Quantum circuits and low-degree polynomials over $\mathbb{F}_2$

Ashley Montanaro*

July 28, 2016

Abstract

In this work we explore a correspondence between quantum circuits and low-degree polynomials over the finite field $\mathbb{F}_2$. Any quantum circuit made up of Hadamard, Z, controlled-Z and controlled-controlled-Z gates gives rise to a degree-3 polynomial over $\mathbb{F}_2$ such that calculating quantum circuit amplitudes is equivalent to counting zeroes of the corresponding polynomial. We exploit this connection, which is especially clean and simple for this particular gate set, in two directions. First, we give proofs of classical hardness results based on quantum circuit concepts. Second, we find efficient classical simulation algorithms for certain classes of quantum circuits based on efficient algorithms for classes of polynomials.

1 Introduction

Quantum computers are believed to outperform classical computers for important tasks as varied as simulation of quantum mechanics and factorisation of large integers. Although no large-scale general-purpose quantum computer has been built as yet, quantum computation can nevertheless already be used as a theoretical tool to study other areas of science and mathematics, without the need for an actual quantum computer.

This work explores a simple correspondence between quantum circuits and low-degree polynomials over the finite field $\mathbb{F}_2$, i.e. the integers modulo 2. By picking the right gate set, it turns out that quantum circuit amplitudes have a close connection to counting zeroes of such polynomials. This correspondence can be exploited in two directions. On the one hand, ideas about quantum circuits can be used to prove purely classical results regarding the computational complexity of counting zeroes of polynomials over finite fields. On the other, known classical results about polynomials can be used to give new algorithms for simulating classes of quantum circuits.

A similar perspective has been taken by a number of previous works. Particularly relevant is prior work of Dawson et al. [13], who showed that quantum circuit amplitudes for circuits of Toffoli and Hadamard gates can be understood in terms of solutions to systems of polynomial equations involving low-degree polynomials over $\mathbb{F}_2$. Here we use a slightly different universal gate set: Hadamard ($= \frac{1}{\sqrt{2}} \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$), Z ($= \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$), controlled-Z ("CZ") and controlled-controlled-Z ("CCZ"). This is essentially equivalent to the gate set of [13], as Toffoli gates are identical to CCZ gates conjugated by a Hadamard gate on the target qubit. However, this small shift in perspective seems to simplify and clarify some of the arguments involved. For example, the connection we use associates a single polynomial with each circuit. Related ideas to [13] were used by Rudolph [36] to give a simple encoding of quantum circuit amplitudes as matrix permanents. The set of circuits

*School of Mathematics, University of Bristol, UK; ashley.montanaro@bristol.ac.uk.

we consider is a very special case of the class of "algebraic quantum circuits" studied by Bacon, van Dam and Russell [6] in some generality.

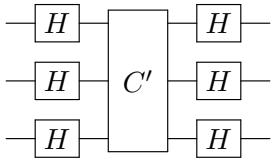
The idea of proving classical results using quantum methods has also been explored previously; see [14] for a survey of many results in this area. Within computational complexity alone, three relevant examples are Aaronson's proof of the computational hardness of computing the matrix permanent using the close connection between the permanent and linear-optical quantum circuits [1]; Kuperberg's proof of the computational hardness of approximately computing Jones polynomials by expressing these in terms of quantum circuits [31]; and Fujii and Morimae's proof of hardness of computing Ising model partition functions, again based on quantum circuits over a suitable gate set [19]. More recently, together with Bremner and Shepherd [11], the present author used a correspondence between low-degree polynomials and a certain class of simple quantum computations, known as IQP circuits [37], to argue that random IQP circuits are unlikely to be efficiently simulable classically. This holds even if the classical simulator is allowed to be approximate, with a fairly generous notion of approximation.

The correspondence between low-degree polynomials and quantum circuits which we investigate here seems particularly simple and direct. We have therefore tried to use it to highlight some of the beautiful ideas present in previous works, and to produce an accessible introduction to computational complexity issues suitable for physicists; and also an introduction suitable for computer scientists to how one can prove classical results using the quantum circuit model.

We begin by introducing the circuit-polynomial correspondence and proving its correctness, and go on to make some simple observations about this connection. Then, in Section 3, we introduce the ideas from computational complexity that we will need, and in Section 3.1 show that the correspondence can be used to prove classical hardness of exactly computing the number of zeroes of low-degree polynomials. Similarly, in Section 3.2 we show that approximate computation of this quantity is closely related to quantum computation. We study a new complexity measure for polynomials motivated by this correspondence – the quantum circuit width – in Section 3.3. Then, in Section 4, we use the circuit-polynomial correspondence to give two simple classical simulation algorithms for classes of quantum circuits: circuits with few CCZ gates (or where the degree-3 part of the polynomial corresponding to the circuit has a small "hitting set", qv), and circuits whose corresponding polynomial can be simplified by a linear transformation. We conclude in Section 5 with some open problems.

2 Circuits and polynomials

In this work, we consider quantum circuit amplitudes of the form $\langle 0|C|0\rangle$, where C is a unitary operator expressed as a circuit on ℓ qubits with $\mathrm{poly}(\ell)$ gates, and we write $|0\rangle = |0\rangle^{\otimes \ell}$ for conciseness throughout. The gates in C are picked from the set $\mathcal{F} = \{\text{Hadamard, Z, CZ, CCZ}\}$[1]. Using the gate set $\mathcal{F}$ will allow us to write $\langle 0|C|0\rangle$ in a particularly concise form. Assume that C begins and ends with a column of Hadamards, i.e. is of the form



[1]In fact, the Z and CZ gates are not necessary, as they can be produced from CCZ gates together with the use of ancillas, but it will be convenient to include them.

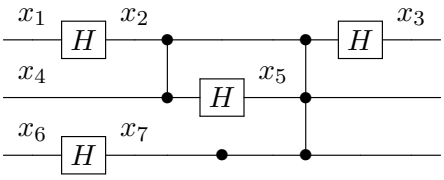
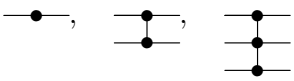


Figure 1: The internal part C' of a circuit C corresponding to the polynomial $x_1x_2 + x_2x_3 + x_4x_5 + x_6x_7 + x_2x_4 + x_2x_5x_7 + x_7$.

for some circuit C'. This is without loss of generality, as we can always add pairs of Hadamards to the beginning or end of each line without changing the unitary operator corresponding to the circuit. Further assume that C' contains at least one gate acting on each qubit. Let h be the number of internal Hadamard gates that C contains, i.e. the number of Hadamards in C'. Set $n = h + \ell$ and define a polynomial $f_C : \{0,1\}^n \to \{0,1\}$ over $\mathbb{F}_2$ as follows. Divide each horizontal wire of the internal part C' into segments, with each segment corresponding to a portion of the wire which is either between two Hadamard gates or to the left/right of all the Hadamard gates. Associate a distinct variable x_i with each segment of each wire. Observe that there are exactly $h+\ell$ variables in total. Each Hadamard gate now joins two segments and associates their corresponding variables, and each Z, CZ, CCZ gate is associated with one, two or three (respectively) variables, corresponding to the segments on which it acts. For each set of variables $x_{i_1}, \ldots, x_{i_k}$ associated with each gate, add the corresponding term $x_{i_1} \ldots x_{i_k}$ to f_C. As we are working over $\mathbb{F}_2$, all addition and multiplication in f_C is taken modulo 2. Note that this procedure never produces polynomials of degree higher than 3.

As a simple example of this construction, consider the labelled circuit C' in Figure 1, where we use the notation



for Z, CZ, CCZ gates respectively.

We now show that the number of zeroes of the polynomial corresponding to C has a close connection to $\langle 0|C|0\rangle$. To be more precise, $\langle 0|C|0\rangle$ is proportional to $\mathrm{gap}(f_C)$, where the gap of a polynomial is the difference between the number of zeroes and ones of that polynomial:

$$\mathrm{gap}(f_C) := \sum_{x \in \{0,1\}^n} (-1)^{f_C(x)} = |\{x : f_C(x) = 0\}| - |\{x : f_C(x) = 1\}|.$$

A similar result was shown in [13] for circuits containing Hadamard and Toffoli gates. However, the argument here seems somewhat simpler. Although there are several ways that the following result can be proven, we choose to highlight a connection to the beautiful results of [10].

Proposition 1. *Let C be a quantum circuit on ℓ qubits consisting of Hadamard, Z, CZ and CCZ gates, starting and ending with a column of Hadamard gates, and containing h internal Hadamard gates. Then*

$$\langle 0|C|0\rangle = \frac{\mathrm{gap}(f_C)}{2^{h/2+\ell}}.$$

Proof. First consider the case where the internal part C' of C does not contain any Hadamard gates (as treated in [11, Appendix B]). Let Z_i denote a Z gate acting on the i'th qubit (and similarly CZ_{ij}, CCZ_{ijk}). Then, for any $x \in \{0,1\}^\ell$, $\langle x|Z_i|x\rangle = (-1)^{x_i}$, $\langle x|CZ_{ij}|x\rangle = (-1)^{x_ix_j}$,

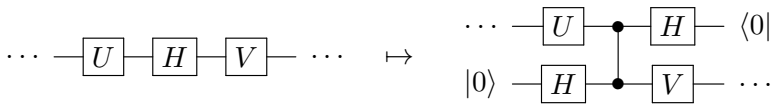


Figure 2: Replacing a Hadamard gate with a controlled-Z gate and postselection [10].

$\langle x|CCZ_{ijk}|x\rangle = (-1)^{x_i x_j x_k}$. As these gates are diagonal, we can obtain $\langle x|C'|x\rangle$ simply by multiplying the expressions $\langle x|G|x\rangle$ for different gates G in C'. Each gate corresponds to a term in f_C as defined above. So, for all $x \in \{0,1\}^\ell$, $\langle x|C'|x\rangle = (-1)^{f_C(x)}$, and hence

$$\langle 0|H^{\otimes \ell} C' H^{\otimes \ell}|0\rangle = \frac{1}{2^\ell} \sum_{x\in\{0,1\}^\ell} \langle x|C'|x\rangle = \frac{1}{2^\ell} \sum_{x\in\{0,1\}^\ell} (-1)^{f_C(x)} = \frac{\mathrm{gap}(f_C)}{2^\ell}.$$

We can remove any Hadamard gates in C' using a trick from [10]. Imagine we have a Hadamard gate on the i'th qubit. We form a new overall circuit C'' from C by introducing a new ancilla qubit a initialised in the state $|0\rangle$, replacing the Hadamard gate with the gadget $G = H_i CZ_{ai} H_a$, and changing all subsequent gates involving the i'th qubit to use qubit a (see Figure 2 for an illustration). Then, by direct calculation, $\langle 0|_i G|0\rangle_a = H/\sqrt{2}$, so $\langle 0|C''|0\rangle = \langle 0|C|0\rangle/\sqrt{2}$. Following this procedure for each of the h Hadamard gates in C', we obtain a circuit on $n = \ell + h$ qubits, where each Hadamard gate corresponds to a product of two variables and relabelling of a qubit as specified in the definition of f_C. Taking into account the normalisation factor of $2^{h/2}$, we obtain

$$\langle 0|C|0\rangle = \frac{1}{2^{h/2+\ell}} \sum_{x\in\{0,1\}^n} (-1)^{f_C(x)} = \frac{\mathrm{gap}(f_C)}{2^{h/2+\ell}}$$

as claimed. □

It is easy to check that the formula of Proposition 1 is accurate for the example in Figure 1 (where $\mathrm{gap}(f_C) = 16$ and $\langle 0|C|0\rangle = 1/2$). The correspondence between circuits and polynomials given in Proposition 1 will be the main tool used throughout this paper. We remark that all the other amplitudes $\langle x|C|y\rangle$, $x, y \in \{0,1\}^\ell$, are also related to polynomials. This is because X gates inserted at the start or end of C can be used to map $|0\rangle \mapsto |y\rangle$ or $|x\rangle \mapsto |0\rangle$, X gates can be commuted through Hadamard gates to produce Z gates, and Z gates give linear terms in the corresponding polynomial. Thus $\langle x|C|y\rangle = \mathrm{gap}(f_C + L_{x,y})/2^{h/2+\ell}$ for some linear function $L_{x,y}$ depending on x, y.

We next make some other simple observations that follow from the circuit-polynomial correspondence.

2.1 Basic observations

Observation 2. *There can be more than one quantum circuit C corresponding to a given polynomial f_C.*

Proof. There are two easy ways to see this. First, as Z, CZ and CCZ gates commute, a consecutive sequence of such gates in C can be reordered arbitrarily while still corresponding to the same polynomial f_C. Second, it is sometimes the case that CZ gates and Hadamards are interchangeable. For example, Figure 3 shows two circuits which both correspond to the polynomial $x_1 x_2$. □

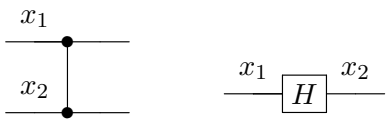


Figure 3: The internal part of two circuits which both correspond to the polynomial x_1x_2.

Observation 3. *For every degree-3 polynomial $f : \{0,1\}^n \to \{0,1\}$ with no constant term, there exists a quantum circuit C on n qubits such that $f = f_C$.*

Proof. Produce the internal part of a circuit C on n qubits by associating a qubit with each variable in f, and include a Z, CZ or CCZ gate between the qubits corresponding to each degree 1, 2, 3 term (respectively) in f. □

We remark that the class of quantum circuits produced from the procedure in Observation 3 are IQP circuits [37]. An IQP circuit ("Instantaneous Quantum Polynomial-time") on n qubits is a circuit of the form $H^{\otimes n}DH^{\otimes n}$, where D is a circuit of poly(n) diagonal gates. It was argued in [11] that it should be hard to sample classically from the output probability distributions of quantum circuits of the form of Observation 3, even up to small total variation distance. The argument was based on a plausible complexity-theoretic conjecture regarding the complexity of approximately computing $\mathrm{gap}(f)$ for random degree-3 polynomials f.

Observation 4. *There exists a degree-3 polynomial $f : \{0,1\}^n \to \{0,1\}$ such that every quantum circuit C corresponding to f requires n qubits.*

Proof. Consider the polynomial containing the term $x_ix_jx_k$ for all $1 \leq i < j < k \leq n$, and no other terms. As there are no degree-2 terms, any corresponding circuit C cannot contain any internal Hadamard gates. Thus C must act on at least n qubits, with one qubit corresponding to each variable. □

Observation 5. *If $f_C : \{0,1\}^n \to \{0,1\}$ corresponds to a quantum circuit C on ℓ qubits, then $|\mathrm{gap}(f_C)| \leq 2^{n/2+\ell/2}$.*

Proof. From Proposition 1, $\langle 0|C|0\rangle = \mathrm{gap}(f_C)/2^{h/2+\ell}$. As $\langle 0|C|0\rangle$ is a quantum circuit amplitude and hence bounded by 1 in absolute value by unitarity, $|\mathrm{gap}(f_C)| \leq 2^{h/2+\ell} = 2^{n/2+\ell/2}$. □

These observations motivate us to define the *quantum circuit width* $w(f)$ of a degree-3 polynomial f over $\mathbb{F}_2$ as the minimal number of qubits required for any quantum circuit which corresponds to f. For example, the family of polynomials f in Observation 4 has $w(f) = n$, whereas the polynomial $f' = x_1x_2 + x_2x_3 + \cdots + x_{n-1}x_n$ has $w(f') = 1$, corresponding to a circuit whose internal part consists of $n-1$ Hadamard gates applied to one qubit.

3 Computational complexity

The theory of computational complexity studies the inherent difficulty of computational problems. One of the main goals of this field is to classify problems into complexity classes: sets of problems of comparable difficulty. We now give a brief, informal introduction to this area; see [35, 5] for a full, formal treatment. The complexity classes used in this work can all be presented in terms of

determining properties of classical or quantum circuits. A classical circuit is a collection of AND, OR and NOT gates connected with wires, which map an input to an output by evaluating the gates in the natural manner. We assume that classical circuits only have one output bit, but potentially many input bits. For each classical circuit C, we let $C(x)$ be the output of C given the bit-string x as input. Then we can define the following natural problems:

- Circuit SAT: given a classical circuit C, determine whether there exists x such that $C(x) = 1$.
- Circuit Counting: given a classical circuit C, output $|\{x : C(x) = 1\}|$.

Each of these problems corresponds to a complexity class. NP ("nondeterministic polynomial-time") is the class of decision problems which reduce to Circuit SAT in polynomial time, while #P ("sharp-P" or "number-P") is the class of functional problems which can be expressed as an instance of Circuit Counting. The closely related class $\mathrm{P}^{\#\mathrm{P}}$ is the class of functional problems which can be solved in polynomial time, given the ability to solve any problem in the class #P. For example, the problem of computing $|\{x : C_1(x) = 1\}| - |\{x : C_2(x) = 1\}|$ for circuits C_1, C_2 is in $\mathrm{P}^{\#\mathrm{P}}$. Here "polynomial time" is short for "in time polynomial in the input size", which is the key notion of efficiency used in computational complexity. For any complexity class $\mathcal{C}$, a problem $\mathcal{P}$ is said to be $\mathcal{C}$-hard if it is at least as hard as every problem in $\mathcal{C}$: in other words, for every problem in $\mathcal{C}$, there is a polynomial-time reduction from that problem to $\mathcal{P}$.

A problem is said to be NP-complete if it is equivalent in difficulty to Circuit SAT, up to polynomial-time reductions. Many important practical problems (such as optimal packing and scheduling, integer programming, and computing ground-state energies of classical physical systems) are known to be NP-complete [21]. The famous $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ problem effectively asks whether Circuit SAT can be solved in time polynomial in the size of the given circuit. Although it is widely believed that the answer is "no", a positive answer would have momentous consequences, implying that any NP-complete problem could be solved in polynomial time. Observe that Circuit Counting is at least as hard as Circuit SAT. In fact, it is conjectured that this problem is much harder. Indeed, if there existed an efficient reduction from Circuit Counting to Circuit SAT, then the infinite tower of complexity classes known as the polynomial hierarchy would collapse [40], a consequence similar to P=NP and considered almost as unlikely.

Many interesting problems in physics and elsewhere are known to be #P-hard: at least as hard as any problem in #P. These include computing Ising model partition functions [24], evaluating Jones and Tutte polynomials [22], and exactly computing the permanent of a 0-1 matrix [41]. The intuitive reason behind the hardness of these problems is that they involve computing a sum of exponentially many terms. However, surprisingly, in some cases such sums can be computed efficiently (exactly or approximately). Examples include exact computation of Ising model partition functions on planar graphs [17, 26, 39], approximate computation of the permanent of a non-negative matrix [25], and Valiant's quantum-inspired "holographic algorithms" for combinatorial problems [42]. Proving #P-hardness of a problem provides strong evidence that a clever efficient algorithm like these should not exist for that problem.

3.1 Computational complexity of low-degree polynomials

We can use the connection between quantum circuits and polynomials to prove #P-hardness results. It was shown by Ehrenfeucht and Karpinski [16] that computing the number of zeroes (equivalently, the gap) of a degree-3 polynomial f over $\mathbb{F}_2$ is #P-hard. This implies that using the circuit-polynomial correspondence is unlikely to give an efficient algorithm for simulating all quantum

circuits classically by computing quantum circuit amplitudes. However, we can go in the other direction, and use the correspondence to obtain a quantum proof of #P-hardness of computing the number of zeroes of f (equivalently, computing $\mathrm{gap}(f)$).

Proposition 6. *It is #P-hard to compute* $\mathrm{gap}(f)$ *for degree-3 polynomials* f*.*

Proof. We will show that the problem of exactly computing $\langle 0|C|0\rangle$ for an arbitrary quantum circuit C containing Hadamard, Z, CZ, and CCZ gates is #P-hard. As computing $\mathrm{gap}(f)$ for arbitrary degree-3 polynomials f would allow us to compute $\langle 0|C|0\rangle$ for arbitrary circuits of this form, this will imply the claim. To achieve this, we first show that computing $\langle 0|C|0\rangle$ for an arbitrary quantum circuit C containing Hadamard, X and Toffoli gates is #P-hard. This can easily be obtained from a similar result of Van den Nest [43]; we include a simple direct proof here for completeness.

It is a fundamental result in the theory of reversible computation that X and Toffoli gates together with ancillas are universal for classical computation, i.e. that given a boolean function $g : \{0,1\}^n \to \{0,1\}$ computed by a classical circuit C of $\mathrm{poly}(n)$ gates, there is a quantum circuit C' of $\mathrm{poly}(n)$ X and Toffoli gates such that $C'|x\rangle_I|0\rangle_O|0\rangle_A^{\otimes a} = |x\rangle_I|g(x)\rangle_O|0\rangle_A^{\otimes a}$, where the circuit acts on a Hilbert space divided into an n-qubit input register I, a 1-qubit output register O, and an a-qubit ancilla register A. Then let the circuit C'' be defined as follows:

1. Apply an X gate to the O register.
2. Apply Hadamard gates to each qubit in the I and O registers.
3. Apply C'.
4. Apply Hadamard gates to each qubit in the I and O registers.
5. Apply an X gate to the O register.

If C'' is applied to the initial state $|0\rangle$, the state prepared after the second step is $|+\rangle_I^{\otimes n}|-\rangle_O|0\rangle_A^{\otimes a}$. When C' is applied in the third step the second and third registers are left unchanged, and the state of the first register becomes

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

Thus $\langle 0|C''|0\rangle = \langle +|^{\otimes n}|\psi_g\rangle = \frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{g(x)} = \mathrm{gap}(g)/2^n$. So computing $\langle 0|C''|0\rangle$ allows us to determine $\mathrm{gap}(g)$, and hence the number of zeroes of g, for functions g computed by arbitrary polynomial-size classical circuits. This problem is #P-hard by definition.

It remains to show that this same conclusion holds for circuits containing Hadamard, Z, CZ, and CCZ gates. But this is immediate, as Toffoli gates can be produced from CCZ gates by conjugating the target qubit by a Hadamard, and similarly $X = HZH$. □

The #P-hardness proof of Ehrenfeucht and Karpinski [16] is not difficult. However, the quantum proof gives a different perspective, and also lends itself to simple generalisations. For example:

Proposition 7. $\mathrm{gap}(f)$ *remains #P-hard to compute for degree-3 polynomials where each variable appears in at most 3 terms.*

Proof. We show that computing $\text{gap}(f)$ for an arbitrary degree-3 polynomial f reduces to computing $\text{gap}(f')$ for a degree-3 polynomial f' where each variable appears in at most 3 terms. Given f, we produce a corresponding quantum circuit C. Then, between each pair of gates, we insert two Hadamard gates on each qubit to produce a new circuit C'. As $H^2 = I$, $\langle 0|C'|0\rangle = \langle 0|C|0\rangle$, so the corresponding polynomial $f_{C'}$ satisfies $\text{gap}(f_{C'}) = \text{gap}(f_C)$, up to an easily computed scaling factor. But each variable in $f_{C'}$ is only contained within at most 3 terms, because the inserted Hadamard gates effectively relabel all the variables between each pair of terms in the polynomial. □

A similar circuit simplification to that of Proposition 7 was previously observed in [36]. Proposition 6 shows that we should not hope to find an efficient algorithm for simulating arbitrary quantum circuits by computing the number of zeroes of low-degree polynomials. However, for some classes of polynomials we can indeed obtain efficient algorithms (see below for some examples of this).

A natural question is whether we can improve Proposition 6 to show that even computing the number of zeroes of degree-2 polynomials is #P-hard. It was already shown by Ehrenfeucht and Karpinski [16] that this is unlikely to be the case, as there is a polynomial-time algorithm for this problem. There is an alternative "quantum" way of seeing this result, as relating to ideas around the well-known Gottesman-Knill theorem [34], which states that any quantum circuit whose gates are all picked from the Clifford group can be efficiently simulated classically. Indeed, for any degree-2 polynomial $f : \{0,1\}^n \to \{0,1\}$, by Observation 3 we can write down a quantum circuit C on n qubits containing only Hadamard, Z and CZ gates such that $\langle 0|C|0\rangle = \text{gap}(f)/2^n$. As the gates in C are all members of the Clifford group, the state $C|0\rangle$ is a stabilizer state, as is the state $|0\rangle$. It is known that the inner product between two arbitrary stabilizer states can be computed in time $O(n^3)$ [2, 20, 9], implying an $O(n^3)$ algorithm for computing $\text{gap}(f)$ for degree-2 polynomials $f : \{0,1\}^n \to \{0,1\}$.

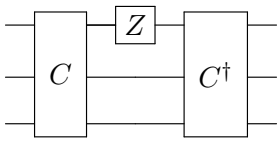
3.2 Approximate computation

Given that we have shown exactly computing $\text{gap}(f)$ to be hard, the next natural question is whether we can approximately compute it. We now show that this question is closely connected to *quantum* computational complexity. The class of decision problems which can be solved efficiently by a quantum computer (i.e. in time polynomial in the size of the input), with success probability $2/3$, is known as BQP [45]. As with the classical complexity classes discussed previously, BQP can be expressed in terms of circuits; however, the circuits are now quantum. Any polynomial-time quantum computation solving a decision problem can be expressed as applying some quantum circuit U, generated from the input in polynomial time, to the initial state $|0\rangle$, then measuring the first qubit, and returning the measurement result.

Proposition 8. *Determining* $\text{gap}(f)$ *for arbitrary degree-3 polynomials* $f : \{0,1\}^n \to \{0,1\}$ *up to absolute error* $\frac{1}{3} \cdot 2^{(n+w(f))/2}$ *is BQP-hard.*

Proof. We first recall that solving decision problems reduces to computing quantum circuit amplitudes (this is an observation of Knill and Laflamme [29]). Assume that we are given some quantum circuit C containing only Hadamard, Z, CZ and CCZ gates, which is applied to the initial state $|0\rangle$, followed by a measurement of the first qubit. We would like to approximately determine the probability that this measurement outputs 1. As the set of gates {Hadamard, CCZ} is universal for quantum computation [38, 3], this is sufficient to solve any problem in BQP. So consider the

following circuit C':



Then $\langle 0|C'|0\rangle = \langle 0|C^\dagger Z_1 C|0\rangle = \operatorname{tr} Z_1 (C|0\rangle\langle 0|C^\dagger)$, which is precisely the difference between the probability that the measurement outputs 0, and the probability that it outputs 1. By the definition of the error bounds in BQP, we have $|\langle 0|C'|0\rangle| \geq 1/3$, so it is sufficient to estimate $\langle 0|C'|0\rangle$ up to absolute error less than $1/3$ to determine whether the answer should be 0 or 1. As discussed in Section 2, we can assume that C' begins and ends with Hadamards on every qubit (equivalently, that C begins with Hadamards on every qubit).

From Proposition 1, there is a degree-3 polynomial $f_{C'} : \{0,1\}^n \to \{0,1\}$, where $n = h + \ell$, h is the number of Hadamard gates in the internal part of C' and ℓ is the number of qubits on which C' acts, such that $\langle 0|C'|0\rangle = \operatorname{gap}(f_{C'})/2^{h/2+\ell}$. So it is sufficient to determine $\operatorname{gap}(f_{C'})$ up to absolute accuracy $\frac{1}{3} \cdot 2^{h/2+\ell} = \frac{1}{3} \cdot 2^{n/2+\ell/2}$ to solve the original decision problem. Observing that $\ell \geq w(f)$ by definition completes the proof. □

We have seen that approximately computing $\operatorname{gap}(f)$ up to accuracy $O(2^{(n+w(f))/2})$ is sufficient to simulate arbitrary quantum computations. This is already sufficient to imply the known complexity class inclusion[2] BQP$\subseteq$P$^{\#\mathrm{P}}$ [7, 13], as it is easy to see that $\operatorname{gap}(f)$ can be computed exactly by counting the number of inputs of a circuit which evaluate to 1, and hence is in P$^{\#\mathrm{P}}$. Does the implication go the other way? That is, can we use quantum computation to approximate $\operatorname{gap}(f)$ up to accuracy $O(2^{(n+w(f))/2})$? If so, this would imply that approximating $\operatorname{gap}(f)$ up to this level of accuracy is effectively equivalent[3] to the complexity class BQP. This would give a new example of a combinatorial problem which characterises the power of quantum computation. Several such examples are known (e.g. [30, 4, 23, 44]), but approximately computing the number of zeroes of degree-3 polynomials would arguably be the simplest yet.

For any quantum circuit C, the Hadamard test [4] can be used to estimate $\langle 0|C|0\rangle$ up to inverse-polynomially small absolute error. So, if we are given a circuit on ℓ qubits corresponding to a polynomial f, we can estimate $\operatorname{gap}(f)$ up to accuracy $O(2^{n/2+\ell/2})$. If $\ell = w(f)$, we have achieved an approximation which matches the bound of Proposition 8. However, it is not clear how to efficiently determine a quantum circuit corresponding to f which acts on $w(f)$ qubits. Indeed, even determining $w(f)$ itself could be NP-complete.

Problem 9. *What is the complexity of computing $w(f)$ for an arbitrary degree-3 polynomial $f : \{0,1\}^n \to \{0,1\}$?*

To achieve a good enough level of accuracy in estimating $\operatorname{gap}(f)$, it would be sufficient to find a circuit on ℓ qubits such that $\ell = w(f) + O(\log n)$. But it is non-obvious how to obtain even this level of accuracy.

We can also relate the quantum circuit width to the complexity of *classical* simulation.

Proposition 10. *Given a degree-3 polynomial $f : \{0,1\}^n \to \{0,1\}$ and a description of a quantum circuit on ℓ qubits corresponding to f, $\operatorname{gap}(f)$ can be calculated exactly classically in time*

[2] In fact, this argument also gives an alternative proof of the tighter complexity class inclusion BQP$\subseteq$AWPP, due to Fortnow and Rogers [18].

[3] Technically, equivalent to the complexity class PromiseBQP [23]: the class of problems which reduce to determining whether the acceptance probability of a quantum computation is greater than 2/3 or less than 1/3, given the promise that exactly one of these is the case.

$O(2^{2\ell}\,\mathrm{poly}(n))$. *Further,* $\mathrm{gap}(f)$ *can be approximated up to additive error* $\epsilon\, 2^n$ *with success probability* $2/3$ *in time* $O(\mathrm{poly}(n)/\epsilon^2)$.

Proof. For any quantum circuit C on ℓ qubits containing m gates, $\langle 0|C|0\rangle$ can be calculated in time $O(2^{2\ell}m)$ simply by multiplying out the matrices. If C represents f, it can be assumed to contain at most $\mathrm{poly}(n)$ gates, so $m = \mathrm{poly}(n)$. For the second part, we can estimate $|\{x : f(x) = 0\}|/2^n$ by taking the average of s random samples from $f(x)$. Each sample can be computed in time $\mathrm{poly}(n)$. By a standard Chernoff bound argument [15], in order for this estimate to be correct up to absolute error ϵ with probability $2/3$, it is sufficient to take $s = O(1/\epsilon^2)$. □

Using the second approach in Proposition 10, we can achieve the same level of approximation accuracy achieved by an optimal quantum circuit by taking $\epsilon = O(2^{(w(f)-n)/2})$, giving a classical algorithm which runs in time $O(2^{n-w(f)}\,\mathrm{poly}(n))$. Thus observe that, if either $w(f) \geq n - O(\log n)$ or $w(f) \leq O(\log n)$, the speedup we could obtain by using a quantum algorithm to compute $\mathrm{gap}(f)$ cannot be super-polynomial (but apparently for different reasons). In the former case, the approximate classical algorithm from Proposition 10 runs in polynomial time; in the latter case, the exact classical algorithm runs in polynomial time.

These results motivate us to further explore the concept of quantum circuit width.

3.3 Quantum circuit width

We first show that most degree-3 polynomials f have high quantum circuit width, and hence that $\mathrm{gap}(f)$ cannot be approximated significantly more efficiently using this quantum circuit approach than is possible classically.

Proposition 11. *The probability that a random degree-3 polynomial* $f : \{0,1\}^n \to \{0,1\}$ *with no constant term has* $w(f) \leq n - 3$ *is at most* $2^{(-3n+1)/2}$.

Proof. We count the number of different functions which can correspond to a circuit on k qubits of the form discussed in this work whose internal part contains $n-k$ Hadamards (giving a polynomial on n variables). Break the internal part of the circuit into $n-k+1$ horizontal blocks such that each Hadamard $H_1, \ldots, H_h$ begins a block. Then slide (commute) all the Z, CZ, CCZ gates in the circuit to the left until they cannot go any further (i.e. come up against a Hadamard). Then, except for the furthest left-hand block, each such gate acts on the qubit corresponding to the Hadamard which begins its block. Therefore, there are at most $2^{\binom{k}{2}+k+1} = 2^{k(k+1)/2+1}$ different possibilities for the combination of gates in each block, except the left-hand block, where there are $2^{\binom{k}{3}+\binom{k}{2}+k} = 2^{k(k^2+5)/6}$ possibilities. There are k^{n-k} possibilities for the vertical position of the Hadamards. Overall, we get an upper bound on the number of functions that can be produced which is equal to

$$2^{(n-k)(k(k+1)/2+1)+k(k^2+5)/6+(n-k)\log_2 k}.$$

Take the rough upper bound $\log_2 k \leq k/2$, valid for large enough k. Then the above quantity is increasing with k and for $k = n-3$ is equal to $2^{(n^3-4n+3)/6}$. On the other hand, there are $2^{\binom{n}{3}+\binom{n}{2}+n} = 2^{(n^3+5n)/6}$ degree-3 polynomials on n variables with no constant term. Thus the fraction of polynomials f such that $w(f) \leq n-3$ is at most exponentially small in n. □

We next relate the quantum circuit width of a polynomial to a combinatorial parameter of a hypergraph associated with the polynomial. A hypergraph $G = (V, E)$ is defined by a set of

vertices V and a set of hyperedges E, where a hyperedge is a subset of at least 2 of the vertices. We can associate a degree-3 polynomial f with a hypergraph $G(f)$ by associating each variable with a vertex, and thinking of each term involving at most 3 variables as a hyperedge between at most 3 vertices. A proper k-colouring of a hypergraph G is an assignment of colours to vertices, picked from a set of colours of size k, such that at least two vertices within each hyperedge are assigned different colours. The chromatic number of a hypergraph, $\chi(G)$, is defined to be the minimal k such that there exists a proper k-colouring of G.

Proposition 12. *For any degree-3 polynomial $f : \{0,1\}^n \to \{0,1\}$, $\chi(G(f)) \leq 2w(f)$, and this inequality can be tight. However, there exists a family of polynomials $f : \{0,1\}^n \to \{0,1\}$, for n even, such that $\chi(G(f)) = 2$ but $w(f) = n/2$.*

Proof. Given a circuit for f using ℓ qubits, each pair of variables which are associated with the same qubit but are not adjacent cannot be included in the same term of f. We can thus properly colour the vertices of $G(f)$ using at most 2ℓ colours by associating a pair of colours (c_i, d_i) with each qubit, and allocating colour c_i (resp. d_i) to those vertices which occur on line i at odd (resp. even) times. Tightness follows from the function $f(x) = x_1x_2 + x_2x_3 + \cdots + x_{n-1}x_n$, which has $w(f) = 1$. As $G(f)$ is a path on n vertices, $\chi(G(f)) = 2$.

For the second part, consider the polynomial $f(x) = x_1x_2 + x_3x_4 + \cdots + x_{n-1}x_n$. The corresponding graph $G(f)$ consists of $n/2$ disjoint edges and hence can be properly coloured with 2 colours. □

4 Polynomials and simulation of quantum circuits

We have seen that, using the construction of Proposition 8, in order to simulate a quantum circuit – i.e. to determine the probability that, at the end of the circuit, the result of measuring the first qubit would be 1 – it is sufficient to compute $\text{gap}(f)$ for a related function f. One can use this idea to easily obtain various simulation results for classes of quantum circuits.

First, as discussed in Section 3.2, any circuit containing only Hadamard, Z and CZ gates can be simulated efficiently classically using the Gottesman-Knill theorem [34]. This result can be generalised to circuits containing a small number of CCZ gates as follows.

Proposition 13. *Let S be a hitting set for the collection of degree-3 terms of $f : \{0,1\}^n \to \{0,1\}$ (in other words, a set of variables such that each degree-3 term contains at least one element of S). Then, given f and S, $\text{gap}(f)$ can be computed in time $O(2^{|S|}\,\text{poly}(n))$.*

Proof. For any variable x_i, let $f_{x_i \leftarrow z}$ denote the function obtained from f by fixing the value of x_i to z. Then it is easy to see that $\text{gap}(f) = \text{gap}(f_{x_i \leftarrow 0}) + \text{gap}(f_{x_i \leftarrow 1})$. Applying this recursively, for any set S of variables, $\text{gap}(f)$ can be computed by summing the gaps of the $2^{|S|}$ functions obtained by fixing each of the variables in S to either 0 or 1. If we choose S to include at least one variable from each of the degree-3 terms in f, each new polynomial produced has degree at most 2, and hence has gap computable in time $O(n^3)$ [16, 2]. □

Observe that, if f contains k degree-3 terms, there is always a hitting set containing k elements (just by taking one variable from each term). More generally, we would like to find a hitting set of minimal size $h(f)$. This is an NP-complete problem [21], but luckily an approximation

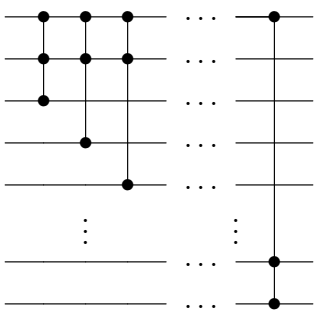


Figure 4: A circuit with a hitting set of size 1 but many non-Clifford gates.

$h'(f) \leq 3h(f)$ can be found in polynomial time (approximating $h(f)$ any better than this is NP-hard [28], assuming the Unique Games Conjecture from complexity theory). We therefore have that $\mathrm{gap}(f)$ can be computed in time $2^{O(h(f))}\,\mathrm{poly}(n)$.

The construction of Proposition 8 produces a circuit C' from any circuit C on ℓ qubits whose corresponding polynomial $f_{C'}$ satisfies $h(f_{C'}) \leq 2h(f_C)$. Therefore, any polynomial-size circuit C can be simulated in time $2^{O(h(f_C))}\,\mathrm{poly}(\ell)$. It was already shown by Aaronson and Gottesman that circuits on ℓ qubits containing k non-Clifford gates can be simulated in time $2^{O(k)}\,\mathrm{poly}(\ell)$ [2], and more recent work has improved the constant hidden in the $O(k)$ term for circuits where the only non-Clifford gate is the T gate [9, 8]. However, the result here is somewhat more general in that there exist circuits with many CCZ gates whose corresponding polynomial has a small hitting set. For example, Figure 4 illustrates a circuit on ℓ qubits containing CCZ gates from the first qubit to every other pair of qubits; this circuit has $\binom{\ell-1}{2}$ gates but a hitting set of size 1.

Also observe that this simulation does not seem to follow immediately from the results of Markov and Shi [33] on simulating quantum circuits by tensor contraction in time exponential in the tree-width of the circuit. Indeed, there exist circuits that contain only Clifford gates but have arbitrarily high tree-width.

4.1 Simulation by linear transformations

In order to calculate $\mathrm{gap}(f)$ more efficiently, we can attempt to transform f into a polynomial which is simpler in some sense. One way of doing this is to apply a linear transformation to f. The following result is well-known in the theory of error-correcting codes [32]; we include the simple proof for completeness.

Proposition 14. *For any degree-3 polynomial $f : \{0,1\}^n \to \{0,1\}$, and any nonsingular linear transformation $L \in GL_n(\mathbb{F}_2)$, let f^L be the polynomial $f^L(x) = f(Lx)$. Then $\deg(f^L) = 3$ and $\mathrm{gap}(f^L) = \mathrm{gap}(f)$.*

Proof. To produce f^L from f, we can replace each term $x_i x_j x_k$ with a term $(Lx)_i (Lx)_j (Lx)_k$ (and similarly for the terms dependent on 1 or 2 variables). As $(Lx)_i$ is a linear function of x over $\mathbb{F}_2$, and similarly for j, k, the product of these functions is a polynomial of degree at most 3. For the second part, as L is nonsingular, there is a one-to-one mapping between the set $\{x : f(x) = 0\}$ and the set $\{x : f^L(x) = 0\}$, so $\mathrm{gap}(f^L) = \mathrm{gap}(f)$. □

In fact, the group $GL_n(\mathbb{F}_2)$ is known to be the *largest* group of transformations which preserves polynomial degree [32]. In some cases, a linear transformation can completely change a function's

quantum circuit width and hence the efficiency with which its gap can be computed using the exact algorithm of Proposition 10. As a very simple example, it is easy to show that the polynomial $x_1 + \cdots + x_n$ has quantum circuit width n, but following a linear transformation that maps $x_1 + \cdots + x_n \mapsto x_1$, the resulting polynomial x_1 has quantum circuit width 1.

Although we do not know a general way of minimising the quantum circuit width of a function by applying a linear transformation, a simpler approach is to minimise the number of variables on which the function depends. Given a polynomial f which depends on v variables, $\mathrm{gap}(f)$ can be computed exactly in time $O(2^v \,\mathrm{poly}(v))$ simply by evaluating f on each of the 2^v possible assignments to the variables. It has been shown by Carlini [12] (see also Appendix B of [27]) that the linear transformation L which minimises the number of variables in f^L can be computed in polynomial time. We therefore obtain the following corollary:

Corollary 15. *Let C be a polynomial-size quantum circuit on ℓ qubits such that there exists a linear transformation L such that f_C^L depends on v variables. Then there is a classical algorithm which computes $\langle 0|C|0\rangle$ exactly in time $O(2^v \,\mathrm{poly}(\ell))$.*

In particular, if there exists L such that f_C^L depends on $O(\log \ell)$ variables, we obtain a polynomial-time classical simulation of C.

5 Conclusions

In this work we have investigated a correspondence between quantum circuits and low-degree polynomials over finite fields, and have shown that by exploiting this correspondence we can obtain classical hardness results, as well as ideas for classical algorithms that simulate quantum circuits. There seem to be many interesting directions in which to further explore this area. For example, as discussed in Section 3.2, what is the complexity of computing or approximating the quantum circuit width $w(f)$? Is it related to other measures of complexity of boolean functions? Low-degree polynomials over $\mathbb{F}_2$ are equivalent to Reed-Muller codes [32] – can ideas from classical coding theory be applied to understand quantum circuits? And finally, can any other useful simulation techniques be developed by taking this perspective – perhaps for other specific classes of quantum circuits?

Acknowledgements

This work was supported by an EPSRC Early Career Fellowship (EP/L021005/1). Some of this work was carried out while the author was at the University of Cambridge. I would like to thank Mick Bremner and Dan Shepherd for discussions on this topic over the last few years, and Scott Aaronson, Miriam Backens and Richard Jozsa for helpful comments on a previous version. Special thanks to Sophie for arriving safely, and providing many helpful distractions from completing this work.

References

[1] S. Aaronson. A linear-optical proof that the permanent is #P-hard. *Proc. Roy. Soc. Ser. A*, 467(2136):3393–3405, 2011. `arXiv:1109.1674`.

[2] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70:052328, 2004. quant-ph/0406196.

[3] D. Aharonov. A simple proof that Toffoli and Hadamard are quantum universal, 2003. quant-ph/0301040.

[4] D. Aharonov, V. Jones, and Z. Landau. A polynomial quantum algorithm for approximating the Jones polynomial. In *Proc. 38th Annual ACM Symp. Theory of Computing*, pages 427–436, 2006. quant-ph/0511096.

[5] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[6] D. Bacon, W. van Dam, and A. Russell. Analyzing algebraic quantum circuits using exponential sums, 2008. https://www.cs.ucsb.edu/~vandam/LeastAction.pdf.

[7] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997.

[8] S. Bravyi and D. Gosset. Improved classical simulation of quantum circuits dominated by Clifford gates. *Phys. Rev. Lett.*, 116:250501, 2016. arXiv:1601.07601.

[9] S. Bravyi, G. Smith, and J. Smolin. Trading classical and quantum computational resources. *Phys. Rev. X*, 6:021043, 2016. arXiv:1506.01396.

[10] M. Bremner, R. Jozsa, and D. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proc. Roy. Soc. Ser. A*, 467(2126):459–472, 2011. arXiv:1005.1407.

[11] M. Bremner, A. Montanaro, and D. Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations, 2015. arXiv:1504.07999.

[12] E. Carlini. Reducing the number of variables of a polynomial. In *Algebraic Geometry and Geometric Modeling*, Mathematics and Visualization, pages 237–247. Springer, 2006.

[13] C. Dawson, H. Haselgrove, A. Hines, D. Mortimer, M. Nielsen, and T. Osborne. Quantum computing and polynomial equations over the finite field Z_2. *Quantum Inf. Comput.*, 5(2):102–112, 2005. quant-ph/0408129.

[14] A. Drucker and R. de Wolf. Quantum proofs for classical theorems. *Theory of Computing Graduate Surveys*, 2:1–54, 2011. arXiv:0910.3376.

[15] D. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

[16] A. Ehrenfeucht and M. Karpinski. The computational complexity of (XOR, AND)-counting problems, 1990. Technical Report 8543-CS.

[17] M. Fisher. Statistical mechanics of dimers on a plane lattice. *Phys. Rev.*, 124:1664–1672, 1961.

[18] L. Fortnow and J. Rogers. Complexity limitations on quantum computation. In *Proc. 13th Annual IEEE Conf. Computational Complexity*, pages 202–209, 1998. cs/9811023.

[19] K. Fujii and T. Morimae. Quantum commuting circuits and complexity of Ising partition functions, 2013. arXiv:1311.2128.

[20] H. García, I. Markov, and A. Cross. On the geometry of stabilizer states. *Quantum Inf. Comput.*, 14(7&8):683–720, 2014.

[21] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-Completeness*. W. H. Freeman, 1979.

[22] F. Jaeger, D. Vertigan, and D. Welsh. On the computational complexity of the jones and tutte polynomials. *Math. Proc. Camb. Phil. Soc.*, 108:35–53, 1990.

[23] D. Janzing and P. Wocjan. A simple PromiseBQP-complete matrix problem. *Theory of Computing*, 3:61–79, 2007. quant-ph/0606229.

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

[25] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004.

[26] P. Kasteleyn. Dimer statistics and phase transitions. *J. Math. Phys.*, 4(2):287–293, 1963.

[27] N. Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Proc. 22nd ACM-SIAM Symp. Discrete Algorithms*, pages 1409–1421, 2011.

[28] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2-\epsilon$. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

[29] E. Knill and R. Laflamme. Power of one bit of quantum information. *Phys. Rev. Lett.*, 81:5672–5675, 1998. quant-ph/9802037.

[30] E. Knill and R. Laflamme. Quantum computation and quadratically signed weight enumerators. *Inf. Proc. Lett.*, 79(4), 2001. quant-ph/9909094.

[31] G. Kuperberg. How hard is it to approximate the Jones polynomial? *Theory of Computing*, 11(6):183–219, 2015. arXiv:0908.0512.

[32] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, 1983.

[33] I. Markov and Y. Shi. Simulating quantum computation by contracting tensor networks. *SIAM J. Comput.*, 38:963–981, 2008. quant-ph/0511069.

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

[35] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[36] T. Rudolph. A simple encoding of a quantum circuit amplitude as a matrix permanent. *Phys. Rev. A*, 80:054302, 2009. arXiv:0909.3005.

[37] D. Shepherd and M. J. Bremner. Temporally unstructured quantum computation. *Proc. Roy. Soc. Ser. A*, 465(2105):1413–1439, 2009. arXiv:0809.0847.

[38] Y. Shi. Both Toffoli and controlled-NOT need little help to do universal quantum computing. *Quantum Inf. Comput.*, 3(1):84–92, 2003. `quant-ph/0205115`.

[39] H. Temperley and M. Fisher. Dimer problem in statistical mechanics – an exact result. *Phil. Mag.*, 6(68):1061–1063, 1961.

[40] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[41] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[42] L. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008.

[43] M. Van den Nest. Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond. *Quantum Inf. Comput.*, 10(3–4):0258–0271, 2010. `arXiv:0811.0898`.

[44] 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`.

[45] J. Watrous. Quantum computational complexity. In *Encyclopedia of Complexity and Systems Science*, pages 7174–7201. Springer New York, 2009. `arXiv:0804.3401`.