THEORY OF COMPUTING LIBRARY
GRADUATE SURVEYS 2 (2011), pp. 1–54
www.theoryofcomputing.org

Quantum Proofs for Classical Theorems

Andrew Drucker* Ronald de Wolf†

Received: October 18, 2009; published: March 9, 2011.

Abstract: Alongside the development of quantum algorithms and quantum complexity theory in recent years, quantum techniques have also proved instrumental in obtaining results in diverse classical (non-quantum) areas, such as coding theory, communication complexity, and polynomial approximations. In this paper we survey these results and the quantum toolbox they use.

ACM Classification: F.1.2

AMS Classification: 81P68

Key words and phrases: quantum arguments, quantum computing, quantum information, polynomial approximation

*Supported by a DARPA YFA grant.

†Partially supported by a Vidi grant from the Netherlands Organization for Scientific Research (NWO), and by the European Commission under the projects Qubit Applications (QAP, funded by the IST directorate as Contract Number 015848) and Quantum Computer Science (QCS).

 DOI: 10.4086/toc.gs.2011.002

Contents

1 Introduction

1.1 Surprising proof methods

Mathematics is full of surprising proofs, and these form a large part of the beauty and fascination of the subject to its practitioners. A feature of many such proofs is that they introduce objects or concepts from beyond the "milieu" in which the problem was originally posed.

As an example from high-school math, the easiest way to prove real-valued identities like

$$\cos(x+y) = \cos x \cos y - \sin x \sin y$$

is to go to *complex* numbers: using the identity $e^{ix} = \cos x + i \sin x$ we have

$$e^{i(x+y)} = e^{ix}e^{iy} = (\cos x + i \sin x)(\cos y + i \sin y) = \cos x \cos y - \sin x \sin y + i(\cos x \sin y + \sin x \cos y) .$$

Taking the real parts of the two sides gives our identity.

Another example is the *probabilistic method*, associated with Paul Erdős and excellently covered in the book of Alon and Spencer [13]. The idea here is to prove the existence of an object with a specific desirable property P by choosing such an object at random, and showing that it satisfies P with positive probability. Here is a simple example: suppose we want to prove that every undirected graph $G = (V, E)$ with $|E| = m$ edges has a cut (a partition $V = V_1 \cup V_2$ of its vertex set) with at least $m/2$ edges crossing the cut.

> *Proof.* Choose the cut at random, by including each vertex i in V_1 with probability 1/2 (independently of the other vertices). For each fixed edge (i, j), the probability that it crosses is the probability that i and j end up in different sets, which is exactly 1/2. Hence by linearity of expectation, the *expected* number of crossing edges for our cut is exactly $m/2$. But then there must exist a specific cut with at least $m/2$ crossing edges. $\square$

The statement of the theorem has nothing to do with probability, yet probabilistic methods allow us to give a very simple proof. Alon and Spencer [13] give many other examples of this phenomenon, in areas ranging from graph theory and analysis to combinatorics and computer science.

Two special cases of the probabilistic method deserve mention here. First, one can combine the language of probability with that of information theory [40]. For instance, if a random variable X is uniformly distributed over some finite set S then its Shannon entropy $H(X) = -\sum_x \Pr[X = x] \log \Pr[X = x]$ is exactly $\log |S|$. Hence upper (resp. lower) bounds on this entropy give upper (resp. lower) bounds on the size of S. Information theory offers many tools that allow us to manipulate and bound entropies in sophisticated yet intuitive ways. The following example is due to Peter Frankl. In theoretical computer science one often has to bound sums of binomials coefficients like

$$s = \sum_{i=0}^{\alpha n} \binom{n}{i} ,$$

say for some $\alpha \leq 1/2$. This s is exactly the size of the set $S \subseteq \{0,1\}^n$ of strings of Hamming weight at most αn. Choose $X = (X_1, \ldots, X_n)$ uniformly at random from S. Then, individually, each X_i is a bit whose probability of being 1 is at most α, and hence $H(X_i) \leq H(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$. Using the sub-additivity of entropy we obtain an essentially tight upper bound on the size of S:

$$\log s = \log|S| = H(X) \leq \sum_{i=1}^{n} H(X_i) \leq nH(\alpha)\,.$$

A second, related but more algorithmic approach is the so-called "incompressibility method," which reasons about the properties of randomly chosen objects and is based on the theory of Kolmogorov complexity [88, Chapter 6]. In this method we consider "compression schemes," that is, injective mappings C from binary strings to other binary strings. The basic observation is that for any C and n, most strings of length n map to strings of length nearly n or more, simply because there aren't enough short descriptions to go round. Thus, if we can design some compression scheme that represents n-bit objects that do *not* have some desirable property P with much fewer than n bits, it follows that most n-bit strings have property P.

Of course one can argue that applications of the probabilistic method are all just counting arguments disguised in the language of probability, and hence probabilistic arguments are not essential to the proof. In a narrow sense this is indeed correct. However, viewing things probabilistically gives a rather different perspective and allows us to use sophisticated tools to bound probabilities, such as large deviation inequalities and the Lovász Local Lemma, as exemplified in [13]. While such tools may be viewed as elaborate ways of doing a counting argument, the point is that we might never think of using them if the argument were phrased in terms of counting instead of probability. Similarly, arguments based on information theory or incompressibility are essentially "just" counting arguments, but the information-theoretic and algorithmic perspective leads to proofs we would not easily discover otherwise.

1.2 A quantum method?

The purpose of this paper is to survey another family of surprising proofs that use the language and techniques of *quantum computing* to prove theorems whose statement has nothing to do with quantum computing.

Since the mid-1990s, especially since Peter Shor's 1994 quantum algorithm for factoring large integers [121], quantum computing has grown to become a prominent and promising area at the intersection of computer science and physics. Quantum computers could yield fundamental improvements in algorithms, communication protocols, and cryptography. This promise, however, depends on physical realization, and despite the best efforts of experimental physicists we are still very far from building large-scale quantum computers.

In contrast, using the language and tools of quantum computing as a proof tool is something we can do today. Here, quantum mechanics is purely a mathematical framework, and our proofs remain valid even if large-scale quantum computers are never built (or worse, if quantum mechanics turns out to be wrong as a description of reality). This paper describes a number of recent results of this type. As with the probabilistic method, these applications range over many areas, from error-correcting codes and complexity theory to purely mathematical questions about polynomial approximations and matrix theory. We hesitate to say that they represent a "quantum method," since the set of tools is far less developed than the probabilistic method. However, we feel that these quantum tools will yield more surprises in the future, and have the potential to grow into a full-fledged proof method.

As we will see below, the language of quantum computing is really just a shorthand for linear algebra: states are vectors and operations are matrices. Accordingly, one could argue that we don't need the quantum language at all. Indeed, one can always translate the proofs given below back to the language of linear algebra. What's more, there is already an extensive tradition of elegant proofs in combinatorics, geometry, and other areas, which employ linear algebra (often over finite fields) in surprising ways. For two surveys of this *linear algebra method*, see the books by Babai and Frank [18] and Jukna [70, Part III]. However, we feel the proofs we survey here are of a different nature than those produced by the classical linear algebra method. Just as thinking probabilistically suggests strategies that might not occur when taking the counting perspective, couching a problem in the language of quantum algorithms and quantum information gives us access to intuitions and tools that we would otherwise likely overlook or consider unnatural. While certainly not a cure-all, for some types of problems the quantum perspective is a very useful one and there is no reason to restrict oneself to the language of linear algebra.

1.3 Outline

The survey is organized as follows. We begin in Section 2 with a succinct introduction to the quantum model and the properties used in our applications. Most of those applications can be conveniently classified in two broad categories. First, there are applications that are close in spirit to the classical information-theory method. They use quantum information theory to bound the dimension of a quantum system, analogously to how classical information theory can be used to bound the size of a set. In Section 3 we give three results of this type. Other applications use quantum algorithms as a tool to define *polynomials* with desired properties. In Section 4 we give a number of applications of this type. Finally, there are a number of applications of quantum tools that do not fit well in the previous two categories; some of these are classical results more indirectly "inspired" by earlier quantum results. These are described in Section 5.

2 The quantum toolbox

The goal of this survey is to show how quantum techniques can be used to analyze non-quantum questions. Of course, this requires at least *some* knowledge of quantum mechanics, which might appear discouraging to those without a physics background. However, the amount of quantum mechanics one needs is surprisingly small and easily explained in terms of basic linear algebra. The first thing we would like to convey is that at the basic level, quantum mechanics is not a full-fledged theory of the universe (containing claims about which objects and forces "really exist"), but rather a *framework* in which to describe physical systems and processes they undergo. Within this framework we can posit the existence of basic units of quantum information ("qubits") and ways of transforming them, just as classical theoretical computer science begins by positing the existence of bits and the ability to perform basic logical operations on them. While we hope this is reassuring, it is nevertheless true that the quantum-mechanical framework has strange and novel aspects—which, of course, is what makes it worth studying in the first place.

In this section we give a bare-bones introduction to the essentials of quantum mechanics and quantum computing. (A more general framework for quantum mechanics is given in the Appendix, but we will not need it for the results we describe.) We then give some specific useful results from quantum information theory and quantum algorithms.

2.1 The quantum model

At a very general level level, any physical system is associated with a *Hilbert space*, and a state of that system is described by an element of that Hilbert space. The Hilbert space corresponding to the combination of two physical systems is the *tensor* product of their respective Hilbert spaces.

Pure states For our purposes, a *pure* quantum state (often just called a *state*) is a unit column vector in a d-dimensional complex vector space $\mathbb{C}^d$. Quantum physics typically used the *Dirac notation*, writing a column vector v as $|v\rangle$, while $\langle v|$ denotes the row vector that is the conjugate transpose of v.

The simplest nontrivial example is the case of a 2-dimensional system, called a *qubit*. We identify the two possible values of a classical bit with the two vectors in the standard orthonormal basis for this space:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

In general, the state of a qubit can be a *superposition* (i. e., linear combination) of these two values:

$$|\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix},$$

where the complex numbers are called *amplitudes*; α_0 is the amplitude of basis state $|0\rangle$, and α_1 is the amplitude of $|1\rangle$. Since a state is a unit vector, we have $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

A 2-qubit space is obtained by taking the *tensor product* of two 1-qubit spaces. This is most easily explained by giving the four basis vectors of the tensor space:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \qquad |01\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

These correspond to the four possible 2-bit strings. More generally, we can form 2^n-dimensional spaces this way whose basis states correspond to the 2^n different n-bit strings.

We also sometimes use d-dimensional spaces without such a qubit-structure. Here we usually denote the d standard orthonormal basis vectors with $|1\rangle, \ldots, |d\rangle$, where $|i\rangle_i = 1$ and $|i\rangle_j = 0$ for all $j \neq i$. For a vector $|\phi\rangle = \sum_{i=1}^{d} \alpha_i |i\rangle$ in this space, $\langle\phi| = \sum_{i=1}^{d} \alpha_i^* \langle i|$ is the row vector that is the conjugate transpose of $|\phi\rangle$. The Dirac notation allows us for instance to conveniently write the standard inner product between states $|\phi\rangle$ and $|\psi\rangle$ as $\langle\phi| \cdot |\psi\rangle = \langle\phi|\psi\rangle$. This inner product induces the Euclidean norm (or "length") of vectors: $\|v\| = \sqrt{\langle v|v\rangle}$.

One can also take tensor products in this space: if $|\phi\rangle = \sum_{i\in[m]} \alpha_i |i\rangle$ and $|\psi\rangle = \sum_{j\in[n]} \beta_j |j\rangle$, then their tensor product $|\phi\rangle \otimes |\psi\rangle \in \mathbb{C}^{mn}$ is

$$|\phi\rangle \otimes |\psi\rangle = \sum_{i\in[m],j\in[n]} \alpha_i \beta_j |i,j\rangle\,,$$

where $[n]$ denotes the set $\{1,\ldots,n\}$ and the vectors $|i,j\rangle = |i\rangle \otimes |j\rangle$ form an orthonormal basis for $\mathbb{C}^{mn}$. This tensor product of states $|\phi\rangle$ and $|\psi\rangle$ is also often denoted simply as $|\phi\rangle|\psi\rangle$. Note that this new state is a unit vector, as it should be.

Not every pure state in $\mathbb{C}^{mn}$ can be expressed as a tensor product in this way; those that cannot are called *entangled*. The best-known entangled state is the 2-qubit *EPR-pair* $(1/\sqrt{2})(|00\rangle + |11\rangle)$, named after the authors of the paper [49]. When two separated parties each hold part of such an entangled state, we talk about *shared* entanglement between them.

Transformations There are two things one can do with a quantum state: transform it or measure it. Actually, as we will see, measurements can transform the measured states as well; however, we reserve the word "transformation" to describe non-measurement change processes, which we describe next. Quantum mechanics allows only *linear* transformations on states. Since these linear transformations must map unit vectors to unit vectors, we require them to be *norm-preserving* (equivalently, *inner-product-preserving*). Norm-preserving linear maps are called *unitary*. Equivalently, these are the $d\times d$ matrices U whose conjugate transpose U^* equals the inverse U^{-1} (physicists typically write $U^\dagger$ instead of U^*). For our purposes, unitary transformations are exactly the transformations that quantum mechanics allows us to apply to states. We will frequently define transformations by giving their action on the standard basis, with the understanding that such a definition extends (uniquely) to a linear map on the entire space.

Possibly the most important 1-qubit unitary is the *Hadamard transform*:

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.1}$$

This maps basis state $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|1\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Two other types of unitaries deserve special mention. First, for any function $f:\{0,1\}^n \to \{0,1\}^n$, define a transformation U_f mapping the joint computational basis state $|x\rangle|y\rangle$ (where $x,y\in\{0,1\}^n$) to $|x\rangle|y\oplus f(x)\rangle$, where "$\oplus$" denotes bitwise addition $(\bmod 2)$ of n-bit vectors. Note that U_f is a permutation on the orthonormal basis states, and therefore unitary. With such transformations we can simulate classical computations. Next, fix a unitary transformation U on a k-qubit system, and consider the $(k+1)$-qubit unitary transformation V defined by

$$V(|0\rangle|\psi\rangle) = |0\rangle|\psi\rangle, \quad V(|1\rangle|\psi\rangle) = |1\rangle U|\psi\rangle\,. \tag{2.2}$$

This V is called a *controlled-U* operation, and the first qubit is called the *control qubit*. Intuitively, our quantum computer uses the first qubit to "decide" whether or not to apply U to the last k qubits.

Finally, just as one can take the tensor product of quantum states in different registers, one can take the tensor product of quantum operations (more generally, of matrices) acting on two registers. If $A = (a_{ij})$

is an $m \times m'$ matrix and B is an $n \times n'$ matrix, then their tensor product is the $mn \times m'n'$ matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1m'}B \\ a_{21}B & \cdots & a_{2m'}B \\ & \ddots & \\ a_{m1}B & \cdots & a_{mm'}B \end{pmatrix}.$$

Note that the tensor product of two vectors is the special case where $m' = n' = 1$, and that $A \otimes B$ is unitary if A and B are. We may regard $A \otimes B$ as the simultaneous application of A to the first register and B to the second register. For example, the n-fold tensor product $H^{\otimes n}$ denotes the unitary that applies the one-qubit Hadamard gate to each qubit of an n-qubit register. This maps any basis state $|x\rangle$ to

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$$

(and vice versa, since H happens to be its own inverse). Here $x \cdot y = \sum_{i=1}^n x_i y_i$ denotes the inner product of bit strings.

Measurement Quantum mechanics is distinctive for having *measurement* built-in as a fundamental notion, at least in most formulations. A measurement is a way to obtain information about the measured quantum system. It takes as input a quantum state and outputs classical data (the "measurement outcome"), along with a new quantum state. It is an inherently probabilistic process that affects the state being measured. Various types of measurements on systems are possible. In the simplest kind, known as *measurement in the computational basis*, we measure a pure state

$$|\phi\rangle = \sum_{i=1}^{d} \alpha_i |i\rangle$$

and see the basis state $|i\rangle$ with probability p_i equal to the squared amplitude $|\alpha_i|^2$ (or more accurately, the squared *modulus* of the amplitude—it is often convenient to just call this the squared amplitude). Since the state is a unit vector these outcome probabilities sum to 1, as they should. After the measurement, the state has changed to the observed basis state $|i\rangle$. Note that if we apply the measurement now a second time, we will observe the same $|i\rangle$ with certainty—as if the first measurement forced the quantum state to "make up its mind."

A more general type of measurement is the *projective* measurement, also known as *Von Neumann measurement*. A projective measurement with k outcomes is specified by $d \times d$ *projector matrices* $P_1, \ldots, P_k$ that form an orthogonal decomposition of the d-dimensional space. That is, $P_i P_j = \delta_{i,j} P_i$, and $\sum_{i=1}^k P_i = I$ is the identity operator on the whole space. Equivalently, there exist orthonormal vectors $v_1, \ldots, v_d$ and a partition $S_1 \cup \cdots \cup S_k$ of $\{1, \ldots, d\}$ such that $P_i = \sum_{j \in S_i} |v_j\rangle\langle v_j|$ for all $i \in [k]$. With some abuse of notation we can identity P_i with the subspace onto which it projects, and write the orthogonal decomposition of the complete space as

$$\mathbb{C}^d = P_1 \oplus P_2 \oplus \cdots \oplus P_k.$$

Correspondingly, we can write $|\phi\rangle$ as the sum of its components in the k subspaces:

$$|\phi\rangle = P_1|\phi\rangle + P_2|\phi\rangle + \cdots + P_k|\phi\rangle \, .$$

A measurement probabilistically picks out one of these components: the probability of outcome i is $\|P_i|\phi\rangle\|^2$, and if we got outcome i then the state changes to the new unit vector $P_i|\phi\rangle/\|P_i|\phi\rangle\|$ (which is the component of $|\phi\rangle$ in the i-th subspace, renormalized).

An important special case of projective measurements is *measurement relative to the orthonormal basis* $\{|v_i\rangle\}$, where each projector P_i projects onto a 1-dimensional subspace spanned by the unit vector $|v_i\rangle$. In this case we have $k = d$ and $P_i = |v_i\rangle\langle v_i|$. A measurement in the computational basis corresponds to the case where $P_i = |i\rangle\langle i|$. If $|\phi\rangle = \sum_i \alpha_i|i\rangle$ then we indeed recover the squared amplitude: $p_i = \|P_i|\phi\rangle\|^2 = |\alpha_i|^2$.

One can also apply a measurement to part of a state, for instance to the first register of a 2-register quantum system. Formally, we just specify a k-outcome projective measurement for the first register, and then tensor each of the k projectors with the identity operator on the second register to obtain a k-outcome measurement on the joint space.

Looking back at our definitions, we observe that if two quantum states $|\phi\rangle, |\psi\rangle$ satisfy $\alpha|\phi\rangle = |\psi\rangle$ for some scalar α (necessarily of unit norm), then for any system of projectors $\{P_i\}$, $\|P_i|\phi\rangle\|^2 = \|P_i|\psi\rangle\|^2$ and so measuring $|\phi\rangle$ with $\{P_i\}$ yields the same distribution as measuring $|\psi\rangle$. More is true: if we make any sequence of transformations and measurements to the two states, the sequence of measurement outcomes we see are identically distributed. Thus the two states are indistinguishable, and we generally regard them as *the same state*.

Quantum-classical analogy For the uninitiated, these high-dimensional complex vectors and unitary transformations may seem baffling. One helpful point of view is the analogy with classical random processes. In the classical world, the evolution of a probabilistic automaton whose state consists of n bits can be modeled as a sequence of 2^n-dimensional vectors $\pi^1, \pi^2, \ldots$. Each π^i is a probability distribution on $\{0,1\}^n$, where π^t_x gives the probability that the automaton is in state x if measured at time t (π^1 is the starting state). The evolution from time t to $t+1$ is describable by a matrix equation $\pi^{t+1} = M_t \pi^t$, where M_t is a $2^n \times 2^n$ stochastic matrix, that is, a matrix that always maps probability vectors to probability vectors. The final outcome of the computation is obtained by sampling from the last probability distribution. The quantum case is similar: an n-qubit state is a 2^n-dimensional vector, but now it is a vector of complex numbers whose *squares* sum to 1. A transformation corresponds to a $2^n \times 2^n$ matrix, but now it is a matrix that preserves the sum of squares of the entries. Finally, a measurement in the computational basis obtains the final outcome by sampling from the distribution given by the squares of the entries of the vector.

2.2 Quantum information and its limitations

Quantum information theory studies the quantum generalizations of familiar notions from classical information theory such as Shannon entropy, mutual information, channel capacities, etc. In Section 3 we give several examples where quantum information theory is used to say something about various non-quantum systems. The quantum information-theoretic results we need all have the same flavor:

they say that a low-dimensional quantum state (i. e., a small number of qubits) cannot contain too much *accessible* information.

Holevo's Theorem The mother of all such results is Holevo's theorem from 1973 [62], which predates the area of quantum computation by many years. Its proper technical statement is in terms of a quantum generalization of mutual information, but the following consequence of it (derived by Cleve et al. [39]) about two communicating parties, suffices for our purposes.

Theorem 2.1 (Holevo, CDNT)**.** *If Alice wants to send n bits of information to Bob via a quantum channel (i. e., by exchanging quantum systems), and they do not share an entangled state, then they have to exchange at least n qubits. If they are allowed to share unlimited prior entanglement, then Alice has to send at least $n/2$ qubits to Bob, no matter how many qubits Bob sends to Alice.*

This theorem is slightly imprecisely stated here, but the intuition is very clear: the first part of the theorem says that if we encode some classical random variable X in an m-qubit state,[1] then no measurement on the quantum state can give more than m bits of information about X. More precisely: the classical mutual information between X and the classical measurement outcome M on the m-qubit system, is at most m. If we encoded the classical information in a *m-bit* system instead of a m-qubit system this would be a trivial statement, but the proof of Holevo's theorem is quite non-trivial. Thus we see that a m-qubit state, despite somehow "containing" 2^m complex amplitudes, is no better than m classical bits for the purpose of storing information (this is in the absence of prior entanglement; if Alice and Bob do share entanglement, then m qubits are no better than $2m$ classical bits).

Low-dimensional encodings Here we provide a "poor man's version" of Holevo's theorem due to Nayak [100, Theorem 2.4.2], which has a simple proof and often suffices for applications. Suppose we have a classical random variable X, uniformly distributed over $[N] = \{1,\dots,N\}$. Let $x \mapsto |\phi_x\rangle$ be some encoding of $[N]$, where $|\phi_x\rangle$ is a pure state in a d-dimensional space. Let $P_1,\dots,P_N$ be the measurement operators applied for decoding; these sum to the d-dimensional identity operator. Then the probability of correct decoding in case $X = x$, is

$$p_x = \|P_x|\phi_x\rangle\|^2 \leq \mathrm{Tr}(P_x)\,.$$

The sum of these success probabilities is at most

$$\sum_{x=1}^{N} p_x \leq \sum_{x=1}^{N} \mathrm{Tr}(P_x) = \mathrm{Tr}\left(\sum_{x=1}^{N} P_x\right) = \mathrm{Tr}(I) = d\,. \tag{2.3}$$

In other words, if we are encoding one of N classical values in a d-dimensional quantum state, then any measurement to decode the encoded classical value has average success probability at most d/N (uniformly averaged over all N values that we can encode).[2] This is optimal. For example, if we encode n

[1] Via an encoding map $x \mapsto |\phi_x\rangle$; we generally use capital letters like X to denote random variables, lower case like x to denote specific values.

[2] For projective measurements the statement is somewhat trivial, since in a d-dimensional space one can have at most d non-zero orthogonal projectors. However, the same proof works for the most general states and measurements that quantum mechanics allows: so-called mixed states (probability distributions over pure states) and POVMs (which are measurements where the operators $P_1,\dots,P_k$ need not be projectors, but can be general positive semidefinite matrices summing to I); see the Appendix for these notions.

bits into m qubits, we will have $N = 2^n$, $d = 2^m$, and the average success probability of decoding is at most $2^m/2^n$.

Random access codes The previous two results dealt with the situation where we encoded a classical random variable X in some quantum system, and would like to recover the original value X by an appropriate measurement on that quantum system. However, suppose $X = X_1 \dots X_n$ is a string of n bits, uniformly distributed and encoded by a map $x \mapsto |\phi_x\rangle$, and it suffices for us if we are able to decode individual bits X_i from this with some probability $p > 1/2$. More precisely, for each $i \in [n]$ there should exist a measurement $\{M_i, I - M_i\}$ allowing us to recover x_i: for each $x \in \{0,1\}^n$ we should have $\||M_i|\phi_x\rangle\|^2 \geq p$ if $x_i = 1$ and $\|M_i|\phi_x\rangle\|^2 \leq 1 - p$ if $x_i = 0$. An encoding satisfying this is called a *quantum random access code*, since it allows us to choose which bit of X we would like to access. Note that the measurement to recover x_i can change the state $|\phi_x\rangle$, so generally we may not be able to decode more than one bit of x.

An encoding that allows us to recover an n-bit string requires about n qubits by Holevo. Random access codes only allow us to recover *each* of the n bits. Can they be much shorter? In small cases they can be: for instance, one can encode two classical bits into one qubit, in such a way that each of the two bits can be recovered with success probability 85% from that qubit [17]. However, Nayak [100] proved that asymptotically quantum random access codes cannot be much shorter than classical (improving upon an $m = \Omega(n/\log n)$ lower bound from [17]).

Theorem 2.2 (Nayak). *Let $x \mapsto |\phi_x\rangle$ be a quantum random access encoding of n-bit strings into m-qubit states such that, for each $i \in [n]$, we can decode X_i from $|\phi_X\rangle$ with success probability p (over a uniform choice of X and the measurement randomness). Then $m \geq (1 - H(p))n$, where $H(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function.*

In fact the success probabilities need not be the same for all X_i; if we can decode each X_i with success probability $p_i \geq 1/2$, then the lower bound on the number of qubits is $m \geq \sum_{i=1}^n (1 - H(p_i))$. The intuition of the proof is quite simple: since the quantum state allows us to predict the bit X_i with probability p_i, it reduces the "uncertainty" about X_i from 1 bit to $H(p_i)$ bits. Hence it contains at least $1 - H(p_i)$ bits of information about X_i. Since all n X_i's are independent, the state has to contain at least $\sum_{i=1}^n (1 - H(p_i))$ bits about X in total. For more technical details see [100] or Appendix B of [74]. The lower bound on m can be achieved up to an additive $O(\log n)$ term, even by classical probabilistic encodings.

2.3 Quantum query algorithms

Different models for quantum algorithms exist. Most relevant for our purposes are the *quantum query algorithms*, which may be viewed as the quantum version of classical decision trees. We will give a basic introduction here, referring to [31] for more details. The model and results of this section will not be needed until Section 4, and the reader might want to defer reading this until they get there.

The query model In this model, the goal is to compute some function $f : A^n \to B$ on a given input $x \in A^n$. The simplest and most common case is $A = B = \{0,1\}$. The distinguishing feature of the query model is the way x is accessed: x is not given explicitly, but is stored in a random access memory, and we are being charged unit cost for each *query* that we make to this memory. Informally, a query asks for and receives the i-th element x_i of the input. Formally, we model a query unitarily as the following 2-register quantum operation O_x, where the first register is n-dimensional and the second is $|A|$-dimensional:

$$O_x : |i,b\rangle \mapsto |i,b+x_i\rangle \,,$$

where for simplicity we identify A with the additive group $\mathbb{Z}_{|A|}$, i. e., addition is modulo $|A|$. In particular, $|i,0\rangle \mapsto |i,x_i\rangle$. This only states what O_x does on basis states, but by linearity determines the full unitary. Note that a quantum algorithm can apply O_x to a superposition of basis states; this gives us a kind of simultaneous access to multiple input variables x_i.

A T-query quantum algorithm starts in a fixed state, say the all-0 state $|0\dots0\rangle$, and then interleaves fixed unitary transformations $U_0,U_1,\dots,U_T$ with queries. It is possible that the algorithm's fixed unitaries act on a workspace-register, in addition to the two registers on which O_x acts. In this case we implicitly extend O_x by tensoring it with the identity operation on this extra register. Hence the final state of the algorithm can be written as the following matrix-vector product:

$$U_T O_x U_{T-1} O_x \cdots O_x U_1 O_x U_0 |0\dots0\rangle \,.$$

This state depends on the input x only via the T queries. The output of the algorithm is obtained by a measurement of the final state. For instance, if the output is Boolean, the algorithm could just measure the final state in the computational basis and output the first bit of the result.

The *query complexity* of some function f is now defined to be the minimal number of queries needed for an algorithm that outputs the correct value $f(x)$ for every x in the domain of f (with error probability at most some fixed value ε). We just count queries to measure the complexity of the algorithm, while the intermediate fixed unitaries are treated as costless. In many cases, including all the ones in this paper, the overall computation time of quantum query algorithms (as measured by the total number of elementary gates, say) is not much bigger than the query complexity. This justifies analyzing the latter as a proxy for the former.

Examples of quantum query algorithms Here we list a number of quantum query algorithms that we will need in later sections. All of these algorithms outperform the best classical algorithms for the given task.

- **Grover's algorithm** [59] searches for a "solution" in a given n-bit input x, i. e., an index i such that $x_i = 1$. The algorithm uses $O(\sqrt{n})$ queries, and if there is at least one solution in x then it finds one with probability at least 1/2. Classical algorithms for this task, including randomized ones, require $\Omega(n)$ queries.

- **ε-error search:** If we want to reduce the error probability in Grover's search algorithm to some small ε, then $\Theta(\sqrt{n\log(1/\varepsilon)})$ queries are necessary and sufficient [30]. Note that this is more efficient than the standard amplification that repeats Grover's algorithm $O(\log(1/\varepsilon))$ times.

- **Exact search:** If we know there are exactly t solutions in our space (i. e., $|x| = t$), then a variant of Grover's algorithm finds a solution with probability 1 using $O(\sqrt{n/t})$ queries [29].

- **Finding all solutions:** If we know an upper bound t on the number of solutions (i. e., $|x| \leq t$), then we can find all of them with probability 1 using $\sum_{i=1}^{t} O(\sqrt{n/i}) = O(\sqrt{tn})$ queries [41].

- **Quantum counting:** The algorithm $Count(x,T)$ of [29] approximates the total number of solutions. It takes as input an $x \in \{0,1\}^n$, makes T quantum queries to x, and outputs an estimate $\tilde{t} \in [0,n]$ to $t = |x|$, the Hamming weight of x. For $j \geq 1$ we have the following concentration bound, implicit in [29]: $\Pr[|\tilde{t} - t| \geq jn/T] = O(1/j)$. For example, using $T = O(\sqrt{n})$ quantum queries we can, with high probability, approximate t up to additive error of $O(\sqrt{n})$.

- **Search on bounded-error inputs:** Suppose the bits $x_1, \ldots, x_n$ are not given by a perfect oracle O_x, but by an imperfect one:

$$O_x : |i,b,0\rangle \mapsto \sqrt{1-\varepsilon_i}\,|i,b \oplus x_i, w_i\rangle + \sqrt{\varepsilon_i}\,|i, \overline{b \oplus x_i}, w_i'\rangle\,,$$

  where we know ε, we know that $\varepsilon_i \leq \varepsilon$ for each x and i, but we do not know the actual values of the ε_i (which may depend on x), or of the "workspace" states $|w_i\rangle$ and $|w_i'\rangle$. We call this an *ε-bounded-error quantum oracle*. This situation arises, for instance, when each bit x_i is itself computed by some bounded-error quantum algorithm. Given the ability to apply O_x as well as its inverse O_x^{-1}, we can still find a solution with high probability using $O(\sqrt{n})$ queries [64]. If the unknown number of solutions is t, then we can still find one with high probability using $O(\sqrt{n/t})$ queries.

From quantum query algorithms to polynomials An *n-variate multilinear polynomial p* is a function $p : \mathbb{C}^n \to \mathbb{C}$ that can be written as

$$p(x_1, \ldots, x_n) = \sum_{S \subseteq [n]} a_S \prod_{i \in S} x_i\,,$$

for some complex numbers a_S. The *degree* of p is $\deg(p) = \max\{|S| : a_S \neq 0\}$. It is well known (and easy to show) that every function $f : \{0,1\}^n \to \mathbb{C}$ has a unique representation as such a polynomial; $\deg(f)$ is defined as the degree of that polynomial. For example, the 2-bit AND function is $p(x_1,x_2) = x_1x_2$, and the 2-bit Parity function is $p(x_1,x_2) = x_1 + x_2 - 2x_1x_2$. Both polynomials have degree 2.

For the purposes of this survey, the crucial property of efficient quantum query algorithms is that the amplitudes of their final state are low-degree polynomials of x [54, 23]. More precisely:

Lemma 2.3. *Consider a T-query algorithm with input $x \in \{0,1\}^n$ acting on an m-qubit space. Then its final state can be written as*

$$\sum_{z \in \{0,1\}^m} \alpha_z(x)|z\rangle\,,$$

where each α_z is a multilinear polynomial in x of degree at most T.

Proof. The proof is by induction on T. The base case ($T = 0$) trivially holds: the algorithm's starting state is independent of x, so its amplitudes are polynomials of degree 0.

For the induction step, note that a fixed linear transformation does not increase the degree of the amplitudes (the new amplitudes are linear combinations of the old amplitudes), while a query to x corresponds to the following map:

$$\alpha_{i,0,w}|i,0,w\rangle + \alpha_{i,1,w'}|i,1,w'\rangle \mapsto ((1-x_i)\alpha_{i,0,w} + x_i\alpha_{i,1,w'})|i,0,w\rangle + (x_i\alpha_{i,0,w} + (1-x_i)\alpha_{i,1,w'})|i,1,w'\rangle\,,$$

which increases the degree of the amplitudes by at most 1: if $\alpha_{i,0,w}$ and $\alpha_{i,1,w'}$ are polynomials in x of degree at most d, then the new amplitudes are polynomials of degree at most $d+1$. Since our inputs are 0/1-valued, we can drop higher degrees and assume without loss of generality that the resulting polynomials are multilinear. □

If we measure the first qubit of the final state and output the resulting bit, then the probability of output 1 is given by

$$\sum_{\substack{z\in\{0,1\}^m,\\ z_1=1}} |\alpha_z|^2\,,$$

which is a real-valued polynomial of x of degree at most $2T$. This is true more generally:

Corollary 2.4. *Consider a T-query algorithm with input $x \in \{0,1\}^n$. Then the probability of a specific output is a multilinear polynomial in x of degree at most $2T$.*

This connection between quantum query algorithms and polynomials has mostly been used as a tool for *lower bounds* [23, 4, 1, 77]: if one can show that every polynomial that approximates a function $f : \{0,1\}^n \to \{0,1\}$ has degree at least d, then every quantum algorithm computing f with small error must use at least $d/2$ queries. We give one example in this spirit in Section 4.5, in which a version of the polynomial method yielded a breakthrough in *classical* lower bounds. However, most of the applications in this survey (in Section 4) work in the other direction: they view quantum algorithms as a means for constructing polynomials with certain desirable properties.

3 Using quantum information theory

The results in this section all use quantum information-theoretic bounds to say something about non-quantum objects.

3.1 Communication lower bound for inner product

The first surprising application of quantum information theory to another area was in *communication complexity*. The basic scenario in this area models 2-party distributed computation: Alice receives some n-bit input x, Bob receives some n-bit input y, and together they want to compute some Boolean function $f(x,y)$, the value of which Bob is required to output (with high probability, in the case of bounded-error protocols). The resource to be minimized is the amount of communication between Alice and Bob, whence the name communication complexity. This model was introduced by Yao [130], and a good overview of (non-quantum) results and applications may be found in the book of Kushilevitz and Nisan [80]. The area is interesting in its own right as a basic complexity measure for distributed computing, but has also found many applications as a tool for lower bounds in areas like data structures, Turing machine complexity, etc. The quantum generalization is quite straightforward: now Alice and Bob can communicate qubits, and possibly start with an entangled state. See [42] for more details and a survey of results.

One of the most studied communication complexity problems is the *inner product* problem, where the function to be computed is the inner product of x and y modulo 2, i. e., $\mathrm{IP}(x,y) = \sum_{i=1}^{n} x_i y_i \bmod 2$. Clearly, n bits of communication suffice for any function—Alice can just send x. However, IP is a good example where one can prove that nearly n bits of communication is also *necessary*. The usual proof for this result is based on the combinatorial notion of "discrepancy," but below we give an alternative quantum-based proof due to Cleve et al. [39].

Intuitively, it seems that unless Alice gives Bob a lot of information about x, he will not be able to guess the value of $\mathrm{IP}(x,y)$. However, in general it is hard to directly lower bound communication complexity by information, since we really require Bob to produce only one bit of output.[3] The very elegant proof of [39] uses quantum effects to get around this problem: it converts a protocol (quantum or classical) that computes IP into a quantum protocol that communicates x from Alice to Bob. Holevo's theorem then easily lower bounds the amount of communication of the latter protocol by the length of x. This goes as follows. Suppose Alice and Bob have some protocol for IP, say it uses c bits of communication. Suppose for simplicity it has no error probability. By running the protocol, putting the answer bit $x \cdot y$ into a phase, and then reversing the protocol to set its workspace back to its initial value, we can implement the following unitary mapping

$$|x\rangle|y\rangle \mapsto |x\rangle(-1)^{x\cdot y}|y\rangle\,.$$

Note that this protocol now uses $2c$ bits of communication: c going from Alice to Bob and c going from Bob to Alice. The trick is that we can run this unitary on a *superposition* of inputs, at a cost of $2c$ *qubits* of communication. Suppose Alice starts with an arbitrary n-bit state $|x\rangle$ and Bob starts with the uniform superposition $\frac{1}{\sqrt{2^n}}\sum_{y\in\{0,1\}^n}|y\rangle$. If they apply the above unitary, the final state becomes

$$|x\rangle\frac{1}{\sqrt{2^n}}\sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle\,.$$

If Bob now applies a Hadamard transform to each of his n qubits, then he obtains the basis state $|x\rangle$, so Alice's n classical bits have been communicated to Bob. Theorem 2.1 now implies that Alice must have sent at least $n/2$ qubits to Bob (even if Alice and Bob started with unlimited shared entanglement). Hence $c \geq n/2$.

With some more technical complication, the same idea gives a linear lower bound on the communication of bounded-error protocols for IP. Nayak and Salzman [101] later obtained optimal bounds for quantum protocols computing IP.

[3] Still, there are also classical techniques to turn this information-theoretic intuition into communication complexity lower bounds [36, 68, 21, 22, 67, 87].

3.2 Lower bounds on locally decodable codes

The development of error-correcting codes is one of the success stories of science in the second half of the 20th century. Such codes are eminently practical, and are widely used to protect information stored on discs, communication over channels, etc. From a theoretical perspective, there exist codes that are nearly optimal in a number of different respects simultaneously: they have constant rate, can protect against a constant noise-rate, and have linear-time encoding and decoding procedures. We refer to Trevisan's survey [123] for a complexity-oriented discussion of codes and their applications.

One drawback of ordinary error-correcting codes is that we cannot efficiently decode small parts of the encoded information. If we want to learn, say, the first bit of the encoded message then we usually still need to decode the whole encoded string. This is relevant in situations where we have encoded a very large string (say, a library of books, or a large database), but are only interested in recovering small pieces of it at any given time. Dividing the data into small blocks and encoding each block separately will not work: small chunks will be efficiently decodable but not error-correcting, since a tiny fraction of well-placed noise could wipe out the encoding of one chunk completely. There exist, however, error-correcting codes that are *locally decodable*, in the sense that we can efficiently recover individual bits of the encoded string. These are defined as follows [72]:

Definition 3.1. $C:\{0,1\}^n \to \{0,1\}^m$ is a (q,δ,ε)-*locally decodable code* (LDC) if there is a classical randomized decoding algorithm A such that

1. A makes at most q queries to an m-bit string y (non-adaptively).
2. For all x and i, and all $y \in \{0,1\}^m$ with Hamming distance $d(C(x),y) \le \delta m$ we have
$$\Pr[A^y(i) = x_i] \ge 1/2 + \varepsilon\,.$$

The notation $A^y(i)$ reflects that the decoder A has two different types of input. On the one hand there is the (possibly corrupted) codeword y, to which the decoder has oracle access and from which it can read at most q bits of its choice. On the other hand there is the index i of the bit that needs to be recovered, which is known fully to the decoder.

The main question about LDCs is the tradeoff between the codelength m and the number of queries q (which is a proxy for the decoding-time). This tradeoff is still not very well understood. We list the best known constructions here. On one extreme, regular error-correcting codes are $(m,\delta,1/2)$-LDCs, so one can have LDCs of linear length if one allows a linear number of queries. Reed-Muller codes allow one to construct LDCs with $m = \mathrm{poly}(n)$ and $q = \mathrm{polylog}(n)$ [37]. For constant q, the best constructions are due to Efremenko [47], improving upon Yekhanin [131]: for $q = 2^r$ one can get codelength roughly $2^{2^{(\log n)^{1/r}}}$, and for $q = 3$ one gets roughly $2^{2^{\sqrt{\log n}}}$. For $q = 2$ there is the Hadamard code: given $x \in \{0,1\}^n$, define a codeword of length $m = 2^n$ by writing down the bits $x \cdot z$ mod 2, for all $z \in \{0,1\}^n$. One can decode x_i with 2 queries as follows: choose $z \in \{0,1\}^n$ uniformly at random and query the (possibly corrupted) codeword at indices z and $z \oplus e_i$, where the latter denotes the string obtained from z by flipping its i-th bit. Individually, each of these two indices is uniformly distributed. Hence for each of them, the probability that the returned bit of is corrupted is at most δ. By the union bound, with probability at least $1-2\delta$, both queries return the uncorrupted values. Adding these two bits modulo 2 gives the correct answer:

$$C(x)_z \oplus C(x)_{z\oplus e_i} = (x\cdot z) \oplus (x \cdot (z \oplus e_i)) = x \cdot e_i = x_i\,.$$

Thus the Hadamard code is a $(2,\delta,1/2-2\delta)$-LDC of exponential length. Can we still do something if we can make only one query instead of two? It turns out that 1-query LDCs do not exist once n is sufficiently large [72].

The only superpolynomial *lower bound* known on the length of LDCs is for the case of 2 queries: there one needs an exponential codelength and hence the Hadamard code is essentially optimal. This was first shown for *linear* 2-query LDCs by Goldreich et al. [57] via a combinatorial argument, and then for general LDCs by Kerenidis and de Wolf [74] via a *quantum* argument.[4] The easiest way to present this argument is to assume the following fact, which states a kind of "normal form" for the decoder.

Fact 3.2 (Katz and Trevisan [72] + folklore)**.** *For every (q,δ,ε)-LDC $C:\{0,1\}^n\to\{0,1\}^m$, and for each $i\in[n]$, there exists a set M_i of $\Omega(\delta\varepsilon m/q^2)$ disjoint tuples, each of at most q indices from $[m]$, and a bit $a_{i,t}$ for each tuple $t\in M_i$, such that the following holds:*

$$\Pr_{x\in\{0,1\}^n}\left[x_i = a_{i,t}\oplus\sum_{j\in t}C(x)_j\right]\geq 1/2+\Omega(\varepsilon/2^q)\,, \tag{3.1}$$

where the probability is taken uniformly over x. Hence to decode x_i from $C(x)$, the decoder can just query the indices in a randomly chosen tuple t from M_i, outputting the sum of those q bits and $a_{i,t}$.

Note that the above decoder for the Hadamard code is already of this form, with $M_i=\{(z,z\oplus e_i)\}$. We omit the proof of Fact 3.2. It uses purely classical ideas and is not hard.

Now suppose $C:\{0,1\}^n\to\{0,1\}^m$ is a $(2,\delta,\varepsilon)$-LDC. We want to show that the codelength m must be exponentially large in n. Our strategy is to show that the following m-dimensional quantum encoding is in fact a quantum random access code for x, with some success probability $p>1/2$:

$$x\mapsto|\phi_x\rangle=\frac{1}{\sqrt{m}}\sum_{j=1}^{m}(-1)^{C(x)_j}|j\rangle\,.$$

Theorem 2.2 then implies that the number of qubits of this state (which is $\lceil\log m\rceil$) is at least $(1-H(p))n=\Omega(n)$, and we are done.

Suppose we want to recover x_i from $|\phi_x\rangle$. We turn each M_i from Fact 3.2 into a measurement: for each pair $(j,k)\in M_i$ form the projector $P_{jk}=|j\rangle\langle j|+|k\rangle\langle k|$, and let $P_{rest}=\sum_{j\notin\cup_{t\in M_i}t}|j\rangle\langle j|$ be the projector on the remaining indices. These $|M_i|+1$ projectors sum to the m-dimensional identity matrix, so they form a valid projective measurement. Applying this to $|\phi_x\rangle$ gives outcome (j,k) with probability $2/m$ for each $(j,k)\in M_i$, and outcome "rest" with probability $\rho=1-\Omega(\delta\varepsilon)$. In the latter case we just output a fair coin flip as our guess for x_i. In the former case the state has collapsed to the following useful superposition:

$$\frac{1}{\sqrt{2}}\left((-1)^{C(x)_j}|j\rangle+(-1)^{C(x)_k}|k\rangle\right)=\frac{(-1)^{C(x)_j}}{\sqrt{2}}\left(|j\rangle+(-1)^{C(x)_j\oplus C(x)_k}|k\rangle\right)$$

[4] The best known lower bounds for general LDCs with $q>2$ queries are only slightly superlinear. Those bounds, and also the best known lower bounds for 2-server *Private Information Retrieval* schemes, are based on similar quantum ideas [74, 126, 128]. The best known lower bound for 3-query LDCs is $m=\Omega(n^2/\log n)$ [128]; for *linear* 3-query LDCs, a slightly better lower bound of $m=\Omega(n^2)$ is known [129].

Doing a 2-outcome measurement in the basis $(1/\sqrt{2})(|j\rangle \pm |k\rangle)$ now gives us the value $C(x)_j \oplus C(x)_k$ with probability 1. By (3.1), if we add the bit $a_{i,(j,k)}$ to this, we get x_i with probability at least $1/2+\Omega(\varepsilon)$. The success probability of recovering x_i, averaged over all x, is

$$p \geq \frac{1}{2}\rho + \left(\frac{1}{2}+\Omega(\varepsilon)\right)(1-\rho) = \frac{1}{2}+\Omega(\delta\varepsilon^2)\,.$$

Now $1-H(1/2+\eta) = \Theta(\eta^2)$ for $\eta \in [0,1/2]$, so after applying Theorem 2.2 we obtain the following:

Theorem 3.3 (Kerenidis and de Wolf)**.** *If $C : \{0,1\}^n \to \{0,1\}^m$ is a $(2,\delta,\varepsilon)$-locally decodable code, then $m = 2^{\Omega(\delta^2\varepsilon^4 n)}$.*

The dependence on δ and ε in the exponent can be improved to $\delta\varepsilon^2$ [74]. This is still the only superpolynomial lower bound known for LDCs. An alternative proof was found later [26], using an extension of the Bonami-Beckner hypercontractive inequality. However, even that proof still follows the outline of the above quantum-inspired proof, albeit in linear-algebraic language.

3.3 Rigidity of Hadamard matrices

In this section we describe an application of quantum information theory to matrix theory from [43]. Suppose we have some $n \times n$ matrix M, whose rank we want to reduce by changing a few entries. The *rigidity* of M measures the minimal number of entries we need to change in order to reduce its rank to a given value r. This notion can be studied over any field, but we will focus here on $\mathbb{R}$ and $\mathbb{C}$. Formally:

Definition 3.4. The *rigidity* of a matrix M is the following function:

$$R_M(r) = \min\{d(M,\widetilde{M}) : \operatorname{rank}(\widetilde{M}) \leq r\}\,,$$

where $d(M,\widetilde{M})$ counts the Hamming distance, i. e., the number of coordinates where M and $\widetilde{M}$ differ. The *bounded rigidity* of M is defined as

$$R_M(r,\theta) = \min\{d(M,\widetilde{M}) : \operatorname{rank}(\widetilde{M}) \leq r, \max_{x,y} |M_{x,y} - \widetilde{M}_{x,y}| \leq \theta\}\,.$$

Roughly speaking, high rigidity means that M's rank is robust: changes in few entries will not change the rank much. Rigidity was defined by Valiant [125, Section 6] in the 1970s with a view to proving circuit lower bounds. In particular, he showed that an explicit $n \times n$ matrix M with $R_M(\varepsilon n) \geq n^{1+\delta}$ for $\varepsilon, \delta > 0$ would imply that log-depth arithmetic circuits (with linear gates) that compute the linear map $M : \mathbb{R}^n \to \mathbb{R}^n$ need superlinear circuit size. This motivates trying to prove lower bounds on rigidity for specific matrices. Clearly, $R_M(r) \geq n-r$ for every full-rank matrix M, since reducing the rank by 1 requires changing at least one entry. This bound is optimal for the identity matrix, but usually far from tight. Valiant showed that most matrices have rigidity $(n-r)^2$, but finding an *explicit* matrix with high rigidity has been open for decades.[5] Similarly, finding explicit matrices with strong lower

[5]Lokam [93] recently found an explicit $n \times n$ matrix with near-maximal rigidity; unfortunately his matrix has fairly large, irrational entries, and is not sufficiently explicit for Valiant's purposes.

bounds on *bounded* rigidity would have applications to areas like communication complexity and learning theory [92, 94].

A very natural and widely studied class of candidates for such a high-rigidity matrix are the *Hadamard matrices*. A Hadamard matrix is an $n \times n$ matrix M with entries $+1$ and -1 that is orthogonal (so $M^T M = nI$). Ignoring normalization, the k-fold tensor product of the matrix from (2.1) is a Hadamard matrix with $n = 2^k$. (It is a longstanding conjecture that Hadamard matrices exist if, and only if, n equals 2 or a multiple of 4.)

Suppose we have a matrix $\widetilde{M}$ differing from the Hadamard matrix M in R positions such that $\operatorname{rank}(\widetilde{M}) \leq r$. The goal in proving high rigidity is to lower-bound R in terms of n and r. Alon [12] proved $R = \Omega(n^2/r^2)$. This was later reproved by Lokam [92] using spectral methods. Kashin and Razborov [71] improved this to $R \geq n^2/256r$. De Wolf [43] later rederived this bound using a quantum argument, with a better constant. We present this argument next.

The quantum idea The idea is to view the rows of an $n \times n$ matrix as a quantum encoding of $[n]$. The rows of a Hadamard matrix M, after normalization by a factor $1/\sqrt{n}$, form an orthonormal set of n-dimensional quantum states $|M_i\rangle$. If Alice sends Bob $|M_i\rangle$ and Bob measures the received state with the projectors $P_j = |M_j\rangle\langle M_j|$, then he learns i with probability 1, since $|\langle M_i|M_j\rangle|^2 = \delta_{i,j}$. Of course, nothing spectacular has been achieved by this—we just transmitted $\log n$ bits of information by sending $\log n$ qubits.

Now suppose that instead of M we have some rank-r $n \times n$ matrix $\widetilde{M}$ that is "close" to M (we are deliberately being vague about "close" here, since two different instantiations of the same idea apply to the two versions of rigidity). Then we can still use the quantum states $|\widetilde{M}_i\rangle$ that correspond to its normalized rows. Alice now sends the normalized i-th row of $\widetilde{M}$ to Bob. Crucially, she can do this by means of an r-dimensional quantum state, as follows. Let $|v_1\rangle, \ldots, |v_r\rangle$ be an orthonormal basis for the row space of $\widetilde{M}$. In order to transmit the normalized i-th row $|\widetilde{M}_i\rangle = \sum_{j=1}^r \alpha_j |v_j\rangle$, Alice sends $\sum_{j=1}^r \alpha_j |j\rangle$ and Bob applies a unitary that maps $|j\rangle \mapsto |v_j\rangle$ to obtain $|\widetilde{M}_i\rangle$. He measures this with the projectors $\{P_j\}$. Then his probability of getting the correct outcome i is

$$p_i = |\langle M_i|\widetilde{M}_i\rangle|^2 .$$

The "closer" $\widetilde{M}$ is to M, the higher these p_i's are. But (2.3) in Section 2.2 tells us that the sum of the p_i's lower-bounds the dimension r of the quantum system. Accordingly, the "closer" $\widetilde{M}$ is to M, the higher its rank has to be. This is exactly the tradeoff that rigidity tries to measure.

This quantum approach allows us to quite easily derive Kashin and Razborov's [71] bound on rigidity, with a better constant.

Theorem 3.5 (de Wolf, improving Kashin and Razborov)**.** *Let M be an $n \times n$ Hadamard matrix. If $r \leq n/2$, then $R_M(r) \geq n^2/4r$.*

Note that if $r \geq n/2$ then $R_M(r) \leq n$, at least for symmetric Hadamard matrices such as $H^{\otimes k}$: then M's eigenvalues are all $\pm\sqrt{n}$, so we can reduce its rank to $n/2$ or less by adding or subtracting the diagonal matrix $\sqrt{n}I$. Hence a superlinear lower bound on $R_M(r)$ cannot be proved for $r \geq n/2$.

Proof. Consider a rank-r matrix $\widetilde{M}$ differing from M in $R = R_M(r)$ entries. By averaging, there exists a set of $a = 2r$ rows of $\widetilde{M}$ with a total number of at most aR/n errors (i. e., changes compared to M). Now consider the submatrix A of $\widetilde{M}$ consisting of those a rows and the $b \geq n - aR/n$ columns that have no errors in those a rows. If $b = 0$ then $R \geq n^2/2r$ and we are done, so we can assume A is nonempty. This A is error-free, hence a submatrix of M itself. We now use the quantum idea to prove the following claim (originally proved by Lokam using linear algebra, see the end of this section):

Claim 3.6 (Lokam)**.** *Every $a \times b$ submatrix A of $n \times n$ Hadamard matrix M has rank $r \geq ab/n$.*

Proof. Obtain the rank-r matrix $\widetilde{M}$ from M by setting all entries outside of A to 0. Consider the a quantum states $|\widetilde{M}_i\rangle$ corresponding to the nonempty rows; they have normalization factor $1/\sqrt{b}$. Alice tries to communicate a value $i \in [a]$ to Bob by sending $|\widetilde{M}_i\rangle$. For each such i, Bob's probability of successfully decoding i is $p_i = |\langle M_i|\widetilde{M}_i\rangle|^2 = |b/\sqrt{bn}|^2 = b/n$. The states $|\widetilde{M}_i\rangle$ are all contained in an r-dimensional space, so (2.3) implies $\sum_{i=1}^{a} p_i \leq r$. Combining both bounds concludes the proof. $\square$

Hence we get

$$r = \operatorname{rank}(\widetilde{M}) \geq \operatorname{rank}(A) \geq \frac{ab}{n} \geq \frac{a(n - aR/n)}{n}.$$

Rearranging gives the theorem. $\square$

Applying the quantum idea in a different way allows us to also analyze *bounded* rigidity:

Theorem 3.7 (Lokam, Kashin and Razborov, de Wolf)**.** *Let M be an $n \times n$ Hadamard matrix and $\theta > 0$. Then*

$$R_M(r,\theta) \geq \frac{n^2(n-r)}{2\theta n + r(\theta^2 + 2\theta)}.$$

Proof. Consider a rank-r matrix $\widetilde{M}$ differing from M in $R = R_M(r,\theta)$ entries, with each entry $\widetilde{M}_{ij}$ differing from M_{ij} by at most θ. As before, define quantum states corresponding to its rows: $|\widetilde{M}_i\rangle = c_i \sum_{j=1}^{n} \widetilde{M}_{i,j}|j\rangle$, where

$$c_i = 1\Big/\sqrt{\sum_j |\widetilde{M}_{i,j}|^2}$$

is a normalizing constant. Note that

$$\sum_j |\widetilde{M}_{i,j}|^2 \leq (n - d(M_i,\widetilde{M}_i)) + d(M_i,\widetilde{M}_i)(1+\theta)^2 = n + d(M_i,\widetilde{M}_i)(\theta^2 + 2\theta),$$

where $d(\cdot,\cdot)$ measures Hamming distance. Alice again sends $|\widetilde{M}_i\rangle$ to Bob to communicate the value $i \in [a]$. Bob's success probability p_i is now

$$p_i = |\langle M_i|\widetilde{M}_i\rangle|^2 \geq \frac{c_i^2}{n}(n - \theta d(M_i,\widetilde{M}_i))^2 \geq c_i^2(n - 2\theta d(M_i,\widetilde{M}_i)) \geq \frac{n - 2\theta d(M_i,\widetilde{M}_i)}{n + d(M_i,\widetilde{M}_i)(\theta^2 + 2\theta)}.$$

Observe that our lower bound on p_i is a convex function of the Hamming distance $d(M_i, \widetilde{M}_i)$. Also, $\mathbb{E}[d(M_i, \widetilde{M}_i)] = R/n$ over a uniform choice of i. Therefore by Jensen's inequality we obtain the lower bound for the average success probability p when i is uniform:

$$p \geq \frac{n - 2\theta R/n}{n + R(\theta^2 + 2\theta)/n}.$$

Now (2.3) implies $p \leq r/n$. Combining and rearranging gives the theorem. □

For $\theta \geq n/r$ we obtain the second result of Kashin and Razborov [71]:

$$R_M(r, \theta) = \Omega(n^2(n-r)/r\theta^2).$$

If $\theta \leq n/r$ we get an earlier result of Lokam [92]:

$$R_M(r, \theta) = \Omega(n(n-r)/\theta).$$

Did we need quantum tools for this? Apart from Claim 3.6 the proof of Theorem 3.5 is fully classical, and that claim itself can quite easily be proved using linear algebra, as was done originally by Lokam [92, Corollary 2.2]. Let $\sigma_1(A), \ldots, \sigma_r(A)$ be the singular values of rank-r submatrix A. Since M is an orthogonal matrix we have $M^T M = nI$, so all M's singular values equal $\sqrt{n}$. The matrix A is a submatrix of M, so all $\sigma_i(A)$ are at most $\sqrt{n}$. Using the Frobenius norm, we obtain the claim:

$$ab = \|A\|_F^2 = \sum_{i=1}^{r} \sigma_i(A)^2 \leq rn.$$

Furthermore, after reading a first version of [43], Midrijanis [98] came up with an even simpler proof of the $n^2/4r$ bound on rigidity for the special case of $2^k \times 2^k$ Hadamard matrices that are the k-fold tensor product of the 2×2 Hadamard matrix.

In view of these simple non-quantum proofs, one might argue that the quantum approach is an overkill here. However, the main point here was not to rederive more or less known bounds, but to show how quantum tools provide a quite different perspective on the problem: we can view a rank-r approximation of the Hadamard matrix as a way of encoding $[n]$ in an r-dimensional quantum system; quantum information-theoretic bounds such as (2.3) can then be invoked to obtain a tradeoff between the rank r and the "quality" of the approximation. The same idea was used to prove Theorem 3.7, whose proof cannot be so easily de-quantized. The hope is that this perspective may help in the future to settle some of the longstanding open problems about rigidity.

4 Using the connection with polynomials

The results of this section are based on the connection explained at the end of Section 2.3: efficient quantum query algorithms give rise to low-degree polynomials.

As a warm-up, we mention a recent application of this. A *formula* is a binary tree whose internal nodes are AND and OR-gates, and each leaf is a Boolean input variable x_i or its negation. The root of the tree computes a Boolean function of the input bits in the obvious way. The size of the formula is its number of leaves. O'Donnell and Servedio [103] conjectured that all formulas of size n have *sign-degree* at most $O(\sqrt{n})$; the sign-degree is the minimal degree among all n-variate polynomials that are positive if, and only if, the formula is 1. Their conjecture implies, by known results, that the class of formulas is *learnable* in the PAC model in time $2^{n^{1/2+o(1)}}$.

Building on a quantum algorithm of Farhi et al. [50] that was inspired by physical notions from scattering theory, Ambainis et al. [16] showed that for every formula there is a quantum algorithm that computes it using $n^{1/2+o(1)}$ queries. By Corollary 2.4, the acceptance probability of this algorithm is an approximating polynomial for the formula, of degree $n^{1/2+o(1)}$. Hence that polynomial minus 1/2 is a sign-representing polynomial for the formula, proving the conjecture of O'Donnell and Servedio up to the $o(1)$ in the exponent. Based on an improved $O(\sqrt{n}\log(n)/\log\log(n))$-query quantum algorithm by Reichardt [112] and some additional analysis, Lee [84] subsequently improved this general upper bound on the sign-degree of formulas to the optimal $O(\sqrt{n})$, fully proving the conjecture (in contrast to [16], he really bounds sign-degree, not approximate degree).

4.1 ε-approximating polynomials for symmetric functions

Our next example comes from [44], and deals with the minimal degree of ε-approximating polynomials for *symmetric* Boolean functions. A function $f:\{0,1\}^n \to \{0,1\}$ is symmetric if its value only depends on the Hamming weight $|x|$ of its input $x \in \{0,1\}^n$. Equivalently, $f(x) = f(\pi(x))$ for all $x \in \{0,1\}^n$ and all permutations $\pi \in S_n$. Examples are OR, AND, Parity, and Majority.

For some specified approximation error ε, let $\deg_\varepsilon(f)$ denote the minimal degree among all n-variate multilinear polynomials p satisfying $|p(x) - f(x)| \le \varepsilon$ for all $x \in \{0,1\}^n$. If one is interested in constant error then one typically fixes $\varepsilon = 1/3$, since approximations with different constant errors can easily be converted into each other. Paturi [104] tightly characterized the 1/3-error approximate degree: if $t \in (0, n/2]$ is the smallest integer such that f is constant for $|x| \in \{t, \dots, n-t\}$, then $\deg_{1/3}(f) = \Theta(\sqrt{tn})$.

Motivated by an application to the inclusion-exclusion principle of probability theory, Sherstov [118] recently studied the dependence of the degree on the error ε. He proved the surprisingly clean result that for all $\varepsilon \in [2^{-n}, 1/3]$,

$$\deg_\varepsilon(f) = \widetilde{\Theta}\left(\deg_{1/3}(f) + \sqrt{n\log(1/\varepsilon)}\right),$$

where the $\widetilde{\Theta}$ notation hides some logarithmic factors (note that the statement is false if $\varepsilon \ll 2^{-n}$, since clearly $\deg(f) \le n$ for all f.) His upper bound on the degree is based on Chebyshev polynomials. De Wolf [44] tightens this upper bound on the degree:

Theorem 4.1 (de Wolf, improving Sherstov)**.** *For every non-constant symmetric function $f:\{0,1\}^n \to \{0,1\}$ and $\varepsilon \in [2^{-n}, 1/3]$:*

$$\deg_\varepsilon(f) = O\left(\deg_{1/3}(f) + \sqrt{n\log(1/\varepsilon)}\right).$$

By the discussion at the end of Section 2.3, to prove Theorem 4.1 it suffices to give an ε-error quantum algorithm for f that uses $O(\deg_{1/3}(f) + \sqrt{n\log(1/\varepsilon)})$ queries. The probability that the algorithm outputs 1 will be our ε-error polynomial. For example, the special case where f is the n-bit OR function follows immediately from the $O(\sqrt{n\log(1/\varepsilon)})$-query search algorithm with error probability ε that was mentioned there.

Here is the algorithm for general symmetric f. It uses some of the algorithms listed in Section 2.3 as subroutines. Let $t = t(f)$ be as in Paturi's result.

1. Use $t-1$ applications of exact Grover to try to find up to $t-1$ distinct solutions in x (remember that a "solution" to the search problem is an index i such that $x_i = 1$). Initially we run an exact Grover assuming $|x| = t-1$, we verify that the outcome is a solution at the expense of one more query, and then we "cross it out" to prevent finding the same solution again in subsequent searches. Then we run another exact Grover assuming there are $t-2$ solutions, etc. Overall, this costs

   $$\sum_{i=1}^{t-1} O(\sqrt{n/i}) = O(\sqrt{tn}) = O(\deg_{1/3}(f))$$

   queries.

2. Use $\varepsilon/2$-error Grover to try to find one more solution. This costs $O(\sqrt{n\log(1/\varepsilon)})$ queries.

3. The same as step 1, but now looking for positions of 0s instead of 1s.

4. The same as step 2, but now looking for positions of 0s instead of 1s.

5. If step 2 did not find another 1, then we assume step 1 found all 1s (i. e., a complete description of x), and we output the corresponding value of f.

   Else, if step 4 did not find another 0, then we assume step 3 found all 0s, and we output the corresponding value of f.

   Otherwise, we assume $|x| \in \{t,\dots,n-t\}$ and output the corresponding value of f.

Clearly the query complexity of this algorithm is $O(\deg_{1/3}(f) + \sqrt{n\log(1/\varepsilon)})$, so it remains to upper bound its error probability. If $|x| < t$ then step 1 finds all 1s with certainty and step 2 will not find another 1 (since there aren't any left after step 1), so in this case the error probability is 0. If $|x| > n-t$ then step 2 finds a 1 with probability at least $1-\varepsilon/2$, step 3 finds all 0s with certainty, and step 4 does not find another 0 (again, because there are none left); hence in this case the error probability is at most $\varepsilon/2$. Finally, if $|x| \in \{t,\dots,n-t\}$ then with probability at least $1-\varepsilon/2$ step 2 will find another 1, and with probability at least $1-\varepsilon/2$ step 4 will find another 0. Thus with probability at least $1-\varepsilon$ we correctly conclude $|x| \in \{t,\dots,n-t\}$ and output the correct value of f. Note that the only property of f used here is that f is constant on $|x| \in \{t,\dots,n-t\}$; the algorithm still works for Boolean functions f that are arbitrary (non-symmetric) when $|x| \notin \{t,\dots,n-t\}$, with the same query complexity $O(\sqrt{tn} + \sqrt{n\log(1/\varepsilon)})$.

4.2 Robust polynomials

In the previous section we saw how quantum query algorithms allow us to construct polynomials (of essentially minimal degree) that ε-approximate symmetric Boolean functions. In this section we show how to construct *robust* polynomial approximations. These are insensitive to small changes in their n input variables. Let us first define more precisely what we mean:

Definition 4.2. Let $p : \mathbb{R}^n \to \mathbb{R}$ be an n-variate polynomial (not necessarily multilinear). Then p ε*-robustly* approximates $f : \{0,1\}^n \to \{0,1\}$ if for every $x \in \{0,1\}^n$ and every $z \in [0,1]^n$ satisfying $|z_i - x_i| \leq \varepsilon$ for all $i \in [n]$, we have $p(z) \in [0,1]$ and $|p(z) - f(x)| \leq \varepsilon$.

Note that we do not restrict p to be multilinear, since the inputs we care about are no longer 0/1-valued. The *degree* of p is its total degree. Note that we require both the z_i's and the value $p(z)$ to be in the interval $[0,1]$. This is just a matter of convenience, because it allows us to interpret these numbers as probabilities; using the interval $[-\varepsilon, 1+\varepsilon]$ instead of $[0,1]$ would give an essentially equivalent definition.

One advantage of the class of robust polynomials over the usual approximating polynomials, is that it is closed under composition: plugging robust polynomials into a robust polynomial gives another robust polynomial. For example, suppose a function $f : \{0,1\}^{n_1 n_2} \to \{0,1\}$ is obtained by composing $f_1 : \{0,1\}^{n_1} \to \{0,1\}$ with n_1 independent copies of $f_2 : \{0,1\}^{n_2} \to \{0,1\}$ (for instance an AND-OR tree). Then we can just compose an ε-robust polynomial for f_1 of degree d_1 with an ε-robust polynomial for f_2 of degree d_2, to obtain an ε-robust polynomial for f of degree $d_1 d_2$. The errors "take care of themselves," in contrast to ordinary approximating polynomials, which may not compose in this fashion.[6]

How hard is it to construct robust polynomials? In particular, does their degree have to be much larger than the usual approximate degree? A good example is the n-bit Parity function. If the n inputs $x_1, \ldots, x_n$ are $0/1$-valued then the following polynomial represents Parity:[7]

$$p(x) = \frac{1}{2} - \frac{1}{2}\prod_{i=1}^{n}(1 - 2x_i)\,. \tag{4.1}$$

This polynomial has degree n, and it is known that any ε-approximating polynomial for Parity needs degree n as well. However, it is clear that this polynomial is not robust: if each $x_i = 0$ is replaced by $z_i = \varepsilon$, then the resulting value $p(z)$ is exponentially close to 1/2 rather than ε-close to the correct value 0. One way to make it robust is to individually "amplify" each input variable z_i, such that if $z_i \in [0, \varepsilon]$ then its amplified version is in, say, $[0, 1/100n]$ and if $z_i \in [1-\varepsilon, 1]$ then its amplified version is in $[1 - 1/100n, 1]$. The following univariate polynomial of degree k does the trick:

$$a(y) = \sum_{j > k/2} \binom{k}{j} y^j (1-y)^{k-j}\,.$$

Note that this polynomial describes the probability that k coin flips, each with probability y of being 1, have majority 1. By standard Chernoff bounds, if $y \in [0, \varepsilon]$ then $a(y) \in [0, \exp(-\Omega(k))]$ and if $y \in [1-\varepsilon, 1]$ then $a(y) \in [1 - \exp(-\Omega(k)), 1]$. Taking $k = O(\log n)$ and substituting $a(z_i)$ for x_i in (4.1) gives an ε-robust polynomial for Parity of degree $\Theta(n \log n)$. Is this optimal? Since Parity crucially depends on each of its n variables, and amplifying each z_i to polynomially small error needs degree $\Theta(\log n)$, one might conjecture robust polynomials for Parity need degree $\Omega(n \log n)$. Surprisingly, this is not the case: there exist ε-robust polynomials for Parity of degree $O(n)$. Even more surprisingly, the only way we know how to construct such robust polynomials is via the connection with quantum algorithms. Based on the quantum search algorithm for bounded-error inputs mentioned in Section 2.3, Buhrman et al. [33] showed the following:

[6] Reichardt [113] showed recently that such a clean composition result also holds for the usual bounded-error quantum query complexity, by going back and forth between quantum algorithms and span programs (which compose cleanly).

[7] If inputs and outputs were ± 1-valued, the polynomial would just be the product of the n variables.

Theorem 4.3 (BNRW)**.** *There exists a quantum algorithm that makes $O(n)$ queries to an ε-bounded-error quantum oracle and outputs $x_1,\ldots,x_n$ with probability at least $1-\varepsilon$.*

The constant in the $O(\cdot)$ depends on ε, but we will not write this dependence explicitly.

Proof (sketch). The idea is to maintain an n-bit string $\widetilde{x}$, initially all-0, and to look for differences between $\widetilde{x}$ and x. Initially this number of differences is $|x|$. If there are t difference points (i. e., $i \in [n]$ where $x_i \neq \widetilde{x}_i$), then the quantum search algorithm A with bounded-error inputs finds a difference point i with high probability using $O(\sqrt{n/t})$ queries. We flip the i-th bit of $\widetilde{x}$. If the search indeed yielded a difference point, then this reduces the distance between $\widetilde{x}$ and x by one. Once there are no differences left, we have $\widetilde{x} = x$, which we can verify by one more run of A. If A only finds difference points, then we would find all differences in total number of queries

$$\sum_{t=1}^{|x|} O(\sqrt{n/t}) = O(\sqrt{|x|n})\,.$$

The technical difficulty is that A errs (i. e., produces an output i where actually $x_i = \widetilde{x}_i$) with constant probability, and hence we sometimes increase rather than decrease the distance between $\widetilde{x}$ and x. The proof details in [33] show that the procedure is still expected to make progress, and with high probability finds all differences after $O(n)$ queries.[8] □

This algorithm implies that we can compute, with $O(n)$ queries and error probability ε, any Boolean function $f : \{0,1\}^n \to \{0,1\}$ on ε-bounded-error inputs: just compute x and output $f(x)$. This is not true for *classical* algorithms running on bounded-error inputs. In particular, classical algorithms that compute Parity with such a noisy oracle need $\Theta(n \log n)$ queries [51].

The above algorithm for f is "robust" in a very similar way as robust polynomials: its output is hardly affected by small errors on its input bits. We now want to derive a robust polynomial from this robust algorithm. However, Corollary 2.4 only deals with algorithms acting on the usual non-noisy type of oracles. We circumvent this problem as follows. Pick a sufficiently large integer m, and fix error-fractions $\varepsilon_i \in [0,\varepsilon]$ that are multiples of $1/m$. Convert an input $x \in \{0,1\}^n$ into $X \in \{0,1\}^{nm} = X_1 \ldots X_n$, where each X_i is m copies of x_i but with an ε_i-fraction of errors (the errors can be placed arbitrarily among the m copies of x_i). Note that the following map is an ε-bounded-error oracle for x that can be implemented by one query to X:

$$|i,b,0\rangle \mapsto |i\rangle \frac{1}{\sqrt{m}} \sum_{j=1}^{m} |b \oplus X_{ij}\rangle |j\rangle = \sqrt{1-\varepsilon_i}\,|i, b\oplus x_i, w_i\rangle + \sqrt{\varepsilon_i}\,|i, \overline{b\oplus x_i}, w_i'\rangle\,.$$

Now consider the algorithm that Theorem 4.3 provides for this oracle. This algorithm makes $O(n)$ queries to X, it is independent of the specific values of ε_i or the way the errors are distributed over X_i, and it has success probability $\geq 1-\varepsilon$ as long as $\varepsilon_i \leq \varepsilon$ for each $i \in [n]$. Applying Corollary 2.4 to this algorithm gives an nm-variate multilinear polynomial p in X of degree $d = O(n)$. This $p(X)$ lies in $[0,1]$ for every

[8] The same idea would work with *classical* algorithms, but gives query complexity roughly $\sum_{t=1}^{|x|} n/t \sim n \ln |x|$.

input $X \in \{0,1\}^{nm}$ (since it is a success probability), and has the property that $p(X_1,\dots,X_n)$ is ε-close to $f(x_1,\dots,x_n)$ whenever $|X_i|/m$ is ε-close to x_i for each i.

It remains to turn each block X_i of m Boolean variables into one real-valued variable z_i. This can be done by the method of *symmetrization* [99] as follows. Define a new polynomial p_1 which averages p over all permutations of the m bits in X_1:

$$p_1(X_1,\dots,X_n) = \frac{1}{m!}\sum_{\pi\in S_m} p(\pi(X_1),X_2,\dots,X_m)\,.$$

Symmetrization replaces terms like $X_{11}\cdots X_{1t}$ by

$$V_t(X_1) = \frac{1}{\binom{m}{t}} \sum_{T\in\binom{[m]}{t}} \prod_{j\in T} X_{1j}\,.$$

Therefore p_1 will be a linear combination of terms of the form $V_t(X_1)r(X_2,\dots,X_n)$ for $t \le d-\deg(r)$. On $X_1 \in \{0,1\}^m$ of Hamming weight $|X_1|$, the sum $V_t(X_1)$ evaluates to

$$\binom{|X_1|}{t} = \frac{|X_1|(|X_1|-1)\cdots(|X_1|-t+1)}{t!}\,,$$

which is a polynomial in $|X_1| = \sum_{j=1}^m X_{1j}$ of degree t. Hence we can define $z_1 = |X_1|/m$, and replace p_1 by a polynomial q_1 of total degree at most d in $z_1,X_2,\dots,X_m$, such that $p_1(X_1,\dots,X_n) = q_1(|X_1|/m,X_2\dots,X_n)$. We thus succeeded in replacing the block X_1 by one real variable z_1. Repeating this for $X_2,\dots,X_n$, we end up with a polynomial $q(z_1,\dots,z_n)$ such that $p(X_1,\dots,X_n) = q(|X_1|/m,\dots,|X_n|/m)$ for all $X_1,\dots,X_n \in \{0,1\}^{nm}$. This q will not be multilinear anymore, but it has degree at most $d = O(n)$ and it ε-robustly approximates f: for every $x\in\{0,1\}^n$ and for every $z\in[0,1]^n$ satisfying $|z_i - x_i| \le \varepsilon$ for all $i\in[n]$, we have that $q(z)$ and $f(x)$ are ε-close. (Strictly speaking we have only dealt with the case where the z_i are multiples of $1/m$, but we can choose m as large as we want and a low-degree polynomial cannot change much if its input varies between i/m and $(i+1)/m$.)

Corollary 4.4 (BNRW). *For every Boolean function f, there exists an n-variate polynomial of degree $O(n)$ that ε-robustly approximates f.*

4.3 Closure properties of PP

The important classical complexity class PP consists of all languages L for which there exists a probabilistic polynomial-time algorithm that accepts an input x with probability at least $1/2$ if $x \in L$, and with probability less than $1/2$ if $x \notin L$. Note that under this criterion, the algorithm's acceptance probabilities may be extremely close to $1/2$, so PP is not a realistic definition of the class of languages feasibly computable with classical randomness. Indeed, it is not hard to see that PP contains NP. Still, PP is worthy of study because of its many relations to other complexity classes.

One of the most basic questions about a complexity class $\mathcal{C}$ is which *closure properties* it possesses. For example, if $L_1, L_2 \in \mathcal{C}$, is $L_1 \cap L_2 \in \mathcal{C}$? That is, is $\mathcal{C}$ *closed under intersection*? In the case of PP, this question was posed by Gill [69], who defined the class, and was open for many years before being answered affirmatively by Beigel et al. [25]. It is now known that PP is closed under significantly more general operations [25, 53, 2]. Aaronson [2] gave a new and arguably more intuitive proof of the known closure properties of PP, by providing a *quantum* characterization of PP.

To describe this result, we first briefly introduce the model of quantum polynomial-time computation. A *quantum circuit* is a sequence of unitary operations $U_1,\ldots,U_T$, applied to the initial state $|x\rangle|0^m\rangle$, where $x\in\{0,1\}^n$ is the input to the circuit and $|0^m\rangle$ is an auxiliary workspace. By analogy with classical circuits, we require that each U_t be a *local* operation which acts on a constant number of qubits. For concreteness, we require each U_t to be a Hadamard gate, or the single-qubit operation

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix},$$

or the two-qubit controlled-NOT gate (which maps computational basis states $|a,b\rangle\mapsto|a,a\oplus b\rangle$). A computation ends by measuring the first workspace qubit. We say that such a circuit computes a function $f_n:\{0,1\}^n\to\{0,1\}$ *with bounded error* if on each $x\in\{0,1\}^n$, the final measurement equals $f_n(x)$ with probability at least $2/3$. BQP is the class of languages computable with bounded error by a logspace-uniform family of polynomial-size quantum circuits. Here, both the workspace size and the number of unitaries are required to be polynomial. The collection of gates we have chosen is *universal*, in the sense that it can efficiently simulate any other collection of local unitaries to within any desired precision [102, Section 4.5.3]. Thus our definition of BQP is a robust one.

In [2], Aaronson investigated the power of a "fantasy" extension of quantum computing in which an algorithm may specify a desired outcome of a measurement in the standard basis, and then condition the quantum state upon seeing that outcome (we require that this event have nonzero probability). Formally, if a quantum algorithm is in the pure state $|\psi\rangle=|\psi_0\rangle|0\rangle+|\psi_1\rangle|1\rangle$ (where we have distinguished a 1-qubit register of interest, and $|\psi_1\rangle$ is non-zero), then the *postselection* transformation carries $|\psi\rangle$ to

$$\frac{|\psi_1\rangle|1\rangle}{\sqrt{\langle\psi_1|\psi_1\rangle}}.$$

The complexity class PostBQP is defined as the class of languages computable with bounded error by a logspace-uniform family of polynomial-size quantum circuits that are allowed to contain postselection gates. We have:

Theorem 4.5 (Aaronson)**.** *PP = PostBQP.*

From Theorem 4.5, the known closure properties of PP follow easily. For example, it is clear that if $L_1,L_2\in$ PostBQP, then we may amplify the success probabilities in the PostBQP algorithms for these languages, then simulate them and take their AND to get a PostBQP algorithm for $L_1\cap L_2$. This shows that PostBQP (and hence also PP) is closed under intersection.

Proof (sketch). We begin with a useful claim about postselection: any quantum algorithm with postselection can be modified to make just a single postselection step after all its unitary transformations (but before its final measurement). We say that such a postselection algorithm is in *canonical form*. To achieve this, given any PostBQP algorithm A for a language L, consider a new algorithm A' which on input x, simulates $A(x)$. Each time A makes a postselecting measurement on a qubit, A' instead records that qubit's value into a fresh auxiliary qubit. At the end of the simulation, A' postselects on the event that all these recorded values are 1, by computing their AND in a final auxiliary qubit $|z\rangle$ and postselecting on $|z\rangle = |1\rangle$. The final state of $A'(x)$ is equivalent to the final state of $A(x)$, so A' is a $\mathsf{PostBQP}$ algorithm for L and is in canonical form. This conversion makes it easy to show that $\mathsf{PostBQP} \subseteq \mathsf{PP}$, by the same techniques which show $\mathsf{BQP} \subseteq \mathsf{PP}$ [5]. We omit the details, and turn to show $\mathsf{PP} \subseteq \mathsf{PostBQP}$.

Let $L \in \mathsf{PP}$, and let M be a probabilistic polynomial-time algorithm witnessing this fact. Say M uses $m = \mathrm{poly}(n)$ random bits on an input x of length n. Then any such input defines a function $g = g_x : \{0,1\}^m \to \{0,1\}$, by the rule

$$g(r) = [M(x) \text{ accepts when using } r \text{ as its random string}] \, .$$

By definition, $x \in L \Leftrightarrow |g^{-1}(1)| \geq 2^{m-1}$. We show how to determine whether $|g^{-1}(1)| \geq 2^{m-1}$, in quantum polynomial time with postselection. Let $s = |g^{-1}(1)|$; we may assume without loss of generality that $s > 0$. The core of the algorithm is the following subroutine, which uses postselection to produce a useful quantum state:

A1. Initialize an $(m+1)$-bit register to $|0^{m+1}\rangle$. Apply $H^{\otimes m}$ to the first m qubits, then apply g to these qubits and add the result into the $(m+1)$-st qubit, yielding the state

$$\frac{1}{\sqrt{2^m}} \sum_{x \in \{0,1\}^m} |x\rangle |g(x)\rangle \, .$$

A2. Apply a Hadamard gate to each qubit of the x-register, yielding

$$\frac{1}{\sqrt{2^m}} \sum_{x \in \{0,1\}^m} \left(\frac{1}{\sqrt{2^m}} \sum_{w \in \{0,1\}^m} (-1)^{w \cdot x} |w\rangle \right) |g(x)\rangle \, ,$$

where $w \cdot x = \sum_i w_i x_i$ denotes inner product of bit strings. Note that $0^m \cdot x = 0$ for all x, so the component of the above state with first register equal to 0^m is

$$\frac{1}{2^m} \left((2^m - s)|0\rangle + s|1\rangle \right) \, .$$

A3. Postselect on the first m qubits measuring to 0^m. This yields the reduced state

$$|\psi\rangle = \frac{(2^m - s)|0\rangle + s|1\rangle}{\sqrt{(2^m - s)^2 + s^2}}$$

on the last qubit.

Using this subroutine, we can prepare fresh copies of $|\psi\rangle$ on demand. Now let (α, β) be a pair of positive reals to be specified later, satisfying $\alpha^2 + \beta^2 = 1$. Using this pair, we define a second subroutine:

B1. Prepare a qubit $|z\rangle = \alpha|0\rangle + \beta|1\rangle$. Perform a controlled-Hadamard operation on a fresh copy of $|\psi\rangle$ with control qubit $|z\rangle$, yielding the joint state $\alpha|0\rangle|\psi\rangle + \beta|1\rangle H|\psi\rangle$. Note that

$$H|\psi\rangle = \frac{\frac{1}{\sqrt{2}}(2^m|0\rangle + (2^m - 2s)|1\rangle)}{\sqrt{(2^m - s)^2 + s^2}} .$$

B2. Now postselect on the second qubit measuring to 1, yielding the reduced state

$$|z'\rangle = \frac{\alpha s|0\rangle + \beta\frac{1}{\sqrt{2}}(2^m - 2s)|1\rangle}{\sqrt{\alpha^2 s^2 + (\beta^2/2)(2^m - 2s)^2}} .$$

Perform a projective measurement relative to the basis

$$\{|+\rangle, |-\rangle\} = \left\{\frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}}\right\},$$

recording the result.

Finally, our algorithm to determine if $s \geq 2^{m-1}$ is as follows: we perform steps B1-B2 for each choice of (α, β) from a collection $\{(\alpha_i, \beta_i)\}_{i=-m}^{m}$ of pairs chosen to satisfy $\alpha_i/\beta_i = 2^i$, $i \in \{-m, \ldots, m\}$. For each choice of i, we repeat steps B1-B2 for $O(\log m)$ trials, and use the results to estimate the probability $|\langle +|z'\rangle|^2$ of outcome $|+\rangle$ under (α_i, β_i).

The idea of the analysis is that, if $s < 2^{m-1}$, then $|z'\rangle$ is in the strictly-positive quadrant of the real plane with axes $|0\rangle, |1\rangle$; here we are using that α, β are positive. Then a suitable choice of (α_i, β_i) will cause $|z'\rangle$ to be closely aligned with $|+\rangle$, and will yield measurement outcome $|+\rangle$ with probability close to 1. On the other hand, if $s \geq 2^{m-1}$, then $|z'\rangle$ is never in the same quadrant as $|+\rangle$ or the equivalent state $-|+\rangle$. Then, for any choice of (α, β) we have $|\langle +|z'\rangle| \leq 1/\sqrt{2}$ and the probability of measuring $|+\rangle$ is at most $1/2$. Thus our repeated trials allow us to determine with high probability whether or not $s \geq 2^{m-1}$. We conclude that $\mathsf{PP} \subseteq \mathsf{PostBQP}$. □

The original proof that PP is closed under intersection [25] crucially relied on ideas from approximation by rational functions, i. e., by ratios of polynomials. Aaronson's proof can also be seen as giving a new method for building rational approximations. To see this, note that the postselection operation also makes sense in the quantum query model, and that in this model as well algorithms may be put in canonical form (with a single postselection). Suppose a canonical-form query algorithm with postselection makes T quantum queries. If its state before the postselection step is $|\psi^T\rangle = |\psi_0^T\rangle|0\rangle + |\psi_1^T\rangle|1\rangle$, then the amplitudes of $|\psi_1^T\rangle$ are degree-T multilinear polynomials in the input x, by Corollary 2.4. Then inspecting the postselected state

$$\frac{|\psi_1^T\rangle|1\rangle}{\sqrt{\langle\psi_1^T|\psi_1^T\rangle}},$$

we see that the *squared* amplitudes are rational functions of x of degree $2T$ in x, that is, each squared amplitude can be expressed as $p(x)/q(x)$, where both numerator and denominator are polynomials of degree at most $2T$. Moreover, all squared amplitudes have the same denominator $q(x) = \langle\psi_1^T|\psi_1^T\rangle$. Thus in the final decision measurement, the acceptance probability is a degree-$2T$ rational function of x.

This may be a useful framework for designing other rational approximations. In fact, one can show the quantum approach to rational approximation always gives an essentially optimal degree: if f is $1/3$-approximated by a rational function of degree d, then f is computed by a quantum query algorithm with postselection, making $O(d)$ queries [45], and vice versa. Such a tight, constant-factor relation between query complexity and degree is known to be false for the case of the usual (non-postselected) quantum query algorithms and the usual approximate degree [15].

4.4 Jackson's theorem

In this section we describe how a classic result in approximation theory, Jackson's Theorem, can be proved using quantum ideas.

The approximation of complicated functions by simpler ones, such as polynomials, has interested mathematicians for centuries. A fundamental result of this type is Weierstrass's Theorem from 1885 [127], which states that any *continuous* function $g : [0,1] \to \mathbb{R}$ can be arbitrarily well-approximated by polynomials. For any function $f : [0,1] \to \mathbb{R}$, let $\|f\|_\infty = \sup_{x\in[0,1]} |f(x)|$; then Weierstrass's Theorem states that for any continuous g and any $\varepsilon > 0$, there exists a polynomial $p(x)$ such that $\|p-g\|_\infty < \varepsilon$.

In 1912 Bernstein gave a simple construction of such polynomials, which can be described in a *probabilistic* fashion [28, 13]. Fix a continuous function g on $[0,1]$ and a value $n \geq 1$, and consider the following algorithm: flip n coins, each taking value 1 with probability x. Let $t \in \{0,\dots,n\}$ be the number of 1s observed in the resulting string $X \in \{0,1\}^n$. Output $g(t/n)$. Note that the expected value of the random variable t/n is exactly x, and its standard deviation is $O(1/\sqrt{n})$. Consider the *expected value* of the output of this procedure, given a bias x:

$$B_{g,n}(x) = \mathbb{E}\left[g\left(|X|/n\right)\right] = \sum_{t=0}^{n} \Pr\left[|X| = t\right] \cdot g\left(t/n\right).$$

We have

$$\Pr\left[|X| = t\right] = \binom{n}{t} x^t (1-x)^{n-t},$$

hence $B_{g,n}(x)$ is a polynomial in x of degree at most n. Moreover, $B_{g,n}(x)$ is intuitively a good estimator for $g(x)$ since t/n is usually close to x. To quantify this, let $\omega_\delta(g)$, the *modulus of continuity of g at scale δ*, be the supremum of $|g(x)-g(y)|$ over all $x, y \in [0,1]$ such that $|x-y| \leq \delta$. Then it can be shown [115] that $\|B_{g,n} - g\|_\infty = O(\omega_{1/\sqrt{n}}(g))$. This is not too surprising since we expect the fraction of heads observed to be concentrated in an interval of length $O(1/\sqrt{n})$ around x. The interval $[0,1]$ is compact, so g is *uniformly* continuous and $\omega_\delta(g) \to 0$ as $\delta \to 0$. Thus the Bernstein polynomials yield a quantitative refinement of Weierstrass's Theorem.

Around the same time as Bernstein's work, an improved result was shown by Jackson [66]:

Theorem 4.6 (Jackson)**.** *If g is continuous, then for all $n \geq 1$ there exists a polynomial $J_{g,n}$ of degree n such that $\|J_{g,n} - g\|_\infty = O(\omega_{1/n}(g))$.*

In Jackson's theorem the quality of approximation is based on the maximum fluctuation of g at a much smaller scale than Bernstein's ($1/n$ instead of $1/\sqrt{n}$). Up to the constant factor, Jackson's Theorem is optimal for approximation guarantees based on the modulus of continuity. The original proof used trigonometric ideas. Drucker and de Wolf [46] gave a proof of Jackson's Theorem that closely follows Bernstein's idea, but replaces his classical estimation procedure with the quantum counting algorithm of [29]. As mentioned in Section 2.3, with M queries this algorithm produces an estimate $\tilde{t}$ of the Hamming weight $t = |X|$ of a string $X \in \{0,1\}^N$, such that for every integer $j \geq 1$ we have $\Pr[|\tilde{t} - t| \geq jN/M] = O(1/j)$. For our purposes we need an even sharper estimate. To achieve this, let $Count'(X,M)$ execute $Count(X,M)$ five times, yielding estimates $\tilde{t}_1, \ldots, \tilde{t}_5$, and output their median, denoted $\tilde{t}_{med}$.[9] Note that for $|\tilde{t}_{med} - t| \geq jN/M$ to hold, we must have $|\tilde{t}_i - t| \geq jN/M$ for at least three of the trials $i \in \{1, \ldots, 5\}$. This happens with probability at most $O(1/j^3)$, which implies

$$\mathbb{E}\left[|\tilde{t}_{med} - t|\right] \leq \sum_{j \geq 1} \frac{jN}{M} \cdot O(1/j^3) = O\left(N/M\right). \tag{4.2}$$

With this estimator in hand, let us fix a continuous function g and $n \geq 1$, and consider the following algorithm $A(x)$ to estimate $g(x)$ for an unknown value x: flip $N = n^2$ independent coins, each taking value 1 with probability x, yielding an N-bit string X. Set $M = \lfloor n/10 \rfloor$ and run $Count'(X,M)$, yielding an estimate $\tilde{t}_{med}$, and output $g(\tilde{t}_{med}/N)$. This algorithm makes $5M \leq n/2$ queries to X, so it follows from Corollary 2.4 that its expected output value (viewed as a function of the N bits of X) is a multilinear polynomial of degree at most n. Define $J_{g,n}(x) = \mathbb{E}[A(x)]$, where the expectation is taken both over the choice of X and over the randomness in the output of the quantum counting procedure. Note that $\mathbb{E}[X_{i_1} \ldots X_{i_d}] = x^d$, since the bits of X are independent coin flips, each with expectation x. Hence $J_{g,n}$ is a degree-n univariate polynomial in x. We bound $|J_{g,n}(x) - g(x)|$ as follows. Consider the random variable $\tilde{x} = \tilde{t}_{med}/N$. Using the definition of $\omega_{1/n}(g)$, we have

$$\begin{aligned}
|J_{g,n}(x) - g(x)| &= |\mathbb{E}[g(\tilde{x})] - g(x)| \\
&\leq \mathbb{E}[|g(\tilde{x}) - g(x)|] \\
&\leq \mathbb{E}[(n \cdot |\tilde{x} - x| + 1) \cdot \omega_{1/n}(g)] \\
&= \left(n \cdot \frac{1}{N} \mathbb{E}[|\tilde{t}_{med} - xN|] + 1\right) \cdot \omega_{1/n}(g) \\
&\leq \left(\frac{1}{n} (\mathbb{E}[|\tilde{t}_{med} - t|] + \mathbb{E}[|t - xN|]) + 1\right) \cdot \omega_{1/n}(g).
\end{aligned}$$

Since $t = |X|$ has expectation xN and variance $x(1-x)N$, we have $\mathbb{E}[|t - xN|] = O(\sqrt{N}) = O(n)$. By (4.2) we have $\mathbb{E}[|\tilde{t}_{med} - t|] = O(N/M) = O(n)$. Plugging these two findings into our expression yields $|J_{g,n}(x) - g(x)| = O(\omega_{1/n}(g))$, for all $x \in [0,1]$. This proves Jackson's Theorem.

At the heart of quantum counting, one finds trigonometric ideas closely related to those used in some classical proofs of Jackson's Theorem (see [46] for a discussion). Thus it is not really fair to call the above a simplified proof. It does, however, show how Bernstein's elegant probabilistic approach to polynomial approximation can be carried further with the use of quantum algorithms.

4.5 Separating strong and weak communication versions of PP

The previous examples all used the connection between polynomials and quantum *algorithms*, more precisely the fact that an efficient quantum (query) algorithm induces a low-degree polynomial. The

[9] A more careful analysis in [46] allows a median of three trials to be used.

last example of this section uses a connection between polynomials and quantum *communication protocols*. Communication complexity was introduced in Section 3.1. Like computational complexity, communication complexity has a host of different models: one can study deterministic, bounded-error, or non-deterministic protocols, and so on. Recall the complexity class PP from Section 4.3. This class has *two* plausible analogues in the communication setting: an unbounded-error version and a weakly-unbounded-error version. In the first, we care about the minimal c such that there is a c-bit randomized communication protocol computing $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ correctly with success probability strictly greater than 1/2 on every input (though the success probability may be arbitrarily close to 1/2). In the second version, we care about the minimal c such that there is a c-bit randomized protocol computing f correctly with success probability at least $1/2+\beta$ on every input, for $\beta \geq 2^{-c}$. The motivation for the second version is that c plays the role that computation-time takes for the computational class PP, and in that class we automatically have $\beta \geq 2^{-c}$: a time-c Turing machine cannot flip more than c fair coins, so all events have probabilities that are multiples of $1/2^c$. Let $\mathsf{UPP}(f)$ and $\mathsf{PP}(f)$ correspond to the minimal communication c in the unbounded and weakly-unbounded cases, respectively. Both measures were introduced by Babai et al. [19], who asked whether they were approximately equal. (One direction is clear: $\mathsf{UPP}(f) \leq \mathsf{PP}(f)$).

These two complexity measures are interesting as models in their own right, but are all the more interesting because they each correspond closely to fundamental complexity measures of Boolean matrices. First, $\mathsf{UPP}(f)$ equals (up to a small additive constant) the log of the *sign-rank* of f [105]. This is the minimal rank among all $2^n \times 2^n$ matrices M satisfying that the sign of the entry $M_{x,y}$ equals $(-1)^{f(x,y)}$ for all inputs. Second, $\mathsf{PP}(f)$ is essentially the *discrepancy* bound [76], which in turn is essentially the *margin complexity* of f [90, 89]. The latter is an important and well-studied concept from computational learning theory. Accordingly, an example where $\mathsf{UPP}(f) \ll \mathsf{PP}(f)$ also separates sign-rank on the one hand from discrepancy and margin complexity on the other.

Simultaneously but independently, Buhrman et al. [34] and Sherstov [119] (later improved in [120]) found functions where $\mathsf{UPP}(f)$ is exponentially smaller than $\mathsf{PP}(f)$, answering the question from [19]. We will give the proof of [34] here. Our quantum tool is the following lemma, first proved (implicitly) in [108] and made more explicit in [77, Section 5].

Lemma 4.7 (Razborov)**.** *Consider a quantum communication protocol on m-bit inputs x and y that communicates q qubits, with outputs 0 and 1 ("reject" and "accept"), and acceptance probabilities denoted by $P(x,y)$. For $i \in \{0,\dots,m/4\}$, define*

$$P(i) = \mathbb{E}_{|x|=|y|=m/4,|x\wedge y|=i}[P(x,y)],$$

where the expectation is taken uniformly over all $x,y \in \{0,1\}^m$ that each have Hamming weight $m/4$ and that have intersection size i. For every $d \leq m/4$ there exists a univariate degree-d polynomial p (over the reals) such that $|P(i)-p(i)| \leq 2^{-d/4+2q}$ for all $i \in \{0,\dots,m/8\}$.

If we choose degree $d = 8q+4\log(1/\varepsilon)$, then p approximates P to within an additive ε for all $i \in \{0,\dots,m/8\}$. This allows us to translate quantum protocols to polynomials. The connection is not as clean as in the case of quantum query algorithms (Corollary 2.4): the relation between the degree and the quantum communication complexity is less tight, and the resulting polynomial only *approximates* certain average acceptance probabilities instead of exactly representing the acceptance probability on each input.

Lemma 4.7 allows us to prove lower bounds on quantum communication complexity by proving lower bounds on polynomial degree. The following result [48, 116] is a very useful degree bound:

Theorem 4.8 (Ehlich and Zeller, Rivlin and Cheney). *Let $p : \mathbb{R} \to \mathbb{R}$ be a polynomial such that $b_1 \leq p(i) \leq b_2$ for every integer $0 \leq i \leq n$, and its derivative has $|p'(x)| \geq c$ for some real $0 \leq x \leq n$. Then $\deg(p) \geq \sqrt{cn/(c+b_2-b_1)}$.*

The function and its unbounded-error protocol To obtain the promised separation, we use a distributed version of the ODD-MAX-BIT function of Beigel [24]. Let $x, y \in \{0,1\}^n$, and $k = \max\{i \in [n] : x_i = y_i = 1\}$ be the rightmost position where x and y both have a 1 (set $k = 0$ if there is no such position). Define $f(x,y)$ to be the least significant bit of k, i. e., whether this k is odd or even. Buhrman et al. [34] proved:

Theorem 4.9 (BVW). *For the distributed ODD-MAX-BIT function we have $\mathsf{UPP}(f) = O(\log n)$ and $\mathsf{PP}(f) = \Omega(n^{1/3})$.*

The upper bound is easy: For $i \in [n]$, define probabilities $p_i = c2^i$, where $c = 1/\sum_{i=1}^n 2^i$ is a normalizing constant. Consider the following protocol. Alice picks a number $i \in [n]$ with probability p_i and sends over (i, x_i) using $\lceil \log n \rceil + 1$ bits. If $x_i = y_i = 1$ then Bob outputs the least significant bit of i, otherwise he outputs a fair coin flip. This computes f with positive (but exponentially small) bias. Hence $\mathsf{UPP}(f) \leq \lceil \log n \rceil + 1$.

Weakly-unbounded-error lower bound It will be convenient to lower-bound the complexity of *quantum* protocols computing f with weakly-unbounded error, since this allows us to use Lemma 4.7.[10] Consider a quantum protocol with q qubits of communication that computes f with bias $\beta > 0$. Let $\beta(x,y) = P(x,y) - 1/2$. Then $\beta(x,y) \geq \beta$ if $f(x,y) = 1$, and $\beta(x,y) \leq -\beta$ if $f(x,y) = 0$. Our goal is to lower-bound $q + \log(1/\beta)$. We will do that below by a process that iteratively fixes some of the input bits, applying Lemma 4.7 to each intermediate function to find a good fixing, in a way that produces a bounded function with larger and larger "swings."

Define $d = \lceil 8q + 4\log(2/\beta) \rceil$ and $m = 32d^2 + 1$. Assume for simplicity that $2m$ divides n. We partition $[n]$ into $n/2m$ consecutive intervals (or "blocks"), each of length $2m$. In the first interval (from the left), fix the bits x_i and y_i to 0 for even i; in the second, fix x_i and y_i to 0 for odd i; in the third, fix x_i and y_i to 0 for even i, etc. In the j-th interval there are m unfixed positions left for each party. Let $x^{(j)}$ and $y^{(j)}$ denote the corresponding m-bit strings in x and y, respectively.

We will define inductively, for all $j = 1, 2, \ldots, n/2m$, particular strings $x^{(j)}$ and $y^{(j)}$ as follows. Let X^j and Y^j denote n-bit strings where the first j blocks are set to $x^{(1)}, \ldots, x^{(j)}$ and $y^{(1)}, \ldots, y^{(j)}$, respectively, and all the other blocks are set to 0^{2m}. In particular, X^0 and Y^0 are all zeros. We will define $x^{(j)}$ and $y^{(j)}$ so that

$$\beta(X^j, Y^j) \geq 2^j \beta \qquad \text{or} \qquad \beta(X^j, Y^j) \leq -2^j \beta$$

[10] We could just analyze classical protocols and use the special case of Razborov's result that applies to classical protocols. However, the classical version of Razborov's lemma was not known prior to [108], and arguably would not have been discovered if it were not for the more general quantum version. We would end up with the same communication lower bound anyway, since quantum and classical weakly-unbounded error complexity turn out to be essentially the same [76, 65].

depending on whether j is odd or even. This holds automatically for $j = 0$, which is the base case of the induction.

Now assume $x^{(1)}, \ldots, x^{(j-1)}$ and $y^{(1)}, \ldots, y^{(j-1)}$ are already defined on previous steps. On the current step, we have to define $x^{(j)}$ and $y^{(j)}$. Without loss of generality assume that j is odd, thus we have $\beta(X^{j-1}, Y^{j-1}) \leq -2^{j-1}\beta$. Consider each $i = 0, 1, \ldots, m/4$. Run the protocol on the following distribution: $x^{(j)}$ and $y^{(j)}$ are chosen randomly subject to each having Hamming weight $m/4$ and having intersection size i; the blocks with indices smaller than j are fixed (on previous steps), and the blocks with indices larger than j are set to zero. Let $P(i)$ denote the expected value (over this distribution) of $\beta(x,y)$ as a function of i. Note that for $i = 0$ we have $P(i) = \beta(X^{j-1}, Y^{j-1}) \leq -2^{j-1}\beta$. On the other hand, for each $i > 0$ the expectation is taken over x, y with $f(x,y) = 1$, because the rightmost intersecting point is in the j-th interval and hence odd (the even indices in the j-th interval have all been fixed to 0). Thus $P(i) \geq \beta$ for those i. Now assume, by way of contradiction, that $\beta(X^j, Y^j) < 2^j\beta$ for all $x^{(j)}, y^{(j)}$ and hence $P(i) < 2^j\beta$ for all such i. By Lemma 4.7, for our choice of d, we can approximate $P(i)$ to within additive error of $\beta/2$ by a polynomial p of degree d. (Apply Lemma 4.7 to the protocol obtained from the original protocol by fixing all bits outside the j-th block.) Let r be the degree-d polynomial

$$r = \frac{p - \beta/2}{2^{j-1}\beta}.$$

From the properties of P and the fact that p approximates P up to $\beta/2$, we see that $r(0) \leq -1$ and $r(i) \in [0,2]$ for all $i \in [m/8]$. But then by the degree lower bound of Theorem 4.8,

$$\deg(r) \geq \sqrt{(m/8)/4} = \sqrt{d^2 + 1/32} > d\,,$$

which is a contradiction. Hence there exists an intersection size $i \in [m/8]$ where $P(i) \geq 2^j\beta$. Thus there are particular $x^{(j)}, y^{(j)}$ with $\beta(X^j, Y^j) \geq 2^j\beta$, concluding the induction step.

For $j = n/2m$ we obtain $|\beta(X^j, Y^j)| \geq 2^{n/2m}\beta$. But for every x, y we have $|\beta(x,y)| \leq 1/2$ because $1/2 + \beta(x,y) = P(x,y) \in [0,1]$, hence $1/2 \geq 2^{n/2m}\beta$. This implies $2m\log(1/2\beta) \geq n$, and therefore

$$(q + \log(1/\beta))^3 \geq (q + \log(1/\beta))^2 \log(1/\beta) = \Omega(m\log(1/\beta)) = \Omega(n)\,.$$

This allows us to conclude $\mathsf{PP}(f) = \Omega(n^{1/3})$, which is exponentially larger than $\mathsf{UPP}(f)$.

5 Other applications

In this section we go over a few further examples of the use of quantum techniques in non-quantum results, examples that do not fit in the two broad categories of the previous two sections. First we give two examples of classical results that were both *inspired* by earlier quantum proofs, but do not explicitly use quantum techniques.[11] Finally, in Section 5.3 we give a brief guide to further literature with more examples.

[11] This is reminiscent of a famous metaphor in Ludwig Wittgenstein's *Tractatus logico-philosophicus* (6.45), about the ladder one discards after having used it to climb to a higher level.

5.1 The relational adversary

In Section 2.3 we described the *polynomial method* for proving lower bounds for quantum query complexity. This method has a strength which is also a weakness: it applies even to a stronger (and less physically meaningful) model of computation where we allow *any linear transformation* on the state space, not just unitary ones. As a result, it does not always provide the strongest possible lower bound for quantum query algorithms.

Ambainis [14, 15], building on an earlier method of Bennett et al. [27], addressed this problem by providing a general method for quantum lower bounds, the *quantum adversary*, which exploits unitarity in a crucial way and which in certain cases yields a provably better bound than the polynomial method [15]. Surprisingly, Aaronson [3] was able to modify Ambainis's argument to obtain a new method, the *relational adversary*, for proving *classical* randomized query lower-bounds. He used this method to give improved lower bounds on the complexity of the "local search" problem, in which one is given a real-valued function F defined on the vertices of a graph G, and must locate a local minimum of F. In this section we state Ambainis's lower-bound criterion, outline its proof, and describe how Aaronson's method follows a similar outline.[12]

Recall from Section 2.3 that a quantum query algorithm is a sequence of unitary operations

$$U_T O_x U_{T-1} O_x \cdots O_x U_1 O_x U_0 \,,$$

applied to the fixed starting state $|0\dots0\rangle$, where the basic "query transformation" O_x depends on the input x and $U_0, U_1, \dots, U_T$ are arbitrary unitaries. Ambainis invites us to look simultaneously at the evolution of our quantum state under all possible choices of x; formally, we let $|\psi_x^t\rangle$ denote the state at time t (i. e., after applying O_x for the t-th time) under input x. In particular, $|\psi_x^0\rangle = |0\dots0\rangle$ for all x (and $\langle\psi_x^0|\psi_y^0\rangle = 1$ for each x,y). Now if the algorithm computes the Boolean function f with success probability $2/3$ on every input, then the final measurement must accept any pair $x \in f^{-1}(0), y \in f^{-1}(1)$ with success probabilities differing by at least $1/3$. It is not hard to verify that this implies $|\langle\psi_x^T|\psi_y^T\rangle| \le 17/18$.[13] This suggests that, for any given relation $R \subseteq f^{-1}(0) \times f^{-1}(1)$ (which one may regard as a bipartite graph), we consider the *progress measure*

$$S_t = \sum_{(x,y)\in R} |\langle\psi_x^t|\psi_y^t\rangle|$$

as a function of t. By our observations, initially we have $S_0 = |R|$, and in the end we must have $S_T \le (17/18)\cdot|R|$. Also, crucially, the progress measure is *unaffected* by each application of a unitary U_t, since each U_t is independent of the input and unitary transformations preserve inner products.

If we can determine an upper-bound Δ on the change $|S_{t+1} - S_t|$ in the progress measure at each step, we can conclude that the number T of queries is at least $|R|/18\Delta$. Ambainis [14, 15] provides a condition on R that allows us to derive such a bound:

Theorem 5.1 (Ambainis)**.** *Suppose that the relation R satisfies*

[12] In both cases we state a simplified ("unweighted") version of the lower-bound method in question, which conveys the essence of the technique. After Ambainis's original paper, a version of the lower bound was formulated that allows even *negative* weights [63]. Reichardt [112] recently proved that this general adversary bound in fact characterizes quantum query complexity.

[13] Contrapositively, and in greater generality, if $|\langle\psi_1|\psi_2\rangle| \ge 1-\varepsilon$ then under any measurement, $|\psi_1\rangle$ and $|\psi_2\rangle$ have acceptance probabilities differing by at most $\sqrt{2\varepsilon}$ (see [102, Section 9.2]).

(i) each $x \in f^{-1}(0)$ appearing in R appears at least m_0 times in R;

(ii) each $y \in f^{-1}(1)$ appearing in R appears at least m_1 times in R;

(iii) for each $x \in f^{-1}(0)$ and $i \in [n]$, there are at most ℓ_0 inputs $y \in f^{-1}(1)$ such that $(x,y) \in R$ and $x_i \neq y_i$;

(iv) for each $y \in f^{-1}(1)$ and $i \in [n]$, there are at most ℓ_1 inputs $x \in f^{-1}(0)$ such that $(x,y) \in R$ and $x_i \neq y_i$.

Let the progress measure S_t be defined relative to a fixed quantum algorithm as above. Then for all $t \geq 0$,

$$|S_{t+1} - S_t| = O\left(\sqrt{\frac{\ell_0}{m_0} \cdot \frac{\ell_1}{m_1}} \cdot |R|\right), \quad \textit{and therefore} \quad T = \Omega\left(\sqrt{\frac{m_0}{\ell_0} \cdot \frac{m_1}{\ell_1}}\right).$$

The art in applying Ambainis's technique lies in choosing the relation R carefully to maximize this quantity. Intuitively, conditions (i)-(iv) imply that $|S_{t+1} - S_t|$ is small relative to $|R|$ by bounding the "distinguishing ability" of any query.

Every classical bounded-error algorithm is also a quantum bounded-error query algorithm, so the above lower-bound also applies in the classical case. However, there are cases where this gives an inferior bound. For example, for the promise problem of inverting a permutation, the above technique yields a query bound of $\Omega(\sqrt{n})$, which matches the true quantum complexity of the problem, while the classical randomized complexity is $\Omega(n)$. In this and similar cases, the particular relation R used in applying Theorem 5.1 for a quantum lower bound, is such that $\max\{m_0/\ell_0, m_1/\ell_1\}$ gives a good estimate of the *classical* query complexity. This led Aaronson to prove in [3] a classical analogue of Ambainis's lower bound, in which the geometric mean of m_0/ℓ_0 and m_1/ℓ_1 is indeed replaced with their maximum.

A sketch of Aaronson's proof is as follows. Fixing the relation R, we use Yao's minimax principle to show that, if there is a randomized T-query bounded-error algorithm for computing f, then there is a *deterministic* algorithm A succeeding with high probability on a specific input distribution determined by R (to be precise, pick a uniformly random pair $(x,y) \in R$ and select x or y with equal probability). We now follow Ambainis, and consider for each input x and $t \leq T$ the state $v_{t,x}$ (which is now a fixed, classical state) of the algorithm A after t steps on input x. Let $I_{t,x,y}$ equal 1 if inputs x and y have not been distinguished by A after t steps, otherwise $I_{t,x,y} = 0$. Define $S_t = \sum_{(x,y)\in R} I_{t,x,y}$ as our progress measure.[14]

Similarly to the quantum adversary, we have $S_0 = |R|$, and Aaronson argues that the success condition of A implies $S_T \leq (1 - \Omega(1))|R|$. A combinatorial argument then yields the following result bounding the maximum possible change $|S_{t+1} - S_t|$ after one (classical) query:

Theorem 5.2 (Aaronson). *Suppose that the relation R obeys conditions (i)-(iv) in Theorem 5.1. Let the progress measure S_t be defined relative to a deterministic algorithm A. Then for all $t \geq 0$,*

$$|S_{t+1} - S_t| = O\left(\min\{\frac{\ell_0}{m_0}, \frac{\ell_1}{m_1}\} \cdot |R|\right) \quad \textit{and therefore} \quad T = \Omega\left(\max\{\frac{m_0}{\ell_0}, \frac{m_1}{\ell_1}\}\right).$$

[14] Actually, Aaronson [3] defines an *increasing* function, but we modify this to show the similarity with Ambainis's proof. We note that if the states $v_{t,x}$ are written as quantum amplitude vectors, and the state of the algorithm A records all the information it sees, $I_{t,x,y}$ can actually be written as an inner product of quantum states just as in Ambainis's proof.

The details of Aaronson's proof are somewhat different from those of Theorem 5.1, and there is no explicit use of quantum states, but the spirit is clearly similar, illustrating that the quantum query model is a sufficiently natural generalization of the classical model for ideas to flow in both directions. Subsequently, Laplante and Magniez [83] gave a different treatment of this based on Kolmogorov complexity, which brings out the analogy between the quantum and classical adversary bounds even more closely.

5.2 Proof systems for the shortest vector problem

A *lattice* is an additive subgroup of $\mathbb{R}^n$ consisting of all integer combinations of a linearly independent set of n vectors. It can be shown that for every lattice L, there exists a value $\lambda(L) > 0$, the *minimum (Euclidean) distance* of the lattice, such that: (i) any two distinct $x, y \in L$ are at distance at least $\lambda(L)$ from each other; (ii) there exists $x \in L$ such that $\|x\| = \lambda(L)$. Lattice vectors of length $\lambda(L)$ are called "shortest (nonzero) vectors" for L, and the problem of computing the minimum lattice distance is also known as the "shortest vector problem" (SVP).

The problem of approximating $\lambda(L)$ to within a multiplicative factor $\gamma(n)$ can be equivalently formulated as a *promise problem* called $\text{GAPSVP}_{\gamma(n)}$, in which we are given a basis for a lattice L, with the "promise" that either $\lambda(L) \leq 1$ or $\lambda(L) \geq \gamma(n)$, and we must determine which case holds. A related problem is the "closest vector problem" (CVP), in which we are given a basis for a lattice L, and a "target" vector $v \notin L$, and would like to approximate the distance $d(v, L)$ from v to the closest vector in the lattice. In the promise problem $\text{GAPCVP}_{\gamma(n)}$, we are given L and v and must distinguish the case where $d(v, L) \leq 1$ from the case where $d(v, L) \geq \gamma(n)$. It is known [58] that $\text{GAPSVP}_{\gamma(n)}$ reduces to $\text{GAPCVP}_{\gamma(n)}$ for any approximation factor $\gamma(n)$.

Approximate solutions to closest and shortest vector problems have numerous applications in pure and applied mathematics. However, $\text{GAPSVP}_{\gamma(n)}$ has been proven intractable even for super-constant approximation factors $\gamma(n)$, albeit under an assumption somewhat stronger than $\mathsf{P} \neq \mathsf{NP}$ [75, 61] (namely that NP-complete problems are not solvable by a randomized algorithm in time $2^{\text{polylog}(n)}$.) Even computing an estimate with an *exponential* approximation guarantee in polynomial time is a highly nontrivial task, achieved by the celebrated LLL algorithm [86]; the best current polynomial-time algorithm gives only slightly subexponential approximation factors [11]. A nearly exponential gap remains between the efficiently achievable approximation ratios and those for which we have complexity-theoretic evidence of hardness. Also, despite intense effort, no quantum algorithms have been found which improve significantly on their classical counterparts.

A breakthrough work of Ajtai [8] initiated a sequence of papers [9, 109, 95, 111, 10, 106] which gave a strong motivation to better understand the approximability of various lattice problems. These papers build cryptosystems which, remarkably, possess strong *average-case* security, based only on the assumption that certain lattice problems are hard to approximate within polynomial factors in the *worst case*.[15] For a fuller discussion of lattice-based cryptography, see [96, 97].

[15] Intriguingly for us, Regev [110] gave one such cryptosystem based on a *quantum* hardness assumption, and recently Peikert [106] built upon ideas in [110] to give a new system based on hardness assumptions against classical algorithms. This is another example of a classical result based on an earlier quantum result. However, the connection is less tight than for the proof systems of Aharonov and Regev, since Peikert replaced the quantum aspect of Regev's earlier work by a fairly unrelated classical approach.

While these hardness assumptions remain plausible, another sequence of papers [81, 20, 56, 6, 7] has given evidence that the shortest vector problem is *not* NP-hard to approximate within polynomial factors (note that a problem may be intractable without being NP-hard). The most recent, due to Aharonov and Regev [7], is a proof system for $\text{GAPCVP}_{c\sqrt{n}}$ for sufficiently large $c > 0$. That is, if we are given a basis for a lattice L with the promise that either $d(v,L) \leq 1$ or $d(v,L) \geq c\sqrt{n}$, there is a proof system to convince us which of the two cases holds. One direction is simple: if $d(v,L) \leq 1$, then exhibiting a lattice vector within distance 1 of v proves this (the promise is not needed in this case). The new contribution of [7] is to supply proofs for the case where $d(v,L) \geq c\sqrt{n}$, showing that $\text{GAPCVP}_{c\sqrt{n}}$ is in $\mathsf{NP} \cap \mathsf{coNP}$ (technically, the promise-problem analogue of this class). It follows that $\text{GAPCVP}_{c\sqrt{n}}$ cannot be NP-hard unless the Polynomial Hierarchy collapses; see [7] for details of this implication.

The proof system of [7] is, in Aharonov and Regev's own description, a "dequantization" of an earlier, *quantum* Merlin-Arthur (QMA) proof system by the same authors [6] for $\text{GAPCVP}_{c\sqrt{n}}$.[16] In a QMA protocol, a computationally unbounded but untrustworthy prover (Merlin) sends a quantum state to a polynomially-bounded quantum verifier (Arthur), who then decides to accept or reject without further communication; unlike NP, QMA allows a small probability of error. Although Aharonov and Regev analyze their second proof system in a completely classical way, the two papers offer an interesting case study of how quantum ideas can be adapted to the classical setting. We now describe some of the ideas involved; we begin with the quantum proof system of [6], then discuss the classical protocol of [7].

The quantum proof system In the QMA protocol, Prover wants to convince Verifier that $d(v,L) \geq c\sqrt{n}$ for some large $c > 0$. Central to the proof system is the following geometric idea: for our given lattice L, one defines a function $F(x) : \mathbb{R}^n \to [0,\infty)$ that is *lattice-periodic* (i. e., $F(x+y) = F(x)$ for all $x \in \mathbb{R}^n$ and $y \in L$) and is heavily concentrated in balls of radius $\ll \sqrt{n}$ around the lattice points. Now for each $z \in \mathbb{R}^n$ we consider the "z-shifted" version of F, $F_z(x) = F(x+z)$. The central idea is that if $d(z,L) \leq 1$, then (informally speaking) F_z has large "overlap" with F, since the centers of mass shift only slightly. On the other hand, if $d(z,L) \geq c\sqrt{n}$, then F_z and F have negligible overlap, since the masses of the two functions are then concentrated on disjoint sets of balls. In the proof system, Prover aims to convince Verifier that this overlap is indeed negligible when $z = v$ is the target vector. To this end, Verifier asks to receive the "state"

$$|\xi\rangle = \sum_{x \in \mathbb{R}^n} F(x)|x\rangle$$

describing the pointwise behavior of $F(x)$. (Here and throughout, we give a simplified, idealized description; the actual protocol uses points x of finite precision, over a bounded region that captures the behavior of the L-periodic function F. Thus all sums become finite, and states can be properly normalized.) We think of $|\xi\rangle$ as the "correct" proof which Verifier hopes to receive from Prover; note that $|\xi\rangle$ is independent of v.

Verifier cannot hope to recover an arbitrary value from among the exponentially many values stored in $|\xi\rangle$. However, given $|\xi\rangle$, an elegant technique allows Verifier to estimate the overlap of F with F_v, if

[16] Actually, for the quantum protocol of [6], we need the slightly stronger promise that when the input satisfies $d(v,L) \geq c\sqrt{n}$, it also holds that $\lambda(L) \geq c\sqrt{n}$. A proof system for this restricted problem still yields a proof system for SVP with approximation factor $c\sqrt{n}$.

the overlap $\langle F, F_v \rangle$ is defined (imprecisely) as

$$\langle F, F_v \rangle = \sum_{x \in \mathbb{R}^n} F(x) F(x+v) \,.$$

Aharonov and Regev show that the overlap measure $\langle F, F_v \rangle$ they define is extremely close to $e^{-\pi d(v,L)^2/4}$ for any v provided that $\lambda(L) \geq c\sqrt{n}$, so that it is indeed a good indicator of distance from the lattice.

To estimate this overlap, Verifier first appends a "control" qubit to $|\xi\rangle$, initialized to $(1/\sqrt{2})\,(|0\rangle + |1\rangle)$. Conditioned on the control qubit being 1, he applies the transformation T_v which takes $|y\rangle$ to $|y-v\rangle$; this yields the state $(1/\sqrt{2})\,(|0\rangle|\xi\rangle + |1\rangle T_v|\xi\rangle)$. He then applies a Hadamard transformation to the control qubit, yielding

$$\frac{1}{2}\left[|0\rangle(|\xi\rangle + T_v|\xi\rangle) + |1\rangle(|\xi\rangle - T_v|\xi\rangle)\right] .$$

Finally, he measures the control qubit, which equals 1 with probability

$$\frac{1}{4}(\langle \xi| - \langle \xi | T_v^*)(|\xi\rangle - T_v|\xi\rangle) = \frac{1}{2}(1 - \mathrm{Re}(\langle \xi | T_v | \xi \rangle)) \,.$$

Consulting the definition of $|\xi\rangle$ and using that F is real-valued, we have $\mathrm{Re}(\langle \xi | T_v | \xi \rangle) = \langle F, F_v \rangle$, so the probability of measuring a 1 is linearly related to the overlap Verifier wants to estimate.

The procedure above allows Verifier to estimate $d(v,L)$, *on the assumption* that Prover supplies the correct quantum encoding of F. But Prover could send any quantum state $|\psi\rangle$, so Verifier needs to test that $|\psi\rangle$ behaves something like the desired state $|\xi\rangle$. In particular, for randomly chosen vectors z within a ball of radius $1/\mathrm{poly}(n)$ around the origin, the overlap $\langle F, F_z \rangle$, estimated by the above procedure, should be about $e^{-\pi \|z\|^2/4}$.

Consider $h_\psi(z) = \mathrm{Re}(\langle \psi | T_z | \psi \rangle)$ as a function of z. If this function could be arbitrary as we range over choices of $|\psi\rangle$, then observing that $h_\psi(z) \approx h_\xi(z)$ for short vectors z would give us no confidence that $h_\psi(v) \approx \langle F, F_v \rangle$. However, Aharonov and Regev show that for every $|\psi\rangle$, the function $h_\psi(z)$ obeys a powerful constraint called *positive definiteness*. They then prove that any positive definite function which is sufficiently "close on average" to the Gaussian $e^{-\pi\|z\|^2/4}$ in a small ball around 0, cannot simultaneously be too close to zero at any point within a distance 1 of L. Thus if $d(v,L) \leq 1$, any state Prover sends must either fail to give the expected behavior around the origin, or fail to suggest negligible overlap of F with F_v. We have sketched the core ideas of the quantum proof system; the actual protocol requires multiple copies of $|\xi\rangle$ to be sent, with additional, delicate layers of verification.

The classical proof system We now turn to see how some of the ideas of this proof system were adapted in [7] to give a classical, deterministic proof system for $\mathrm{GAPCVP}_{c\sqrt{n}}$. The first new insight is that for any lattice L, there is an L-periodic function $f(x)$ which behaves similarly to $F(x)$ in the previous protocol, and which can be efficiently and accurately approximated at any point, using a polynomial-sized *classical* advice string. (The function $f(x)$ is defined as a sum of Gaussians, one centered around each point in L.) This advice is not efficiently constructible given a basis for L, but we may still hope to verify this advice if it is supplied by a powerful prover.

These approximations are derived using ideas of Fourier analysis. First we introduce the *dual lattice* L^*, which consists of all $w \in \mathbb{R}^n$ satisfying $\langle w, y \rangle \in \mathbb{Z}$ for all $y \in L$ (here we use $\langle \cdot, \cdot \rangle$ to denote inner product). For example, we have $(\mathbb{Z}^n)^* = \mathbb{Z}^n$. Note that for $w \in L^*$, $r_w(x) = \cos(2\pi\langle w,x\rangle)$ is an L-periodic function. In fact, any sufficiently smooth L-periodic real-valued function, such as $f(x)$, can be uniquely expressed as

$$f(x) = \sum_{w \in L^*} \widehat{f}(w)\cos(2\pi\langle w,x\rangle)\,,$$

where the numbers $\widehat{f}(w) \in \mathbb{R}$ are called the *Fourier coefficients* of f. Our choice of f has particularly nicely-behaved Fourier coefficients: they are nonnegative and sum to 1, yielding a probability distribution (denoted by $\widehat{f}$). Thus

$$f(x) = \mathbb{E}_{w \sim \widehat{f}}[\cos(2\pi\langle w,x\rangle)]\,.$$

It follows from standard sampling ideas that a large enough ($N = \mathrm{poly}(n)$-sized) set of samples $W = w_1,\dots,w_N$ from $\widehat{f}$ will, with high probability, allow us to accurately estimate $f(x)$ at every point in a large, fine grid about the origin, by the approximation rule

$$f_W(x) = \frac{1}{N}\sum_{i=1}^{N}\cos(2\pi\langle w_i,x\rangle)\,.$$

Since f is L-periodic and smooth, this yields good estimates everywhere. So, we let the proof string consist of a matrix W of column vectors $w_1,\dots,w_N$. Since Prover claims that the target vector v is quite far from the lattice, and f is concentrated around the lattice points, Verifier expects to see that $f_W(x)$ is small, say, less than $1/2$.

As usual, Verifier must work to prevent Prover from sending a misleading proof. The obvious first check is that the vectors w_i sent are indeed in L^*. For the next test, a useful fact (which relies on properties of the Gaussians used to define f) is that samples drawn from $\widehat{f}$ are not too large and "tend not to point in any particular direction": $\mathbb{E}_{w\sim\widehat{f}}[\langle u,w\rangle^2] \le 1/2\pi$ for any unit vector u. This motivates a second test to be performed on the samples $w_1,\dots,w_N$: check that the largest eigenvalue of WW^T is at most $3N$.

In a sense, this latter test checks that f_W has the correct "shape" in a neighborhood of the origin, and plays a similar role to the random-sampling test from the quantum protocol.[17] Like its quantum counterpart, this new test is surprisingly powerful: if W is *any* collection of dual-lattice vectors satisfying the eigenvalue constraint, and $d(v,L) \le 1/100$, then $f_W(v) \ge 1/2$ and Prover cannot use W to make v pass the test. On the other hand, if $d(v,L) \ge c\sqrt{n}$ for large enough c, then choosing the columns of W according to $\widehat{f}$ yields a witness that with high probability passes the two tests and satisfies $f_W(v) < 1/2$. Scaling by a factor 100 gives a proof system for $\textsc{GapCVP}_{100c\sqrt{n}}$.

5.3 A guide to further literature

In this section we give pointers to other examples where quantum techniques are used to obtain classical results in some way or other:

- **Data structure lower bounds:** Using linear-algebraic techniques, Radhakrishnan et al. [107] proved lower bounds on the bit-length of data structures for the set membership problem with

[17] Indeed, the testing ideas are similar, due to the fact that for any W, the function f_W obeys the same "positive definiteness" property used to analyze the quantum protocol.

quantum decoding algorithms. Their bounds of course also apply to classical decoding algorithms, but are in fact stronger than the previous classical lower bounds of Buhrman et al. [32]. Sen and Venkatesh did the same for data structures for the predecessor problem [117], proving a "round elimination" lemma in the context of quantum communication complexity that is stronger than the best known classical round elimination lemmas. A further strengthening of their lemma was subsequently used by Chakrabarti and Regev [35] to obtain optimal lower bounds for the approximate nearest neighbour problem.

- **Formula lower bounds:** Recall that a formula is a binary tree whose internal nodes are AND and OR-gates, and each leaf is a Boolean input variable x_i or its negation. The root of the tree computes a Boolean function of the inputs in the obvious way. Proving superpolynomial formula lower bounds for specific explicit functions in NP is a long-standing open problem in complexity theory, the hope being that such a result would be a stepping-stone towards the superpolynomial circuit lower bounds that would separate P from NP (currently, not even superlinear bounds are known). The best proven formula-size lower bounds are nearly cubic [60], but a large class of quadratic lower bounds can be obtained from a quantum result: Laplante et al. [82] showed that the formula size of f is lower bounded by the square of the quantum adversary bound for f (mentioned in Section 5.1). Since random functions, as well as many specific functions like Parity and Majority, have linear adversary bounds, one obtains many quadratic formula lower bounds this way.

- **Circuit lower bounds:** Kerenidis [73] describes an approach to prove lower bounds for classical circuit *depth* using quantum multiparty communication complexity. Roughly speaking, the idea is to combine two classical parties into one quantum party, prove lower bounds in the resulting quantum model, and then translate these back to strong lower bounds for the classical model. (Unfortunately, it seems hard to prove good lower bounds for the resulting quantum model [85].)

- **Horn's problem** is to characterize the triples μ, ν, and λ of vectors of integers for which there exist Hermitian operators A and B such that μ, ν, and λ are the spectra of A, B, and $A+B$, respectively. It is known (and quite non-trivial) that this problem is equivalent to a question about the representation theory of the group $\mathsf{GL}(d)$ of invertible $d \times d$ complex matrices. Christandl [38] reproved a slightly weaker version of this equivalence based on quantum information theory.

- **Secret-key distillation:** In cryptography, Gisin, Renner, and Wolf [55] used an analogy with "quantum bound entanglement" to provide evidence against the conjecture that the "intrinsic information" in a random variable shared by Alice, Bob, and eavesdropper Eve always equals the amount of secret key that Alice and Bob can extract from this; later this conjecture was indeed disproved [114], though without using quantum methods.

- **Equivalence relations and invariants:** A function f is said to be a *complete invariant* for the equivalence relation R if $(x,y) \in R \Leftrightarrow f(x) = f(y)$. It is an interesting question whether every polynomial-time computable equivalence relation has a polynomial-time computable complete invariant. Fortnow and Grochow [52, Theorem 4.3] show this would imply that the class UP reduces to Simon's problem [122] (and hence would be in BQP, which seems unlikely).

- **Learning positive semidefinite matrices:** Tsuda, Rätsch, and Warmuth [124] study online learning of positive semidefinite matrices using properties of von Neumann divergence (also known as quantum relative entropy) in their analysis in order to measure differences between density matrices.

- **Johansson's theorem** asymptotically equates the expected shape of the semi-standard tableau produced by a random word in k letters, with the spectrum of a certain random matrix. Kuperberg [78] provides a proof of this theorem by treating the random matrix as a quantum random variable on the space of random words. In another paper, Kuperberg [79] also proved a central limit theorem for non-commutative random variables in a von Neumann algebra with a tracial state, based on quantum ideas.

6 Conclusion

In this paper we surveyed the growing list of applications of quantum computing techniques to non-quantum problems, in areas ranging from theoretical computer science to pure mathematics. These proofs build on current research in quantum computing, but do not depend on whether a large-scale quantum computer will ever be built. One could even go further, and use mathematical frameworks that go "beyond quantum" as a proof-tool. The use of PostBQP in Section 4.3 is in this vein, since we don't expect postselection to be physically implementable.

We feel that "thinking quantumly" can be a source of insight and of charming, surprising proofs: both proofs of new results and simpler proofs of known results (of course, "simplicity" is in the eye of the beholder here). While the examples in this survey do not constitute a full-fledged proof method yet, our hope is that both the quantum toolbox and its range of applications will continue to grow.

A The most general quantum model

The ingredients introduced in Section 2.1 are all the quantum mechanics we need for the applications of this survey. However, a more general formalism exists, which we will explain here with a view to future applications—who knows what these may need!

First we generalize pure states. In the classical world we often have uncertainty about the state of a system, which can be expressed by viewing the state as a random variable that has a certain probability distribution over the set of basis states. Similarly we can define a *mixed* quantum state as a probability distribution (or "mixture") over pure states. While pure states are written as vectors, it is most convenient to write mixed states as *density matrices*. A pure state $|\phi\rangle$ corresponds to the rank-one density matrix $|\phi\rangle\langle\phi|$, which is the outer product of the vector $|\phi\rangle$ with itself. A mixed state that is in (not necessarily orthogonal) pure states $|\phi_1\rangle,\dots,|\phi_k\rangle$ with probabilities $p_1,\dots,p_k$, respectively, corresponds to the density matrix $\sum_{i=1}^k p_i|\phi_i\rangle\langle\phi_i|$. The set of density matrices is exactly the set of positive semidefinite (PSD) matrices of trace 1. A mixed state is pure if, and only if, its density matrix has rank 1.

The most general quantum operation on density matrices is a *completely-positive, trace-preserving (CPTP) map*. By definition, this is a linear map that sends density matrices to density matrices, even when tensored with the identity operator on another space. Alternatively, a map $\mathcal{S} : \rho \mapsto S(\rho)$ from

$d \times d$-matrices to $d' \times d'$-matrices is a CPTP map if, and only if, it has a *Kraus-representation*: there are $d' \times d$ matrices $M_1, \ldots, M_k$, satisfying $\sum_{i=1}^k M_i^* M_i = I$, such that $\mathcal{S}(\rho) = \sum_{i=1}^k M_i \rho M_i^*$ for every $d \times d$ density matrix ρ. A unitary map U corresponds to $k = 1$ and $M_1 = U$, so unitaries act on mixed states by conjugation: $\rho \mapsto U\rho U^*$. Note that a CPTP map can change the dimension of the state. For instance, the map that traces out ("throws away") the second register of a 2-register state is a CPTP map. Formally, this map is defined on tensor-product states as $\mathrm{Tr}_2(A \otimes B) = A$, and extended to all 2-register states by linearity.

CPTP maps also include measurements as a special case. For instance, a projective measurement with projectors $P_1, \ldots, P_k$ that writes the classical outcome in a second register, corresponds to a CPTP map $\mathcal{S}$ with Kraus operators $M_i = P_i \otimes |i\rangle$. We now have

$$\mathcal{S}(\rho) = \sum_{i=1}^{k} M_i \rho M_i^* = \sum_{i=1}^{k} \frac{P_i \rho P_i^*}{\mathrm{Tr}(P_i \rho P_i^*)} \otimes \mathrm{Tr}(P_i \rho P_i^*) |i\rangle\langle i| \,.$$

This writes the classical value i in the second register with probability $\mathrm{Tr}(P_i \rho P_i^*)$, and "collapses" ρ in the first register to its normalized projection in the subspace corresponding to P_i.

While this framework of mixed states and CPTP maps looks more general than the earlier framework of pure states and unitaries, philosophically speaking it is not: every CPTP map can be implemented unitarily on a larger space. What this means is that for every CPTP map $\mathcal{S}$, there exists a state ρ_0 on an auxiliary space, and a unitary on the joint space, such that for every ρ, the state $\mathcal{S}(\rho)$ equals what one gets by tracing out the auxiliary register from the state $U(\rho \otimes \rho_0)U^*$. We refer to the book of Nielsen and Chuang [102, Section 8.2] for more details.

Acknowledgements

We thank Richard Lipton for his blog entry on these techniques [91], which was one of the things that motivated us to write this survey. We thank Scott Aaronson, Joshua Brody, Iordanis Kerenidis, Greg Kuperberg, Troy Lee, Frédéric Magniez, Ashwin Nayak, Alexander Sherstov, and Shengyu Zhang for helpful comments and pointers to the literature. Finally, many thanks to the anonymous ToC referees for their very helpful comments.

References

[1] SCOTT AARONSON: Limitations of quantum advice and one-way communication. *Theory of Computing*, 1:1–28, 2005. Earlier version in Proc. 19th IEEE Conf. Comput. Complex., 2004 and arxiv:quant-ph/0402095. [doi:10.4086/toc.2005.v001a001] 14

[2] SCOTT AARONSON: Quantum computing, postselection, and probabilistic polynomial-time. In *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, volume A461(2063), pp. 3473–3482, 2005. Earlier version in arxiv:quant-ph/0412187. [doi:10.1098/rspa.2005.1546] 27

[3] SCOTT AARONSON: Lower bounds for local search by quantum arguments. *SIAM J. Comput.*, 35(4):804–824, 2006. Earlier version in STOC '03 and arxiv:quant-ph/0307149. [doi:10.1137/S0097539704447237] 35, 36

[4] SCOTT AARONSON AND YAOYUN SHI: Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004. [doi:10.1145/1008731.1008735] 14

[5] LEONARD M. ADLEMAN, JONATHAN DEMARRAIS, AND MING-DEH A. HUANG: Quantum computability. *SIAM J. Comput.*, 26(5):1524–1540, 1997. [doi:10.1137/S0097539795293639] 28

[6] DORIT AHARONOV AND ODED REGEV: A lattice problem in quantum NP. In *Proc. 44th FOCS*, pp. 210–219. IEEE Comp. Soc. Press, 2003. Earlier version in arxiv:quant-ph/0307220. [doi:10.1109/SFCS.2003.1238195] 38

[7] DORIT AHARONOV AND ODED REGEV: Lattice problems in NP intersect coNP. *J. ACM*, 52(5):749–765, 2005. Earlier version in FOCS '04. [doi:10.1145/1089023.1089025] 38, 39

[8] MIKLÓS AJTAI: Generating hard instances of lattice problems (extended abstract). In *Proc. 28th STOC*, pp. 99–108. ACM Press, 1996. [doi:10.1145/237814.237838] 37

[9] MIKLÓS AJTAI AND CYNTHIA DWORK: A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. 29th STOC*, pp. 284–293. ACM Press, 1997. [doi:10.1145/258533.258604] 37

[10] MIKLÓS AJTAI AND CYNTHIA DWORK: The first and fourth public-key cryptosystems with worst-case/average-case equivalence. *Electron. Colloq. on Comput. Complexity (ECCC)*, 14(097), 2007. [ECCC:TR07-097] 37

[11] MIKLÓS AJTAI, RAVI KUMAR, AND D. SIVAKUMAR: A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd STOC*, pp. 601–610. ACM Press, 2001. [doi:10.1145/380752.380857] 37

[12] NOGA ALON: On the rigidity of an Hadamard matrix. Manuscript. His proof may be found in [70, Section 15.1.2], 1990. 19

[13] NOGA ALON AND JOEL H. SPENCER: *The Probabilistic Method*. Wiley-Interscience, third edition, 2008. 3, 4, 30

[14] ANDRIS AMBAINIS: Quantum lower bounds by quantum arguments. *J. Comput. System Sci.*, 64(4):750–767, 2002. Earlier versions in STOC '00 and arxiv:quant-ph/0002066. [doi:10.1006/jcss.2002.1826] 35

[15] ANDRIS AMBAINIS: Polynomial degree vs. quantum query complexity. *J. Comput. System Sci.*, 72(2):220–238, 2006. Earlier versions in FOCS '03 and arxiv:quant-ph/0305028. [doi:10.1016/j.jcss.2005.06.006] 30, 35

[16] ANDRIS AMBAINIS, A. CHILDS, BEN W. REICHARDT, ROBERT ŠPALEK, AND S. ZHANG: Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. *SIAM J. Comput.*, 39(6):2513–2530, 2010. Earlier version in FOCS '07. [doi:10.1137/080712167] 22

[17] ANDRIS AMBAINIS, ASHWIN NAYAK, AMNON TA-SHMA, AND UMESH VAZIRANI: Dense quantum coding and quantum finite automata. *J. ACM*, 49(4):496–511, 2002. Earlier version in STOC '99. [doi:10.1145/581771.581773] 11

[18] LÁSZLÓ BABAI AND PÉTER FRANKL: *Linear Algebra Methods in Combinatorics, with Applications to Geometry and Computer Science*. University of Chicago, 1992. Unpublished manuscript, available from `http://www.cs.uchicago.edu/research/publications/combinatorics`. 5

[19] LÁSZLÓ BABAI, PÉTER FRANKL, AND JANOS SIMON: Complexity classes in communication complexity theory. In *Proc. 27th FOCS*, pp. 337–347. IEEE Comp. Soc. Press, 1986. [doi:10.1109/SFCS.1986.15] 32

[20] W. BANASZCZYK: New bounds in some transference theorems in the geometry of numbers. *Math. Ann.*, 296(4):625–635, 1993. [doi:10.1007/BF01445125] 38

[21] ZIV BAR-YOSSEF, T. S. JAYRAM, RAVI KUMAR, AND D. SIVAKUMAR: Information theory methods in communication complexity. In *Proc. 17th IEEE Conf. Comput. Complex., 2002*, pp. 93–102, 2002. [doi:10.1109/CCC.2002.1004344] 15

[22] ZIV BAR-YOSSEF, T. S. JAYRAM, RAVI KUMAR, AND D. SIVAKUMAR: An information statistics approach to data stream and communication complexity. *J. Comput. System Sci.*, 68(4):702–732, 2004. Earlier version in FOCS '02. [doi:10.1016/j.jcss.2003.11.006] 15

[23] ROBERT BEALS, HARRY BUHRMAN, RICHARD CLEVE, MICHELE MOSCA, AND RONALD DE WOLF: Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. Earlier versions in FOCS '98 and arxiv:quant-ph/9802049. [doi:10.1145/502090.502097] 13, 14

[24] RICHARD BEIGEL: Perceptrons, PP, and the polynomial hierarchy. *Comput. Complexity*, 4:339–349, 1994. [doi:10.1007/BF01263422] 33

[25] RICHARD BEIGEL, NICK REINGOLD, AND DANIEL SPIELMAN: PP is closed under intersection. *J. Comput. System Sci.*, 50(2):191–202, 1995. [doi:10.1006/jcss.1995.1017] 27, 29

[26] AVRAHAM BEN-AROYA, ODED REGEV, AND RONALD DE WOLF: A hypercontractive inequality for matrix-valued functions with applications to quantum computing and LDCs. In *Proc. 49th FOCS*, pp. 477–486. IEEE Comp. Soc. Press, 2008. Earlier version in arxiv:quant-ph/0705.3806. [doi:10.1109/FOCS.2008.45] 18

[27] CHARLES H. BENNETT, ETHAN BERNSTEIN, GILLES BRASSARD, AND UMESH VAZIRANI: Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. Earlier version in arxiv:quant-ph/9701001. [doi:10.1137/S0097539796300933] 35

[28] S. N. BERNSTEIN: Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités. *Communications de la Société Mathematique de Kharkov*, 13:1–2, 1912. 30

[29] GILLES BRASSARD, P. HØYER, MICHELE MOSCA, AND ALAIN TAPP: Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pp. 53–74. American Mathematical Society, 2002. Earlier version in arxiv:quant-ph/0005055. 12, 13, 31

[30] HARRY BUHRMAN, RICHARD CLEVE, RONALD DE WOLF, AND CHRISTOF ZALKA: Bounds for small-error and zero-error quantum algorithms. In *Proc. 40th FOCS*, pp. 358–368. IEEE Comp. Soc. Press, 1999. Earlier version in arxiv:cs.CC/9904019. [doi:10.1109/SFFCS.1999.814607] 12

[31] HARRY BUHRMAN AND RONALD DE WOLF: Complexity measures and decision tree complexity: A survey. *Theoret. Comput. Sci.*, 288(1):21–43, 2002. [doi:10.1016/S0304-3975(01)00144-X] 11

[32] HARRY BUHRMAN, P. B. MILTERSEN, JAIKUMAR RADHAKRISHNAN, AND S. VENKATESH: Are bitvectors optimal? *SIAM J. Comput.*, 31(6):1723–1744, 2002. Earlier version in STOC '00. [doi:10.1137/S0097539702405292] 41

[33] HARRY BUHRMAN, ILAN NEWMAN, HEIN RÖHRIG, AND RONALD DE WOLF: Robust polynomials and quantum algorithms. *Theory Comput. Syst.*, 40(4):379–395, 2007. Special issue on STACS 2005. Earlier version in arxiv:quant-ph/0309220. [doi:10.1007/s00224-006-1313-z] 24, 25

[34] HARRY BUHRMAN, NIKOLAY VERESHCHAGIN, AND RONALD DE WOLF: On computation and communication with small bias. In *Proc. 22nd IEEE Conf. Comput. Complex.*, pp. 24–32, 2007. [doi:10.1109/CCC.2007.18] 32, 33

[35] AMIT CHAKRABARTI AND ODED REGEV: An optimal randomised cell probe lower bound for approximate nearest neighbour searching. In *Proc. 45th FOCS*, pp. 473–482. IEEE Comp. Soc. Press, 2004. [doi:10.1137/080729955] 41

[36] AMIT CHAKRABARTI, YAOYUN SHI, ANTHONY WIRTH, AND ANDREW CHI-CHIH YAO: Informational complexity and the direct sum problem for simultaneous message complexity. In *Proc. 42nd FOCS*, pp. 270–278. IEEE Comp. Soc. Press, 2001. [doi:10.1109/SFCS.2001.959901] 15

[37] BENNY CHOR, ODED GOLDREICH, EYAL KUSHILEVITZ, AND MADHU SUDAN: Private information retrieval. *J. ACM*, 45(6):965–981, 1998. Earlier version in FOCS '95. [doi:10.1145/293347.293350] 16

[38] MATTHIAS CHRISTANDL: A quantum information-theoretic proof of the relation between Horn's problem and the Littlewood-Richardson coefficients. In *Proc. 4th Conf. Computability in Europe (CiE)*, volume 5028 of *Lecture Notes in Comput. Sci.*, pp. 120–128, 2008. [doi:10.1145/293347.293350] 41

[39] RICHARD CLEVE, WIM VAN DAM, MICHAEL NIELSEN, AND ALAIN TAPP: Quantum entanglement and the communication complexity of the inner product function. In *Proc. 1st NASA Intern. Conf. Quantum Comput. Quantum Comm.*, volume 1509 of *Lecture Notes in Comput. Sci.*, pp. 61–74. Springer, 1998. Earlier version in arxiv:quant-ph/9708019. [doi:10.1007/3-540-49208-9_4] 10, 15

[40] THOMAS M. COVER AND JOY A. THOMAS: *Elements of Information Theory*. Wiley, 1991. 3

[41] MART DE GRAAF AND RONALD DE WOLF: On quantum versions of the Yao principle. In *Proc. 19th Ann. Symp. Theor. Aspects of Comput. Sci. (STACS)*, volume 2285 of *Lecture Notes in Comput. Sci.*, pp. 347–358. Springer, 2002. Earlier version in arxiv:quant-ph/0109070. [doi:10.1007/3-540-45841-7_28] 13

[42] RONALD DE WOLF: Quantum communication and complexity. *Theoret. Comput. Sci.*, 287(1):337–353, 2002. [doi:10.1016/S0304-3975(02)00377-8] 15

[43] RONALD DE WOLF: Lower bounds on matrix rigidity via a quantum argument. In *Proc. 33rd Intern. Colloq. Autom. Lang. Program. (ICALP)*, volume 4051 of *Lecture Notes in Comput. Sci.*, pp. 62–71, 2006. Earlier version in arxiv:quant-ph/0505188. [doi:10.1007/11786986_7] 18, 19, 21

[44] RONALD DE WOLF: A note on quantum algorithms and the minimal degree of ε-error polynomials for symmetric functions. *Quantum Inf. Comput.*, 8(10):943–950, 2008. Earlier version in arxiv:quant-ph/0802.1816. 22

[45] RONALD DE WOLF: Rational degree equals quantum query complexity with postselection. Unpublished note, Oct 2008. 30

[46] ANDREW DRUCKER AND RONALD DE WOLF: Uniform approximation by (quantum) polynomials. *Quantum Information and Computation*, (3&4):215–225, 2011. `http://www.rintonpress.com/journals/qiconline.html#v11n34`. Earlier version in arxiv:1008.1599. 31

[47] KLIM EFREMENKO: 3-query locally decodable codes of subexponential length. In *Proc. 41st STOC*, pp. 39–44. ACM Press, 2009. [doi:10.1145/1536414.1536422] 16

[48] H. EHLICH AND K. ZELLER: Schwankung von Polynomen zwischen Gitterpunkten. *Math. Z.*, 86:41–44, 1964. 33

[49] A. EINSTEIN, B. PODOLSKY, AND N. ROSEN: Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, 1935. 7

[50] EDWARD FARHI, JEFFREY GOLDSTONE, AND SAM GUTMANN: A quantum algorithm for the Hamiltonian NAND tree. *Theory of Computing*, 4(1):169–190, 2008. Earlier version in arxiv:quant-ph/0702144. [doi:10.4086/toc.2008.v004a008] 22

[51] URIEL FEIGE, PRABHAKAR RAGHAVAN, DAVID PELEG, AND ELI UPFAL: Computing with noisy information. *SIAM J. Comput.*, 23(5):1001–1018, 1994. Earlier version in STOC '90. [doi:10.1137/S0097539791195877] 25

[52] LANCE FORTNOW AND JOSHUA A. GROCHOW: Complexity classes of equivalence problems revisited, 2009. Earlier version in arXiv:0907.4775. 41

[53] LANCE FORTNOW AND NICK REINGOLD: PP is closed under truth-table reductions. *Inform. and Comput.*, 124(1):1–6, 1996. [doi:10.1137/S0097539791195877] 27

[54] LANCE FORTNOW AND JOHN ROGERS: Complexity limitations on quantum computation. *J. Comput. System Sci.*, 59(2):240–252, 1999. Earlier version in Proc. 13th IEEE Conf. Comput. Complex., 1998 and arxiv:cs.CC/9811023. [doi:10.1006/jcss.1999.1651] 13

[55] NICOLAS GISIN, RENATO RENNER, AND STEFAN WOLF: Linking classical and quantum key agreement: Is there a classical analog to bound entanglement? *Algorithmica*, 34(4):389–412, 2002. Earlier version in Crypto '00. [doi:10.1007/s00453-002-0972-7] 41

[56] ODED GOLDREICH AND SHAFI GOLDWASSER: On the limits of nonapproximability of lattice problems. *J. Comput. System Sci.*, 60(3):540–563, 2000. Earlier version in STOC '98. [doi:10.1137/S0097539702405292] 38

[57] ODED GOLDREICH, HOWARD KARLOFF, LEONARD J. SCHULMAN, AND LUCA TREVISAN: Lower bounds for linear locally decodable codes and private information retrieval. *Comput. Complexity*, 15(3):263–296, 2006. Earlier version in Proc. 17th IEEE Conf. Comput. Complex., 2002. [doi:10.1007/s00037-006-0216-3] 17

[58] ODED GOLDREICH, DANIELE MICCIANCIO, S. SAFRA, AND J-P. SEIFERT: Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inform. Process. Lett.*, 71(2):55–61, 1999. [doi:10.1016/S0020-0190(99)00083-6] 37

[59] LOV K. GROVER: A fast quantum mechanical algorithm for database search. In *Proc. 28th STOC*, pp. 212–219. ACM Press, 1996. Earlier version in arxiv:quant-ph/9605043. [doi:10.1145/237814.237866] 12

[60] JOHAN HÅSTAD: The shrinkage exponent of de Morgan formulas is 2. *SIAM J. Comput.*, 27(1):48–64, 1998. [doi:10.1137/S0097539794261556] 41

[61] ISHAY HAVIV AND ODED REGEV: Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proc. 39th STOC*, pp. 469–477. ACM Press, 2007. [doi:10.1145/1250790.1250859] 37

[62] A. S. HOLEVO: Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973. English translation in *Problems Inform. Transmission*, 9:177–183, 1973. `http://mi.mathnet.ru/eng/ppi903`. 10

[63] P. HØYER, TROY LEE, AND ROBERT ŠPALEK: Negative weights make adversaries stronger. In *Proc. 39th STOC*, pp. 526–535. ACM Press, 2007. Earlier version in arxiv:quant-ph/0611054. [doi:10.1145/1250790.1250867] 35

[64] P. HØYER, MICHELE MOSCA, AND RONALD DE WOLF: Quantum search on bounded-error inputs. In *Proc. 30th Intern. Colloq. Autom. Lang. Program. (ICALP)*, volume 2719 of *Lecture Notes in Comput. Sci.*, pp. 291–299. Springer, 2003. Earlier version in arxiv:quant-ph/0304052. [doi:10.1007/3-540-45061-0_25] 13

[65] KAZUO IWAMA, HARUMICHI NISHIMURA, RUDY RAYMOND, AND SHIGERU YAMASHITA: Unbounded-error classical and quantum communication complexity. In *Proc. 18th Intern. Symp. Algorithms Comput. (ISAAC)*, volume 4835 of *Lecture Notes in Comput. Sci.*, pp. 100–111. Springer, 2007. Earlier version in arXiv:0709.2761. [doi:10.1007/978-3-540-77120-3_11] 33

[66] D. JACKSON: *Über die Genauigkeit der Annäherung stetiger Funktionen durch ganze rationale Funktionen gegebenen Grades und trigonometrische Summen gegebener Ordnung*. PhD thesis, University of Göttingen, 1911. 30

[67] T. S. JAYRAM, S. KOPPARTY, AND P. RAGHAVENDRA: On the communication complexity of read-once AC^0 formulae. In *Proc. 24th IEEE Conf. Comput. Complex.*, pp. 329–340, 2009. [doi:10.1109/CCC.2009.39] 15

[68] T. S. JAYRAM, RAVI KUMAR, AND D. SIVAKUMAR: Two applications of information complexity. In *Proc. 35th STOC*, pp. 673–682. ACM Press, 2003. [doi:10.1145/780542.780640] 15

[69] III JOHN T. GILL: *Probabilistic Turing Machines and Complexity of Computation*. PhD thesis, UC Berkeley, Berkeley, California, 1972. 26

[70] STASYS JUKNA: *Extremal Combinatorics, With Applications in Computer Science*. EATCS Series. Springer, 2001. 5, 44

[71] B. KASHIN AND ALEXANDER A. RAZBOROV: Improved lower bounds on the rigidity of Hadamard matrices. *Mat. Zametki*, 63(4):535–540, 1998. In Russian. 19, 21

[72] JONATHAN KATZ AND LUCA TREVISAN: On the efficiency of local decoding procedures for error-correcting codes. In *Proc. 32nd STOC*, pp. 80–86. ACM Press, 2000. [doi:10.1145/335305.335315] 16, 17

[73] IORDANIS KERENIDIS: Quantum multiparty communication complexity and circuit lower bounds. In *Proc. 4th Ann. Conf. Theory and Appl. Models of Comput. (TAMC)*, volume 4484 of *Lecture Notes in Comput. Sci.*, pp. 306–317. Springer, 2007. Earlier version in arxiv:quant-ph/0504087. [doi:10.1007/978-3-540-72504-6_28] 41

[74] IORDANIS KERENIDIS AND RONALD DE WOLF: Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. Comput. System Sci.*, 69(3):395–420, 2004. Earlier versions in STOC '03 and arxiv:quant-ph/0208062. [doi:10.1016/j.jcss.2004.04.007] 11, 17, 18

[75] SUBHASH KHOT: Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 52(5):789–808, 2005. Earlier version in FOCS '04. [doi:10.1145/1089023.1089027] 37

[76] HARTMUT KLAUCK: Lower bounds for quantum communication complexity. *SIAM J. Comput.*, 37(1):20–46, 2007. Earlier version in FOCS '01 and arxiv:quant-ph/0106160. [doi:10.1137/S0097539702405620] 32, 33

[77] HARTMUT KLAUCK, ROBERT ŠPALEK, AND RONALD DE WOLF: Quantum and classical strong direct product theorems and optimal time-space tradeoffs. *SIAM J. Comput.*, 36(5):1472–1493, 2007. Earlier version in FOCS '04 and arxiv:quant-ph/0402123. [doi:10.1137/05063235X] 14, 32

[78] GREG KUPERBERG: Random words, quantum statistics, central limits, random matrices. *Methods Appl. Anal.*, 9(1):99–118, 2002. http://www.intlpress.com/MAA/p/2002/09_1/MAA-09-1-099-118.pdf. Earlier version in arxiv:math/9909104. 42

[79] GREG KUPERBERG: A tracial quantum central limit theorem. *Trans. Amer. Math. Soc.*, 357:459–471, 2005. Earlier version in arxiv:math-ph/0202035. [doi:10.1090/S0002-9947-03-03449-4] 42

[80] EYAL KUSHILEVITZ AND NOAM NISAN: *Communication Complexity*. Cambridge University Press, 1997. 14

[81] J. C. LAGARIAS, H. W. LENSTRA, JR., AND C.-P. SCHNORR: Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333–348, 1990. [doi:10.1007/BF02128669] 38

[82] SOPHIE LAPLANTE, TROY LEE, AND MARIO SZEGEDY: The quantum adversary method and classical formula size lower bounds. *Comput. Complexity*, 15(2):163–196, 2006. Earlier version in Proc. 20th IEEE Conf. Comput. Complex., 2005. [doi:10.1007/s00037-006-0212-7] 41

[83] SOPHIE LAPLANTE AND FREDERIC MAGNIEZ: Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. In *Proc. 19th IEEE Conf. Comput. Complex.*, pp. 294–304, 2004. Earlier version in arxiv:quant-ph/0311189. [doi:10.1109/CCC.2004.1313852] 37

[84] TROY LEE: A note on the sign degree of formulas, 2009. [arXiv:0909.4607] 22

[85] TROY LEE, GIDEON SCHECHTMAN, AND ADI SHRAIBMAN: Lower bounds on quantum multiparty communication complexity. In *Proc. 24th IEEE Conf. Comput. Complex.*, pp. 254–262, 2009. [doi:10.1109/CCC.2009.24] 41

[86] A. K. LENSTRA, H. W. LENSTRA, JR., AND L. LOVÁSZ: Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982. 37

[87] NIKOS LEONARDOS AND MICHAEL SAKS: Lower bounds on the randomized communication complexity of read-once functions. *Comput. Complexity*, 19(2):153–181, 2010. [doi:10.1007/s00037-010-0292-2] 15

[88] MING LI AND PAUL VITÁNYI: *An Introduction to Kolmogorov Complexity and its Applications*. Springer, Berlin, third edition, 2008. 4

[89] NATI LINIAL AND ADI SHRAIBMAN: Learning complexity vs communication complexity. *Combin. Probab. Comput.*, 18(1-2):227–245, 2009. Earlier version in Proc. 22nd IEEE Conf. Computat. Complex., 2008. [doi:10.1017/S0963548308009656] 32

[90] NATI LINIAL AND ADI SHRAIBMAN: Lower bounds in communication complexity based on factorization norms. *Random Structures Algorithms*, 34(3):368–394, 2009. Earlier version in STOC '07. [doi:10.1002/rsa.20232] 32

[91] RICHARD LIPTON: Erdős and the quantum method, March 28, 2009. Blog entry http://rjlipton.wordpress.com/2009/03/28/erds-and-the-quantum-method/. 43

[92] SATYANARAYANA V. LOKAM: Spectral methods for matrix rigidity with applications to size-depth trade-offs and communication complexity. *J. Comput. System Sci.*, 63(3):449–473, 2001. Earlier version in FOCS ’95. [doi:10.1006/jcss.2001.1786] 19, 21

[93] SATYANARAYANA V. LOKAM: Quadratic lower bounds on matrix rigidity. In *Proc. 3rd Ann. Conf. Theory and Appl. Models of Comput. (TAMC)*, volume 3959 of *Lecture Notes in Comput. Sci.*, pp. 295–307. Springer, 2006. [doi:10.1007/11750321_28] 18

[94] SATYANARAYANA V. LOKAM: Complexity lower bounds using linear algebra. *Found. Trends Theor. Comput. Sci.*, 4(1–2):1–155, 2008. [doi:10.1561/0400000011] 19

[95] DANIELE MICCIANCIO: Almost perfect lattices, the covering radius problem, and applications to Ajtai’s connection factor. *SIAM J. Comput.*, 34(1):118–169, 2004. [doi:10.1137/S0097539703433511] 37

[96] DANIELE MICCIANCIO AND SHAFI GOLDWASSER: *Complexity of Lattice Problems: A cryptographic perspective*. Volume 671 of *Kluwer Internat. Ser. Engrg. Comput. Sci.* Kluwer Academic Publishers, Boston, Massachusetts, 2002. 37

[97] DANIELE MICCIANCIO AND ODED REGEV: Lattice-based cryptography. In D. J. BERNSTEIN, J. BUCHMANN, AND E. DAHMEN, editors, *Post-quantum Cryptography*, pp. 147–191. Springer, 2009. [doi:10.1007/978-3-540-88702-7_5] 37

[98] GATIS MIDRIJANIS: Three lines proof of the lower bound for the matrix rigidity. Technical report, Arxiv e-print, June 2005. [arXiv:cs.CC/0506081] 21

[99] MARVIN MINSKY AND SEYMOUR PAPERT: *Perceptrons*. MIT Press, Cambridge, MA, 1968. Second, expanded edition 1988. 26

[100] ASHWIN NAYAK: Optimal lower bounds for quantum automata and random access codes. In *Proc. 40th FOCS*, pp. 369–376. IEEE Comp. Soc. Press, 1999. Earlier version in arxiv:quant-ph/9904093. [doi:10.1109/SFFCS.1999.814608] 10, 11

[101] ASHWIN NAYAK AND JULIA SALZMAN: Limits on the ability of quantum states to convey classical messages. *J. ACM*, 53(1):184–206, 2006. Earlier version in STOC ’02. [doi:10.1145/1120582.1120587] 15

[102] MICHAEL NIELSEN AND ISAAC L. CHUANG: *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 27, 35, 43

[103] RYAN O’DONNELL AND ROCCO SERVEDIO: New degree bounds for polynomial threshold functions. In *Proc. 35th STOC*, pp. 325–334. ACM Press, 2003. [doi:10.1145/780542.780592] 22

[104] RAMAMOHAN PATURI: On the degree of polynomials that approximate symmetric Boolean functions. In *Proc. 24th STOC*, pp. 468–474. ACM Press, 1992. [doi:10.1145/129712.129758] 22

[105] RAMAMOHAN PATURI AND JANOS SIMON: Probabilistic communication complexity. *J. Comput. System Sci.*, 33(1):106–123, 1986. Earlier version in FOCS '84. [doi:10.1016/0022-0000(86)90046-2] 32

[106] CHRIS PEIKERT: Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. 41st STOC*, pp. 333–342. ACM Press, 2009. [doi:10.1145/1536414.1536461] 37

[107] JAIKUMAR RADHAKRISHNAN, PRANAB SEN, AND S. VENKATESH: The quantum complexity of set membership. *Algorithmica*, 34(4):462–479, 2002. Earlier versions in FOCS '00 and arvix:quant-ph/0007021. [doi:10.1007/s00453-002-0979-0] 40

[108] ALEXANDER A. RAZBOROV: Quantum communication complexity of symmetric predicates. *Izvestiya of the Russian Academy of Sciences, Mathematics*, 67(1):145–159, 2003. Earlier version in arxiv:quant-ph/0204025. [doi:10.1070/IM2003v067n01ABEH000422] 32, 33

[109] ODED REGEV: New lattice based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004. Earlier version in STOC '03. [doi:10.1145/1039488.1039490] 37

[110] ODED REGEV: Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, 2004. Earlier version in FOCS '02. [doi:10.1137/S0097539703440678] 37

[111] ODED REGEV: On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009. Earlier version in STOC '05. [doi:10.1145/1568318.1568324] 37

[112] BEN W. REICHARDT: Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proc. 50th FOCS*, pp. 544–551. IEEE Comp. Soc. Press, 2009. [doi:10.1109/FOCS.2009.55] 22, 35

[113] BEN W. REICHARDT: Span programs and quantum query algorithms. Technical report, ECCC Report TR10–110, 2010. [ECCC:TR10-110] 24

[114] RENATO RENNER AND STEFAN WOLF: New bounds in secret-key agreement: The gap between formation and secrecy extraction. In *Proc. Advances in Cryptology — Eurocrypt*, volume 2656 of *Lecture Notes in Comput. Sci.*, pp. 562–577. Springer, 2003. [doi:10.1007/3-540-39200-9_35] 41

[115] THEODORE J. RIVLIN: *An Introduction to the Approximation of Functions*. Blaisdell Publishing Company, 1969. 30

[116] THEODORE J. RIVLIN AND E. W. CHENEY: A comparison of uniform approximations on an interval and a finite subset thereof. *SIAM J. Numer. Anal.*, 3(2):311–320, 1966. 33

[117] PRANAB SEN AND S. VENKATESH: Lower bounds for predecessor searching in the cell probe model. *J. Comput. System Sci.*, 74(3):364–385, 2008. [doi:10.1016/j.jcss.2007.06.016] 41

[118] ALEXANDER A. SHERSTOV: Approximate inclusion-exclusion for arbitrary symmetric functions. In *Proc. 23rd IEEE Conf. Comput. Complex.*, pp. 112–123, 2008. [doi:10.1109/CCC.2008.18] 22

[119] ALEXANDER A. SHERSTOV: Halfspace matrices. *Comput. Complexity*, 17(2):149–178, 2008. [doi:10.1007/s00037-008-0242-4] 32

[120] ALEXANDER A. SHERSTOV: The pattern matrix method. *SIAM J. Comput.*, 2010. To appear. Earlier version in STOC '08. 32

[121] PETER W. SHOR: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. Earlier versions in FOCS '94 and arxiv:quant-ph/9508027. [doi:10.1137/S0097539795293172] 4

[122] DANIEL R. SIMON: On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997. Earlier version in FOCS '94. [doi:10.1137/S0097539796298637] 41

[123] LUCA TREVISAN: Some applications of coding theory in computational complexity. *Quad. Mat.*, 13:347–424, 2004. 16

[124] KOJI TSUDA, GUNNAR RÄTSCH, AND MANFRED WARMUTH: Matrix exponentiated gradient updates for on-line learning and Bregman projection. *J. Mach. Learn. Res.*, 6:995–1018, 2005. http://www.jmlr.org/papers/volume6/tsuda05a/tsuda05a.pdf. 42

[125] LESLIE G. VALIANT: Graph-theoretic arguments in low-level complexity. In *Proc. 6th Conf. Math. Found. Comput. Sci. (MFCS)*, volume 53 of *Lecture Notes in Comput. Sci.*, pp. 162–176. Springer, 1977. [doi:10.1007/3-540-08353-7_135] 18

[126] STEPHANIE WEHNER AND RONALD DE WOLF: Improved lower bounds for locally decodable codes and private information retrieval. In *Proc. 32nd Intern. Colloq. Autom. Lang. Program. (ICALP)*, volume 3580 of *Lecture Notes in Comput. Sci.*, pp. 1424–1436, 2005. Earlier version in arxiv:quant-ph/0403140. [doi:10.1007/11523468_115] 17

[127] KARL WEIERSTRASS: Über die analytische Darstellbarkeit sogenannter willkürlicher Funktionen reeller Argumente. In *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin, II*, volume 3. 1885. 30

[128] DAVID P. WOODRUFF: New lower bounds for general locally decodable codes. Technical report, ECCC Report TR07–006, 2007. [ECCC:TR07-006] 17

[129] DAVID P. WOODRUFF: A quadratic lower bound for three-query linear locally decodable codes over any field. In *Proc. 13th Intern. Workshop Approx. Randomization Comb. Optim. (APPROX)*, Lecture Notes in Comput. Sci., pp. 766–779. Springer, 2010. [doi:10.1007/978-3-642-15369-3_57] 17

[130] ANDREW CHI-CHIH YAO: Some complexity questions related to distributive computing. In *Proc. 11th STOC*, pp. 209–213. ACM Press, 1979. [doi:10.1145/800135.804414] 14

[131] SERGEY YEKHANIN: Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1), 2008. Earlier version in STOC '07. [doi:10.1145/1326554.1326555] 16

AUTHORS

Andrew Drucker
graduate student
MIT, Cambridge, MA
adrucker@mit.edu
http://people.csail.mit.edu/andyd/home.html

Ronald de Wolf
senior researcher
CWI, Amsterdam, Netherlands
rdewolf@cwi.nl
http://homepages.cwi.nl/~rdewolf/

ABOUT THE AUTHORS

ANDREW DRUCKER is a Ph. D. student in computer science at M.I.T., advised by Scott Aaronson. He has broad interests within complexity theory, and also enjoys running, live jazz, and the game of Go.

RONALD DE WOLF graduated from the University of Amsterdam and C.W.I. in 2001. His advisors were Harry Buhrman and Paul Vitányi. His CS interests include quantum computing, complexity theory, and learning theory. More details can be found at his homepage. He also holds a degree in philosophy, and enjoys classical music and literature.