Advanced methods in Simulation: Part III

Pascal Viot

November 6, 2020

Critical slowing down

But, close to the critical point, large scale fluctuations are present and the typical relaxation time increases accordingly; This time is related to the growing correlation length by the scaling relation

$$\tau \sim (\xi(t))^z$$

where z is a new exponent, the so-called dynamical exponent. This exponent depends on the dynamics of the system.
In Molecular Dynamics, this time is a real physical time.
Dynamics of the Monte Carlo simulation is very versatile, because it depends on the algorithm. The Metropolis algorithm mimics in part the real physical time and typically varies between 2 and 5 for pure systems.

Critical slowing down

In simulation, the system is finite. the correlation length is bound by the linear system size L and the relaxation time increases as

$$\tau \sim L^z;$$

Far from a phase transition, the computer time increases as the system size, because we need to keep the number of elemetary steps per particle constant. Close a phase transition, the simulation time increases as L^{d+z}. The relaxation becomes prohibitive!

Critical slowing down

In simulation, the system is finite. the correlation length is bound by the linear system size L and the relaxation time increases as

$$\tau \sim L^z;$$

Far from a phase transition, the computer time increases as the system size, because we need to keep the number of elemetary steps per particle constant. Close a phase transition, the simulation time increases as L^{d+z}. The relaxation becomes prohibitive!
Monte Carlo simulation can reduce the dynamical by choosing an appropriate method.

First-order phase transition

For a finite system, close the phase transition, and for a lattice size larger than the finite correlation length, the equilibrium energy distribution is a bimodal distribution. The ratio between the minimum and the maximum is proportional to the free energy interface.
Therefore, the characteristic time of the relaxation goes

$$\tau \propto \exp(AL^{d-1})$$

where L is the linear size of the simulation box d the space dimension.
Because the simulation time is proportionnal to the volume of the simulation box , the simulation time goes as $L^d \exp(AL^{d-1})$, which leads to a worst situation than for a second-order phase transition.
Once again, the Metropolis algorithm is not appropriate for studying phase transition due to a fast increase of the relation time.

Reweighting Method

To obtain a complete phase of a given system, a large number of simulations are required for scanning the entire range of temperatures, or for the range of external fields. When one attempts to locate a phase transition precisely, the number of simulation runs can increase as the size of the simulation box. become prohibitive. How to solve these problems?

Reweighting Method

To obtain a complete phase of a given system, a large number of simulations are required for scanning the entire range of temperatures, or for the range of external fields. When one attempts to locate a phase transition precisely, the number of simulation runs can increase as the size of the simulation box. become prohibitive. How to solve these problems?

Consider the partition function

$$\begin{aligned} Z(\beta, N) &= \sum_{\{\alpha\}} \exp(-\beta H(\{\alpha\})) \\ &= \sum_{i=1} g(i) \exp(-\beta E(i)) \end{aligned}$$

where $\{\alpha\}$ denotes all available states, the index i runs over all available energies of the system, $g(i)$ denotes the density of states of energy $E(i)$.

Reweighting Method (2)

For a given inverse temperature $\beta' = 1/(k_B T')$, the partition function can be expressed as

$$\begin{aligned} Z(\beta', N) &= \sum_i g(i) \exp(-\beta' E(i)) \\ &= \sum_i g(i) \exp(-\beta E(i)) \exp(-(\beta' - \beta) E(i)) \\ &= \sum_i g(i) \exp(-\beta E(i)) \exp(-(\Delta\beta) E(i)), \end{aligned} \tag{1}$$

where $\Delta\beta = (\beta' - \beta)$.

Reweighting Method (2)

Similarly, a thermal average, like the mean energy, is expressed as:

$$\begin{aligned}\langle E(\beta', N)\rangle &= \frac{\sum_i g(i)E(i)\exp(-\beta' E(i))}{\sum_i g(i)\exp(-\beta' E(i))} \\ &= \frac{\sum_i g(i)E(i)\exp(-\beta E(i))\exp(-(\Delta\beta)E(i))}{\sum_i g(i)\exp(-\beta E(i))\exp(-(\Delta\beta)E(i))}. \end{aligned} \tag{2}$$

Reweighting Method (2)

Similarly, a thermal average, like the mean energy, is expressed as:

$$\begin{aligned}\langle E(\beta', N)\rangle &= \frac{\sum_i g(i)E(i)\exp(-\beta' E(i))}{\sum_i g(i)\exp(-\beta' E(i))} \\ &= \frac{\sum_i g(i)E(i)\exp(-\beta E(i))\exp(-(\Delta\beta)E(i))}{\sum_i g(i)\exp(-\beta E(i))\exp(-(\Delta\beta)E(i))}.\end{aligned} \tag{2}$$

In a Monte Carlo simulation, one obtains the energy histogram associated with visited configurations.
This histogram, denoted $D_\beta(i)$, is proportional to the density of states $g(i)$ weighted by the Boltzmann factor,

$$D_\beta(i) = C(\beta)g(i)\exp(-\beta E(i)) \tag{3}$$

where $C(\beta)$ is a temperature dependent constant.

Reweighting Method (3)

The mean energy is estimated at a different temperature β'.

$$\langle E(\beta', N) \rangle = \frac{\sum_i D_\beta(i) E(i) \exp(-(\Delta\beta) E(i))}{\sum_i D_\beta(i) \exp(-(\Delta\beta) E(i))}.$$

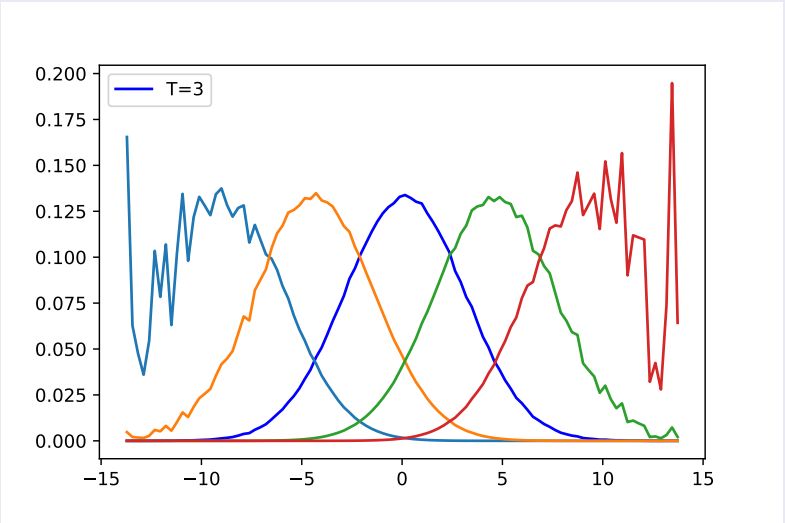
All moments can be also estimated, and therefore the specific heat. The energy distribution can be also estimated and can display the anomalies which appears on the tails of the distribution

Reweighting Method (3)

The mean energy is estimated at a different temperature β'.

$$\langle E(\beta', N) \rangle = \frac{\sum_i D_\beta(i) E(i) \exp(-(\Delta\beta) E(i))}{\sum_i D_\beta(i) \exp(-(\Delta\beta) E(i))}.$$

All moments can be also estimated, and therefore the specific heat. The energy distribution can be also estimated and can display the anomalies which appears on the tails of the distribution



Cluster Algorithm

We consider below the Ising model, but be generalized the method can be done for all locally symmetric Hamiltonian (for the Ising model, this corresponds to the up/down symmetry). Fluctuations consists in large regions of same values of spins.
Generate a flip involving large number of spins. With Metropolis algorimth, low acceptance rate.
Satisfy the balance equation

$$\frac{P_{gc}(o)W(o \to n)}{P_{gc}(n)W(n \to o)} = \exp(-\beta(U(n) - U(o))) \tag{4}$$

where $P_{gc}(o)$ is the probability of generating a bond configuration from o to n (o and n denote the old and new configurations, respectively)

Cluster Algorithm(2)

One wishes to accept all new configurations ($W(i \to j) = 1$), whatever configurations i and j, one must have the following ratio

$$\frac{P_{gc}(o)}{P_{gc}(n)} = \exp(-\beta(U(n) - U(o))) \tag{5}$$

Cluster Algorithm(2)

One wishes to accept all new configurations ($W(i \to j) = 1$), whatever configurations i and j, one must have the following ratio

$$\frac{P_{gc}(o)}{P_{gc}(n)} = \exp(-\beta(U(n) - U(o))) \tag{5}$$

In order to build such an algorithm, let us note that energy is simply expressed as

$$U = (N_a - N_p)J$$

where N_p is the number of pairs of nearest neighbor spins of the same sign and N_a the number of pairs of nearest neighbor spins of the opposite sign.
This formula, beyond its simplicity, has the advantage of being exact whatever the dimensionality of the system. The method relies on rewriting of the partition function given by Fortuyn and Kasteliyn (1969)

Cluster Algorithm(3)

For building clusters, one defines bonds between nearest neighbor spins with the following rules

1. If two nearest neighbor spins have opposite signs, they are not connected.
2. If two nearest neighbor spins have the same sign, they are connected with a probability p and disconnected with a probability $(1 - p)$.

   This rule assumes that J is positive (ferromagnetic system)

Cluster Algorithm(3)

For building clusters, one defines bonds between nearest neighbor spins with the following rules

1. If two nearest neighbor spins have opposite signs, they are not connected.
2. If two nearest neighbor spins have the same sign, they are connected with a probability p and disconnected with a probability $(1-p)$.

   This rule assumes that J is positive (ferromagnetic system)

Let consider a configuration of N_p pairs of spins of same sign, the probability of having n_c pairs connected (and consequently $n_b = N_p - n_c$ pairs of spins are disconnected) is given by the relation

$$P_{gc}(o) = p^{n_c}(1-p)^{n_b}. \tag{6}$$

Cluster Algorithm(4)

Once bonds are set, a cluster is a set of spins connected by at least one bond between them. If one flips a cluster ($o \to n$), the number of bonds between pairs of spins of same and opposite signs is changed and one has

$$N_p(n) = N_p(o) + \Delta$$

and

$$N_a(n) = N_a(o) - \Delta$$

Cluster Algorithm(4)

Once bonds are set, a cluster is a set of spins connected by at least one bond between them. If one flips a cluster ($o \to n$), the number of bonds between pairs of spins of same and opposite signs is changed and one has

$$N_p(n) = N_p(o) + \Delta$$

and

$$N_a(n) = N_a(o) - \Delta$$

The energy of the new configuration $U(n)$ is simply related to the energy of the old configuration $U(o)$ by the equation:

$$\mathcal{U}(n) = \mathcal{U}(o) - 2J\Delta.$$

Cluster Algorithm(5)

Consider the probability of the inverse process. One wishes to generate the same cluster structure, but one starts from a configuration where one has $N_p + \Delta$ parallel pairs and $N_a - \Delta$ antiparallel pairs. Antiparallel bonds are assumed broken. One wishes that the bond number n_c' is equal to n_c because one needs to have the same number of connected bonds for generating the same cluster. The difference with the previous configuration is the number of bonds to break. Indeed, one obtains

$$\begin{aligned} N_p(n) &= n_c' + n_b' \\ N_p(o) &= n_c + n_b \\ N_p(n) &= N_p(o) + \Delta \\ &= n_c + n_b + \Delta, \end{aligned}$$

which immediately gives

$$n_b' = n_b + \Delta.$$

Cluster Algorithm(6)

Going from the new bond configuration to the old configuration , the probability is given by

$$P_{gc}(n) = p^{n_c}(1-p)^{n_b+\Delta}$$

Let us recall that

$$\frac{P_{gc}(o)}{P_{gc}(n)} = \exp(-\beta(U(n) - U(o))) \tag{7}$$

$$(1-p)^{-\Delta} = \exp(2\beta J\Delta) \tag{8}$$

One can solve this equation for p, which gives

$$(1-p) = \exp(-2\beta J), \tag{9}$$

Finally, the probability p is given by

$$p = 1 - \exp(-2\beta J).$$

Therefore, if the probability is chosen as p new configuration is always accepted!

Cluster Algorithm(7)

- The virtue of this algorithm is not only its ideal acceptance rate, but one can also show that, in the vicinity of the critical point, the critical slowing down drastically decreases drastically.

  For instance, the critical exponent of the two-dimensional Ising model is equal to 2.1 with the Metropolis algorithm while it is 0.2 with a cluster algorithm.

Cluster Algorithm(7)

- The virtue of this algorithm is not only its ideal acceptance rate, but one can also show that, in the vicinity of the critical point, the critical slowing down drastically decreases drastically.

  For instance, the critical exponent of the two-dimensional Ising model is equal to 2.1 with the Metropolis algorithm while it is 0.2 with a cluster algorithm.

- Conversely, a cluster algorithm is slower than the Metropolis algorithm far from the critical point.

Cluster Algorithm(7)

- The virtue of this algorithm is not only its ideal acceptance rate, but one can also show that, in the vicinity of the critical point, the critical slowing down drastically decreases drastically.

  For instance, the critical exponent of the two-dimensional Ising model is equal to 2.1 with the Metropolis algorithm while it is 0.2 with a cluster algorithm.

- Conversely, a cluster algorithm is slower than the Metropolis algorithm far from the critical point.

- Two kinds of implementation

Cluster Algorithm(7)

- The virtue of this algorithm is not only its ideal acceptance rate, but one can also show that, in the vicinity of the critical point, the critical slowing down drastically decreases drastically.

  For instance, the critical exponent of the two-dimensional Ising model is equal to 2.1 with the Metropolis algorithm while it is 0.2 with a cluster algorithm.

- Conversely, a cluster algorithm is slower than the Metropolis algorithm far from the critical point.

- Two kinds of implementation

  1. In Swendsen-Wang algorithm, all clusters are built on the laatice and flipped

Cluster Algorithm(7)

- The virtue of this algorithm is not only its ideal acceptance rate, but one can also show that, in the vicinity of the critical point, the critical slowing down drastically decreases drastically.

  For instance, the critical exponent of the two-dimensional Ising model is equal to 2.1 with the Metropolis algorithm while it is 0.2 with a cluster algorithm.

- Conversely, a cluster algorithm is slower than the Metropolis algorithm far from the critical point.

- Two kinds of implementation

  1. In Swendsen-Wang algorithm, all clusters are built on the laatice and flipped
  2. In Wolf algorithm, a site is chosen randomly, and a cluster is built from this seed. Small cluster flips are avoided and the algorithm is generally faster than the Swenden-Wang algorithm.