Quantum Simulators, Continuous-Time Automata, and Translationally Invariant Systems

K. G. H. Vollbrecht and J. I. Cirac
Max-Planck Institut für Quantenoptik, Hans-Kopfermann-Str. 1, Garching, D-85748, Germany

(Received 2 May 2007; published 10 January 2008)

The general problem of finding the ground state energy of lattice Hamiltonians is known to be very hard, even for a quantum computer. We show here that this is the case even for translationally invariant systems in 1D. We also show that a quantum computer can be built in a 1D chain with a fixed, translationally invariant Hamitonian consisting of nearest-neighbor interactions only. The result of the computation is obtained after a prescribed time with high probability.

DOI: 10.1103/PhysRevLett.100.010501 PACS numbers: 03.67.Lx, 05.50.+q

The difficulty of simulating the dynamics of quantum systems by classical means was recognized by Feynman [1] more than two decades ago. He proposed to use another system, a quantum simulator, to overcome this problem, introducing several visionary ideas about quantum computation. At the same time, very powerful classical techniques have been discovered, which allow us to tackle important problems in many-body physics. One of the most important questions in this context is to decide which kind of problems can (or cannot) be efficiently simulated or solved by quantum or classical simulators.

Systems in Nature typically possess certain symmetries (like homogeneity, or translational invariance). Thus, it is relevant to consider the above questions under the restrictions imposed by some of those symmetries. In this work we concentrate on 1D spin chains with a translationally invariant (TI) Hamiltonian and address the following questions: (i) Is it possible, given a quantum computer, to simulate those systems? (ii) Can such a physical system perform arbitrary quantum computations?

Regarding the first question, we show that if it would be possible to find the ground state energy of any TI Hamiltonian in 1D, then one could solve all QMA problems. Those are the ones belonging to the complexity class called ‘‘quantum Merlin Arthur’’ [2], which is the quantum analog of NP (nondeterministic polynomial time problem), and which is believed to be different from BQP (bounded error, quantum, polynomial time problem), the one corresponding to the problems a quantum computer can solve efficiently. In other words, if it were possible, then a quantum computer could solve efficiently (in poly time) very difficult problems, something which is believed not to be true by computer scientists. In order to derive this result, we build on the recent work [3]. There they show that all QMA problems could be solved efficiently if one would be able to find the ground state energy of (general, i.e., not restricted to be TI) 1D chains with nearest-neighbor interactions.

Regarding the second question, we show that indeed it is possible to perform arbitrary quantum computations if one has a certain fixed, TI Hamiltonian. One can encode the program in the initial state, and the result is obtained with very high probability after a certain time, with at most a polynomial overhead in the number of qubits with respect to a standard quantum computer. We stress that, as opposed to previous works [4], the Hamiltonian we build is fixed and independent of the computation to be performed. This may be relevant in certain implementations of quantum computation, like in atoms with optical lattices [5]. In that case, one can think of preparing a particular initial (product state) in a regime where the atoms are well separated [6], and then bring them closer (by changing the angle between the lasers that create the lattice) such that they experience a constant interaction (mediated by virtual transitions and cold collision). Note that our work does not contradict the nonexistence of a ‘‘general purpose programmable quantum computer’’ [7], since this result was about the stronger assumption of applying the full continuous group of unitary operators. Our scheme can be seen as a combination of a cellular automata [8] approach with a continuous-time quantum walk [1]. In fact, we will refer to our model as ‘‘continuous cellular automaton.’’ On the other hand, as opposed to the standard quantum cellular automata, ours has no classical analog. Furthermore, our model can be seen as a different paradigm to the standard gate based quantum-computer since: (i) it is continuous in time; (ii) it has a fixed and universal Hamiltonian, independent of the program to be run; (iii) the Hamiltonian is translationally invariant.

Ground state energy in TI systems.—We start out by showing that finding the ground state energy E_0 of any TI Hamiltonian in a 1D chain is very difficult (as we increase the number of sites), even for a quantum computer. In fact, we show that for each QMA-complete problem one can find a TI Hamiltonian, H_n, such that: $E_0 = 0$ if the answer to the problem is 'yes'; $E_0 > 1/[\mathrm{Poly}(n)]$, if the answer is no. Here Poly(n) denotes a polynomial in n. Our result heavily relies on the recent discovery [2,3] that QMA-complete problems can be encoded in the ground state energy of a Hamiltonian h_n describing a 1D chain of n system with $d = 12$ levels each and with nearest-neighbor interactions. This is, roughly speaking, done by encoding a complete quantum computation, including every time step and the time itself, into the ground state of the Hamiltonian. Each local Hamiltonian adds a penalty in energy, if one step of the computation is not carried out (encoded) correctly.

Our strategy is to build a TI H_n out of the non-TI h_n by increasing the dimension d by a factor of 2, i.d., by adding a qubit at each site. To prove this, let us assume a chain of length n with periodic boundary conditions. Furthermore, we assume that h_n [2,3] has a minimum eigenvalue $E_0 = \lambda$ (and corresponding eigenvector $|\Phi_0\rangle$) that defines the solution to a (QMA) problem Q in the above sense. We will now construct a TI Hamiltonian H_n having a minimum eigenvalue that still solves the same problem. H_n acts on a chain where every sites contains a spin (s) and, in addition, one extra qubit (q).

Let us define a Hamiltonian on this chain by $H' = \sum_{i=1}^{n} |1\rangle\langle 1|^{(q;i)} \otimes h_n^{(s;i,\ldots,i+n)}$, where the label ($q$; i) indicates that the operator is applied to the qubit-system at site i and (s; $i, \ldots, i+n$) that the operator is applied to all the spins at the sites ($i, \ldots, i+n$) in this order, where we identify the sites separated by n sites. That is, if the qubit at site i is in the state $|1\rangle$, then the Hamiltonian h_n is applied to all spins at ($i, \ldots, i+n$). Obviously, H' has an n-degenerated eigenvalue of λ with corresponding eigenvectors $|\psi_i\rangle = |100\ldots0\rangle^{(q;i,\ldots,i+n)} \otimes |\Phi_0\rangle^{(s;i,\ldots,i+n)}$, where the superindexes indicate the positions in the same sense as for the operators above. This eigenvalue would give the right answer to the problem Q, but it is not the minimum eigenvalue; H' has a zero subspace whenever all qubits are $|0\rangle$. To avoid this we define $P = 1 - \sum_k |1\rangle\langle 1|^{(q;k)} + \sum_{k''\neq k'} |1\rangle\langle 1|^{(q,k')} \otimes |1\rangle\langle 1|^{(q;k'')}$, where k, k', k'' run from site 1 to n. This operator penalizes any configuration where the number of qubits in sate $|1\rangle$ is not equal to one. Finally, we define H_n as $H_n = \frac{1}{n}H' + \frac{1}{n(n-1)}P$. H_n is TI, and contains up to three-body interactions only. In case h_n has a zero eigenvalue the same holds for H_n which is proved by the corresponding eigenvectors $|\psi_i\rangle$. Otherwise, since H' is lower bounded by λ and P by 1, H_n is lower bounded by $\min(\frac{\lambda}{n}, \frac{1}{n(n-1)})$. Note, that this is still decreasing only polynomially in n. Therefore, H_n gives the same answer to the problem Q as h_n did, q.e.d. Note that unlike the original Hamiltonian h_n this TI Hamiltonian is not a nearest-neighbor Hamiltonian any more. But to relax the condition of only nearest-neighbor interactions is required for a TI Hamiltonian, because otherwise its number of parameters do not scale with the length of the chain, which makes the system trivial in the sense of a complexity theoretical description.

Continuous cellular automaton.—Now, we introduce a programmable quantum computation scheme for an infinite chain of quantum systems of dimension $d = 30$. The program is encoded into the initial state, while the time evolution is fixed and given by a universal TI Hamiltonian with nearest-neighbor interaction only. This could be considered as a proof of principle for an new TI quantum computation scheme. We will start by introducing a simple quantum computing scheme. Later on, we will show how this quantum computer can be simulated by a continuous-time evolution. We consider a simple quantum computer with an n-qubit "hard disk" and a read and write head, that we call the pointer. The pointer can be moved such that we can address single qubits. Furthermore, the pointer has an internal quantum state, a qubit. To perform any quantum computation we write a program consisting of five different commands: (L) The pointer is moved one site to the left; (R) same but to the right; (S) the qubit at the position of the pointer and the internal state of the pointer are swapped; (G) a G Gate [8] on those two qubits is applied. The G gate allows for arbitrary quantum computations (within the standard gate model) if it can be applied between any two qubits. In this simple model this can be accomplished by loading qubits with the S command into the internal pointer state, which can then be moved to any other qubit. We now encode this quantum computer into a higher dimensional chain: Every site in the chain has three registers. A "qubit" register (q) having dimension 2. It acts like the normal qubit of the quantum computer. A "pointer" register (p) having dimension 3. Here we encode the pointer, where 0 indicates no pointer and 1 or 2 the presence of a pointer with an internal qubit state. Finally, a "program" register (c) having dimension 5: One for "e = empty" and the rest for commands $\{L, R, S, G\}$. The total dimension of one site is then 30. We choose n neighboring sites to be the "quantum computer", i.e., the qubit-registers of those sites correspond one to one to the qubits of the quantum-computer we want to simulate [see Fig. 1(a)]. The pointer registers are in state $|0\rangle$ everywhere, except for one site which contains $|1\rangle$. The program is written in the program register in an area to the left of the quantum computer, where the commands are arranged in the order they should be executed from right to left. The rest of the program registers are filled with $|e\rangle$.

We assume now a TI Hamiltonian with nearest-neighbor interactions, $H = \sum_i H_i$, where H_i denotes a two-site Hamiltonian acting only on the sites i and $i+1$ and is given by $H_i = \sum_{C\in\{L,R,S,G\}} |e\rangle\langle C|^{(c;i)} \otimes |C\rangle\langle e|^{(c;i+1)} \otimes U_C^{(pq;i,i+1)}$ + H.c. The first two operators only act on the two program registers at sites i, and $i+1$, respectively,

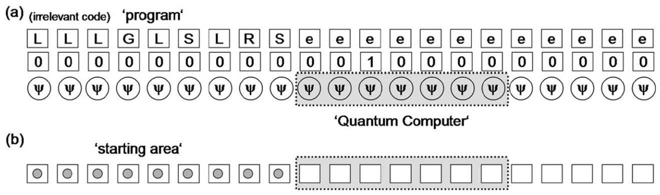

FIG. 1. (a) A 7-qubit quantum computer encoded into a 1D chain. The pointer is represented by the 1, the program is encoded to the left of the quantum computer. The sites are counted from the left to the right. (b) The system can be mapped onto another system, where every command correspond to an electron that is allowed to hop to nearby sites.

while U_C's are unitary operators acting both on the qubit and pointer register at those two sites. The unitaries are now defined as follows: The operations U_S and U_G are controlled unitaries which implement the two-qubit S and G gates between the qubit and pointer registers only if there is a pointer present at site i. In this case a swap-gate or G gate, is applied on the qubit register and the internal qubit state ($|1\rangle$, $|2\rangle$) of the pointer register. If no pointer is present ($|0\rangle$) nothing happens. U_L swaps the two pointer sites while U_R swaps the two-qubit sites. Let us assume now, that we apply one of those unitaries on every pair of sites $(i, i+1)$ starting from the left end of the chain going step by step to the right. U_S and U_G will do nothing until we reach the position where a pointer is present in the first of the two participating sites. In this case, the corresponding gates are applied. The U_L gate will swap the two pointer states at every position. As a result the pointer, marked by $|1\rangle$, is moved one step to the left. Finally, U_R will swap the two qubit registers. As a result the whole qubit-register chain is moved one step to the left. Relative to thequantum computer position, the pointer is moved one step to the right. Therefore, if such a sequence of unitaries is applied from left to right the corresponding set of commands is applied to the quantum computer. Note, that it is sufficient to start this sequence of unitaries at any position to the left of the quantum commuter and to stop it once they are to its right. Our claim is now that this basically is what the time evolution does.

To this aim, we consider the time evolution operator $e^{-iHt} = \sum_n \frac{(-iHt)^n}{n!}$. Applied to the initial state $|\text{start}\rangle$ we just end up in a linear combination of states of the form $H^n|\text{start}\rangle$. Since H is defined as a sum of terms of the form $h_i^C := |e, C\rangle\langle C, e|^{(c;i,i+1)} \otimes U_C^{(pq;i,i+1)}$ (and $h_i^{C\dagger}$), we can write the result of a time evolution as a linear combination of states of the form $h_n \cdots h_3 h_2 h_1|\text{start}\rangle$, where every $h_k \in \{h_i^C, h_i^{C\dagger}\}$. The effect of any such h_i^C on a state is the following: it either moves the command C from place i to $(i+1)$ and applies U_C to the sites $(i, i+1)$, or it maps the state to zero if no C is at i and no empty space $|e\rangle$ at $(i+1)$. In the same manner $h_i^{C\dagger}$ results in a state where the command C is moved from $(i+1)$ to i while $U_C^\dagger$ is applied. Therefore $h_n \cdots h_3 h_2 h_1|\text{start}\rangle$ is either zero, or a state where several of the commands have moved while applying the assigned unitaries (or their conjugates) all the way from their initial to their final position. Note that such a command can only move, if the a neighboring site is in the state $|e\rangle$, i.d., up to some $|e\rangle$'s in between the order of the commands in the program register never change. If a command moves to the right and afterwards to the left, we end up in the same state since U_C and $U_C^\dagger$ cancel each other. This implies that two states $h_n \cdots h_3 h_2 h_1|\text{start}\rangle$, $g_m \cdots g_3 g_2 g_1|\text{start}\rangle$ are equal, iff the configuration of commands in the program registers are equal. That is, a measurement of all the command registers leads exactly to a state of the form $h_n \cdots h_3 h_2 h_1|\text{start}\rangle$, where $h_n \cdots h_3 h_2 h_1$ can be chosen to be any combination that maps the initial commands to the measured combination.

In particular, suppose that we measure after some time evolution a configuration where all the commands that were initially to the left of the quantum computer [see Fig. 1(a)], are found to its right. In this case the whole program has been executed (in the right order) by the quantum computer and we can read out the result. If this is not the case, we can continue the time evolution and repeat until we found a positive result. To boost the probability of success, we can increase the program code by some irrelevant code [see Fig. 1(a)], e.g., by adding $|L\rangle's$ that do not affect the result of the computation if carried out after the real program. This irrelevant code will act as a kind of barrier, that prevents the real program code to move to the left and forces it to do the computation. In conclusion, the computation scheme has three steps: (i) preparing the initial state; (ii) applying a fixed time evolution; (iii) measuring the command register to verify a successful computation [otherwise repeating (ii)-(iii)].

In order to verify that this is an effective way of carrying out quantum computations, we have to guarantee that the probability of finding the whole program to the right of the quantum computer is finite after a finite (polynomially) time. To this end we have to calculate (a lower bound to) the probability of success. Since the states $h_n \cdots h_3 h_2 h_1|\text{start}\rangle$ are completely defined by only the configuration of the commands, we can map our system onto a chain of qubits, where $|0\rangle$ means 'empty' and $|1\rangle$ means "command." We do not have to specify the 'command' because the order of the commands stays unchanged under the evolution so that we can identify a command by its relative position. The Hamiltonian simplifies to $H = \sum_i H_i$ with $H_i = |0_i 1_{i+1}\rangle\langle 1_i 0_{i+1}| + \text{H.c.}$. We see that the efficiency of the computation does not depend on the specific program we run. Let us take this system as (spinless) electrons in a lattice [see Fig. 1(b)], where $|1\rangle$ stands for an electron and $|0\rangle$ for an empty site and H_i is a hopping term. Thus, we end up with noninteracting fermions in second quantization. Now we go back to first quantization, where we just have to consider a single electron Hamiltonian, $H^S = \sum_k |k+1\rangle\langle k| + \text{H.c.}$, where $|k\rangle$ denotes now the electron sitting at site k. The total Hamiltonian can then be written in first quantization as $H = \sum_i H^S_{\langle i\rangle}$, where $\langle i\rangle$ is now labeling the different electrons. The single electron problem with Hamiltonian H^S can be solved for periodic boundary conditions. Let us assume M sites with periodic boundary conditions, then the eigenvectors are given by $\psi_q = \frac{1}{\sqrt{M}} \sum_x e^{i(2\pi/M)xq}|x\rangle$ with corresponding eigenvalues $\varepsilon(q) = 2\cos(\frac{2\pi}{M} q)$. One electron sitting at site y is written in terms of the eigenvectors as $\phi_y = \frac{1}{M} \sum_q \psi_q e^{-i(2\pi/M)yq} = \frac{1}{M} \sum_{q,x} e^{i(2\pi/M)(x-y)q}|x\rangle$. After the time evolution, the state is changed into $\phi_{y,t} = \frac{1}{M} \sum_{q,x} e^{i(2\pi/M)(x-y)q + i\varepsilon(q)t}|x\rangle$. Since we are dealing with non−interacting fermions, we get the solution for N electrons, up to fermionic antisymmetrization, by the tensor product. We assume now N electrons sitting in the first N sites leading to the state $\psi_0 = S[|\phi_1, \phi_2, \phi_3, \ldots, \phi_N\rangle]$, where S denotes the fermionic antisymmetrization operator. After waiting a time t we end up in the state $\psi_t = S[|\phi_{1,t}, \phi_{2,t}, \phi_{3,t}, \ldots, \phi_{N,t}\rangle]$.

Now we compute the probability p_1 to find electron $\langle 1\rangle$ at time t still in one of the first N sites. The initial vectors ϕ_y are orthogonal and stay orthogonal under an unitary time evolution. The reduction of ψ_t to electron $\langle 1\rangle$ gives us $(1/N)\sum_y|\phi_{y,t}\rangle\langle\phi_{y,t}|$. The off-diagonal terms $|\phi_{y_1,t}\rangle\langle\phi_{y_2,t}|$ vanish due to the orthogonality of the permutated $\phi_{y,t}$ for the electrons $\langle 2\rangle$ to $\langle N\rangle$. The probability p_1 to find electron $\langle 1\rangle$ in one of the first N sides yields $p_1 = \frac{1}{N} \times \sum_{y=1}^{N}\sum_{x=1}^{N}|\langle\phi_{y,t}|x\rangle|^2$. In [9] it is shown that by choosing t to be proportional to N, e.g. $t = 5000N$, we can bound this probability to be smaller than a fixed constant, e.g., $p_1 < 0.3$. So we can guarantee to find electron $\langle 1\rangle$ after a polynomial time with probability $p > 0.7$ outside the starting area. Note that electron $\langle 1\rangle$ is not localized in site one at time zero, but is spread over the whole starting area due to the fermionic antisymmetrization. The same calculations holds for all the other electrons. The expected number of electrons inside the starting area is given by Np_1, whereas the ones that move outside this areas is given by $N(1 - p_1) = Np$.

For a successful computation a fixed number k of electrons have to leave their starting area. Let us assume that we find in every single shot q electrons with probability p_q, with $\sum_q qp_q = Np$, $\sum_q p_q = 1$. If the number of electrons is bigger than k the computation is successfully done. The success probability is $p_s = \sum_{q>k}p_q$ under the above constrains. To get an lower bound for p_s let us assume the worst case scenario and minimize p_s. The minimum is attained when we either get N electrons with probability $p_N = p_s$ or $(k-1)$ electrons with probability $p_{(k-1)} = 1 - p_s$ under the condition that $p_{(k-1)}(k-1) + p_sN = Np$. From this we can conclude that the success probability $p_s > \frac{1-k+Np}{1-k+N}$ which can be made arbitrarily close to p by choosing N to grow polynomially with k. Now let us apply this to our model. We have to distinguish between the electron leaving the starting area to the left and to the right, because only the ones going to the right will carry out the program. But, due to the reflection symmetry of the problem, we can assume all electrons moving in the right direction, what can be corrected by an irrelevant factor of 2 in the following discussion. We have a program of length l_p and a quantum computer of length l_q. Instead of searching for l_p electrons to the left of the quantum computer we can search for $l_p + l_q$ "electrons" leaving the starting area. The extra l_q electrons will be just part of the irrelevant code; that guarantees that the real program completely passed the quantum computer. Then we choose N to be, e.g., $(l_p + l_q)^2$. The above calculation then tells us that after a time of $5000(l_p + l_q)^2$ the computation is successful with a probability higher than $p = 0.7$. Therefore the evolution time grows only polynomially in the size of the quantum computer and the length of the program.

Conclusion.—We have derived two results regarding the use of a quantum computer in the presence of translational invariance, a typical property of Nature: (i) If one could simulate any translationally invariant 1D chain, then one could solve all QMA problems. (ii) It is possible to build continuous-time automata that are as powerful as quantum computers. The first result implies that, if as it is believed, BQP $\neq$ QMA, then not even a quantum computer could solve translationally invariant 1D problems, i.e., solving general TI systems is much harder than BQP or even NP. The second one implies that quantum computation is possible with a spin chain that has a fixed universal Hamiltonian. The computation itself requires only enough patience but no active further control. This last result should be considered as a proof of principle, since experimentally it will be very difficult—if not impossible—to implement such a Hamiltonian. However, it may be possible that with simpler Hamiltonians the same result is true, something which may have very interesting experimental implications.

We acknowledges support from EU projects (SCALA and CONQUEST), DFG-Forschergruppe No. 635, and the excellence clusters MAP and NIM. K. V. thanks Frank Verstraete for helpful discussions.

[1] R. Feynman, Int. J. Theor. Phys. **21**, 467 (1982); R. Feynman, Found. Phys. **16**, 507 (1986).
[2] A. Kitaev, A. H. Shen, and M. N. Vialyi, *Classical and Quantum Computation* (Americal Mathematical Society, Providence, 2002); J. Kempe, A. Kitaev, and O. Regev, SIAM J. Comput. **35**, 1070 (2006); R. Oliveira and B. Terhal, arXiv:quant-ph/0504050.
[3] D. Aharonov, D. Gottesman, and J. Kempe, arXiv:0705.4077.
[4] K. G. H. Vollbrecht, E. Solano, and J. I. Cirac, Phys. Rev. Lett. **93**, 220502 (2004); K. G. H. Vollbrecht and J. I. Cirac, Phys. Rev. A **73**, 012324 (2006).
[5] D. Jaksch, H.-J. Briegel, J. I. Cirac, C. W. Gardiner, and P. Zoller, Phys. Rev. Lett. **82**, 1975 (1999).
[6] K. D. Nelson, X. Li, and D. S. Weiss, Nature Phys. **3**, 556 (2007).
[7] M. Nielsen and I. Chuang, Phys. Rev. Lett. **79**, 321 (1997).
[8] D. J. Shepherd and T. Franz, R. F. Werner, Phys. Rev. Lett. **97**, 020502 (2006); Diego de Falco and Dario Tamascelli, J. Phys. A **39**, 5873 (2006).
[9] K. Vollbrecht and J. I. Cirac, arXiv:0704.3432.