CMSC 657 - Introduction to Quantum Information Processing **Due: Dec 14**

Final Report: Quantum Simulation of Spin Models

Instructor: Xiaodi Wu Student: Jue Xu, Shucheng Zhu

Simulation by quantum computers [1] is a promising method to investigate quantum systems which are intractable to classical computers. However, it is not explicit how quantum computers outperform classical ones in practice. In this project, we will answer this question as follows: in the first and the second section, we mainly discuss the motivation of quantum simulation and different models of quantum simulation; in the third and fourth section, we survey the recent research on the simulation of Hamiltonian dynamics; on the other hand, we illustrate how to simulate the equilibrium properties of quantum systems by implementing quantum metropolis algorithm. For simplicity, We will focus on the spin models and will not touch the detail of the physical realization.

1 Why Quantum Simulation

Firstly, *quantum simulation problems* are to find the state of a quantum system described by the wave function $|\psi\rangle$ at some time t and computing the value of some physical quantity of interest, e.g. energy levels of an atom, the critical point of phase transition. Focusing for simplicity on time-independent Hamiltonians (denoted $\hat{H}$), the solution of the Schrödinger equation:

$$i\hbar\frac{d}{dt}|\Psi(\mathbf{r},t)\rangle = \hat{H}|\Psi(\mathbf{r},t)\rangle \tag{1}$$

is given by $|\Psi(t)\rangle = \exp\Big(-i\hbar\hat{H}t\Big)|\Psi(0)\rangle$. We can write the operator of total energy, Hamiltonian $\hat{H}$ explicitly,

$$\hat{H}|\Psi(\mathbf{r})\rangle = \big[-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r})\big]|\Psi(\mathbf{r})\rangle = E|\Psi(\mathbf{r})\rangle \tag{2}$$

The first term in the Hamiltonian is the kinetic energy and the second term is potential energy. E on the right hand side is the eigenvalue of the Hamiltonian $\hat{H}$, namely, the value of system's total energy. In many cases, we are interested in the energy levels of a system, i.e., all eigenvalues of the Hamiltonian. For example, the energy levels of atoms are of great importance because they carry almost all important information of elements and materials.

Example 1 (Two-level system)**.** A *two-level system* is a quantum system that can exist in any quantum superposition of two independent (physically distinguishable) quantum states.

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle, \quad c_0, c_1 \in \mathbb{C} \tag{3}$$

It may be the simplest nontrivial quantum system. A well-known example is the spin of a spin-1/2 particle such as an electron, whose spin can have values $\hbar/2$ or $-\hbar/2$. Its energy in a magnetic field $\mathbf{B}$ (assume the direction of $\mathbf{B}$ is parallel to the direction of spin) is given by

$$\hat{H}_{spin} = -\mathbf{B}\cdot\sigma_z = \begin{pmatrix} -B & 0 \\ 0 & B \end{pmatrix}. \tag{4}$$

Then, its time evolution is easily to get

$$|\psi(t)\rangle = e^{-i\hbar \hat{H}_{spin}t}|\psi(0)\rangle = e^{i\hbar Bt}c_0|0\rangle + c_1 e^{-i\hbar Bt}|1\rangle. \tag{5}$$

Note that any two-level systems are inherent candidates of qubits that serve as registers in quantum computer.

Unluckily, two-level system is one of the very limited number of quantum systems that can solved analytically, such as one particle in a harmonic potential and Hydrogen (an electron interacting with the coulomb potential of a proton) [2]. These systems are solvable because their potential have ideal forms and only two-body interaction is considered. Many approximation methods, such as perturbation theory, were proposed to solve more specific Hamiltonians. For instance, when we include the spin-orbit coupling, relativistic correction and spin-spin coupling in the Hydrogen's total Hamiltonian instead of only considering the coulomb potential, the energy levels of Hydrogen cannot be solved analytically. The approximation methods are effective to tackle this kind of problem, but more serious problem occurs when we deal with the many body systems. In many body systems, a large number of particles interact with each other so that the Hamiltonian is too complicated to be solved analytically or numerically with brute force. However, many body systems are of great practical interests because we live in the macroscopic world. For instance, we are curious about the critical temperature of a bulk of superconductor instead of how single electron moves in it. This requires that we deal with the systems at thermodynamic limit, namely, the number of particles in a system is almost infinite ($\sim 10^{23}$). It is impossible to cope with these systems by solving eq. (2) and eq. (1). This is the reason why we need statistical mechanics. With statistical mechanics, we do statistics over all possible configurations of many body systems instead of tracking all particles' movement to obtain some useful information of the whole system. The computational problem in statistical mechanics becomes the calculation of partition function Z, with which we are able to calculate all thermodynamic quantities

$$P(c) = \frac{e^{-\beta E(c)}}{Z}, \quad Z = \sum_{c\in\Omega} e^{-\beta E(c)}, \quad \langle A\rangle = \sum_{c\in\Omega} A(c)P(c) \tag{6}$$

$P(c)$ is the probability of the system in certain configuration c. Z is the partition function, the sum of the weight $e^{-\beta E(c)}$ over configuration space Ω. This weight is also called Boltzmann factor. β is a parameter related with temperature $\beta = 1/k_B T$. $\langle A\rangle$ is the expectation value of certain observable. Note that the above defintion of partition function is for classical discrete system. The quantum version is

$$Z = \mathrm{tr}\Big(e^{-\beta\hat{H}}\Big), \quad \langle A\rangle = \frac{1}{Z}\,\mathrm{tr}\Big(Ae^{-\beta\hat{H}}\Big). \tag{7}$$

the second equation is the thermal average of an observable A. No matter quantum or classical, the size of configuration space increases exponentially with the size of the system.

Example 2 (Ising model)**.** The simplest model of many body system may be one-dimensional *Ising model*: the spins are arranged in a lattice, allowing each spin to interact with its neighbors. A *spin configuration* σ is an assignment of spin value to each lattice site. The classical Hamiltonian of such a spin configuration is

$$H_n(\sigma) = -\sum_{\langle i,j\rangle} J_{ij}\sigma_i\sigma_j - B\sum_i \sigma_i, \quad \sigma \in \{1, -1\} \tag{8}$$

The lower index n of Hamiltonian is to denote it is a n-body Hamiltonian. The first term in it is the interaction between adjacent spins and J_{ij} can be considered as a constant. The second term is the interaction between external magnetic field $\mathbf{B}$ and each spin. It should be noted that the interaction between spins is called ferromagnetic if $J_{ij} > 0$; it is called antiferromagnetic if $J_{ij} < 0$. Though they only differ by a sign, it makes big difference in computational complexity when we do simulation. The antiferromagnetic one is also called frustrated spins, which favoring rather simple, but different structures lead to qubit complex structures. The quantum version of Ising model is

$$\hat{H}_n(\sigma) = -\sum_{\langle i,j\rangle} J_{ij}\sigma_i^x\sigma_j^x - B\sum_i \sigma_i^x \tag{9}$$

where σ_i^x are Pauli matrices. Navielly, representing the configuration space Ω of n spin Ising model requires 2^n bits classically.

Although the partition function of one-dimensional and two-dimensional Ising models have exact solutions, higher dimensional Ising models or more complicated spin models need more advanced techniques to tackle with, such as mean-field theory and quantum field theory. With the development of computer science, an efficient and generic way to calculate partition function is Monte Carlo algorithm with Metropolis method[3], which samples the important configuration with high probability. Its main advantage is that it allows phase space integrals of partition function to be evaluated in a time that scales only polynomially with the size of system. It means that the thermal average of an observable is approximated by the sample mean within a statistical error in polynomial time.

$$\langle A\rangle \approx \bar{A} = \frac{1}{M}\sum_{i=1}^{M} A(c_i) \tag{10}$$

The same technique can be applied to reduce the exponential scaling of the problem in quantum case, but only after mapping the quantum model to a classical one. One approach to this mapping is a Taylor expansion:

$$\begin{aligned} Z = \operatorname{tr}\Big(e^{-\beta\hat{H}}\Big) &= \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \operatorname{tr}\hat{H}^n \\ &= \sum_{n=0}^{\infty}\sum_{i_1,\ldots,i_n} \frac{(-\beta)^n}{n!} \langle i_1|H|i_2\rangle\langle i_2|H|i_3\rangle\ldots\langle i_n|H|i_1\rangle \\ &\equiv \sum_{n=0}^{\infty}\sum_{i_1,\ldots,i_n} p(i_1,\cdots,i_n) \equiv \sum_c p(c) \end{aligned} \tag{11}$$

The weight $p(c)$ here is the corresponding product of matrix elements of $\hat{H}$ and the term $(-\beta)^n/n!$ instead of normal Boltzman factor $e^{-\beta E(c)}$. If all the weights $p(c)$ are positive, standard Monte

Carlo methods can be applied, as it is the non-frustrated quantum magnets and bosonic systems. The problem comes with fermonic systems or frustrated systems. Negative weights $p(c) < 0$ arise from the Pauli exclusion principle. The standard way of dealing with the negative weights of the fermionic system is to sample with respect to the bosonic system by using the absolute values of the weights $|p(c)|$ and to assign the sign to the quantity being sampled. While this allows Monte Carlo simulations to be performed, the errors increase exponentially with the system size and the inverse temperature β. It is called *sign problem*. It has been proved that sign problem is **NP**-hard[4]. Although this problem is intractable to classical computer, a quantum algorithm allows for direct sampling from the eigenstates of the Hamiltonian, overcoming sign problem[5]. This is a good example illustrating the supremacy of quantum simulation.

2 Quantum Simulation Models

There are mainly three types of quantum simulation: (1) digital quantum simulation; (2) analog quantum simulation; (3) quantum-information-inspired algorithms for the classical simulation of quantum systems. In this project, we will focus on digital quantum simulation that is more widely analyzed.

We denote the state of the simulated system by $|\varphi\rangle$. The system evolves from the initial state $|\varphi(0)\rangle$ to $|\varphi(t)\rangle$ via the unitary transformation $U = \exp\{-i\hbar H_{sys}t\}$, where H_{sys} is the Hamiltonian of the system. The quantum simulator is a controllable system: the initial state $|\psi(0)\rangle$ can be prepared, the desired unitary evolution $U' = \exp\{-i\hbar H_{sim}t\}$ with H_{siim} being the controllable Hamiltonian of the simulator can be engineered, and the final state $|\psi(t)\rangle$ can be measured. If a mapping between the system and the simulator exists, then the system can be simulated.[6]

2.1 Digital Quantum Simulation (DQS)

Most quantum algorithms we learned in the textbook [7] are based on circuit model. The wave function $|\varphi\rangle$ has to be encoded using the computational basis, i.e., as a superposition of binary bit strings. The simple example is the simulation of spin-1/2 particles. The three-spin state $|\varphi\rangle = |\uparrow\downarrow\downarrow\rangle$ is represented in the simulator by $|\psi\rangle = |100\rangle$. To obtain $|\psi(t)\rangle = \exp\{-i\hbar Ht\}|\psi(0)\rangle$, the complicated many qubit-unitary transformation U is implemented through the application of a sequence of universal gates, such as $\{T, CNOT, H\}$. Although any unitary operation can be written in terms of universal quantum gates, it should be noted that not all unitary operation can be simulated efficiently. The advantage of digital quantum simulation is that rich experience and results in classical digital computation can be used for reference.

2.2 Analog Quantum Simulation (AQS)

Another method is analog quantum simulation, in which one quantum system emulates the other one. The Hamiltonian of the system to be simulated H_{sys} is directly mapped onto the Hamiltonian of the simulator H_{sim}, which can be controlled at least to some extent. The choice of the mapping depends on what needs to be simulated and on the capabilities of the simulator. In AQS, one is usually emulating an effective many-body model of the simulated system. It should be noted that the simulator may only partly reproduce the dynamics of the system. A controllable toy model of the system is used to reproduce the certain property of interest. An important advantage of AQS is that it could be useful even in the presence of errors, up to a certain tolerance level. For example, one is sometimes interested in knowing whether a certain set of physical conditions leads to a given quantum phase transition. Even without having the full quantitative details, a qualitative answer can be quite valuable in this context.

3 Hamiltonian Simulation with Product Formula Algorithm

In this section we give a brief review on a spcific digital simulation algorithm, namely the product formula algorithm. Given a quantum algorithm for Hamiltonian simulation, people investigate its performance algorithmically [8, 9] and empirically [10]. Algorithmically, query complexity measures the frequency a blackbox oracle being invoked through execution.In particular, the blackbox is an oracle that spits out the entries of the simulated Hamiltonian. Empirically, software packages were used to synthesize circuit with a set of universal quantum gates. The number of quantum gates used by the synthesized circuits indicates the complexity of the algorithm. In this section we delve into a most basic quantum algorithm, namely the product formula algorithm.
The generic Hamiltonian for spin models is $H = \sum_i \alpha_i H_i$, hence the evolution operator is $e^{-it\sum_i \alpha_i H_i}$. Instead of simulating the whole operator for time t, it is natural to break the operator into product of operators, and simulate each one piece at a time. If we know the total number of pieces n and number of oracle calls m to get matrix entries for each e^{-itH_i}, then the query complexity would exactly be $O(mn)$. However, by Baker-Campbell-Hausdorff formula it does not hold true that $e^{-it\sum_i \alpha_i H_i} = \prod_i e^{-itH_i}$, unless $[H_i, H_j] = 0, \forall i, j$. The remedy for it is to apply Suzuki's formula [11], such that the matrix norm between e^{-iHt} and higher order product of exponentials with infinitesimal time increment is sufficiently small. [10] rephrased [8]'s result in a cleaner way:

$$\left\| e^{-it\sum_i^L \alpha_j H_j} - [S_{2k}(-\frac{it}{r})]^r \right\| = O(\frac{(L\Gamma t)^{2k+1}}{r^{2k}}) \,, \tag{12}$$

where Γ is a constant, S_{2k} is the recursive definition of Suzuki's formula for product of exponentials and k is an integer that specifies the order of Suzuki's formula. The asymptotic behavior tells that good approximation can be achieved either by applying higher order Suzuki's formula or making infinitesimal time evolution for each piece of exponential operator. So for a fixed matrix norm error, there is a trade off between k and r to result in best query complexity.

3.1 Algorithm query complexity analysis

For each piece of exponential operator e^{-iH_jt}, with the assumption that H_j is sparse, [8] showed that each query to H_j can be simulated by making $\log N$ queries to H. To investigate the total number N_{exp} of exponential operators for approximating e^{-iHt} within matrix norm ϵ, [8] stated that

$$N_{exp} \leq 5^{2k} L^2 \|Ht\| (\frac{L\|Ht\|}{\epsilon})^{1/2k} \,, \text{with } \epsilon \leq 1 \leq 2L5^{k-1}\|Ht\| \,. \tag{13}$$

It is interesting to observe that for large k, the number of exponential operators is linear in terms of time t. Hence the total query complexity is $O(N_{exp} \log N)$, which is also linear in time. Later work by [12] has been focusing on optimizing query complexity with respect to the lattice size L, they brought down the $L^2 \log N$ factor in N_{exp} to $L(\sqrt{L} + \log(N))$ using star decomposition technique.

3.2 Circuit size analysis

[10] synthesized digital circuit with a set of universal quantum gates, and counted the optimized number of gates in the end as algorithm's gate complexity. Firstly a theoretical gate estimation was made. In contrast to the calculation of algorithm query complexity, where the total query complexity is the multiple between number of exponentials and number of queries to H for each exponential. The gate complexity is calculated by the multiple between the number of slices on evolution time and the number of gates used to simulate all exponentials under infinitesimal time increment. The hard part is to make the number of slices as small as possible while maintaining matrix norm under target error.

Up to $k = 8$th order of Suzuki's formula was applied to approximate e^{-iHt}. For given target matrix norm error ϵ, the number of time slice r for each exponential operator could be determined in terms of lattice size L, targeted error ϵ, simulation time t and order k. However, the requirement on r can be relaxed if there are terms in H that are weakly correlated, i.e. $[H_i, H_j] \sim 0$. This comes straight from the Baker-Campbell-Hausdorff formula, where the first order approximation becomes exact if all Hamiltonian components commute. With that in mind, [10] exploited commutation relations in Heisenberg Hamiltonian which can be written as sum of Pauli matrices. They managed to bring down the matrix norm between e^{-iHt} and product of exponentials, and concluded that with simulation error $\epsilon = O(1)$ and simulation time $t = L$, the number of slices $r = O(L^{2+2/(2k+1)})$. Hence in total the theoretical gate complexity is $O(L^{3+2/(2k+1)})$, because each slice $S_{2k}(-\frac{it}{r})$ takes lattice size $O(L)$ number of gates to simulate.

To empirically estimate the gate complexity, [10] first fix the order k, and target error $\epsilon = 0.001$. They numerically determined number of slices r for lattice size varying from 5 to 12, and did an exponential fit between r and lattice size L, i.e. $r = \alpha L^{\beta}$. Therefore, the empirical complexity is $O(L^{\beta+1})$. It turns out that the empirical estimation yields a better upper bound, hence there can be improvement on the theoretical analysis.

4 Lattice Hamiltonian simulation

For real physical spin systems, generally the problems are categorized by the strength of interactions between atoms sitting on lattice sites.

One is system of short range interactions where only the nearest neighbor interactions are considered. For short range interactions, the most famous classical simulation technique is Density Functional Theory. There are large number of software packages utilizing this theory to simulate the properties of spin systems with high accuracy. In the following subsection, we will present a quantum algorithm [13] that simulates short range spin system with gate complexity $O(nT\, polylog(nT/\epsilon))$, and this is an optimal solution as a matching lower bound was found.

Spin systems become more complicated to analyze when long range interactions are taken into account, since there are very few terms that can be dropped or simplified in the Hamiltonian. In this case the number of terms in Hamiltonian is proportional to exponential of number of sites, hence the Hamiltonian is intractable by any classical calculations. The most successful method to tackle strong coupling is Dynamic Mean Field Theory that maps lattice model to a impurity model which is solvable by classical means.

4.1 Short range interaction

[13] has the most optimal gate approximation for simulating Hamiltonian with short range interactions. The result states that evolution of Hamiltonian for time T on n qubits (lattice sites) up to error ϵ uses $O(nT\,polylog(nT/\epsilon))$ quantum gates. The overall idea is quite similar to the product formula method, where e^{-iHt} can be broken up into product of pieces of unitary operators e^{-iH_jt}. The difference is the method used to provide the error guarantee. Product formula utilizes recursive definition of Suzuki's formula, or fundamentally the Baker-Campbell-Hausdorff commutation relation, to bound the error as we discussed earlier. [13] applies the implication of another commutation relation, Lieb-Robinson bound, to bound the error of product of pieces of unitaries. Another key difference between them is the product of unitaries by [13] contains both forward and backward evolution, i.e. e^{-iH_jt} and e^{iH_jt}, whereas product formula method contains only forward evolution unitaries. In the following paragraphs we briefly and informally describe the implication of Lieb-Robinson bound to error guarantee.

Lieb-Robinson bound compares the matrix norm of difference of a local Hermitian operator evolved by a full Hamiltonian and a spatially restricted Hamiltonian. Let U_t^H be e^{-iHt}, it states as

$$\left\| (U_t^H)^\dagger O_X U_t^H - (U_t^{H_\Omega})^\dagger O_X U_t^{H_\Omega} \right\| \leq |X| \left\| O_X \right\| e^{-\mu l} \tag{14}$$

where $l = dist(X, \Lambda \setminus \Omega)$. Analogues to Baker-Campbell-Hausdorff relation implying Suzuki's formula, the above relation implies a more useful result that can be applied to product decomposition:

$$\left\| U_t^{H_{A\cup B}} (U_t^{H_B})^\dagger U_t^{H_{B\cup C}} - U_t^{H_{A\cup B\cup C}} \right\| \leq O(e^{vt-\mu\, dist(A,C)}) \sum_X \left\| h_X \right\| \tag{15}$$

where $X \subseteq A \cup B \cup C$ and $X \nsubseteq A \cup B$ and $X \nsubseteq C$. With the above formula to bound the error, any e^{-iHt} can be brought into product of three pieces. The mean result [13] iteratively applies such formula to break e^{-iHt} into $O(n\log(nT/\epsilon))$ pieces for evolution time $O(1)$, each piece is spatially supported by size $O(log(nT/\epsilon))$.

4.2 Long range interaction

[14] showed a new Lieb-Robinson bound and used it for unitary decomposition. However, they argue that the gate complexity from their unitary decomposition is still not tight, indicated by the result from numerical estimation. Therefore the gate complexity of long range interaction system still remains open.

A brief idea for the unitary decomposition is as follow. Suppose A, B and C are neighboring regions without any intersection. An evolution of operator O_X is $U_{ABC}^\dagger O_X U_{ABC}$. This can be approximated by $U_{AB}^\dagger U_B U_{BC}^\dagger O_X U_{BC} U_B^\dagger U_{AB}$. If we assume that X has no overlap with B and C, then O_X commutes with U_B, U_{BC}, hence $U_{ABC}^\dagger O_X U_{ABC} \sim U_{AB}^\dagger O_X U_{AB}$. It is intuitively true that the error of such approximate comes from two sources. First the interaction between A and C are ignored, therefore to estimate this type error it is sufficient to sum over all interactions on sites A and C. Second error comes from the fact that unitaries of subsystems on AB, B and BC may not commute.

With the above error bounded unitary decomposition, [14] obtained a Lieb-Robinson bound for long range interaction in one dimension:

$$\left\|[O_X(T), O_Y]\right\| = O(\frac{T^{\alpha-1}}{R^{\alpha-2}}) \tag{16}$$

Therefore a upper bound of gate complexity of such system can be inferred. However, a technique for finding a matching lower bound remains an open problem.

5 Quantum Simulation of Equilibrium Properties

So far, we have discussed how to simulate the evolution of different Hamiltonians and their complexities. However, the equilibrium properties of physical systems are also of great importance so that it is expected to speedup its calculation with quantum computers.

5.1 Measuring an operator

It is possible to view a Hermitian operator, not as the generator of dynamics, but as a quantity to be measured. For example, we might want to measure the energy of a system, i.e., the eigenvalues of the Hamiltonian. It turns out that any Hermitian operator that can be efficiently simulated (viewing it as the Hamiltonian of a quantum system) can also be efficiently measured using a formulation of the quantum measurement process given by von Neumann. We can find that von Neumann's procedure is exactly the same as quantum phase estimation.

In von Neumann's description of the measurement process, a measurement is performed by coupling the system of interest to an ancillary system, which we call the pointer. Suppose that the pointer is a one-dimensional free particle and that the system-pointer interaction Hamiltonian is $H \otimes p$, where p is the momentum of the particle. Furthermore, suppose that the mass of the particle is sufficiently large that we can neglect the kinectic term. Then the resulting evolution is

$$e^{-itH\otimes p} = \sum_a |E_a\rangle\langle E_a| \otimes e^{-itE_a p}, \tag{17}$$

where $|E_a\rangle$ are the eigenstates of H with eigenvalues E_a. Suppose we prepare the pointer in the state $|x = 0\rangle$, a narrow wave packet centered at $x = 0$. Since the momentum operator generates translations in position, the above evolution perform the transform

$$|E_a\rangle \otimes |x = 0\rangle \to |E_a\rangle \otimes |x = tE_a\rangle. \tag{18}$$

Now, the energy E_a can be measured by measuring the position of the pointer.

Note that the discretized von Neumann measurement procedure is equivalent to quantum phase estimation algorithm. Recall that in the phase estimation problem, we are given an eigenvector $|\psi\rangle$ of a unitary operator U and asked to determine its eigenvalue $e^{i\phi}$. If we let $U = e^{-iHt}$, this can be viewed as the transformation induced by $e^{-i(H\otimes p)t}$. Thus we see that the quantum phase estimation algorithm for a unitary operator U is exactly von Neumann's prescription for measuring $i \ln U$[15].

5.2 Quantum Metropolis Algorithm

Classically, Monte Carlo Metropolis algorithm is an effective solution to simulation of equilibrium properties for many physical systems. The key idea is sampling the more important configurations with higher probabilities. Its procedure is as following

1. Firstly, initialize an arbitrary configuration.
2. Randomly change the configuration by a little.
3. Compare the total energies of these two configurations.
4. If the new energy is lower than the previous one, we always accept this new configuration as the input to the next iteration. Otherwise, we accept it with some probability.
5. By repeating 2,3,4 steps, this algorithm will find the ground state (i.e., the lowest energy) and sample the whole configuration space.

For many realistic physical systems, this algorithm is efficient, but this algorithm fails in the anti-ferromagnetic systems, which is known as sign problem. A quantum version of the Metropolis algorithm[5] has been proposed to solve the sign problem. It permits sampling directly from the eigenstates of the Hamiltonian, and evades the sign problem.

In analogous to the classical one, we firstly prepare a random initial state $|\psi_0\rangle$ of the simulated system and the other three registers are initialized as $|0\rangle$. Then, we can utilize the our familiar quantum phase estimation algorithm to transform the initial state into a superposition of energy eigenstates and the eigenvalules E_i are encoded in the second register

$$|\psi_0\rangle_S|0\rangle_1^r|0\rangle_2^r|0\rangle_M \overset{QPE}{\longmapsto} \sum_i |\psi_i\rangle_S|E_i\rangle_1^r|0\rangle_2^r|0\rangle_M. \tag{19}$$

By measuring the second register in the basis of eigenvalues, this state collapses to one of the eigenstates. The second step is to randomly transform the state by a little, which can be realized by a random local unitary transformation

$$C|\psi_i\rangle_S = \sum_k x_k^i|\psi_k\rangle_S \tag{20}$$

Then, estimate the energy of new state and store it in the third register.

$$\sum_k x_k^i|\psi_k\rangle_S|E_i\rangle_1^r|0\rangle_2^r|0\rangle_M \overset{QPE}{\longmapsto} \sum_k x_k^i|\psi_k\rangle_S|E_i\rangle_1^r|E_k\rangle_2^r|0\rangle_M \tag{21}$$

Next step is to accept or reject the new state following the Metropolis rule. Classically, we can reject the state by keeping a copy of original state in computer memory and restore it. However, measurement is irreversible and cloning a state is impossible in the quantum mechanics. The solution is to make a measurement that reveals as little information as possible. So, this algorithm only use the last one qubit to represent acceptance or rejection. By rotating the last qubit, we can prepare a coherent superposition of accepting the new state and rejecting it.

$$\sum_k x_k^i|\psi_k\rangle_S|E_i\rangle_1|E_k\rangle_2^r|0\rangle_M \overset{W(E_k,E_i)}{\longmapsto} \sum_k x_k^i\sqrt{f_k^i}|\psi_k\rangle_S|E_i\rangle_1^r|E_k\rangle_2^r|1\rangle_M+\sum_k x_k^i\sqrt{1-f_k^i}|\psi_k\rangle_S|E_i\rangle_1^r|E_k\rangle_2^r|0\rangle_M \tag{22}$$

where $f_{ik} = \min(1, \exp(-\beta(E_k - E_i)))$ are exactly transition probabilities. Now, the measurement is completed by measuring the last qubit in the computational basis. The outcome 1 will project the superposition to the new state and we use this state as the input to the next Metropolis step, If the outcome is 0, it means we have to reject the new state and recover it to the original one. By repeating the Metropolis step, this algorithm is expected to find the ground state of the system.

5.3 Implementation of a toy model with a quantum language(Qiskit)

The simplest but non-trivial example is to simulate the Gibbs states of two Heisenberg ferromagnetic 1/2-spins, i.e.

$$H_2 = -\frac{1}{2}(\sigma_1^x \otimes \sigma_2^x + \sigma_1^y \otimes \sigma_2^y + \sigma_1^z \otimes \sigma_2^z) \tag{23}$$

With the appropriate energy offset, this Hamiltonian has the spectrum 0,2, where the eigenvalue 0 is threefold degenerate. This means that an exact phase estimation algorithm can be set up with just a single qubit of accuracy. Such a phase estimation requires simulating the Hamiltonian for a time $t = \pi/2$. One sees that this unitary corresponds exactly to the SWAP gate.

$$U(\frac{\pi}{2}) = e^{-i\frac{\pi}{2}H_2} = \text{SWAP} \tag{24}$$

In the quantum Metropolis algorithm, we need to implement the controlled version of this SWAP, which is exactly the Fredkin gate. The second gate to be implemented is the controlled Metropolis unitary W. The Metropolis unitary can be implemented with two controlled R_y rotations:

$$\begin{aligned} W(\theta_\beta) &= R_y(-\theta_\beta)_C \ X \ R_y(\theta_\beta)_C \\ &= \begin{pmatrix} \cos\frac{\theta}{2} & \sin\frac{\theta}{2} \\ -\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \\ &= \begin{pmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{pmatrix} \end{aligned} \tag{25}$$

where $\cos(\theta_\beta) = e^{-\beta}$. If the last register collapses to $|1\rangle$, we accept the new state and use it as the input of next iteration. Otherwise, we reject the new state and recover it to the original state.

It is worthwhile to note that like the classical Metropolis algorithm, the quantum Metropolis algorithm is not expected to reach the ground state of an arbitrary Hamiltonian in polynomial time. The ability to prepare the ground state of a general Hamiltonian in polynomial time would allow the

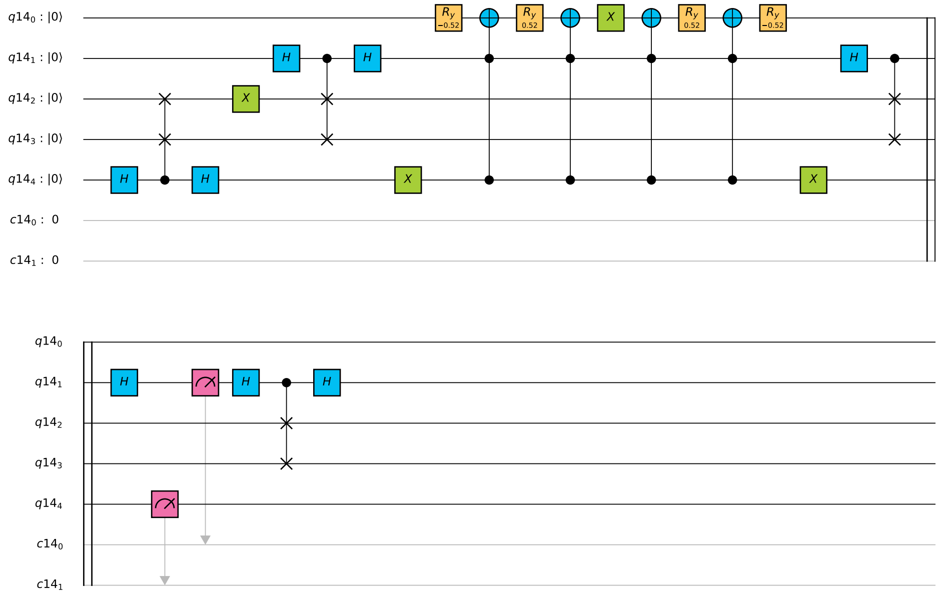

Figure 1: The circuit of quantum metropolis algorithm for two Heisenberg ferromagnetic 1/2-spins

solution to quantum Merlin Arthur (QMA)-complete problems, which is highly unlikely. However, for realistic physical systems, the convergence rate of the classical Metropolis algorithm is often very good, and it is conceivable that the same is also true for the quantum Metropolis algorithm.

6 Conclusions

In this project, we discussed two broad types of quantum simulations, namely simulation of evolution of Hamiltonian, and simulation of equilibrium properties. Hamiltonian simulation was achieved by unitary decomposition that recursively breaks a single evolution operator into multiple unitary operators. Lieb-Robinson bound was applied to achieve the tight bound for gate complexity of short range interactions. Meantime, a tight upper bound of gate complexity for long range interaction remains an open problem. The speedup of simulation of equilibrium properties is illustrated by introducing quantum version of Monte Carlo Metropolis algorithm. At last we implement a simple example of quantum Metropolis algorithm with a IBM's quantum programming language.
We appreciate professor Wu's insightful suggestions for our project.

References

[1] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6-7):467–488, 1982.

[2] David J Griffiths and Darrell F Schroeter. *Introduction to quantum mechanics*. Cambridge University Press, 2018.

[3] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[4] Matthias Troyer and Uwe-Jens Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Physical review letters*, 94(17):170201, 2005.

[5] Kristan Temme, Tobias J Osborne, Karl G Vollbrecht, David Poulin, and Frank Verstraete. Quantum metropolis sampling. *Nature*, 471(7336):87, 2011.

[6] IM Georgescu, Sahel Ashhab, and Franco Nori. Quantum simulation. *Reviews of Modern Physics*, 86(1):153, 2014.

[7] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

[8] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *SpringerLink*, 2006.

[9] Seth Lloyd. Universal quantum simulators. *Science*, 1996.

[10] Childs, Andrew M., Maslov, Dmitri, Ross, Neil J., and Yuan. Toward the first quantum simulation with quantum speedup, 2017.

[11] Masuo Suzuki. General theory of fractal path integrals with applications to manybody theories and statistical physics. *Journal of Mathematical Physics*, 32(2):400407, 1991.

[12] Andrew M. Childs and Robin Kothari. Simulating sparse hamiltonians with star decompositions. *Theory of Quantum Computation, Communication, and Cryptography Lecture Notes in Computer Science*, page 94103, 2011.

[13] Jeongwan, Hastings, Matthew B., Robin, Low, and Guang Hao. Quantum algorithm for simulating real time evolution of lattice hamiltonians, Sep 2018.

[14] Minh C Tran, Andrew Y Guo, Yuan Su, James R Garrison, Zachary Eldredge, Michael Foss-Feig, Andrew M Childs, and Alexey V Gorshkov. Locality and digital quantum simulation of power-law interactions. *arXiv preprint arXiv:1808.05225*, 2018.

[15] Andrew Childs. Simulating hamiltonian dynamics.