Quantum speedup of Monte Carlo methods

Ashley Montanaro*

July 11, 2017

Abstract

Monte Carlo methods use random sampling to estimate numerical quantities which are hard to compute deterministically. One important example is the use in statistical physics of rapidly mixing Markov chains to approximately compute partition functions. In this work we describe a quantum algorithm which can accelerate Monte Carlo methods in a very general setting. The algorithm estimates the expected output value of an arbitrary randomised or quantum subroutine with bounded variance, achieving a near-quadratic speedup over the best possible classical algorithm. Combining the algorithm with the use of quantum walks gives a quantum speedup of the fastest known classical algorithms with rigorous performance bounds for computing partition functions, which use multiple-stage Markov chain Monte Carlo techniques. The quantum algorithm can also be used to estimate the total variation distance between probability distributions efficiently.

1 Introduction

Monte Carlo methods are now ubiquitous throughout science, in fields as diverse as statistical physics [37], microelectronics [30] and mathematical finance [23]. These methods use randomness to estimate numerical properties of systems which are too large or complicated to analyse deterministically. In general, the basic core of Monte Carlo methods involves estimating the expected output value μ of a randomised algorithm $\mathcal{A}$. The natural algorithm for doing so is to produce k samples, each corresponding to the output of an independent execution of $\mathcal{A}$, and then to output the average $\widetilde{\mu}$ of the samples as an approximation of μ. Assuming that the variance of the random variable corresponding to the output of $\mathcal{A}$ is at most σ^2, the probability that the value output by this estimator is far from the truth can be bounded using Chebyshev's inequality:

$$\Pr[|\widetilde{\mu} - \mu| \geq \epsilon] \leq \frac{\sigma^2}{k\epsilon^2}.$$

It is therefore sufficient to take $k = O(\sigma^2/\epsilon^2)$ to estimate μ up to additive error ϵ with, say, 99% success probability. This simple result is a key component in many more complex randomised approximation schemes (see e.g. [50, 37]).

Although this algorithm is fairly efficient, its quadratic dependence on σ/ϵ seems far from ideal: for example, if $\sigma = 1$, to estimate μ up to 4 decimal places we would need to run $\mathcal{A}$ over 100 million times. Unfortunately, it can be shown that, without any further information about $\mathcal{A}$, the sample complexity of this algorithm is asymptotically optimal [15] with respect to its scaling with σ and ϵ, although it can be improved by a constant factor [29].

[*] Department of Computer Science, University of Bristol, UK; ashley.montanaro@bristol.ac.uk.

We show here that, using a quantum computer, the number of uses of $\mathcal{A}$ required to approximate μ can be reduced almost quadratically beyond the above classical bound. Assuming that the variance of the output of the algorithm $\mathcal{A}$ is at most σ^2, we present a quantum algorithm which estimates μ up to additive error ϵ, with 99% success probability, using $\mathcal{A}$ only $\widetilde{O}(\sigma/\epsilon)$ times[1]. It follows from known lower bounds on the quantum complexity of approximating the mean [45] that the runtime of this algorithm is optimal, up to polylogarithmic factors. This result holds for an *arbitrary* algorithm $\mathcal{A}$ used as a black box, given only an upper bound on the variance.

An important aspect of this construction is that the underlying subroutine $\mathcal{A}$ need not be a classical randomised procedure, but can itself be a quantum algorithm. This enables any quantum speedup obtained by $\mathcal{A}$ to be utilised within the overall framework of the algorithm. A particular case in which this is useful is quantum speedup of Markov chain Monte Carlo methods [38]. Classically, such methods use a rapidly mixing Markov chain to approximately sample from a probability distribution corresponding to the stationary distribution of the chain. Quantum walks are the quantum analogue of random walks (see e.g. [57] for a review). In some cases, quantum walks can reduce the mixing time quadratically (see e.g. [3, 58]), although it is not known whether this can be achieved in general [48, 6, 18]. We demonstrate that this known quadratic reduction can be combined with our algorithm to speed up the fastest known general-purpose classical algorithm with rigorous performance bounds [50] for approximately computing partition functions up to small relative error, a fundamental problem in statistical physics [37]. As another example of how our algorithm can be applied, we substantially improve the runtime of a quantum algorithm for estimating the total variation distance between two probability distributions [13].

1.1 Prior work

The topic of quantum estimation of mean output values of algorithms with bounded variance connects to several previously-explored directions. First, it generalises the problem of approximating the mean, with respect to the uniform distribution, of an arbitrary bounded function. This has been addressed by a number of authors. The first asymptotically optimal quantum algorithm for this problem, which uses $O(1/\epsilon)$ queries to achieve additive error ϵ, seems to have been given by Heinrich [27]; an elegant alternative optimal algorithm was later presented by Brassard et al. [11]. Previous algorithms, which are optimal up to lower-order terms, were described by Grover [25], Aharonov [2] and Abrams and Williams [1]. Using similar techniques to Brassard et al., Wocjan et al. [59] described an efficient algorithm for estimating the expected value of an arbitrary bounded observable. It is not difficult to combine these ideas to approximate the mean of arbitrary bounded functions with respect to nonuniform distributions (see Section 2.1).

One of the main technical ingredients in the present paper is based on an algorithm of Heinrich for approximating the mean, with respect to the uniform distribution, of functions with bounded L^2 norm [27]. Section 2.2 describes a generalisation of this result to nonuniform distributions, using similar techniques. This is roughly analogous to the way that amplitude amplification [12] generalises Grover's quantum search algorithm [24].

The related problem of quantum estimation of expectation values of observables, an important task in the simulation of quantum systems, has been studied by Knill, Ortiz and Somma [36]. These authors give an algorithm for estimating $\mathrm{tr}(A\rho)$ for observables A such that one can efficiently implement the operator e^{-iAt}. The algorithm is efficient (i.e. achieves runtimes close to $O(1/\epsilon)$) when the tails of the distribution $\mathrm{tr}(A\rho)$ decay quickly. However, in the case where one only knows

[1] The $\widetilde{O}$ notation hides polylogarithmic factors.

Algorithm	Precondition	Approximation of μ	Uses of $\mathcal{A}$ and $\mathcal{A}^{-1}$
1	$v(\mathcal{A}) \in [0,1]$	Additive error ϵ	$O(1/\epsilon)$
3	$\mathrm{Var}(v(\mathcal{A})) \leq \sigma^2$	Additive error ϵ	$\widetilde{O}(\sigma/\epsilon)$
4	$\mathrm{Var}(v(\mathcal{A}))/(\mathbb{E}[v(\mathcal{A})])^2 \leq B$	Relative error ϵ	$\widetilde{O}(B/\epsilon)$

Table 1: Summary of the main quantum algorithms presented in this paper for estimating the mean output value μ of an algorithm $\mathcal{A}$. (Algorithm 2, omitted, is a subroutine used in Algorithm 3.)

an upper bound on the variance of this distribution, the algorithm does not achieve a better runtime than classical sampling. Yet another related problem, that of exact Monte Carlo *sampling* from a desired probability distribution, was addressed by Destainville, Georgeot and Giraud [17]. Their quantum algorithm, which uses Grover's algorithm as a subroutine, achieves roughly a quadratic speedup over classical exact sampling. This algorithm's applicability is limited by the fact that its runtime scaling can be as slow as $O(\sqrt{N})$, where N is the number of states of the system; we often think of N as being exponential in the input size.

Quantum algorithms have been used previously to approximate classical partition functions and solve related problems. In particular, a number of authors [40, 39, 4, 56, 21, 7, 22, 16, 43] have considered the complexity of computing Ising and Potts model partition functions. These works in some cases achieve exponential quantum speedups over the best known classical algorithms. Unfortunately, they in general either produce an approximation accurate up to a specified *additive* error bound, or only work for specific classes of partition function problems with restrictions on interaction strengths and topologies, or both. Here we aim to approximate partition functions up to small relative error in a rather general setting.

Using related techniques to the present work, Somma et al. [49] used quantum walks to accelerate classical simulated annealing processes, and quantum estimation of partition functions up to small relative error was addressed by Wocjan et al. [59]. Their algorithm, which is based on the use of quantum walks and amplitude estimation, achieves a quadratic speedup over classical algorithms with respect to both mixing time and accuracy. However, it cannot be directly applied to accelerate the most efficient classical algorithms for approximating partition function problems, which use so-called Chebyshev cooling schedules (discussed in Section 3). This is essentially because these algorithms are based around estimating the mean of random variables given only a bound on the variance. This was highlighted as an open problem in [59], which we resolve here.

Several recent works have developed quantum algorithms for the quantum generalisation of sampling from a Gibbs distribution: producing a Gibbs state $\rho \propto e^{-\beta H}$ for some quantum Hamiltonian H [53, 47, 52, 60]. Given such a state, one can measure a suitable observable to compute some quantity of interest about H. Supplied with an upper bound on the variance of such an observable, the procedure detailed here can be used (as for any other quantum algorithm) to reduce the number of repetitions required to estimate the observable to a desired accuracy.

1.2 Techniques

We now give an informal description of our algorithms, which are summarised in Table 1 (for technical details and proofs, see Section 2). For any randomised or quantum algorithm $\mathcal{A}$, we write $v(\mathcal{A})$ for the random variable corresponding to the value computed by $\mathcal{A}$, with the expected value of $v(\mathcal{A})$ denoted $\mathbb{E}[v(\mathcal{A})]$. For concreteness, we think of $\mathcal{A}$ as a quantum algorithm which operates on n qubits, each initially in the state $|0\rangle$, and whose quantum part finishes with a measurement of k

of the qubits in the computational basis. Given that the measurement returns outcome $x \in \{0,1\}^k$, the final output is then $\phi(x)$, for some fixed function $\phi : \{0,1\}^k \to \mathbb{R}$. If $\mathcal{A}$ is a classical randomised algorithm, or a quantum circuit using (for example) mixed states and intermediate measurements, a corresponding unitary quantum circuit of this form can be produced using standard reversible-computation techniques [5]. As is common in works based on quantum amplitude amplification and estimation [12], we also assume that we have the ability to execute the algorithm $\mathcal{A}^{-1}$, which is the inverse of the unitary part of $\mathcal{A}$. If we do have a description of $\mathcal{A}$ as a quantum circuit, this can be achieved simply by running the circuit backwards, replacing each gate with its inverse.

We first deal with the special case where the output of $\mathcal{A}$ is bounded between 0 and 1. Here a quantum algorithm for approximating $\mu := \mathbb{E}[v(\mathcal{A})]$ quadratically faster than is possible classically can be found by combining ideas from previously known algorithms [27, 11, 59]. We append an additional qubit and define a unitary operator W on $k+1$ qubits which performs the map $|x\rangle|0\rangle \mapsto |x\rangle(\sqrt{1-\phi(x)}|0\rangle + \sqrt{\phi(x)}|1\rangle)$. If the final measurement of the algorithm $\mathcal{A}$ is replaced with performing W, then measuring the added qubit, the probability that we receive the answer 1 is precisely μ. Using quantum amplitude estimation [12] the probability that this measurement returns 1 can be estimated to higher accuracy than is possible classically. Using t iterations of amplitude estimation, we can output an estimate $\widetilde{\mu}$ such that $|\widetilde{\mu} - \mu| = O(\sqrt{\mu}/t + 1/t^2)$ with high probability [12]. In particular, $O(1/\epsilon)$ iterations of amplitude estimation are sufficient to produce an estimate $\widetilde{\mu}$ such that $|\widetilde{\mu} - \mu| \le \epsilon$ with, say, 99% probability.

The next step is to use the above algorithm as a subroutine in a more general procedure that can deal with algorithms $\mathcal{A}$ whose output is non-negative, has bounded ℓ_2 norm, but is not necessarily bounded between 0 and 1. That is, algorithms for which we can control the expression $\|v(\mathcal{A})\|_2 := \sqrt{\mathbb{E}[v(\mathcal{A})^2]}$. The procedure for this case generalises, and is based on the same ideas as, a previously known result for the uniform distribution [27].

The idea is to split the output of $\mathcal{A}$ up into disjoint intervals depending on size. Write $\mathcal{A}_{p,q}$ for the "truncated" algorithm which outputs $v(\mathcal{A})$ if $p \le v(\mathcal{A}) < q$, and otherwise outputs 0. We estimate μ by applying the above algorithm to estimate $\mathbb{E}[v(\mathcal{A}_{p,q})]$ for a sequence of $O(\log 1/\epsilon)$ intervals which are exponentially increasing in size, and summing the results. As the intervals $[p,q)$ get larger, the accuracy with which we approximate $\mathbb{E}[v(\mathcal{A}_{p,q})]$ decreases, and values $v(\mathcal{A})$ larger than about $1/\epsilon$ are ignored completely. However, the overall upper bound on $\|v(\mathcal{A})\|_2$ allows us to infer that these larger values do not affect the overall expectation μ much; indeed, if μ depended significantly on large values in the output, the ℓ_2 norm of $v(\mathcal{A})$ would be high.

The final result is that for $\|v(\mathcal{A})\|_2 = O(1)$, given appropriate parameter choices, the estimate $\widetilde{\mu}$ satisfies $|\widetilde{\mu} - \mu| = O(\epsilon)$ with high probability, and the algorithm uses $\mathcal{A}$ $\widetilde{O}(1/\epsilon)$ times in total. This scaling is a near-quadratic improvement over the best possible classical algorithm.

We next consider the more general case of algorithms $\mathcal{A}$ which have bounded variance, but whose output need not be non-negative, nor bounded in ℓ_2 norm. To apply the previous algorithm, we would like to transform the output of $\mathcal{A}$ to make its ℓ_2 norm low. If $v(\mathcal{A})$ has mean μ and variance upper-bounded by σ^2, a suitable way to achieve this is to subtract μ from the output of $\mathcal{A}$, then divide by σ. The new algorithm's output would have ℓ_2 norm upper-bounded by 1, and estimating its expected value up to additive error ϵ/σ would give us an estimate of μ up to ϵ. Unfortunately, we of course do not know μ initially, so cannot immediately implement this idea. To approximately implement it, we first run $\mathcal{A}$ once and use the output $\widetilde{m}$ as a proxy for μ. Because $\mathrm{Var}(v(\mathcal{A})) \le \sigma^2$, $\widetilde{m}$ is quite likely to be within distance $O(\sigma)$ of μ. Therefore, the algorithm $\mathcal{B}$ produced from $\mathcal{A}$ by subtracting $\widetilde{m}$ and dividing by σ is quite likely to have ℓ_2 norm upper-bounded by a constant. We can thus efficiently estimate the positive and negative parts of $\mathbb{E}[v(\mathcal{B})]$ separately, then combine and rescale them. The overall algorithm achieves accuracy ϵ in time $\widetilde{O}(\sigma/\epsilon)$.

A similar idea can be used to approximate the expected output value of algorithms for which we have a bound on the relative variance, namely that $\mathrm{Var}(v(\mathcal{A})) = O(\mu^2)$. In this setting it turns out that $\widetilde{O}(1/\epsilon)$ uses of $\mathcal{A}$ suffice to produce an estimate $\widetilde{\mu}$ accurate up to *relative* error ϵ, i.e. for which $|\widetilde{\mu} - \mu| \leq \epsilon\mu$. This is again a near-quadratic improvement over the best possible classical algorithm.

1.3 Approximating partition functions

In this section we discuss (with details in Section 3) how these algorithms can be applied to the problem of approximating partition functions. Consider a (classical) physical system which has state space Ω, together with a Hamiltonian $H : \Omega \to \mathbb{R}$ specifying the energy of each configuration[2] $x \in \Omega$. Here we will assume that H takes integer values in the set $\{0, \dots, n\}$. A central problem is to compute the partition function

$$Z(\beta) = \sum_{x \in \Omega} e^{-\beta\, H(x)}$$

for some inverse temperature β defined by $\beta = 1/(k_B T)$, where T is the temperature and k_B is Boltzmann's constant. As well as naturally encapsulating various models in statistical physics, such as the Ising and Potts models, this framework also encompasses well-studied problems in computer science, such as counting the number of valid k-colourings of a graph. In particular, $Z(\infty)$ counts the number of configurations x such that $H(x) = 0$. It is often hard to compute $Z(\beta)$ for large β but easy to approximate $Z(\beta) \approx |\Omega|$ for $\beta \approx 0$. In many cases, such as the Ising model, it is known that computing $Z(\infty)$ exactly falls into the #P-complete complexity class [34], and hence is unlikely to admit an efficient quantum or classical algorithm.

Here our goal will be to approximate $Z(\beta)$ up to relative error ϵ, for some small ϵ. That is, to output $\widetilde{Z}$ such that $|\widetilde{Z} - Z(\beta)| \leq \epsilon\, Z(\beta)$, with high probability. For simplicity, we will focus on $\beta = \infty$ in the following discussion, but it is easy to see how to generalise to arbitrary β.

Let $0 = \beta_0 < \beta_1 < \cdots < \beta_\ell = \infty$ be a sequence of inverse temperatures. A standard classical approach to design algorithms for approximating partition functions [55, 19, 10, 50, 59] is based around expressing $Z(\beta_\ell)$ as the telescoping product

$$Z(\beta_\ell) = Z(\beta_0) \frac{Z(\beta_1)}{Z(\beta_0)} \frac{Z(\beta_2)}{Z(\beta_1)} \cdots \frac{Z(\beta_\ell)}{Z(\beta_{\ell-1})}.$$

If we can compute $Z(\beta_0) = |\Omega|$, and can also approximate each of the ratios $\alpha_i := Z(\beta_{i+1})/Z(\beta_i)$ accurately, taking the product will give a good approximation to $Z(\beta_\ell)$. Let π_i denote the Gibbs (or Boltzmann) probability distribution corresponding to inverse temperature β_i, where

$$\pi_i(x) = \frac{1}{Z(\beta_i)} e^{-\beta_i H(x)}.$$

To approximate α_i we define the random variable

$$Y_i(x) = e^{-(\beta_{i+1} - \beta_i) H(x)}.$$

Then one can readily compute that $\mathbb{E}_{\pi_i}[Y_i] = \alpha_i$, so sampling from each distribution π_i allows us to estimate the quantities α_i. It will be possible to estimate α_i up to small relative error efficiently

[2] We use x to label configurations rather than the more standard σ to avoid confusion with the variance.

if the ratio $\mathbb{E}[Y_i^2]/\mathbb{E}[Y_i]^2$ is low. This motivates the concept of a *Chebyshev cooling schedule* [50]: a sequence of inverse temperatures β_i such that $\mathbb{E}[Y_i^2]/\mathbb{E}[Y_i]^2 = O(1)$ for all i. It is known that, for any partition function problem as defined above such that $|\Omega| = A$, there exists a Chebyshev cooling schedule with $\ell = \widetilde{O}(\sqrt{\log A})$ [50].

It is sufficient to approximate $\mathbb{E}[Y_i]$ up to relative error $O(\epsilon/\ell)$ for each i to get an overall approximation accurate up to relative error ϵ. To achieve this, the quantum algorithm presented here needs to use at most $\widetilde{O}(\ell/\epsilon)$ samples from Y_i. Given a Chebyshev cooling schedule with $\ell = \widetilde{O}(\sqrt{\log A})$, the algorithm thus uses $\widetilde{O}((\log A)/\epsilon)$ samples in total, a near-quadratic improvement in terms of ϵ over the complexity of the fastest known classical algorithm [50].

In general, we cannot exactly sample from the distributions π_i. Classically, one way of approximately sampling from these distributions is to use a Markov chain which mixes rapidly and has stationary distribution π_i. For a reversible, ergodic Markov chain, the time required to produce such a sample is controlled by the *relaxation time* $\tau := 1/(1 - |\lambda_1|)$ of the chain, where λ_1 is the second largest eigenvalue in absolute value [38]. In particular, sampling from a distribution close to π_i in total variation distance requires $\Omega(\tau)$ steps of the chain.

It has been known for some time that quantum walks can sometimes mix quadratically faster [3]. One case where efficient mixing can be obtained is for sequences of Markov chains whose stationary distributions π are close [58]. Further, for this special case one can approximately produce coherent "quantum sample" states $|\pi\rangle = \sum_{x\in\Omega} \sqrt{\pi(x)}|x\rangle$ efficiently. Here we can show (Section 3.2) that the Chebyshev cooling schedule condition implies that each distribution in the sequence $\pi_1, \dots, \pi_{\ell-1}$ is close enough to its predecessor that we can use techniques of [58] to approximately produce any state $|\pi_i\rangle$ using $\widetilde{O}(\ell\sqrt{\tau})$ quantum walk steps each. Using similar ideas we can approximately reflect about $|\pi_i\rangle$ using only $\widetilde{O}(\sqrt{\tau})$ quantum walk steps.

Approximating $\mathbb{E}[Y_i]$ up to relative error $O(\epsilon/\ell)$ using our algorithm requires one quantum sample approximating $|\pi_i\rangle$, and $\widetilde{O}(\ell/\epsilon)$ approximate reflections about $|\pi_i\rangle$. Therefore, the total number of quantum walk steps required for each i is $\widetilde{O}(\ell\sqrt{\tau}/\epsilon)$. Summing over i, we get a quantum algorithm for approximating an arbitrary partition function up to relative error ϵ using $\widetilde{O}((\log A)\sqrt{\tau}/\epsilon)$ quantum walk steps. The fastest known classical algorithm [50] exhibits quadratically worse dependence on both τ and ϵ.

In the above discussion, we have neglected the complexity of computing the Chebyshev cooling schedule itself. An efficient classical algorithm for this task is known [50], which runs in time $\widetilde{O}((\log A)\tau)$. Adding the complexity of this part, we finish with an overall complexity of $\widetilde{O}((\log A)\sqrt{\tau}(\sqrt{\tau} + 1/\epsilon))$. We leave the interesting question open of whether there exists a more efficient quantum algorithm for finding a Chebyshev cooling schedule.

1.4 Applications

We now sketch several representative settings (for details, see Section 3.4) in which our algorithm for approximating partition functions gives a quantum speedup.

- The **ferromagnetic Ising model** above the critical temperature. This well-studied statistical physics model is defined in terms of a graph $G = (V, E)$ by the Hamiltonian $H(z) = -\sum_{(u,v)\in E} z_u z_v$, where $|V| = n$ and $z \in \{\pm 1\}^n$. The Markov chain known as the Glauber dynamics is known to mix rapidly above a certain critical temperature and to have as its stationary distribution the Gibbs distribution. For example, for any graph with maximum degree $O(1)$, the mixing time of the Glauber dynamics for sufficiently low inverse temperature β is $O(n \log n)$ [44]. In this case, as $A = 2^n$, the quantum algorithm approximates $Z(\beta)$ to within relative error ϵ in $\widetilde{O}(n^{3/2}/\epsilon + n^2)$ steps. The corresponding classical algorithm [50] uses $\widetilde{O}(n^2/\epsilon^2)$ steps.

- **Counting colourings.** Here we are given a graph G with n vertices and maximum degree d. We seek to approximately count the number of valid k-colourings of G, where a colouring of the vertices is valid if all pairs of neighbouring vertices are assigned different colours. In the case where $k > 2d$, the use of a rapidly mixing Markov chain gives a quantum algorithm approximating the number of colourings of G up to relative error ϵ in time $\widetilde{O}(n^{3/2}/\epsilon + n^2)$, as compared with the classical $\widetilde{O}(n^2/\epsilon^2)$ [50].

- **Counting matchings.** A matching in a graph G is a subset M of the edges of G such that no pair of edges in M shares a vertex. In statistical physics, matchings are studied under the name of monomer-dimer coverings [26]. Our algorithm can approximately count the number of matchings on a graph with n vertices and m edges in $\widetilde{O}(n^{3/2}m^{1/2}/\epsilon + n^2 m)$ steps, as compared with the classical $\widetilde{O}(n^2 m/\epsilon^2)$ [50].

Finally, as another example of how our algorithm can be applied, we improve the accuracy of an existing quantum algorithm for estimating the total variation distance between probability distributions. In this setting, we are given the ability to sample from probability distributions p and q on n elements, and would like to estimate the distance between them up to additive error ϵ. A quantum algorithm of Bravyi, Harrow and Hassidim solves this problem using $O(\sqrt{n}/\epsilon^8)$ samples [13], while no classical algorithm can achieve sublinear dependence on n [54].

Quantum mean estimation can significantly improve the dependence of this quantum algorithm on ϵ. The total variation distance between p and q can be described as the expected value of the random variable $R(x) = \frac{|p(x)-q(x)|}{p(x)+q(x)}$, where x is drawn from the distribution $r = (p+q)/2$ [13]. For each x, $R(x)$ can be computed up to accuracy ϵ using $\widetilde{O}(\sqrt{n}/\epsilon^{3/2})$ iterations of amplitude estimation. Wrapping this within $O(1/\epsilon)$ iterations of the mean-estimation algorithm, we obtain an overall algorithm running in time $\widetilde{O}(\sqrt{n}/\epsilon^{5/2})$. See Section 4 for details.

2 Algorithms

We now give technical details, parameter values and proofs for the various algorithms described informally in Section 1.2. Recall that, for any randomised or quantum algorithm $\mathcal{A}$, we let $v(\mathcal{A})$ be the random variable corresponding to the value computed by $\mathcal{A}$. We assume that $\mathcal{A}$ takes no input directly, but may have access to input (e.g. via queries to some black box or "oracle") during its execution. We further assume throughout that $\mathcal{A}$ is a quantum algorithm of the following form: apply some unitary operator to the initial state $|0^n\rangle$; measure $k \le n$ qubits of the resulting state in the computational basis, obtaining outcome $x \in \{0,1\}^k$; output $\phi(x)$ for some easily computable function $\phi : \{0,1\}^k \to \mathbb{R}$. We finally assume that we have access to the inverse of the unitary part of the algorithm, which we write as $\mathcal{A}^{-1}$.

Lemma 1 (Powering lemma [35])**.** *Let $\mathcal{A}$ be a (classical or quantum) algorithm which aims to estimate some quantity μ, and whose output $\widetilde{\mu}$ satisfies $|\mu - \widetilde{\mu}| \le \epsilon$ except with probability γ, for some fixed $\gamma < 1/2$. Then, for any $\delta > 0$, it suffices to repeat $\mathcal{A}$ $O(\log 1/\delta)$ times and take the median to obtain an estimate which is accurate to within ϵ with probability at least $1 - \delta$.*

We will also need the following fundamental result from [12]:

Theorem 2 (Amplitude estimation [12])**.** *There is a quantum algorithm called* **amplitude estimation** *which takes as input one copy of a quantum state $|\psi\rangle$, a unitary transformation $U = 2|\psi\rangle\langle\psi| - I$, a unitary transformation $V = I - 2P$ for some projector P, and an integer t. The algorithm outputs $\widetilde{a}$, an estimate of $a = \langle\psi|P|\psi\rangle$, such that*

$$|\widetilde{a} - a| \leq 2\pi\frac{\sqrt{a(1-a)}}{t} + \frac{\pi^2}{t^2}$$

with probability at least $8/\pi^2$, using U and V t times each.

The success probability of $8/\pi^2$ can be improved to $1 - \delta$ for any $\delta > 0$ using the powering lemma at the cost of an $O(\log 1/\delta)$ multiplicative factor.

2.1 Estimating the mean with bounded output values

We first consider the problem of estimating $\mathbb{E}[v(\mathcal{A})]$ in the special case where $v(\mathcal{A})$ is bounded between 0 and 1. The algorithm for this case is effectively a combination of elegant ideas of Brassard et al. [11] and Wocjan et al. [59]. The former described an algorithm for efficiently approximating the mean of an arbitrary function with respect to the uniform distribution; the latter described an algorithm for approximating the expected value of a particular observable, with respect to an arbitrary quantum state. The first quantum algorithm achieving optimal scaling for approximating the mean of a bounded function under the uniform distribution was due to Heinrich [27].

Input: an algorithm $\mathcal{A}$ such that $0 \leq v(\mathcal{A}) \leq 1$, integer t, real $\delta > 0$.
Assume that $\mathcal{A}$ is a quantum algorithm which makes no measurements until the end of the algorithm; operates on initial input state $|0^n\rangle$; and its final measurement is a measurement of the last $k \leq n$ of these qubits in the computational basis.

1. If necessary, modify $\mathcal{A}$ such that it makes no measurements until the end of the algorithm; operates on initial input state $|0^n\rangle$; and its final measurement is a measurement of the last $k \leq n$ of these qubits in the computational basis.

2. Let W be the unitary operator on $k+1$ qubits defined by

   $$W|x\rangle|0\rangle = |x\rangle\left(\sqrt{1-\phi(x)}|0\rangle + \sqrt{\phi(x)}|1\rangle\right),$$

   where each computational basis state $x \in \{0,1\}^k$ is associated with a real number $\phi(x) \in [0,1]$ such that $\phi(x)$ is the value output by $\mathcal{A}$ when measurement outcome x is received.

3. Repeat the following step $O(\log 1/\delta)$ times and output the median of the results:

   (a) Apply t iterations of amplitude estimation, setting $|\psi\rangle = (I \otimes W)(\mathcal{A} \otimes I)|0^{n+1}\rangle$, $P = I \otimes |1\rangle\langle 1|$.

Algorithm 1: Approximating the mean output value of algorithms bounded between 0 and 1 (cf. [11, 27, 59])

Theorem 3. *Let $|\psi\rangle$ be defined as in Algorithm 1 and set $U = 2|\psi\rangle\langle\psi| - I$. Algorithm 1 uses $O(\log 1/\delta)$ copies of the state $\mathcal{A}|0^n\rangle$, uses U $O(t\log 1/\delta)$ times, and outputs an estimate $\widetilde{\mu}$ such* *that*

$$|\widetilde{\mu} - \mathbb{E}[v(\mathcal{A})]| \le C\left(\frac{\sqrt{\mathbb{E}[v(\mathcal{A})]}}{t} + \frac{1}{t^2}\right)$$

with probability at least $1-\delta$, where C is a universal constant. In particular, for any fixed $\delta > 0$ and any ϵ such that $0 \le \epsilon \le 1$, to produce an estimate $\widetilde{\mu}$ such that with probability at least $1-\delta$, $|\widetilde{\mu} - \mathbb{E}[v(\mathcal{A})]| \le \epsilon\,\mathbb{E}[v(\mathcal{A})]$ it suffices to take $t = O(1/(\epsilon\sqrt{\mathbb{E}[v(\mathcal{A})]}))$. To achieve $|\widetilde{\mu} - \mathbb{E}[v(\mathcal{A})]| \le \epsilon$ with probability at least $1-\delta$ it suffices to take $t = O(1/\epsilon)$.

Proof. The complexity claim follows immediately from Theorem 2. Also observe that W can be implemented efficiently, as it is a controlled rotation of one qubit dependent on the value of $\phi(x)$ [59]. It remains to show the accuracy claim. The final state of $\mathcal{A}$, just before its last measurement, can be written as

$$|\psi'\rangle = \mathcal{A}|0^n\rangle = \sum_x \alpha_x |\psi_x\rangle|x\rangle$$

for some normalised states $|\psi_x\rangle$. If we then attach an ancilla qubit and apply W, we obtain

$$|\psi\rangle = (I \otimes W)(\mathcal{A} \otimes I)|0^n\rangle|0\rangle = \sum_x \alpha_x|\psi_x\rangle|x\rangle\left(\sqrt{1-\phi(x)}|0\rangle + \sqrt{\phi(x)}|1\rangle\right).$$

We have

$$\langle\psi|P|\psi\rangle = \sum_x |\alpha_x|^2\phi(x) = \mathbb{E}[v(\mathcal{A})].$$

Therefore, when we apply amplitude estimation, by Theorem 2 we obtain an estimate $\widetilde{\mu}$ of $\mu = \mathbb{E}[v(\mathcal{A})]$ such that

$$|\widetilde{\mu} - \mu| \le 2\pi\frac{\sqrt{\mu(1-\mu)}}{t} + \frac{\pi^2}{t^2}$$

with probability at least $8/\pi^2$. The powering lemma (Lemma 1) implies that the median of $O(\log 1/\delta)$ repetitions will lie within this accuracy bound with probability at least $1-\delta$. □

Observe that $U = 2|\psi\rangle\langle\psi| - I$ can be implemented with one use each of $\mathcal{A}$ and $\mathcal{A}^{-1}$, and $V = I - 2P$ is easy to implement.

It seems likely that the median-finding algorithm of Nayak and Wu [45] could also be generalised in a similar way, to efficiently compute the median of the output values of any quantum algorithm. As we will not need this result here we do not pursue this further.

2.2 Estimating the mean with bounded ℓ_2 norm

We now use Algorithm 1 to give an efficient quantum algorithm for approximating the mean output value of a quantum algorithm whose output has bounded ℓ_2 norm. In what follows, for any algorithm $\mathcal{A}$, let $\mathcal{A}_{<x}$, $\mathcal{A}_{x,y}$, $\mathcal{A}_{\ge y}$, be the algorithms defined by executing $\mathcal{A}$ to produce a value $v(\mathcal{A})$ and:

- $\mathcal{A}_{<x}$: If $v(\mathcal{A}) < x$, output $v(\mathcal{A})$, otherwise output 0;
- $\mathcal{A}_{x,y}$: If $x \le v(\mathcal{A}) < y$, output $v(\mathcal{A})$, otherwise output 0;
- $\mathcal{A}_{\ge y}$: If $y \le v(\mathcal{A})$, output $v(\mathcal{A})$, otherwise output 0.

In addition, for any algorithm $\mathcal{A}$ and any function $f : \mathbb{R} \to \mathbb{R}$, let $f(\mathcal{A})$ be the algorithm produced by evaluating $v(\mathcal{A})$ and computing $f(v(\mathcal{A}))$. Note that Algorithm 1 can easily be modified to compute $\mathbb{E}[f(v(\mathcal{A}))]$ rather than $\mathbb{E}[v(\mathcal{A})]$, for any function $f : \mathbb{R} \to [0,1]$, by modifying the operation W.

Our algorithm and correctness proof are a generalisation of a result of Heinrich [27] for computing the mean with respect to the uniform distribution of functions with bounded L^2 norm, and are based on the same ideas. Write $\|v(\mathcal{A})\|_2 := \sqrt{\mathbb{E}[v(\mathcal{A})^2]}$.

```
Input: an algorithm $\mathcal{A}$ such that $v(\mathcal{A}) \geq 0$, and an accuracy $\epsilon < 1/2$.
  1. Set $k = \lceil \log_2 1/\epsilon \rceil$, $t_0 = \left\lceil \frac{D\sqrt{\log_2 1/\epsilon}}{\epsilon} \right\rceil$, where $D$ is a universal constant to be chosen later.
  2. Use Algorithm 1 with $t = t_0$, $\delta = 1/10$ to estimate $\mathbb{E}[v(\mathcal{A}_{0,1})]$. Let the estimate be $\widetilde{\mu}_0$.
  3. For $\ell = 1, \dots, k$:
     (a) Use Algorithm 1 with $t = t_0$, $\delta = 1/(10k)$ to estimate $\mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})/2^\ell]$. Let the estimate be $\widetilde{\mu}_\ell$.
  4. Output $\widetilde{\mu} = \widetilde{\mu}_0 + \sum_{\ell=1}^k 2^\ell \widetilde{\mu}_\ell$.
```

Algorithm 2: Approximating the mean of positive functions with bounded ℓ_2 norm

Lemma 4. *Let $|\psi\rangle = \mathcal{A}|0^n\rangle$, $U = 2|\psi\rangle\langle\psi| - I$. Algorithm 2 uses $O(\log(1/\epsilon)\log\log(1/\epsilon))$ copies of $|\psi\rangle$, uses U $O((1/\epsilon)\log^{3/2}(1/\epsilon)\log\log(1/\epsilon))$ times, and estimates $\mathbb{E}[v(\mathcal{A})]$ up to additive error $\epsilon(\|v(\mathcal{A})\|_2 + 1)^2$ with probability at least $4/5$.*

Proof. We first show the resource bounds. Algorithm 1 is run $\Theta(\log 1/\epsilon)$ times, each time with parameter $\delta = \Omega(1/(\log 1/\epsilon))$. By Theorem 3, each use of Algorithm 1 consumes $O(\log\log 1/\epsilon)$ copies of $|\psi\rangle$ and uses U $O((1/\epsilon)\sqrt{\log(1/\epsilon)}\log\log(1/\epsilon))$ times. The total number of copies of $|\psi\rangle$ used is $O(\log(1/\epsilon)\log\log(1/\epsilon))$, and the total number of uses of U is $O((1/\epsilon)\log^{3/2}(1/\epsilon)\log\log(1/\epsilon))$.

All of the uses of Algorithm 1 succeed, except with probability at most $1/5$ in total. To estimate the total error in the case where they all succeed, we write

$$\mathbb{E}[v(\mathcal{A})] = \mathbb{E}[v(\mathcal{A}_{0,1})] + \sum_{\ell=1}^{k} 2^\ell \mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})/2^\ell] + \mathbb{E}[v(\mathcal{A}_{\geq 2^k})]$$

and use the triangle inequality term by term to obtain

$$|\widetilde{\mu} - \mathbb{E}[v(\mathcal{A})]| \leq |\widetilde{\mu}_0 - \mathbb{E}[v(\mathcal{A}_{0,1})]| + \sum_{\ell=1}^{k} 2^\ell |\widetilde{\mu}_\ell - \mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})/2^\ell]| + \mathbb{E}[v(\mathcal{A}_{\geq 2^k})].$$

Let $p(x)$ denote the probability that $\mathcal{A}$ outputs x. We have

$$\mathbb{E}[v(\mathcal{A}_{\geq 2^k})] = \sum_{x \geq 2^k} p(x)x \leq \frac{1}{2^k}\sum_x p(x)x^2 = \frac{\|v(\mathcal{A})\|_2^2}{2^k}.$$

By Theorem 3,

$$|\widetilde{\mu}_0 - \mathbb{E}[v(\mathcal{A}_{0,1})]| \leq C\left(\frac{\sqrt{\mathbb{E}[v(\mathcal{A}_{0,1})]}}{t_0} + \frac{1}{t_0^2}\right)$$

and similarly

$$|\widetilde{\mu}_\ell - \mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})/2^\ell]| \le C\left(\frac{\sqrt{\mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})]}}{t_0\,2^{\ell/2}} + \frac{1}{t_0^2}\right).$$

So the total error is at most

$$C\left(\frac{\sqrt{\mathbb{E}[v(\mathcal{A}_{0,1})]}}{t_0} + \frac{1}{t_0^2} + \sum_{\ell=1}^{k} 2^\ell \left(\frac{\sqrt{\mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})]}}{t_0\,2^{\ell/2}} + \frac{1}{t_0^2}\right)\right) + \frac{\|v(\mathcal{A})\|_2^2}{2^k}.$$

We apply Cauchy-Schwarz to the first part of each term in the sum:

$$\sum_{\ell=1}^{k} 2^{\ell/2}\sqrt{\mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})]} \le \sqrt{k}\left(\sum_{\ell=1}^{k} 2^\ell \mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})]\right)^{1/2} \le \sqrt{2k}\,\|v(\mathcal{A})\|_2\,,$$

where the second inequality follows from

$$\mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})] = \sum_{2^{\ell-1}\le x<2^\ell} p(x)x \le \frac{1}{2^{\ell-1}} \sum_{2^{\ell-1}\le x<2^\ell} p(x)x^2 = \frac{\|v(\mathcal{A}_{2^{\ell-1},2^\ell})\|_2^2}{2^{\ell-1}}.$$

Inserting this bound and using $\mathbb{E}[v(\mathcal{A}_{0,1})] \le 1$, we obtain

$$|\widetilde{\mu} - \mathbb{E}[v(\mathcal{A})]| \le C\left(\frac{1}{t_0} + \frac{1}{t_0^2} + \frac{\sqrt{2k}\,\|v(\mathcal{A})\|_2}{t_0} + \frac{2^{k+1}}{t_0^2}\right) + \frac{\|v(\mathcal{A})\|_2^2}{2^k}.$$

Inserting the definitions of t_0 and k, we get an overall error bound

$$\begin{aligned}
&|\widetilde{\mu} - \mathbb{E}[v(\mathcal{A})]| \\
&\quad\le \frac{C}{D}\left(\frac{\epsilon}{\sqrt{\log_2 1/\epsilon}} + \frac{\epsilon^2}{D\log_2 1/\epsilon} + \sqrt{2}\epsilon\,\|v(\mathcal{A})\|_2\left(1 + \frac{1}{\log_2 1/\epsilon}\right)^{1/2} + \frac{4\epsilon}{D\log_2 1/\epsilon}\right) + \epsilon\,\|v(\mathcal{A})\|_2^2 \\
&\quad\le \frac{C}{D}\left(\epsilon + \frac{\epsilon}{D} + 2\epsilon\,\|v(\mathcal{A})\|_2 + \frac{4\epsilon}{D}\right) + \epsilon\,\|v(\mathcal{A})\|_2^2 \\
&\quad= \epsilon\left(\frac{C}{D}\left(1 + \frac{5}{D} + 2\,\|v(\mathcal{A})\|_2\right) + \|v(\mathcal{A})\|_2^2\right)
\end{aligned}$$

using $0 < \epsilon < 1/2$ in the second inequality. For a sufficiently large constant D, this is upper-bounded by $\epsilon(\|v(\mathcal{A})\|_2 + 1)^2$ as claimed. □

Observe that, if $\mathbb{E}[v(\mathcal{A})^2] = O(1)$, to achieve additive error ϵ the number of uses of $\mathcal{A}$ that we need is $O((1/\epsilon)\log^{3/2}(1/\epsilon)\log\log(1/\epsilon))$. By the powering lemma, we can repeat Algorithm 2 $O(\log 1/\delta)$ times and take the median to improve the probability of success to $1-\delta$ for any $\delta > 0$.

2.3 Estimating the mean with bounded variance

We are now ready to formally state our algorithm for estimating the mean output value of an arbitrary algorithm with bounded variance. For clarity, some of the steps are reordered as compared with the informal description in Section 1.2. Recall that, in the classical setting, if we wish to estimate $\mathbb{E}[v(\mathcal{A})]$ up to additive error ϵ for an arbitrary algorithm $\mathcal{A}$ such that

$$\mathrm{Var}(v(\mathcal{A})) := \mathbb{E}[(v(\mathcal{A}) - \mathbb{E}[v(\mathcal{A})])^2] \le \sigma^2,$$

we need to use $\mathcal{A}$ $\Omega(\sigma^2/\epsilon^2)$ times [15].

```
Input: an algorithm $\mathcal{A}$ such that $\mathrm{Var}(v(\mathcal{A})) \leq \sigma^2$ for some known $\sigma$, and an accuracy $\epsilon$ such that $\epsilon < 4\sigma$.

1. Set $\mathcal{A}' = \mathcal{A}/\sigma$.

2. Run $\mathcal{A}'$ once and let $\widetilde{m}$ be the output.

3. Let $\mathcal{B}$ be the algorithm produced by executing $\mathcal{A}'$ and subtracting $\widetilde{m}$.

4. Apply Algorithm 2 to algorithms $-\mathcal{B}_{<0}/4$ and $\mathcal{B}_{\geq 0}/4$ with accuracy $\epsilon/(32\sigma)$ and failure
   probability $1/9$, to produce estimates $\widetilde{\mu}^-$, $\widetilde{\mu}^+$ of $\mathbb{E}[v(-\mathcal{B}_{<0})/4]$ and $\mathbb{E}[v(\mathcal{B}_{\geq 0})/4]$, respectively.

5. Set $\widetilde{\mu} = \widetilde{m} - 4\widetilde{\mu}^- + 4\widetilde{\mu}^+$.

6. Output $\sigma\,\widetilde{\mu}$.
```

Algorithm 3: Approximating the mean with bounded variance

Theorem 5. *Let $|\psi\rangle = \mathcal{A}|0^n\rangle$, $U = 2|\psi\rangle\langle\psi| - I$. Algorithm 3 uses $O(\log(\sigma/\epsilon)\log\log(\sigma/\epsilon))$ copies of $|\psi\rangle$, uses U $O((\sigma/\epsilon)\log^{3/2}(\sigma/\epsilon)\log\log(\sigma/\epsilon))$ times, and estimates $\mathbb{E}[v(\mathcal{A})]$ up to additive error ϵ with success probability at least $2/3$.*

Proof. First, observe that $\widetilde{m}$ is quite close to $\mu' := \mathbb{E}[v(\mathcal{A}')]$ with quite high probability. As $\mathrm{Var}(v(\mathcal{A}')) = \mathrm{Var}(v(\mathcal{A}))/\sigma^2 \leq 1$, by Chebyshev's inequality we have

$$\Pr[|v(\mathcal{A}') - \mu'| \geq 3] \leq \frac{1}{9}.$$

We therefore assume that $|\widetilde{m} - \mu'| \leq 3$. In this case we have

$$\begin{aligned} \|v(\mathcal{B})\|_2 &= \mathbb{E}[v(\mathcal{B})^2]^{1/2} = \mathbb{E}[((v(\mathcal{A}') - \mu') + (\mu' - \widetilde{m}))^2]^{1/2} \\ &\leq \mathbb{E}[(v(\mathcal{A}') - \mu')^2]^{1/2} + \mathbb{E}[(\mu' - \widetilde{m})^2]^{1/2} \\ &\leq 4, \end{aligned}$$

where the first inequality is the triangle inequality. Thus $\|v(\mathcal{B})/4\|_2 \leq 1$, which implies that $\|v(-\mathcal{B}_{<0})/4\|_2 \leq 1$ and $\|v(\mathcal{B}_{\geq 0})/4\|_2 \leq 1$.

The next step is to use Algorithm 2 to estimate $\mathbb{E}[v(-\mathcal{B}_{<0})/4]$ and $\mathbb{E}[v(\mathcal{B}_{\geq 0})/4]$ with accuracy $\epsilon/(32\sigma)$ and failure probability $1/9$. By Lemma 4, if the algorithm succeeds in both cases the estimates are accurate up to $\epsilon/(8\sigma)$. We therefore obtain an approximation of each of $\mathbb{E}[v(-\mathcal{B}_{<0})]$ and $\mathbb{E}[v(\mathcal{B}_{\geq 0})]$ up to additive error $\epsilon/(2\sigma)$. As we have

$$\mathbb{E}[v(\mathcal{A})] = \sigma\,\mathbb{E}[v(\mathcal{A}')] = \sigma(\widetilde{m} - \mathbb{E}[v(-\mathcal{B}_{<0})] + \mathbb{E}[v(\mathcal{B}_{\geq 0})])$$

by linearity of expectation, using a union bound we have that $\sigma\,\widetilde{\mu}$ approximates $\mathbb{E}[v(\mathcal{A})]$ up to additive error ϵ with probability at least $2/3$. □

2.4 Estimating the mean with bounded relative error

It is often useful to obtain an estimate of the mean output value of an algorithm which is accurate up to small relative error, rather than the absolute error achieved by Algorithm 3. Assume that we have the bound on the relative variance that $\mathrm{Var}(v(\mathcal{A}))/(\mathbb{E}[v(\mathcal{A})])^2 \leq B$, where we normally think of B as small, e.g. $B = O(1)$. Classically, it follows from Chebyshev's inequality that the simple classical algorithm described in the Introduction approximates $\mathbb{E}[v(\mathcal{A})]$ up to additive error $\epsilon \, \mathbb{E}[v(\mathcal{A})]$ with $O(B/\epsilon^2)$ uses of $\mathcal{A}$. In the quantum setting, we can improve the dependence on ϵ near-quadratically.

```
Input: An algorithm $\mathcal{A}$ such that $v(\mathcal{A}) \geq 0$ and $\mathrm{Var}(v(\mathcal{A}))/(\mathbb{E}[v(\mathcal{A})])^2 \leq B$ for some $B \geq 1$,
and an accuracy $\epsilon < 27B/4$.

  1. Run $\mathcal{A}$ $k = \lceil 32B \rceil$ times, receiving output values $v_1, \dots, v_k$, and set $\widetilde{m} = \frac{1}{k} \sum_{i=1}^{k} v_i$.

  2. Apply Algorithm 2 to $\mathcal{A}/\widetilde{m}$ with accuracy $2\epsilon/(3(2\sqrt{B}+1)^2)$ and failure probability $1/8$.
     Let $\widetilde{\mu}$ be the output of the algorithm, multiplied by $\widetilde{m}$.

  3. Output $\widetilde{\mu}$.
```

Algorithm 4: Approximating the mean with bounded relative error

Theorem 6. *Let $|\psi\rangle = \mathcal{A}|0^n\rangle$, $U = 2|\psi\rangle\langle\psi| - I$. Algorithm 4 uses $O(B + \log(1/\epsilon) \log\log(1/\epsilon))$ copies of $|\psi\rangle$, uses U $O((B/\epsilon)\log^{3/2}(B/\epsilon)\log\log(B/\epsilon))$ times, and outputs an estimate $\widetilde{\mu}$ such that*

$$\Pr[|\widetilde{\mu} - \mathbb{E}[v(\mathcal{A})]| \geq \epsilon \, \mathbb{E}[v(\mathcal{A})]] \leq 1/4.$$

Proof. The complexity bounds follow from Lemma 4; we now analyse the claim about accuracy. $\widetilde{m}$ is a random variable whose expectation is $\mathbb{E}[v(\mathcal{A})]$ and whose variance is $\mathrm{Var}(v(\mathcal{A}))/\lceil 32B \rceil$. By Chebyshev's inequality, we have

$$\Pr[|\widetilde{m} - \mathbb{E}[\widetilde{m}]| \geq |\mathbb{E}[\widetilde{m}]|/2] \leq \frac{4\,\mathrm{Var}(\widetilde{m})}{\mathbb{E}[\widetilde{m}]^2} = \frac{4\,\mathrm{Var}(v(\mathcal{A}))}{\lceil 32B \rceil \mathbb{E}[v(\mathcal{A})]^2} \leq \frac{1}{8}.$$

We can thus assume that $\mathbb{E}[v(\mathcal{A})]/2 \leq \widetilde{m} \leq 3\,\mathbb{E}[v(\mathcal{A})]/2$. In this case, when we apply Algorithm 2 to $\mathcal{A}/\widetilde{m}$, we receive an estimate of $\mathbb{E}[v(\mathcal{A})]/\widetilde{m}$ which is accurate up to additive error

$$\frac{2\epsilon(\|v(\mathcal{A})\|_2/\widetilde{m} + 1)^2}{3(2\sqrt{B}+1)^2} \leq \frac{\epsilon\,\mathbb{E}[v(\mathcal{A})](2\,\|v(\mathcal{A})\|_2/\mathbb{E}[v(\mathcal{A})] + 1)^2}{\widetilde{m}(2\sqrt{B}+1)^2} \leq \frac{\epsilon\,\mathbb{E}[v(\mathcal{A})]}{\widetilde{m}}$$

except with probability $1/8$, where we use $\|v(\mathcal{A})\|_2/\mathbb{E}[v(\mathcal{A})] \leq \sqrt{B}$. Multiplying by $\widetilde{m}$ and taking a union bound, we get an estimate of $\mathbb{E}[v(\mathcal{A})]$ which is accurate up to ϵ except with probability at most $1/4$. □

Once again, using the powering lemma we can repeat Algorithms 3 and 4 $O(\log 1/\delta)$ times and take the median to improve their probabilities of success to $1 - \delta$ for any $\delta > 0$.

To see that Algorithms 3 and 4 are close to optimal, we can appeal to a result of Nayak and Wu [45]. Let $\mathcal{A}$ be an algorithm which picks an integer x between 1 and N uniformly at random, for some large N, and outputs $f(x)$ for some function $f : \{1, \dots, N\} \to \{0, 1\}$. Then $\mathbb{E}[v(\mathcal{A})] = |\{x : f(x) = 1\}|/N$. It was shown by Nayak and Wu [45] that any quantum algorithm which computes this quantity for an arbitrary function f up to (absolute or relative) error ϵ must make at most $\Omega(1/\epsilon)$ queries to f in the case that $|\{x : f(x) = 1\}| = N/2$. As the output of $\mathcal{A}$ for any such function has variance $1/4$, this implies that Algorithms 2 and 4 are optimal in the black-box setting in terms of their scaling with ϵ, up to polylogarithmic factors. By rescaling, we get a similar near-optimality claim for Algorithm 3 in terms of its scaling with σ.

3 Partition function problems

In this section we formally state and prove our results about partition function problems. We first recall the definitions from Section 1.3. A partition function Z is defined by

$$Z(\beta) = \sum_{x \in \Omega} e^{-\beta\, H(x)}$$

where β is an inverse temperature and H is a Hamiltonian function taking integer values in the set $\{0, \dots, n\}$. Let $0 = \beta_0 < \beta_1 < \cdots < \beta_\ell = \infty$ be a sequence of inverse temperatures and assume that we can easily compute $Z(\beta_0) = |\Omega|$. We want to approximate $Z(\infty)$ by approximating the ratios $\alpha_i := Z(\beta_{i+1})/Z(\beta_i)$ and using the telescoping product

$$Z(\beta_\ell) = Z(\beta_0) \frac{Z(\beta_1)}{Z(\beta_0)} \frac{Z(\beta_2)}{Z(\beta_1)} \cdots \frac{Z(\beta_\ell)}{Z(\beta_{\ell-1})}.$$

Finally, a sequence of Gibbs distributions π_i is defined by

$$\pi_i(x) = \frac{1}{Z(\beta_i)} e^{-\beta_i H(x)}.$$

3.1 Chebyshev cooling schedules

We start by motivating, and formally defining, the concept of a Chebyshev cooling schedule [50]. To approximate α_i we define the random variable

$$Y_i(x) = e^{-(\beta_{i+1} - \beta_i) H(x)}.$$

Then

$$\mathbb{E}[Y_i] := \mathbb{E}_{\pi_i}[Y_i] = \frac{1}{Z(\beta_i)} \sum_{x \in \Omega} e^{-\beta_i H(x)} e^{-(\beta_{i+1}-\beta_i)H(x)} = \frac{1}{Z(\beta_i)} \sum_{x \in \Omega} e^{-\beta_{i+1} H(x)} = \frac{Z(\beta_{i+1})}{Z(\beta_i)} = \alpha_i.$$

The following result was shown by Dyer and Frieze [19] (see [50] for the statement here):

Theorem 7. *Let $Y_0, \dots, Y_{\ell-1}$ be independent random variables such that $\mathbb{E}[Y_i^2]/\mathbb{E}[Y_i]^2 \le B$ for all i, and write $\overline{Y} = \mathbb{E}[Y_0]\mathbb{E}[Y_1] \dots \mathbb{E}[Y_{\ell-1}]$. Let $\widetilde{\alpha}_i$ be the average of $16B\ell/\epsilon^2$ independent samples from Y_i, and set $\widetilde{Y} = \widetilde{\alpha}_0\, \widetilde{\alpha}_1 \dots \widetilde{\alpha}_{\ell-1}$. Then*

$$\Pr[(1-\epsilon)\overline{Y} \le \widetilde{Y} \le (1+\epsilon)\overline{Y}] \ge 3/4.$$

Thus a classical algorithm can approximate $Z(\infty)$ up to relative error ϵ using $O(B\ell^2/\epsilon^2)$ samples in total, assuming that $Z(0)$ can be computed without using any samples and that we have $\mathbb{E}[Y_i^2]/\mathbb{E}[Y_i]^2 \le B$. To characterise the latter constraint, observe that we have

$$\mathbb{E}[Y_i^2] = \frac{1}{Z(\beta_i)} \sum_{x \in \Omega} e^{-\beta_i H(x)} e^{-2(\beta_{i+1}-\beta_i)H(x)} = \frac{1}{Z(\beta_i)} \sum_{x \in \Omega} e^{(\beta_i - 2\beta_{i+1})H(x)} = \frac{Z(2\beta_{i+1} - \beta_i)}{Z(\beta_i)},$$

so

$$\frac{\mathbb{E}[Y_i^2]}{(\mathbb{E}[Y_i])^2} = \frac{Z(2\beta_{i+1} - \beta_i) Z(\beta_i)}{Z(\beta_{i+1})^2}.$$

This motivates the following definition:

Definition 1 (Chebyshev cooling schedules [50])**.** *Let Z be a partition function. Let $\beta_0, \dots, \beta_\ell$ be a sequence of inverse temperatures such that $0 = \beta_0 < \beta_1 < \dots < \beta_\ell = \infty$. The sequence is called a B-Chebyshev cooling schedule for Z if*

$$\frac{Z(2\beta_{i+1} - \beta_i)Z(\beta_i)}{Z(\beta_{i+1})^2} \leq B$$

for all i, for some fixed B.

Assume that we have a sequence of estimates $\widetilde{\alpha}_i$ such that, for all i, $|\widetilde{\alpha}_i - \alpha_i| \leq (\epsilon/2\ell)\, \alpha_i$ with probability at least $1 - 1/(4\ell)$. We output as a final estimate

$$\widetilde{Z} = Z(0)\, \widetilde{\alpha}_0\, \widetilde{\alpha}_1 \dots \widetilde{\alpha}_{\ell-1}.$$

By a union bound, all of the estimates $\widetilde{\alpha}_i$ are accurate to within $(\epsilon/2\ell)\, \alpha_i$, except with probability at most $1/4$. Assuming that all the estimates are indeed accurate, we have

$$1 - \epsilon/2 \leq (1 - \epsilon/(2\ell))^\ell \leq \frac{\widetilde{Z}}{Z(\infty)} \leq (1 + \epsilon/(2\ell))^\ell \leq e^{\epsilon/2} \leq 1 + \epsilon$$

for $\epsilon < 1$. Thus $|\widetilde{Z} - Z(\infty)| \leq \epsilon\, Z(\infty)$ with probability at least $3/4$.

Using these ideas, we can formalise the discussion in Section 1.3.

Theorem 8. *Let Z be a partition function with $|\Omega| = A$. Assume that we are given a B-Chebyshev cooling schedule $0 = \beta_0 < \beta_1 < \dots < \beta_\ell = \infty$ for Z. Further assume that we have the ability to exactly sample from the distributions π_i, $i = 1, \dots, \ell - 1$. Then there is a quantum algorithm which outputs an estimate $\widetilde{Z}$ such that*

$$\Pr[(1-\epsilon)Z(\infty) \leq \widetilde{Z} \leq (1+\epsilon)Z(\infty)] \geq 3/4.$$

using

$$O\left(\frac{B\ell \log \ell}{\epsilon} \log^{3/2}\left(\frac{B\ell}{\epsilon}\right) \log\log\left(\frac{B\ell}{\epsilon}\right)\right) = \widetilde{O}\left(\frac{B\ell^2}{\epsilon}\right)$$

samples in total.

Proof. For each $i = 1, \dots, \ell - 1$, we use Algorithm 4 to estimate $\mathbb{E}[Y_i]$ up to additive error $(\epsilon/(2\ell))\mathbb{E}[Y_i]$ with failure probability $1/(4\ell)$. As the β_i form a B-Chebyshev cooling schedule, $\mathbb{E}[Y_i^2]/\mathbb{E}[Y_i]^2 \leq B$, so $\mathrm{Var}(Y_i)/\mathbb{E}[Y_i]^2 \leq B$. By Theorem 6, each use of Algorithm 4 requires

$$O\left(\frac{B\ell}{\epsilon} \log^{3/2}\left(\frac{B\ell}{\epsilon}\right) \log\log\left(\frac{B\ell}{\epsilon}\right) \log \ell\right)$$

samples from π_i to achieve the desired accuracy and failure probability. The total number of samples is thus

$$O\left(\frac{B\ell^2 \log \ell}{\epsilon} \log^{3/2}\left(\frac{B\ell}{\epsilon}\right) \log\log\left(\frac{B\ell}{\epsilon}\right)\right)$$

as claimed. □

3.2 Approximate sampling

It is unfortunately not always possible to exactly sample from the distributions π_i. However, one classical way of approximately sampling from each of these distributions is to use a (reversible, ergodic) Markov chain which has unique stationary distribution π_i. Assume the Markov chain has relaxation time τ, where $\tau := 1/(1-|\lambda_1|)$, and λ_1 is the second largest eigenvalue in absolute value. Then one can sample from a distribution $\widetilde{\pi}_i$ such that $\|\widetilde{\pi}_i - \pi_i\| \le \epsilon$ using $O(\tau \log(1/(\epsilon \pi_{\min,i})))$ steps of the chain, where $\pi_{\min,i} = \min_x |\pi_i(x)|$ [38]. We would like to replace the classical Markov chain with a quantum walk, to obtain a faster mixing time. A construction due to Szegedy [51] defines a quantum walk corresponding to any ergodic Markov chain, such that the dependence on τ in the mixing time can be improved to $O(\sqrt{\tau})$ [48]. Unfortunately, it is not known whether in general the dependence on $\pi_{\min,i}$ can be kept logarithmic [48, 18]. Indeed, proving such a result is likely to be hard, as it would imply a polynomial-time quantum algorithm for graph isomorphism [6].

Nevertheless, it was shown by Wocjan and Abeyesinghe [58] (improving previous work on using quantum walks for classical annealing [49]) that one can achieve relatively efficient quantum sampling if one has access to a sequence of slowly varying Markov chains.

Theorem 9 (Wocjan and Abeyesinghe [58])**.** *Let $M_0, \dots, M_r$ be classical reversible Markov chains with stationary distributions $\pi_0, \dots, \pi_r$ such that each chain has relaxation time at most τ. Assume that $|\langle \pi_i | \pi_{i+1} \rangle|^2 \ge p$ for some $p > 0$ and all $i \in \{0, \dots, r-1\}$, and that we can prepare the state $|\pi_0\rangle$. Then, for any $\epsilon > 0$, there is a quantum algorithm which produces a quantum state $|\widetilde{\pi}_r\rangle$ such that $\||\widetilde{\pi}_r\rangle - |\pi_r\rangle|0^a\rangle\| \le \epsilon$, for some integer a. The algorithm uses*

$$O(r\sqrt{\tau} \log^2(r/\epsilon)(1/p)\log(1/p))$$

steps in total of the quantum walk operators W_i corresponding to the chains M_i.

In addition, one can approximately reflect about the states $|\pi_i\rangle$ more efficiently still, with a runtime that does not depend on r. This will be helpful because Algorithm 4 uses significantly more reflections than it does copies of the starting state.

Theorem 10 (Wocjan and Abeyesinghe [58], see [59] for version here)**.** *Let $M_0, \dots, M_r$ be classical reversible Markov chains with stationary distributions $\pi_0, \dots, \pi_r$ such that each chain has relaxation time at most τ. For each i there is an approximate reflection operator $\widetilde{R}_i$ such that*

$$\widetilde{R}_i|\phi\rangle|0^b\rangle = (2|\psi\rangle\langle\psi| - I)|\phi\rangle|0^b\rangle + |\xi\rangle,$$

where $|\phi\rangle$ is arbitrary, $b = O((\log\tau)(\log 1/\epsilon))$, and $|\xi\rangle$ is a vector with $\||\xi\rangle\| \le \epsilon$. The algorithm uses $O(\sqrt{\tau}\log(1/\epsilon))$ steps of the quantum walk operator W_i corresponding to the chain M_i.

In our setting, we can easily create the quantum state $|\pi_0\rangle$, which is the uniform superposition over all configurations x. We now show that the overlaps $|\langle \pi_i | \pi_{i+1} \rangle|^2$ are large for all i. We go via the chi-squared divergence

$$\chi^2(\nu, \pi) := \sum_{x \in \Omega} \pi(x) \left(\frac{\nu(x)}{\pi(x)} - 1 \right)^2 = \sum_{x \in \Omega} \frac{\nu(x)^2}{\pi(x)} - 1.$$

As noted in [50], one can calculate that

$$\chi^2(\pi_{i+1}, \pi_i) = \frac{Z(\beta_i) Z(2\beta_{i+1} - \beta_i)}{Z(\beta_{i+1})^2} - 1. \tag{1}$$

Therefore, if the β_i values form a Chebyshev cooling schedule, $\chi^2(\pi_{i+1}, \pi_i) \leq B - 1$ for all i. For any distributions ν, π, we also have

$$\frac{1}{\sqrt{\chi^2(\nu,\pi)+1}} = \frac{1}{\sqrt{\sum_{x\in\Omega} \nu(x)\frac{\nu(x)}{\pi(x)}}} \leq \sum_{x\in\Omega} \nu(x)\sqrt{\frac{\pi(x)}{\nu(x)}} = \langle \nu|\pi\rangle$$

by applying Jensen's inequality to the function $x \mapsto 1/\sqrt{x}$. So, for all i, $|\langle \pi_i|\pi_{i+1}\rangle|^2 \geq 1/B$. Note that in [50] it was necessary to introduce the concept of a reversible Chebyshev cooling schedule to facilitate "warm starts" of the Markov chains used in the algorithm. That work uses the fact that one can efficiently sample from π_{i+1}, given access to samples from π_i, if $\chi^2(\pi_i, \pi_{i+1}) = O(1)$; this is the reverse of the condition (1). Here we do not need to reverse the schedule as the precondition $|\langle \pi_i|\pi_{i+1}\rangle|^2 \geq \Omega(1)$ required for Theorem 9 is already symmetric.

We are now ready to formally state our result about approximating partition functions. We assume that ϵ is relatively small to simplify the bounds; this is not an essential restriction.

Theorem 11. *Let Z be a partition function. Assume we have a B-Chebyshev cooling schedule $\beta_0 = 0 < \beta_1 < \beta_2 < \cdots < \beta_\ell = \infty$ for $B = O(1)$. Assume that for every inverse temperature β_i we have a reversible ergodic Markov chain M_i with stationary distribution π_i and relaxation time upper-bounded by τ. Further assume that we can sample directly from M_0. Then, for any $\delta > 0$ and $\epsilon = O(1/\sqrt{\log \ell})$, there is a quantum algorithm which uses*

$$O((\ell^2\sqrt{\tau}/\epsilon)\log^{5/2}(\ell/\epsilon)\log(\ell/\delta)\log\log(\ell/\epsilon)) = \widetilde{O}(\ell^2\sqrt{\tau}/\epsilon)$$

steps of the quantum walks corresponding to the M_i chains and outputs $\widetilde{Z}$ such that

$$\Pr[(1-\epsilon)Z(\infty) \leq \widetilde{Z} \leq (1+\epsilon)Z(\infty)] \geq 1-\delta.$$

Proof. For each i, we use Algorithm 4 to approximate α_i up to relative error $\epsilon/(2\ell)$, with failure probability γ, for some small constant γ. This would require R reflections about the state $|\pi_{\beta_i}\rangle$, for some R such that $R = O((\ell/\epsilon)\log^{3/2}(\ell/\epsilon)\log\log(\ell/\epsilon))$, and $O(\log(\ell/\epsilon)\log\log(\ell/\epsilon))$ copies of $|\pi_{\beta_i}\rangle$.

Instead of performing exact reflections and using exact copies of the states $|\pi_i\rangle$, we use approximate reflections and approximate copies of $|\pi_i\rangle$. By Theorem 10, $O(\sqrt{\tau}\log(1/\epsilon_r))$ walk operations are sufficient to reflect about $|\pi_i\rangle$ up to an additive error term of order ϵ_r. By Theorem 9, as we have a Chebyshev cooling schedule, a quantum state $|\widetilde{\pi}_i\rangle$ such that $\||\widetilde{\pi}_i\rangle - |\pi_i\rangle|0^b\rangle\| \leq \epsilon_s$ can be produced using $O(\ell\sqrt{\tau}\log^2(\ell/\epsilon_s))$ steps of the quantum walks corresponding to the Markov chains $M_0, \dots, M_i$.

We choose $\epsilon_r = \gamma/R$, $\epsilon_s = \gamma$. Then the final state of Algorithm 4 using approximate reflections and starting with the states $|\widetilde{\pi}_i\rangle$ rather than $|\pi_i\rangle$ can differ from the final state of an exact algorithm by at most $R\epsilon_r + \epsilon_s = 2\gamma$ in ℓ_2 norm. This implies that the total variation distance between the output probability distributions of the exact and inexact algorithms is at most 2γ, and hence by a union bound that the approximation is accurate up to relative error $\epsilon/(2\ell)$ except with probability 3γ. For each i, we then take the median of $O(\log(\ell/\delta))$ estimates to achieve an estimate which is accurate up to relative error $\epsilon/(2\ell)$ except with probability at most δ/ℓ. By a union bound, all the estimates are accurate up to relative error $\epsilon/(2\ell)$ except with probability at most δ, so their product is accurate to relative error ϵ except with probability at most δ.

The total number of steps needed to produce all the copies of the states $|\widetilde{\pi}_i\rangle$ required is thus

$$O(\ell \cdot \ell\sqrt{\tau}(\log^2 \ell) \cdot \log(\ell/\epsilon)\log\log(\ell/\epsilon) \cdot \log(\ell/\delta))$$

and the total number of steps needed to perform the reflections is

$$O(\ell \cdot \sqrt{\tau}(\log R) \cdot R \cdot \log(\ell/\delta)).$$

Adding the two, substituting the value of R, and using $\epsilon = O(1/\sqrt{\log \ell})$, we get an overall bound of

$$O((\ell^2\sqrt{\tau}/\epsilon)\log^{5/2}(\ell/\epsilon)\log(\ell/\delta)\log\log(\ell/\epsilon)) = \widetilde{O}(\ell^2\sqrt{\tau}/\epsilon)$$

as claimed. □

We remark that, in the above complexities, we have chosen to take the number of quantum walk steps used as our measure of complexity. This is to enable a straightforward comparison with the classical literature, which typically uses a random walk step as its elementary operation for the purposes of measuring complexity [50]. To implement each quantum walk step efficiently and accurately, two possible approaches are to use efficient state preparation [14] or recently developed approaches to efficient simulation of sparse Hamiltonians [9].

Finally, we mention that one could also consider a more general setting for approximate sampling. Imagine that we would like to approximate the mean μ of some random variable chosen according to some distribution π, but only have access to samples from a distribution $\widetilde{\pi}$ that approximates π (using some method which, for example, might not be a quantum walk). In this case, one can show that the estimation algorithm does not notice the difference between $\widetilde{\pi}$ and π and hence allows efficient estimation of μ. See Appendix A for the details.

3.3 Computing a Chebyshev cooling schedule

We still need to show that, given a particular partition function, we can actually find a Chebyshev cooling schedule. For this we simply use a known classical result:

Theorem 12 (Štefankovič, Vempala and Vigoda [50])**.** *Let Z be a partition function. Assume that for every inverse temperature β we have a Markov chain M_β with stationary distribution π_β and relaxation time upper-bounded by τ. Further assume that we can sample directly from M_0. Then, for any $\delta > 0$ and any $B = O(1)$, we can produce a B-Chebyshev cooling schedule of length*

$$\ell = O(\sqrt{\log A}(\log n)(\log\log A))$$

with probability at least $1-\delta$, using at most

$$Q = O((\log A)((\log n) + \log\log A)^5 \tau \log(1/\delta))$$

steps of the Markov chains.

We remark that a subsequent algorithm [28] improves the polylogarithmic terms and the hidden constant factors in the complexity. However, this algorithm assumes that we can efficiently generate independent samples from distributions approximating π_β for arbitrary β. The most efficient general algorithm known [50] for approximately sampling from arbitrary distributions π_β uses "warm starts" and hence does not produce independent samples.

Combining all the ingredients, we have the following result:

Corollary 13. *Let Z be a partition function and let $\epsilon > 0$ be a desired precision such that $\epsilon = O(1/\sqrt{\log\log A})$. Assume that for every inverse temperature β we have a Markov chain M_β with* *stationary distribution π_β and relaxation time upper-bounded by τ. Further assume that we can sample directly from M_0. Then, for any $\delta > 0$, there is a quantum algorithm which uses*

$$\begin{aligned}O(((\log A)(\log^2 n)(\log\log A)^2\sqrt{\tau}/\epsilon)\log^{5/2}((\log A)/\epsilon)\log((\log A)/\delta)\log\log((\log A)/\epsilon)\\ +(\log A)((\log n)+\log\log A)^5\tau\log(1/\delta)))\\ =\widetilde{O}((\log A)\sqrt{\tau}(1/\epsilon+\sqrt{\tau}))\end{aligned}$$

steps of the M_β chains and their corresponding quantum walk operations, and outputs $\widetilde{Z}$ such that

$$\Pr[(1-\epsilon)Z(\infty) \le \widetilde{Z} \le (1+\epsilon)Z(\infty)] \ge 1-\delta.$$

The best comparable classical result known is $\widetilde{O}((\log A)\tau/\epsilon^2)$ [50]. We therefore see that we have achieved a near-quadratic reduction in the complexity with respect to both τ and ϵ, assuming that $\epsilon \le 1/\sqrt{\tau}$. Otherwise, we still achieve a near-quadratic reduction with respect to ϵ.

Note that, if we could find a quantum algorithm that outputs a Chebyshev cooling schedule using $\widetilde{O}((\log A)\sqrt{\tau})$ steps of the Markov chains, Corollary 13 would be improved to a complexity of $\widetilde{O}((\log A)\sqrt{\tau}/\epsilon)$. It is instructive to note why this does not seem to be immediate. The classical algorithm for this problem [50] needs to approximately sample from Markov chains M_β for arbitrary values of β. To do this, it starts by fixing a nonadaptive Chebyshev cooling schedule $0 < \beta'_1 < \beta'_2 < \cdots < \beta'_\ell = \infty$ of length $\ell = \widetilde{O}(\log A)$. When the algorithm wants to sample from M_β with $\beta'_i < \beta < \beta'_{i+1}$, the algorithm uses an approximate sample from $M_{\beta'_i}$ as a "warm start". To produce one sample corresponding to each β'_i value requires $\widetilde{O}(\ell\tau)$ samples, because each $M_{\beta'_i}$ also provides a warm start for $M_{\beta'_{i+1}}$. But, in the quantum case, this does not work because, by no-cloning, the states $|\pi_{\beta'_i}\rangle$ cannot be reused in this way to provide warm starts for multiple runs of the algorithm.

3.4 Some partition function problems

In this section we describe some representative applications of our results to problems in statistical physics and computer science.

The ferromagnetic Ising model. This well-studied statistical physics model is defined in terms of a graph $G = (V, E)$ by the Hamiltonian

$$H(z) = -\sum_{(u,v)\in E} z_u z_v,$$

where $|V| = n$ and $z \in \{\pm 1\}^n$. A standard method to approximate the partition function of the Ising model uses the Glauber dynamics. This is a simple Markov chain with state space $\{\pm 1\}^n$, each of whose transitions involves only updating individual sites, and whose stationary distribution is the Gibbs distribution

$$\pi_\beta(z) = \frac{1}{Z(\beta)} e^{-\beta H(z)}.$$

This Markov chain, which has been intensively studied for decades, is known to mix rapidly in certain regimes [41]. Here we mention just one representative recent result:

Theorem 14 (Mossel and Sly [44])**.** *For any integer $d > 2$, and inverse temperature $\beta > 0$ such that $(d-1)\tanh\beta < 1$, the mixing time of the Glauber dynamics on any graph of maximum degree d is $O(n\log n)$.*

(More precise results than Theorem 14 are known for certain specific graphs such as lattices [42].) As we have $A = 2^n$, in the regime where $(d-1)\tanh\beta < 1$ the quantum algorithm approximates $Z(\beta)$ to within ϵ relative error in $\widetilde{O}(n^{3/2}/\epsilon + n^2)$ steps. The fastest known classical algorithm with rigorously proven performance bounds [50] uses time $\widetilde{O}(n^2/\epsilon^2)$. We remark that an alternative approach of Jerrum and Sinclair [34], which is based on analysing a different Markov chain, gives a polynomial-time classical algorithm which works for any temperature, but is substantially slower.

Counting colourings. Here we are given as input a graph G with n vertices and maximum degree d. We seek to approximately count the number of valid k-colourings of G, where a colouring of the vertices is valid if all pairs of neighbouring vertices are assigned different colours, and $k = O(1)$. In physics, this problem corresponds to the partition function of the Potts model evaluated at zero temperature. It is known that the Glauber dynamics for the Potts model mixes rapidly in some cases [20]. One particularly clean result of this form is work of Jerrum [31] showing that this Markov chain mixes in time $O(n \log n)$ if $k > 2d$. As here $A = k^n$, we obtain a quantum algorithm approximating the number of colourings of G up to relative error ϵ in $\widetilde{O}(n^{3/2}/\epsilon + n^2)$ steps, as compared with the classical $\widetilde{O}(n^2/\epsilon^2)$ [50].

Counting matchings. A matching in a graph G is a subset M of the edges of G such that no pair of edges in M shares a vertex. In statistical physics, matchings are often known as monomer-dimer coverings [26]. To count the number of matchings, we consider the partition function

$$Z(\beta) = \sum_{M \in \mathcal{M}} e^{-\beta |M|},$$

where $\mathcal{M}$ is the set of matchings of G. We have $Z(0) = |\mathcal{M}|$, while $Z(\infty) = 1$, as in this case the sum is zero everywhere except the empty matching ($0^0 = 1$). Therefore, in this case we seek to approximate $Z(0)$ using a telescoping product which starts with $Z(\infty)$. In terms of the cooling schedule $0 = \beta_0 < \beta_1 < \cdots < \beta_\ell = \infty$, we have

$$Z(\beta_0) = Z(\beta_\ell) \frac{Z(\beta_{\ell-1})}{Z(\beta_\ell)} \frac{Z(\beta_{\ell-2})}{Z(\beta_{\ell-1})} \cdots \frac{Z(\beta_0)}{Z(\beta_1)}.$$

As we have reversed our usage of the cooling schedule, rather than looking for it to be a B-Chebyshev cooling schedule we instead seek the bound

$$\frac{Z(2\beta_i - \beta_{i+1}) Z(\beta_{i+1})}{Z(\beta_i)^2} \leq B$$

to hold for all $i = 0, \dots, \ell - 1$. That is, the roles of β_i and β_{i+1} have been reversed as compared with (1). However, the classical algorithm for printing a cooling schedule can be modified to output a "reversible" schedule where this constraint is satisfied too, with only a logarithmic increase in complexity [50]. In addition, it was shown by Jerrum and Sinclair [33, 32] that, for any β, there is a simple Markov chain which has stationary distribution π, where

$$\pi(M) = \frac{1}{Z(\beta)} \sum_{M \in \mathcal{M}} e^{-\beta |M|},$$

and which has relaxation time $\tau = O(nm)$ on a graph with n vertices and m edges. Finally, in the setting of matchings, $A = O(n! 2^n)$. Putting these parameters together, we obtain a quantum complexity $\widetilde{O}(n^{3/2} m^{1/2}/\epsilon + n^2 m)$, as compared with the lowest known classical bound $\widetilde{O}(n^2 m/\epsilon^2)$ [50].

4 Estimating the total variation distance

Here we give the technical details of our improvement of the accuracy of a quantum algorithm of Bravyi, Harrow and Hassidim [13] for estimating the total variation distance between probability distributions. In this setting, we are given the ability to sample from probability distributions p and q on n elements, and would like to estimate $\|p-q\| := \frac{1}{2}\|p-q\|_1 = \frac{1}{2}\sum_{x\in[n]}|p(x)-q(x)|$ up to additive error ϵ. Classically, estimating $\|p-q\|$ up to error, say, 0.01 cannot be achieved using $O(n^\alpha)$ samples for any $\alpha < 1$ [54], but in the quantum setting the dependence on n can be improved quadratically:

Theorem 15 (Bravyi, Harrow and Hassidim [13])**.** *Given the ability to sample from p and q, there is a quantum algorithm which estimates $\|p-q\|$ up to additive error ϵ, with probability of success $1-\delta$, using $O(\sqrt{n}/(\epsilon^8\delta^5))$ samples.*

Here we will use Theorem 3 to improve the dependence on ϵ and δ of this algorithm. We will approximate the mean output value of the following algorithm, which was a subroutine previously used in [13].

Let p and q be probability distributions on n elements and let $r = (p+q)/2$.

1. Draw a sample $x \in [n]$ according to r.
2. Use amplitude estimation with t queries, for some t to be determined, to obtain estimates $\widetilde{p}(x)$, $\widetilde{q}(x)$ of the probability of obtaining outcome x under distributions p and q.
3. Output $|\widetilde{p}(x)-\widetilde{q}(x)|/(\widetilde{p}(x)+\widetilde{q}(x))$.

Algorithm 5: Subroutine for estimating the total variation distance

If the estimates $\widetilde{p}(x)$, $\widetilde{q}(x)$ were precisely accurate, the expected output of the subroutine would be

$$E := \sum_{x\in[n]}\left(\frac{p(x)+q(x)}{2}\right)\frac{|p(x)-q(x)|}{p(x)+q(x)} = \frac{1}{2}\sum_{x\in[n]}|p(x)-q(x)| = \|p-q\|.$$

We now bound how far the expected output $\widetilde{E}$ of the algorithm is from this exact value. By linearity of expectation,

$$|\widetilde{E}-E| = \left|\sum_{x\in[n]} r(x)\mathbb{E}[\widetilde{d}(x)-d(x)]\right| \leq \sum_{x\in[n]} r(x)\mathbb{E}[|\widetilde{d}(x)-d(x)|]$$

where $d(x) = |p(x)-q(x)|/(p(x)+q(x))$, $\widetilde{d}(x) = |\widetilde{p}(x)-\widetilde{q}(x)|/(\widetilde{p}(x)+\widetilde{q}(x))$. Note that $\widetilde{d}(x)$ is a random variable. Split $[n]$ into "small" and "large" parts according to whether $r(x) \leq \epsilon/n$. Then

$$\begin{aligned}|\widetilde{E}-E| &\leq \sum_{x,r(x)\leq\epsilon/n} r(x)\mathbb{E}[|\widetilde{d}(x)-d(x)|] + \sum_{x,r(x)\geq\epsilon/n} r(x)\mathbb{E}[|\widetilde{d}(x)-d(x)|]\\ &\leq \epsilon + \sum_{x,r(x)\geq\epsilon/n} r(x)\mathbb{E}[|\widetilde{d}(x)-d(x)|]\end{aligned}$$

using that $0 \leq d(x), \widetilde{d}(x) \leq 1$. From Theorem 2, for any $\delta > 0$ we have

$$|\widetilde{p}(x)-p(x)| \leq 2\pi\frac{\sqrt{p(x)}}{t} + \frac{\pi^2}{t^2}$$

except with probability at most δ, using $O(t\log 1/\delta)$ samples from p. If $t \geq 4\pi/(\eta\sqrt{p(x)+q(x)})$ for some $0 \leq \eta \leq 1$, this implies that

$$|\widetilde{p}(x) - p(x)| \leq \frac{2\pi\eta\sqrt{p(x)}\sqrt{p(x)+q(x)}}{4\pi} + \frac{\pi^2\eta^2(p(x)+q(x))}{16\pi^2} \leq \eta(p(x)+q(x))$$

except with probability at most δ. A similar claim also holds for $|\widetilde{q}(x) - q(x)|$. We now use the following technical result from [13]:

Proposition 16. *Consider a real-valued function $f(p,q) = (p-q)/(p+q)$ where $0 \leq p,q \leq 1$. Assume that $|p-\widetilde{p}|, |q-\widetilde{q}| \leq \eta(p+q)$ for some $\eta \leq 1/5$. Then $|f(p,q) - f(\widetilde{p},\widetilde{q})| \leq 5\eta$.*

By Proposition 16, for all x such that $t \geq 4\pi/(\eta\sqrt{p(x)+q(x)})$ we have $|\widetilde{d}(x) - d(x)| \leq 5\eta$, except with probability at most 2δ. We now fix $t = \lceil 10\sqrt{2}\pi\sqrt{n}\epsilon^{-3/2}\rceil$. Then, for all x such that $p(x)+q(x) \geq 2\epsilon/n$, $|\widetilde{d}(x) - d(x)| \leq \epsilon$ except with probability at most 2δ. Thus, for all x such that $r(x) \geq \epsilon/n$,

$$\mathbb{E}[|\widetilde{d}(x) - d(x)|] \leq 2\delta + (1-2\delta)\epsilon \leq 2\delta + \epsilon.$$

Taking $\delta = \epsilon$, we have

$$|\widetilde{E} - E| \leq 4\epsilon$$

for any ϵ, using $O(\sqrt{n}\epsilon^{-3/2}\log(1/\epsilon))$ samples. It therefore suffices to use $O(\sqrt{n}\epsilon^{-3/2}\log(1/\epsilon))$ samples to achieve $|\widetilde{E} - E| \leq \epsilon/2$. As the output of this subroutine is bounded between 0 and 1, to approximate $\widetilde{E}$ up to additive error $\epsilon/2$ with failure probability δ, it suffices to use the subroutine $O((1/\epsilon)\log(1/\delta))$ times by Theorem 3. So the overall complexity is $O((\sqrt{n}/\epsilon^{5/2})\log(1/\epsilon)\log(1/\delta))$. For small ϵ and δ this is a substantial improvement on the $O(\sqrt{n}/(\epsilon^8\delta^5))$ complexity stated by Bravyi, Harrow and Hassidim [13].

Acknowledgements

This work was supported by the UK EPSRC under Early Career Fellowship EP/L021005/1. I would like to thank Aram Harrow for helpful conversations and pointing out references, and Daniel Lidar for supplying further references. I would also like to thank several anonymous referees for their helpful comments. Special thanks to Tongyang Li for pointing out an error in Section 4.

A Stability of Algorithm 3

It is often the case that one wishes to estimate some quantity of interest defined in terms of samples from some probability distribution π, but can only sample from a distribution $\widetilde{\pi}$ which is close to π in total variation distance (for example, using Markov chain Monte Carlo methods). We now show that, if Algorithm 3 is given access to samples from $\widetilde{\pi}$ rather than π, it does not notice the difference. We will need the following claim.

Claim 17. *For any $x, y \in [0,1]$,*

$$|\arcsin x - \arcsin y| \leq \frac{\pi}{2}\sqrt{|x^2 - y^2|}.$$

Proof. We use a standard addition formula for arcsin to obtain

$$\begin{aligned} |\arcsin x - \arcsin y| &= |\arcsin(x\sqrt{1-y^2} - y\sqrt{1-x^2})| \\ &\leq \frac{\pi}{2}|\sqrt{x^2(1-y^2)} - \sqrt{y^2(1-x^2)}| \\ &\leq \frac{\pi}{2}\sqrt{|x^2-y^2|}, \end{aligned}$$

where the first inequality is $\sin\theta \geq (2/\pi)\theta$ for all $\theta \in [0, \pi/2]$, and the second inequality is

$$|a-b| \leq \sqrt{|a-b|(a+b)} = \sqrt{|a^2-b^2|},$$

valid for all non-negative a and b. □

Lemma 18. *Let $\mathcal{A}$ and $\mathcal{B}$ be algorithms with distributions $\mathcal{D}_{\mathcal{A}}$ and $\mathcal{D}_{\mathcal{B}}$ on their output values, such that $\|\mathcal{D}_{\mathcal{A}} - \mathcal{D}_{\mathcal{B}}\| \leq \gamma$, for some γ. Assume that Algorithm 3 is applied to $\mathcal{A}$, and uses the operator $U = 2|\psi\rangle\langle\psi| - I$ T times, where $|\psi\rangle = \mathcal{A}|0\rangle$. Then the algorithm estimates $\mathbb{E}[v(\mathcal{B})]$ up to additive error ϵ except with probability at most $3/10 + \frac{\pi^2}{\sqrt{6}}T\sqrt{\gamma}$.*

Lemma 18 is reminiscent of the hybrid argument for proving lower bounds on quantum query complexity [8]: if the distributions $\mathcal{D}_{\mathcal{A}}$ and $\mathcal{D}_{\mathcal{B}}$ are close, and the amplitude amplification algorithm makes few queries, it cannot distinguish them. However, here the quantifiers appear in a different order: whereas in the setting of lower-bounding quantum query complexity we wish to show that there exist pairs of distributions which are indistinguishable by any possible algorithm, here we wish to show that one fixed algorithm cannot distinguish any pair of close distributions.

Also note that Wocjan et al. [59] proved a similar result in the setting where we are given access to an approximate rotation $\widetilde{U} \approx U$. However, the result here is more general, in that we do not assume that $|\phi\rangle = \mathcal{B}|0\rangle$ is close to $|\psi\rangle$, but merely that the measured probability distributions are close.

Proof. We first use the calculations for the output probabilities of the amplitude estimation algorithm from [12] when applied as in Theorem 3 with t queries to an algorithm with mean output value μ_A, and another with mean output value μ_B.

For $x, y \in \mathbb{R}$, define $d(x,y) = \min_{z\in\mathbb{Z}} |z+x-y|$. $2\pi d(x,y)$ is the length of the shortest arc on the unit circle between $e^{2\pi i x}$ and $e^{2\pi i y}$. Let ω_A and ω_B be defined by $\sin^2\omega_A = \mu_A$, $\sin^2\omega_B = \mu_B$, and set $\Delta = d(\omega_A, \omega_B)$. Finally, let $\mathcal{M}_{\mathcal{A}}$ and $\mathcal{M}_{\mathcal{B}}$ be the distributions over the measurement outcomes when amplitude estimation is applied to estimate μ_A, μ_B.

The distribution on the measurement outcomes of the amplitude estimation algorithm after t uses of the input operator, when applied to a phase of ω, is equivalent [12] to that obtained by measuring the state

$$|\mathcal{S}_t(\omega)\rangle := \frac{1}{\sqrt{t}}\sum_{y\in[t]} e^{2\pi i\omega y}|y\rangle,$$

so the total variation distance between the distributions $\mathcal{M}_{\mathcal{A}}$ and $\mathcal{M}_{\mathcal{B}}$ obeys the bound

$$\|\mathcal{M}_{\mathcal{A}} - \mathcal{M}_{\mathcal{B}}\|^2 \leq 1 - |\langle\mathcal{S}_t(\omega_A)|\mathcal{S}_t(\omega_B)\rangle|^2 = 1 - \frac{\sin^2(\pi t\Delta)}{t^2\sin^2(\pi\Delta)},$$

where the first equality is standard [46] and the second equality is [12, Lemma 10]. Using the inequalities

$$\theta - \frac{\theta^3}{6} \leq \sin\theta \leq \theta,$$

valid for $\theta \geq 0$, we obtain

$$\|\mathcal{M}_{\mathcal{A}} - \mathcal{M}_{\mathcal{B}}\|^2 \leq 1 - \left(\frac{\pi t\Delta - (\pi t\Delta)^3/6}{t\pi\Delta}\right)^2 = 1 - \left(1 - \frac{(\pi t\Delta)^2}{6}\right)^2 \leq \frac{(\pi t\Delta)^2}{3}.$$

As we have

$$\Delta = \min_{z\in\mathbb{Z}} |z + \omega_A - \omega_B| \leq |\omega_A - \omega_B| \leq \frac{\pi}{2}\sqrt{|\mu_A - \mu_B|}$$

by Claim 17, we have

$$\|\mathcal{M}_{\mathcal{A}} - \mathcal{M}_{\mathcal{B}}\| \leq \frac{\pi^2}{2\sqrt{3}} t\sqrt{|\mu_A - \mu_B|}.$$

Within Algorithm 2, Theorem 3 is applied to $v(\mathcal{A}_{2^{\ell-1},2^\ell})/2^\ell$ for various values of ℓ. We have

$$\begin{aligned}|\mathbb{E}[v(\mathcal{A}_{2^{\ell-1},2^\ell})/2^\ell] - \mathbb{E}[v(\mathcal{B}_{2^{\ell-1},2^\ell})/2^\ell]| &= \frac{1}{2^\ell}\sum_{2^{\ell-1}\leq x<2^\ell} x|\Pr[v(\mathcal{A}) = x] - \Pr[v(\mathcal{B}) = x]| \\ &\leq \sum_x |\Pr[v(\mathcal{A}) = x] - \Pr[v(\mathcal{B}) = x]| \\ &= 2\|\mathcal{D}_{\mathcal{A}} - \mathcal{D}_{\mathcal{B}}\| \leq 2\gamma.\end{aligned}$$

Thus, for each run of the algorithm which uses $\mathcal{A}$ t times,

$$\|\mathcal{M}_{\mathcal{A}} - \mathcal{M}_{\mathcal{B}}\| \leq \frac{\pi^2}{\sqrt{3}} t\sqrt{\gamma}.$$

This is equivalent to the output of the algorithm being a probabilistic mixture of $\mathcal{M}_{\mathcal{B}}$ and some other distribution $\mathcal{M}$, where the probability of it being $\mathcal{M}$ is at most $\frac{\pi^2}{\sqrt{3}} t\sqrt{\gamma}$.

Algorithm 3 uses $\mathcal{A}$ T times in total. Each use of $\mathcal{A}$ is either within Algorithm 2 or one separate sample from $v(\mathcal{A})$ in Algorithm 3. We can similarly think of this sample as being taken from $\mathcal{B}$, except with probability at most $\gamma \leq \frac{\pi^2}{\sqrt{3}}\sqrt{\gamma}$. Taking a union bound over all uses of $\mathcal{A}$, we get the claimed bound. □

References

[1] D. Abrams and C. Williams. Fast quantum algorithms for numerical integrals and stochastic processes, 1999. `quant-ph/9908083`.

[2] D. Aharonov. Quantum computation. In *Annual Reviews of Computational Physics VI*, chapter 7, pages 259–346. World Scientific, 1998. `quant-ph/9812037`.

[3] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani. Quantum walks on graphs. In *Proc. 33rd Annual ACM Symp. Theory of Computing*, pages 50–59, 2001. `quant-ph/0012090`.

[4] D. Aharonov, I. Arad, E. Eban, and Z. Landau. Polynomial quantum algorithms for additive approximations of the Potts model and other points of the Tutte plane, 2007. `quant-ph/0702008`.

[5] D. Aharonov, A. Kitaev, and N. Nisan. Quantum circuits with mixed states. In *Proc. 30th Annual ACM Symp. Theory of Computing*, pages 20–30, 1998.

[6] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation. *SIAM J. Comput.*, 37(1):47–82, 2007. `quant-ph/0301023`.

[7] I. Arad and Z. Landau. Quantum computation and the evaluation of tensor networks. *SIAM J. Comput.*, 39:3089–3121, 2010. `arXiv:0805.0040`.

[8] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. `quant-ph/9701001`.

[9] D. Berry, A. Childs, and R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Proc. 56th Annual Symp. Foundations of Computer Science*, pages 792–809, 2015. `arXiv:1501.01715`.

[10] I. Bezáková, D. Štefankovič, V. Vazirani, and E. Vigoda. Accelerating simulated annealing for the permanent and combinatorial counting problems. *SIAM J. Comput.*, 37(5):1429–1454, 2008.

[11] G. Brassard, F. Dupuis, S. Gambs, and A. Tapp. An optimal quantum algorithm to approximate the mean and its application for approximating the median of a set of points over an arbitrary distance, 2011. `arXiv:1106.4267`.

[12] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. *Quantum Computation and Quantum Information: A Millennium Volume*, pages 53–74, 2002. `quant-ph/0005055`.

[13] S. Bravyi, A. W. Harrow, and A. Hassidim. Quantum algorithms for testing properties of distributions. *IEEE Trans. Inform. Theory*, 57(6):3971–3981, 2011. `arXiv:0907.3920`.

[14] C.-F. Chiang, D. Nagaj, and P. Wocjan. Efficient circuits for quantum walks. *Quantum Inf. Comput.*, 10(5&6):420–424, 2010. `arXiv:0903.3465`.

[15] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for Monte Carlo estimation. *SIAM J. Comput.*, 29(5):1484–1496, 2000.

[16] G. De las Cuevas, W. Dür, M. van den Nest, and M. Martin-Delgado. Quantum algorithms for classical lattice models. *New J. Phys.*, 13:093021, 2011. `arXiv:1104.2517`.

[17] N. Destainville, B. Georgeot, and O. Giraud. Quantum algorithm for exact Monte Carlo sampling. *Phys. Rev. Lett.*, 104:250502, 2010. `arXiv:1003.1862`.

[18] V. Dunjko and H. Briegel. Sequential quantum mixing for slowly evolving sequences of Markov chains, 2015. `arXiv:1503.01334`.

[19] M. Dyer and A. Frieze. Computing the volume of convex bodies: a case where randomness provably helps. In *Probabilistic Combinatorics and Its Applications*, volume 44 of *Proceedings of Symposia in Applied Mathematics*, pages 123–170. American Mathematical Society, 1992.

[20] A. Frieze and E. Vigoda. A survey on the use of Markov chains to randomly sample colourings. In *Combinatorics, Complexity and Chance*, pages 53–71. Oxford University Press, 2007.

[21] J. Geraci and D. Lidar. On the exact evaluation of certain instances of the Potts partition function by quantum computers. *Comm. Math. Phys.*, 279:735–768, 2008. `quant-ph/0703023`.

[22] J. Geraci and D. Lidar. Classical Ising model test for quantum circuits. *New J. Phys.*, 12:075026, 2010. `arXiv:0902.4889`.

[23] P. Glasserman. *Monte Carlo methods in financial engineering.* Springer, New York, 2003.

[24] L. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79(2):325–328, 1997. `quant-ph/9706033`.

[25] L. Grover. A framework for fast quantum mechanical algorithms. In *Proc. 30th Annual ACM Symp. Theory of Computing*, pages 53–62, 1998. `quant-ph/9711043`.

[26] O. Heilmann and E. Lieb. Theory of monomer-dimer systems. *Comm. Math. Phys.*, 25:190–232, 1972.

[27] S. Heinrich. Quantum summation with an application to integration. *Journal of Complexity*, 18(1):1–50, 2001. `quant-ph/0105116`.

[28] M. Huber. Approximation algorithms for the normalizing constant of Gibbs distributions, 2012. `arXiv:1206.2689`.

[29] M. Huber. Improving Monte Carlo randomized approximation schemes, 2014. `arXiv:1411.4074`.

[30] C. Jacoboni and P. Lugli. *The Monte Carlo method for semiconductor device simulation.* Springer-Verlag, Wien-New York, 1989.

[31] M. Jerrum. A very simple algorithm for estimating the number of k-colourings of a low-degree graph. *Random Structures and Algorithms*, 7(2):157–165, 1995.

[32] M. Jerrum. *Counting, sampling and integrating: algorithms and complexity.* Birkhäuser Verlag, Basel, 2003.

[33] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989.

[34] M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM J. Comput.*, 22(5):1087–1116, 1993.

[35] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43(2–3):169–188, 1986.

[36] E. Knill, G. Ortiz, and R. Somma. Optimal quantum measurements of expectation values of observables. *Phys. Rev. A*, 75:012328, 2007. `quant-ph/0607019`.

[37] W. Krauth. *Statistical Mechanics: Algorithms and Computations.* Oxford University Press, Oxford, 2006.

[38] D. Levin, Y. Peres, and E. Wilmer. *Markov chains and mixing times.* American Mathematical Society, 2009.

[39] D. Lidar. On the quantum computational complexity of the Ising spin glass partition function and of knot invariants. *New J. Phys.*, 6:167, 2004. `quant-ph/0309064`.

[40] D. Lidar and O. Biham. Simulating Ising spin glasses on a quantum computer. *Phys. Rev. E*, 56:3661, 1997. `quant-ph/9611038`.

[41] F. Martinelli. Lectures on Glauber dynamics for discrete spin models. In *Lectures on probability theory and statistics (Saint-Flour, 1997)*, volume 1717 of *Lecture Notes in Mathematics*, pages 93–191. Springer, 1997.

[42] F. Martinelli and E. Olivieri. Approach to equilibrium of Glauber dynamics in the one phase region. *Comm. Math. Phys.*, 161(3):447–486, 1994.

[43] A. Matsuo, K. Fujii, and N. Imoto. Quantum algorithm for an additive approximation of Ising partition functions. *Phys. Rev. A*, 90:022304, 2014. `arXiv:1405.2749`.

[44] E. Mossel and A. Sly. Exact thresholds for Ising-Gibbs samplers on general graphs. *The Annals of Probability*, 41(1):294–328, 2013.

[45] A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. In *Proc. 31st Annual ACM Symp. Theory of Computing*, pages 384–393, 1999. `quant-ph/9804066`.

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

[47] D. Poulin and P. Wocjan. Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer. *Phys. Rev. Lett.*, 103:220502, 2009. `0905.2199`.

[48] P. Richter. Quantum speedup of classical mixing processes. *Phys. Rev. A*, 76:042306, 2007. `quant-ph/0609204`.

[49] R. Somma, S. Boixo, H. Barnum, and E. Knill. Quantum simulations of classical annealing processes. *Phys. Rev. Lett.*, 101(13):130504, 2008. `arXiv:0804.1571`.

[50] D. Štefankovič, S. Vempala, and E. Vigoda. Adaptive simulated annealing: a new connection between sampling and counting. *J. ACM*, 56(3):18:1–18:36, 2009. `cs.DS/0612058`.

[51] M. Szegedy. Quantum speed-up of Markov chain based algorithms. In *Proc. 45th Annual Symp. Foundations of Computer Science*, pages 32–41, 2004. `quant-ph/0401053`.

[52] K. Temme, T. Osborne, K. Vollbrecht, D. Poulin, and F. Verstraete. Quantum Metropolis sampling. *Nature*, 471:87–90, 2011. `arXiv:0911.3635`.

[53] R. Tucci. Use of quantum sampling to calculate mean values of observables and partition function of a quantum system, 2009. `arXiv:0912.4402`.

[54] P. Valiant. Testing symmetric properties of distributions. *SIAM J. Comput.*, 40(6):1927–1968, 2011.

[55] J. Valleau and D. Card. Monte Carlo estimation of the free energy by multistage sampling. *J. Chem. Phys.*, 57:5457, 1972.

[56] M. Van den Nest, W. Dür, R. Raussendorf, and H. Briegel. Quantum algorithms for spin models and simulable gate sets for quantum computation. *Phys. Rev. A*, 80:052334, 2008. `arXiv:0805.1214`.

[57] S. Venegas-Andraca. Quantum walks: a comprehensive review. *Quantum Information Processing*, 11(5):1015–1106, 2012. `arXiv:1201.4780`.

[58] P. Wocjan and A. Abeyesinghe. Speedup via quantum sampling. *Phys. Rev. A*, 78:042336, 2008. `arXiv:0804.4259`.

[59] P. Wocjan, C.-F. Chang, D. Nagaj, and A. Abeyesinghe. Quantum algorithm for approximating partition functions. *Phys. Rev. A*, 80:022340, 2009. `arXiv:0811.0596`.

[60] M.-H. Yung and A. Aspuru-Guzik. A quantum-quantum Metropolis algorithm. *Proceedings of the National Academy of Sciences*, 109(3):754–759, 2012. `arXiv:1011.1468`.