work in the eigenbasis of Q_1, the individual 2 × 2 block matrices take the form

$$Q_1^k = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad R_1^k = \begin{pmatrix} d_k & \sqrt{d_k(1-d_k)} \\ \sqrt{d_k(1-d_k)} & 1-d_k \end{pmatrix}. \tag{2.12}$$

Since $|\psi\rangle$ is left invariant by R_1, we have that $|\psi\rangle = \sum_k c_k |r_k\rangle$, where in this basis every $|r_k\rangle = (\sqrt{d_k} \quad \sqrt{1-d_k})^T$ by Lemma 5. We are therefore left with

$$p_{\text{fail}}(m) = \sum_k |c_k|^2 \left(1-d_k\right)\left(1-2d_k(1-d_k)\right)^m , \tag{2.13}$$

with $d_k \in [0,1]$ and $\sum_k |c_k|^2 = 1$.

Since $(1-x) \le e^{-x}$, we may bound $(1-d_k)(1-2d_k(1-d_k))^m \le (1-d_k)e^{-2md_k(1-d_k)}$. Furthermore, we have that $(1-d_k)e^{-2md_k(1-d_k)} \le 1/2md_k$ by Taylor expansion. If we now choose the largest factor $(2md_k)^{-1} \le (2md_{\min})^{-1}$, we can bound the total failure probability by Eq. (2.11). □

We can use this to bound the overall runtime.

Theorem 8 (Runtime)**.** *Let A^v be G-symmetric tensors defining a PEPS on an N-vertex lattice. A state in the subspace spanned by the corresponding G-injective PEPS $|A^1 \dots A^N; K\rangle$ can be prepared on a quantum computer with probability $1-\epsilon$ in time $O(N^4 \kappa_G^2 \Delta^{-1} \epsilon^{-1})$, with additional classical processing $O(Nd^6)$. $\Delta = \min_t(\Delta_t)$ is the minimal spectral gap of the family of parent Hamiltonians H_t for $|A^1 \dots A^t\rangle$ ($t = 1 \dots N$), and $\kappa_G = \max_t \kappa(A^t|_{S_G})$.*

Proof. The algorithm in Fig. 1 first prepares the initial G-isometric PEPS, which can be done exactly in time $O(N \log N)$ [Aguado and Vidal, 2008], and then transforms this step by step into the G-injective PEPS, with one step for each of the N vertices of $\mathcal{G}$. Each step has a probability of failure $p_{\text{fail}}(m)$ if we repeat the back-and-forth measurement scheme m times. We need to ensure that the total success probability is lower-bounded by $(1-p_{\text{fail}}(m))^N \ge 1-\epsilon$.

Since $(1-x)^N \geq 1 - Nx$, we can use Lemma 7 to bound $(1 - p_{\text{fail}}(m))^N \geq 1 - \frac{N}{2md_{\min}}$, so we want $N/2md_{\min} \leq \epsilon$. Since $d_{\min} \geq \kappa_G^{-2}$ by Lemma 6, we choose $m \geq N\kappa_G^2/2\epsilon$ at each step. We therefore need to perform $O(N^2\kappa_G^2\epsilon^{-1})$ quantum phase estimation procedures, each of which has runtime $\tilde{O}(N^2/\Delta^{-1})$ to ensure that we are able to resolve the energy gap of the parent Hamiltonian [Harrow et al., 2009]. (Note that the notation $\tilde{O}(\cdot)$ suppresses more slowly growing terms such as $\exp(\sqrt{\ln N/\Delta})$.) The classical bookkeeping required to keep track of the Hamiltonians is the same as in [Schwarz et al., 2012]. Putting all this together, we arrive at the total runtime stated in the theorem. □

2.4 Conclusion

We have shown how the Marriott-Watrous rewinding technique combined with the unique structure of G-injective PEPS can be used to transition from one state to the next (see Fig. 1), successively building up the desired quantum state, even when the state has topological order and the ground states are degenerate. A number of alternative techniques could potentially be substituted for the measurement rewinding approach used here. In each case, the key to proving an efficient runtime is our Lemma 6. In many cases the existing results in the literature assume non-degenerate ground states, so would need to be generalized before they would apply to the topologically degenerate ground states considered here.

Standard adiabatic state preparation requires a polynomial energy gap along a continuous path joining the initial Hamiltonian with the final one. But the 'jagged adiabatic lemma' of Ref. [Aharonov and Ta-Shma, 2007] shows that such a path connecting a discrete set of gapped Hamiltonians always exists if the ground states are unique, and each has sufficient overlap with the next. This is precisely what we prove in Lemma 6. For the 'injective' case of [Schwarz et al., 2012], this is sufficient to show that adiabatic state preparation is an efficient alternative to the "rewinding" technique. Our results suggest it may be possible to generalize the jagged adiabatic lemma to degenerate ground states.

More general are the methods of [Boixo et al., 2010], which subsume the jagged adiabatic lemma and the Marriott-Watrous technique. The results in [Boixo et al., 2010] do not immediately apply to degenerate ground states, but if they can be generalized they could potentially improve the polynomial dependence on the required error probability to a logarithmic one. Similarly, the quantum rejection sampling technique of [Ozols et al., 2012] gives a quadratic improvement over Marriott-Watrous rewinding by a clever use of amplitude amplification. Finally, the spectral gap amplification technique of [Somma and Boixo, 2013], which cites injective PEPS preparation [Schwarz et al., 2012] as a potential application, may also be applicable. In all three cases, the techniques would first need to be generalized to handle degenerate ground states. Our Lemma 6 would then imply efficiency of the resulting algorithm.

The conditions required for efficient preparation in Theorem 8 (inverse-polynomial scaling of the spectral gaps of the partial parent Hamiltonians, and polynomial scaling of the condition numbers of the PEPS projectors) are reminiscent of the conditions (local gap and local topological quantum order) required for stability of the spectral gap of local Hamiltonians [Michalakis, 2012]. It is also conjectured that the spectral gap of the parent Hamiltonian should be closely related to the condition number of the PEPS projectors. It would be interesting to understand better the relationships between these various conditions.

The technique introduced in this letter, of constructing a complex many-body quantum state by starting from an easily-constructible state and successively transforming it into the desired state, is very general. Although we have applied it here to G-injective PEPS, as a class of states including many important topological quantum states, our algorithm can be generalized to other classes of tensor network states, such as string-net models [Schuch, 2012] and models constructed from Hopf algebras [Buerschaper et al., 2013].

Acknowledgements DPG and TSC thank the Centro de Ciencias Pedro Pascual and the Petronilla facility, Benasque, where part of this work was done. They were funded by Spanish grants QUITEMAD, I-MATH, and MTM2008-01366. TSC was supported by a Juan de la Cierva fellowship, MS by Austrian SFB grant FoQuS F4014, KT by an Erwin Schrödinger fellowship and Austrian FWF: J 3219-N16, FV by EU grants QUERG, and Austrian FWF SFB grants FoQuS and ViCoM.

Chapter 3

An Information-Theoretic Proof of the Constructive Commutative Quantum Lovász Local Lemma

Synopsis:

The Quantum Lovász Local Lemma (QLLL) [Ambainis et al., 2012] establishes non-constructively that any quantum system constrained by a local Hamiltonian has a zero-energy ground state, if the local Hamiltonian terms overlap only in a certain restricted way. In this paper, we present an efficient quantum algorithm to prepare this ground state for the special case of commuting projector terms. The related classical problem has been open for more than 34 years. Our algorithm follows the breakthrough ideas of Moser's [Moser, 2009] classical algorithm and lifts his information theoretic argument to the quantum setting. A similar result has been independently published by Arad and Sattath [Arad and Sattath, 2013] recently.

Based on:
M. Schwarz, T.S. Cubitt and F. Verstraete,
e-print arXiv:1311.6474, 2013, submitted

Changes compared to published version: minor corrections.

3.1 Introduction

In 1973 László Lovász proved a remarkable probabilistic lemma nowadays known as *Lovász Local Lemma (LLL)* [Erdős and Lovász, 1975, Spencer, 1977] . Informally, it says that whenever events in a set of probability events are only locally dependent (i.e. each event depends on at most a constant number of other events), then with positive probability *none* of them occurs. This probability might be extremely small, nevertheless the lemma shows that such an event *exists*. Lovász and Erdős applied this lemma with great success to prove the existence of various rare combinatorial objects, an approach which came to be known as the *probabilistic method* [Alon and Spencer, 2000]. Their method has one drawback: even though the LLL shows the existence of certain objects, it doesn't provide any clue of how to *construct* such objects efficiently – the lemma is non-constructive. Things started to change when in 1991 Beck was the first to give an efficient algorithm to construct such objects, but only under assumptions stronger than the LLL [Beck, 1991]. After a sequence of improvements on Beck's work, Moser's breakthrough in 2009 finally gave us a constructive and efficient proof of the LLL under the same assumptions as the original one [Moser, 2009]. First, he proved a widely used variant called the *symmetric LLL*, and then jointly with Tardos gave a fully general constructive and efficient proof of the LLL [Moser and Tardos, 2010]. The symmetric LLL considers the special case, where the probabilities of all of the dependent events are bounded by the same constant, and can be stated as follows:

Lemma 9 (Symmetric Lovász Local Lemma)**.** *Let $A_1, A_2, \ldots, A_m$ be a set of events such that each event occurs with probability at most p. If each event is independent of all others except for at most $d-1$ of them, and*

$$epd \le 1,$$

then

$$\Pr\left[\, \overline{A_1} \cap \overline{A_2} \cap \ldots \cap \overline{A_m} \,\right] > 0.$$

The symmetric LLL is often used in the context of constraint satisfaction problems (CSPs) to prove the existence of an object specified by a list of local constraints. In this case one considers, say, n bit strings X chosen uniformly at random. The events are given by the local constraint functions $A_i = f_i(X)$, where each function f_i is k-local in the sense that it depends only on k of the n bits; the event occurs if the constraint is satisfied. If these A_i meet the constraints of the symmetric LLL, the LLL implies that an x satisfying all the constraints *exists*, and Moser's algorithm can be used to construct such an x efficiently. In this way the LLL also implies that the set of k-SAT instances, where each variable occurs in at most $d < 2^k/ek$ clauses, is *always* satisfiable. (Without this restriction on variable occurrence, deciding satisfiability is of course the archetypical NP-complete problem.)

During the STOC 2009 presentation of his result, Moser presented a beautiful information-theoretic argument, valid under very slightly stronger conditions, which underlies the more complicated but tight result in [Moser, 2012]. It is this argument that the present paper generalizes to the quantum setting.

The non-constructive proof of the LLL has recently been generalized to the quantum case by Ambainis, Kempe, and Sattath [Ambainis et al., 2012]. In this setting, events are replaced by orthogonal projectors of rank 1 (or rank r in general) onto k-local subsystems, and the authors achieve a non-constructive proof of a Quantum Lovász Local Lemma (QLLL) with exactly the same constants as in the classical version.

Lemma 10 (Symmetric Quantum Lovász Local Lemma [Ambainis et al., 2012])**.** *Let* $\{\Pi_1, ..., \Pi_n\}$ *be a set of* k*-local projectors of rank at most* r*. If every qubit appears in at most* $d < 2^k/(e{\cdot}r{\cdot}k)$ *projectors, then the instance is satisfiable.*

In this paper, we generalize Moser's algorithm to the quantum setting in the special case of commuting projectors, yielding an efficient proof of Lemma 10 for this case. While all of our projectors are diagonal in a common basis, the basis vectors will in general be highly entangled quantum states. The (classical) constructive LLL does not immediately apply in the diagonal basis. Indeed, the preparation of such highly entangled ground states is far from trivial and subject to active research in the field of quantum Hamiltonian complexity theory [Aharonov and Eldar, 2011, 2013, Bravyi and Vyalyi, 2005, Hastings, 2013, Osborne, 2012, Schuch, 2011, Schwarz et al., 2013].

Furthermore, we improve upon Moser's argument and make it tight up to the assumptions of the non-constructive symmetric (Q)LLL. Of course, this also implies a tight algorithmic result for the classical special case. Our argument relies on a simple universal method to compress a binary classical bit sequence, which yields the tight result. In the process of generalizing the result to the quantum setting, we explicitly bound the run-time and error probabilities using (a tight special case of) the *strong converse of the typical subspace theorem* [Winter, 1999] as an indispensible ingredient, which is a fundamental result of quantum information theory.

More precisely, we prove the following efficient symmetric Quantum Lovász Local Lemma for commuting projectors with the same parameters as the original LLL and QLLL. Our proof is a quantum information-theoretic argument, but by restricting to classical constraints our argument immediately specializes to a tight classical information-theoretic proof.

Theorem 11 (Efficient symmetric commutative QLLL)**.** *Let* $\Pi_1, \Pi_2, ..., \Pi_m$ *be a set of commuting* k*-local projectors of rank at most* r *acting on a system of* n *qubits. If each projector intersects with at most* $d-1$ *of the others, where* $d \leq \frac{2^k}{re}$*, then for any* $\varepsilon > 0$ *there exists a quantum algorithm with run-time* $O\big(m + \log(\frac{1}{\varepsilon})\big)$ *that returns a quantum state* σ *with probability* $1-\varepsilon$*, such that* σ *has energy zero, i.e.* $\forall i, 1 \leq i \leq m : tr(\Pi_i \sigma) = 0$.

It might be interesting to note that for non-commuting projectors our proof still implies that the algorithm terminates within the same run-time bound, but the argument about the energy of the state returned (Lemma 16) is no longer applicable. Lemma 12 (see also Lemma 16) is the crucial and only place in the proof where commutativity of the projectors is used.

In Section 3.2, we fix the notation and review Moser's classical algorithm. In Section 3.2.2 we present the key ideas of our quantum generalization, and give a simple quantum information-theoretic analysis in Section 3.3 which leads to the main result. (A manifestly unitary variant of the recursive algorithm, complete with technical details, is given in Section 3.6.1.) We conclude in Section 3.4.

3.2 The Algorithm

In this section we describe our quantum version of Moser's algorithm. Before we do so, we quickly review Moser's classical original algorithm. We will start by setting up some notation, where we try to keep the notational differences between the quantum and classical case at a minimum.

The input to the classical (quantum) algorithm consists of a k-(Q)SAT instance. Each k-(Q)SAT instance is defined on n (qu)bits and consists of m clauses (projectors of rank at most r) $\{\Pi_i\}_{1\leq i\leq m}$. Each clause (projector) is k-local, i.e. it acts non-trivially only on a subset of k (qu)bits and as the identity on the $n-k$ remaining qubits. Given an instance $\{\Pi_i\}$, the *exclusive neighborhood function* $\Gamma(\Pi_i)$ returns an ordered tuple of projectors sharing at least one qubit with Π_i. Furthermore we define the inclusive neighborhood function $\Gamma^+(\Pi_i) = \Gamma(\Pi_i) \cup \Pi_i$. The j^{th} neighbor of Π_i is then defined as $\Gamma^+(\Pi_i)_j$. To simplify the notation, we sometimes write $\Gamma^+(i,j)$ instead of $\Gamma^+(\Pi_i)_j$. In the special case of a k-QSAT instance where all $\{\Pi_i\}$ are diagonal in the standard basis, it reduces to a classical k-SAT instance and projectors reduce to clauses. All logarithms in this paper use base 2.

3.2.1 Moser's classical algorithm

We will now quickly review Moser's classical algorithm to set the scene for our quantum generalization. In Algorithm 2 we assume a classical k-SAT instance as input. The algorithm operates on a register of n bits sampled from a uniformly random source. The main procedure *solve_lll()* iterates over the clauses $\Pi_1, \Pi_2, \ldots, \Pi_m$ and calls subroutine *fix(*Π_i*)* on each. Procedure *fix(*Π_i*)* checks if Π_i is satisfied, records the outcome to a logging register L ("the log") and returns if it is. Otherwise *fix()* resamples the bits of the unsatisfied clause from the uniformly random source and recurses on all neighbors in $\Gamma^+(\Pi_i)$ in turn. Throughout the paper *fixing a clause* or *fixing a projector* will mean entering such a recursion. Whenever we observe a clause not to be satisfied, we say the measurement of the clause has *failed* (or *succeeded* otherwise.) In the quantum case, whenever a projective measurement $\{\Pi_i, (\mathbb{1} - \Pi_i)\}$ has outcome Π_i we say it has failed (or *succeeded* if the outcome is $(\mathbb{1} - \Pi_i)$.)

Moser's key insight was to understand Algorithm 2 as a compression algorithm, that draws entropy from a uniformly random source and compresses it into the log register L. He shows that the random initial state of n bits and all entropy drawn from the source during execution of

Algorithm 2 Classical and quantum information-theoretic LLL solver

```
procedure solve_lll($\Pi_1, \Pi_2, \ldots, \Pi_m$)
    $W \leftarrow n$ uniformly random bits                    ▷ initial state
    $R \leftarrow kN$ uniformly random bits                   ▷ source of randomness
    $t \leftarrow 0$, $L \leftarrow 0...0$                    ▷ book keeping registers
    for $i \leftarrow 1$ to $m$ do
        fix($\Pi_i$)
    end for
    return (SUCCESS, W)
end procedure
procedure fix($\Pi_i$)
    measure $\Pi_i$ on $W$
    append the binary result to the execution log, $L$
    if $\Pi_i$ was violated then
        swap subsystem of $\Pi_i$ with block $t$ in $R$
        apply $U_i$ to rotate the state of the swapped subsystem in $R$
        $t \leftarrow t + 1$
        for all $\Pi_j \in \Gamma^+(\Pi_i)$ do
            fix($\Pi_j$)
        end
    end if
end procedure
```

the algorithm can be losslessly compressed into the log and the output state. By showing that each failed measurement yields a tighter bound on the entropy of the system, he argues that the algorithm must terminate with high probability after $O(m)$ measurements, as otherwise the entropy of the system was compressed below the entropy drawn from the source. Furthermore, each time *fix()* returns, one more projector is satisfied. Thus, once the algorithm terminates, all projectors are satisfied and the output state must therefore have energy zero.

In Moser's algorithm the log is introduced merely as a bookkeeping device to facilitate the correctness proof of the algorithm. It is not necessary to produce the log in "real world" implementations; the log is merely a proof device to allow one to argue about the entropy of the system by constructing a reversible compression scheme. Since a quantum algorithm in the standard quantum circuit model is unitary, thus in particular reversible, and the concept of reversible lossless compression is central to Moser's proof, this proof approach is a natural fit, and an ideal starting point to develop an efficient quantum algorithm for the QLLL based on a quantum information-theoretic argument. Once unitarity is *required*, the log is no longer an optional, fictitious device. Instead, it becomes a natural and *necessary* by-product of any unitary (or even reversible) implementation.

3.2.2 The quantum algorithm

Although we have to modify the analysis somewhat, our quantum algorithm is just a coherent version of the original classical algorithm of Algorithm 2. In this section, we show how a beautifully simple quantum information-theoretic analysis of this coherent algorithm gives the desired result. A fully detailed version of the proof based on a manifestly unitary version of Algorithm 2 (i.e. Algorithm 3) is given in Section 3.6.1.

Unsurprisingly, the quantum algorithm operates on four registers: an n-qubit work register W, an T-qubit log register L consisting of qubits labeled $j_1, ..., j_T$, a kN-qubit randomness register R, and a $\log N$-qubit register t counting the number of failed measurements.[1] Henceforth, j_l will denote the l^{th} qubit of L, and R_t will denote the t^{th} *block* of k qubits in R. We will use W_i to denote the k qubits in W on which the i^{th} projector acts non-trivially. We use Π_i to denote both the projector on W_i, and the projector $\Pi_i \otimes \mathbb{1}$ extended to the whole of W; when not indicated explicitly, it will be clear from context which we mean. We initialize the quantum registers to the state

$$|\psi_0^{x,y}\rangle = |x\rangle_W \, |y\rangle_R \, |0_1, ..., 0_T\rangle_L \, |0\rangle_t \tag{3.1}$$

where x, y are uniformly random bit strings of sizes n and kN, respectively. Algorithm 2 proceeds by *coherently measuring* projectors on the work register and appending the measurement outcomes to the log register. More precisely, a "coherent measurement" of Π_i is the following unitary operation between the work register and the next unused qubit in the log register.[2]:

$$C_i = \Pi_{W_i} \otimes X_{j_l} + (\mathbb{1} - \Pi_i)_{W_i} \otimes \mathbb{1}_{j_l}. \tag{3.2}$$

If $l - 1$ measurements have been performed so far, the next coherent measurement writes its outcome to the l^{th} qubit in the log register L.

As is well known [Nielsen and Chuang, 2000, Wilde, 2013], when applied to an arbitrary state of the work register W and a $|0\rangle_{j_l}$ in the log register j_l, the unitary C_i prepares a coherent superposition of the two measurement outcomes in the log register j_l, entangled with the corresponding post-measurement state in the work register. The square-amplitudes of the two components are the probabilities of the corresponding measurement outcomes.

If a projector Π_i is violated (outcome "1"), we know that the state of the subsystem W_i is contained in the subspace Π_i. In this case, we proceed by taking the next k qubits from the randomness register, and swapping them with the k work-qubits we've just measured. The state of the measured qubits must be in the r-dimensional subspace projected onto by Π_i. We can therefore apply a unitary U_i to rotate the measured qubits (which are now in the randomness register) into a fixed r-dimensional subspace which is independent of the particular Π_i

[1] The algorithm will also have to store some additional data for classical book-keeping, which however we neglect here as it isn't important in the analysis. Full details are given in Section 3.6.1.

[2] The algorithm necessarily keeps track of the index of the next unused log register qubit, as part of the classical bookkeeping implicit in Algorithm 2.

measured. We identify this subspace with the rank-r projector $P_r = diag(1,...,1,0,...,0)$. The unitary U_i can be computed classically for each i by diagonalizing Π_i, i.e. $U_i \Pi_i U_i^\dagger = P_i \leq P_r$ with equality if $rk(P_i) = rk(P_r) = r$. Let us denote this sequence of unitary swap-and-rotate operations as R_i. Note that the measured, swapped, and rotated k qubits $|\varphi_i\rangle$ have support on subspace P_r only, since

$$U_i \Pi_i |\varphi_i\rangle_R = U_i \Pi_i U_i^\dagger U_i |\varphi_i\rangle_R = P_r U_i |\varphi_i\rangle_R \tag{3.3}$$

The following partial isometry implements this swap-and-rotate procedure (it can be extended to a unitary in the usual way):

$$R_i = |1\rangle\langle 1|^{j_l} \otimes (\mathbb{1}^{W_i} \otimes U_i^{R_t} \cdot U_{\text{SWAP}}^{W_i R_t}) + |0\rangle\langle 0|^{j_l} \otimes \mathbb{1}^{W_i R_t}. \tag{3.4}$$

We will always apply R_i immediately after each coherent measurement, so for brevity we refer to the whole isometry $R_i C_i$ as a “measurement operation”. Whenever we get a violation, we increment the count register t.

The recursive algorithm now proceeds analogously to the classical algorithm Algorithm 2. The only differences are that we interpret Π_i as commuting projectors (not necessarily diagonal in the computational basis), and that ‘measure’ in Line 11 is interpreted as a coherent measurement causing the state (and thus the control flow) to split into a superposition depending on the measurement outcomes.[1]

Note that any computational basis state describing a sequence of measurement outcomes *uniquely determines* the next measurement to perform; i.e. there is a deterministic function $f : \{0,1\}^* \mapsto \{[m], \perp\}$ from finite sequences $j_1, \dots, j_{l-1}$ of previous measurement outcomes to the index i_l of the next measurement (i.e. $i_l = f(j_1, \dots, j_{l-1})$). If there is no further measurement to perform ($\perp$), the measurement sequence terminates (i.e. $f(\dots, \perp) = \perp$). It is not difficult to see that this function can be computed efficiently classically. By linearity, we can extend this to a unitary operation on arbitrary superpositions of a specific number of measurement outcomes.

Apart from the measurement operations, the rest of the algorithm involves purely classical processing to determine the next measurement, and thus *is diagonal in the computational basis*. Furthermore, each measurement operation acts on a fresh log qubit. Thus orthogonal states of the log remain orthogonal for the rest of the computation. This allows us to view the execution of the algorithm as a coherent superposition of *histories*, which may be analyzed independently. Lemma 14 in Section 3.6.1 makes this precise, and shows that after T coherent measurements, the state (essentially) has the form

$$|\psi_T^{x,y}\rangle = \sum_{j_1,\dots,j_T \in \{0,1\}} P_r^{\otimes t_{j_1,\dots,j_T}} |\varphi_{j_1,\dots,j_T}\rangle_{W,R} |j_1, \dots, j_T\rangle_L |t_{j_1,\dots,j_T}\rangle_t . \tag{3.5}$$

[1] For an explicit, manifestly unitary description that includes *all* the classical bookkeeping in the quantum description, see Algorithm 3 in Section 3.6.1.

Henceforth, we refer to any term in Eq. (3.5) indexed by $j_1, \dots, j_T$ as a *history*. Note the tensor product structure among the registers in each history.

We let the algorithm run for a total of $T = m+Nd$ measurement steps, for some N chosen in advance. If the recursion in Algorithm 2 has reached a maximum of N failed measurements or terminates early, the algorithm (coherently) does nothing for the remaining steps. Finally, after running for this many steps, we measure the log register L in order to collapse the superposition of measurement outcomes to a particular measurement sequence.

3.3 Analysis

To show that our algorithm efficiently finds a state in the kernel of all Π_i with high probability, we need to prove two properties captured in Lemma 12 and Lemma 13, that together imply the desired result:

1. If the sequence of measurement outcomes terminates, the corresponding state of the work register is in the kernel of all Π_i (Lemma 12).
2. The probability that the measurement sequence terminates goes exponentially to 1 for $N > m/(k - \log(der))$ (Lemma 13).

Lemma 12. *Let $|\varphi_l\rangle_W = |\varphi_{j_1,\dots,j_l}\rangle_W$ be the state of register W in a history where the algorithm has obtained a failure in the l^{th} measurement outcome, thereby starting a recursion. Assuming that the recursion eventually terminates, let $|\varphi_m\rangle = |\varphi_{j_1,\dots,j_m}\rangle$ be the state of register W when the algorithm has just returned from that recursion after measurement $m \geq l + k$. Then*

1. *all satisfied projectors Π_i stay satisfied, i.e. if $\Pi_i |\varphi_l\rangle = 0$, then also $\Pi_i |\varphi_m\rangle = 0$. ,*
2. *the originally unsatisfied projector Π_l is now satisfied, i.e. if $\Pi_l |\varphi_l\rangle = |\varphi_l\rangle$, then $\Pi_l |\varphi_m\rangle =$ 0.*

Proof. We prove Lemma 12 by induction on the recursion level s. Let Π_s be the projector that shall be fixed in the level s.

Base case: Consider the deepest level of recursion, which necessarily exists since, by assumption, the recursion eventually terminates. After the failed Π_l measurement, the algorithm performs the swap-and-rotate operation followed by measurements of all projectors in $\Gamma^+(\Pi_l)$ on the state $\Pi_l |\varphi_l\rangle$. These must succeed, since the algorithm is already at the deepest level of recursion. Thus the algorithm returns yielding the state $|\varphi_m\rangle$. Since $\Pi_l \in \Gamma^+(\Pi_l)$ and all Π_l commute, 2 follows. To show 1, note that all previously satisfied $\Pi_i \in \Gamma^+(\Pi_l)$ clearly stay satisfied, i.e. $\forall \Pi_i \in \Gamma^+(\Pi_l) : \Pi_i |\varphi_m\rangle = 0$. For all other $\Pi_i \notin \Gamma^+(\Pi_l)$, notice that Π_i commutes with the swap-and-rotate operation as they act on disjoint subsystems, yielding $\forall \Pi_i \notin \Gamma^+(\Pi_l) : \Pi_i |\varphi_m\rangle = 0$, which proves the base case.

Inductive step: As induction hypotheses, assume 1 and 2 are true for any originally unsatisfied projector Π_{s+1} after the algorithm returns from recursion level $s+1$. At level s of the recursion, after a failed measurement Π_l the algorithm performs the swap-and-rotate operation followed by measurements of all projectors in $\Gamma^+(\Pi_l)$ on the state $\Pi_l |\varphi_l\rangle$. For any failed measurement, the algorithm will recurse to level $s+1$ and return with 1 and 2 satisfied by the induction hypothesis. Thus, after returning from the recursion, one additional $\Pi_i \in \Gamma^+(\Pi_l)$ is satisfied. For any successful measurement, again one additional Π_i is satisfied due to commutativity of the Π_i. Thus, once the iteration over the neighborhood is complete, the algorithm returns the state $|\varphi_m\rangle$ with all $\Pi_i \in \Gamma^+(\Pi_l)$ satisfied. Since $\Pi_l \in \Gamma^+(\Pi_l)$, 2 follows. To see that 1 also holds, note that all previously satisfied $\Pi_i \in \Gamma^+(\Pi_l)$ stay satisfied, i.e. $\forall \Pi_i \in \Gamma^+(\Pi_l) : \Pi_i |\varphi_m\rangle = 0$. For all other $\Pi_i \notin \Gamma^+(\Pi_l)$, notice that Π_i commutes with the swap-and-rotate operation as they act on disjoint subsystems, yielding $\forall \Pi_i \notin \Gamma^+(\Pi_l) : \Pi_i |\varphi_m\rangle = 0$. This establishes the inductive step, and the lemma follows. □

Property 1 follows from Lemma 12 and the fact that the algorithm measures each projector Π_i once at the top level of the recursion. Property 2 is the content of the following lemma.

Lemma 13. *If we let the algorithm run for $T = m + Nd$ steps, the probability that the measurement sequence terminated within this number of steps is $\geq 1 - 2^{-N(k - \log der) + m + \log N}$.*

Proof. The proof rests on three simple facts: (i) The initial state is maximally-mixed on $n + kN$ qubits (tensor a pure state on the rest). (ii) The algorithm is unitary. (iii) If a total of M violations occurred, the information stored in the log register L can be compressed to $m + M\log(de)$ qubits.

Consider a computational basis state $|\sigma\rangle_L |M\rangle_t$ of the log and count registers, describing a particular (classical) history σ with a total of M violations. Since the count register is incremented each time the algorithm measures a violation, σ must contain exactly M 1s. By encoding σ as the *index* $\iota(\sigma; M)$ of σ in the lexicographically-ordered set of length-N bit strings that contain exactly M ones, we could losslessly and deterministically compress σ to $m + M\log(de)$ bits [Cover and Thomas, 2006, Ch. 13.2]. (Note that we do not need to actually perform this compression step as part of the algorithm; it is sufficient that it is possible.) By linearity, we can extend this lossless compression to a unitary operation on the log and count registers L and t:

$$U_C |\sigma\rangle_L |M\rangle_t = \underbrace{|\iota(\sigma; M)\rangle |0\rangle^{\otimes(T - m - M\log(de))}}_{L} \underbrace{|M\rangle}_{t}. \tag{3.6}$$

Furthermore, since M violations occurred, each of the M subsystems of k qubits in the register R only has support on the r-dimensional subspace P_r (in the respective subsystem) by Eq. (3.3). Given this, if we apply U_C to the state of the log and count registers L and t, the following projector projects onto measurement histories with $M = N$:

$$P_N = U_C^\dagger P U_C, \tag{3.7}$$

where

$$P = \underbrace{\mathbb{1}^{\otimes n}}_{W} \otimes \underbrace{P_r{}^{\otimes N}}_{R} \otimes \underbrace{\mathbb{1}^{\otimes m+N\log de} \otimes |0\rangle\langle 0|^{\otimes(T-m-N\log de)}}_{L} \otimes \underbrace{\mathbb{1}^{\otimes \log N}}_{t}. \tag{3.8}$$

Meanwhile, from Eq. (3.1), the initial state of the registers is

$$\rho_0 = \frac{1}{2^{n+kN}} \sum_{x,y} |\psi_0^{x,y}\rangle\langle\psi_0^{x,y}| = \frac{\mathbb{1}_W}{2^n} \otimes \frac{\mathbb{1}_R}{2^{kN}} \otimes |0\rangle\langle 0|_L \otimes |0\rangle\langle 0|_l \, |0\rangle\langle 0|_t\,. \tag{3.9}$$

Let U denote the overall unitary describing the algorithm. The probability of measuring P_N on the final state of the algorithm is then[1]

$$\begin{aligned} \mathrm{Tr}[P_N U \rho_0 U^\dagger] &= 2^{-n-kN}\,\mathrm{Tr}\left[P U_C U (\mathbb{1}_{WR} \otimes |0\rangle\langle 0|_{L,l,t}) U^\dagger U_C^\dagger\right] \\ &\leq 2^{-n-kN}\,\mathrm{Tr}\,P = 2^{-N(k-\log der)+m+\log N}. \end{aligned} \tag{3.10}$$

Now, any measurement sequence where *less* than N measurements failed must have terminated early, since the total number of measurement steps $T = m + Nd$ is clearly sufficient to return from any recursion with less than N failed measurements (cf. Algorithm 2). Thus the projector $\mathbb{1} - P_N$ projects onto histories in which the sequence of measurement outcomes terminated, and the lemma follows. □

Choosing $N = O\left(\frac{m+\log(\frac{1}{\varepsilon})}{k-\log(der)}\right)$ in Lemma 13 suffices to produce the desired output state in register W with success probability $1-\varepsilon$. Together with Lemma 12, this proves Properties 1 and 2, and hence Theorem 11.

3.4 Conclusions

We have presented a quantum generalization of Moser's algorithm and information-theoretic analysis to efficiently construct a zero-energy ground state of certain local Hamiltonians. The existence of such ground states has been established by the non-constructive Quantum Lovász Local Lemma [Ambainis et al., 2012]. Our algorithm requires the additional assumption that the Hamiltonian is a sum of commuting projectors. In fact, for this special case, our algorithm is a *constructive proof* of the Quantum Lovász Local Lemma, as our argument does not depend

[1] Note that this inequality is none other than a sharp version of the strong converse of the typical subspace theorem [Winter, 1999, Lemma I.9], for the simple case of the maximally mixed state.

on the non-constructive result of [Ambainis et al., 2012]. After completion of this work, we have learned about a similar result of Arad and Sattath [Arad and Sattath, 2013]. Their proof uses an entropy-counting argument, which is arguably even simpler, but yields only constant probability of success.

The obvious open question is whether Theorem 11 can be generalized to the non-commuting case. The crucial (and only) place in our proof where commutativity is used and where the argument fails is Lemma 12 (see also Lemma 16 in Section 3.6.1). If the quantum algorithm is executed with non-commuting projectors, the present proof still shows that the algorithm terminates, i.e. the final measurement will project with high probability onto a subspace of terminated histories after the stated number of iterations. But we are not able to show that the state returned by the algorithm has low energy. Without commutativity, because measurements disturb quantum states, subsystems already checked at higher levels of the recursion may be messed up when fixing lower levels.

A further open question is whether Moser and Tardos' combinatorial proof [Moser and Tardos, 2010] of the Lovász Local Lemma for the more general, asymmetric case can be generalized to the quantum setting. It is interesting to note, that the dissipative algorithm of [Verstraete et al., 2009] is precisely the quantum analogue of Moser and Tardos' algorithm for the general, asymmetric Lovász Local Lemma written in the language of CP-maps. Thus, [Verstraete et al., 2009] already gives a way to prepare the ground state implied by the non-constructive QLLL [Ambainis et al., 2012]. What is missing is an argument supporting a polynomial-time convergence rate of the given CP-map. A first attempt in this direction for the case of commuting projectors has been made by the first and second author in [Cubitt and Schwarz, 2011]. While the specific argument has an unresolved gap in the proof, the general framework based on dissipative CP-maps still appears as a promising approach and might lead to a complete proof in the future.

3.5 Acknowledgements

The authors wish to thank Julia Kempe, Or Sattath, and Robin Moser for valuable discussions.

MS is supported by Austrian SFB grant FoQuS F4014, TSC is supported by the Royal Society, and FV by EU grants QUERG, and Austrian FWF SFB grants FoQuS and ViCoM.

3.6 Appendix

3.6.1 Detailed algorithm and proof

We are now ready to introduce the more detailed quantum Algorithm 3. Algorithm 3 is a manifestly unitary version of Algorithm 2 expanding all quantum registers necessary for bookkeeping, unrolling the recursion into a unitary loop, and uncomputing auxiliary variables whenever necessary for the rigorous argument. Furtheremore, we explicitly bound the number of iterations necessary, such that with high probability all relevant histories in superposition have actually returned from the (unrolled) recursion and terminated individually.

As already mentioned, our goal is to construct a unitary version of Moser's algorithm. Since projective measurements are not unitary and can only be performed at the end of a standard quantum circuit, our approach is to replace them by *coherent measurements* [Wilde, 2013, Ch. 5.4]. A coherent measurement of a binary observable $\{\Pi_i^0, \Pi_i^1\}$, with $\Pi_i^0 + \Pi_i^1 = \mathbb{1}$, on a subsystem will correlate the state of a target qubit with the two possible measurement outcomes in a unitary way. This coherent measurement operation is performed by the following operator that is easily checked to be unitary:

$$C_i = \Pi_i^0 \otimes \mathbb{1} + \Pi_i^1 \otimes X \tag{3.11}$$

where X is the Pauli matrix σ_x.

Algorithm 3 operates on a quantum system consisting of register $W, R, L, F, term, S, s, l, t, live$ summarized in Table 3.1 at the end of the paper. We assume registers W, R are initialized in the completely mixed state. Register W is the *work register* in which our algorithm will prepare a state σ satisfying the symmetric QLLL conditions. Register R is the source of randomness that is fed into the work register by the algorithm appropriately. Register L is called the *log register* holding an array of qubits $|j_1, \ldots, j_T\rangle$ that store the binary coherent measurement outcomes for a chosen projector Π_i in each iteration of the algorithm. Register F is an array recording whether a recursion level has terminated. While the information in this register is strictly redundant (relative to L), we find it necessary to first compute and later uncompute the contents of this register to achieve an *efficient* unitary implementation of the algorithm that is provably correct up to the symmetric QLLL condition simultaneously. Register $term$ is an array of qubits used to signal the termination of a measurement history in the coherent superposition of histories. Once the qubit $term[l]$ is set to $|1\rangle$ in iteration l in a particular history, further iterations will just be idle in that history until the overall algorithm terminates. The stack S is an array of *pairs* of registers, $proj$ and nbr. At recursion level i, register $S[i].proj$ refers to the projector $\pi_{S[i].proj}$ being fixed in level i, where $S[i].nbr$ indicates the index (relative $S[i].nbr$) of the *neighboring* projector currently being verified $(0..k-1)$. To simplify the presentation of the algorithm, we treat the top-level of the recursion by pretending that some fiduciary clause had failed that intersected with all clauses. In this way we can deal with the top-level iteration just like with any other level. To this effect we initialize the content of register $S[0].proj = 0$ and define the special projector Π_0 to act non-trivially on *all* n qubits intersecting with *all* projectors $\{\Pi_i\}_{1 \le i \le m}$. This defines the top level of the recursion. Register s is the stack pointer referring to the current recursion level. Register l is the log pointer, indicating the current iteration of procedure *iteration()* and the target qubit $L[l]$ for the coherent measurement in that iteration. Register t is the randomness pointer. It counts the number of failed measurements in a particular measurement history and points to the next available block of k random qubits starting at $R[tk]$. Finally, register $live$ is a parameter to procedure *iteration()* controlling whether operations among the W subsystem and the rest of the system should be performed ($live = 1$) or skipped ($live = 0$). This is used to facilitate uncomputation of redundant information in the above registers.

We will now describe the operation of Algorithm 3 in detail. It consists of two procedures. The main procedure *QLLL_solver()* (Line 1), and procedure *iteration()* (Line 19), which is called from *QLLL_solver()*. *QLLL_solver()* starts by executing procedure *iteration()* T times in the forward and T times in the reverse direction, as indicated by the dagger symbol in Line 10. In the forward direction procedure *iteration()* (invoked with $live = 1$) applies a coherent measurements of one of the k-QSAT projectors to the assignment register W and stores the coherent measurement outcome at the current position l in the log register L. Based on the measurement outcome, the stack and other bookkeeping registers are updated coherently as well. During the uncomputation phase we invoke procedure *iteration()* with parameter $live$ set to 0 such that all bookkeeping registers are uncomputed, *except* the log L itself as the coherent “unmeasurements" are skipped. Indeed, the contents of the large F and $term$ registers has been completely uncomputed, as they can be reconstructed from L alone. Note, that after the completion of the reverse iterations (before executing Line 12), all registers are back to their initial states, except the W, L, and R registers. Once all redundancy in the bookkeeping registers has been removed by uncomputation, procedure *compress()* compresses the R, L, t registers as explained in more detail in the next section. The function will return with the quantum state of register t recomputed. Finally a projective measurement on the subspace of histories with $t < N$ failed measurements is performed, in which case the algorithm returns SUCCESS and the subsystem W of quantum state $(\mathbb{1} - P_N)\sigma(\mathbb{1} - P_N)$, or FAILURE otherwise.

We will now describe the procedure *iteration()*. Unless the algorithm has terminated (or the function is not called with $live = 1$) each iteration of the algorithm performs exactly one coherent measurement (Line 22) and all necessary update actions on the state variables to simulate the recursive procedure of Moser’s algorithm. Since the measurement is coherent, the *execution splits into a superposition* of two possible measurement outcomes whenever this line of the algorithm is executed, unless all projectors are classical. In the case that the measurement *fails* (and *iteration()* is called with $live = 1$) the procedure *swap_and_rotate()* (Line 26) is invoked, denoted R_i below.

We are free to restrict our analysis to one particular history $|j_1, \ldots, j_l\rangle$ since the quantum state is just a superposition of all possible such histories. To see this, we proof the following

Lemma 14. *For any initial state*

$$|\psi_0^{x,y}\rangle = |x\rangle^W |y\rangle^R |0_1, \ldots, 0_T\rangle^L |0\rangle^l |0\rangle^t |0\rangle^{F,S,s,live,term} \tag{3.12}$$

with randomly chosen bit strings x, y, the quantum state produced by Algorithm 3 after $T > 0$

iterations has the following structure:

$$|\psi_T^{x,y}\rangle = \sum_{t=0}^{T} P_r^{\otimes t} \sum_{\substack{j_1+\cdots+j_T=t \\ j_i\in\{0,1\}}} |\varphi_{j_1,\ldots,j_T}\rangle^{W,R} |j_1,\ldots,j_T\rangle^L |T\rangle^l |t\rangle |z_{j_1,\ldots,j_T}\rangle^{F,S,s,live,term} \tag{3.13}$$

$$= \sum_{j_1,\ldots,j_T\in\{0,1\}} P_r^{\otimes t_{j_1,\ldots,j_T}} |\varphi_{j_1,\ldots,j_T}\rangle^{W,R} |j_1,\ldots,j_T\rangle^L |T\rangle^l |t_{j_1,\ldots,j_T}\rangle |z_{j_1,\ldots,j_T}\rangle \tag{3.14}$$

where $t_{j_1,\ldots,j_T} = \sum_{i=1}^T j_i$*, and where* $P_r^{\otimes t}$ *acts only non-trivially on the first* kt *qubits of register* R*. That is, the state can be written as a (non-uniform) superposition of* 2^T *orthogonal states enumerating all* T*-bit computational basis states* $|j_1,\ldots,j_T\rangle$ *in the* L *register, each of which is entangled with some quantum state* $|\varphi_{j_1,\ldots,j_T}\rangle$ *in the* W *and* R *registers, and* computational basis states *in the* $t, F, S, s, live,$ *and* $term$ *registers. Furthermore, the* R*-register components of this state live in a subspace of rank at most* $rk(P_r^{\otimes t}) = r^t$*, where* $t = \sum_{i=1}^T j_i$*.*

Proof. The proof proceeds by induction over T. The initial state of the algorithm (i.e. $T = 0$ iterations) is

$$|\psi_0\rangle = |x\rangle^W |y\rangle^R |0_1,\ldots,0_T\rangle^L |0\rangle^l |0\rangle^t |0\rangle^{F,S,s,live,term} \tag{3.15}$$

We claim that after $1 \le l \le T$ iterations the state has the following slightly more general structure:

$$|\psi_l\rangle = \sum_{t=0}^{l} P_r^{\otimes t} \sum_{\substack{j_1+\cdots+j_l=t \\ j_i\in\{0,1\}}} |\varphi_{j_1,\ldots,j_T}\rangle^{W,R} |j_1,\ldots,j_l,0_{l+1},\ldots,0_T\rangle^L |l\rangle |t\rangle |z_{j_1,\ldots,j_l}\rangle^{F,S,s,live,term} \tag{3.16}$$

Clearly, for $l = T$ the lemma follows. To prove the base case $l = 1$, notice that after the first iteration the state evolves to

$$|\psi_1\rangle = R_0C_0|\psi_0\rangle = R_0C_0|x,y\rangle^{W,R}|0_1,\ldots,0_T\rangle^L|0\rangle^l|0\rangle^t|0\rangle^{F,S,s,live,term} \tag{3.17}$$

$$= R_0\Pi_0^0|x,y\rangle^{W,R}|0_1,0_2\ldots,0_T\rangle^L|1\rangle^l|0\rangle^t|z_0\rangle \tag{3.18}$$

$$+ R_0\Pi_0^1|x,y\rangle^{W,R}|1_1,0_2\ldots,0_T\rangle^L|1\rangle^l|1\rangle^t|z_1\rangle$$

$$= R_0|\varphi_0\rangle^{W,R}|0_1,0_2\ldots,0_T\rangle^L|1\rangle^l|0\rangle^t|z_0\rangle \tag{3.19}$$

$$+ R_0|\varphi_1'\rangle^{W,R}|1_1,0_2\ldots,0_T\rangle^L|1\rangle^l|1\rangle^t|z_1\rangle$$

$$= |\varphi_0\rangle^{W,R}|0_1,0_2\ldots,0_T\rangle^L|1\rangle^l|0\rangle^t|z_0\rangle \tag{3.20}$$

$$+ P_r|\varphi_1\rangle^{W,R}|1_1,0_2\ldots,0_T\rangle^L|1\rangle^l|1\rangle^t|z_1\rangle$$

$$= P_r^{\otimes 0}|\varphi_0\rangle^{W,R}|0_1,0_2,\ldots,0_T\rangle^L|1\rangle^l|0\rangle^t|z_0\rangle \tag{3.21}$$

$$+ P_r^{\otimes 1}|\varphi_1\rangle^{W,R}|1_1,0_2,\ldots,0_T\rangle^L|1\rangle^l|1\rangle^t|z_1\rangle$$

$$= \sum_{t=0}^{1} P_r^{\otimes t} \sum_{\substack{j_1=t \\ j_1\in\{0,1\}}} |\varphi_{j_1}\rangle^{W,R}|j_1,0_2\ldots,0_T\rangle^L|1\rangle^l|t\rangle|z_{j_1}\rangle. \tag{3.22}$$

where in Eq. (3.18) we expand the coherent measurement C_0 using Section 3.6.1. We denote classical bookkeeping states in registers $F, S, s, L, live, term$ collectively as $|z_i\rangle$ henceforth. Note that the projectors act on W while $\mathbb{1}$ and X act on qubit $l = 0$ in L, respectively. We see that the state splits into a superposition of two states, with orthogonal states in qubit $|j_1\rangle$. In Eq. (3.19) we label the projected states by $\Pi_0^0 |x, y\rangle = |\varphi_0\rangle$, and $\Pi_0^1 |x, y\rangle = |\varphi_1'\rangle$. In Eq. (3.20) we apply the *swap_and_rotate()* operation R_0, which acts at the identity on the first term. On the second term, the projected qubits are swapped from the W into the R register and then rotated into the P_r subspace, transforming the state into

$$|\varphi_1\rangle = U_0^{R_0} \cdot U_{\text{SWAP}}^{W_0 R_0} |\varphi_1'\rangle = U_0^{R_0} \cdot U_{\text{SWAP}}^{W_0 R_0} \Pi_0^1 |x, y\rangle = P_r^{W_0} U_0^{R_0} \cdot U_{\text{SWAP}}^{W_0 R_0} |x, y\rangle . \tag{3.23}$$

which follows from Eqs. (3.3) and (3.4). Furthermore, this also implies that $|\varphi_1\rangle = P_r^{R_0} |\varphi_1\rangle$, so we are justified in explicitly extracting the projector P_r in Eq. (3.20). In Eq. (3.21) we insert the fiducial projector $P_r^{\otimes 0} = \mathbb{1}$ in order to rewrite the equation into a sum of the desired structure in Eq. (3.22). Thus, the state $|\psi_1\rangle$ has the required structure with $l = 1$, which proves the base case.

In subsequent iterations, we denote the operations of Algorithm 3 by operators $C_{j_1,\ldots,j_l}$ (*coherent measurement*), and $R_{j_1,\ldots,j_l}$ (*swap_and_rotate*), respectively. These are controlled by the content of the L and l registers. All further bookkeeping operations are to be considered to be part of $R_{j_1,\ldots,j_l}$. We need to show that the state has the structure of Eq. (3.16) for all l. This is indeed the case, since

$$|\psi_{l+1}\rangle = R_{j_1,\ldots,j_l} C_{j_1,\ldots,j_l} |\psi_l\rangle \tag{3.24}$$

$$= R_{j_1,\ldots,j_l} C_{j_1,\ldots,j_l} \sum_{t=0}^{l} P_r^{\otimes t} \sum_{\substack{j_1+\cdots+j_l=t \\ j_i \in \{0,1\}}} |\varphi_{j_1,\ldots,j_T}\rangle^{W,R} |j_1, \ldots, j_l, 0_{l+1}, \ldots, 0_T\rangle^L |l\rangle |t\rangle |z_{j_1,\ldots,j_l}\rangle \tag{3.25}$$

$$= R_{j_1,\ldots,j_l} \sum_{t=0}^{l} P_r^{\otimes t} \sum_{\substack{j_1+\cdots+j_l=t \\ j_i \in \{0,1\}}} (\Pi^0_{j_1,\ldots,j_l} |\varphi_{j_1,\ldots,j_l}\rangle^{W,R} |j_1, \ldots, j_l, 0, 0_{l+2} \ldots, 0_T\rangle^L |l+1\rangle |t\rangle |z_{j_1,\ldots,j_l}\rangle$$
$$+ \Pi^1_{j_1,\ldots,j_l} |\varphi_{j_1,\ldots,j_l}\rangle^{W,R} |j_1, \ldots, j_l, 1, 0_{l+1} \ldots, 0_T\rangle^L |l+1\rangle |t+1\rangle |z_{j_1,\ldots,j_l}\rangle) \tag{3.26}$$

$$= R_{j_1,\ldots,j_l} \sum_{t=0}^{l} P_r^{\otimes t} \sum_{\substack{j_1+\cdots+j_l=t \\ j_i \in \{0,1\}}} (|\varphi_{j_1,\ldots,j_l,0}\rangle^{W,R} |j_1, \ldots, j_l, 0, 0_{l+2} \ldots, 0_T\rangle^L |l+1\rangle |t\rangle |z_{j_1,\ldots,j_l}\rangle$$
$$+ |\varphi'_{j_1,\ldots,j_l,1}\rangle^{W,R} |j_1, \ldots, j_l, 1, 0_{l+1} \ldots, 0_T\rangle^L |l+1\rangle |t+1\rangle |z_{j_1,\ldots,j_l}\rangle) \tag{3.27}$$

$$= \sum_{t=0}^{l} P_r^{\otimes t} \sum_{\substack{j_1+\cdots+j_l=t \\ j_i \in \{0,1\}}} (|\varphi_{j_1,\ldots,j_l,0}\rangle^{W,R} |j_1, \ldots, j_l, 0, 0_{l+2} \ldots, 0_T\rangle^L |l+1\rangle |t\rangle |z_{j_1,\ldots,j_l}\rangle$$
$$+ {P_r}^{R_t} |\varphi_{j_1,\ldots,j_l,1}\rangle^{W,R} |j_1, \ldots, j_l, 1, 0_{l+1} \ldots, 0_T\rangle^L |l+1\rangle |t+1\rangle |z_{j_1,\ldots,j_l}\rangle) \tag{3.28}$$

$$= \sum_{t=0}^{l+1} P_r^{\otimes t} \sum_{\substack{j_1+\dots+j_{l+1}=t \\ j_i\in\{0,1\}}} |\varphi_{j_1,\dots,j_l,j_{l+1}}\rangle^{W,R} |j_1,\dots,j_l,j_{l+1},0_{l+2}\dots,0_T\rangle^L |l+1\rangle|t\rangle|z_{j_1,\dots,j_l}\rangle \tag{3.29}$$

where, again, in Eq. (3.26) we expand the coherent measurement $C_{j_1,\dots,j_l}$ using Section 3.6.1, where the projectors act on W while $\mathbb{1}$ and X act on qubit l in L, respectively. We see that the state splits into a superposition of two states orthogonal in the state of this qubit. Register l is increased by one in both states. In Eq. (3.27) we label the projected states by $\Pi^0_{j_1,\dots,j_l}|\varphi_{j_1,\dots,j_l}\rangle = |\varphi_{j_1,\dots,j_l,0}\rangle$, and $\Pi^1_{j_1,\dots,j_l}|\varphi_{j_1,\dots,j_l}\rangle = |\varphi'_{j_1,\dots,j_l,1}\rangle$. In Eq. (3.28) we apply the *swap_and_rotate()* operation $R_{j_1,\dots,j_l}$, which acts at the identity on the first term. On the second term, the projected qubits are swapped from the W into the R register and then rotated into the P_r subspace, transforming the state into

$$|\varphi_{j_1,\dots,j_l,1}\rangle = U_i^{R_t}\cdot U_{\text{SWAP}}^{W_iR_t}|\varphi'_{j_1,\dots,j_l,1}\rangle = U_i^{R_t}\cdot U_{\text{SWAP}}^{W_iR_t}\Pi_0^1|\varphi_{j_1,\dots,j_l}\rangle = P_r^{W_0}U_i^{R_t}\cdot U_{\text{SWAP}}^{W_iR_t}|\varphi_{j_1,\dots,j_l}\rangle . \tag{3.30}$$

which follows from Eqs. (3.3) and (3.4). Furthermore, this also implies that $|\varphi_{j_1,\dots,j_l,1}\rangle = P_r^{R_t}|\varphi_{j_1,\dots,j_l,1}\rangle$, thus we are justified in explicitly extracting the projector P_r in Eq. (3.20). Finally, in Eq. (3.29) we rewrite the state by adding the binary index j_{l+1} in the inner sum. Furthermore, we sum t up to $l+1$ accommodating the additional measurement. Evidently, the state has now the form claimed for $|\psi_{l+1}\rangle$. By induction, the state has the required form of Eq. (3.13) for all $1 \le l \le T$, yielding the lemma. □

3.6.2 Proof of Theorem 11

Proof. By Lemma 14 we know that after T iterations of Algorithm 3 the state has the form

$$|\psi_T^{x,y}\rangle = \sum_{t=0}^{T} P_r^{\otimes t} \sum_{\substack{j_1+\dots+j_T=t \\ j_i\in\{0,1\}}} |\varphi_{j_1,\dots,j_T}\rangle^{W,R} |j_1,\dots,j_T\rangle^L |T\rangle^l |t\rangle |z_{j_1,\dots,j_T}\rangle^{F,S,s,live,term} \tag{3.31}$$

After uncomputing the redundant registers, this simplifies to

$$|\psi_U^{x,y}\rangle = \sum_{t=0}^{T} P_r^{\otimes t} \sum_{\substack{j_1+\dots+j_T=t \\ j_i\in\{0,1\}}} |\varphi_{j_1,\dots,j_T}\rangle^{W,R} |j_1,\dots,j_T\rangle^L |T\rangle^l |t\rangle |0\rangle^{F,S,l,s,live,term} \tag{3.32}$$

One way to view state $|\psi_U^{x,y}\rangle$ is as a superposition of all possible measurement histories $j_1,\dots,j_T$, which were the result if we had performed projective rather than coherent measurements. By the *principle of deferred measurement* [Nielsen and Chuang, 2000], we can still measure all qubits in L to project onto one of these histories. Consequently, we call each term in the sum of Eq. (3.31) a *history* and identify histories by the outcomes $|j_1,\dots,j_T\rangle$ in register L.

Let us make a few observations about each history $|j_1,\dots,j_T\rangle^L$. If $j_1,\dots,j_T$ contains t failed measurement outcomes, we know from Lemma 14 that Algorithm 3 has projected the first t blocks of k qubits in R into the subspace $P_r^{\otimes t}$. Line 30 enforces that $t \le N \le (T-m)/d$, i.e. a maximum number N of failed measurements, which we will choose later on. Thus by terminating execution once the maximum admissible number of N failed measurements has been reached, we accept that some histories in superposition in $|\psi_T^{x,y}\rangle$ may not have returned from the recursion. On the other hand, for all histories with $t < N$ it is clear that they must have returned to the top-level of the recursion and terminated at iteration $T = m + dt$, since to the m top-level measurements exactly d more measurements are added for each of the t failed outcomes. Therefore, within the T bits of L at most t bits are in state $|1\rangle$. Thus the Shannon entropy of bit string $j_1, \ldots, j_T$ relative to t is at most $\log\binom{m+dt}{t} \leq m+\log\binom{dt}{t} \leq m+\log\left(\frac{det}{t}\right)^t = m+t\log(de)$ bits. By encoding L by the *index* of $j_1, \ldots, j_T$ in the lexicographically ordered set of bit strings of length T with t ones we can achieve compression of L to the above bound, relative to t [Cover and Thomas, 2006, Ch. 13.2]. This classical compression is performed reversibly in Line 12 by procedure $compress(j_1, \ldots, j_T, t)$ for each history $|j_1, \ldots, j_T\rangle$, in superposition.[1] Let us denote the state after the compression as

$$\begin{aligned}|\psi_C^{x,y}\rangle &= U_{\text{compress}}\,|\psi_U^{x,y}\rangle = \\ &= \sum_{t=0}^{T} P_r^{\otimes t} \sum_{\substack{j_1+\cdots+j_T=t\\ j_i\in\{0,1\}}} |\varphi_{j_1,\ldots,j_T}\rangle^{W,R} \left(|L_{j_1,\ldots,j_T}\rangle\,|0\rangle^{\otimes(T-m-t\log(de))}\right)^L |t\rangle\,|0\rangle \end{aligned} \tag{3.33}$$

where $L_{j_1,\ldots,j_T} = compress(j_1, \ldots, j_T, t)$. We formalize our knowledge about $|\psi_C^{x,y}\rangle$ by constructing a projector P_M onto the subspace with $t \geq M$:

$$P_M = \tag{3.34}$$

$$\underbrace{\mathbb{1}^{\otimes n}}_{W} \otimes \underbrace{P_r^{\otimes M} \otimes \mathbb{1}^{\otimes(N-M)k}}_{R} \otimes \underbrace{\mathbb{1}^{\otimes m+M\log(de)} \otimes (|0\rangle\langle 0|)^{T-m-M\log(de)}}_{L} \otimes \underbrace{\left(\sum_{\tau=M}^{N} |\tau\rangle\langle\tau|\right)}_{t} \otimes \underbrace{|0\rangle\langle 0|}_{F,S,s,l,live}$$

We now show that for $M > \Omega\left(\frac{m+\log(N)}{k-\log(der)}\right)$ the probability of successfully projecting the state

$$\rho_C = \frac{1}{2^{n+Nk}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^{Nk}-1} |\psi_C^{x,y}\rangle\langle\psi_C^{x,y}| \tag{3.35}$$

i.e. $|\psi_C^{x,y}\rangle$ mixed over all x, y, onto P_M is very low. Clearly, mixing over x, y injects $n + Nk$ bits of initial entropy. Let $V = U_{\text{compress}} U_0^{\dagger T} U_1^T$, and since $|\psi_C^{x,y}\rangle = V\,|\psi_0^{x,y}\rangle$, we have

$$\rho_C = \frac{1}{2^{n+Nk}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^{Nk}-1} V\,|\psi_0^{x,y}\rangle\langle\psi_0^{x,y}|\,V^\dagger \tag{3.36}$$

$$= \frac{1}{2^{n+Nk}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^{Nk}-1} V(|x\rangle\langle x|^W\,|y\rangle\langle y|^R\,|0\rangle\langle 0|^{L,l,t,F,S,s,live,term})V^\dagger \tag{3.37}$$

$$= \frac{1}{2^{n+Nk}} V(\mathbb{1} \otimes |0\rangle\langle 0|)V^\dagger \tag{3.38}$$

[1] Note a minor technicality: at the instant *compress(L,t)* is invoked, the t register has actually been uncomputed (like all other auxiliary variables) and must be recomputed within the function by simply counting the $t \leq N$ ones in each $|j_1, \ldots, j_T\rangle$. It could also have been copied before uncomputation. The recomputed value of t remains in the register as the function returns as the compression $|j_1, \ldots, j_T\rangle$ is relative to $|t\rangle$.

We now apply the following simple special case of the *strong converse of the typical subspace theorem* [Winter, 1999] to get an upper bound for the overlap of ρ_C with P_M. Note, that the following bound for this special case is slightly stronger than the original bound of [Winter, 1999].

Lemma 15. *Let Q be a projector on any subspace of $(\mathbb{C}^2)^{\otimes(n+m)}$ of dimension at most 2^{nR}, where $R < 1$ is fixed and $\frac{\mathbb{1}}{2^n} \otimes (|0\rangle\langle 0|)^m$ a completely mixed state with pure ancillas. Then,*

$$\operatorname{Tr}\left(Q\left(\frac{\mathbb{1}}{2^n} \otimes (|0\rangle\langle 0|)^{\otimes m}\right)\right) \leq \operatorname{Tr}\left(Q\frac{\mathbb{1}}{2^n}\right) = 2^{-n}\operatorname{Tr}(Q) \leq 2^{-n+nR} \tag{3.39}$$

Proof. The proof is immediate in Lemma 15. □

Thus we achieve the bound

$$\begin{aligned}
\operatorname{Tr}(P_M\rho_C) &= 2^{-(n+Nk)}\operatorname{Tr}(P_M V(\mathbb{1}\otimes|0\rangle\langle 0|)V^\dagger) && (3.40)\\
&\leq 2^{-(n+Nk)}\operatorname{Tr}(P_M) && (3.41)\\
&\leq 2^{m+\log(N)-M(k-\log(r)-\log(de))} && (3.42)\\
&\leq 2^{m+\log(N)-M(k-\log(der))} && (3.43)
\end{aligned}$$

On the other hand when $N = M$ we conclude, that the projector $(\mathbb{1} - P_N)$ onto histories with $t < N$ has overlap exponentially close to 1 with ρ_S. In other words, Algorithm 3 returns SUCCESS in Line 14 with

$$Pr[SUCCESS, \sigma] \geq 1 - 2^{m+\log(N)-N(k-\log(der))} \tag{3.44}$$

It follows that choosing N such that

$$N \geq \frac{m + \log(\frac{1}{\varepsilon})}{k - \log(der)} + \frac{\log(N)}{k - \log(der)} \tag{3.45}$$

suffices to push the error below $1-\varepsilon$. But this bound for N is not yet explicit. To get an explicit bound we define $c = (k - \log(der))^{-1}$, and $d = \frac{m+\log(\frac{1}{\varepsilon})}{k-\log(der)}$, and set (Line 4)

$$N = d + 3c(\log(d) + 1) \tag{3.46}$$

or, equivalently but more verbosely,

$$N = \frac{m + \log(\frac{1}{\varepsilon})}{k - \log(der)} + \frac{3(\log(\frac{m+\log(\frac{1}{\varepsilon})}{k-\log(der)}) + 1)}{k - \log(der)} \tag{3.47}$$

satisfying Eq. (3.43) as shown in Lemma 17 in the appendix. Thus we conclude that after $T = m + Nd$ (Line 5) iterations of Algorithm 3,

$$Pr[SUCCESS, \sigma] \geq 1 - \varepsilon \tag{3.48}$$

as claimed.

In summary, we have shown that either the algorithm achieves a compression of its state below the entropy of the initial state, which is unlikely, or in all histories in superposition the number of failed measurements is upper bounded by N and thus the histories must have terminated in the state returned by Algorithm 3. Furthermore, the probability of the latter outcome can be pushed exponentially close to 1. All that is left to show is that the state, once projected into the $(\mathbb{1} - P_N)$ subspace, satisfies the symmetric QLLL condition. By Lemma 16 shown below we know that each terminated history $j_1, \dots, j_T$ is correlated to a state $|\varphi_{j_1,\dots,j_T}\rangle$ with energy exactly zero. Thus it follows that the W subsystem of state $(\mathbb{1} - P_N)\rho_C(\mathbb{1} - P_N)$ returned by Algorithm 3 is just a mixture of zero energy states and has thus energy zero itself, which completes the proof, i.e. formally let

$$\rho_P = \frac{(\mathbb{1} - P_N)\rho_C(\mathbb{1} - P_N)}{1 - \mathrm{Tr}(P_N \rho_C)} \tag{3.49}$$

where the denominator is exponentially close to 1 due to Eq. (3.40). Then, expanding the definition of ρ_C and recognizing that the projector on $\mathbb{1} - P_N$ just changes the upper bound of the sum over t (and t') from T to N, we have

$$\begin{aligned}
\mathrm{Tr}_{\overline{W,R}}(\rho_P) &\propto \mathrm{Tr}_{\overline{W,R}}((\mathbb{1} - P_N)\rho_C(\mathbb{1} - P_N)) && (3.50)\\
&= \mathrm{Tr}_{\overline{W,R}}((\mathbb{1} - P_N)\,|\psi_S^{x,y}\rangle\langle\psi_S^{x,y}|\,(\mathbb{1} - P_N)) && (3.51)\\
&= \frac{1}{2^{n+Nk}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^{Nk}-1} \sum_{t=0}^{N} \sum_{\substack{j_1+\dots+j_N=t\\ j_i\in\{0,1\}}} \sum_{t'=0}^{N} \sum_{\substack{j'_1+\dots+j'_N=t'\\ j'_i\in\{0,1\}}} && (3.52)\\
&\quad \mathrm{Tr}_{\overline{W,R}}(|\varphi_{j_1,\dots,j_N}\rangle\langle\varphi_{j'_1,\dots,j'_N}|^{W,R}\,(|L'_{j_1,\dots,j_N}\rangle\langle L'_{j'_1,\dots,j'_N}|\,|0\rangle\langle 0|)^L |t\rangle\langle t'|\,|0\rangle\langle 0|) \\
&= \frac{1}{2^{n+Nk}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^{Nk}-1} \sum_{t=0}^{N} \sum_{\substack{j_1+\dots+j_N=t\\ j_i\in\{0,1\}}} |\varphi_{j_1,\dots,j_N}\rangle\langle\varphi_{j_1,\dots,j_N}|^{W,R} && (3.53)\\
&\quad \mathrm{Tr}(|L'_{j_1,\dots,j_N}\rangle\langle L'_{j_1,\dots,j_N}|)\,\mathrm{Tr}(|0\rangle\langle 0|)\,\mathrm{Tr}(|t\rangle\langle t|)\,\mathrm{Tr}(|0\rangle\langle 0|) \\
&= \frac{1}{2^{n+Nk}} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^{Nk}-1} \sum_{t=0}^{N} \sum_{\substack{j_1+\dots+j_N=t\\ j_i\in\{0,1\}}} |\varphi_{j_1,\dots,j_N}\rangle\langle\varphi_{j_1,\dots,j_N}|^{W,R} && (3.54)
\end{aligned}$$

where in Eq. (3.53) we distribute the partial trace over the tensor factors. Since orthogonal states evaluate to zero in each factor, only terms of factors with matching indices survive in the sum, in which case these factors happen to be projectors of trace 1. Thus Eq. (3.54) follows, which is clearly a mixture of states $|\varphi_{j_1,\dots,j_N}\rangle$ as claimed. Since every $|\varphi_{j_1,\dots,j_N}\rangle$ is a state associated to a terminated history, we know the recursion of Algorithm 3 has returned to the top level, in which all m initial projectors Π_i have been measured. Thus by Lemma 16 we conclude that $\Pi_i^1\,|\varphi_{j_1,\dots,j_N}\rangle = 0$ for all histories $j_1, \dots, j_N$. □

Note that the following lemma is the crucial (and only) place in the proof where commutativity of the projectors $\{\Pi_i\}$ is assumed.

Lemma 16. *According to Lemma 14, consider a history $|j_1, \dots, j_l\rangle$ in the superposition after l coherent measurements*

$$|\psi_l\rangle = |\varphi_{j_1,\dots,j_l}\rangle^{W,R} |j_1, \dots, j_l, 0_{l+1}, \dots, 0_T\rangle^L |l\rangle |t_{j_1,\dots,j_l}\rangle |z_{j_1,\dots,j_l}\rangle^{F,S,s,live,term} \tag{3.55}$$

where the last measurement has failed, i.e $j_l = 1$. In this state Algorithm 3 has started a new recursion level and will coherently measure all projectors $\Pi_k \subseteq \Gamma^+(\Pi^1_{j_1,\dots,j_l})$ in subsequent iterations. For some iteration $m \geq l + k$, let

$$|\psi_m\rangle = |\varphi_{j_1,\dots,j_m}\rangle^{W,R} |j_1, \dots, j_m, 0_{m+1}, \dots, 0_T\rangle^L |m\rangle |t_{j_1,\dots,j_m}\rangle |z_{j_1,\dots,j_m}\rangle^{F,S,s,live,term} \tag{3.56}$$

be an extension of history $|\psi_l\rangle$ (i.e. with matching $j_1, \dots, j_l$) where Algorithm 3 has just returned from that recursion. Then

1. *all satisfied projectors Π^1_i stay satisfied, i.e. if $\Pi^1_i |\varphi_{j_1,\dots,j_l}\rangle = 0$, then also $\Pi^1_i |\varphi_{j_1,\dots,j_m}\rangle = 0$.*
2. *the originally unsatisfied projector is now satisfied, i.e. $\Pi^1_{j_1,\dots,j_l} |\varphi_{j_1,\dots,j_m}\rangle = 0$.*

Proof. We first prove Item 1 by induction on the stack level s of Algorithm 3, starting from the deepest level, which must exist because the algorithm returns by assumption.[1] The recursive call can only return if all $\Pi^1_i \in \Gamma^+(\Pi(s))$ are satisfied, i.e. $\Pi^1_i |\varphi_{j_1,\dots,j_m}\rangle = 0$. For all $\Pi^1_q \notin \Gamma^+(\Pi(s))$ with $\Pi^1_q |\varphi_{j_1,\dots,j_l}\rangle = 0$, we have

$$\Pi^1_q |\varphi_{j_1,\dots,j_m}\rangle |\xi'\rangle = \Pi^1_q \prod_{i \in \Gamma^+} \Pi^0_i R_{j_1,\dots,j_l} \Pi^1_{j_1,\dots,j_l} |\varphi_{j_1,\dots,j_l}\rangle |\xi\rangle \tag{3.57}$$

$$= \prod_{i \in \Gamma^+} \Pi^0_i R_{j_1,\dots,j_l} \Pi^1_{j_1,\dots,j_l} \Pi^1_q |\varphi_{j_1,\dots,j_l}\rangle |\xi\rangle = 0 \tag{3.58}$$

where we expand $|\varphi_{j_1,\dots,j_m}\rangle$ by the action of Algorithm 3 in the first equality, where $|\xi'\rangle$ represents the state of subsystems other than W, R. In the second equality we commute Π^1_q through, which is possible, because Π^1_q and all Π_i commute by assumption, and Π^1_q and $R_{j_1,\dots,j_l} \Pi^1_{j_1,\dots,j_l}$ commute because they act on different subsystems. Finally, the last equation follows since $\Pi^1_q |\varphi_{j_1,\dots,j_l}\rangle = 0$ is the precondition under which we need to prove Item 1 of Lemma 16. This proves the base case of the induction. The inductive step follows from exactly the same arguments, thus Item 1 follows. To show Item 2 of Lemma 16, it suffices to note that $\Pi^1_{j_1,\dots,j_l} \in \Gamma^+(\Pi^1_{j_1,\dots,j_l})$, thus $\Pi^1_{j_1,\dots,j_l} |\varphi_{j_1,\dots,j_m}\rangle = 0$ is true since the algorithm just returned from a recursive call on a failed measurement of $\Pi^1_{j_1,\dots,j_l}$ by assumption: i.e. in the iterations $< m$ just before the algorithm has returned, all $\Pi^1_q \in \Gamma^+(\Pi^1_{j_1,\dots,j_l})$ had been measured to be satisfied (or fixed and then satisfied by Item 1). Since all Π^1_q commute, this implies $\Pi^1_{j_1,\dots,j_l} |\varphi_{j_1,\dots,j_m}\rangle = 0$. □

[1] In the main algorithm we apply this lemma only to histories in the subspace $(\mathbb{1} - P_N)$, where we have already shown that all histories terminate.

3.6.3 Upper bound on N

In this section we compute an upper bound for N defined implicitly by

$$N = \frac{\log(N)}{k - \log(der)} + \frac{m + \log(\frac{1}{\varepsilon})}{k - \log(der)} \tag{3.59}$$

Lemma 17. *Define* $a = (k - \log(der))^{-1}$, $b = \frac{m+\log(\frac{1}{\varepsilon})}{k-\log(der)}$, *and* $N = t + a\log(t)$, *then*

$$N \le b + a(\log(a+1) + \log(b + a\log(a+1))) \le b + 3a(\log(b) + 1) \tag{3.60}$$

Proof. We start with Section 3.6.3 as the implicit definition of N to derive the upper bound. Expanding the substitutions reduces Section 3.6.3 to

$$t + a\log(t) = a\log(t + a\log(t)) + b \tag{3.61}$$

Then we bound $\log(t) \le t$ coarsely on the r.h.s., which yields

$$t + a\log(t) \le a\log(t(a+1)) + b \tag{3.62}$$

$$t + a\log(t) \le a\log(a+1) + a\log(t) + b \tag{3.63}$$

$$t \le a\log(a+1) + b \tag{3.64}$$

Thus

$$N \le b + a(\log(a+1) + \log(b + a\log(a+1))) \tag{3.65}$$

which can be evaluated explicitly. Relaxing the bound further yields

$$N \le b + 3a(\log(b) + 1) \tag{3.66}$$

□

As asymptotic bounds we also have $N \le \frac{m+\log(\frac{1}{\varepsilon})}{k-\log(de)} + O(\log(m + \log(\frac{1}{\varepsilon})))$ or $N \le O(m + \log(\frac{1}{\varepsilon}))$.

Algorithm 3 Quantum information-theoretic QLLL solver

```
procedure QLLL_solver
    $a := 1/\log(k - de)$
    $b := (m + \log(1/\varepsilon))/\log(k - de)$
    $N := b + 3a(\log(b) + 1)$
    $T := m + Nd$
    for $l := 0$ to $T - 1$ do
        iteration($live = 1$)
    end for
    for $l := T - 1$ to 0 do
        iteration$^\dagger$($live = 0$)
    end for
    compress($L, t$)
    if measure($\{P_N, \mathbb{1} - P_N\}$)=($\mathbb{1} - P_N$) then
        return SUCCESS, $W$
    else
        return FAILURE
    end if
end procedure
procedure iteration (live)
    if not $term[l]$ then
        if $live$ then
            $L[l] \leftarrow$ measure_coherently($\Gamma^+(S[s].proj, S[s].nbr)$)
        end if
        if $L[l]$ then
            if $live$ then
                swap_and_rotate($\Gamma^+(S[s].proj, S[s].nbr), R[tk]$)
            end if
            $t \leftarrow t + 1$
            if $t = N$ then
                $term[l + 1] \leftarrow term[l + 1] + 1$
            end if
            $S[s + 1].proj \leftarrow S[s + 1].proj + \Gamma^+(S[s].proj, S[s].nbr)$
            $s \leftarrow s + 1$
        else
            if $s = 0$ then
                $S[s].nbr \leftarrow S[s].nbr + 1 \mod m$
                if $[s].nbr = 0$ then
                    $term[l + 1] \leftarrow term[l + 1] + 1$
                end if
            else
                $S[s].nbr \leftarrow S[s].nbr + 1 \mod k$
                if $S[s].nbr = 0$ then
                    $F[l] \leftarrow F[l] + 1$
                end if
            end if
            if $F[l]$ then
                $s \leftarrow s - 1$
                $S[s + 1].proj \leftarrow S[s + 1].proj - \Gamma^+(S[s].proj, S[s].nbr)$
            end if
        end if
    else
        $term[l + 1] \leftarrow term[l + 1] + 1$
    end if
end procedure
```

register	description	size (qubits)	initial value	comment
W	work register	n	$\mathbb{1}/2^n$	random initial assignment
R	randomness register	Nk	$\mathbb{1}/2^{Nk}$	source of entropy
L	recursion log register	$m + Nd$	$\|0\ldots,0\rangle$	indicates a failed measurements and thus the start of recursion
F	return flag register	$m + Nd$	$\|0\ldots,0\rangle$	indicates return from recursion
$term$	termination register	$m + Nd$	$\|0\ldots,0\rangle$	indicates *no further operations* need to be performed
S	stack register	$2\log(N)\log(m)$	$\|0,0\rangle\ldots\|0,0\rangle$	$\log(N)$ pairs of registers labeled $(S[i].proj, S[i].nbr)$ used to indicate the projector we're fixing and the current neighbor we're checking
s	stack pointer	$\log(N)$	$\|0\rangle$	indicates the recursion level.
l	log pointer	$\log(N)$	$\|0\rangle$	indicates the next empty record.
t	randomness pointer	$\log(N)$	$\|0\rangle$	indicates the next available block of k random bits, also the number of failed coherent measurements.
$live$	modify W, R?	1	$\|0\rangle$	indicates if changes to W, R are executed (1) or skipped (0).

Table 3.1: The quantum registers and the initial state of Algorithm 3

Chapter 4

Simulating Quantum Circuits with Sparse Output Distributions

Synopsis:

We show that several quantum circuit families can be simulated efficiently classically if it is promised that their output distribution is approximately sparse i.e. the distribution is close to one where only a polynomially small, a priori unknown subset of the measurement probabilities are nonzero. Classical simulations are thereby obtained for quantum circuits which—without the additional sparsity promise—are considered hard to simulate. Our results apply in particular to a family of Fourier sampling circuits (which have structural similarities to Shor's factoring algorithm) but also to several other circuit families, such as IQP circuits. Our results provide examples of quantum circuits that cannot achieve exponential speed-ups due to the presence of too much destructive interference i.e. too many cancelations of amplitudes. The crux of our classical simulation is an efficient algorithm for approximating the significant Fourier coefficients of a class of states called computationally tractable states. The latter result may have applications beyond the scope of this work. In the proof we employ and extend sparse approximation techniques, in particular the Kushilevitz-Mansour algorithm, in combination with probabilistic simulation methods for quantum circuits.

Based on:
M. Schwarz, M. Van den Nest,
e-print arXiv:1310.6749, 2013, submitted

Changes compared to published version: minor corrections.

4.1 Introduction

In this paper we present classical algorithms for the simulation of several related classes of quantum circuits containing blocks of Quantum Fourier Transforms (QFTs). In particular, we consider n-qubit circuits with a QFT-Toffoli-QFT^{-1} block structure followed by a (partial) measurement immediately after the final QFT. Circuits of this kind are used in various quantum algorithms, most notably Shor's factoring algorithm. Whereas the circuits considered in this paper are unlikely to have an efficient classical simulation in general, the aim of this work is to analyze under which additional conditions an efficient classical simulation becomes possible. This provides an approach to identify features which are essential in the (believed) superpolynomial speed-ups achieved by, say, the factoring algorithm. In this paper we will in particular place restrictions on the *output distribution* of the circuit. In short, our results are as follows: given the promise that the output distribution is *approximately sparse* (or "*peaked*")—in the sense that only $O(\text{poly}(n))$ of the $O(2^n)$ probabilities have significant magnitude of $\Omega(1/\text{poly}(n))$—then an efficient classical simulation algorithm is provided. Not unexpectedly, Shor's algorithm does not satisfy such sparseness promise i.e. its output distribution is "superpolynomially flat". Our results thus imply that the approximate sparseness promise alone suffices to bring down the (believed) superpolynomial speed-up achieved by the factoring algorithm to the realm of a classically simulatable quantum computation. Below we provide a discussion of how our findings shed light on the factoring algorithm (see Section 4.2).

The implications of our results are twofold. First, they pose restrictions on the design of fast quantum algorithms. For example, our results show that any *exact* quantum algorithm adopting the QFT-Toffoli-QFT^{-1} block structure (or more generally the structures considered in Theorems 20-23) which has as its output state a single computational basis state containing the answer of the problem, can never achieve an exponential quantum speed-up. Given the generality of the class of circuits considered, we believe that these classical simulation results may provide useful insights for the quantum algorithms community. Second, the present results have conceptual implications as follows: the exponential speed-up found in quantum algorithms is often related to the availability of interference of probability amplitudes in this model. Indeed in several quantum algorithms, first a superposition of states is created using a QFT, then amplitudes are manipulated in some nontrivial way using reversible (classical) gates, such that in a final QFT, by means of interference, only desired basis states survive whereas the amplitudes for undesired states cancel out. Our results imply that this qualitative picture has to be refined, since too much cancelation leading to only a few classical output states (let alone a single one!) can in fact be simulated efficiently classically, and thus cannot offer exponential speed-up. Indeed, our results imply that the final probability distribution must *necessarily have super-polynomially large support* (e.g. in the same order as the full state space), in order to allow for exponential speed-up. Finally, since only polynomially many measurements can be performed efficiently on the output state—and thus only a small fraction of the necessarily large number of states can be sampled—the output distribution must have a special structure such that meaningful information can be recovered from just a few measurements. Notably, the coset state produced by Shor's algorithm (and its generalizations) has group structure which is indeed exploited in the classical post-processing step to recover the entire state space from just a few measurements (cf. Section 4.2).

The proof techniques we use to obtain our results are twofold. First, we use randomized classical simulation methods for Computationally Tractable (CT) states as developed in [Van den Nest, 2011]. Furthermore the latter methods are combined with algorithms for sublinear sparse Fourier transforms (SFTs), which have been pioneered in seminal work by Goldreich-Levin [Goldreich and Levin, 1989] and Kushilevitz-Mansour [Kushilevitz and Mansour, 1991] and which have been refined throughout the last two decades [Akavia, 2010, Akavia et al., 2003, 2006, Gilbert et al., 2005, 2002, Hassanieh et al., 2012a,b, Iwen, 2010, Mansour, 1995]. Our work also provides further extensions of the above sparse approximation techniques.

Whereas to our knowledge this is the first paper which analyzes the effect of (approximate) sparseness of the output distribution on the classical simulability of quantum circuits, from a more general point of view several works are related to the present paper (e.g. in terms of the class of quantum circuits considered or in terms of the techniques used). For example, a relevant series of papers regards [Aharonov et al., 2006, Browne, 2007, Yoran and Short, 2007], that all focus on efficient classical simulation of the QFT with the aim of understanding better the workings of Shor's factoring algorithm. In the latter context, see also [Bermejo-Vega and Nest, 2012, Van den Nest, 2012] for classical simulations of a class of circuits involving QFTs over finite abelian groups supplemented with a particular family of group-theoretic operations (Normalizer circuits). Classical simulation of CT states were considered in [Van den Nest, 2011] by one of us. In the latter work, the algorithms from Goldreich-Levin [Goldreich and Levin, 1989] and Kushilevitz-Mansour [Kushilevitz and Mansour, 1991] were applied in the context of classical simulation, albeit in a rather different context compared to the present paper, namely to analyze the role of the classical postprocessing for quantum speed-ups (more particularly in Simon's algorithm). Further work on CT states is done in [Stahlke, 2013]; the latter work also analyzes the role of interference effects in quantum speed-ups (although from a different perspective then the present paper). Below we will also make statements about classical simulability of IQP (Instanteneous Quantum Polynomial-time) circuits. In [Bremner et al., 2011] it was shown (roughly speaking) that general IQP circuits cannot be simulated efficiently, unless the polynomial hierarchy collapses. In contrast, here we show that IQP circuits with an additional sparseness promise on the output distribution, are efficiently simulable classically. Finally, in [Montanaro and Osborne, 2010] the authors consider and generalize prior work on SFTs in a different direction i.e. unrelated to classical simulation issues; they prove a quantum Goldreich-Levin theorem and use it for efficient quantum state tomography for quantum states that are approximately sparse in the Pauli product operator basis.

4.2 Main results: statements and discussion

We prove four theorems, all similar in spirit, about efficient classical simulability of classes of quantum circuits with a promise on the (approximate) sparseness of the output distributions and/or the output states. We call a probability distribution over 2^n events t-*sparse*, if only t probabilities are nonzero, and ε-*approximately* t-*sparse* if the probability distribution is ε-close in ℓ_1-distance to a t-sparse one. Throughout this paper we will work with qubit systems and sometimes indicate where generalizations of definitions and results to d-level systems are possible. The computational basis states of an n-qubit system are denoted by $|x\rangle$ where $x = x_1 \cdots x_n$ is an bit string. The set of n-bit strings will be denoted by B_n.

A key concept we build upon in this work are *computationally tractable* states introduced in [Van den Nest, 2011], which capture two key properties of simulable quantum states:

Definition 18 (Computationally Tractable (CT) states)**.** *An n-qubit state $|\psi\rangle$ is called 'computationally tractable' (CT) if the following conditions hold:*

1. *it is possible to sample in poly(n) time with classical means from the probability distribution $\mathcal{P} = \{p_x : x \in B_n\}$ defined by $p_x = |\langle x|\psi\rangle|^2$, and*

2. *upon input of any bit string x, the coefficient $\langle x|\psi\rangle$ can be computed in poly(n) time on a classical computer.*

The definition of CT states is straightforwardly generalized to states of systems of qudits. Several important state families are CT: matrix product states with polynomial bond dimension, states generated by poly-size Clifford circuits, states generated by poly-size nearest-neighbor matchgate circuits, states generated by bounded tree-width circuits (where all aforementioned circuits act on standard basis inputs). For definitions of these classes and proofs that they are CT states, we refer to [Van den Nest, 2011]. Further examples of CT states are states generated by normalizer circuits over finite Abelian groups (acting on coset states) [Bermejo-Vega and Nest, 2012, Van den Nest, 2012].

Example 19. *For our purposes it will be especially useful to point out that the following classes of states are CT [Van den Nest, 2011].*

(i) Let $|x\rangle$ be an arbitrary n-qubit computational basis state, let $\mathcal{F}$ denote the quantum Fourier transform over $\mathbb{Z}_{2^k}$ for some $k \leq n$ (acting on any subset of k qubits) and let $\mathcal{T}$ be a poly-size circuit of classical reversible gates (e.g. Toffoli gates), then the state $\mathcal{T}\mathcal{F}|x\rangle$ is CT.

(ii) Let $f : B_n \to \{1, -1\}$ be a classically efficiently computable function, then the state $|\psi_f\rangle = \frac{1}{\sqrt{2^n}} \sum f(x)|x\rangle$, where the sum is over all n-bit strings x, is CT.

One may also consider a notion of CT states in the presence of oracles (see also [Brandão and Horodecki, 2013]). We say that an n-qubit state $|\psi\rangle$ is f-CT given access to an oracle $f : \{0,1\}^m \to \{0,1\}$ (with $m = \text{poly}(n)$) if conditions (a)-(b) in Definition 18 hold when allowing, instead of poly-time classical computations, poly-many queries to the oracle. For example, if the function f in (ii) is given as an oracle, the state $|\psi_f\rangle$ in Example 19 is trivially f-CT.

Based on these definitions, we are now ready to state our main results.

Sparse output distributions

Theorem 20. *Consider a unitary n-qubit quantum circuit composed of two blocks $\mathcal{C} = U_2U_1$ with input state $|\psi_{in}\rangle$. Suppose that the following conditions are fulfilled:*

(a) the state $U_1|\psi_{in}\rangle$ obtained after applying the first block is CT;

(b) the second block U_2 is a QFT (or QFT^{-1}) modulo 2^k, for some $k \leq n$, applied to any subset S of k qubits. The circuit is followed by a measurement of the qubits in S in the computational basis, giving rise to a probability distribution $\mathcal{P}$.

(c) The distribution $\mathcal{P}$ is promised to be ε-approximately t-sparse for some $\varepsilon \leq 1/6$ and for some t (and otherwise no information about $\mathcal{P}$ is available).

Then there exists a randomized classical algorithm with runtime poly$(n, t, 1/\varepsilon, \log\frac{1}{\delta})$ which outputs (by means of listing all nonzero probabilities) an s-sparse probability distribution $\mathcal{P}'$ where $s = O(t/\varepsilon)$; with probability at least $1 - \delta$, the distribution $\mathcal{P}'$ is $O(\varepsilon)$-close to $\mathcal{P}$. Furthermore, it is possible to sample $\mathcal{P}'$ on a classical computer in poly$(n, t, 1/\varepsilon)$ time.

Thus, if the sparseness t is at most polynomially large in n, if the error ε is at worst polynomially small in n, and if $\delta = 2^{-\text{poly(n)}}$, then the classical simulation is efficient i.e. it runs in poly(n) time, and the probability of failure is exponentially small.

We emphasize that, apart from the promise (c), no information about the structure of $\mathcal{P}$ is a priori available. For example, suppose that $\mathcal{P}$ is promised to be approximately 1-sparse, where a distribution is 1-sparse if there exists a single bit string x^* which occurs with probability 1 and all other bit strings have probability 0. Then, crucially, we do not assume knowledge of the bit string x^*, i.e a priori all (potentially exponentially many in n!) bit strings are equally likely. Perhaps surprisingly, Theorem 20 implies that a good approximation of $\mathcal{P}$ can nevertheless be efficiently computed.

Since several circuit families satisfy condition (a) (recall examples above and see [Van den Nest, 2011]), Theorem 20 yields an efficient classical simulation of various types of circuits. For example, letting $|\psi_{\text{in}}\rangle$ be an arbitrary computational basis input, the block U_1 may be e.g. any poly-size Clifford circuit, nearest-neighbor matchgate circuit or bounded-treewidth circuit. A particularly interesting class of circuits, denoted by $\mathcal{A}_{\text{Shor}}$, is depicted in Figure 4.1. Note

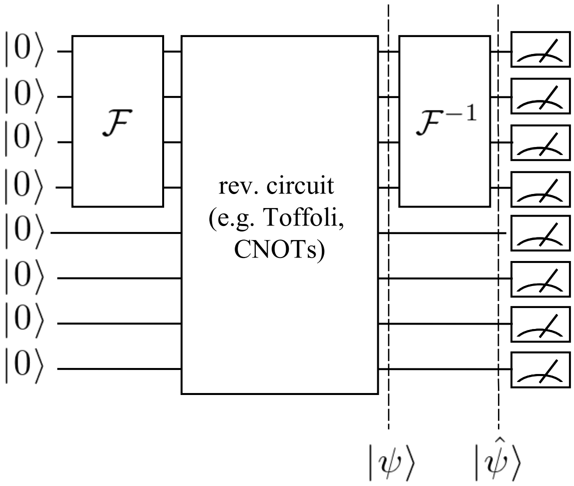

Figure 4.1: *Shor's algorithm [Shor, 1999] consists of (1) a quantum Fourier transform (QFT) on a subset of qubits, (2) a block of reversible gates (a modular exponentiation circuit), and (3) an inverse QFT on the same subset qubits. Note that the state $|\psi\rangle$ obtained after the first QFT is a computationally tractable (CT) state. Thus conditions (a) and (b) of Theorem 20 are satisfied. However the output distribution of Shor's algorithm is* not *sparse in general, as required by our algorithm (cf. condition (c)).*

that Shor's factoring algorithm belongs to the class $\mathcal{A}_{\text{Shor}}$. It is easily verified that, for any $\mathcal{A}_{\text{Shor}}$ circuit, the state of the quantum register immediately before the second QFT is CT (recall Example 19 (i) from above). Thus any circuit in $\mathcal{A}_{\text{Shor}}$ *which, in addition, satisfies the sparseness condition (c) of Theorem 20* can be simulated efficiently classically. Upon closer inspection of Shor's factoring algorithm, one finds that its output distribution $\mathcal{P}_{\text{Shor}}$ generally contains super-polynomially many nonzero probabilities and thus (non-surprisingly) Theorem 20 does not yield an efficient classical simulation of the factoring algorithm. More precisely, the size of the support of the flat distribution $\mathcal{P}_{\text{Shor}}$ equals the multiplicative order r of a randomly chosen integer x modulo N. For a general integer N, the order is conjectured to be $\Omega(N/\log(N))$ on average over all N [Arnold, 2005, Kurlberg and Pomerance, 2013]. In the case of RSA, with $N = pq$, the primes p and q might be *chosen* such that w.h.p. $r \approx N/4$ [Shor, 2011]. Nevertheless it is interesting that the mere promise of (approximate) sparsity of the output distribution suffices to arrive at an efficient classical simulation for all $\mathcal{A}_{\text{Shor}}$ circuits, without otherwise restricting the allowed operations. This implies that the feature that $\mathcal{P}_{\text{Shor}}$ is sufficiently flat is an essential ingredient in the (believed) superpolynomial speed-up achieved by Shor's factoring algorithm.

Another observation is the following. Any quantum circuit $\mathcal{A}$ satisfying (a)-(b) in Theorem 20 (for example any $\mathcal{A}_{\text{Shor}}$ circuit) which, when implemented on a quantum computer, aspires to deliver a superpolynomial speed-up over classical computers, must generate a distribution $\mathcal{P}$ which cannot be well-approximated by a poly(n)-sparse distribution. At the same time, at most poly(n) repetitions of $\mathcal{A}$ are allowed if the total computational cost is to be polynomially bounded, yielding only poly(n) samples of $\mathcal{P}$. In other words, one only has access to 'few' samples of a distribution which has support on a 'large' number of outputs. Yet somehow these few samples should contain sufficient information to extract the final result of the computation with high probability (working within the standard bounded-error setting). This point is nicely illustrated by considering again the factoring algorithm (or more generally the abelian hidden subgroup algorithm). Here the output distribution is (close to) the uniform distribution over an unknown *group* H (and determining this group is essentially the goal of the algorithm) and the final measurement only yields a small set of $O(\log |H|)$ randomly chosen elements of H. However, since such a small set of randomly generated group elements is with high probability a generating set of the group, a small number of measurements indeed suffices to determine the entire group H.

Theorem 20 can be extended by allowing the block U_2 to comprise tensor product operations, instead of the QFT:

Theorem 21. *The conclusions of Theorem 20 also apply if condition (b) is replaced by*

(b') the second block U_2 is an arbitrary tensor product unitary operation $U_2 = u_1 \otimes \cdots \otimes u_n$. The circuit is followed by a measurement of an arbitrary subset of qubits S in the computational basis, giving rise to a probability distribution $\mathcal{P}$.

In addition, the conclusions of Theorem 20 also apply when U_2 is a tensor product operation as in (b'), but now for quantum algorithms operating on the Hilbert space $\mathcal{H} = \mathbb{C}_{d_1} \otimes \cdots \otimes \mathbb{C}_{d_n}$ with $d_i = O(1)$ but otherwise arbitrary, i.e. $\mathcal{H}$ is a system of n qudits of possibly different dimensions.

A first example of the setting considered in Theorem 21 regards the family of IQP circuits (Instantaneous Quantum Polynomial time [Shepherd and Bremner, 2009]). Here the input is an n-qubit computational basis state $|x\rangle$ and the circuit consists of gates of the form $\exp[i\theta T]$ where θ is an arbitrary real parameter and where T is a tensor product of the form $T = T_1 \otimes \cdots \otimes T_n$ with $T_i \in \{I, X\}$. Since $X = HZH$, every IQP circuit $\mathcal{C}$ can be written as $\mathcal{C} = H^{\otimes n}\mathcal{C}' H^{\otimes n}$ where $\mathcal{C}'$ is obtained by replacing each gate $\exp[i\theta T]$ by $\exp[i\theta T']$ where $T' = T_1' \otimes \cdots \otimes T_n'$ with $T_i' = HT_iH$. Thus T' is a tensor product of Z operators and identity gates and hence each gate $e^{i\theta T'}$ is diagonal in the computational basis. Setting $U_1 := \mathcal{C}' H^{\otimes n}$ and $U_2 := H^{\otimes n}$ we find that conditions (a)-(b') of Theorem 21 are fulfilled; indeed it is straightforward to show that $\mathcal{C}' H^{\otimes n}|x\rangle$ is a CT state. Thus Theorem 21 shows that any IQP circuit with an approximately sparse output distribution can be simulated efficiently classically. This result is particularly interesting when compared to a hardness-of-simulation result obtained for general IQP circuits (i.e. without sparseness promise) in [Bremner et al., 2011]. In the latter work it was shown that an efficient, approximate classical simulation of IQP circuits (w.r.t. a certain multiplicative approximation) would imply a collapse of the polynomial hierarchy.

A second example of the setting considered in Theorem 21 is the following. Consider a finite, possibly non-abelian group G given as a direct product of n individual groups, $G = G_1 \times \cdots \times G_n$ where the order of each G_i is $O(1)$. Define a Hilbert space $\mathcal{H}_G$ with computational basis vectors $|g\rangle = |g_1\rangle \otimes \cdots \otimes |g_n\rangle$ labeled by group elements $g = (g_1, \dots, g_n) \in G$. The space $\mathcal{H}_G$ is naturally associated with a tensor product of n individual spaces, each of constant dimension. We may now consider quantum circuits of the following kind. The total Hilbert space is $\mathcal{H}_G \otimes \mathcal{H}_n$ where $\mathcal{H}_n$ is an n-qubit system. In analogy to Figure 4.1, we consider circuits of the block structure $\mathcal{C} = A_3 A_2 A_1$ where A_1 is the QFT over G acting on the register $\mathcal{H}_G$, A_2 is an arbitrary poly-size circuit of classical reversible gates acting on the entire system and A_3 is the inverse QFT over G. The input is $|1_G, 0_n\rangle$ where 1_G is the neutral element in G and 0_n denotes the all-zeros n-bit string; the circuit is followed by measurement of the system $\mathcal{H}_G$ in the basis $\{|g\rangle\}$. Circuits of this kind are of interest in the context of quantum algorithms for the (non-abelian) Hidden subgroup problem (see e.g. [Alagic et al., 2007, Lomont, 2004]). For a definition of the QFT over a finite group we refer to e.g. [Moore et al., 2006]; here it suffices to mention that the QFT over a product group $G = G_1 \times \cdots \times G_n$ is a tensor product operator. Furthermore it is easily verified (recall also the discussion on CT states above) that condition (a) in Theorem 20 is satisfied with $U_1 \equiv A_2 A_1$. Thus Theorem 21 implies that any quantum circuit of this kind which has an approximately sparse output distribution can be simulated classically. This gives an example of a quantum circuit family comprising non-abelian QFTs (albeit of a restricted kind) which can be simulated classically. For other examples of simulations of non-Abelian QFTs we refer to [Bermejo-Vega, 2011].

Sparse output states

Let us present two more results regarding quantum circuits of the kinds considered in Theorem 20 and Theorem 21, when promised that the output *state* is approximately sparse. In this case we show how an approximation of the latter output state can be efficiently determined by means of a classical randomized algorithm.

An n-qubit state $|\varphi\rangle$ is called ε-approximately t-sparse if there exists a state $|\varphi'\rangle$ which is ε-close to $|\psi\rangle$ and for which at most t amplitudes $\langle x|\varphi'\rangle$ (with $|x\rangle$ computational basis states) are nonzero (see also section 4.4).

Theorem 22. *Consider a unitary n-qubit quantum circuit composed of two blocks $\mathcal{C} = U_2 U_1$ with input state $|\psi_{in}\rangle$. Suppose that the following conditions are fulfilled:*

(a) the state $U_1|\psi_{in}\rangle$ obtained after applying the first block is CT;

(b) the second block U_2 is the QFT modulo 2^n or its inverse.

(c) The final state $|\psi_{out}\rangle = \mathcal{C}|\psi_{in}\rangle$ is promised to be $\sqrt{\varepsilon}$-approximately t-sparse for some $\varepsilon \le 1/6$ and some t.

Then there exists a randomized classical algorithm with runtime poly$(n, t, 1/\varepsilon, \log\frac{1}{\delta})$ *which outputs (by means of listing all nonzero amplitudes) an* s*-sparse state* $|\psi\rangle$ *which, with probability at least* $1-\delta$*, is* $O(\sqrt{\varepsilon})$*-close to* $|\psi_{out}\rangle$*, where* $s = O(t/\varepsilon)$*.*

Theorem 23. *The conclusions of Theorem 22 also apply if condition (b) is replaced by*

(b') the second block U_2 *is an arbitrary tensor product unitary operation* $U_2 = u_1 \otimes \cdots \otimes u_n$*.*

In addition, the conclusions of Theorem 22 also apply when U_2 *is a tensor product operation as in (b'), but now for quantum algorithms operating on the Hilbert space* $\mathcal{H} = \mathbb{C}_{d_1} \otimes \cdots \otimes \mathbb{C}_{d_n}$ *with* $d_i = O(1)$ *but otherwise arbitrary.*

Theorem 22 and Theorem 23 are closely connected to an important result in theoretical computer science, namely the Kushilevitz-Mansour (KM) algorithm [Kushilevitz and Mansour, 1991]: if one has oracle access to a Boolean function $f : B_n \to \{1, -1\}$ which is promised to have an approximately sparse Fourier spectrum, it is possible to compute a sparse approximation of f in polynomial time. We connect our result to Kushilevitz-Mansour by considering Theorem 23 for an n-qubit system where

$$|\psi_{\text{in}}\rangle \equiv |\psi_f\rangle = \frac{1}{2^{n/2}} \sum_x f(x)|x\rangle \tag{4.1}$$

is a CT state, $U_1 \equiv I$ and $U_2 \equiv H^{\otimes n}$ where H is the Hadamard gate. Then Theorem 23 implies that if $H^{\otimes n}|\psi_f\rangle$ is promised to be approximately sparse, then a sparse approximation of the latter state can be computed efficiently. This is effectively (a version of) the KM result, stated in the language of quantum computing. Similarly, Theorem 22 relates to a version of the KM result [Mansour, 1995] considered for transformations of Boolean functions under the Fourier transform over $\mathbb{Z}_{2^n}$. The proof method of the KM theorem, suitably generalized to our setting at hand, will be an important tool for us.

Computing significant weights

Whereas Theorems 20 to 23 involve a promise about the approximate sparseness of the output distributions/states, our final result does not. The following theorem asserts that, for CT states expanded in the Fourier basis, it is possible to efficiently determine (in a suitable approximate and probabilistic sense) all Fourier coefficients which are larger than some threshold value; a similar result also holds for CT states expanded in product bases. The result is in the present paper mainly used as a technique in the proof of Theorems 22 and 23 (similar to the proof of Kushilevitz-Mansour). However we believe it may be of independent interest, given the broadness of the class of CT states and the frequent usage of Fourier transforms.

Let $\mathbb{Z}_{2^n}$ denote the cyclic group of integers modulo 2^n. Any n-bit string x is identified with an element of $\mathbb{Z}_{2^n}$ via the binary expansion. Recall that the quantum Fourier transform over $\mathbb{Z}_{2^n}$ is the following n-qubit unitary operator:

$$\mathcal{F}_{2^n} = \frac{1}{\sqrt{2^n}} \sum_{x,y \in \mathbb{Z}_{2^n}} \exp\left(\frac{2\pi i x y}{2^n}\right) |x\rangle\langle y|\,. \tag{4.2}$$

and the *Fourier basis* is simply the orthonormal basis $\{|F_x\rangle : x \in B_n\}$ defined by $|F_x\rangle = \mathcal{F}_{2^n}|x\rangle$.

Theorem 24. *Let $|\psi\rangle$ be an n-qubit CT state and consider its expansion in the Fourier basis:*

$$|\psi\rangle = \sum \hat{\psi}_x |F_x\rangle. \tag{4.3}$$

There exists a randomized classical algorithm with runtime poly$(n, \frac{1}{\theta}, \log\frac{1}{\pi})$ which outputs a list $L = \{x^1, \ldots, x^l\}$ where $l \le 2/\theta$ and where each x^i is an n-bit string such that, with probability at least $1 - \pi$:

(a) for all $y \in L$, it holds that $|\hat{\psi}_x|^2 \ge \frac{\theta}{2}$;

(b) every k-bit string x satisfying $|\hat{\psi}_x|^2 \ge \theta$ belongs to the list L;

Furthermore, given any $x \in B_n$, there exists a classical algorithm with runtime poly$(n, 1/\varepsilon, \log\frac{1}{\delta})$ which, with probability at least $1 - \delta$, outputs an ε-approximation of $\hat{\psi}_x$. Finally, the above results also holds if the Fourier basis is replaced by a product basis $\{U|x\rangle\}$ where $U = U_1 \otimes \cdots \otimes U_n$ is an arbitrary tensor product unitary operator.

4.3 Proof outline and organization of the paper

In Section 4.4 we discuss ε-approximately t-sparse distributions and states. A key property will be Lemma 26 where we show that the large probabilities contain most of the information of an approximately sparse distribution i.e. discarding the small probabilities does not introduce too much error.

It will be a key point in our proofs that the output distributions of the quantum circuits considered in Theorems 20 to 23, as well as a suitable subset of their marginal distributions, are what will be called here *additively approximable*. The latter are distributions whose individual probabilities can be efficiently approximated with a randomized classical algorithm with a performance in terms of error and success probability which is similar to the one given by the Chernoff bound. Our analysis of additively approximable distributions (Section 4.5 and Section 4.6), which is a significant component in the proofs of our main results, will not make reference to quantum computing (the latter is done as of Section 4.7). In Section 4.5, we introduce the notion of additively approximable distributions and develop their properties. An important feature will be established in Theorem 30 where we show that, for any probability distribution which is itself additively approximable and for which a designated subset of its marginals are additively approximable as well, it is possible to efficiently determine (in a suitable approximate sense) those probabilities which are larger than some given, sufficiently large, threshold value. This lemma, in combination with Lemma 26 mentioned above, will yield an efficient algorithm to (approximately) sample any ε-approximately t-sparse distribution which is additively approximable and whose marginals are as well; this algorithm is given in Section 4.6 (Theorem 31). The results developed in Section 4.5 to Section 4.6 will follow the general proof idea of the Kushilevitz-Mansour theorem [Goldreich and Levin, 1989, Kushilevitz and Mansour, 1991].

In Section 4.7 we recall classical simulation properties of CT states. Finally, in Section 4.8 the proofs of our main results are given: the main strategy is to show that the output distributions of the circuits considered in our main theorems, as well as their marginals, are additively approximable.

4.4 Approximate sparseness

4.4.1 Basic definitions

We call a quantum state $|\varphi\rangle$ *t-sparse* (relative to the computational basis), if at most t amplitudes $\langle x|\varphi\rangle$ are nonzero. We will use the standard ℓ_2-norm as as the natural distance measure for two pure states. Thus we will call two quantum states $|\varphi\rangle, |\psi\rangle$ *ε-close*, if $|| \, |\varphi\rangle - |\psi\rangle \, ||_2 \leq \varepsilon$. We call a normalized pure state $|\varphi\rangle$ *ε-approximately t-sparse* if there exists a, not necessarily normalized, t-sparse vector which is ε-close to $|\varphi\rangle$. In this paper we will mostly be interested in a sparseness t which scales at most polynomially with the number of qubits n, and in an error ε which is worst polynomially small in n. Note that in the definition of approximate sparseness we allow the t-sparse vector to be an unnormalized state (this will be a convenient definition in our proofs). However, if $|\varphi\rangle$ is ε-approximately t-sparse and if ε is sufficiently small (namely $\varepsilon \leq 0.5$), there always exists a *normalized* t-sparse state $|\varphi'\rangle$ which is $O(\varepsilon)$-close to $|\varphi\rangle$ as well (see Section 4.4.2).

Similar to sparse quantum states, we call a probability distribution $\mathcal{P} = \{p_x : x \in B_n\}$ on the set of n-bit strings t-sparse if at most t of its probabilities p_x are nonzero. The distance between two probability distributions $\mathcal{P}$ and $\mathcal{P}'$ will be measured in terms of the total variation distance, defined by

$$\|\mathcal{P} - \mathcal{P}'\|_1 = \sum |p_x - p'_x|. \tag{4.4}$$

We say that $\mathcal{P}$ is ε-approximately t-sparse if there exists a t-sparse vector $v = (v_x : x \in B_n)$ such that $\sum |p_x - v_x| \leq \varepsilon$. The entries v_x may a priori be arbitrary complex numbers. However, similar to above, if $\mathcal{P}$ is ε-approximately t-sparse and if ε is sufficiently small, there always exists a *normalized probability distribution* $\mathcal{P}'$ which is t-sparse and such that $\|\mathcal{P} - \mathcal{P}'\|_1 \leq O(\varepsilon)$ (see Section 4.4.2).

The support of a probability distribution $\mathcal{P} = \{p_x : x \in B_n\}$ is the set of all x for which $p_x \neq 0$. If $A \subseteq B_n$, the restriction of $\mathcal{P}$ to A is the subnormalized distribution $\{q_x : x \in B_n\}$ defined by

$$q_x = \begin{cases} p_x & \text{if } x \in A \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Similarly, the support of an n-qubit state is the set of all x for which $\langle x|\varphi\rangle \neq 0$. If $A \subseteq B_n$, the restriction of $|\varphi\rangle$ to A is the subnormalized state

$$\sum_{x \in A} \langle x|\varphi\rangle |x\rangle. \tag{4.6}$$

4.4.2 Basic properties

Let $\mathcal{P} = \{p_x : x \in B_n\}$ be an arbitrary probability distribution. Let $A_t \subseteq B_n$ be a subset which, roughly speaking, contains t bit strings corresponding to the t largest probabilities of $\mathcal{P}$. More formally, A_t satisfies the properties (i) $|A_t| = t$ and (ii) $p_x \geq p_y$ for all $x \in A_t$ and $y \notin A_t$. Note that there may be more than one set A_t with this property (e.g. if multiple probabilities happen to be equal). For our purposes the particular choice of A_t will however be irrelevant. Let $\mathcal{P}[t]$ denote the restriction of $\mathcal{P}$ to A_t. Note that $\mathcal{P}[t]$ is t-sparse. Furthermore it is straightforward to show that, for any t-sparse vector $v = (v_x : x \in B_n)$ (where the v_x may be arbitrary complex numbers), one has $\|\mathcal{P}[t] - \mathcal{P}\|_1 \leq \|v - \mathcal{P}\|_1$ i.e. $\mathcal{P}[t]$ has minimal distance to $\mathcal{P}$ among all such t-sparse v's. It follows that $\mathcal{P}$ is ε-approximately t-sparse iff

$$\|\mathcal{P} - \mathcal{P}[t]\|_1 \leq \varepsilon. \tag{4.7}$$

Next we show that, for any ε-approximately t-sparse distribution $\mathcal{P}$ with $\varepsilon \leq 0.5$ there always exists a t-sparse *normalized* distribution $\mathcal{P}'$ which is $O(\varepsilon)$-close to $\mathcal{P}$. To see this, set $\mathcal{P}' := \mathcal{P}[t]/\|\mathcal{P}[t]\|_1$. Owing to Eq. (4.7) we have

$$1 - \varepsilon \leq \|\mathcal{P}[t]\|_1 \leq 1. \tag{4.8}$$

We then find

$$\begin{aligned} \|\mathcal{P}' - \mathcal{P}\|_1 &= \frac{\|\mathcal{P}[t] - \|\mathcal{P}[t]\|_1 \cdot \mathcal{P}\|_1}{\|\mathcal{P}[t]\|_1} \leq \frac{\|\mathcal{P}[t] - \|\mathcal{P}[t]\|_1 \cdot \mathcal{P}\|_1}{1-\varepsilon} \\ &\leq \frac{\|\mathcal{P}[t] - \mathcal{P}\|_1}{1-\varepsilon} + \frac{(1 - \|\mathcal{P}[t]\|_1) \cdot \|\mathcal{P}\|_1}{1-\varepsilon} \leq \frac{2\varepsilon}{1-\varepsilon}. \end{aligned} \tag{4.9}$$

Here in the equality we used the definition of $\mathcal{P}[t]$; in the first inequality we used Eq. (4.8); in the second inequality we used the triangle inequality; finally we used Eq. (4.7) and Eq. (4.8). Then, if $\varepsilon \leq 0.5$, we have $\|\mathcal{P}' - \mathcal{P}\|_1 \leq 4\varepsilon$.

Let $|\varphi\rangle$ be an n-qubit state. In analogy with above, let $A_t \subseteq B_n$ be a subset which, roughly speaking, contains t bit strings corresponding to the t largest amplitudes of $|\varphi\rangle$. More formally, A_t satisfies (i) $|A_t| = t$ and (ii) $|\langle x|\varphi\rangle| \geq |\langle y|\varphi\rangle|$ for all $x \in A_t$ and $y \notin A_t$. Letting $|\varphi[t]\rangle$ denote the restriction of $|\varphi\rangle$ to A_t, it is straightforward to show that $|\varphi[t]\rangle$ has minimal ℓ_2-distance to $|\varphi\rangle$ among all t-sparse vectors. It follows that $|\varphi\rangle$ is ε-approximately t-sparse iff

$$\||\varphi\rangle - |\varphi[t]\rangle\|_2 \leq \varepsilon. \tag{4.10}$$

Fully analogous to above, for any ε-approximately t-sparse state $|\varphi\rangle$ with $\varepsilon \leq 0.5$ there always exists a t-sparse *normalized* state $|\varphi'\rangle$ which is $O(\varepsilon)$-close to $|\varphi\rangle$. The state $|\varphi'\rangle := |\varphi[t]\rangle/\||\varphi[t]\rangle\|_2$ does the job.

Let $|\varphi\rangle$ be an n-qubit pure state and let $\mathcal{P}$ be the probability distribution arising from measuring all qubits of $|\varphi\rangle$ in the computational basis. We may then ask whether $\mathcal{P}$ is approximate sparse or whether the full state $|\varphi\rangle$ is approximately sparse, where in the former case closeness is measured w.r.t. total variation distance and in the latter case it is measured w.r.t. ℓ_2 distance. Next we show that both notions of approximate sparseness are equivalent up to a square-root rescaling of the accuracy ε (which is mostly harmless if one is ultimately interested in $\varepsilon = 1/\text{poly}(n)$, as we will mostly be in this paper).

Lemma 25. *Let $|\varphi\rangle$ be an n-qubit pure state and let $\mathcal{P}$ be the probability distribution arising from measuring all qubits of $|\varphi\rangle$ in the computational basis. Then $|\varphi\rangle$ is $\sqrt{\varepsilon}$-approximately t-sparse (relative to the ℓ_2-distance, as above) iff $\mathcal{P}$ is ε-approximately t-sparse (relative to the total variation distance, as above).*

Proof. Define $p_x = |\langle x|\varphi\rangle|^2$ for all x. As above, let A_t be a set of t n-bit string satisfying $p_x \geq p_y$ for all $x \in A_t$ and $y \notin A_t$. This is (trivially) equivalent to $|\langle x|\varphi\rangle| \geq |\langle y|\varphi\rangle|$ for all $x \in A_t$ and $y \notin A_t$. Let $\mathcal{P}[t]$ denote the restriction of $\mathcal{P}$ to A_t and similarly $|\varphi[t]\rangle$ is the restriction of $|\varphi\rangle$ to A_t. Recall that $|\varphi\rangle$ is $\sqrt{\varepsilon}$-approximately t-sparse iff $\||\varphi\rangle - |\varphi[t]\rangle\|_2 \leq \sqrt{\varepsilon}$ and that $\mathcal{P}$ is ε-approximately t-sparse iff $\|\mathcal{P} - \mathcal{P}[t]\|_1 \leq \varepsilon$. A straightforward application of definitions now shows that $\||\varphi\rangle - |\varphi[t]\rangle\|^2 = \|\mathcal{P} - \mathcal{P}[t]\|$, since both expressions coincide with

$$\sum_{x \notin A_t} |\langle x|\varphi\rangle|^2. \tag{4.11}$$

This shows that $\||\varphi\rangle - |\varphi[t]\rangle\|_2 \leq \sqrt{\varepsilon}$ iff $\|\mathcal{P} - \mathcal{P}[t]\|_1 \leq \varepsilon$. □

4.4.3 Sparse distributions have large coefficients

The next lemma shows that, for an approximately sparse probability distribution, the 'small' probabilities can be ignored without introducing much error. This property will be important in the proof of our main results, in combination with Theorem 30 which states that the large probabilities can be efficiently computed for certain distributions. The following lemma is also closely related to [Kushilevitz and Mansour, 1991, Lemma 3.11]

Lemma 26. *Let $\mathcal{P} = \{p_x : x \in B_n\}$ be an ε-approximately t-sparse probability distribution. Define $B_{\varepsilon,t}$ to be the subset of all bit strings x such that $p_x \geq \varepsilon/t$. Define the subnormalized distribution $\mathcal{Q}_{\varepsilon,t}$ to be the restriction of $\mathcal{P}$ to $B_{\varepsilon,t}$. Then $\mathcal{Q}_{\varepsilon,t}$ is $O(\varepsilon)$-close to $\mathcal{P}$. More precisely $\|\mathcal{Q}_{\varepsilon,t} - \mathcal{P}\|_1 \leq 2\varepsilon$.*

Proof. Let $A_t \subseteq B_n$ and $\mathcal{P}[t]$ be defined as in Section 4.4.2. Recall that $\|\mathcal{P}[t] - \mathcal{P}\|_1 \leq \varepsilon$ owing to the approximate sparseness of $\mathcal{P}$. Furthermore construct $\mathcal{P}'$ as follows: start from $\mathcal{P}[t]$ and set all probabilities with magnitudes $\leq \frac{\varepsilon}{t}$ to zero; let C denote the support of $\mathcal{P}'$. Note that $C \subseteq A_t$ and thus $|A_t \setminus C| \leq t$. Furthermore $p_x \leq \varepsilon/t$ for every $x \in A_t \setminus C$. Then

$$\begin{aligned} \left\|\mathcal{P} - \mathcal{P}'\right\|_1 &\leq \left\|\mathcal{P} - \mathcal{P}[t]\right\|_1 + \left\|\mathcal{P}[t] - \mathcal{P}'\right\|_1 \\ &\leq \varepsilon + \sum_{x \in A_t \setminus C} p_x \leq \varepsilon + t \cdot \frac{\varepsilon}{t} = 2\varepsilon. \end{aligned} \tag{4.12}$$

Note also that $C \subseteq B_{\varepsilon,t}$ since $p_x \geq \varepsilon/t$ for all $x \in C$. Thus both $\mathcal{P}'$ and $\mathcal{Q}_{\varepsilon,t}$ are restrictions of $\mathcal{P}$, and that the support $B_{\varepsilon,t}$ of $\mathcal{Q}_{\varepsilon,t}$ contains the support C of $\mathcal{P}'$. This implies that $||\mathcal{P} - \mathcal{Q}_{\varepsilon,t}||_1 \leq ||\mathcal{P} - \mathcal{P}'||_1$. Together with (4.12) this proves the result. □

An analogous result holds for approximately sparse quantum states. We do not make it explicit here since it will not be needed in our proofs of the main results.

4.5 Additively approximable probability distributions

4.5.1 Definition and basic properties

The Chernoff-Hoeffding bound is a basic tool in probability theory which will be used in this work. Whereas the bound is usually stated for real-valued random variables, here we state a simple generalization to the complex-valued case, which follows from the real-valued case by bounding real and imaginary parts of independently.

Lemma 27 (Chernoff-Hoeffding bound)**.** *Let $X_1, \ldots, X_T$ be i.i.d. complex-valued random variables with $E := \mathbf{E}X_i$ and $|X_i| \leq 1$ for every $i = 1, \ldots, T$. Then with $T = \frac{4}{\varepsilon^2}\log(\frac{4}{\delta})$ we have*

$$\Pr\left\{\left|\frac{1}{T}\sum_{i=1}^{T} X_i - E\right| \leq \varepsilon\right\} \geq 1 - \delta$$

A proof of Lemma 27 can be found in Section 4.10.1. The main application of the Chernoff bound used in this work will be in the following context. Let $F : B_n \to \mathbb{C}$ be an efficiently computable complex function (i.e. computable in polynomial time on a deterministic classical computer) satisfying $|F(x)| \leq 1$ for all $x \in B_n$ and let $\mathcal{P} := \{p_x : x \in B_n\}$ be a probability distribution on the set of n-bit strings which can be sampled in poly(n) time on a randomized classical computer. Then a direct application of the Chernoff-Hoeffding bound shows that there exists a classical randomized algorithm to estimate

$$\langle F \rangle := \sum p_x F(x) \tag{4.13}$$

with error ε and probability at least $1 - \delta$ in poly$(n, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ time. This means that in poly(n) time it is possible to achieve an accuracy $\varepsilon = 1/\text{poly}(n)$ and exponentially small failure probability $\delta = 2^{-\text{poly(n)}}$.

Next we introduce a definition for functions that are approximable with randomized classical algorithms having a performance in terms of error ε and failure probability δ that is analogous to those obtained by applying the Chernoff bound (see also [Bordewich et al., 2005] for a related notion of additive approximations).

Definition 28. *A function $f : B_n \to \mathbb{C}$ is said to be additively approximable if there exists a randomized classical algorithm with runtime poly$(n, 1/\varepsilon, \log\frac{1}{\delta})$ which, on input of an n-bit bit string x, outputs with probability at least $1 - \delta$ an ε-approximation of $f(x)$. A probability* *distribution $\mathcal{P} = \{p_x\}$ on the set of n-bit strings is said to be additively approximable if the function $x \to p_x$ is additively approximable.*

Note that any $\mathcal{P}$ which can be sampled classically in poly(n) time is additively approximable since each individual probability can essentially be computed by sampling the distribution. More precisely, to estimate p_x, write $p_x = \sum \delta(x, y)p_y$ where $\delta(x, y)$ equals 1 if $x = y$ and 0 otherwise. We have thus rewritten p_x as the expectation value of $F \equiv \delta(x, \cdot)$ which is a poly(n)-time computable function satisfying $|F(x)| \leq 1$ for all $x \in B_n$. The discussion above Definition 28 then immediately implies that $\mathcal{P}$ is additively approximable.

In the example discussed in Eq. (4.13) we found that $\langle F\rangle$ can be efficiently approximated provided that F was efficiently computable on a deterministic computer. In the following lemma it is shown that the same performance in estimating $\langle F\rangle$ can be achieved even when F is only additively approximable. The argument is a basic application of the Chernoff bound.

Lemma 29. *Let $F : B_n \to \mathbb{C}$ be an additively approximable function and let $\mathcal{P} := \{p_x : x \in B_n\}$ be a probability distribution which can be sampled in poly(n) time on a classical computer. Then there exists a classical randomized algorithm to estimate $\langle F\rangle := \sum p_x F(x)$ with error ε and probability $1 - \delta$ in poly$(n, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ time.*

Proof. By generating $K = O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ bit strings $x^1, \ldots, x^K$ from the distribution $\mathcal{P}$, the inequality

$$\left|\frac{1}{K}\sum_{i=1}^{K} F(x^i) - \langle F\rangle\right| \leq \varepsilon/2 \tag{4.14}$$

holds with probability at least $1 - \delta/2$, owing to the Chernoff bound. Then, for each x^i we compute a complex number c_i satisfying $|c_i - F(x^i)| \leq \frac{\varepsilon}{2}$ with probability at least $1 - \delta/(2K)$. Since F is additively approximable, each c_i can be computed in time

$$T = \text{poly}(n, \tfrac{2}{\varepsilon}, \log\tfrac{2K}{\delta}) = \text{poly}(n, \tfrac{1}{\varepsilon}, \log\tfrac{1}{\delta}). \tag{4.15}$$

Thus the total runtime of computing all values c_i is $KT = \text{poly}(n, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$. The total probability that each c_i is $\frac{\varepsilon}{2}$-close to $F(x^i)$ *and* that (4.14) holds is at least

$$(1 - \tfrac{\delta}{2}) \cdot (1 - \tfrac{\delta}{2K})^K \geq (1 - \tfrac{\delta}{2}) \cdot (1 - \tfrac{\delta}{2}) \geq 1 - \delta \tag{4.16}$$

where we have repeatedly used that $(1 - a)^r \geq 1 - ra$ for all positive integers r and for all $a \in [0, 1]$. It follows that, with probability at least $1 - \delta$, we have

$$|\frac{1}{K}\sum_{i=1}^{K} c_i - \langle F\rangle| \leq \varepsilon \tag{4.17}$$

by using the triangle inequality.

□

4.5.2 Estimating large coefficients

The following theorem contains the property of additive approximations which is most important for our purposes. It is a statement that, for distributions which are additively approximable and for which also (a designated subset of) the marginals are additively approximable, there exists an efficient algorithm to determine those probabilities which are larger than some given threshold value. The proof technique is a type of binary search algorithm which is a direct generalization of the proof of the Kushilevitz-Mansour algorithm [Kushilevitz and Mansour, 1991].

Theorem 30. *Let $\mathcal{P} = \{p_x : x \in B_k\}$ be a probability distribution. Let $\mathcal{P}_m$ denote the marginal probability distribution of the first m bits, for every m ranging from 1 to k (with $\mathcal{P}_k \equiv \mathcal{P}$). Suppose that all distributions $\mathcal{P}_m$ are additively approximable. Then the following holds: given $\theta, \pi > 0$, there exists a randomized classical algorithm with runtime poly$(k, \frac{1}{\theta}, \log\frac{1}{\pi})$ which outputs a list $L = \{x^1, \ldots, x^l\}$ where $l \le 2/\theta$ and where each x^i is an k-bit string such that, with probability at least $1 - \pi$:*

(a) for all $y \in L$, it holds that $p(y) \ge \frac{\theta}{2}$;

(b) every k-bit string x satisfying $p(x) \ge \theta$ belongs to the list L;

Proof. For any integer $m \le k$ we denote by $p(x_1 \cdots x_m)$ the marginal probability of the bit string $x_1 \cdots x_m$. We point out the basic fact that

$$p(x_1 \cdots x_{m-1}) \ge p(x_1 \cdots x_{m-1} x_m) \tag{4.18}$$

for all m and for all x_j's.

The algorithm will consist of k steps. In each step we construct a list L_m containing a certain collection of m-bit strings, where m ranges from 1 to k. The final list L_k will satisfy (a)-(b) with probability at least $1 - \pi$. In the algorithm we will repeatedly invoke that each $\mathcal{P}_m$ is additively approximable; whenever an additive approximation of any $\mathcal{P}_m$ will be considered, we will set the required probability of success to be at least $1 - \delta$ with $\delta := \theta\pi/2k$ and the accuracy to be $\varepsilon := \theta/4$. Each single estimate of such a probability can be done in time

$$N_{\text{single}} = \text{poly}(k, \frac{1}{\varepsilon}, \log\frac{1}{\delta}) = \text{poly}(k, \frac{1}{\theta}, \log\frac{1}{\pi}). \tag{4.19}$$

Step 1. The list $L_1 \subseteq B_1 \equiv \{0, 1\}$ is computed as follows. We use that $\mathcal{P}_1$ is additively approximable and compute $p(0)$ (i.e. the probability of the outcome 0 on the first bit). More formally, we compute a number $c(0)$ satisfying

$$|c(0) - p(0)| \le \theta/4 \tag{4.20}$$

with probability at least $1 - \delta$. If $c(0) \ge 3\theta/4$ then define the bit 0 to belong to the list L_1. Analogously we compute $c(1)$ as an approximation of $p(1)$ and add the bit 1 to L_1 if $c(1) \ge 3\theta/4$.

Step 2. To compute the list $L_2 \subseteq B_2 \equiv \{00, 01, 10, 11\}$ we use that $\mathcal{P}_2$ is additively approximable as follows. For every $x \in L_1$ and $u \in \{0, 1\}$ we compute an $\theta/4$-approximation of $p(xu)$ with probability at least $1 - \delta$, yielding a number $c(xu)$ in analogy to Step 1. If $c(xu) \geq 3\theta/4$ then we add the bit pair xu to the list L_2.

Steps 3-k. The above procedure is continued for all $m = 3 \cdots k$ where in the m-th step we use that $\mathcal{P}_m$ is additively approximable. To compute the list $L_m \subseteq B_m$, for every $x_1 \cdots x_{m-1} \in L_{m-1}$ and $u \in \{0, 1\}$ we compute $c(x_1 \cdots x_{m-1} u)$, which is an $\theta/4$-approximation of $p(x_1 \cdots x_{m-1} u)$ with probability at least $1 - \delta$. If $c(x_1 \cdots x_{m-1} u) \geq 3\theta/4$ then we add the bit string $x_1 \cdots x_{m-1} u$ to the list L_m.

Finally, if at some point in the above algorithm one of the lists L_m contains strictly more than $2/\theta$ elements, the algorithm is halted and all subsequent lists $L_{m+1}, \ldots, L_k$ are defined to be empty. With this extra constraint, we ensure that at most $2k/\theta$ probabilities are estimated. It follows that the total runtime of the algorithm is

$$\frac{2k}{\theta} \cdot N_{\text{single}} = \operatorname{poly}(k, \frac{1}{\theta}, \log \frac{1}{\pi}). \tag{4.21}$$

Furthermore, since at most $2k/\theta$ probabilities are estimated, each succeeding with probability $1 - \delta$, the probability that all estimates succeed is at least $(1 - \delta)^{\frac{2k}{\theta}} \geq 1 - \frac{2k}{\theta}\delta = 1 - \pi$.

From this point on we consider the case that all estimates succeed, and claim that in this case the list L_k satisfies (a)-(b). We make the following observations. First, for every m we prove property (a'): *For all $x_1 \cdots x_m \in L_m$ it holds that $p(x_1 \cdots x_m) \geq \theta/2$.* This is true since $c(x_1 \cdots x_m)$ is an $\frac{\theta}{4}$-approximation of $p(x_1 \cdots x_m)$ and since $x_1 \cdots x_m$ was only added to L_m if $c(x_1 \cdots x_m) \geq 3\theta/4$. Property (a') implies that the list L_k satisfies (a). Furthermore, property (a') implies that every list L_m contains at most $2/\theta$ bit strings (since probability distributions are normalized to sum up to 1). This shows that, as long as all estimates of the probabilities are successful, the halting procedure described above need never be applied (indeed, the latter is only incorporated in the algorithm to ensure that successive failed estimations of probabilities do not result in an (exponentially) long runtime).

Second, we argue that each L_m satisfies property (b'): *If $p(x_1 \cdots x_m) \geq \theta$ then $x_1 \cdots x_m \in L_m$.* To see this, we argue by induction on m. For $m = 1$, property (b') follows immediately from the definition of L_1. Furthermore suppose that $y = y_1 \cdots y_m$ satisfies $p(y) \geq \theta$. Then, using Eq. (4.18) we have $p(y_1 \cdots y_{m-1}) \geq \theta$ and thus, by induction, we have $y_1 \cdots y_{m-1} \in L_{m-1}$. The definition of L_m now immediately implies that $y_1 \cdots y_m \in L_m$. This shows that property (b') holds for all L_m, so that L_k satisfies (b) as desired. □

4.6 Algorithm for additively approximable, approximately sparse distributions

We now arrive at an efficient algorithm which, on input of a probability distribution $\mathcal{P}$ which is promised to be approximately sparse *and* which satisfies the conditions of Theorem 30, outputs an (exactly) sparse distribution $\mathcal{P}'$ which is close to $\mathcal{P}$. In addition, the distribution $\mathcal{P}'$ can be sampled efficiently. The proof will be obtained by combining Theorem 30 and Lemma 26. The argument is straightforward but somewhat tedious since some care is required in choosing suitable epsilons and deltas. We also note that Theorem 31 is closely related to theorem 3.11 in [Kushilevitz and Mansour, 1991], which provides a randomized classical algorithm for computing representations of Boolean functions which are promised to be approximately sparse.

Theorem 31. *Let $\mathcal{P}$ be a distribution on B_k which satisfies the following conditions:*

(i) $\mathcal{P}$ is promised to be ε-approximately t-sparse, where $\varepsilon \leq 1/6$.

(ii) $\mathcal{P}$ and its marginals $\mathcal{P}_m$ $(m = 1, \dots, k)$ are additively approximable as in Theorem 30.

Then there exists a randomized classical algorithm with runtime poly$(k, t, \frac{1}{\varepsilon}, \log \frac{1}{\delta})$ which outputs (by means of listing all nonzero probabilities) an s-sparse probability distribution $\mathcal{P}' = \{p'_x\}$ where $s = O(t/\varepsilon)$ such that, with probability at least $1 - \delta$, $\mathcal{P}'$ is $O(\varepsilon)$-close to $\mathcal{P}$ (more precisely $\|\mathcal{P} - \mathcal{P}'\|_1 \leq 12\varepsilon$). Furthermore, $p'_x \geq \varepsilon/8t$ for all p'_x which are nonzero. Finally, it is possible to sample $\mathcal{P}'$ on a classical computer in poly$(k, t, 1/\varepsilon)$ time.

Proof. First we invoke Theorem 30 with $\theta := \varepsilon/t$ and

$$\pi := \tfrac{\delta}{2t/\varepsilon + 1}. \tag{4.22}$$

This yields, with probability at least $1 - \pi$, a list L of k-bit strings satisfying conditions (a)-(b), within a runtime

$$N_1 = \text{poly}(k, \tfrac{1}{\theta}, \log \tfrac{1}{\pi}) = \text{poly}(k, t, \tfrac{1}{\varepsilon}, \log \tfrac{1}{\delta}). \tag{4.23}$$

Note that $|L| \leq 2t/\varepsilon$. Second, since $\mathcal{P}$ is additively approximable, each individual probability p_x with $x \in L$ can be computed with success probability at least $1 - \pi$ and with an error ε' set to

$$\varepsilon' := \min\{\varepsilon/|L|, \varepsilon/4t\} \tag{4.24}$$

in time

$$N_2 = \text{poly}(k, \tfrac{1}{\varepsilon'}, \log \tfrac{1}{\pi}) = \text{poly}(k, t, \tfrac{1}{\varepsilon}, \log \tfrac{1}{\delta}). \tag{4.25}$$

This yields a list of numbers $\{c_x : x \in L\}$ such that $|p_x - c_x| \leq \varepsilon'$ for all $x \in L$ if all evaluations were successful. Up to this point, the runtime of the algorithm is $N = N_1 + |L|N_2$ which scales as poly$(k, t, \frac{1}{\varepsilon}, \log \frac{1}{\delta})$, and the total success probability is at least

$$(1 - \pi)^{|L|+1} \geq 1 - (|L| + 1)\pi \geq 1 - \delta \tag{4.26}$$

where we have used (4.22) and the property $|L| \leq 2t/\varepsilon$. From this point on, the entire algorithm proceeds deterministically.

Define c_x to be 0 for all $x \notin L$ and let $\mathcal{C} = \{c_x : x \in B_k\}$ denote the resulting list of 2^k coefficients. Now let $\mathcal{Q}_{\varepsilon,t} = \{q_x\}$ be the restriction of $\mathcal{P}$ to $B_{\varepsilon,t}$, where $B_{\varepsilon,t}$ is the set of strings satisfying $p_x \geq \varepsilon/t$, as defined in Lemma 26. Note that $B_{\varepsilon,t} \subseteq L$ (recall condition (b) of Theorem 30 and the fact that here $\theta = \varepsilon/t$). Then

$$\begin{aligned} \|\mathcal{C} - \mathcal{P}\|_1 &= \sum_{x \in L} |c_x - p_x| + \sum_{x \notin L} p_x \leq |L| \cdot \varepsilon' + \sum_{x \notin L} p_x \\ &\leq \varepsilon + \sum_{x \notin L} p_x \leq \varepsilon + \sum_{x \notin B_{\varepsilon,t}} p_x = \varepsilon + \|\mathcal{P} - \mathcal{Q}_{\varepsilon,t}\|_1 \leq 3\varepsilon. \end{aligned} \tag{4.27}$$

Here in the first inequality we used that $|c_x - p_x| \leq \varepsilon'$ for all $x \in L$; in the second, we used the definition of ε'; in the third, we used $B_{\varepsilon,t} \subseteq L$; in the equality, we used the definition of $\mathcal{Q}_{\varepsilon,t}$; finally, we used Lemma 26.

Since $|c_x - p_x| \leq \varepsilon' \leq \varepsilon/4t$ (recall the definition of ε') and since $p_x \geq \varepsilon/2t$ owing to condition (a) of Theorem 30, we have $c_x \geq \varepsilon/4t$ for every $x \in L$; in particular, all c_x are nonnegative. Finally, we set $\mathcal{P}'$ to be $\mathcal{C}$ divided by its 1-norm $\|\mathcal{C}\|_1 = \sum |c_x|$, so that $\mathcal{P}'$ is a proper probability distribution. Since $\mathcal{P}'$ is $|L|$-sparse, computing $\mathcal{P}'$ from $\mathcal{C}$ can be done in $O(|L|)$ =poly$(t, 1/\varepsilon)$ time. Putting everything together, the total runtime for computing $\mathcal{P}'$ scales as poly$(k, t, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$. We now show that $\mathcal{P}'$ is also $O(\varepsilon)$-close to $\mathcal{P}$. The argument is straightforward and fully analogous to the one in Section 4.4.2, cf. (4.8)-(4.9). Since $\|\mathcal{C} - \mathcal{P}\|_1 \leq 3\varepsilon$ and $\|\mathcal{P}\|_1 = 1$ we have

$$1 - 3\varepsilon \leq \|\mathcal{C}\|_1 \leq 1 + 3\varepsilon. \tag{4.28}$$

We then find

$$\begin{aligned} \|\mathcal{P}' - \mathcal{P}\|_1 &= \frac{\|\mathcal{C} - \|\mathcal{C}\|_1 \cdot \mathcal{P}\|_1}{\|\mathcal{C}\|_1} \leq \frac{\|\mathcal{C} - \|\mathcal{C}\|_1 \cdot \mathcal{P}\|_1}{1 - 3\varepsilon} \\ &\leq \frac{\|\mathcal{C} - \mathcal{P}\|_1}{1 - 3\varepsilon} + \frac{|1 - \|\mathcal{C}\|_1| \cdot \|\mathcal{P}\|_1}{1 - 3\varepsilon} \leq \frac{6\varepsilon}{1 - 3\varepsilon}. \end{aligned} \tag{4.29}$$

Then, for $\varepsilon \leq 1/6$, we have $\|\mathcal{P}' - \mathcal{P}\|_1 \leq 12\varepsilon$. Note also that $p'_x \geq \varepsilon/8t$ for all $x \in L$ follows by combining the inequalities $c_x \geq \varepsilon/4t$ and $\|\mathcal{C}\| \leq 1 + 3\varepsilon$ and $\varepsilon \leq 1/6$.

Finally, we show how to sample $\mathcal{P}'$. For a bit string $x_1 \cdots x_m$ with m between 1 and k, let $p'(x_1 \cdots x_m)$ denote the marginal probability of $\mathcal{P}'$ for obtaining $x_1 \cdots x_m$ on the first m bits. Since $\mathcal{P}$ is s-sparse with $s = O(t/\varepsilon)$, each $p'(x_1 \cdots x_m)$ can be computed from $\mathcal{P}'$ in poly(s) = poly$(t, 1/\varepsilon)$ time on input of $x_1 \cdots x_m$. By a standard argument, the property that all such marginals can be computed, allows to sample $\mathcal{P}'$ in poly$(k, t, 1/\varepsilon)$ time [Jerrum et al., 1986, Terhal and DiVincenzo, 2004, Valiant, 2002]. □

4.7 Classical simulation of CT states

Here we review two classical simulation results for CT states which will be used in the proofs of our results. An n-qubit unitary operator U is said to be efficiently computable basis-preserving if there exist efficiently computable functions $f, f' : B_n \to B_n$ and $g, g' : B_n \to \mathbb{C}$ where $|g(x)| = 1 = |g'(x)|$ for all $x \in B_n$, such that, for every computational basis state $|x\rangle$, one has

$$U|x\rangle = g(x)|f(x)\rangle \quad \text{and} \quad U^\dagger|x\rangle = g'(x)|f'(x)\rangle \tag{4.30}$$

A notable example of efficiently computable basis preserving operations is given by operators comprising tensor products of Pauli matrices $\mathbb{1}, X, Y, Z$.

Lemma 32 ([Van den Nest, 2011])**.** *Let $|\psi\rangle$ and $|\varphi\rangle$ be CT n-qubit states and let A be an efficiently computable basis-preserving n-qubit operation. Then there exists a randomized classical algorithm with runtime poly$(n, 1/\varepsilon, \log\frac{1}{\delta})$ which outputs an approximation of $\langle\psi| A |\varphi\rangle$ with accuracy ε and success probability at least $1 - \delta$.*

Lemma 33 ([Van den Nest, 2011])**.** *Let $|\psi\rangle$ and $|\varphi\rangle$ be CT n-qubit states, let $|\xi\rangle$ and $|\chi\rangle$ be CT k-qubit states with $k \le n$. Then there exists a randomized classical algorithm with runtime poly$(n, 1/\varepsilon, \log\frac{1}{\delta})$ which outputs an approximation of $\langle\varphi| [|\xi\rangle\langle\chi| \otimes \mathbb{1}] |\psi\rangle$ with accuracy ε and success probability at least $1 - \delta$.*

The above results are slightly more detailed then the corresponding results in [Van den Nest, 2011] since the latter reference does not provide explicit information about the scaling with ε and δ. For completeness, proofs of Lemma 32 and Lemma 33 (which are straightforward extensions of the proofs in [Van den Nest, 2011]) are given in Section 4.10.2.

4.8 Proofs of main results

4.8.1 Proof of Theorem 20

The proof will be obtained by showing that the output distribution of any quantum circuit considered in Theorem 20 satisfies the conditions of Theorem 31. We introduce some further basic definitions. For any positive integer d, let X_d, Z_d be *generalized Pauli operators* (also known as *Weyl operators*) [Gottesman, 1999], which act on the d-level computational basis states $|x\rangle$ (with $x \in \mathbb{Z}_d$) as follows

$$X_d\,|x\rangle = |x + 1\rangle \tag{4.31}$$

$$Z_d\,|x\rangle = e^{\frac{2\pi i}{d}x}\,|x\rangle \tag{4.32}$$

where $x + 1$ is defined modulo d. Note that the order of both X_d is d (i.e. is the smallest integer $r \ge 2$ satisfying $X_d^r = I$ is precisely d), as is the order of Z_d. Let $\mathcal{F}_d$ denote the Fourier transform over $\mathbb{Z}_d$. A straightforward application of definitions [Gottesman, 1999] shows that

$$\mathcal{F}_d^\dagger Z_d \mathcal{F}_d = X_d. \quad \text{and} \quad \mathcal{F}_d Z_d \mathcal{F}_d^\dagger = X_d^\dagger. \tag{4.33}$$

Theorem 20 now follows immediately from Theorem 31 in combination with the following result:

Lemma 34. *Let $\mathcal{P}$ be a probability distribution on B_k arising from a quantum circuit satisfying conditions (a)-(b) in Theorem 20. Let $\mathcal{P}_m$ denote the marginal distributions arising from measurement of the first m qubits, for $m = 1, \ldots, k$ (with $\mathcal{P} \equiv \mathcal{P}_m$). Then each $\mathcal{P}_m$ is additively approximable.*

Proof. Without loss of generality we let S be the set of first k qubits. For a k-bit string $x = (x_1, \ldots, x_k)$, consider the associated k-bit integer $\hat{x} := x_1 2^0 + x_2 2 + \cdots + x_k 2^{k-1}$. The standard basis states of a k-qubit system will be labeled both by the set of k-bit strings x and the associated integers $\hat{x}$ depending on which formulation is most convenient. Below we will use the basic fact that, for any $m = 1, \ldots, k$,

$$\hat{x} \mod 2^m = x_1 2^0 + \cdots + x_m 2^{m-1}. \tag{4.34}$$

Let $m \in \{1, \ldots, k\}$. For an m-bit string $y = y_1 \cdots y_m$, consider the projector (acting on k qubits)

$$|y_1 \cdots y_m\rangle\langle y_1 \cdots y_m| \otimes I \equiv P(y) \tag{4.35}$$

where I denotes the identity on the last $k - m$ qubits. Thus $P(y)$ is the projector onto those k-qubit computational basis states $|x\rangle$ where the first m bits of x coincide with y. Owing to (4.34), this means that $P(y)$ is the projector on those computational $|x\rangle$ satisfying $\hat{x}$ mod $2^m = \hat{y}$, where $\hat{y} := y_1 2^0 + \cdots + y_m 2^{m-1}$. Let $Z_{2^k} \equiv Z$ and $X_{2^k} \equiv X$ denote the generalized Pauli operators acting on $\mathbb{C}^{2^k}$. A straightforward application of the definition of Z shows that

$$\hat{x} \mod 2^m = \hat{y} \quad \text{iff} \quad \alpha^{\hat{y}} Z^{2^{k-m}} |\hat{x}\rangle = |\hat{x}\rangle \quad \text{with } \alpha := e^{-\frac{2\pi i}{2^m}}. \tag{4.36}$$

This implies that $P(y)$ coincides with the projector onto the eigenspace of $M := \alpha^{\hat{y}} Z^{2^{k-m}}$ with eigenvalue 1. This projector can be obtained by averaging over all powers of M; since the order of M is 2^m (recall that the order of Z is 2^k), this implies that

$$P(y) = \frac{1}{2^m} \sum_{u=0}^{2^m - 1} M^u. \tag{4.37}$$

Let $\mathcal{F} \equiv \mathcal{F}_{2^k}$ denote the Fourier transform modulo 2^k. We consider the scenario where $\mathcal{F}$ is applied in the block U_2; the case where $\mathcal{F}^\dagger$ is applied is treated in full analogy and is omitted here. Denoting $N := \alpha^{\hat{y}} X^{2^{k-m}}$ (i.e. we replace Z by $X \equiv X_{2^k}$ in the definition of M) and recalling the first identity of Eq. (4.32) we find

$$\mathcal{F}^\dagger P(y) \mathcal{F} = \frac{1}{2^m} \sum_{u=0}^{2^m - 1} N^u. \tag{4.38}$$

Now denote the n-qubit CT state generated after application of the block U_1 by $|\text{CT}\rangle$. Furthermore denote the marginal probability of obtaining the bit string y when measuring the first m qubits at the end of the circuit by $p(y)$. Then

$$p(y) = \langle \text{CT}| [\mathcal{F}^\dagger P(y) \mathcal{F}] \otimes I |\text{CT}\rangle \tag{4.39}$$

where I denotes the identity acting on the last $n - k$ qubits. Using Lemma 34 we find

$$p(y_1 \cdots y_m) = \frac{1}{2^m} \sum_{u=0}^{2^m - 1} \langle \mathrm{CT} | N^u \otimes I | \mathrm{CT} \rangle. \tag{4.40}$$

It easily follows from the definition of N that each $N^u \otimes I$ is efficiently computable basis-preserving (as defined in section 4.7). Together with Lemma 32 this implies that the function $u \in \mathbb{Z}_{2^m} \to \langle \mathrm{CT} | N^u \otimes I | \mathrm{CT} \rangle$ is additively approximable. But then Lemma 29 implies that $y \to p(y)$ is additively approximable as well. □

4.8.2 Proof of Theorem 21

Similar to the proof of Theorem 20, also the proof of Theorem 21 follows immediately by showing that the output distribution of any quantum circuit considered in Theorem 21 satisfies the conditions of Theorem 31. The latter is done next.

Lemma 35. *Let $\mathcal{P}$ be a probability distribution on B_k arising from a quantum circuit satisfying conditions (a)-(b') in Theorem 21. Let $\mathcal{P}_m$ denote the marginal distributions arising from measurement of the first m qubits, for $m = 1, \ldots, k$ (with $\mathcal{P} \equiv \mathcal{P}_m$). Then each $\mathcal{P}_m$ is additively approximable.*

Proof. We prove the result for qubit systems; the proof will carry over straightforwardly to systems of qudits of potentially different dimensions. Without loss of generality we let S be the set of first k qubits. For an m-bit string $y = y_1 \cdots y_m$ with $m \le k$, let $p(y)$ denote the marginal probability of the outcome $y_1 \cdots y_m$ when measuring the first m qubits at the end of the circuit. We need to show that the function $y \to p(y)$ is additively approximable. Denote the CT state generated after application of the block U_1 by $|\mathrm{CT}\rangle$. Since $U_2 = u_1 \otimes \cdots \otimes u_n$ is a tensor product operator and since $|y\rangle$ is a product state, we have

$$p(y) = \langle \mathrm{CT} | \, U^\dagger [|y\rangle\langle y| \otimes \mathbb{1}] U \, | \mathrm{CT} \rangle = \langle \mathrm{CT} | \, |\alpha\rangle\langle\alpha| \otimes \mathbb{1} \, | \mathrm{CT} \rangle \tag{4.41}$$

for some m-qubit tensor product state $|\alpha\rangle$ (with efficiently computable description). Since product states are CT, Lemma 33 immediately implies that $y \to p(y)$ is additively approximable. □

4.8.3 Proof of Theorem 22 and Theorem 23

Lemma 36. *Let $|CT\rangle$ be an n-qubit CT state, let $U = U_1 \otimes \cdots \otimes U_n$ be a unitary tensor product operator and let $\mathcal{F}$ denote the Fourier transform modulo 2^n. Then the following functions are additively approximable (where $x = x_1 \cdots x_n$ is an n-bit string):*

$$x \to \langle x | \mathcal{F} | CT \rangle \tag{4.42}$$

$$x \to \langle x | \mathcal{F}^\dagger | CT \rangle \tag{4.43}$$

$$x \to \langle x | U | CT \rangle. \tag{4.44}$$

The last function is still additively approximable when generalized to tensor product operators acting on n qudit systems with potentially different dimensions.

Proof. A straightforward application of definitions shows that the states $\mathcal{F}|x\rangle$, $\mathcal{F}^{\dagger}|x\rangle$ and $U|x\rangle$ are CT. The result then immediately follows from Lemma 32 (with A being the identity). □

Lemma 37. *Let c, c' be two complex numbers satisfying $c \neq 0$ and $|c - c'| \leq \alpha$ for some $\alpha > 0$. Let $c = \theta|c|$ where θ is the phase of c and similarly $c' = \theta'|c'|$. Then $|\theta - \theta'| \leq 2\alpha/|c|$.*

Proof. Since $|c - c'| \leq \alpha$, we have $||c| - |c'|| \leq \alpha$. Then

$$|\theta - \theta'||c| = |c - \theta'|c|| \leq |c - c'| + |c' - \theta'|c|| = |c - c'| + ||c'| - |c|| \leq 2\alpha. \tag{4.45}$$

□

Next we prove Theorem 22 and Theorem 23. Let $|\psi_{\text{out}}\rangle$ denote the final state in any of the settings considered in Theorem 22 and Theorem 23. We write $\langle x|\psi_{\text{out}}\rangle = \gamma_x\sqrt{p_x}$ where γ_x is the phase and p_x the modulus squared, so that $\mathcal{P} = \{p_x\}$ is the probability distribution arising from measuring all qubits of $|\psi_{\text{out}}\rangle$ in the computational basis. Since $|\psi_{\text{out}}\rangle$ is $\sqrt{\varepsilon}$-approximately t-sparse, $\mathcal{P}$ is ε-approximately t-sparse by Lemma 25. Recalling Lemma 34 and Lemma 35, we find that all conditions of Theorem 31 are fulfilled. Thus there exists a randomized classical algorithm with runtime $\text{poly}(n, t, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ which outputs an s-sparse probability distribution $\mathcal{P}' = \{p'_x\}$ where $s = O(t/\varepsilon)$ such that, with probability at least $1 - \delta$, $\|\mathcal{P}' - \mathcal{P}\|_1 \leq 12\varepsilon$. Let L be the list of bit strings as in the proof of Theorem 31. Recall from the latter proof also the following properties: $|L| \leq 2t/\varepsilon$; L is precisely the support of $\mathcal{P}'$; $p_x \geq \varepsilon/2t$ for every $x \in L$.

Thus far we have computed an approximation $\mathcal{P}'$ of the probability distribution $\mathcal{P}$. Next we will also approximately compute the amplitudes of $|\psi_{\text{out}}\rangle$ by employing Lemma 36. For every $x \in L$ we compute a complex number a_x satisfying

$$|a_x - \langle x|\psi_{\text{out}}\rangle| \leq \sqrt{\varepsilon^3/8t}. \tag{4.46}$$

Owing to Lemma 36, the function $x \to \langle x|\psi_{\text{out}}\rangle$ is additively approximable. Therefore each individual a_x can be computed with success probability at least $1-\delta/|L|$ in time $N = \text{poly}(n, t, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$. Thus the total runtime for computing all a_x is $|L|T = \text{poly}(n, t, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ and the total success probability is at least $1 - \delta$. We then compute the complex phase θ_x of each a_x (which requires $O(|L|)$ computational steps in total) and define the state

$$|\varphi\rangle := \sum_{x\in L} \theta_x\sqrt{p'_x}|x\rangle. \tag{4.47}$$

Note that $|\varphi\rangle$ has 2-norm equal to 1: indeed $\||\varphi\rangle\|_2^2$ coincides with $\sum_{x\in L} p'_x$ which equals 1 since L coincides with the support of $\mathcal{P}'$. Next we prove that $|\varphi\rangle$ is $O(\sqrt{\varepsilon})$-close to $|\psi_{\text{out}}\rangle$. The idea of the argument is rather straightforward but the details will be somewhat tedious.

First we show that the phase θ_x is close to γ_x for every $x \in L$ (recall that the latter is the phase of $\langle x|\psi_{\text{out}}\rangle$): using Lemma 37 and recalling that $p_x \geq \varepsilon/2t$, we have

$$|\theta_x - \gamma_x| \leq 2 \cdot \sqrt{\frac{\varepsilon^3}{8t}} \cdot \frac{1}{\sqrt{p_x}} \leq \varepsilon. \tag{4.48}$$

This implies that

$$\| \sum_{x\in L} (\theta_x - \gamma_x)\sqrt{p'_x}|x\rangle\|_2^2 = \sum_{x\in L} |\theta_x - \gamma_x|_2^2 p'_x \leq \varepsilon^2 \sum_{x\in L} p'_x \leq \varepsilon^2. \tag{4.49}$$

For every two numbers $a, b \geq 0$ we have $|a-b|^2 \leq |a^2 - b^2|$. This implies that

$$\sum |\sqrt{p'_x} - \sqrt{p_x}|^2 \leq \sum |p'_x - p_x| = \|\mathcal{P}' - \mathcal{P}\|_1 \leq 12\varepsilon \tag{4.50}$$

where the sums are over all $x \in B_n$. Hence

$$\begin{aligned}\||\psi_{\text{out}}\rangle - \sum_{x\in L} \gamma_x \sqrt{p'_x}|x\rangle\|_2^2 &= \sum_{x\in L} |\gamma_x\sqrt{p_x} - \gamma_x\sqrt{p'_x}|^2 + \sum_{x\notin L} p_x \\ &= \sum_{x\in L} |\sqrt{p_x} - \sqrt{p'_x}|^2 + \sum_{x\notin L} p_x \\ &= \sum_{x\in B_n} |\sqrt{p_x} - \sqrt{p_x}'|^2 \leq 12\varepsilon \end{aligned} \tag{4.51}$$

where in the last equality we used that $p'_x = 0$ for all $x \notin L$. Writing

$$|\varphi\rangle = \sum_{x\in L} \gamma_x \sqrt{p'_x}|x\rangle + \sum_{x\in L} (\theta_x - \gamma_x)\sqrt{p'_x}|x\rangle \tag{4.52}$$

and using the triangle inequality, we then find

$$\begin{aligned}\||\psi_{\text{out}}\rangle - |\varphi\rangle\|_2 &\leq \||\psi_{\text{out}}\rangle - \sum_{x\in L} \gamma_x\sqrt{p'_x}|x\rangle\|_2 + \|\sum_{x\in L}(\theta_x - \gamma_x)\sqrt{p'_x}|x\rangle\|_2 \\ &\leq \sqrt{12\varepsilon} + \varepsilon \leq 5\sqrt{\varepsilon}. \end{aligned} \tag{4.53}$$

4.8.4 Proof of Theorem 24

Denote by $\mathcal{P} = \{p_x : x \in B_n\}$ the probability distribution arising from a standard basis measurement of all n qubits performed on the state $\mathcal{F}_{2^n}^\dagger|\psi\rangle$. Then $p_x = |\hat{\psi}_x|^2$. It follows from Lemma 34 that $\mathcal{P}$ and its marginals $\mathcal{P}_m$ fulfill all conditions of Theorem 30. The latter result then immediately implies the existence of a classical algorithm with runtime $\text{poly}(k, \frac{1}{\theta}, \log\frac{1}{\pi})$ which outputs a list $L = \{x^1, \ldots, x^l\}$ where $l \leq 2/\theta$ such that, with probability at least $1 - \pi$, conditions (a) and (b) in Theorem 24 are fulfilled. Furthermore, Lemma 36 implies that, given any $x \in B_n$, there exists a classical algorithm with runtime $\text{poly}(n, 1/\varepsilon, \log\frac{1}{\delta})$ which, with probability at least $1 - \delta$, outputs an ε-approximation of $\hat{\psi}_x$, since $\hat{\psi}_x = \langle x|\mathcal{F}_{2^n}^\dagger|\psi\rangle$.

Fully analogously, for $U = U_1 \otimes \cdots \otimes U_n$ let $\mathcal{P} = \{p_x\}$ be the probability distribution arising from a standard basis measurement of all n qubits performed on the state $U^\dagger|\psi\rangle$. The extension of Theorem 24 to the product basis $\{U|x\rangle\}$ is now obtained by combining Lemma 35, Theorem 30, and Lemma 36.

4.9 Further research

In the classical simulation algorithms given in this paper, we have not optimized the degree or constants involved in the polynomial-time simulation. While our algorithm is a generalization of [Goldreich and Levin, 1989, Kushilevitz and Mansour, 1991], for optimal performance one could try to adapt the more advanced, query-optimal algorithm of [Hassanieh et al., 2012b] to our setting.

4.10 Appendix

4.10.1 Proof of lemma 27

We recall the standard Chernoff-Hoeffding bound for real-valued random variables.

Theorem 38 (Chernoff-Hoeffding bound)**.** *Let $X_1, \dots, X_T$ be i.i.d. real random variables. Assume that $|X_i| \leq 1$ and denote $E := \mathbf{E}X_i$. Then*

$$Prob\left\{\left|\frac{1}{T}\sum_{i=1}^{T} X_i - E\right| \leq \varepsilon\right\} \geq 1 - 2e^{-\frac{T\varepsilon^2}{2}}. \tag{4.54}$$

The proof of the complex-valued version of the Chernoff-Hoeffding bound as given in lemma 27 is an immediate corollary of the real-valued version, as follows. For complex-valued random variables $X_1, \dots, X_T$ we apply Theorem 38 independently to the real and imaginary parts of the X_i, where we choose $\tilde{\varepsilon} = \frac{\varepsilon}{\sqrt{2}}$. Denoting $Y := \frac{1}{T}\sum_{i=1}^{T} X_i - E$, this yields lower bounds for the probabilities that $Re(Y) \leq \tilde{\varepsilon}$ and $Im(Y) \leq \tilde{\varepsilon}$. Putting things together we find

$$\text{Prob}\left\{\left|\frac{1}{T}\sum_{i=1}^{T} X_i - E\right| \leq \varepsilon\right\} \geq 1 - 4e^{-\frac{T\varepsilon^2}{4}}. \tag{4.55}$$

4.10.2 Proofs of lemmas 32 and 33

In this section we give explicit quantitative versions of the definition and theorems about CT states, which were only stated implicitly in [Van den Nest, 2010].

Definition 39 (Computationally Tractable (CT) states)**.** *An n-qubit state $|\psi\rangle$ is called 'computationally tractable' (CT) if the following conditions hold:*

1. *[Sample] it is possible to sample in time $s_{|\psi\rangle} = O(poly(n))$ with classical means from the probability distribution $Prob(x) = |\langle x|\psi\rangle|^2$ on the set of n-bit strings x.*

2. *[Query] upon input of any bit string x, the coefficient $\langle x|\psi\rangle$ can be computed in $c_{|\psi\rangle} = O(poly(n))$ time on a classical computer.*

The proof of lemma 32 will follow immediately from the following result:

Lemma 40. *Let $|\psi\rangle$ and $|\varphi\rangle$ be two CT n-qubit states and let $s = s_{|\psi\rangle} + s_{|\varphi\rangle}$, $c = c_{|\psi\rangle} + c_{|\varphi\rangle}$. Then there exists a randomized classical algorithm to compute μ such that $|\langle\varphi|\psi\rangle - \mu| \leq \varepsilon$ in time $O(\frac{s+c}{\varepsilon^2}\log(\frac{4}{\delta}))$ with error probability δ.*

Proof. Denote $p_x := |\langle x|\psi\rangle|^2$ and $q_x := |\langle x|\varphi\rangle|^2$. Since $|\psi\rangle$ and $|\varphi\rangle$ are CT states, it is possible to sample from the probability distributions $\{p_x\}$ and $\{q_x\}$ in time s (Definition 39, Item 1). Define the function $\alpha : \{0,1\}^n \mapsto \{0,1\}$ by $\alpha(x) = 1$ if $p_x \geq q_x$ and $\alpha(x) = 0$ otherwise, for every n-bit string x, and define the function β by $\beta(x) := 1 - \alpha(x)$. Then α and β can be computed in time $O(c)$ since p_x and q_x can be computed in time c each by Item 2 in Definition 39. The overlap $\langle\varphi|\psi\rangle$ is equal to

$$\langle\varphi|\psi\rangle = \sum\langle\varphi|x\rangle\langle x|\psi\rangle\alpha(x) + \sum\langle\varphi|x\rangle\langle x|\psi\rangle\beta(x) \tag{4.56}$$

where the sums are over all n-bit strings x. Defining the functions F and G by

$$F(x) = \frac{\langle\varphi|x\rangle\langle x|\psi\rangle}{p_x}\alpha(x), \quad G(x) = \frac{\langle\varphi|x\rangle\langle x|\psi\rangle}{q_x}\beta(x) \tag{4.57}$$

we have $\langle\varphi|\psi\rangle = \langle F\rangle + \langle G\rangle$, where $\langle F\rangle = \sum p_x F(x)$ and $\langle G\rangle = \sum p_x G(x)$. It follows from the query property (Definition 39, Item 2) of CT states, that F and G can be evaluated in time $O(c)$. Furthermore, both $|F(x)|$ and $|G(x)|$ are not greater than 1. It thus follows from Lemma 27, that both $\langle F\rangle$ and $\langle G\rangle$ can be approximated with accuracy $\varepsilon/2$ and error probability at most $\delta/2$ by estimating the averages over samples from the distributions p_x and q_x, respectively. More precisely, let X_i, $1 \leq i \leq T$, be samples drawn from distribution $\{p_x\}$ with $T = \frac{16}{\varepsilon^2}\log(\frac{8}{\delta})$, and let $\mu_F = \frac{1}{T}\sum_{i=1}^T F(X_i)$, (and similarly for samples Y_i drawn from $\{q_x\}$, $\mu_G = \frac{1}{T}\sum_{i=1}^T G(Y_i)$), then it follows from Lemma 27 that

$$\Pr\{|\mu_F - \langle F\rangle| \leq \varepsilon/2\} \geq 1 - \delta/2 \tag{4.58}$$

$$\Pr\{|\mu_G - \langle G\rangle| \leq \varepsilon/2\} \geq 1 - \delta/2 \tag{4.59}$$

Thus we conclude that $\langle\varphi|\psi\rangle$ can be approximated by $\mu = \mu_F + \mu_G$ in time $O(\frac{s+c}{\varepsilon^2}\log(\frac{4}{\delta}))$ such that

$$\Pr\{|\mu - \langle\varphi|\psi\rangle| \leq \varepsilon\} \geq 1 - \delta \tag{4.60}$$

□

The proof of lemma 33 is obtained by noting that any partial overlap of n-qubit CT states (as considered in lemma 33) can be re-expressed (via a poly(n) time classical reduction) as a complete overlap $\langle\phi|\phi'\rangle$ where $|\phi\rangle$ and $|\phi'\rangle$ are CT states on $O(n)$ qubits. Invoking lemma 32 then proves the result.

Conclusions and Outlook

In this thesis we have made progress towards understanding the computational power and possibilities provided by a quantum computer. In one direction, we have explicitly designed quantum algorithms that construct non-trivial ground states of certain Local Hamiltonians. While this problem is QMA-complete in general (and thus hard even for a quantum computer), we have identified several conditions that suffice to make the problem tractable and put it into BQP. In this sense, we have explored the border between QMA and BQP and improved our understanding of what allows quantum computers to excel. In another direction, we have explored the border between BQP and BPP with the goal of improving our understanding when and why quantum computers could be classically simulated. Just as in the first direction, we have identified a condition that results in an efficient classical simulation algorithm for a class of non-trivial quantum circuits.

More specifically, towards the first direction (finding useful quantum algorithms) we have shown in Chapter 1 how to construct quantum states described by injective PEPS in polynomial time by first reducing the problem to the generation of a sequence of unique ground states of certain Hamiltonians and then preparing that sequence. In follow-up work, [Hauke et al., 2012] have found our algorithm to be a useful application for small-scale quantum computers, so-called discrete quantum simulators, due to its frugal use of Hilbert space dimensions. Furthermore, [Somma and Boixo, 2013] have used their general spectral gap amplification technique to achieve a quadratic improvement of the run-time of our algorithm in the spectral gap parameter, while [Ozols et al., 2012] have used their quantum rejection sampling technique to achieve a quadratic speed-up of our algorithm in the condition number parameter.

Furthermore, we have shown in Chapter 2 how to generalize our algorithm to topological, degenerate ground states by exploiting the unique structure of G-injective PEPS. The technique (introduced in Chapters 1 and 2) of constructing a complex many-body quantum state by starting from an easily-constructible state and successively transforming it into the desired state, is very general. Although we have applied it here to G-injective PEPS, as a class of states including many important topological quantum states, our algorithm can probably be generalized to other classes of tensor network states, such as string-net models [Schuch, 2012] and models constructed from Hopf algebras [Buerschaper et al., 2013].

Next, in Chapter 3, we have developed a quantum generalization of Moser's algorithm and information theoretic analysis to efficiently construct a zero-energy ground state of certain local Hamiltonians. The existence of such ground states has been established by the non-constructive Quantum Lovász Local Lemma [Ambainis et al., 2012]. Our algorithm requires the additional assumption that the Hamiltonian is a sum of commuting projectors. In fact, for this special case, our algorithm is a *constructive proof* of the Quantum Lovász Local Lemma, as our argument does not depend on the non-constructive result of [Ambainis et al., 2012]. The obvious open question is whether Theorem 11 can be generalized to the non-commuting case. A further open question is whether Moser and Tardos' combinatorial proof [Moser and Tardos, 2010] of the Lovász Local Lemma for the more general, asymmetric case can be generalized to the quantum setting. It is interesting to note that the dissipative algorithm of [Verstraete et al., 2009] is precisely the quantum analogue of Moser and Tardos' algorithm for the general, asymmetric Lovász Local Lemma written in the language of CP-maps. Thus, [Verstraete et al., 2009] already gives a way to prepare the ground state implied by the non-constructive QLLL [Ambainis et al., 2012]. What is still missing is an argument supporting a polynomial-time convergence rate of the given CP-map.

Our second direction concerns the classical simulation of quantum computers. We have found in Chapter 4 that circuits with a structure similar to Shor's algorithm can be classically simulated assuming approximate sparseness of the output distribution generated by the circuit. The implications of our results are twofold. First, they pose restrictions on the design of fast quantum algorithms. For example, our results show that any *exact* quantum algorithm adopting the QFT-Toffoli-QFT^{-1} block structure (or more generally the structures considered in Theorems 20-23) which has as its output state a single computational basis state containing the answer of the problem, can *never* achieve an exponential quantum speed-up. Second, the present results have conceptual implications: the exponential speed-up found in quantum algorithms is often related to the availability of interference of probability amplitudes in this model. Indeed, in several quantum algorithms, first a superposition of states is created using a QFT, then amplitudes are manipulated in some nontrivial way using reversible (classical) gates, such that in a final QFT, by means of interference, only desired basis states survive, whereas the amplitudes for undesired states cancel out. Our results imply that this qualitative picture has to be refined, since too many cancellations leading to only a few classical output states (let alone a single one!) can in fact be simulated efficiently classically, and thus cannot offer an exponential speed-up. Indeed, our results imply that the final probability distribution must *necessarily have super-polynomially large support* (e.g. in the same order as the full state space), in order to allow for exponential speed-up. Finally, since only polynomially many measurements can be performed efficiently on the output state—and thus only a small fraction of the necessarily large number of states can be sampled—the output distribution must have a special structure such that meaningful information can be recovered from just a few measurements. Notably, the coset state produced by Shor's algorithm (and its generalizations) has group structure which is indeed exploited in the classical post-processing step to recover the entire state space from just a few measurements.

Bibliography

M. Aguado and G. Vidal. Entanglement renormalization and topological order. *Phys. Rev. Lett.*, 100:070404, 2008. doi: 10.1103/PhysRevLett.100.070404.

D. Aharonov and L. Eldar. On the complexity of Commuting Local Hamiltonians, and tight conditions for Topological Order in such systems. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 334–343. IEEE, 2011.

D. Aharonov and L. Eldar. The commuting local Hamiltonian on locally-expanding graphs is in NP. *arXiv preprint arXiv:1311.7378*, 2013.

D. Aharonov and T. Naveh. Quantum NP-A Survey, 2002.

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

D. Aharonov, Z. Landau, and J. Makowsky. The quantum FFT can be classically simulated. *arXiv:quant-ph/0611156*, 2006.

A. Akavia. Deterministic Sparse Fourier Approximation via Fooling Arithmetic Progressions. In *Proceedings of the 2010 Conference on Learning Theory, AT Kalai and M. Mohri, eds., Omnipress*, pages 381–393, 2010.

A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, volume 44, pages 146–157, Oct. 2003.

A. Akavia, O. Goldreich, S. Goldwasser, and D. Moshkovitz. On basing one-way functions on NP-hardness. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 701–710. ACM, 2006.

G. Alagic, C. Moore, and A. Russell. Quantum algorithms for Simon's problem over general groups. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1217–1224. Society for Industrial and Applied Mathematics, 2007.

N. Alon and J. H. Spencer. *The probabilistic method*. Wiley-Interscience, 2000.

A. Ambainis, J. Kempe, and O. Sattath. A Quantum Lovasz Local Lemma. *Journal of the ACM (JACM)*, 59(5):24:1–24:24, Nov. 2012. ISSN 0004-5411. doi: 10.1145/2371656.2371659. URL http://doi.acm.org/10.1145/2371656.2371659.

I. Arad and Z. Landau. Quantum Computation and the Evaluation of Tensor Networks. *SIAM Journal on Computing*, 39:3089, 2010.

I. Arad and O. Sattath. A Constructive Quantum Lovász Local Lemma for Commuting Projectors. *arXiv preprint arXiv:1310.7766*, 2013.

V. Arnold. Number-theoretical turbulence in Fermat–Euler arithmetics and large Young diagrams geometry statistics. *Journal of Mathematical Fluid Mechanics*, 7:S4–S50, 2005.

S. Arora and B. Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

A. Aspuru-Guzik and P. Walther. Photonic quantum simulators. *Nature Physics*, 8:285, 2012.

F. Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241, 1982.

J. Beck. An Algorithmic Approach to the Lovász Local Lemma. I. *Random Structures & Algorithms*, 2(4):343–365, 1991.

J. Bermejo-Vega. Classical simulations of non-abelian quantum Fourier transforms. Master's thesis, Technische Universität München, 2011.

J. Bermejo-Vega and M. V. d. Nest. Classical simulations of Abelian-group normalizer circuits with intermediate measurements. *arXiv preprint arXiv:1210.3637*, 2012.

E. Bernstein and U. Vazirani. Quantum complexity theory. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, page 20. ACM, 1993.

D. Berry, G. Ahokas, R. Cleve, and B. Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *Communications in Mathematical Physics*, 270:359–371, 2007. ISSN 0010-3616. URL http://dx.doi.org/10.1007/s00220-006-0150-x. 10.1007/s00220-006-0150-x.

R. Blatt and C. F. Roos. Quantum simulations with trapped ions. *Nature Physics*, 8:277, 2012.

I. Bloch, J. Dalibard, and S. Nascimbène. Quantum simulations with ultracold quantum gases. *Nature Physics*, 8:267, 2012.

S. Boixo. personal communication, 2011.

S. Boixo and R. D. Somma. Necessary condition for the quantum adiabatic approximation. *Physical Review A*, 81(3):032308, 2010.

S. Boixo, E. Knill, and R. D. Somma. Fast quantum algorithms for traversing paths of eigenstates. *Arxiv preprint arXiv:1005.3034*, 2010.

M. Bordewich, M. Freedman, L. Lovász, and D. Welsh. Approximate counting and quantum computation. *Combinatorics Probability and Computing*, 14(5):737–754, 2005.

F. G. S. L. Brandão and M. Horodecki. Exponential Quantum Speed-ups are Generic. *Quantum Information and Computation*, 13:0901–0924, 2013.

S. Bravyi and M. Vyalyi. Commutative version of the local Hamiltonian problem and common eigenspace problem. *Quantum Information & Computation*, 5(3):187–215, 2005.

M. J. Bremner, R. Jozsa, and D. J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 467(2126):459–472, 2011.

D. E. Browne. Efficient classical simulation of the quantum fourier transform. *New Journal of Physics*, 9(5):146, 2007.

O. Buerschaper, J. M. Mombelli, M. Christandl, and M. Aguado. A hierarchy of topological tensor network states. *Journal of Mathematical Physics*, 54:012201, 2013.

Clay Mathematics Institute. Millennium Problems: P vs NP Problem. `http://www.claymath.org/millenium-problems/p-vs-np-problem`, 2000.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, second edition, 2006.

T. S. Cubitt and M. Schwarz. A constructive commutative quantum Lovász Local Lemma, and beyond. *arXiv preprint arXiv:1112.1413*, 2011.

P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets*, II:609–627, 1975.

R. P. Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6):467–488, 1982.

A. Gilbert, S. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal sparse fourier representations. In *Proceedings of SPIE*, volume 5914, page 59141A, 2005.

A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal sparse fourier representations via sampling. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 152–161. ACM, 2002.

O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32. ACM, 1989.

D. Gottesman. Fault-tolerant quantum computation with higher-dimensional systems. In *Quantum Computing and Quantum Communications*, pages 302–313. Springer, 1999.

L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, page 219. ACM, 1996.

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

H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse fourier transform. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1183–1194. SIAM, 2012a.

H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse fourier transform. In *Proceedings of the 44th symposium on Theory of Computing*, pages 563–578. ACM, 2012b.

M. B. Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007:P08024, 2007.

M. B. Hastings. Trivial low energy states for commuting Hamiltonians, and the quantum PCP conjecture. *Quantum Information & Computation*, 13(5-6):393–429, 2013.

P. Hauke, F. M. Cucchietti, L. Tagliacozzo, I. Deutsch, and M. Lewenstein. Can one trust quantum simulators? *Reports on Progress in Physics*, 75(8):082401, 2012.

A. A. Houck, H. E. Türeci, and J. Koch. On-chip quantum simulation with superconducting circuits. *Nature Physics*, 8:292, 2012.

R. Impagliazzo and A. Wigderson. P = BPP if E Requires Exponential Circuits: Derandomizing the XOR Lemma. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 220–229, New York, NY, USA, 1997. ACM. ISBN 0-89791-888-6. doi: 10.1145/258533.258590. URL http://doi.acm.org/10.1145/258533.258590.

M. A. Iwen. Combinatorial sublinear-time Fourier algorithms. *Foundations of Computational Mathematics*, 10(3):303–338, 2010.

M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

C. Jordan. Essai sur la géométrie à n dimensions. *Bull. Soc. Math. France*, 3:103–174, 1875.

S. Jordan. Quantum Algorithm Zoo. *Quantum algorithm list available at http://math.nist.gov/quantum/zoo*, 2013.

A. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2 – 30, 2003. ISSN 0003-4916. doi: 10.1016/S0003-4916(02)00018-0. URL http://www.sciencedirect.com/science/article/pii/S0003491602000180.

A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classsical and quantum computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Soc., 2002.

E. Knill, G. Ortiz, and R. D. Somma. Optimal quantum measurements of expectation values of observables. *Physical Review A*, 75:012328, 2007. doi: 10.1103/PhysRevA.75.012328.

P. Kurlberg and C. Pomerance. On a problem of Arnold: The average multiplicative order of a given integer. *Algebra & Number Theory*, 7(4):981–999, 2013.

E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 455–464. ACM, 1991.

C. Lomont. The hidden subgroup problem-review and open problems. *arXiv preprint quant-ph/0411037*, 2004.

Y. Mansour. Randomized interpolation and approximation of sparse polynomials. *SIAM Journal on Computing*, 24(2):357–368, 1995.

C. Marriott and J. Watrous. Quantum Arthur–Merlin games. *Computational Complexity*, 14 (2):122–152, 2005. ISSN 1016-3328.

S. Michalakis. Stability of the Area Law for the Entropy of Entanglement. *arXiv preprint arXiv:1206.6900*, 2012.

A. Montanaro and T. J. Osborne. Quantum boolean functions. *Chicago Journal OF Theoretical Computer Science*, 1:1–45, 2010.

C. Moore, D. Rockmore, and A. Russell. Generic quantum fourier transforms. *ACM Transactions on Algorithms (TALG)*, 2(4):707–723, 2006.

R. A. Moser. A constructive proof of the Lovász Local Lemma. In *Proceedings of the 41st annual ACM Symposium on Theory Of Computing (STOC)*, pages 343–350. ACM, 2009.

R. A. Moser. *Exact Algorithms for Constraint Satisfaction Problems*. PhD thesis, ETH Zürich, 2012.

R. A. Moser and G. Tardos. A constructive proof of the general Lovász Local Lemma. *Journal of the ACM (JACM)*, 57(2):1–15, 2010.

D. Nagaj, P. Wocjan, and Y. Zhang. Fast Amplification of QMA. *Quantum Information and Computation (QIC)*, 9(11&12):1053–1068, 2009.

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

T. J. Osborne. Hamiltonian complexity. *Reports on Progress in Physics*, 75(2):22001–22010, 2012.

M. Ozols, M. Roetteler, and J. Roland. Quantum rejection sampling. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 290–308, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1115-1. doi: 10.1145/2090236.2090261. URL http://doi.acm.org/10.1145/2090236.2090261.

D. Perez-Garcia, F. Verstraete, J. I. Cirac, and M. M. Wolf. PEPS as unique ground states of local Hamiltonians. *Quant. Inf. Comp*, 8:0650–0663, July 2008.

D. Poilblanc, N. Schuch, D. Peréz-García, and J. I. Cirac. Topological and entanglement properties of resonating valence bond wave functions. *Physical Review B*, 86:014404, 2012. doi: 10.1103/PhysRevB.86.014404.

S. Rommer and S. Östlund. Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Physical Review B*, 55(4): 2164, 1997.

C. Schön, E. Solano, F. Verstraete, J. I. Cirac, and M. M. Wolf. Sequential generation of entangled multiqubit states. *Physical review letters*, 95(11):110503, 2005. ISSN 1079-7114.

N. Schuch. Complexity of commuting Hamiltonians on a square lattice of qubits. *Quantum Information & Computation*, 11(11-12):901–912, 2011.

N. Schuch. (private communication), 2012.

N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac. Computational complexity of projected entangled pair states. *Physical review letters*, 98(14):140506, 2007. ISSN 1079-7114.

N. Schuch, I. Cirac, and D. Perez-Garcia. PEPS as ground states: Degeneracy and topology. *Annals of Physics*, 325:2153–2192, 2010. doi: 10.1016/j.aop.2010.05.008.

N. Schuch, D. Perez-Garcia, and I. Cirac. Classifying quantum phases using matrix product states and projected entangled pair states. *Phys. Rev. B*, 84:165139, 2011.

M. Schwarz, K. Temme, and F. Verstraete. Preparing projected entangled pair states on a quantum computer. *Phys. Rev. Lett.*, 108:110502, 2012. doi: 10.1103/PhysRevLett.108.110502.

M. Schwarz, K. Temme, F. Verstraete, D. Perez-Garcia, and T. S. Cubitt. Preparing topological projected entangled pair states on a quantum computer. *Physical Review A*, 88(032321), 2013.

D. Shepherd and M. J. Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 465(2105):1413–1439, 2009.

P. Shor. Lower bounds on the period in integer factorization? http://cstheory.stackexchange.com/questions/7043/lower-bounds-on-the-period-in-integer-factorization, 2011.

P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

D. R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5): 1474–1483, 1997.

R. D. Somma and S. Boixo. Spectral gap amplification. *SIAM Journal on Computing*, 42(2): 593–610, 2013.

J. Spencer. Asymptotic lower bounds for Ramsey functions. *Discrete Mathematics*, 20(0): 69 – 76, 1977. ISSN 0012-365X. doi: 10.1016/0012-365X(77)90044-9. URL http://www.sciencedirect.com/science/article/pii/0012365X77900449.

D. Stahlke. Quantum interference as a resource for quantum speedup. *arXiv preprint arXiv:1305.2186*, 2013.

K. Temme, T. J. Osborne, K. G. Vollbrecht, D. Poulin, and F. Verstraete. Quantum metropolis sampling. *Nature*, 471(7336):87–90, 2011.

B. M. Terhal and D. P. DiVincenzo. Adptive quantum computation, constant depth quantum circuits and arthur-merlin games. *Quantum Information & Computation*, 4(2):134–145, 2004.

L. G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM Journal on Computing*, 31(4):1229–1254, 2002.

M. Van den Nest. Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond. *Quantum Information and Computation*, 10(3-4):0258–0271, 2010.

M. Van den Nest. Simulating quantum computers with probabilistic methods. *Quantum Information and Computation*, 11(9-10):784–812, 2011.

M. Van den Nest. Efficient classical simulations of quantum fourier transforms and normalizer circuits over abelian groups. *arXiv preprint arXiv:1201.4867*, 2012.

F. Verstraete and J. I. Cirac. Valence-bond states for quantum computation. *Phys. Rev. A*, 70: 060302, Dec 2004. ISSN 1094-1622. doi: 10.1103/PhysRevA.70.060302. URL http://link.aps.org/doi/10.1103/PhysRevA.70.060302.

F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Physical Review B*, 73(9):94423, 2006. ISSN 1550-235X.

F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac. Criticality, the area law, and the computational power of projected entangled pair states. *Physical review letters*, 96(22): 220601, 2006. ISSN 1079-7114.

F. Verstraete, M. M. Wolf, and J. I. Cirac. Quantum computation, quantum state engineering, and quantum phase transitions driven by dissipation. *Nature Physics*, 5(9):633–636, 2009.

J. Watrous. Quantum computational complexity. *Encyclopedia of Complexity and Systems Science. Springer*, 2008.

M. M. Wilde. *Quantum Information Theory*. Cambridge University Press, 2013.

A. Winter. *Coding Theorems of Quantum Information Theory*. PhD thesis, University of Bielefeld, 1999.

N. Yoran and A. J. Short. Efficient classical simulation of the approximate quantum fourier transform. *Phys. Rev. A*, 76:042321, Oct 2007. doi: 10.1103/PhysRevA.76.042321. URL http://link.aps.org/doi/10.1103/PhysRevA.76.042321.

Lebenslauf

Persönliche Daten

Name:	Dipl.-Ing. Martin Schwarz
Geburtsdatum:	15. Februar 1979
Geburtsort:	Wiener Neustadt, Österreich
Nationalität:	Österreich

Ausbildung und Tätigkeiten

05 / 1997	Reifeprüfung am Bundesrealgymnasium Wiener Neustadt
10 / 1997 – 06 / 1998	verpflichtender Grundwehrdienst beim Österreichischen Bundesheer
10 / 1998 – 01 / 2003	Studium der Technischen Informatik an der Technischen Universität Wien
07 / 1999 – 03 / 2003	Software Engineer bei TTTech Computertechnik AG, Wien
03 / 2003 – 09 / 2004	On-site Engineer bei Honeywell Aerospace Electronic Systems, Kansas City
05 / 2004 – 12 / 2005	Project Manager Aerospace bei TTTech AG, Wien
01 / 2006 – 04 / 2007	Team Lead, Aerospace Embedded Software bei TTTech AG, Wien
04 / 2007 – 03 / 2013	Advanced Technology / Product Manager bei TTTech AG, Wien
10 / 2009 – 12 / 2013	Doktorand an der Fakultät für Physik der Universität Wien
04 / 2013 – heute	Team Lead, Chip IP Product Management bei TTTech AG, Wien

Publikationsliste

- M. Schwarz, K. Temme, and F. Verstraete,
  "Preparing Projected Entangled-Pair States on a Quantum Computer", *Phys. Rev. Lett.* **108**, 110502 (2012)

- M. Schwarz, K. Temme, F. Verstraete, D. Pérez-García, T. S. Cubitt,
  "Preparing Topological Projected Entangled-Pair States on a Quantum Computer", *Phys. Rev. A* **88**, 032321 (2013)

- M. Schwarz, M. Van den Nest,
  "Simulating Quantum Circuits with Sparse Output Distributions", *e-print* arXiv:1310.6749, 2013, submitted

- M. Schwarz, T.S. Cubitt and F. Verstraete,
  "An Information-Theoretic Proof of the Constructive Commutative Quantum Lovász Local Lemma", *e-print* arXiv:1311.6474, 2013, submitted