Chapter 7

Cluster quantum Monte Carlo algorithms for lattice models

7.1 World line representations for quantum lattice models

All quantum Monte Carlo algorithms are based on a mapping of a d-dimensional quantum system to a $(d+1)$-dimensional classical system using a path-integral formulation. We then perform classical Monte Carlo updates on the world lines of the particles. We will now introduce one modern algorithm for lattice models, the "loop-algorithm" which is a generalization of the classical cluster algorithms for the Ising model to quantum models.

We will discuss the loop algorithm for a spin-1/2 quantum XXZ model with the Hamiltonian

$$\begin{aligned} H &= -\sum_{\langle i,j\rangle} \left(J_z S_i^z S_j^z + J_{xy}(S_i^x S_j^x + S_i^y S_j^y) \right) \\ &= -\sum_{\langle i,j\rangle} \left(J_z S_i^z S_j^z + \frac{J_{xy}}{2}(S_i^+ S_j^- + S_i^- S_j^+) \right). \end{aligned} \tag{7.1}$$

For $J \equiv J_z = J_{xy}$ we have the Heisenberg model ($J > 0$ is ferromagnetic, $J < 0$ antiferromagnetic). $J_{xy} = 0$ is the (classical) Ising model and $J_z = 0$ the quantum XY model.

Continuous-time world lines

In contrast to models in continuous space, where a discrete time step was needed for the path integral, for lattice models the continuum limit can be taken. The spatial lattice is sufficient to regularize any ultraviolet divergencies.

We start by still discretizing discretize the imaginary time (inverse temperature) direction and subdivide $\beta = M\Delta\tau$:

$$e^{-\beta H} = \left(e^{-\Delta\tau H}\right)^M = (1 - \Delta\tau H)^M + O(\Delta\tau) \tag{7.2}$$

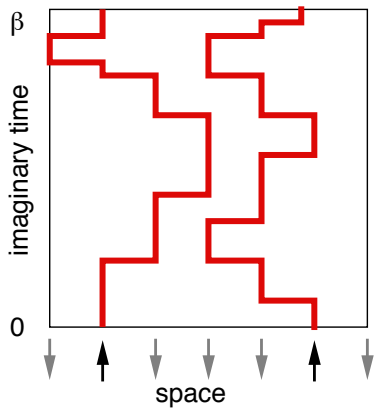


Figure 7.1: Example of a world line configuration for a spin-1/2 quantum Heisenberg model. Drawn are the world lines for up-spins only. Down spin world lines occupy the rest of the configuration.

In the limit $M \to \infty$ ($\Delta\tau \to 0$) this becomes exact. We will take the limit later, but stay at finite $\Delta\tau$ for the derivation.

The next step is to insert the identity matrix, represented by a sum over all basis states $\sum_i |i\rangle\langle i|$ between all operators $(1 - \Delta\tau H)$:

$$\begin{aligned} Z &= \mathrm{Tr} e^{-\beta H} = \mathrm{Tr}\,(1 - \Delta\tau H)^M + O(\Delta\tau) \\ &= \sum_{i_1,\ldots,i_M} \langle i_1|1 - \Delta\tau H|i_2\rangle\langle i_2|1 - \Delta\tau H|i_3\rangle \cdots \langle i_M|1 - \Delta\tau H|i_1\rangle + O(\Delta\tau) \\ =: P_{i_1,\ldots,i_M} \end{aligned} \tag{7.3}$$

and similarly for the measurement, obtaining

$$\langle A\rangle = \sum_{i_1,\ldots,i_M} \frac{\langle i_1|A(1 - \Delta\tau H)|i_2\rangle}{\langle i_1|1 - \Delta\tau H|i_2\rangle} P_{i_1,\ldots,i_M} + O(\Delta\tau). \tag{7.4}$$

If we choose the basis states $|i\rangle$ to be eigenstates of the local S^z operators we end up with an Ising-like spin system in one higher dimension. Each choice $i_1, \ldots, i_M$ corresponds to one of the possible configurations of this classical spin system. The trace is mapped to periodic boundary conditions in the imaginary time direction of this classical spin system. The probabilities are given by matrix elements $\langle i_n|1-\Delta\tau H|i_{n+1}\rangle$. We can now sample this classical system using classical Monte Carlo methods.

However, most of the matrix elements $\langle i_n|1-\Delta\tau H|i_{n+1}\rangle$ are zero, and thus nearly all configurations have vanishing weight. The only non-zero configurations are those where neighboring states $|i_n\rangle$ and $|i_{n+1}\rangle$ are either equal or differ by one of the off-diagonal matrix elements in H, which are nearest neighbor exchanges by two opposite spins. We can thus uniquely connect spins on neighboring "time slices" and end up with world lines of the spins, sketched in Fig. 7.1. Instead of sampling over all configurations of local spins we thus have to sample only over all world line configurations (the others have vanishing weight). Our update moves are not allowed to break world lines but have to lead to new valid world line configurations.

Finally we take the continuous time limit $\Delta\tau \to 0$. Instead of storing the configurations at all $M \to \infty$ time steps, we will not just store the times τ where a spin flips as the consequence of an off-diagonal operator acting at that time. The number of such events stays finite as $M \to \infty$: as can be seen from equation (7.2) the probability of H acting at a given time is proportional to $1/M$, and hence the total number of spin flips will stay finite although $M \to \infty$

7.2 Cluster updates

Before discussing cluster updates for quantum systems, we will review the cluster algorithms for the classical Ising model which should be known from the computational statistical physics course.

7.2.1 Kandel-Domany framework

To provide a general framework, which can be extended to quantum systems, we use the Fortuin-Kastelyn representation of the Ising model, as generalized by Kandel and Domany. The phase space of the Ising model is enlarged by assigning a set $\mathcal{G}$ of possible "graphs" to each configuration C in the set of configurations $\mathcal{C}$. We write the partition function as

$$Z = \sum_{C\in\mathcal{C}} \sum_{G\in\mathcal{G}} W(C,G) \tag{7.5}$$

where the new weights $W(C,G) > 0$ are chosen such that Z is the partition function of the original model by requiring

$$\sum_{G\in\mathcal{G}} W(C,G) = W(C) := \exp(-\beta E[C]), \tag{7.6}$$

where $E[C]$ is the energy of the configuration C.

The algorithm now proceeds as follows. First we assign a graph $G \in \mathcal{G}$ to the configuration C, chosen with the correct probability

$$P_C(G) = W(C,G)/W(C). \tag{7.7}$$

Then we choose a new configuration C' with probability $p[(C,G) \to (C',G)]$, keeping the graph G fixed; next a new graph G' is chosen

$$C \to (C,G) \to (C',G) \to C' \to (C',G') \to \ldots \tag{7.8}$$

What about detailed balance? The procedure for choosing graphs with probabilities P_G obeys detailed balance trivially. The non-trivial part is the probability of choosing a new configuration C'. There detailed balance requires:

$$W(C,G)p[(C,G) \to (C',G)] = W(C',G)p[(C',G) \to (C,G)], \tag{7.9}$$

which can be fulfilled using either the heat bath algorithm

$$p[(C,G) \to (C',G)] = \frac{W(C',G)}{W(C,G) + W(C',G)} \tag{7.10}$$

Table 7.1: Local bond weights for the Kandel-Domany representation of the Ising model.

	$c = \uparrow\uparrow$	$c = \downarrow\uparrow$	$c = \uparrow\downarrow$	$c = \downarrow\downarrow$	V(g)
$\Delta(c, \text{discon.})$	1	1	1	1	$\exp(-\beta J)$
$\Delta(c, \text{con.})$	1	0	0	1	$\exp(\beta J) - \exp(-\beta J)$
w(c)	$\exp(\beta J)$	$\exp(-\beta J)$	$\exp(-\beta J)$	$\exp(\beta J)$	

or by again using the Metropolis algorithm:

$$p[(C, G) \to (C', G)] = \min(W(C', G)/W(C, G), 1) \tag{7.11}$$

The algorithm simplifies a lot if we can find a graph mapping such that the graph weights do not depend on the configuration whenever it is nonzero in that configuration. This means, we want the graph weights to be

$$W(C, G) = \Delta(C, G)V(G), \tag{7.12}$$

where

$$\Delta(C, G) := \begin{cases} 1 \text{ if } W(C, G) \neq 0, \\ 0 \text{ otherwise.} \end{cases} . \tag{7.13}$$

Then equation (7.10) simply becomes $p = 1/2$ and equation (7.11) reduces to $p = 1$ for any configuration C' with $W(C', G) \neq 0$.

7.2.2 The cluster algorithms for the Ising model

Let us now show how this abstract and general algorithm can be applied to the Ising model. Our graphs will be bond-percolation graphs on the lattice. Spins pointing into the same direction can be connected or disconnected. Spins pointing in opposite directions will always be disconnected. In the Ising model we can write the weights $W(C)$ and $W(C, G)$ as products over all bonds b:

$$W(C) = \prod_b w(C_b) \tag{7.14}$$

$$W(C, G) = \prod_b w(C_b, G_b) = \prod_b \Delta(C_b, G_b)V(G_b) \tag{7.15}$$

where the local bond configurations C_b can be one of $\{\uparrow\uparrow, \downarrow\uparrow, \uparrow\downarrow, \downarrow\downarrow\}$

and the local graphs can be "connected" or "disconnected". The graph selection can thus be done locally on each bond.

Table 7.1 shows the local bond weights $w(c, g)$, $w(c)$, $\Delta(c, g)$ and $V(g)$. It can easily be checked that the sum rule (7.6) is satisfied.

The probability of a connected bond is $[\exp(\beta J) - \exp(-\beta J)]/\exp(\beta J) = 1 - \exp(-2\beta J)$ if two spins are aligned and zero otherwise. These connected bonds group the spins into clusters of aligned spins.

A new configuration C' with the same graph G can differ from C only by flipping clusters of connected spins. Thus the name "cluster algorithms". The clusters can be

flipped independently, as the flipping probabilities $p[(C, G) \to (C', G)]$ are configuration independent constants.

There are two variants of cluster algorithms that can be constructed using the rules derived above.

The Swendsen-Wang algorithm

The Swendsen-Wang or multi-cluster algorithm proceeds as follows:

i) Each bond in the lattice is assigned a label "connected" or "disconnected" according to above rules. Two aligned spins are connected with probability $1 - \exp(-2\beta J)$. Two antiparallel spins are never connected.

ii) Next a cluster labeling algorithm, like the Hoshen-Kopelman algorithm is used to identify clusters of connected spins.

iii) Measurements are performed, using improved estimators discussed in the next section.

iv) Each cluster of spins is flipped with probability 1/2.

The Wolff single-cluster algorithm

The Swendsen Wang algorithm gets less efficient in dimensions higher than two as the majority of the clusters will be very small ones, and only a few large clusters exist. The Wolff algorithm is similar to the Swendsen-Wang algorithm but builds only one cluster starting from a randomly chosen point. As the probability of this point being on a cluster of size s is proportional to s the Wolff algorithm builds preferedly larger clusters. It works in the following way:

i) Choose a random spin as the initial cluster.

ii) If a neighboring spin is parallel to the initial spin it will be added to the cluster with probability $1 - \exp(-2\beta J)$.

iii) Repeat step ii) for all points newly added to the cluster and repeat this procedure until no new points can be added.

iv) Perform measurements using improved estimators.

v) Flip all spins in the cluster.

We will see in the next section that the linear cluster size diverges with the correlation length ξ and that the average number of spins in a cluster is just χT. Thus the algorithm adapts optimally to the physics of the system and the dynamical exponent $z \approx 0$, thus solving the problem of critical slowing down. Close to criticality these algorithms are many orders of magnitudes (a factor L^2) better than the local update methods. Away from criticality sometimes a hybrid method, mixing cluster updates and local updates can be the ideal method.

7.2.3 Improved Estimators

In this section we present a neat trick that can be used in conjunction with cluster algorithms to reduce the variance, and thus the statistical error of Monte Carlo measurements. Not only do these "improved estimators" reduce the variance. They are also much easier to calculate than the usual "simple estimators".

To derive them we consider the Swendsen-Wang algorithm. This algorithm divides the lattice into N_c clusters, where all spins within a cluster are aligned. The next possible configuration is any of the 2^{N_c} configurations that can be reached by flipping any subset of the clusters. The idea behind the "improved estimators" is to measure not only in the new configuration but in all equally probable 2^{N_c} configurations.

As simplest example we consider the average magnetization $\langle m \rangle$. We can measure it as the expectation value $\langle \sigma_{\vec{i}} \rangle$ of a single spin. As the cluster to which the spin belongs can be freely flipped, and the flipped cluster has the same probability as the original one, the improved estimator is

$$\langle m \rangle = \langle \frac{1}{2}(\sigma_{\vec{i}} - \sigma_{\vec{i}}) \rangle = 0. \tag{7.16}$$

This result is obvious because of symmetry, but we saw that at low temperatures a single spin flip algorithm will fail to give this correct result since it takes an enormous time to flip all spins. Thus it is encouraging that the cluster algorithms automatically give the exact result in this case.

Correlation functions are not much harder to measure:

$$\langle \sigma_{\vec{i}} \sigma_{\vec{j}} \rangle = \begin{cases} 1 & \text{if } \vec{i} \text{ und } \vec{j} \text{ are on the same cluster} \\ 0 & \text{otherwise} \end{cases} \tag{7.17}$$

To derive this result consider the two cases and write down the improved estimators by considering all possible cluster flips.

Using this simple result for the correlation functions the mean square of the magnetization is

$$\langle m^2 \rangle = \frac{1}{N^2} \sum_{\vec{i},\vec{j}} \langle \sigma_{\vec{i}} \sigma_{\vec{j}} \rangle = \frac{1}{N^2} \langle \sum_{cluster} S(cluster)^2 \rangle, \tag{7.18}$$

where $S(cluster)$ is the number of spins in a cluster. The susceptibility above T_c is simply given by $\beta \langle m^2 \rangle$ and can also easily be calculated by above sum over the squares of the cluster sizes.

In the Wolff algorithm only a single cluster is built. Above sum (7.18) can be rewritten to be useful also in case of the Wolff algorithm:

$$\begin{aligned} \langle m^2 \rangle &= \frac{1}{N^2} \langle \sum_{cluster} S(cluster)^2 \rangle \\ &= \frac{1}{N^2} \sum_{\vec{i}} \frac{1}{S(\text{cluster containing } \vec{i})} S(\text{cluster containing } \vec{i})^2 \\ &= \frac{1}{N^2} \sum_{\vec{i}} S(\text{cluster containing } \vec{i}) = \frac{1}{N} \langle S(\text{cluster}) \rangle. \end{aligned} \tag{7.19}$$

Table 7.2: The six local configurations for an XXZ model and their weights.

configuration	weight
$S_i(\tau+d\tau)\uparrow$ $S_j(\tau+d\tau)\uparrow$ $S_i(\tau+d\tau)\downarrow$ $S_j(\tau+d\tau)\downarrow$ $S_i(\tau)\uparrow$ $S_j(\tau)\uparrow$, $S_i(\tau)\downarrow$ $S_j(\tau)\downarrow$	$1+\frac{J_z}{4}d\tau$
$S_i(\tau+d\tau)\uparrow$ $S_j(\tau+d\tau)\downarrow$ $S_i(\tau+d\tau)\downarrow$ $S_j(\tau+d\tau)\uparrow$ $S_i(\tau)\uparrow$ $S_j(\tau)\downarrow$, $S_i(\tau)\downarrow$ $S_j(\tau)\uparrow$	$1-\frac{J_z}{4}d\tau$
$S_i(\tau+d\tau)\downarrow$ $S_j(\tau+d\tau)\uparrow$ $S_i(\tau+d\tau)\uparrow$ $S_j(\tau+d\tau)\downarrow$ $S_i(\tau)\uparrow$ $S_j(\tau)\downarrow$, $S_i(\tau)\downarrow$ $S_j(\tau)\uparrow$	$\frac{J_{xy}}{2}d\tau$

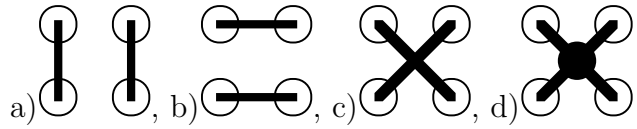


Figure 7.2: The four local graphs: a) vertical, b) horizontal c) crossing and d) freezing (connects all four corners).

The expectation value for m^2 is thus simply the mean cluster size. In this derivation we replaced the sum over all clusters by a sum over all sites and had to divide the contribution of each cluster by the number of sites in the cluster. Next we can replace the average over all lattice sites by the expectation value for the cluster on a randomly chosen site, which in the Wolff algorithm will be just the one Wolff cluster we build.

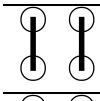
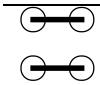
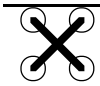
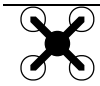
7.2.4 The loop algorithm for quantum spins

We will now generalize these cluster algorithms to quantum systems and present the loop algorithm. [1]

This algorithm is best described by first taking the continuous time limit $M \to \infty$ ($\Delta\tau \to d\tau$) and by working with infinitesimals. Similar to the Ising model we look at two spins on neigboring sites i and j at two neighboring times τ and $\tau + d\tau$, as sketched in Tab. 7.2. There are a total of six possible configurations, having three different probabilities. The total probabilities are the products of all local probabilities, like in the classical case. This is obvious for different time slices. For the same time slice it is also true since, denoting by H_{ij} the term in the Hamiltonian H acting on the bond between sites i and j we have $\prod_{\langle i,j\rangle}(1 - d\tau H_{ij}) = 1 - d\tau \sum_{\langle i,j\rangle} H_{ij} = 1 - d\tau H$. In the following we focus only on such local four-spin plaquettes. Next we again use the Kandel-Domany framework and assign graphs. As the updates are not allowed to break world lines only four graphs, sketched in Fig. 7.2 are allowed. Finally we have to find Δ functions and graph weights that give the correct probabilities. The solution for the

[1] H. G. Evertz et al., Phys. Rev. Lett. **70**, 875 (1993); B. B. Beard and U.-J. Wiese, Phys. Rev. Lett. **77**, 5130 (1996); B. Ammon, H. G. Evertz, N. Kawashima, M. Troyer and B. Frischmuth, Phys. Rev. B **58**, 4304 (1998).

Table 7.3: The graph weights for the quantum-XY model and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero.

G	$\Delta(\uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, G)$ $= \Delta(\uparrow\downarrow, G)$	graph weight
	1	1	–	$1 - \frac{J_{xy}}{4} d\tau$
	–	1	1	$\frac{J_{xy}}{4} d\tau$
	1	–	1	$\frac{J_{xy}}{4} d\tau$
	0	0	0	0
total weight	1	1	$\frac{J_{xy}}{2} d\tau$	

XY-model, ferromagnetic and antiferromagnetic Heisenberg model and the Ising model is shown in Tables 7.3 - 7.6.

Let us first look at the special case of the Ising model. As the exchange term is absent in the Ising model all world lines run straight and can be replaced by classical spins. The only non-trivial graph is the "freezing", connecting two neighboring world lines. Integrating the probability that two neighboring sites are *nowhere* connected along the time direction we obtain: times:

$$\prod_{\tau=0}^{\beta} (1 - d\tau J/2) = \lim_{M\to\infty} (1 - \Delta\tau J/2)^M = \exp(-\beta J/2) \tag{7.20}$$

Taking into account that the spin is $S = 1/2$ and the corresponding classical coupling $J_{cl} = S^2 J = J/4$ we find for the probability that two spins are *connected*: $1 - \exp(-2\beta J_{cl})$. We end up exactly with the cluster algorithm for the classical Ising model!

The other cases are special. Here each graph connects two spins. As each of these spins is again connected to only one other, all spins connected by a cluster form a closed loop, hence the name "loop algorithm". Only one issue remains to be explained: how do we assign a horizontal or crossing graph with infinitesimal probability, such as $(J/2)d\tau$. This is easily done by comparing the assignment process with radioactive decay. For each segment the graph runs vertical, except for occasional decay processes occuring with probability $(J/2)d\tau$. Instead of asking at every infinitesimal time step whether a decay occurs we simply calculate an exponentially distributed decay time t using an exponential distribution with decay constant $J/2$. Looking up the equation in the lecture notes of the winter semester we have $t = -(2/J)\ln(1-u)$ where u is a uniformly distributed random number.

Table 7.4: The graph weights for the ferromagnetic quantum Heisenberg model and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero.

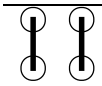
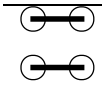
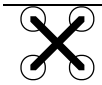
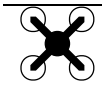
G	$\Delta(\uparrow\uparrow, \uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, \downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, \uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, \downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, \uparrow\downarrow, G)$ $= \Delta(\uparrow\downarrow, \downarrow\uparrow, G)$	graph weight
	1	1	–	$1 - \frac{J}{4}d\tau$
	–	0	0	0
	1	–	1	$\frac{J}{2}d\tau$
	0	0	0	0
total weight	$1 + \frac{J}{4}d\tau$	$1 - \frac{J}{4}d\tau$	$\frac{J}{2}d\tau$	

Table 7.5: The graph weights for the antiferromagnetic quantum Heisenberg model and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero. To avoid the sign problem (see next subsection) we change the sign of J_{xy}, which is allowed only on bipartite lattices.

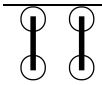
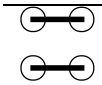
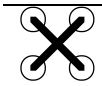
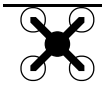
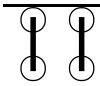
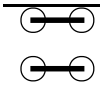
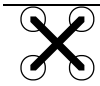
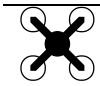
G	$\Delta(\uparrow\uparrow, \uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, \downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, \uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, \downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, \uparrow\downarrow, G)$ $= \Delta(\uparrow\downarrow, \downarrow\uparrow, G)$	graph weight
	1	1	–	$1 - \frac{\lvert J\rvert}{4}d\tau$
	–	1	1	$\frac{\lvert J\rvert}{2}d\tau$
	0	–	0	0
	0	0	0	0
total weight	$1 - \frac{\lvert J\rvert}{4}d\tau$	$1 + \frac{\lvert J\rvert}{4}d\tau$	$\frac{\lvert J\rvert}{2}d\tau$	

Table 7.6: The graph weights for the ferromagnetic Ising model and the Δ function specifying whether the graph is allowed. The dash – denotes a graph that is not possible for a configuration because of spin conservation and has to be zero.

G	$\Delta(\uparrow\uparrow, G)$ $= \Delta(\downarrow\downarrow, G)$	$\Delta(\uparrow\downarrow, G)$ $= \Delta(\downarrow\uparrow, G)$	$\Delta(\downarrow\uparrow, G)$ $= \Delta(\uparrow\downarrow, G)$	graph weight
	1	1	–	$1 - \frac{J_z}{4} d\tau$
	–	0	0	0
	0	–	0	0
	1	0	0	$\frac{J_z}{2} d\tau$
total weight	$1 + \frac{J_z}{4} d\tau$	$1 - \frac{J_z}{4} d\tau$	0	

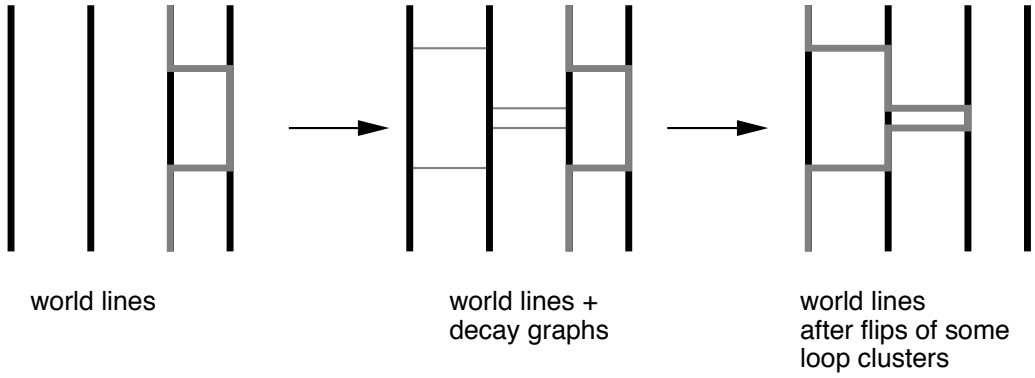


Figure 7.3: Example of a loop update. In a first step decay paths are inserted where possible at positions drawn randomly according to an exponential distribution and graphs are assigned to all exchange terms (hoppings of world lines). In a second stage (not shown) the loop clusters are identified. Finally each loop cluster is flipped with probability 1/2 and one ends up with a new configuration.

The algorithm now proceeds as follows (see Fig. 7.3): for each bond we start at time 0 and calculate a decay time. If the spins at that time are oriented properly and an exchange graph is possible we insert one. Next we advance by another randomly chosen decay time along the same bond and repeat the procedure until we have reached the extent β. This assigns graphs to all infinitesimal time steps where spins do not change. Next we assign a graph to all of the (finite number of) time steps where two spins are exchanged. In the case of the Heisenberg models there is always only one possible graph to assign and this is very easy. In the next step we identify the loop-clusters and then flip them each with probability 1/2. Alternatively a Wolff-like algorithm can be constructed that only builds one loop-cluster.

Improved estimators for measurements can be constructed like in classical models. The derivation is similar to the classical models. I will just mention two simple ones for the ferromagnetic Heisenberg model. The spin-spin corelation is

$$S_i^z(\tau)S_j^z(\tau') = \begin{cases} 1 & \text{if } (i,\tau) \text{ und } (j,\tau') \text{ are on the same cluster} \\ 0 & \text{otherwise} \end{cases} \tag{7.21}$$

and the uniform susceptibilty is

$$\chi = \frac{1}{N\beta}\sum_c S(c)^2, \tag{7.22}$$

where the sum goes over all loop clusters and $S(c)$ is the length of all the loop segments in the loop cluster c.

7.3 The negative sign problem

Now that we have an algorithm with no critical slowing down we could think that we have completely solved the problem of quantum many body problems. However, in this section we will show that the sign problem is **NP**-hard, following the paper M. Troyer and U.J. Wiese, Phys. Rev. Lett. **94**, 170201 (2005).

The difficulties in finding polynomial time solutions to the sign problem are reminiscent of the apparent impossibility to find polynomial time algorithms for nondeterministic polynomial (**NP**)-complete decision problems, which could be solved in polynomial time on a hypothetical non-deterministic machine, but for which no polynomial time algorithm is known for deterministic classical computers. A hypothetical non-deterministic machine can always follow both branches of an if-statement simultaneously, but can never merge the branches again. It can, equivalently, be viewed as having exponentially many processors, but without any communication between them. In addition, it must be possible to check a positive answer to a problem in **NP** on a classical computer in polynomial time.

Many important computational problems in the complexity class **NP**, including the traveling salesman problem and the problem of finding ground states of spin glasses have the additional property of being **NP**-hard, forming the subset of **NP**-complete problems, the hardest problems in **NP**. A problem is called **NP**-hard if any problem in **NP** can be mapped onto it with polynomial complexity. Solving an **NP**-hard problem

is thus equivalent to solving any problem in **NP**, and finding a polynomial time solution to any of them would have important consequences for all of computing as well as the security of classical encryption schemes. In that case all problems in **NP** could be solved in polynomial time, and hence **NP**=**P**.

As no polynomial solution to any of the **NP**-complete problems was found despite decades of intensive research, it is generally believed that **NP**≠**P** and no deterministic polynomial time algorithm exists for these problems. The proof of this conjecture remains as one of the unsolved millennium problems of mathematics for which the Clay Mathematics Institute has offered a prize of one million US$. In this section we will show that the sign problem is **NP**-hard, implying that unless the **NP**≠**P** conjecture is disproved there exists no generic solution of the sign problem.

Before presenting the details of our proof, we will give a short introduction to classical and quantum Monte Carlo simulations and the origin of the sign problem. In the calculation of the phase space average of a quantity A, instead of directly evaluating the sum

$$\langle A\rangle = \frac{1}{Z}\sum_{c\in\Omega} A(c)p(c)\ , \quad Z=\sum_{c\in\Omega} p(c), \tag{7.23}$$

over a high-dimensional space Ω of configurations c, a classical Monte Carlo method chooses a set of M configurations $\{c_i\}$ from Ω, according to the distribution $p(c_i)$. The average is then approximated by the sample mean

$$\langle A\rangle \approx \overline{A} = \frac{1}{M}\sum_{i=1}^{M} A(c_i), \tag{7.24}$$

within a statistical error $\Delta A = \sqrt{\mathrm{Var}A(2\tau_A+1)/M}$, where $\mathrm{Var}A$ is the variance of A and the integrated autocorrelation time τ_A is a measure of the autocorrelations of the sequence $\{A(c_i)\}$. In typical statistical physics applications, $p(c)=\exp(-\beta E(c))$ is the Boltzmann weight, $\beta = 1/k_BT$ is the inverse temperature, and $E(c)$ is the energy of the configuration c.

Since the dimension of configuration space Ω grows linearly with the number N of particles, the computational effort for the direct integration Eq. (7.23) scales exponentially with the particle number N. Using the Monte Carlo approach the same average can be estimated to any desired accuracy in *polynomial time*, as long as the autocorrelation time τ_A does not increase faster than polynomially with N.

In a quantum system with Hamilton operator H, instead of an integral like Eq. (7.23), an operator expression

$$\langle A\rangle = \frac{1}{Z}\mathrm{Tr}[A\exp(-\beta H)]\ , \quad Z=\mathrm{Tr}\exp(-\beta H) \tag{7.25}$$

needs to be evaluated in order to calculate the thermal average of the observable A (represented by a self-adjoint operator). Monte Carlo techniques can again be applied to reduce the exponential scaling of the problem, but only after mapping the quantum model to a classical one, for example using world line configurations c with weights $p(c)$. If all the weights $p(c)$ are positive, standard Monte Carlo methods can be applied, as

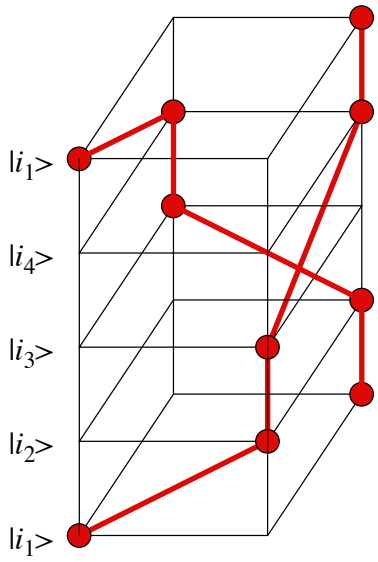


Figure 7.4: A configuration of a fermionic lattice model on a 4-site square. The configuration has negative weight, since two fermions are exchanged in the sequence $|i_1\rangle \to |i_2\rangle \to |i_3\rangle \to |i_4\rangle \to |i_1\rangle$. World lines connecting particles on neighboring slices are drawn as thick lines.

it is the case for non-frustrated quantum magnets and bosonic systems. In fermionic systems negative weights $p(c) < 0$ arise from the Pauli exclusion principle, when along the sequence $|i_1\rangle \to |i_2\rangle \to \cdots \to |i_n\rangle \to |i_1\rangle$ two fermions are exchanged, as shown in Fig. 7.4.

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 $s(c) \equiv \operatorname{sign} p(c)$ to the quantity being sampled:

$$\begin{aligned} \langle A \rangle &= \frac{\sum_c A(c)p(c)}{\sum_c p(c)} \\ &= \frac{\sum_c A(c)s(c)|p(c)| \,/\sum_c |p(c)|}{\sum_c s(c)|p(c)| \,/\sum_c |p(c)|} \equiv \frac{\langle As\rangle'}{\langle s\rangle'}. \end{aligned} \tag{7.26}$$

While this allows Monte Carlo simulations to be performed, the errors increase exponentially with the particle number N and the inverse temperature β. To see this, consider the mean value of the sign $\langle s\rangle = Z/Z'$, which is just the ratio of the partition functions of the fermionic system $Z = \sum_c p(c)$ with weights $p(c)$ and the bosonic system used for sampling with $Z' = \sum_c |p(c)|$. As the partition functions are exponentials of the corresponding free energies, this ratio is an exponential of the differences Δf in the free energy densities:$\langle s\rangle = Z/Z' = \exp(-\beta N \Delta f)$. As a consequence, the relative error $\Delta s/\langle s\rangle$ increases exponentially with increasing particle number and inverse temperature:

$$\frac{\Delta s}{\langle s\rangle} = \frac{\sqrt{\left(\langle s^2\rangle - \langle s\rangle^2\right)/M}}{\langle s\rangle} = \frac{\sqrt{1-\langle s\rangle^2}}{\sqrt{M}\langle s\rangle} \sim \frac{e^{\beta N \Delta f}}{\sqrt{M}}. \tag{7.27}$$

Similarly the error for the numerator in Eq. (7) increases exponentially and the time needed to achieve a given relative error scales exponentially in N and β.

In order to avoid any misconception about what would constitute a "solution" of the sign problem, we start by giving a precise definition:

- A quantum Monte Carlo simulation to calculate a thermal average $\langle A \rangle$ of an observable A in a quantum system with Hamilton operator H is defined to suffer from a *sign problem* if there occur negative weights $p(c) < 0$ in the classical representation.

- The related *bosonic system* of a fermionic quantum system is defined as the system where the weights $p(c)$ are replaced by their absolute values $|p(c)|$, thus ignoring the minus sign coming from fermion exchanges:

$$\langle A \rangle' = \frac{1}{Z'} \sum_c A(c)|p(c)|. \tag{7.28}$$

- An algorithm for the stochastic evaluation of a thermal average such as Eq. (7.28) is defined to be of *polynomial complexity* if the computational time $t(\epsilon, N, \beta)$ needed to achieve a relative statistical error $\epsilon = \Delta A/\langle A \rangle$ in the evaluation of the average $\langle A \rangle$ scales polynomially with the system size N and inverse temperature β, i.e. if there exist integers n and m and a constant $\kappa < \infty$ such that

$$t(\epsilon, N, \beta) < \kappa \epsilon^{-2} N^n \beta^m. \tag{7.29}$$

- For a quantum system that suffers from a sign problem for an observable A, and for which there exists a polynomial complexity algorithm for the related bosonic system Eq. (7.28), we define a *solution of the sign problem* as an algorithm of polynomial complexity to evaluate the thermal average $\langle A \rangle$.

It is important to note that we only worry about the sign problem if the bosonic problem is easy (of polynomial complexity) but the fermionic problem hard (of exponential complexity) due to the sign problem. If the bosonic problem is already hard, e.g. for spin glasses [2], the sign problem will not increase the complexity of the problem. Also, changing the representation so that all $p(c) \geq 0$ might not sufficient to solve the sign problem if the scaling remains exponential, since then we just map the sign problem to another exponentially hard problem. Only a polynomial complexity algorithm counts as a solution of the sign problem.

At first sight such a solution seems feasible since the sign problem is not an intrinsic property of the quantum model studied but is representation-dependent: it depends on the choice of basis sets $\{|i\rangle\}$, and in some models it can be solved by a simple local basis change.. Indeed, when using the eigen basis in which the Hamilton operator H is diagonal, there will be no sign problem. This diagonalization of the Hamilton operator is, however, no solution of the sign problem since its complexity is exponential in the number of particles N.

[2] F. Barahona, J. Phys. A **15**, 3241 (1982)

We now construct a quantum mechanical system for which the calculation of a thermal average provides the solution for one and thus all of the **NP**-complete problems. This system exhibits a sign problem, but the related bosonic problem is easy to solve. Since, for this model, a solution of the sign problem would provide us with a polynomial time algorithm for an **NP**-complete problem, the sign problem is **NP**-hard. Of course, it is expected that the corresponding thermal averages cannot be calculated in polynomial time and the sign problem thus cannot be solved. Otherwise we would have found a polynomial time algorithm for the **NP**-complete problems and would have shown that **NP**=**P**.

The specific **NP**-complete problem we consider is to determine whether a state with energy less than or equal to a bound E_0 exists for a classical three-dimensional Ising spin glass with Hamilton function

$$H = -\sum_{\langle j,k \rangle} J_{jk}\sigma_j\sigma_k. \tag{7.30}$$

Here the spins σ_j take the values ± 1, and the couplings J_{jk} between nearest neighbor lattice points j and k are either 0 or $\pm J$.

This problem is in the complexity class **NP** since the non-deterministic machine can evaluate the energies of all configurations c in polynomial time and test whether there is one with $E(c) \leq E_0$. In addition, the validity of a positive answer (i.e. there is a configuration c) can be tested on a deterministic machine by evaluating the energy of that configuration. The evaluation of the partition function $Z = \sum_c \exp(-\beta E(c))$ is, however, not in **NP** since the non-deterministic machine cannot perform the sum in polynomial time.

This question whether there is a state with energy $E(c) \leq E_0$ can also be answered in a Monte Carlo simulation by calculating the average energy of the spin glass at a large enough inverse temperature β. Since the energy levels are discrete with spacing J it can easily be shown that by choosing an inverse temperature $\beta J \geq N \ln 2 + \ln(12N)$ the thermal average of the energy will be less than $E_0 + J/2$ if at least one configuration with energy E_0 or less exists, and larger than $E_0 + J$ otherwise.

In this classical Monte Carlo simulation, the complex energy landscape, created by the frustration in the spin glass (Fig. 2a), exponentially suppresses the tunneling of the Monte Carlo simulation between local minima at low temperatures. The autocorrelation times and hence the time complexity of this Monte Carlo approach are exponentially large $\tau \propto \exp(aN)$, as expected for this **NP**-complete problem.

We now map this classical system to a quantum system with a sign problem. We do so by replacing the classical Ising spins by quantum spins. Instead of the common choice in which the classical spin configurations are basis states and the spins are represented by diagonal σ_j^z Pauli matrices we choose a representation in which the spins point in the $\pm x$ direction and are represented by σ_j^x Pauli matrices:

$$H = -\sum_{\langle j,k \rangle} J_{jk}\sigma_j^x\sigma_k^x, \tag{7.31}$$

Here the random signs of the couplings are mapped to random signs of the off-diagonal matrix elements which cause a sign problem (see Fig. 2b). The related bosonic model

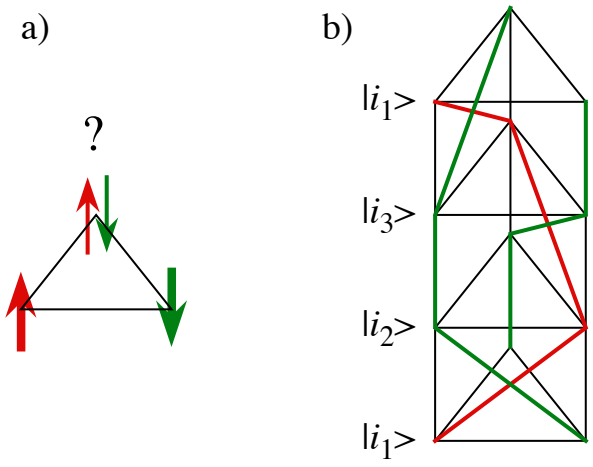


Figure 7.5: a) A classically frustrated spin configuration of three antiferromagnetically coupled spins: no configuration can simultaneously minimize the energy of all three bonds. b) A configuration of a frustrated quantum magnet with negative weights: three antiferromagnetic exchange terms with negative weights are present in the sequence $|i_1\rangle \rightarrow |i_2\rangle \rightarrow |i_3\rangle \rightarrow |i_1\rangle$. Here up-spins with z-component of spin $\sigma_j^z = 1$ and down-spins with $\sigma_j^z = -1$ are connected with differently colored world lines.

is the ferromagnet with all couplings $J_{jk} \geq 0$ and efficient cluster algorithms with polynomial time complexity are known for this model. Since the bosonic version is easy to simulate, the *sign problem is the origin of the* **NP**-*hardness* of a quantum Monte Carlo simulation of this model. A generic solution of the sign problem would provide a polynomial time solution to this, and thus to all, **NP**-complete problems, and would hence imply that **NP**=**P**. Since it is generally believed that **NP**≠**P**, we expect that such a solution does not exist.

By constructing a concrete model we have shown that the sign problem of quantum Monte Carlo simulations is **NP**-hard. This does not exclude that a specific sign problem can be solved for a restricted subclass of quantum systems. This was indeed possible using the meron-cluster algorithm [3] for some particular lattice models. Such a solution must be intimately tied to properties of the physical system and allow an essentially bosonic description of the quantum problem. A generic approach might scale polynomially for some cases but will in general scale exponentially.

In the case of fermions or frustrated quantum magnets, solving the sign problem requires a mapping to a bosonic or non-frustrated system – which is, in general, almost certainly impossible for physical reasons. The origin of the sign problem is, in fact, the distinction between bosonic and fermionic systems. The brute-force approach of taking the absolute values of the probabilities means trying to sample a frustrated or fermionic system by simulating a non-frustrated or bosonic one. As for large system sizes N and low temperatures the relevant configurations for the latter are not the relevant ones for the former, the errors are exponentially large.

[3] S. Chandrasekharan and U.-J. Wiese, Phys. Rev. Lett. **83**, 3116 (1999)

Given the **NP**-hardness of the sign problem one promising idea for the simulation of fermionic systems is to use ultra-cold atoms in optical lattices to construct well-controlled and tunable implementations of physical systems, such as the Hubbard model, and to use these "quantum simulators" to study the phase diagrams of correlated quantum systems. But even these quantum simulators are most likely not a generic solution to the sign problem since there exist quantum systems with exponentially diverging time scales and it is at present not clear whether a quantum computer could solve the **NP**-complete problems.

7.4 Worm and directed loop updates

The quantum loop algorithm, like the classical cluster algorithms work only for models with spin-inversion symmetry: a configuration of spins and the flipped configuration need to have the same weight. Applying a magnetic field breaks this spin-inversion symmetry, and the loop algorithm cannot be applied. For that case, Nikolay Prokof'ev and coworkers invented the worm algorithm.[4] The worm algorithm is based on a very simple idea: while local updates on world lines are not ergodic (since knots cannot be created or undone), the worm algorithm proceeds by cutting a world line, and then moves the open ends in local updates until they meet again and the world line is glued together once more.

[4] N. V. Prokof'ev, B. V. Svistunov and I. S. Tupitsyn, Sov. Phys - JETP **87**, 310 (1998).