PHYSICAL REVIEW LETTERS

Computational Complexity of Projected Entangled Pair States

Norbert Schuch,[1] Michael M. Wolf,[1] Frank Verstraete,[2] and J. Ignacio Cirac[1]

[1]*Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Str. 1, D-85748 Garching, Germany*
[2]*Fakultät für Physik, Universität Wien, Boltzmanngasse 5, A-1090 Wien, Austria*

We determine the computational power of preparing projected entangled pair states (PEPS), as well as the complexity of classically simulating them, and generally the complexity of contracting tensor networks. While creating PEPS allows us to solve *PP* problems, the latter two tasks are both proven to be #*P*-complete. We further show how PEPS can be used to approximate ground states of gapped Hamiltonians and that creating them is easier than creating arbitrary PEPS. The main tool for our proofs is a duality between PEPS and postselection which allows us to use existing results from quantum complexity.

Introduction.—Computing the properties of correlated quantum many-body systems is a central task in many fields in physics. Its complexity stems mainly from the large dimension of the Hilbert space which grows exponentially in the system size. In the last decades, the Density Matrix Renormalization Group (DMRG) method has proven extremely successful in the description of one-dimensional phenomena [1]. Recently, it has been shown that from the perspective of quantum information, DMRG can be described as a variational method over the class of Matrix Product States (MPS) [2]. MPS structure the state space into a hierarchy of states with polynomial description complexity [3], and it turns out that already the lowest levels of this hierarchy approximate many physical states of interest extremely well. MPS have a natural extension to two and higher dimensional lattices, called Projected Entangled Pair States (PEPS), which also have an efficient description and are promising candidates for variational methods in higher dimensions [4]. It has been shown that MPS can be created efficiently by a quantum computer [5], and that they also can be simulated efficiently classically [6]. In contrast, in two or more dimensions, it seems to be hard to create arbitrary PEPS as well as to classically compute expectation values. In fact, it has been shown that there exist 2D PEPS which encode solutions to *NP*-complete problems [7], thus posing lower bounds on their complexity and computational power.

In the present Letter, we determine both the power of creating PEPS and the complexity of classically simulating them. We investigate which kind of problems we could solve if we had a way to efficiently create PEPS, and find that these are exactly the problems in the complexity class *PP* (deciding whether a boolean formula has more satisfying than nonsatisfying assignments). Second, we show that classically computing local expectation values on PEPS is a #*P*-complete problem (counting the satisfying assignments of a boolean formula). This result can be extended

to the contraction of arbitrary tensor networks, which turns out to be #*P*-complete as well.

The main tool in our proofs is a duality between PEPS and postselection, which permits us to use existing results from quantum complexity [8]: any PEPS can be created by a postselected quantum circuit, and any output of such a circuit can be written as a PEPS. We also apply this duality to show that ground states of gapped local Hamiltonians in *D* dimensions can be efficiently approximated by the boundary of a $(D + 1)$-dimensional PEPS. Finally, we compare the power of creating PEPS to the power of creating ground states of local Hamiltonians. While in general they are equally hard, we find that when restricting to gapped Hamiltonians, creating ground states becomes easier: it is in the weaker class *QMA*, the quantum analogue of *NP*.

PEPS and postselection.—We start by recalling the definition of PEPS [9]. Consider an arbitrary undirected graph where each of the vertices corresponds to a quantum system (a spin) of Hilbert space dimension *d*. A PEPS on these *N* spins is constructed by placing as many virtual spins of dimension *D* on each vertex as there are adjacent edges. Along each edge, these virtual spins form maximally entangled states $\sum_{i=1}^{D} |i\rangle|i\rangle$. The physical spins are now obtained from the virtual ones by applying a linear map $P^{[v]}:\mathbb{C}^D \otimes \ldots \otimes \mathbb{C}^D \to \mathbb{C}^d$ at each vertex *v*. For the sake of readability, we will mostly suppress the dependence of *P* on *v*. The graph underlying the PEPS will usually be chosen according to the physical setup, typically a two or higher dimensional lattice.

Let us now turn to postselected quantum circuits [8]. Roughly speaking, postselection means that we can measure a qubit with the promise of obtaining a certain outcome. More precisely, the postselected circuits we consider start from the $|0\cdots0\rangle$ state, perform a sequence of unitary one- and two-qubit gates, and postselect on the first qubit being $|0\rangle$. Thereby, the state $\alpha|0\rangle|\phi_0\rangle + \beta|1\rangle|\phi_1\rangle$ is pro-

jected onto the state $|\phi_0\rangle$, which is the state created by the postselected quantum circuit. Note that a state with $\alpha = 0$ will not be considered a valid input.

In the following, we show that the output of a postselected quantum circuit can be expressed efficiently as a PEPS on a 2D square lattice with both $D = d = 2$. We start by briefly recalling the concept of measurement based quantum computation [10]: One starts from the 2D cluster state (which is a PEPS with $D = d = 2$ [9]) and implements the quantum circuit by a sequence of projective measurements on the individual spins. Finally, the output is found in the unmeasured qubits, up to Pauli corrections which depend on the previous measurement outcomes. In order to express the output of a postselected circuit as a PEPS, we therefore start by implementing its unitary part in the measurement based model. We do this by projecting each qubit on the outcome $|a\rangle$ which does not give a Pauli correction, by replacing the original cluster projector P_C with $|a\rangle\langle a|P_C$. This leaves us with a set of qubits holding the output of the circuit, and by projecting the first qubit on $|0\rangle$, we obtain the output of the postselected quantum circuit. The transformation between the representations can be carried out efficiently, and the resulting PEPS has a size polynomial in the length of the circuit.

Conversely, any PEPS can be efficiently created by a postselected quantum computer since any linear operator P can be implemented deterministically using postselection. This is accomplished by implementing a POVM measurement [18] containing P as an element and postselecting the ancilla register on the outcome corresponding to P.

In summary, on the one side, we have that any postselected quantum circuit can be translated efficiently into a 2D PEPS with $D = d = 2$, while conversely there is also an efficient transform from any PEPS to a postselected quantum circuit. In turn, this shows that all the features and the full complexity of PEPS can already be found in the simplest case of two-dimensional PEPS, making them an even more interesting subject for investigations.

The power of creating PEPS.—Let us first briefly introduce the complexity classes #P and PP [11]. Consider an efficiently computable boolean function $f:\{0, 1\}^N \rightarrow \{0, 1\}$, and let $s \equiv s(f) := |\{x:f(x) = 1\}|$ be the number of satisfying assignments. Then, finding s defines the counting class #P, while determining whether $s \geq 2^{n-1}$ (i.e., finding the first bit of s) defines the decision class PP. This class contains NP and BQP as well as QMA, the quantum version of NP.

First, we investigate the computational power of creating PEPS. More precisely, we consider the scenario of Fig. 1: We want to know which decision problems we can solve with one use of a PEPS oracle, i.e., a black box which creates the quantum state from its classical PEPS description, together with efficient classical preprocessing and quantum postprocessing.

We now use the PEPS-postselection duality to show that the power of creating PEPS equals PP. It has been shown
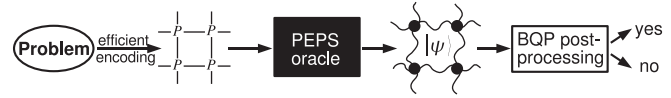

FIG. 1. The power of creating PEPS: The original decision problem is transformed into a PEPS description by a polynomial-time algorithm. The black box creates the corresponding quantum state, and an efficient quantum postprocessing returns the solution. Which kind of problems can we solve this way?

that $PostBQP$—the class of decision problems which can be solved by a postselected quantum computer—equals PP, $PostBQP = PP$ [8]. This readily implies that a PEPS oracle allows us to solve PP problems instantaneously by preparing the output of the postselected circuit as a PEPS and just measuring one output qubit in the computational basis. On the other hand, this is the best we can do with a single use of the PEPS oracle, since every PEPS can be generated efficiently by a postselected quantum computer. BQP postprocessing instead of a simple one-qubit measurement does not increase the computational power, since it commutes with the postselection and can thus be incorporated in the PEPS.

The fact that creating PEPS allows us to solve PP-complete problems strongly suggests the existence of PEPS which cannot be created efficiently by a quantum computer. Note however that the states which appear in the PP-hardness proof above are not of this type: once the corresponding counting problem is solved, they can be easily constructed. While it appears very unlikely that all PEPS can be constructed efficiently from some normal form (it would imply $QMA = QCMA$ and $BQP/qpoly = BQP/poly$ [12]), an example of such a state is still missing.

The classical complexity of PEPS.—Let us now investigate the complexity of classically simulating PEPS and its generalization, the contraction of tensor networks. For the case of PEPS, there are at least three possible definitions of the problem: compute the normalization of the PEPS (NORM), compute the unnormalized expectation value of some observable (UEV), and compute the normalized expectation value (NEV). Since they can be transformed easily into each other, we will use whichever is most appropriate.

We first show that contracting PEPS is #P-hard, i.e., that for any (polynomial) boolean function f, $s(f)$ can be found by simulating a PEPS. Therefore, we take a quantum circuit which creates $\sum_x |x\rangle_A |f(x)\rangle_B$ and encode it in a PEPS. Then, the normalized expectation value of σ_z of B allows us to compute $s(f)$.

To show that the simulation of PEPS is inside #P, we have to show that the normalization of the PEPS, or equivalently the success probability for the postselection, can be computed by counting the satisfying assignments of some boolean function. This can be done by adapting well-established quantum complexity techniques (see [8] and

references therein): First, approximate the postselected circuit using only Toffoli and Hadamard gates [13,14]. The probability p_x for a state $|x\rangle$ before postselection is obtained as a kind of path integral [15], by summing the amplitudes for all possible "computational paths" $\zeta = (\zeta_1, \ldots, \zeta_{T-1})$, where $|\zeta_t\rangle$ is the state at step t and T the length of the circuit:

$$p_x = \left| \sum_\zeta \alpha_{x,\zeta} \right|^2 = \sum_{\zeta, \zeta'} \alpha_{x,\zeta} \alpha^*_{x,\zeta'},$$

with $\alpha_{x,\zeta}$ a product over transition amplitudes $A_{\zeta_t \rightarrow \zeta_{t+1}}$ along the path ζ. The normalization of the PEPS is obtained as the sum over all states where the postselection succeeds, $\sum_{\bar{x}} p_{(0,\bar{x})}$. This can be rewritten as the sum over an efficiently computable function $f(\bar{x}, \zeta, \zeta') = \alpha_{(0,\bar{x}),\zeta} \alpha^*_{(0,\bar{x}),\zeta'}$ which takes values in $\{0, \pm 1\}$, as the circuit consisted only of Toffoli and Hadamard gates. Now, this sum can be computed by counting the satisfying assignments of the function $f_{\text{bool}}(\xi, z) := (f(\xi) \geq z)$, $z \in \{0, 1\}$, which shows that the simulation of PEPS is in #P. Together, we find that the classical simulation of PEPS is #P-complete under weakly parsimonious reductions (see [16]).

It is natural to ask whether this also shows that contracting general tensor networks is in #P. For a tensor network T, let us denote its contraction by $\mathcal{C}(T) \in \mathbb{C}$. Since the contraction of PEPS is a special case, it is clear that the problem is #P-hard. To place it within #P, observe first that $|\mathcal{C}(T)|^2 = \mathcal{C}(T \otimes T^*)$ can be found by attaching a physical system of dimension one to each site and computing the normalization of the resulting PEPS. To determine the phase of $\mathcal{C}(T)$, observe that $\mathcal{C}(T \oplus T') = \mathcal{C}(T) + \mathcal{C}(T')$. Thus, by setting $T' = T^*$, we get $|\text{Re}[\mathcal{C}(T)]|$, while the sign can be determined by adding another $T'' \equiv c > 0$. This proves that contracting tensor networks is #P-complete.

The obtained hardness results are stable under approximations. To see why, note that any counting problem can be reduced to any of our three primitives with only linear postprocessing, and thus approximating these primitives is as hard as approximating counting problems can be. For NEV, this again works by preparing $\sum |x\rangle_A |f(x)\rangle_B$ and computing the expectation value of B. For NORM and thus UEV, note that the output of any normal quantum circuit and thus $\sum |x\rangle_A |f(x)\rangle_B$ has a known norm when written as a PEPS, since the success probability of each cluster projector is known, and the probability of the two measurement outcomes in the cluster is unbiased [10]. Thus, the probability for $|1\rangle_B$ can be readily determined from the norm of the PEPS where we postselected on $|1\rangle_B$.

PEPS and ground states.—The interest in MPS and PEPS stems mainly from the fact that those states perform extremely well in approximating ground states. In the following, we use the PEPS–postselection duality, and a relation between postselection and cooling, to shed new light on the connection between PEPS and ground states. In

particular, we show that the unique ground state of a gapped Hamiltonian on a D-dimensional lattice can be approximated efficiently by the border of a PEPS with $D + 1$ dimensions.

Consider a Hamiltonian on N spins, $H = \sum_i H_i$, where each H_i acts on a finite number of spins, with a unique ground state and a polynomial energy gap $\Delta \geq 1/\text{poly}(N)$. Starting from a random state $|\chi\rangle$, the ground state can be efficiently approximated via $|\psi_0\rangle \approx \exp[-\beta H]|\chi\rangle$. The imaginary time evolution can in turn be approximated using the Trotter decomposition, which only requires operations $\exp[-\beta/NH_i]$ acting on finitely many spins. Since those operations are linear, they can be implemented using postselection, and we see that postselection can be used to cool into the ground state. By embedding the postselected cooling procedure in a PEPS, the ground state of any gapped N-particle Hamiltonian can be approximated up to ϵ by the boundary of a PEPS, where the extra dimension has depth $M \sim \text{poly}(N, 1/\epsilon)$. In case the H_i are local, the PEPS can be simplified considerably since any local linear operation can be implemented directly on the level of the PEPS without the need for ancilla qubits.

The power of creating ground states.—As we have seen, PEPS can encapsulate problems as hard as PP. However, these PEPS are quite artificial, while in practice one is often interested in PEPS in connection with ground states. Therefore, let us have a look at the computational power of a ground state oracle, i.e., a black box which creates the ground state from the Hamiltonian.

First, let us introduce the complexity class QMA [17]. Colloquially, QMA is the quantum version of NP; i.e., it contains all decision problems where for the "yes" instance, there exists an efficiently checkable quantum proof, while there is no proof for any "no" instance. In a seminal work, Kitaev [17,18] has shown that the problem of determining ground state energies of local Hamiltonians up to polynomial accuracy is QMA-complete. More precisely, in LOCAL HAMILTONIAN one is given an N-qubit local Hamiltonian $H = \sum H_i$ with the promise that the ground state energy $E_0 < a$ or $E_0 > b$, $b - a > 1/\text{poly}(N)$, and the task is to decide whether $E_0 < a$. Clearly, the ground state of H serves as a proof for a yes instance. In successive works, the class of Hamiltonians has been restricted down to two-particle nearest neighbor Hamiltonians on a 2D lattice of qubits [19].

Let us briefly reconsider our cooling protocol in the light of QMA. It is easy to see that the QMA proof need not necessarily be the ground state, as long as it is close enough in energy (depending on the verifier). Since our cooling protocol suppresses higher energy levels exponentially, the correspondence between postselection and cooling shows that a postselected quantum computer can be used to create proofs for QMA problems, or differently speaking, that any QMA proof can be efficiently expressed as a PEPS.

In the following, we give some observations which indicate that creating ground states of gapped Hamil-

tonians is easier than creating PEPS. First, note that a ground state oracle for arbitrary Hamiltonians is still as powerful as PP. To see why, take a PP problem and encapsulate it in a PEPS. By perturbing the P's randomly by a small amount, one obtains a PEPS which is the unique ground state of a local Hamiltonian, which can be derived efficiently from the P's [3,20]. This shows that an unrestricted ground state oracle enables us to solve PP problems. However, the gap Δ of the above Hamiltonian will be exponentially small: if not, one could add a small penalty, say $\Delta/100$, on the "answer" qubit and use that the original Hamiltonian has ground state energy $E_0 = 0$: Then, determining the value of that qubit could be solved in QMA, thus proving $QMA = PP$ which is considered unlikely [21].

Since ground states of general Hamiltonians are not easier to create than PEPS, let us now assume an oracle which only works for local Hamiltonians with a unique ground state, known ground state energy, and a polynomial spectral gap to the first excited state. (Alternatively, one could consider "proof oracles" for the LOCAL HAMILTONIAN problem.) It is easy to see that this restricted oracle, even with BQP postprocessing, is at most as powerful as QMA. The proof is the ground state, and the verifier is constructed as follows. Let V_1 be the verifier for the ground state, it accepts the ground state with p_{GS}, and any excited state with probability at most $p_{ES} = p_{GS} - \Delta$, $\Delta = 1/\text{poly}(N)$. Further, let V_2 be the postprocessing circuit which has a polynomial separation between the yes and the no answer if applied to the ground state, $p_{yes} = 1/2 + \delta$, $\delta = 1/\text{poly}(N)$. Take $Q = \frac{\Delta/2+1}{\Delta+1}$ and construct the complete verifier as follows: with probability Q, run V_1, and with $(1 - Q)$, run V_2. One can readily check that this gives a polynomial separation between the cases where the proof is the ground state and the postprocessing returns "yes," and the cases where either the proof is not the ground state or the postprocessing returns "no." The same strategy can be used to show that a PEPS oracle cannot be tested on all inputs unless $QMA = PP$: Otherwise, one could take a PP-hard PEPS and construct a verifier which either runs the testing routine or reads out the PP solution.

These observations show that imposing a constraint on the spectral gap of a Hamiltonian has direct implications on its computational complexity, and we think that the complexity properties of gapped Hamiltonians are worth being considered. On the one side, in the above scenario, it is not clear whether all QMA problems can be solved using this oracle; on the other side, it is not clear how important the knowledge of the ground state energy is—note however that we also had this knowledge in the PP-hard case. It is also an interesting question whether the problem LOCAL HAMILTONIAN remains QMA-complete when restricting to polynomially gapped Hamiltonians. If not, GAPPED LOCAL HAMILTONIAN should be a natural candidate for a physically motivated class of problems weaker than QMA.

[1] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).
[2] U. Schollwöck, Rev. Mod. Phys. **77**, 259 (2005).
[3] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, quant-ph/0608197.
[4] F. Verstraete and J. I. Cirac, cond-mat/0407066.
[5] C. Schön, E. Solano, F. Verstraete, J. I. Cirac, and M. M. Wolf, Phys. Rev. Lett. **95**, 110503 (2005).
[6] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).
[7] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac, Phys. Rev. Lett. **96**, 220601 (2006).
[8] S. Aaronson, Proc. R. Soc. A **461**, 3473 (2005).
[9] F. Verstraete and J. I. Cirac, Phys. Rev. A **70**, 060302 (2004).
[10] R. Raussendorf and H. J. Briegel, Phys. Rev. Lett. **86**, 5188 (2001); R. Raussendorf, D. E. Browne, and H. J. Briegel, Phys. Rev. A **68**, 022312 (2003).
[11] C. M. Papadimitriou, *Computational Complexity* (Addison-Wesley, Reading, MA, 1994).
[12] S. Aaronson, Theory of Computing **1**, 1 (2005).
[13] Y. Shi, Quantum Inf. Comput. **3**, 84 (2003).
[14] D. Aharonov, quant-ph/0301040.
[15] C. M. Dawson *et al.*, Quantum Inf. Comput. **5**, 102 (2005).
[16] NORM is UEV of $\mathbb{1}$. UEV of A is obtained from NORM via $\langle\psi|A|\psi\rangle = \langle\tilde{\psi}|\tilde{\psi}\rangle - \|A\| \langle\psi|\psi\rangle$, where $|\tilde{\psi}\rangle$ is derived from $|\psi\rangle$ by replacing the relevant P by $(A + \|A\| \mathbb{1})^{1/2}P$. Clearly, NEV can be computed using UEV and NORM. To compute NORM from NEV, take the PEPS with the post-selection omitted—this has a known norm [10], and compute the NEV of $\text{diag}(1, 0)$ on the qubit to be post-selected. All reductions are weakly parsimonious: problem A can be solved by *one* call to problem B, with efficient preprocessing of the input and postprocessing of the output. Note that two (or more) parallel #P-queries can be encoded in a single one, by considering $h(x, y, b)$, defined as $f(x)$ for $b = 0$ and $g(y)$ for $b = 1$, $x = 0$.
[17] D. Aharonov and T. Naveh, quant-ph/0210077.
[18] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (American Mathematical Society, Providence, Rhode Island, 2002).
[19] R. Oliveira and B. M. Terhal, quant-ph/0504050.
[20] D. Perez-Garcia, M. M. Wolf, F. Verstraete, and J. I. Cirac (to be published).
[21] M. Vyalyi, Electronic Colloquium on Computational Complexity **10**, Report TR03-021 (2003).