Quantum Information and Computation, Vol. 11, No. 9&10 (2011) 0784–0812

SIMULATING QUANTUM COMPUTERS WITH PROBABILISTIC METHODS

MAARTEN VAN DEN NEST
Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Straße 1
85748 Garching, Germany

Received April 21, 2011
Revised July 5, 2011

We investigate the boundary between classical and quantum computational power. This work consists of two parts. First we develop new classical simulation algorithms that are centered on sampling methods. Using these techniques we generate new classes of classically simulatable quantum circuits where standard techniques relying on the exact computation of measurement probabilities fail to provide efficient simulations. For example, we show how various concatenations of matchgate, Toffoli, Clifford, bounded-depth, Fourier transform and other circuits are classically simulatable. We also prove that sparse quantum circuits as well as circuits composed of CNOT and $\exp[i\theta X]$ gates can be simulated classically. In a second part, we apply our results to the simulation of quantum algorithms. It is shown that a recent quantum algorithm, concerned with the estimation of Potts model partition functions, can be simulated efficiently classically. Finally, we show that the exponential speed-ups of Simon's and Shor's algorithms crucially depend on the very last stage in these algorithms, dealing with the classical postprocessing of the measurement outcomes. Specifically, we prove that both algorithms would be classically simulatable if the function classically computed in this step had a sufficiently peaked Fourier spectrum.

Keywords: Quantum computation, classical simulations, sampling methods, quantum algorithms
Communicated by: R Jozsa & B Terhal

1 Introduction

What is the power of quantum computers compared to classical ones? Understanding this fundamental but difficult question is one of the great challenges in the field of quantum computation.

A fruitful approach to tackle this problem is to study classes of quantum computations that do *not* offer any computational benefits over classical computation. Indeed, such investigations shed light on the essential features of quantum mechanics that are responsible for quantum computational power. At the same time, understanding which classes of quantum computations can be simulated classically provides useful insights in the difficult task of constructing novel quantum algorithms, potentially yielding indications on where to look for new algorithmic primitives.

In recent years several non-trivial classes of quantum computations have been identified for which an efficient classical simulation can be achieved. For example, certain computations are classically simulatable due to their low degrees of entanglement (quantified appropriately in terms of suitable entanglement measures) [1, 2, 3, 4, 5]. Other well known results are the

Gottesman-Knill theorem [6, 7, 8, 9, 10] and the classical simulation of matchgate circuits [11, 12, 13, 14, 15]. The latter two classes of results provide key illustrations of the fascinating and puzzling relation between classical and quantum computational power, as they e.g. regard computations that may exhibit large degrees of entanglement, interference, superposition, etc.—i.e. the ingredients that supposedly provide QC with its increased power—but which nevertheless cannot achieve any computational speed-up over classical computers.

A common element in many existing classical simulation results and methods is the notion of classical simulation that is, sometimes implicitly, adopted in these works. When a quantum computation is to be simulated classically, the goal may be to either classically compute measurement probabilities (or expectation values) with high precision in polynomial time ("strong simulation"), *or* to classically sample in polynomial time from the resulting output probability distribution ("weak simulation"). Given the intrinsic probabilistic nature of quantum mechanics, it is readily motivated that weak simulation is the more natural notion of what a classical simulation should constitute. Furthermore, one may easily construct examples of quantum circuit classes for which strong simulation is intractable whereas weak simulation is achieved by elementary sampling methods (see e.g. [10])—hence showing that a gap between strong and weak simulations manifests itself already in elementary scenarios. The latter gap moreover highlights that any serious attempt to compare classical with quantum computational power should not be based on strong simulation methods.

In spite of these basic and well-known insights, the majority of existing results on classical simulation of QC regard the strong variant, and weak simulation techniques seem to date largely unexplored. The goal of the present work is to develop new classical simulation algorithms that are based on sampling methods and to therewith initiate an investigation of the potential of weak simulation of quantum computation. Next we state more precisely the contributions of this work.

2 Statement of results

Classical simulation of QC with probabilistic methods

In a first part of the paper, we develop tools to investigate weak classical simulation of QC (henceforth the notion "classical simulation" will always refer to efficient weak classical simulation). A central ingredient in our analysis will be a certain class of quantum states, called here *computationally tractable states* (CT states). Colloquially speaking, a state is CT if it is possible to classically simulate computational basis measurements on $|\psi\rangle$ *and* if the coefficients of $|\psi\rangle$ in this basis can be efficiently computed. As we will see, many important state families—matrix product states, stabilizer states, states generated by polynomial size matchgate circuits, and several others—turn out to be CT. A second element will be the notion of efficiently computable sparse operators (ECS). An n-qubit operation is ECS if its matrix representation in the standard basis has at most $\mathrm{poly}(n)$ nonzero entries per row and per column, and if these entries can be determined efficiently. For example, all Pauli products, k-local operators with $k = O(\log n)$, as well as operators that can be written as polynomial size circuits of Toffoli gates, are ECS. We will prove the following result.

Theorem 1 *Consider a polynomial size quantum circuit U acting on a state $|\psi\rangle$ and followed by measurement of an observable O where $\|O\| \leq 1$. If $|\psi\rangle$ is CT and if $U^\dagger OU$ is ECS, then*

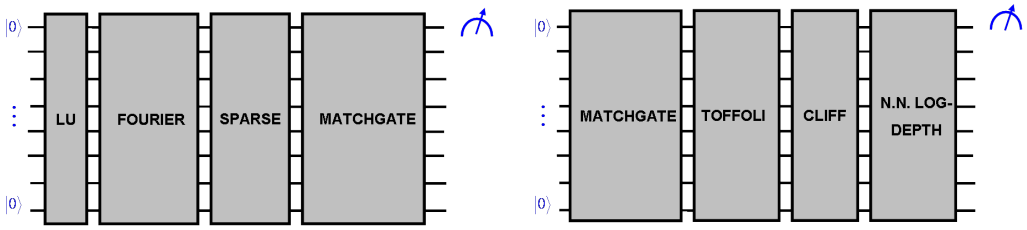

Fig. 1. The above concatenated quantum circuits can be efficiently simulated classically via an application of theorem 1. See section 5.2 for a discussion of these examples.

this quantum computation can be simulated classically.

An immediate remark to be made is that the unitary operation U itself is *not* required to be sparse—only its action on O is to yield an ECS operation, which is a significantly distinct requirement. For example, if U is a polynomial size circuit consisting of nearest neighbor matchgates—which is generally not sparse at all—then $U^\dagger Z_1 U$ is a linear combination of poly(n) Pauli products, which is an ECS operation (here Z denotes the Pauli σ_z operator acting on qubit 1).

Theorem 1 identifies a general scenario in which quantum circuits can be simulated efficiently classically. This result turns out to be rather versatile and will be useful in a number of contexts. In this work we highlight the following particular applications (however, it is likely that this result has applications beyond the ones considered here):

- **Sparse circuits.** A simple instance of theorem 1 is obtained by considering a product input state (which is trivially CT) and the Z observable on, say, the first qubit, and by letting the circuit U itself be an ECS operation (in which case $U^\dagger Z U$ is ECS as well). Then, by virtue of theorem 1, the resulting quantum computation can be simulated classically. In fact, one can immediately extend this result by composing m efficiently computable s-sparse[a]unitary operations with $s^m = \text{poly}(n)$. Then the overall circuit will still be ECS, as can easily be verified, and thus can be simulated classically due to theorem 1.

  Sparse unitary operations are of interest because they highlight the role of *interference* in quantum computation, as opposed to *entanglement*. In particular, sparse operations may produce highly entangled states but the interference exhibited in any sparse unitary evolution is always limited. As we will show, this absence of a high degree of interference can be exploited to construct an efficient classical simulation algorithm, in spite of the potentially complex entangled states produced throughout the computation. This provides (yet another) illustration that the presence of entanglement is by no means sufficient to guarantee quantum computational speed-ups. Sparse operations furthermore provide examples of a class of QCs where weak classical simulation is efficiently possible, whereas strong simulation is intractable (#P-hard). In other words, adopting the notion of weak simulation constitutes a necessary ingredient in the simulation

[a]An operator is s-sparse if its standard basis matrix representation has at most s nonzero entries per row and per column.

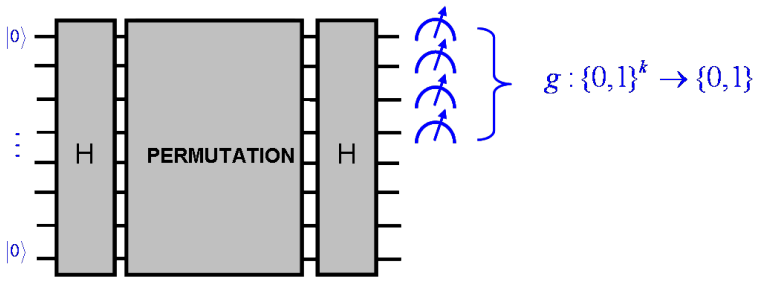

Fig. 2. Both the factoring algorithm and Simon's algorithm can be implemented by a circuit with the following structure. The first and third round in the circuit consist of collections of Hadamard operations applied to certain subsets of the qubits; the second round is a unitary operation that acts as a permutation on the computational basis. The circuit is followed by a $\{|0\rangle, |1\rangle\}$ measurement of a subset of the qubits. The algorithm concludes with classical postprocessing of the measurement results.

of sparse circuits whereas strong simulation methods such as e.g. tensor contraction schemes cannot (unless #P = P) yield an efficient classical simulation.

- **Composability.** Instead of letting $|\psi\rangle$ be a simple product input state, we may also consider more complicated CT states which are e.g. the result of an earlier quantum computation, i.e. $|\psi\rangle = U'|\psi_{\text{in}}\rangle$ for some simple (e.g. standard basis) input $|\psi_{\text{in}}\rangle$. As long as $|\psi\rangle$ is CT *and* subsequently a circuit U is applied followed by measurement of O such that $U^\dagger OU$ is ECS, the overall quantum circuit UU', acting on $|\psi_{\text{in}}\rangle$ and followed by measurement of O, can be simulated classically by theorem 1. One hence arrives at a criterion to asses when the *concatenation* of two quantum circuits can be simulated classically.

  Since the majority of existing efficiently simulatable circuits turn out to generate CT states when acting on suitable inputs *and* since at the same time many simulatable operations yield ECS operations when acting on suitable observables, the above composability result is applicable to a wide variety of settings. In particular, this result applies to Clifford operations, matchgate circuits, bounded-depth circuits, classical circuits, bounded-treewidth circuits, the quantum Fourier transform, and others. This leads to sometimes surprising examples of concatenated circuits that can be simulated classically (cf. Fig. 1). As illustrated in these examples, the concatenation of simulatable blocks of very different nature may remain efficiently simulatable classically (consider e.g. the concatenation of a Clifford with a matchgate circuit).

  It is interesting to compare the examples in Fig. 1 to powerful quantum algorithms such as Simon's and Shor's. Strikingly, the latter algorithms are implemented with particularly simple circuitry—arguably even simpler than the classically simulatable circuits displayed in Fig. 1. In particular, it is known that both the factoring algorithm and Simon's algorithm can be efficiently implemented by a circuit with the very simple structure of Fig. 2 [17, 18]. Intriguingly, this circuit is the composition of only three blocks, each of which is elementary. Nevertheless, our simulation techniques *cannot* be successfully be applied to yield an efficient classical simulation of this circuit class. In

the second part of this work we investigate the hardness of simulating these circuits and, by extension, Simon's and Shor's algorithms, in more detail.

- **CNOT-$e^{i\theta X}$ circuits.** As a further application of theorem 1, we will show that polynomial size circuits composed of CNOT and $e^{i\theta X}$ gates, acting on product inputs and followed by measurement of Z on any single qubit, can be simulated classically. This result is of interest since it is known that CNOT together with any single *real* one-qubit gate V such that V^2 is not basis-preserving, is *universal* for quantum computation [19]. In contrast to this, here it is found that there is a class of non-trivial *complex* gates $e^{i\theta X}$ that can be added to the CNOT gate while retaining efficient classical simulation.

  The above result is also interesting from a conceptual point of view. In particular, its proof will follow from a variant of theorem 1 where states $|\psi\rangle$ and operations $U^\dagger OU$ are considered that are CT, resp. ECS, with respect to bases other than the standard basis. Letting $|\psi\rangle$ be a product input and U a polynomial size circuit composed of CNOT and $e^{i\theta X}$ gates, it will be shown that $|\psi\rangle$ and $U^\dagger Z_1 U$ are CT, resp. ECS, *with respect tho the* $\{|\pm\rangle\}$ *basis of* X *eigenstates*. Hence, viewing the entire computation in this basis and applying theorem 1 shows that classical simulation is efficiently possible. In contrast, a direct application of theorem 1, i.e. with respect to the standard basis, is not possible as $U^\dagger Z_1 U$ is generally not ECS w.r.t this basis.

Classical simulation of quantum algorithms

In a second part of the paper, the above results are applied in the context of quantum algorithms. Depending on the case at hand, the goal will be to either show that certain algorithms can be simulated classically or to deepen our insight into why certain algorithms achieve exponential (oracle) speed-ups over classical computation. We will analyze three different quantum algorithms:

- a quantum algorithm to estimate partition functions of classical lattice models [20];
- a general class of quantum algorithms containing the Deutsch-Jozsa algorithm [21];
- Simon's algorithm [18].

The first two classes of quantum algorithms will be proved to be classically simulatable using the methods developed in this paper. We refer to the relevant sections in the text for a discussion. For the time being, we limit ourselves to discussing our results in the context of Simon's algorithm, which we consider the most interesting application.

Recall that in Simon's problem one has oracle access to a function $f : \{0,1\}^n \to \{0,1\}^n$; it is promised that there exists an unknown n-bit string a such that $f(x) = f(y)$ if and only if $y = x + a$ (addition modulo 2). The goal is to find a. Classically one needs at least $O(2^{\frac{n}{2}})$ queries, whereas a quantum computer can solve the problem with $O(n)$ queries—i.e. Simon's algorithm achieves an exponential oracle separation between BQP and BPP. In spite of its computational power, Simon's algorithm is implemented with very simple circuitry, as displayed in Fig. 2. What are the essential ingredients responsible for the power of this algorithm?

In standard considerations, the interplay between the Fourier transform (i.e. the second layer of Hadamards in Fig. 2) and the oracle f is emphasized. After the oracle is applied,

the system is in the state $\sum |x\rangle|f(x)\rangle$. The Fourier transform then creates interference in the system and "picks out" the relevant computational basis states, such that a subsequent measurement of the system yields the desired information about the unknown bit-string a. This rather delicate relation between oracle and Fourier transform is usually considered to be among the main origins of the hardness of classically simulating Simon's algorithm. In this work we will show that this point of view is not the end of the story: in particular, we will find that the interplay between the same Fourier transform and the function g computed during the round of *classical postprocessing* is an equally important element in the speed-up achieved by the algorithm. Specifically, we will prove the following result.

Theorem 2 (rough version) *Consider a quantum circuit displaying the structure as depicted in Fig. 2. If the function g computed in the round of classical postprocessing is promised to have a sufficiently "peaked" Fourier spectrum, then the entire circuit can be simulated efficiently classically, independent of the specific forms of the other rounds.*

In particular, if the final classical round in Simon's algorithm happened to regard a function with sufficiently peaked fourier spectrum, then the entire quantum computation could be simulated efficiently (i.e. in poly(n) time using poly(n) queries to the oracle f)—independent of the details of e.g. the oracle computed in an earlier stage of the computation, and independent of e.g. the entanglement produced by the quantum circuit. This result hence exposes the double role played by the Fourier transform, which is to act appropriately on *both* the oracle f *and* the function computed in the postprocessing, in order to achieve a quantum speed-up. These observations highlight that the power of a quantum algorithm can only be understood by taking the entire computation into account including the classical postprocessing round, even though the latter may at first sight look rather innocuous. Indeed, note that—strikingly—in Simon's algorithm this round 'only' involves solving a simple system of *linear* equations over $\mathbf{Z}_2$! Nevertheless, this simple classical computation is associated with a function having a very *flat* spectrum (as we will see), hence ensuring the exponential speed up achieved by Simon's algorithm.

Remark: in the formulation of theorem 2, no knowledge of the Fourier spectrum of the function in question is assumed, except the promise that this spectrum is "peaked". Using remarkable results of Boolean learning theory, enough information of the spectrum can be efficiently reconstructed in order to achieve the polynomial time classical simulation as stated in the theorem. ◇

Finally, the factoring algorithm can also be implemented with a circuit displaying the structure of Fig. 2. Therefore, the classical postprocessing also plays a similar crucial role in this algorithm. As the technical considerations in Simon's algorithm are more transparent than in Shor's, here we will focus on the former—keeping in mind that our conclusions also apply to the latter.

Matchgate circuits and polynomial time classical computation

Somewhat unrelated to the above context, we prove a "byproduct result" that we find noteworthy. We will arrive at a complexity-theoretic result regarding the computational power of matchgate circuits. Roughly speaking, we will show the following (see theorem 4 for a precise

statement):

> *The class of functions that can be efficiently computed by nearest-neighbor matchgate circuits is strictly contained within P.*

Perhaps the most interesting aspect regarding this result here is its proof method. Surprisingly, the result will be obtained by combining the classical query lower bound of Simon's problem with our theorem 1. In particular, we will show that if the class of matchgate-computable functions comprised all of P, then a quantum algorithm for Simon's problem would exist which turns out to be efficiently simulatable classically (using theorem 1). Hence an efficient *classical* algorithm would exist which solves Simon's problem with poly(n) classical oracle queries, yielding a contradiction. Remark that it is striking how utterly unrelated matchgate circuits and Simon's problem seem at first sight!

Some conventions

In this paper, when we refer to a quantum circuit, we will always implicitly mean a uniformly generated family of quantum circuits. Further, by observable we mean any Hermitian operator O. When a measurement of an observable is considered at the end of a quantum circuit, we will always implicitly assume that this regards an observable that can be measured efficiently. The notion of 'simulation' will be synonymous to 'classical simulation'. The notion 'efficient' will be synonymous to 'in polynomial time'. For clarity, all results are stated in terms of qubit systems, but generalizations to arbitrary finite-dimensional quantum systems are immediate. Our standard notation for the computational basis of an n-qubit system will be $\{|x\rangle\}$, where $x = (x_1, \ldots, x_n)$ ranges over all n-bit strings and $|x\rangle = |x_1\rangle \otimes \ldots \otimes |x_n\rangle$. Finally, the spectral norm of an operator A is denoted by $\|A\|$.

3 Classical simulation of quantum computation

In this section we discuss the definition of classical simulation that will be adopted in the present work. Suppose that an n-qubit polynomial size quantum circuit produces an output state $|\psi_{\text{out}}\rangle$ and is followed by a measurement of an observable O, assuming that O can efficiently be measured. Then, repeating the computation $K = \text{poly}(n)$ times, recording the measurement outcome o_i in each run (i.e. each o_i is one of the eigenvalues of O) one obtains an estimate $\sigma = K^{-1} \sum_{i=1}^{K} o_i$ of the expectation value $\langle O \rangle = \langle \psi_{\text{out}} | O | \psi_{\text{out}} \rangle$. The accuracy of this approximation is dictated by the Chernoff-Hoeffding bound (we refer to the Appendix for a statement and discussion of this bound). In particular, this bound implies the following: for every $\epsilon = p(n)^{-1}$, where $p(n)$ represents an arbitrary polynomial in n, there exists a K that scales as a suitable polynomial in n such that the inequality $|\sigma - \langle O \rangle| \leq \epsilon$ holds with a probability that is exponentially (in n) close to 1. In other words, by taking poly(n) runs of the computation—and this is all that is allowed in an efficient quantum computation—it is possible to estimate $\langle O \rangle$ with an error that scales as an arbitrary inverse polynomial in n. We denote this type of estimate as an approximation with 'polynomial accuracy' or a 'polynomial approximation'. Note that a polynomial approximation achieves an estimate of $\langle O \rangle$ up to $O(\log n)$ significant bits.

The above method hence represents an efficient quantum algorithm to estimate $\langle O \rangle$ with polynomial accuracy with a success probability that lies exponentially close to 1. We now say

that this quantum algorithm can be efficiently simulated classically if there exists an efficient classical algorithm to provide a polynomial approximation of $\langle O\rangle$, again with a probability that lies exponentially close to 1. That is, we require the classical simulation algorithm to approximate $\langle O\rangle$ in polynomial time *with the same accuracy that is achieved by the quantum algorithm*. This notion of simulation is sometimes called *weak simulation*. The latter is to be regarded as opposed to the much more stringent requirement of *strong* simulation, where it is asked to construct a classical algorithm to approximate $\langle O\rangle$ in poly(m, n) time up to m significant bits (i.e. with exponential precision).

Note that the notion of weak simulation is more true to the concept of what a classical simulation actually constitutes since, colloquially speaking, it requires the classical simulation to achieve 'the same result' as the quantum algorithm. In contrast, in the strong scenario one is asked to construct an efficient classical algorithm that approximates $\langle O\rangle$ far more accurate than the quantum algorithm itself could generally achieve in polynomial time. Even though it has been realized previously that the weak variant is a valid and natural notion of classical simulation of QC (see e.g. [1, 14]), it seems that this notion is to date largely unexplored. In particular, the vast majority of classical simulation results use the strong variant. In [10] it was pointed out that there exists simple examples of quantum circuits for which weak classical simulation is possible with elementary methods, whereas strong simulation of the same circuits is a #P-hard problem and hence intractable. This highlights the presence of a significant gap between strong and weak simulation.

Remark: When the notion of polynomial approximation is used in the following, we will always mean a polynomial approximation which is achieved with a probability that is exponentially close to one. ◇

4 Computationally tractable states

The objective of this section is to develop the notion of computationally tractable (CT) states and to prove theorem 1. To do this, first we first define CT states and discuss some of their elementary properties; this is done in section 4.1. In section 4.2 we consider basis-preserving operations, which are identified as a class of operations that map CT states to CT states. In section 4.3 we consider sparse operations; the main technical contribution in this section is theorem 3 regarding the efficient classical estimation of matrix elements $\langle\varphi|A|\psi\rangle$, where $|\psi\rangle$ and $|\varphi\rangle$ are CT and A is an (efficiently computable) sparse operation. This theorem will immediately lead to the proof of theorem 1.

4.1 Definition of CT states

Throughout this paper, we will deal with n-qubit state families $\{|\psi_n\rangle : n = 1, 2, \ldots\}$, where $|\psi_n\rangle$ is an n-qubit state. When considering such a state family $\{|\psi_n\rangle\}$, we will mostly refer to a single state $|\psi_n\rangle \equiv |\psi\rangle$ with the silent assumption that this actually denotes a family. We now consider the following definition.

Definition 1 *An n-qubit state $|\psi\rangle$ is called 'computationally tractable' (CT) if the following conditions hold:*

(a) it is possible to sample in poly(n) time with classical means from the probability distribution Prob$(x) = |\langle x|\psi\rangle|^2$ on the set of n-bit strings x, and

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

For convenience, in (b) we require the coefficients $\langle x|\psi\rangle$ to be computable with perfect precision, a notion which may lead to rather pathological situations when e.g. irrational numbers are involved. The results in this paper can however straightforwardly be generalized to the case where $\langle x|\psi\rangle$ can be computed efficiently with exponential precision, i.e. up to m significant bits in poly(n, m) time. As in the present work the distinction between these two types of accuracies is not essential (in contrast to the distinction between polynomial and exponential precision, which *is* crucial), for clarity we state all results w.r.t. the notion of perfect accuracy. Also in other places in the text where we refer to 'perfect accuracy', the results in question immediately generalize to the case of exponential precision.

Note that (a) and (b) are highly dependent on the classical description of the state $|\psi\rangle$ that is provided. Therefore, strictly speaking it would be more precise to call a state $|\psi\rangle$ CT *relative to this classical description.* In this paper we will only encounter situations where each state has a natural (efficient) description that will be obvious from the context. It will always be assumed that this particular description is provided. For example, the classical description of a state generated by a polynomial size quantum circuit acting on, say, the all-zeroes input, will always be assumed to be the circuit that generates the state. As another example, for every complete product state $|\psi\rangle = |\psi_1\rangle \otimes \ldots \otimes |\psi_n\rangle$ we will assume $|\psi\rangle$ to be specified in terms of the 'obvious' description of $|\psi\rangle$ consisting if the $2n$ complex coefficients $\langle 0|\psi_i\rangle$ and $\langle 1|\psi_i\rangle$.

Even though conditions (a) and (b) are similar in nature, we provide evidence that these conditions are incomparable. In particular, the following complexity theoretic argument implies that it is highly likely that there exists states satisfying (b) but not (a). Consider any efficiently computable function $f : \{0,1\}^n \to \{0,1\}$ for which it is promised that there exists a unique x_0 such that $f(x_0) = 1$, and define the n-qubit state $|\psi\rangle = \sum_x f(x)|x\rangle = |x_0\rangle$. Note that the state $|\psi\rangle$ satisfies condition (b). Assuming that (b) implies (a), it follows that it is possible to efficiently sample from the distribution $\{|\langle x|\psi\rangle|^2\}$. But this distribution assigns a zero probability to each bit string x except x_0, which has unit probability. Hence, the possibility of efficiently sampling from this distribution implies that x_0 can be determined efficiently. Regarding f as a verifier circuit for an NP problem, it would immediately follow that every problem in NP with a unique witness is in P. This last property is not likely to be true [22].

Next we state a useful sufficient (but not necessary) criterion to assess whether condition (a) holds for a given state. To state this result, we need the following notation. For an n-qubit state $|\psi\rangle$, let $p_{S,y}(|\psi\rangle) \equiv p_{S,y}$ denote the probability of obtaining the bit string $y = (y_i : i \in S)$ as an outcome when measuring the qubits in the set $S \subseteq \{1, \ldots, n\}$. We can then state the following lemma; a proof can be found in e.g. [11].

Lemma 1 *Let $|\psi\rangle$ be an n-qubit state. Suppose that, on input of an arbitrary S and y, the probability $p_{S,y}$ can be computed in poly(n) time. Then it is possible to sample in poly(n) time from the probability distribution $\{|\langle x|\psi\rangle|^2\}$.*

Several important state families turn out to be CT, as illustrated next.

- **Examples of CT states:**

– Product states are trivially CT.

– Every state of the form $|\psi\rangle \propto \sum_x e^{i\theta(x)}|x\rangle$, where the sum is over all n-bit strings x and where $x \to \theta(x) \in \mathbf{R}$ represents an arbitrary efficiently computable function, is trivially CT. Every state obtained by applying a polynomial size circuit family consisting of Toffoli gates to an arbitrary product state is CT as well, as can easily be proved (this property will also follow from lemma 2).

– Every matrix product state (MPS) of polynomial bond dimension is CT. A state $|\psi\rangle$ is an MPS of poly bond dimension if there exist $2n$ $N \times N$ matrices $A_i[0], A_i[1]$ with $N = \mathrm{poly}(n)$ such that $\langle x|\psi\rangle = \mathrm{Tr}(A_1[x_1] \dots A_n[x_n])$, for every n-bit string $x = (x_1, \dots, x_n)$. Property (b) follows immediately from this definition. Property (a) holds since the conditions of lemma 1 are satisfied for all MPS of polynomial bond dimension [23]. Tree tensor states [24] are generalizations of MPS with similar properties and are also CT.

– A Clifford circuit is a quantum circuit composed of Hadamard, CNOT and PHASE gates, where PHASE $= \mathrm{diag}(1, i)$. An n-qubit stabilizer state is any state that is generated by applying a polynomial size Clifford circuit to the state $|0\rangle^n$. Every stabilizer state is a CT state. Property (a) is the content of the Gottesman-Knill theorem [6]. Property (b) is proved in [7] (see also [10]).

– A (unitary, two-qubit) matchgate G is any two-qubit gate of the form

$$G = \begin{bmatrix} a & & & b \\ & u & v & \\ & x & y & \\ c & & & d \end{bmatrix}, \quad A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad B = \begin{bmatrix} u & v \\ x & y \end{bmatrix}, \tag{1}$$

where $A, B \in SU(2)$. Every state obtained by applying a polynomial size matchgate circuit to a computational basis state, where all gates are restricted to act on nearest neighbors (assuming a one-dimensional ordering of the qubits) is a CT state. Properties (a) and (b) are proved in [11].

– Any n-qubit state that is obtained by applying the quantum Fourier transform (over the integers modulo 2^n) to an arbitrary product state, is a CT state. See e.g. [25] for a simple proof of this property (see also [26, 27] for related results).

– We briefly mention a general class of classical simulation results related to efficient tensor contraction schemes. This approach relies on the topology of (a graph associated with) the quantum circuit in question. If this topology displays a sufficiently tree-like structure (quantified in terms of the graph invariant *tree-width*) then classical simulation of such circuits can be achieved [28]. It can be shown that the output states of quantum circuits with logarithmically scaling tree-width (acting on product input states), are CT states; the proof essentially contained in [28] and is omitted here (see also [4] for related work).

4.2 Basis-preserving operations

Next we investigate which operations map the family of CT states to itself. In this context, the operations that preserve the computational basis play an important role. An n-qubit operation M is called 'basis-preserving' if every computational basis state $|x\rangle$ is mapped to $M|x\rangle = \gamma_x|\pi(x)\rangle$, for some permutation π of the set of n-bit strings and some complex number γ_x. The operation M is efficiently computable if the functions $x \to \gamma_x$, $x \to \pi(x)$ and $x \to \pi^{-1}(x)$ can be evaluated in poly(n) time. For example, every Pauli product [b] is efficiently computable basis-preserving, as well as every operation of the form $O = \sum_x (-1)^{f(x)}|x\rangle\langle x|$, where $f : \{0,1\}^n \to \{0,1\}$ is an efficiently computable function. Every polynomial size circuit composed of elementary basis-preserving gates (e.g. Toffoli gates, diagonal gates) is also efficiently computable basis-preserving.

The relevance of efficiently computable basis-preserving *unitary* operations in the present context is that these operations preserve the class of CT states:

Lemma 2 *If $|\psi\rangle$ is a CT n-qubit state and if M is an efficiently computable unitary basis-preserving operation, then $|\psi'\rangle = M|\psi\rangle$ is again CT.*

Proof: Let the permutation π and the coefficients γ_x be defined as above. Note that $|\gamma_x| = 1$ for every x since M is unitary. The coefficients of $|\psi'\rangle$ are given by $\langle x|\psi'\rangle = \gamma_{\pi^{-1}(x)}\langle\pi^{-1}(x)|\psi\rangle$. Property (b) now follows immediately from the properties that M is efficiently computable and that $|\psi\rangle$ is CT. To show (a), we have to find an efficient classical method to sample from the probability distribution defined by $\text{Prob}(x) = |\langle x|\psi'\rangle|^2 = |\langle\pi^{-1}(x)|\psi\rangle|^2$. To do so, consider the following procedure. First sample from the distribution $\{|\langle y|\psi\rangle|^2\}$, yielding a bit string y with probability $|\langle y|\psi\rangle|^2$, and subsequently output the bit string $x := \pi(y)$. This procedure is efficient since $|\psi\rangle$ is CT and $y \to \pi(y)$ is efficiently computable. Moreover, every bit string x is generated with probability $|\langle\pi^{-1}(x)|\psi\rangle|^2$ as desired. □

Note that the basis-preserving operation M may drastically change the entanglement properties of $|\psi\rangle$. Consider e.g. the case where $|\psi\rangle$ is a complete product state and M a polynomial size circuit of CPHASE and/or Toffoli operations, yielding a state $|\psi'\rangle$ that may be highly entangled. Nevertheless, both $|\psi\rangle$ and $|\psi'\rangle$ are CT and equal up to a basis-preserving operation.

4.3 Sparse operations

Next we consider sparse operations. Such operations are sufficiently close to basis-preserving operations that their action on CT states remains manageable. An n-qubit operation A is s-sparse if for every basis state $|x\rangle$, each of the vectors $A|x\rangle$ and $A^T|x\rangle$ is a linear combination of at most s computational basis states. The quantity s is called the sparseness of A. We will consider n-qubit operations A (both unitary operations and observables) with sparseness $s \leq$ poly(n), which will simply be called 'sparse operations'. Note that the notion of sparseness is defined w.r.t. to the number of nonzero entries per row/column and *not* the total number of nonzero entries in the matrix, the latter not being required to be small. In particular, a sparse n-qubit operation generically has a total number of nonzero entries that scales *exponentially*

[b] Recall that an n-qubit Pauli operator (or Pauli product) has the form $P = P_1 \otimes \ldots \otimes P_n$, where each P_i is either the 2×2 identity or one of the Pauli matrices X, Y or Z.

with n.

For every s-sparse n-qubit operation A, define $2s$ functions $\alpha_i : \{0,1\}^n \to \mathbf{C}$ and $r_i : \{0,1\}^n \to \{0,1\}^n$ $(i = 1, \ldots, s)$ as follows: the n-bit string $r_i(x)$ is defined to be the row index of A associated with the i-th non-zero entry in the column indexed by x (when traversing this column from top to bottom), if an i-th nonzero entry exists within this column; we denote this entry by $\alpha_i(x)$. If an i-th nonzero entry does not exist in this column, then $r_i(x)$ is set to be the all-zeroes string and $\alpha_i(x)$ is set to zero. With the above definitions, one simply has

$$A|x\rangle = \alpha_1(x)|r_1(x)\rangle + \ldots + \alpha_s(x)|r_s(x)\rangle. \tag{2}$$

Similar definitions can be given regarding the rows of A, leading to $2s$ functions $\beta_i : \{0,1\}^n \to \mathbf{C}$ and $c_i : \{0,1\}^n \to \{0,1\}^n$ $(i = 1, \ldots, s)$ that are the natural counterparts of the α_i and r_i, respectively.

A sparse n-qubit operation A is efficiently column-computable if, on input of an arbitrary n-bit string x, it is possible to list the (at most $s = \text{poly}(n)$) nonzero entries within the column of A indexed by x together with the row indices associated with each of these non-zero entries, all in $\text{poly}(n)$ time. Equivalently, A is efficiently column-computable if it is possible to compute the $2s$ quantities $\alpha_i(x)$ and $r_i(x)$ $(i = 1, \ldots, s)$ in polynomial time. The operation A is called efficiently row-computable if A^T is efficiently column-computable. Finally, A is called efficiently computable sparse (ECS) if it is both efficiently row- and column-computable. All ECS unitary operations can be implemented efficiently on a quantum computer [29]. In this paper we will only consider sparse operations that are efficiently computable.

The following are some examples of ECS operations.

- **Examples of ECS operations:**

– Every efficiently computable basis-preserving operation is ECS.

– Every d-qubit gate G acting within an n-qubit circuit, represented by the matrix $G \otimes I$ where I denotes the identity acting on $n - d$ qubits, is 2^d-sparse. If $d = O(\log n)$ then such an operation is ECS.

– Every operation that is a linear combination of $\text{poly}(n)$ ECS operations, is ECS. It follows that every operator $H = \sum_{i=1}^m H_i$ which is a sum of $m = \text{poly}(n)$ d-local observables H_i (with $d = O(\log n)$) is ECS. This means that observables such as Hamiltonians and correlation operators are typically ECS.

– Let U represent an n-qubit polynomial size circuit of basis-preserving elementary gates (e.g. Toffoli, CNOT, PHASE, CPHASE, etc.), interspersed with k gates $V_1, \ldots, V_k$ at arbitrary places in the circuit, each of which acts on at most d qubits. It is required that $kd = O(\log n)$; otherwise the V_i are arbitrary. Then U is ECS. To see this, expand each gate V_i as a linear combination of 4^d Pauli products and note that every Pauli product is efficiently computable basis-preserving. Consequently, U can be written as a linear combination of $4^{dk} = \text{poly}(n)$ efficiently computable basis-preserving operations, showing that U is ECS.

– ECS operations often arise in the context of quantum algorithms, related e.g. to unitary group representations; see e.g. [29] and references within.

– A product of d ECS operations, where d is some constant, is again ECS.

We are now in a position to state the following result, which constitutes the main technical ingredient in this work regarding the use of sampling techniques in classical simulation.

Theorem 3 *Let $|\psi\rangle$ and $|\varphi\rangle$ be CT n-qubit states and let A be an ECS (not necessarily unitary) n-qubit operation with $\|A\| \leq 1$. Then there exists an efficient classical algorithm to approximate $\langle\varphi|A|\psi\rangle$ with polynomial accuracy.*

Note that theorem 1 immediately follows from theorem 3. Before proving this result in its most general form, as a warm-up we prove a special instance, taking A to be the identity. Hence, we are concerned with the estimation of overlaps between CT states. This special case is proved beforehand to illustrate the sampling methods used in this work, without the more technically involved arguments required in the proof of theorem 3. Thus, we set out to prove the following property, formulated as a lemma.

Lemma 3 *Let $|\psi\rangle$ and $|\varphi\rangle$ be two CT n-qubit states. Then there exists an efficient classical algorithm to approximate $\langle\varphi|\psi\rangle$ with polynomial accuracy.*

Proof: Denote $p_x := |\langle x|\psi\rangle|^2$ and $q_x := |\langle x|\varphi\rangle|^2$. Since $|\psi\rangle$ and $|\varphi\rangle$ are CT states, it is possible to sample efficiently from the probability distributions $\{p_x\}$ and $\{q_x\}$. Define the function $\delta : \{0,1\}^n \to \{0,1\}$ by $\delta(x) = 1$ if $p_x \geq q_x$ and $\delta(x) = 0$ otherwise, for every n-bit string x, and define $\epsilon = 1 - \delta$. Then δ and ϵ can be evaluated efficiently since p_x and q_x can be efficiently evaluated by assumption (b) in the definition of CT states. The overlap $\langle\varphi|\psi\rangle$ is therefore equal to

$$\langle\varphi|\psi\rangle = \sum \langle\varphi|x\rangle\langle x|\psi\rangle\delta(x) + \sum \langle\varphi|x\rangle\langle x|\psi\rangle\epsilon(x), \tag{3}$$

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

$$F(x) = \frac{\langle\varphi|x\rangle\langle x|\psi\rangle}{p_x}\ \delta(x), \quad G(x) = \frac{\langle\varphi|x\rangle\langle x|\psi\rangle}{q_x}\ \epsilon(x), \tag{4}$$

we have $\langle\varphi|\psi\rangle = \langle F\rangle + \langle G\rangle$ where $\langle F\rangle = \sum p_x F(x)$ and $\langle G\rangle = \sum q_x G(x)$. It follows from assumption (b) in the definition of CT states that F and G can be efficiently evaluated. Furthermore, both $|F(x)|$ and $|G(x)|$ are not greater than 1. It thus follows from the Chernoff-Hoeffding bound that both $\langle F\rangle$ and $\langle G\rangle$ can be approximated efficiently with polynomial accuracy. This implies that $\langle\varphi|\psi\rangle$ can be estimated with polynomial accuracy as well. This completes the proof. □

Lemma 3 shows that the overlap $\langle\varphi|\psi\rangle$, representing a 'joint' property of the states $|\psi\rangle$ and $|\varphi\rangle$, may be estimated efficiently classically even when only an efficient simulation of quantum processes resulting in $|\psi\rangle$ and $|\varphi\rangle$ *individually* is available—in particular, the techniques leading to the proofs of (a)-(b) (cf. definition of CT states) for $|\psi\rangle$ and $|\varphi\rangle$, may be completely different. For example, the overlap between a matrix product state and a stabilizer state can be estimated efficiently classically with polynomial accuracy, even though such states are CT due to very different argumentations.

We are now in a position to prove theorem 3.

Proof of theorem 3: consider CT states $|\psi\rangle$ and $|\varphi\rangle$. Let $s = \text{poly}(n)$ denote the sparseness of A. Using the notation of (2), we have $\langle\varphi|A|\psi\rangle = \sum_{i=1}^{n} \sigma_i$, where we denote

$$\sigma_i := \sum_x \alpha_i(x)\langle\varphi|r_i(x)\rangle\langle x|\psi\rangle. \tag{5}$$

Note that $|\alpha_i(x)| \leq 1$. It is sufficient to prove that each of the s quantities σ_i can be estimated efficiently with polynomial accuracy, for then $\sum_{i=1}^{s} \sigma_i$ can also be estimated with polynomial accuracy as $s = \text{poly}(n)$. To do so, write $p_x := |\langle x|\psi\rangle|^2$ and $q_x := |\langle x|\varphi\rangle|^2$. Define a function δ_i by $\delta_i(x) = 1$ if $p_x \geq q_{r_i(x)}$ and $\delta_i(x) = 0$ otherwise, for every n-bit string x, and define $\epsilon_i = 1 - \delta_i$. Then δ_i and ϵ_i can be evaluated efficiently since $|\psi\rangle$ and $|\varphi\rangle$ are CT and A is ECS. We split σ_i in two parts by inserting $\delta_i(x) + \epsilon_i(x) = 1$:

$$\sigma_i = \sum \langle\varphi|r_i(x)\rangle\langle x|\psi\rangle\alpha_i(x)\delta_i(x) + \sum \langle\varphi|r_i(x)\rangle\langle x|\psi\rangle\alpha_i(x)\epsilon_i(x). \tag{6}$$

The function F_i defined by

$$F_i(x) = \frac{\langle\varphi|r_i(x)\rangle\langle x|\psi\rangle}{p_x}\ \alpha_i(x)\delta_i(x) \tag{7}$$

is efficiently computable and satisfies $|F_i(x)| \leq 1$ for every x. The first term in the r.h.s. of (6) is hence equal to $\langle F_i\rangle = \sum p_x F_i(x)$, which can be estimated to polynomial accuracy efficiently due to the Chernoff-Hoeffding bound. To estimate the second term in the r.h.s. of (6), one needs to be careful since the function r_i may not be injective. We proceed as follows. Define the following function G_i:

$$G_i(y) = \sum_{x:\ r_i(x)=y \ \text{ and } \ \alpha_i(x)\neq 0} \frac{\langle\varphi|y\rangle\langle x|\psi\rangle}{q_y}\ \alpha_i(x)\epsilon_i(x) \tag{8}$$

with the additional convention that $G_i(y)$ is zero if there are no x such that $r_i(x) = y$ and $\alpha_i(x) \neq 0$. With this definition, the second term in the r.h.s. of (6) is equal to $\langle G_i\rangle = \sum_y q_y G_i(y)$. We now make the following claims. *Claim 1*: the function G_i is efficiently computable; and *Claim 2*: $|G_i(y)| \leq s$ for every y. A proof of claims 1 and 2 implies that $\langle G_i\rangle$ can be estimated in polynomial time with polynomial accuracy due to the Chernoff-Hoeffding bound. But then σ_i can also be estimated efficiently, thus completing the proof.

We now prove Claim 1. Since A is s-sparse, every row y has at most s non-zero entries. Equivalently, the following set contains at most s strings x:

$$\{x : \exists j \in \{1, \ldots, s\} \text{ s.t. } y = r_j(x) \text{ and } \alpha_j(x) \neq 0\}. \tag{9}$$

Hence, a fortiori, for every fixed i there are at most s different x such that $r_i(x) = y$ and $\alpha_i(x) \neq 0$. Moreover, given an arbitrary y it is possible to efficiently determine all these x's and the corresponding coefficients $\alpha_i(x)$. This is done in two steps: first, since A is efficiently (row-)computable, given a row index y it is possible to compute all (at most s) strings x in the set (9) in polynomial time; second, for all those x one computes $r_i(x)$ and $\alpha_i(x)$—this is possible in polynomial time since A is efficiently column-computable—and verifies whether $r_i(x)$ is equal to y; those x for which $r_i(x) = y$ are kept, the others discarded.

It follows that $G_i(y)$ is a sum of at most $s = \text{poly}(n)$ terms, each of which is efficiently computable. Thus, Claim 1 is proved. Moreover, Claim 2 now immediately follows as well, since the modulus of every term in the sum (8) is smaller than one and there are at most s terms in the sum. This proves theorem 3. □

Remark: poly-ECS operations.— In the definition of ECS operations and in the subsequent statement of theorem 3, we have required that the non-zero entries of A can be computed efficiently with perfect precision. Theorem 3 also holds for sparse operations where, instead, these coefficients can be estimated efficiently with *polynomial* accuracy, which is a significant relaxation. Call an n-qubit operation A ($\|A\| \leq 1$) *poly-ECS* if it is sparse, and if (i) on input of an arbitrary column index x, it is possible to determine in polynomial time all those row indices y such that $\langle y|A|x\rangle \neq 0$ and if the corresponding nonzero entries $\langle y|A|x\rangle$ can be estimated in polynomial time with polynomial accuracy, and (ii) similarly for the row indices y. Theorem 3 then also holds for poly-ECS operations. The proof is completely analogous to the above proof of theorem 3. The only difference is that now the functions $F_i(x)$ and $G_i(x)$ can no longer be computed exactly, but only with polynomial accuracy. However, this suffices to invoke the Chernoff-Hoeffding bound (cf. the Appendix). This remark will play an important role in the discussion of Simon's algorithm i.e. in the proof of theorem 2.◇

We conclude this section with two corollaries of theorem 3. Corollary 1 shows that expectation values of local observables can be estimated efficiently classically for every CT state. This result may potentially be of use in e.g. variational Monte Carlo studies of strongly correlated systems (this is work in progress). Corollary 2 will be of use when we discuss the Deutsch-Jozsa algorithm in section 6.2.

Corollary 1 *Let $|\psi\rangle$ be an n-qubit CT state and let O be a d-local observable with $d = O(\log n)$ and $\|O\| \leq 1$. Then there exists an efficient classical algorithm to estimate $\langle\psi|O|\psi\rangle$ with polynomial accuracy.*

Proof: this result follows immediately from theorem 3 since every d-local O with $d = O(\log n)$ is ECS. Here we provide a short alternative proof that does not require the formalism used in the proof of theorem 3. Every observable O of the form considered can be written as a linear combination of $N = \text{poly}(n)$ Pauli operators: $O = \sum_{i=1}^{N} a_i P_i$, with $|a_i| \leq 1$. Consequently,

$$\langle O\rangle := \langle\psi|O|\psi\rangle = \sum a_i \langle\psi|P_i|\psi\rangle. \tag{10}$$

As each P_i is an efficiently computable basis-preserving unitary operation, each state $P_i|\psi\rangle$ is CT due to lemma 2. Invoking lemma 3, the overlap between $P_i|\psi\rangle$ and $|\psi\rangle$ can be estimated classically with polynomial accuracy. Hence, $\langle O\rangle$ can also be estimated classically with polynomial accuracy. This proves the result. □

Corollary 2 *Let $|\psi\rangle$ and $|\varphi\rangle$ be CT n-qubit states, let $|\xi\rangle$ and $|\chi\rangle$ be CT k-qubit states (with $k \leq n$) and let A and B be ECS n-qubit operations with $\|A\|, \|B\| \leq 1$. Then there exists an efficient classical algorithm to approximate $\langle\varphi|A[|\xi\rangle\langle\chi| \otimes I]B|\psi\rangle$ with polynomial accuracy.*

Proof: The proof uses a technique related to the SWAP test. Denote $|\psi'\rangle := B|\psi\rangle$ and $|\varphi'\rangle := A^\dagger|\varphi\rangle$ (which are potentially unnormalized states) and consider the following identity:

$$\langle\varphi'|[|\xi\rangle\langle\chi|\otimes I]|\psi'\rangle = [\langle\chi|\langle\varphi'|]U_{\mathrm{SWAP}}[|\xi\rangle|\psi'\rangle], \tag{11}$$

where the unitary operator U_{SWAP} swaps qubit i with qubit $i+k$, for every $i = 1, \ldots, k$. The identity (11) can easily be verified. Hence, we have

$$\langle\varphi|A[|\xi\rangle\langle\chi|\otimes I]B|\psi\rangle = [\langle\chi|\langle\varphi|][I\otimes A]U_{\mathrm{SWAP}}[I\otimes B][|\xi\rangle|\psi\rangle]. \tag{12}$$

Note that the $(k+n)$-qubit states $|\xi\rangle|\psi\rangle$ and $|\chi\rangle|\varphi\rangle$ are CT. Moreover, it can easily be verified that U_{SWAP} is ECS. This implies that the operation $[I\otimes A]U_{\mathrm{SWAP}}[I\otimes B]$ is ECS as well, being a product of three ECS operations. Theorem 3 can now be applied. □

Note that, as a special case of this last result, it follows that partial overlaps $\langle\varphi|[|\xi\rangle\langle\chi|\otimes I]|\psi\rangle$ between CT states can be estimated efficiently classically.

5 Applications of theorem 1

Next we discuss three applications of theorem 1 as announced in the introduction. These applications regard sparse circuits, composability, and CNOT-$e^{i\theta X}$ circuits.

5.1 Classical simulation of sparse circuits

The following is a formal statement of the classical simulation of sparse circuits which was announced in the introduction.

Corollary 3 *Let U be a circuit composed of m efficiently computable s-sparse unitary operations with $s^m = poly(n)$. The circuit acts on an arbitrary product input state and is followed by a Z measurement of the first qubit. Then this quantum computation can be simulated efficiently classically.*

Proof: Let $|\psi\rangle$ denote the product input state and let Z_1 denote the Z observable acting on the first qubit. The expectation value of Z_1 is given by $\langle Z_1\rangle = \langle\psi|U^\dagger ZU|\psi\rangle$. Note that U is ECS due to the restrictions on s and m; but then the observable $O := U^\dagger ZU$ is also ECS, being a product of three ECS operations. Moreover, $|\psi\rangle$ is a product state and hence CT. Theorem 1 can now be applied. □

As briefly alluded to in the introduction, sparse operations highlight the role of interference—as opposed to entanglement—in quantum computation. Note that sparse operations may generically produce highly entangled states. Consider e.g. the simple case where the input is $|+\rangle^n$ and the entire circuit U is composed of poly(n) CPHASE gates (which are basis-preserving gates and thus particularly simple examples of sparse operations). With such circuits, it is possible to efficiently generate e.g. the highly entangled cluster states [30]. On the other hand, if a sparse operation U acts on a state $|\psi\rangle$ then each coefficient of $U|\psi\rangle$ in the standard basis is a linear combination of at most poly(n) coefficients of $|\psi\rangle$. Hence, the "interference" in the process $|\psi\rangle \to U|\psi\rangle$ is limited (we use the notion of interference in a

colloquial sense and do not adopt any technical definition). Corollary 3 states that quantum computational processes where the interference is "small" in this sense, cannot offer any speed-up compared to classical computers, in spite of the high degrees of entanglement that may be generated throughout the computation. Corollary 3 may thus be regarded as complementary to a class of results stating that quantum computations that generate a low amount of entanglement (quantified appropriately) can be classically simulated efficiently (see e.g. [1, 2, 3, 4, 5]).

Finally, note that in corollary 3 one cannot hope for an improvement of the bound $s^m = \text{poly}(n)$ to e.g. $m = \text{poly}(n)$ and s constant (unless BQP = BPP) since *every* polynomial size quantum circuit is a product of $m = \text{poly}(n)$ single- and two-qubit gates, each of which is an s-sparse operation with s constant.

5.2 Composability

Theorem 1 immediately leads to a criterion to assess when the composition of two quantum circuits can be simulated classically. Formally, we have:

Corollary 4 *Consider polynomial size n-qubit quantum circuits U_1 and U_2, an input state $|\psi_{in}\rangle$ and an observable O with $\|O\| \leq 1$ such that: (i) the state $U_1|\psi_{in}\rangle$ is CT and (ii) the operation $U_2^\dagger O U_2$ is ECS. Then the circuit $U = U_2 U_1$, acting on $|\psi_{in}\rangle$ and followed by measurement of O, can be simulated efficiently classically.*

Next we provide some illustrations of this result. First we provide some examples of pairs (U, O) such that $U^\dagger O U$ is ECS. All circuit families U below are polynomial size.

- **Examples of pairs (U, O) where $U^\dagger O U$ is ECS:**

– Let U be a circuit of constant depth and let the observable O act nontrivially on $O(\log n)$ qubits. Then $U^\dagger O U$ also acts nontrivially on $O(\log n)$ qubits and is hence an ECS observable.

– Let U represent a Clifford circuit and let O be any observable that is a linear combination of $N = \text{poly}(n)$ Pauli products: $O = \sum_{i=1}^{N} a_i P^i$ with $|a_i| \leq 1$ and P^i Pauli operators. Then $U^\dagger O U$ is again a linear combination of N Pauli products, and hence ECS.

– Let U be a circuit composed of nearest-neighbor matchgates and let Z_1 denote the Pauli Z operation acting on the first qubit. Then U maps Z_1 (under conjugation) to a linear combination of $\text{poly}(n)$ Pauli products (see e.g. [14]), which is an ECS operation.

Next we explicitly describe two concatenated circuits that can be simulated efficiently using our results; see also Fig 1. In both examples, the circuit acts on the all-zeroes computational basis state and is followed by measurement of Z on the first qubit.

- **Examples of corollary 4:**

– Consider a quantum circuit $V = V_4 V_3 V_2 V_1$ where V_1 is an arbitrary local unitary operation, V_2 represents the quantum Fourier transform (over $\mathbf{Z}_{2^n}$), V_3 is an arbitrary ECS unitary, and V_4 is an arbitrary polynomial size (nearest-neighbor) matchgate circuit.

Then this circuit can be simulated efficiently classically due to corollary 4. In particular, we show that corollary 4 can be applied by taking $U_1 \equiv V_2V_1$ and $U_2 \equiv V_4V_3$. To see this, note first that V_2V_1 acting on the input yields a CT state. Further, $(V_4V_3)^\dagger Z(V_4V_3)$ is ECS: indeed, $V_4^\dagger ZV_4$ is a sum of poly(n) Pauli products and hence ECS, and thus $(V_4V_3)^\dagger Z(V_4V_3)$ is ECS as well, being a product of three ECS operations. Corollary 4 can now be applied.

– Consider a quantum circuit $V = V_4V_3V_2V_1$ where V_1 is an arbitrary polynomial size matchgate circuit, V_2 is a polynomial size circuit of Toffoli gates, V_3 is an arbitrary polynomial size Clifford circuit and V_4 is an arbitrary log-depth circuit consisting of nearest-neighbor gates. We show that corollary 4 can be applied by taking $U_1 \equiv V_1$ and $U_2 \equiv V_4V_3V_2$. To see this, note first that V_1 acting on the input yields a CT state. Further, $(V_4V_3V_2)^\dagger Z(V_4V_3V_2)$ is ECS: $V_4^\dagger ZV_4$ acts nontrivially on $O(\log n)$ qubits and is hence is a linear combination of poly(n) Pauli products; but then $V_3^\dagger V_4^\dagger ZV_4V_3$ is also a linear combination of poly(n) Pauli products (and hence ECS) since V_3 is a Clifford operation; finally, it follows that $(V_4V_3V_2)^\dagger Z(V_4V_3V_2)$ is ECS as this operation is a product of three ECS operations. Corollary 4 thus again yields the desired result.

Several other examples of the above nature can easily be generated.

5.3 Rotated bases and CNOT-$e^{i\theta X}$ circuits

In our definition of CT states and sparse operations, as well as in the resulting theorem 1, we have singled out a particular basis—i.e. the computational basis. Note, however, that in the vast majority of all arguments we have never relied on the specific form of this basis. Therefore, we may consider a generalized definition of CT states, sparse operations, etc., stated *relative to a arbitrary basis* $\mathcal{B}$, and carry out an analogous program as done so far, leading a much broader class of results. Results such as theorem 1 can be transferred in an obvious way, and will be omitted. Here we limit ourselves to discussing an example that can be understood using this generalized notion of CT states. This example regards the simulation of circuits composed of CNOTs and $e^{i\theta X}$ gates. Other examples of similar nature can easily be constructed.

Let $\mathcal{B} = \{|b_x\rangle\}$ denote the $|\pm\rangle$ product basis, defined by $|b_x\rangle \propto \bigotimes_{i=1}^n [|0\rangle + (-1)^{x_i}|1\rangle]$ for every n-bit string $x = (x_1, \ldots, x_n)$. A state is called 'computationally tractable in the basis $\mathcal{B}$' if it is possible to sample in poly(n) time with classical means from the probability distribution $\text{Prob}(x) = |\langle b_x|\psi\rangle|^2$, and if the coefficients $\langle b_x|\psi\rangle$ can be computed in poly(n) time classically. It is clear that $|\psi\rangle$ is CT in $\mathcal{B}$ iff $H^{\otimes n}|\psi\rangle$ is CT in the computational basis. For example, it can easily be shown that every stabilizer state, as well as any MPS $|\psi\rangle$ is CT in the $|\pm\rangle$-basis $\mathcal{B}$ as $H^{\otimes n}|\psi\rangle$ is in both cases CT in the computational basis.

Similarly, the notion of ECS operations w.r.t. $\mathcal{B}$ is defined in the natural way. Obviously, A is ECS w.r.t. $\mathcal{B}$ iff $H^{\otimes n}AH^{\otimes n}$ is ECS in the computational basis. For example, let U denote an arbitrary polynomial size n-qubit circuit composed of CNOT and $e^{i\theta X}$ gates, where θ may be any (real) angle. Whereas U is generally *not* ECS in the computational basis, this circuit is *always* ECS in the $|\pm\rangle$ basis $\mathcal{B}$. This can be seen as follows. Let CNOT_{ab} denote a CNOT gate with control a and target b. One then has the pair of identities

$$H^{\otimes 2}\text{CNOT}_{ab}H^{\otimes 2} = \text{CNOT}_{ba} \quad \text{and} \quad He^{i\theta X}H = e^{i\theta Z}, \tag{13}$$

both of which are easily verified. These identities imply that $M := H^{\otimes n} U H^{\otimes n}$ is a polynomial size circuit consisting entirely of CNOT and $e^{i\theta Z}$ gates and is thus ECS (even basis-preserving) in the computational basis. This shows that U is ECS in the $|\pm\rangle$ product basis.

One can now consider a generalized form of theorem 1, now stated relative to the $|\pm\rangle$ basis (or any other basis):

> **Theorem 1'** *Let $|\psi_{in}\rangle$ be an n-qubit state, let U denote a polynomial size n-qubit circuit and let O denote an observable with $\|O\| \leq 1$. If $|\psi\rangle$ is CT in $\mathcal{B}$ and if $U^\dagger O U$ is ECS in $\mathcal{B}$, then the circuit U, acting on $|\psi_{in}\rangle$ and followed by measurement of O, can be simulated efficiently classically.*

Now consider a CNOT-$e^{i\theta X}$ circuit U as above. The circuit U acts on an arbitrary product input $|\alpha\rangle$ and is followed by measurement of Z_1. We now claim that this computation can be simulated efficiently classically, using the above variant of theorem 1. To see this, first note that $|\alpha\rangle$ is CT in $\mathcal{B}$. Second, $O := U^\dagger Z_1 U$ is ECS in $\mathcal{B}$: to show this, note that $H^{\otimes n} O H^{\otimes n} = M^\dagger X_1 M$. Here, as before, $M := H^{\otimes n} U H^{\otimes n}$ is a polynomial size circuit consisting entirely of CNOT and $e^{i\theta Z}$ gates, and X_1 denotes the Pauli X operation acting on the first qubit. The operation $M^\dagger X_1 M$ is basis-preserving in the computational basis, hence $O = H^{\otimes n}[M^\dagger X_1 M] H^{\otimes n}$ is basis-preserving in $\mathcal{B}$. This proves the claim; note that we have hence proved:

Corollary 5 *Every polynomial size circuit composed of CNOT and $e^{i\theta X}$ gates (for arbitrary real θ), acting on an arbitrary product input and followed by measurement of Z_1, can be simulated efficiently classically.*

6 Simulating quantum algorithms

In this section we apply our results in the context of quantum algorithms. The idea is to consider e.g. theorems 1 and 3 and corollary 2 as a collection of 'tests' that every quantum algorithm claiming to achieve an exponential speed-up needs to pass. We will consider the three classes of algorithms mentioned in the introduction.

6.1 Potts models

Here we point out that a recently proposed quantum algorithm [20], concerned with estimating partition functions of classical spin systems such as the Potts model, can be simulated efficiently classically. Letting $\mathcal{Z}$ denote the Potts model partition function defined on some (arbitrary) lattice, the quantum algorithm in [20] provides a polynomial approximation of the quantity $\mathcal{Z}/\Delta$. Here Δ denotes a particular, easy-to-compute normalization factor that depends on the couplings of the model (see [20], Cor. 5.9, for the precise form of Δ); Δ is sometimes called the 'approximation scale' of the algorithm. On the other hand, in [31] mappings were established which allow to express the same quantity $\mathcal{Z}/\Delta$ as the overlap between a suitable product state $|\alpha\rangle$ and stabilizer state $|\psi\rangle$: $\mathcal{Z}/\Delta = \langle\alpha|\psi\rangle$. Note that both stabilizer states and product states are CT (see section 4). Using theorem 3 (in fact: the special instance $A = I$ of lemma 3, dealing with overlaps between CT states), we find that overlaps between stabilizer states and product states can also efficiently be estimated with polynomial accuracy with *classical* methods. Hence, the quantity $\mathcal{Z}/\Delta$ can also be estimated with polynomial

accuracy in polynomial time using classical means, showing that the quantum algorithm in question can be simulated efficiently classically.

We emphasize that the work [20] contains several quantum algorithms besides the partition function algorithm focused on here (in particular, the latter does not constitute the main result of [20]), including algorithms for BQP-complete problems, to which our classical simulation techniques do not apply.

6.2 Deutsch-Jozsa

An application of corollary 2 is found by considering the Deutsch-Jozsa (DJ) algorithm [21]. Recall that in the DJ problem one considers a black-box function $f : \{0,1\}^n \to \{0,1\}$ which is promised to be either constant or balanced [c] The task is to determine which possibility holds. Classically, any deterministic solution to the problem requires exponentially many oracle calls, whereas a randomized classical algorithm can solve the DJ problem with exponentially small probability of failure using $O(n)$ queries. The DJ quantum algorithm constitutes a deterministic solution to the problem using a single query of the oracle.

Thus, it is well known that DJ has an efficient classical solution when an exponentially small probability of failure is allowed. Here we will reproduce this result, showing that it immediately follows from corollary 2. Moreover, we will find that a large class of generalizations (to be specified below) can be efficiently simulated as well. The argument is very general and mainly regards the *structure* of the involved circuits.

Going through the steps in the DJ algorithm, it is easily verified that DJ is implemented by a circuit belonging to the following general class (the system is initialized in the state $|0\rangle^n$):

Round 1: apply a local unitary operator V_1 (i.e. $V_1 = V_{11} \otimes \ldots \otimes V_{1n}$);

Round 2: apply an ECS operation V_2;

Round 3: apply another local unitary operator V_3 ;

Round 4: measure the observable $O = |0\rangle\langle 0|^k \otimes I$, for some $k \leq n$.

Using corollary 2, we now immediately find that such a computation can be simulated efficiently classically. Indeed, the state obtained after Round 1 is a a product state and hence CT. Moreover, the operation in round 2 is ECS. Finally, the observable $O' := V_3^\dagger O V_3$ has the form $|\gamma\rangle\langle\gamma| \otimes I$ for some k-qubit product—and hence CT—state $|\gamma\rangle$. Corollary 2 now implies that the circuit can be simulated efficiently.

Remark. Strictly speaking, in DJ the operation V_2 is ECS in a slightly more general sense compared to the definition in section 4.3, due to the presence of the oracle. That is, V_2 is sparse with matrix elements that are classically computable in poly(n) time using poly(n) queries to the oracle. Remark that the results of section 4 carry over straightforwardly to the oracular case. In this context an "efficient" classical simulation of a quantum circuit involving oracles refers to a classical simulation that runs in poly(n) time using poly(n) queries to the classical oracle. ◇

[c] A function f is constant if $f(x) = f(0)$ for every n-bit string x; f is balanced if exactly 2^{n-1} inputs x satisfy $f(x) = 0$.

Note that, in the argument, the specific form of the function f (computed in Round 2) is completely irrelevant. This shows that the lack in computational power of the DJ algorithm is a *structural* feature of the circuit. In particular, this computational weakness cannot be overcome by e.g. changing the form of the oracle, but must involve a more drastic alteration of the circuit structure.

6.3 Simon's algorithm

Lastly, we consider Simon's algorithm [18]. As this algorithm has the admirable feature of being a very simple quantum algorithm that nevertheless achieves an exponential speed-up, it is an ideal candidate to compare quantum and classical computational power. Simon's algorithm is worth investigating from a number of angles. As a comprehensive study would lead us too far, here we single out one particular aspect, namely the surpising role of the round of classical postprocessing in the algorithm taking place *after* the measurement. We will show that this seemingly innocuous round of classical computation plays a rather determining role in the performance of the algorithm.

We first give a short review of Simon's algorithm in section 6.3.1. In section 6.3.2 we take small detour, discussing aspects of Fourier analysis of Boolean functions, which will be necessary to prove theorem 2; the latter is done in section 6.3.3.

6.3.1 Review of Simon's algorithm

Here we will focus on a decision problem version of Simon's problem, where it is asked to determine the i-th bit a_i of the unknown string a for some i. We will fix $i = 1$ in the following for concreteness.

Simon's quantum algorithm consists of the following steps. There are two registers, each consisting of n qubits, each initially prepared in the state $|0\rangle^n$. First a Hadamard operation is applied to every qubit in the first register. Second, the oracle operator U_f is applied, yielding $\sum_x |x\rangle|f(x)\rangle$. Third, again a Hadamard operation is applied to every qubit in the first register. This yields a state of the form $|\psi_{\text{out}}\rangle \propto \sum_{u\in\mathcal{V}} |u\rangle|\psi_u\rangle$. Here the sum is over all n-bit strings u that are orthogonal to a (w.r.t. modulo-2 arithmetic). We denote by $\mathcal{V}$ the subspace over $\mathbf{Z}_2$ of all such u. The $|\psi_u\rangle$ are (irrelevant) normalized states. Next, all qubits in the first register are measured in the computational basis, yielding a bit string u which is drawn uniformly at random from the subspace $\mathcal{V}$. Running this procedure N times, one generates the (Nn)-qubit state $|\psi_{\text{out}}\rangle^N$ and one subsequently obtains N bit strings $u^1, \ldots, u^N$, each drawn randomly from $\mathcal{V}$. We assemble these vectors as the rows as an $N \times n$ matrix, denoted by $\mathbf{u}$. If $N = O(n)$ then the probability that $u^1, \ldots, u^N$ do *not* span the entire space $\mathcal{V}$ is exponentially small in n. In the final step in the algorithm, one uses a classical computer to compute a solution x to the linear system of equations $\mathbf{u}x = 0$. More precisely, in the decision problem version of Simon's algorithm, a function $g : \{0,1\}^{nN} \to \{0,1\}$ is computed which takes the entries of the matrix $\mathbf{u}$ as input and which outputs 1 if there exists a solution x where the first bit of x is equal to 1; the output is zero otherwise. Note that g is efficiently computable classically. If the matrix $\mathbf{u}$ has rank $n-1$—which happens in all cases except for an exponentially small fraction—then there is a unique nontrivial solution i.e. $x = a$, in which case the function $g(\mathbf{u})$ correctly outputs the first bit of a.

In summary, Simon's algorithm can be implemented with an (Nn)-qubit circuit (where $Nn = \text{poly}(n)$) displaying the following structure; the circuit acts on the all-zeroes computa-

tional basis state.

Round 1: apply a Hadamard gate to some subset of qubits;

Round 2: apply an efficiently computable basis-preserving unitary operation;

Round 3: apply another round of Hadamard gates to some subset of the qubits; the latter subset is denoted by S;

Round 4: perform a computational basis measurement on all qubits in S. Denote by $\mathbf{u}$ the bit string containing all measurement outcomes.

Round 5: classically compute the value $g(\mathbf{u})$—which represents the output of the algorithm—where g is some efficiently computable Boolean function with a single-bit output.

In Simon's algorithm, round 2 is efficiently computable basis-preserving in a generalized sense due to the oracle, see the remark in section 6.2 regarding ECS operations in the presence of oracles.

For the time being, we will consider the above class of 5-round circuits in full generality, and ignore the specific forms of e.g. the functions f and g needed in Simon's algorithm.

6.3.2 Intermezzo: learning theory

In order to formally state and prove theorem 2, we briefly need to discuss some elementary concepts related to learning theory of Boolean functions (see e.g. [32]). Readers familiar with these concepts may immediately skip to section 6.3.3.

1. A Boolean function is any function $g : \{0,1\}^m \to \{0,1\}$. Every Boolean function can be written in a unique way as a multivariate polynomial $g(x) = \sum_S a_S x^S$ over $\mathbf{Z}_2$. In this expression, the sum ranges over all subsets $S \subseteq \{1,\ldots,m\}$. Moreover one has $a_S \in \mathbf{Z}_2$ and $x^S := \prod_{i\in S} x_i$ for every S, and arithmetic is performed over $\mathbf{Z}_2$. The ($\mathbf{Z}_2$-)degree of g is the size of the largest set S such that $a_S = 1$.

2. The Fourier transform $\hat{g} : \{0,1\}^m \to \mathbf{R}$ of g is defined as follows:

$$\hat{g}(u) = \sum_x (-1)^{u^T x + g(x)}, \tag{14}$$

for every m-bit string u. The quantities $\hat{g}(u)$ are called the Fourier coefficients of g. If the function g is computable in polynomial time (or provided as an oracle), and if a bit string u is provided as an input, then there exists an elementary polynomial time classical algorithm to estimate the quantity $2^{-m}\hat{g}(u)$ with polynomial accuracy. To see this, simply note that $2^{-m}\hat{g}(u)$ coincides with the expectation value of the (efficiently computable) function $x \to (-1)^{g(x)+u^T x}$ w.r.t. the uniform distribution, such that a polynomial approximation of $2^{-m}\hat{g}(u)$ can be achieved in polynomial time due to the Chernoff-Hoeffding bound.

3. A Boolean function is said to be s-sparse if it has precisely s nonzero Fourier coefficients. It is easily verified that every linear function is 1-sparse. In addition, it has been shown that every Boolean function corresponding to a polynomial of degree d is at least

2^d-sparse [33]. In this sense the sparseness of a Boolean function is an indication of its nonlinearity, since high-degree polynomials necessarily have many nonzero Fourier coefficients [d] A (family of) function(s) g is simply called 'sparse' if its sparseness satisfies $s \leq \text{poly}(m)$.

4. Interestingly, there exists an efficient algorithm to determine all Fourier coefficients of g that are greater than a given threshold value, in the following sense:

Lemma 4 *[35, 34] Suppose that one has access to an oracle computing a Boolean function g. Let $p(m)$ denote an arbitrary polynomial in m. Then there exists a polynomial time algorithm that outputs a collection of m-bit strings $\mathcal{T} \subseteq \{0,1\}^m$ of size poly(m) containing all u such that $2^{-m}|\hat{g}(u)| \geq (p(m))^{-1}$.*

Together with the remark made in 2, it follows that there exists a polynomial time algorithm that outputs the set $\mathcal{T}$ together with polynomial approximations of all the quantities $2^{-m}\hat{g}(u)$, for every $u \in \mathcal{T}$. Note that lemma 4 is a nontrivial result: indeed, a priori it is not obvious that the coefficients $\hat{g}(u)$ that lie above a certain threshold can be determined efficiently, since in principle there is an exponentially large space of bit strings u to be searched.

6.3.3 Proof of theorem 2

We are now in a position to formally state theorem 2:

Theorem 2 *Consider a quantum circuit displaying the 5-round structure as in section 6.3.1. If the function g computed in the round of classical postprocessing is sparse, then the entire circuit can be simulated efficiently classically.*

An important ingredient in the proof of theorem 2 will be the m-qubit operator W_g (where m denotes the number of bits on which g acts) defined by

$$\langle u|W_g|v\rangle = 2^{-m}\hat{g}(u+v) \quad \text{for every } u,v \in \{0,1\}^m. \tag{15}$$

Note that each row and each column of W_g contains precisely s non-zero entries, where s is the sparseness of g; in other words, *the Boolean sparseness of g and the sparseness of the operator W_g coincide*. This correspondence prompts the question of when the operator W_g is *efficiently computable* sparse. It can easily be seen that W_g is ECS if and only if (i) g is sparse and (ii) there exists an efficient algorithm to determine all those strings u such that $\hat{g}(u) \neq 0$ *and* the values of the corresponding coefficients $\hat{g}(u)$. Note however, that finding all u such that $\hat{g}(u) \neq 0$ is highly nontrivial since some of the non-zero Fourier coefficients may be exponentially small, yet nonzero. Moreover, for general (efficiently computable) g the problem of computing $\hat{g}(u)$ with exponential precision is #P-hard. Therefore, requiring W_g to be ECS is highly stringent.

[d] The converse, however, is not true. For example, there exist degree-2 polynomials with sparseness $s = 2^n$ (the inner product function is an example of this).

Fortunately, for our purposes the relevant question will be when W_g can be well-approximated by an ECS operation A with polynomial accuracy; moreover, A itself need not be ECS in the exact sense, but poly-ECS as discussed in the remark below theorem 3—these are much less stringent demands. Approximating W_g by such an A is actually possible *for every sparse function* g. This is shown in the following lemma; the proof relies on lemma 4.

Lemma 5 *Let g be a sparse Boolean function acting on m bits that is provided as an oracle, let the operator W_g be defined as in (15) and let $p(m)$ be an arbitrary polynomial. Then there exists a polynomial time classical algorithm that outputs a poly-ECS m-qubit operation A such that $\|W_g - A\| \leq p(m)^{-1}$.*

Proof: Let $s \leq \mathrm{poly}(m)$ denote the sparseness of g. Let $\theta > 0$ and let W_g^θ denote the matrix obtained by replacing all entries of W_g that are smaller in absolute value than θ, by zero. That is: $\langle u|W_g^\theta|v\rangle$ is equal to $2^{-m}\hat{g}(u+v)$ if $|2^{-m}\hat{g}(u+v)| \geq \theta$, and zero otherwise. For now, θ is arbitrary but below we will choose θ to be a suitable polynomial in m. Since W_g is s-sparse, the matrices W_g^θ and $W_g - W_g^\theta$ are s-sparse as well. Due to lemma 4 and the remark below it, for every $\theta = 1/\mathrm{poly}(m)$, the operator W_g^θ is poly-ECS. Next we show that θ can be tuned appropriately such $\|W_g - W_g^\theta\| \leq p(m)^{-1}$ is satisfied. To do so, let $\|\cdot\|_r$ ($\|\cdot\|_c$) denote the maximum row (column) sum norm [e]; these norms are related to the spectral norm $\|\cdot\|$ via the inequality $\|X\|^2 \leq \|X\|_r\|X\|_c$ for every matrix X [36]. As the matrix $W - W_g^\theta$ is s-sparse and as every entry of this matrix is at most θ in absolute value, it holds that $\|W - W_g^\theta\|_r \leq s\theta$ and $\|W - W_g^\theta\|_c \leq s\theta$, and hence

$$\|W - W_g^\theta\|^2 \leq \|W - W_g^\theta\|_r\|W - W_g^\theta\|_c \leq (s\theta)^2. \tag{16}$$

By choosing $\theta := (sp(m))^{-1}$ and setting $A := W_g^\theta$ with this choice of θ, we have found a matrix A satisfying the desired conditions. This completes the proof. □

Lemma 5 will be the key ingredient in the proof of theorem 2, which is provided next.

Proof of theorem 2: The analysis will be simplified by considering a slightly alternative version of the 5-round circuits in question, where now the entire computation is performed coherently and there is only a single measurement at the end of the computation. To achieve this, first one goes through rounds 1-3 as indicated. Second, the function $\mathbf{u} \to g(\mathbf{u})$ is computed coherently on the relevant registers, realized by a unitary operation U_g mapping $U_g : |\mathbf{u}\rangle \to |g(\mathbf{u})\rangle|\xi_\mathbf{u}\rangle$ for some (irrelevant) states $|\xi_\mathbf{u}\rangle$ [f]. Finally, the first qubit is measured in the computational basis. The overall circuit is denoted by U_T. Letting g be an arbitrary sparse function, we thus have to show that there exists an efficient classical algorithm to approximate $\langle Z_1\rangle = \langle \mathbf{0}|U_T^\dagger Z_1 U_T|\mathbf{0}\rangle$ (where $|\mathbf{0}\rangle = |00\ldots\rangle$) with polynomial accuracy. For further reference, we denote by $|\psi_2\rangle$ the state obtained after round 2; furthermore, $\mathcal{H}$ denotes the tensor product of Hadamard gates applied in round 3. Moreover, let $p(n)$ denote an arbitrary polynomial in n.

[e] That is, $\|X\|_r := \max_i \sum_{j=1}^N |X_{ij}|$ and $\|X\|_c := \max_j \sum_{i=1}^N |X_{ij}|$, for every $N \times N$ matrix X.

[f] We remark that in the definition of U_g, in the most general case one must allow U_g to use $m = \mathrm{poly}(n)$ ancillary qubits prepared in, say, the state $|0\rangle$, i.e. $U_g : |\mathbf{u}\rangle|0\rangle^m \to |g(\mathbf{u})\rangle|\xi_\mathbf{u}\rangle$. For clarity, we have not incorporated this in the proof; the argument can be generalized appropriately without significant complications.

First, remark that the state $|\psi_2\rangle$ is CT. Denoting $O := \mathcal{H}U_g^\dagger Z_1 U_g \mathcal{H}$, one has $\langle Z_1\rangle = \langle\psi_2|O|\psi_2\rangle$. *It is now crucial to note that $O = W_g$, where W_g is defined in Eq. (15)*; this identity can easily be verified. This allows us to invoke lemma 5, yielding in polynomial time a poly-ECS operation A satisfying $\|W_g - A\| \leq p(n)^{-1}$. Since A is poly-ECS and since $|\psi_2\rangle$ is CT, according to theorem 3 (cf. also the remark below it) it is possible to approximate $\langle\psi_2|A|\psi_2\rangle$ with polynomial accuracy in polynomial time with classical means. In particular, it is possible to efficiently generate a number c such that $|c - \langle\psi_2|A|\psi_2\rangle| \leq p(n)^{-1}$. Since $\langle Z_1\rangle = \langle\psi_2|W_g|\psi_2\rangle$, we then have

$$\begin{aligned} |c - \langle Z_1\rangle| &\leq |c - \langle\psi_2|A|\psi_2\rangle| + |\langle\psi_2|(W_g - A)|\psi_2\rangle| \\ &\leq p(n)^{-1} + \|W_g - A\| \leq 2p(n)^{-1}. \end{aligned} \tag{17}$$

In the first inequality we have used the triangle inequality; in the second inequality we have used that $|\langle\psi_2|(W_g - A)|\psi_2\rangle|$ is not greater $\|W_g - A\|$; in the third inequality, we have used that $\|W_g - A\| \leq p(n)^{-1}$. This hence shows that a polynomial approximation of $\langle Z_1\rangle$ can be achieved in polynomial time, thus proving the claim. □

We now specialize the discussion to Simon's algorithm. Note that the classical postprocessing in this algorithm is particularly simple, as it merely involves solving a system of linear equations over $\mathbf{Z}_2$. Nevertheless, the function g needed in Simon's algorithm is highly non-sparse. The intuition of the argument is that the function $g(\mathbf{u})$ is related to the computation of the determinant of a suitable matrix (or an analogous function in the case of non-square matrices), since the function g decides whether there exists a nontrivial solution to a certain system of linear equations. It is known that the determinant function $X \to \det(X)$ corresponds to a polynomial of degree k in the case of $k \times k$ matrices X, i.e. the degree of the polynomial is the square root of the input size k^2 of the determinant function. As the degree of a polynomial provides a lower bound to the logarithm of the sparseness (see point 3 in section 6.3.2), it follows that the determinant function has exponentially high sparseness $s \geq 2^k$. An analogous argument can be used to show that the function g considered in Simon's algorithm has high sparseness parameter s.

Looking at the problem differently, one can in fact use the proved $O(2^{\frac{n}{2}})$ classical oracle lower bound for Simon's problem to immediately *infer* that the function g *cannot be sparse*. Indeed, if g were sparse then our classical simulation results would imply the existence of a classical algorithm to solve Simon's problem using poly(n) classical oracle queries, which is provably not possible. Note that it is remarkable that the classical query lower bound for the oracle f can hence be used to infer properties of another function g!

7 Matchgates and polynomial time classical computation

We conclude this paper with a result regarding the computational power of matchgate circuits. While seemingly disconnected from the rest of the paper, this result will actually follow from our discussion of Simon's algorithm.

Call a family of functions $f_n : \{0,1\}^n \to \{0,1\}$ *efficiently matchgate-computable* if there exists a family of nearest-neighbor matchgate-circuits U_n acting on $M_n = \text{poly}(n)$ qubits $(n = 1, 2, \ldots)$, such that U_n, acting on $|x\rangle|0\rangle^{M_n - n}$ and followed by a $\{|0\rangle, |1\rangle\}$ measurement on the first qubit, yields the output $f(x)$ with probability $p \geq 2/3$, for all n-bit strings x. The

circuit family is to be polynomial time uniformly generated in the sense that the description of U_n is to be polynomial time computable from the number n. Our result is the following.

Theorem 4 *There exist functions that are efficiently computable classically (i.e. functions in P) that are not efficiently matchgate-computable.*

An interesting feature of this result is its proof method. Surprisingly, the proof will follow from our analysis of Simon's algorithm—even though the latter seems to have nothing to do with matchgates! Roughly speaking, we will show that if theorem 4 were *false*, then there would exist a quantum circuit to solve Simon's problem that can be simulated classically with our methods—hence resulting in a *classical* algorithm for Simon's problem that requires only poly(n) queries to the oracle. As the latter has been proved to be an impossibility, this will show that theorem 4 has to be true.

In the proof of theorem 4 we will need the following simple application of corollary 4.

> **Fact 1:** Consider an n-qubit quantum circuit $V = V_4V_3V_2V_1$ where both V_1 and V_3 represent collections of Hadamards applied to subsets of the qubits, V_2 is efficiently computable basis-preserving, and V_4 is a polynomial size (nearest-neighbor) matchgate circuit. Then any such circuit (acting on $|0\rangle^n$ and followed by measurement of Z_1) can be simulated efficiently classically due to corollary 4, taking $V_2V_1 \equiv U_1$ and $V_4V_3 \equiv U_2$. Indeed, $V_2V_1|0\rangle^n$ is CT and $(V_4V_3)^\dagger Z_1(V_4V_3)$ is a linear combination of poly(n) Pauli products [14] and hence ECS.

Proof of theorem 4: Consider the following variant $\tilde{g}$ of the function g computed in the classical postprocessing in Simon's algorithm: $\tilde{g}$ takes an $N \times n$ matrix $\mathbf{u}$ together with an integer i between 1 and n (specified in terms of $\log n$ bits) as its inputs, and outputs 1 if and only if there exists a bit string $x = (x_1, \ldots, x_n)$ satisfying $\mathbf{u}x = 0$ and $x_i = 1$. Note that $\tilde{g}$ is efficiently computable classically. We claim that $\tilde{g}$ is *not* efficiently matchgate-computable. To prove this, we show that the converse leads to a contradiction. Suppose that $\tilde{g}$ is matchgate-computable and let U denote the (family of) matchgate circuit(s) that computes $\tilde{g}$. Now consider the following quantum algorithm $\mathcal{A}$: first prepare the state $|i\rangle \otimes |\psi_{\text{out}}\rangle^{\otimes N}$, where $N = O(n)$ and where $|\psi_{\text{out}}\rangle \propto \sum_{u\in\mathcal{V}} |u\rangle|\psi_u\rangle$ as in section 6.3.1; up to a permutation of the qubits, at this point the state of the quantum register has the form $\sum_{\mathbf{u}} |i\rangle|\mathbf{u}\rangle|\chi_{\mathbf{u}}\rangle$ for some (irrelevant) normalized $|\chi_{\mathbf{u}}\rangle$, and where the sum is over all $N \times n$ matrices $\mathbf{u}$ for which each row belongs to $\mathcal{V}$. Second, apply the matchgate circuit U on the relevant registers in order to compute $|i, \mathbf{u}\rangle \to |\tilde{g}(i, \mathbf{u})\rangle$ in superposition—note that at this point it is crucial that U only depends on the input size but not on the entire input. Finally, measure Z_1 and let $\langle Z_1\rangle$ denote the expectation value of Z_1. Recall that all but an exponentially small fraction of the matrices $\mathbf{u}$ have rows which span the space orthogonal to a. Therefore, if the i-th bit of the unknown string a in Simon's problem is equal to 1, $\tilde{g}$ will evaluate to 1 for all but exponentially few $|\mathbf{u}\rangle$. Since U computes g with success probability at least 2/3 (cf. definition of matchgate-computability), the total probability $p(1)$ that the quantum circuit outputs 1 satisfies $p(1) \geq 2/3 - \delta$ for some exponentially small $\delta \geq 0$, so that $\langle Z_1\rangle = 1 - 2p(1) \leq -1/3 + O(\delta)$. On the other hand, if the i-th bit of a is 0 then $\langle Z_1\rangle \geq 1/3 - O(\delta)$. It is now easily verified that the algorithm $\mathcal{A}$ is implemented with a circuit displaying the structure considered in Fact

1. Hence a polynomial approximation of $\langle Z_1 \rangle$ can be classically achieved in polynomial time with exponentially small probability of failure, for every i. Note that such an approximation allows to decide whether $\langle Z_1 \rangle$ lies exponentially close to $1/3$ or $-1/3$. This hence leads to a polynomial time *classical* algorithm to determine a. This comprises a contradiction, given the $O(2^{\frac{n}{2}})$ classical query lower bound for Simon's problem. Hence, g cannot be efficiently matchgate-computable. □

Acknowledgements

I am very grateful to R. Jozsa for discussions and suggestions on the manuscript, and to H. Briegel, I. Cirac, W. Dür, G. Giedke, B. Kraus, R. Renner, N. Schuch and K. Vollbrecht for discussions. Work supported by the excellence cluster MAP.

References

1. R. Jozsa and N. Linden (2002), *On the role of entanglement in quantum computational speed-up*, arXiv:quant-ph/0201143.
2. G. Vidal (2003), *Efficient Classical Simulation of Slightly Entangled Quantum Computations*, Phys. Rev. Lett. **91**, 147902.
3. N. Yoran and A. Short (2006), *Classical simulation of limited-width cluster-state quantum computation*, quant-ph/060117.
4. R. Jozsa (2006), *On the simulation of quantum circuits*, arXiv:quant-ph/0603163 .
5. M. Van den Nest, W. Dür, G. Vidal and H. J. Briegel (2007), *Classical simulations versus universality in measurement-based quantum computation*, Phys. Rev. A **75**, 012337.
6. D. Gottesman (1998), *The Heisenberg Representation of Quantum Computers*, talk at International Conference on Group Theoretic Methods in Physics, arXiv:quant-ph/9807006.
7. J. Dehaene and B. De Moor (2003), *The Clifford group, stabilizer states, and linear and quadratic operations over GF(2)*, Phys. Rev. A **68**, 042318.
8. S. Aaronson and D. Gottesman (2004), *Improved Simulation of Stabilizer Circuits*, Phys. Rev. A 70:052328.
9. S. Clark, R. Jozsa, N. Linden (2007), *Generalised Clifford groups and simulation of associated quantum circuits*, arXiv:quant-ph/0701103.
10. M. Van den Nest (2010), *Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond*, Quant. Inf. Comp. **10** No. 3-4, pp0258-0271.
11. L. G. Valiant (2002), *Quantum Computers that can be Simulated Classically in Polynomial Time*, SIAM J. Comput. **31**, No. 4, p. 1229.
12. D. DiVincenzo and B. Terhal (2002), *Classical simulation of noninteracting-fermion quantum circuits*, Phys. Rev. A **65**, 032325/1-10.
13. S. Bravyi (2009), *Contraction of matchgate tensor networks on non-planar graphs*, Cont. Math., Vol. 482, pp. 179-211.
14. R. Jozsa and A. Miyake (2008), *Matchgates and classical simulation of quantum circuits*, Proc. R. Soc. A 464, 3089-3106.
15. R. Jozsa, B. Kraus, A. Miyake, J. Watrous (2010), *Matchgate and space-bounded quantum computations are equivalent*, Proc. R. Soc. A **466**, 809-830.
16. D. J. Brod and E. F. Galvao (2011), *Extending matchgates into universal quantum computation*, arXiv:1106.1863.
17. A. Y. Kitaev (1995), *Quantum measurements and the Abelian Stabilizer Problem*, quant-ph/9511026.
18. D. Simon (1997), *On the power of quantum computation*, SIAM J. Computing 26, 1474-1483.
19. Y. Shi (2002), *Both Toffoli and Controlled-NOT need little help to do universal quantum computation*, arXiv:quant-ph/0205115.

20. I. Arad and Z. Landau (2008), *Quantum computation and the evaluation of tensor networks*, arXiv:0805.0040.
21. D. Deutsch and R. Jozsa (1992), *Rapid solution of problems by quantum computation*, Proc. Roy. Soc. A 439: 553.
22. L. G. Valiant and V. V. Vazirani (1985), *NP is as easy as detecting unique solutions*, ACM Press New York, NY, USA.
23. D. Perez-Garcia, F. Verstraete, M.M. Wolf, J.I. Cirac (2007), *Matrix Product State Representations*, Quantum Inf. Comput. **7**, 401.
24. Y. Shi, L.-M. Duan and G. Vidal (2006), *Classical simulation of quantum many-body systems with a tree tensor network*, Phys. Rev. A **74**, 022320.
25. D. E. Browne (2007), *Efficient classical simulation of the semi-classical Quantum Fourier Transform*, New J. Phys. **9** 146.
26. N. Yoran and A. Short (2007), *Efficient classical simulation of the approximate quantum Fourier transform*, Phys. Rev. A **76**, 042321.
27. D. Aharonov, Z. Landau and J. Makowsky (2006), *The quantum FFT can be classically simulated*, quant-ph/0611156.
28. I. Markov and Y. Shi (2008), *Simulating quantum computation by contracting tensor networks*, SIAM J. Comp., 38(3):963-981.
29. S. P. Jordan and P. Wocjan (2009), *Efficient quantum circuits for arbitrary sparse unitaries*, arXiv:0904.2211.
30. M. Hein et al., *Entanglement in Graph States and its Applications*, In *Proceedings of the International School of Physics 'Enrico Fermi' on 'Quantum Computers, Algorithms and Chaos'* (2005); arXiv:quant-ph/0602096.
31. M. Van den Nest, W. Dür and H. J. Briegel (2007), *Classical spin models and the quantum stabilizer formalism*, Phys. Rev. Lett. **98**, 117207.
32. Y. Mansour (1994), *Learning Boolean Functions via the Fourier Transform*, Theoretical Advances in Neural Computation and Learning, 391-424, Kluwer Academic Publishers.
33. A. Bernasconi and B. Codenotti (1999), *Spectral Analysis of Boolean Functions as a Graph Eigenvalue Problem*, IEEE Trans. Computers **48**, 3, 345351.
34. Oded Goldreich and Leonid A. Levin (1989), *A hard-core predicate for all one-way functions*, In Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing, pp. 2532.
35. E. Kushilevitz and Y. Mansour (1993), *Learning decision trees using the Fourier spec- trum*, Siam J. Comput. 22, no. 6, 13311348.
36. R. A. Horn and C. R. Johnson (1990), *Matrix Analysis*, Cambridge Univ. Press.

Appendix A Sampling and the Chernoff-Hoeffding bound

The Chernoff-Hoeffding bound is a tool to assess with which precision the expectation value of a random variable may be approximated in terms of 'sample averages'. This bound asserts the following. Let $X_1, \ldots X_K$ be i.i.d. real-valued random variables with $E := \mathbf{E}X_i$ and $X_i \in [-1, 1]$. Then

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

In the case of complex-valued random variables X_i, a similar bound can be obtained for $|X_i| \leq 1$ by splitting X_i in its real and imaginary part and using (A.1) on both of these parts. In this work we will consider the Chernoff-Hoeffding bound in the following context. Let $\mathcal{P} := \{p_x\}$ be a probability distribution on the set of n-bit strings $x \in \{0,1\}^n$ and let $x \to F(x) \in \mathbf{C}$ be a complex function such that $|F(x)| \leq 1$ for every x. Let $\langle F\rangle = \sum_x p_x F(x)$

denote the expectation value of F. The goal is to approximate $\langle F \rangle$ by sampling from the distribution $\mathcal{P}$. To do so, consider K n-bit strings $x^1, \ldots, x^K$ drawn (independently) from the distribution $\mathcal{P}$, and denote the average $\sigma := K^{-1} \sum_{i=1}^{K} F(x^i)$. The Chernoff-Hoeffding bound then implies the following. For every $\epsilon = p(n)^{-1}$, where $p(n)$ represents an arbitrary polynomial in n, there exists a K that scales at most polynomially with n, such that the inequality $|\sigma - \langle F \rangle| \leq \epsilon$ holds with a probability that is exponentially (in n) close to 1. In other words, by taking poly(n) samples x^i it is possible to estimate $\langle F \rangle$ with an error that scales as $p(n)^{-1}$ for every choice of $p(n)$. We will henceforth denote this type of estimate as an approximation with 'polynomial accuracy' or a 'polynomial approximation'. Note that a polynomial approximation achieves an estimate of $\langle F \rangle$ up to $O(\log n)$ significant bits.

Moreover, if the function F can be evaluated in polynomial time *and* if it is possible to sample in polynomial time from $\mathcal{P}$, then the quantity σ can be computed in polynomial time. Hence, an overall efficient method is achieved to compute a polynomial approximation of $\langle F \rangle$ with exponentially small probability of failure. In this paper we will mostly ignore the fact that the Chernoff-Hoeffding bound yields polynomial approximations that do not succeed with unit probability but rather with a probability that is exponentially close to one. When the notion of a polynomial approximation is considered in the text, we will mean a polynomial approximation that is achieved with a probability that is exponentially close to one.

We discuss two immediate generalizations of the above arguments. First, above we have required that the function F can be evaluated with perfect precision in polynomial time. Such perfect accuracy is in this context not necessary. In particular, with similar methods as above, a polynomial approximation of $\langle F \rangle$ can be achieved in polynomial time if $F(x)$ *itself can be approximated with polynomial accuracy in polynomial time.* This can be seen as follows. Suppose that, on input of an arbitrary x, a polynomial approximation of $F(x)$ can be achieved in polynomial time. Let $p(n)$ be an arbitrary polynomial and consider K n-bit strings $x^1, \ldots, x^K$ drawn from the distribution $\mathcal{P}$ as before. Then for large enough K (where K scales as a polynomial in n with suitably high degree), $K^{-1} \sum_{i=1}^{K} F(x^i)$ lies ϵ-close to $\langle F \rangle$, where $\epsilon = (2p(n))^{-1}$. As each of the K quantities $F(x^i)$ can be approximated with polynomial accuracy in polynomial time by assumption, it is possible to efficiently generate K complex numbers c^i $(i = 1, \ldots, K)$ such that $|c^i - F(x^i)| \leq (2p(n))^{-1}$. Using the triangle inequality and denoting $c := K^{-1} \sum_{i=1}^{K} c^i$, it then easily follows that $|\langle F \rangle - c| \leq p(n)^{-1}$.

Second, so far we have considered functions F satisfying $\|F\| := \max_x |F(x)| \leq 1$. Note that similar conclusions can be reached for functions satisfying $\|F\| \leq \text{poly}(n)$.

The discussion in the present section can be summarized as follows.

Theorem 5 (Chernoff-Hoeffding bound) *Suppose that it is possible to sample in polynomial time with classical means from a probability distribution $\{p_x\}$ on the set of n-bit strings. Let $F : \{0,1\}^n \to \mathbf{C}$ denote a function satisfying $\|F\| \leq$ poly(n). Moreover, suppose that it is possible to efficiently estimate $x \to F(x)$ with polynomial accuracy on a classical computer. Then there exists an efficient classical algorithm to estimate $\langle F \rangle$ with polynomial accuracy.*