Quantum Simulation a New Perspective

Quantum Computing Group@UW
Department of Computer Science & Engineering,
University of Washington, Seattle, WA USA 98195 and
Department of Physics, University of Washington, Seattle, WA USA 98195

Happy 150^{th} birthday

I. QUANTUM ON QUANTUM SIMULATION

Computer Science is the study of algorithms [1], yet computer engineering might be another completely different story. When we submit a computation job, say calculating the addition of one apple plus two apples or 1+2, to our computer, there is no such thing like apple existing inside a computer. Instead we actually use some mathematically meaningful but physically unrelated quantities like electrical pulses with different peak values to *represent* the values we want to calculate. Neither is there a *plus* sign, which physically is an adder composed of AND, OR and NO gates.

It was Feynman [2] who first realized that we could actually take the cue from classical world to use a controllable quantum device, probably called quantum computer or quantum simulator, to simulate the time evolution or physical process of another quantum system. Unfortunately the ubiquity of decoherence processes makes quantum computers more sensitive to noise and errors than classical computers. Traditionally there are two proposed methods to accomplish quantum computation fault tolerantly. One is using quantum error correction codes and using quantum gates to repeatedly correct errors due to noise and errors introduced by these quantum gates itself. The other one is using physical systems which is intrinsically immune to errors or whose quantum processes are automatically fault tolerant, such as the 4-dimension toric code Hamiltonian.

At the turn of last century, Llyod *et al.* [3] proposed an intermediate method of fault tolerant quantum computation, that is to use quantum gates to simulate fault-tolerant quantum systems which are immune to local errors. They introduced a computable mapping M from the system Hilbert space into the simulator Hilbert space such that $M^\dagger M = \mathbf{1}$. Hence if the system Hamiltonian is $H = \sum_{i\in\mathcal{N}} H_i$ where $\mathcal{N}$ is the set of local neighbors, then the overall simulator Hamiltonian would be $H^s = MHM^\dagger = \sum_{i\in\mathcal{N}} H_i^s$, in which $H_i^s = MH_iM^\dagger$. Correspondingly the state ρ of system would be mapped into the state $\rho^s = M\rho M^\dagger$ of the simulator... The good is that the simulation is shown to preserve the fault tolerance of the simulated system, because errors are automatically eliminated by natural error-correcting feature of the system simulated. The bad is that they didn't consider the errors introduced by the quantum gates themselves.

Here we think about quantum simulator(or quantum computer) from a new perspective inspired its classical sibling. Similar to the classical world, we have a quantum system A called *original* system, whose process we want to simulate. We also have another quantum system B, called *simulated* system, which will be used to simulate the process of system A. We have to use different physical quantities or parameters to label states for different systems, say A and B. For example, the state of system A might be a space part tensor product with its spin part but the state of system B might additionally need its orbital momentum as another physical quantity to describe its state. They are not simply related by unitary mapping as Lloyd *et al.* suggested. So how are we going to represent the mapping between the original and simulated quantum systems? What are the criteria such that system B could be used to simulate the original system A?

Without lose of generality, let's consider our problem in the Schordinger picture, where operators are invariant along with time and is basis independent. Analogue to the classical case, if there is a bijective relation between states of system A and states of B, we say they are *computationally equivalent*, as shown in Figure 1. Since every state in A could be represented in some basis $\{\rho_i^A\}$, we want a mapping f such that $f : \rho_i^A \to \rho_i^B$ where $\{\rho_i^B\}$ are basis of system B. In other words, mapping f is a bijective function from the Hamiltonian space of the *original* system to the Hamiltonian space of the *simulated* system. Hence if the original system has a state $\rho^A = \sum_i c_i\rho_i^A$, then there is one and only one corresponding state $\sum_i c_i f(\rho^A) = \sum_i c_i\rho_i^B = \rho^B$ in the simulated system. According to our assumption commutation or anti-commutation relations are conserved under mapping since operators are invariant. Then we claim that a stabilizer code in the original system is still a stabilizer code in the simulated system. To see this, we start from some stabilizer code, generated by $\{s_i\}$, in the original Hamiltonian. Hence for some state $|\psi\rangle$, in system A we have $s_i|\psi\rangle = |\psi\rangle$ for all i. We could expand the state to be $|\psi\rangle = \sum_j c_j|\alpha_j\rangle$. Then for all i, $s_i|\psi\rangle = \sum_j c_j s_i|\alpha_j\rangle = \sum_j c_j|\alpha_j\rangle$. So we have $s_i|\alpha_j\rangle = |\alpha_j\rangle$ as expected. Hence the original stabilizer code is also a valid stabilizer code in the mapped system B. Therefore if the original system is fault tolerant, the simulated system is automatically fault tolerant, guaranteed by our construction.

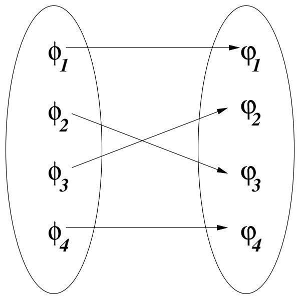

FIG. 1. Bijective relation between states of original system A and simulated system B

II. TO DO

- What is the general form for the states like master equation? Or how to express the master equation in terms of mapped quantities?
- Is our requirement still too restrictive? What if we just need the aforementioned mapping works at some measurement point?
- How do we find the mapping? Is there a general purpose quantum computer that could simulate any quantum processes?
- Mapping seems to have some weird property that is $f(\sum_i A_i) = \sum_i f(A_i)$.

Appendix A

If it is true that $H' = M^\dagger H M$, then M is diagonal. $f(H) = H'$ bijective.

Dave used Clifford gate U such that $H' = UHU^\dagger$.

The commutation relation is preserved under mapping. If operators A and B commute, then it is easy to prove that $A' = MAM^\dagger$ and $B' = MBM^\dagger$ also commute. If they anti-commute, their mapped operators also anti-commute.

[1] I. Pohl and A. C. Shaw, *The nature of computation : an introduction to computer science* (Computer Science Press, Potomac, MD, 1981).

[2] R. Feynman, Intl. J. Theor. Phys. **21**, 467 (1982).

[3] S. Lloyd, B. Rahn, and C. Ahn, arxiv.org/abs/quant-ph/9912040 (1999).