Monte Carlo, importance sampling through Markov chain and simulated annealing

Let us introduce the concept of importance sampling method by application to classical many-particle system (like Ising model or classical gas).

The basic idea of Monte Carlo Simulation:

The simulation is performed by random walk through **very large** configuration space. The probability to make a move has to be such that the system gets to thermal equilibrium (and remains in thermal equilibrium) at certain temperature T (is usually a parameter) after a lot of Monte Carlo steps.

The basic idea of simulated annealing:

Slowly decreasing the temperature of the system leads to the ground state of the system. When using this type of cooling down by Monte Carlo, one can find a global minimum of a general minimization problem. This is called simulated annealing.

Monte Carlo importance sampling and Markov chain

If a configuration in phase space is denoted by X, the probability for configuration according to Boltzman is

$$\rho(X) \propto e^{-\beta E(X)} \qquad \beta = \frac{1}{T} \tag{1}$$

How to sample over the whole phase space for a general problem? How to generate configurations?

- **Brute force**: generate a truly random configuration X and accept it with probability $e^{-\beta E(X)}$ where all $E > 0$. Successive X are **statistically independent**. *VERY INEFFICIENT*
- **Markov chain**: Successive configurations X_i, X_{i+1} are **NOT** statistically independent but are distributed according to Boltzman distribution.

What is the difference between Markov chain and uncorrelated sequence?

- *Truly random* or *uncorrelated* sequence of configurations satisfies the identity

$$P(X_1, X_2, \cdots, P_{X_N}) = P_1(X_1)P_1(X_2)\cdots P_1(X_N)$$

Metropolis

- *Markov chain* satisfies the equation

  $$P(X_1, X_2, \cdots, P_{X_N}) = P_1(X_1)T(X_1 \to X_2)T(X_2 \to X_3)\cdots T(X_{N-1} \to X_N)$$

  where the transition probabilities $T(X \to X')$ are normalized

  $$\sum_{X'} T(X \to X') = 1$$

We want to generate Markov chain where distribution of states is proportional to $e^{-\beta E(X)}$ and this distribution should be independent of the position within the chain and independent of the initial configuration.

The necessary conditions for generating such Markov chain is that every configuration in phase space should be accesible from any other configuration within finite number of steps (*connectedness* or *irreducibility*) - (Be careful to check this condition when choosing Monte Carlo step!)

We need to find transition probability $T(X \to X')$ which leads to a given **stationary** distribution $\rho(X)$ (in this case $\rho(X) \propto e^{-\beta E(X)}$).

Metropolis

The probability for X decreases, if system goes from X to any other X': $-\sum_{X'} \rho(X)T(X \to X')$ and increases if X configuration is visited from any other state X': $\sum_{X'} \rho(X')T(X' \to X)$. The change of probability for X is therefore

$$\rho(X,t+1) - \rho(X,t) = -\sum_{X'} \rho(X)T(X \to X') + \sum_{X'} \rho(X')T(X' \to X) \quad (2)$$

We look for stationary solution, i.e., $\rho(X,t+1) - \rho(X,t) = 0$ and therefore

$$\sum_{X'} \rho(X)T(X \to X') = \sum_{X'} \rho(X')T(X' \to X) \quad (3)$$

General solution of this equation is not accesible, but a particular solution is obvious

$$\rho(X)T(X \to X') = \rho(X')T(X' \to X) \quad (4)$$

This solution is called **DETAIL BALANCE** solution.

Metropolis

To construct algorithm, we devide transition prob. $T(X \to X') = \omega_{XX'} A_{XX'}$:

- *trial step probability* $\omega_{XX'}$ which is symmetric, i.e., $\omega_{XX'} = \omega_{X'X}$ (for example spin flip in ising: $\omega_{XX'}$ is $1/L^2$ if X and X' differ for a single spin flip and zero otherwise) and
- *acceptance probability* $A_{XX'}$ (for example accepting of rejecting new configuration with probability proportional to $min(1, \exp(-\beta(E(X') - E(X))))$).

Detail balance condition becomes

$$\frac{\rho(X')}{\rho(X)} = \frac{A_{XX'}}{A_{X'X}}$$

Metropolis chooses

$$\begin{aligned} A_{XX'} &= 1 && \text{if } \rho(X') > \rho(X) \\ A_{XX'} &= \frac{\rho(X')}{\rho(X)} && \text{if } \rho(X') < \rho(X). \end{aligned} \qquad (5)$$

Obviously, this acceptance probability satisfies detail balance condition and therefore leads to desired Markov chain with stationary probability for any configuration $X \propto \rho(X)$ for long times.

Metropolis

To summarize Metropolis algorithm

- $T(X \to X') = \omega_{XX'} A_{XX'}$
- $\sum_{X'} \omega_{XX'} = 1; \omega_{XX'} = \omega_{X'X}$
- $\omega_{XX'} > 0$ for all X, X' after finite number of steps
- $A_{XX'} = min(1, \frac{\rho(X')}{\rho(X)})$

How to accept a step with probability $A_{XX'} < 1$? One can generate a random number $r \in [0, 1]$ and accept the step if $r < A_{XX'}$.

Keep in mind:

- Configurations that are generated by Markov chain are correlated. The theory guarantees that we arrive at invariant distribution ρ for long times.
- Two configurations are statistically independent only if they are far apart in the Markov chain. This distance is called *correlation time* (*Be careful: To meassure distance in Markov chain, every step counts, not only successful.*)

The average of any quantity can be calculated as usual

$$\overline{A} = \frac{1}{n-n_0}\sum_{i>n_0}^{n} A_i$$

where n_0 steps are used to ”warm-up”.

The error of the quantity, however, is much bigger than the following quantity

$$\frac{1}{n-n_0}\sum_{i>n_0}^{n}(A_i-\overline{A})^2$$

Imagine the extreme limit of correlations when all values A_i are the same. We would estimate that standard deviation is zero regardless of the actual error!

To compute standard deviation, we need to group meassurements within the correlation time into bins and than estimate the standard deviation of the bins:

$$B_l = \frac{1}{N_0}\sum_{i=N_l}^{i<N_l+N_0} A_i \tag{6}$$

Metropolis

$$\sigma^2 = \frac{1}{M}\sum_{j=0}^{M-1}(B_j - \overline{A})^2 \tag{7}$$

where we took into account that $\overline{A} = \overline{B}$. The correlation time (here denoted by N_0) is not very easy to estimate. Maybe the best algorithm is to compute σ^2 for few different N_0 and as long as σ^2 is increasing with N_0, the correlation time is still larger than N_0. When σ^2 stops changing with increasing N_0, we reached correlation time and σ^2 is a good estimation of standard deviation.

Example: Ising Model

Below we sketch algorithm for the two dimensional Ising model

$$H = -\frac{1}{2}\sum_{ij} J_{ij} S_i S_j. \tag{8}$$

We will take $J_{ij} = 1$ for the nearest neighbors and 0 otherwise. S_i can take the values ± 1.

On an $L \times L$ lattice, we can choose

$$\omega_{XX'} = 1/L^2$$

if X and X' differ by one spin and $\omega_{XX'} = 0$ otherwise. This can be realized by selecting a spin at random and try to flip it (X' differs from X by single spin flip).

The energy diference $\Delta E = E(X') - E(X)$ in this case is cheap to calculate because it depends only on the nearest neighbour bonds. If $\Delta E < 0$ the trial state is accepted and if $\Delta E > 0$, the trial step is accepted with probability $\exp(-\beta \Delta E)$.

Keep in min

- One can keep track of total energy and total magnetization at every step. The total energy and magnetization needs to be computed from the scratch only at the beginning. After that they can be updated (add difference due to spin flip).

- the quantity $\exp(-\beta\Delta E)$ takes only 5 different values at fixed temperature. It is advisable to store those five numbers and not recompute them at every step.
- Simulation can be started with random configuration corresponding to infinite temperature. The temperature can be slowly decreased through transition which is around $\beta J \approx 0.44$.

Metropolis

Algorithm

- Choose temperature T and precompute the exponents $[\exp(-8J/T), \exp(-4J/T), 1, \exp(4J/T), \exp(8J/T)]$.
- Generate a random configuration X_0 (this is equivalent to infinite temperature).
- Compute the energy and magnetization of the configuration X_0.1
- Iterate the following steps many times
  - Select randomly a spin and compute the Weiss-Field felt by that spin, i.e., $W_i = \sum_{j\ neighbor\ i} J_{ij} S_j$. The energy cost to flip the spin is $\Delta E = 2 S_i W_i$. Use the precomputed exponents to evaluate acceptance probability $P = min(1, \exp(-\Delta E/T))$.
  - Accept the trial step with probability P. If accepted, update spin lattice configuration, update the current energy ($E = E + \Delta E$), and the current magnetization $M = M - 2S_i$.
  - Ones the Markow chain equilibrated, meassure the total energy and magnetization (also E^2, and M^2) every few Monte Carlo steps. ***Be carefull:*** *Do not meassure only the accepted steps. Every step counts. Meassure outside the acceptance loop.*

Metropolis

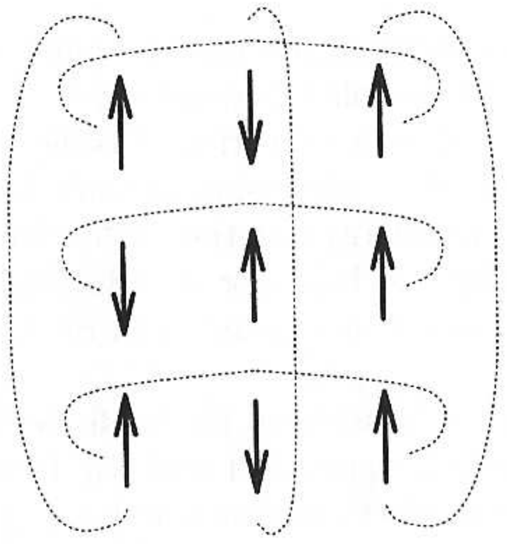
- Print the following quantities:
  - $\langle E \rangle$, $\langle M \rangle$
  - $c_V = (\langle E^2 \rangle - \langle E \rangle^2)/T^2$
  - $\chi = (\langle M^2 \rangle - \langle M \rangle^2)/T$

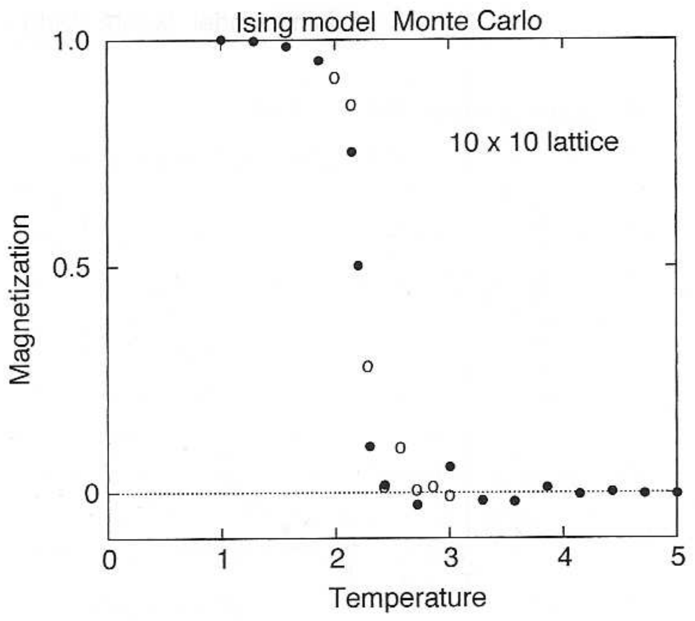
The relation for specific heat can be derived from the following identity

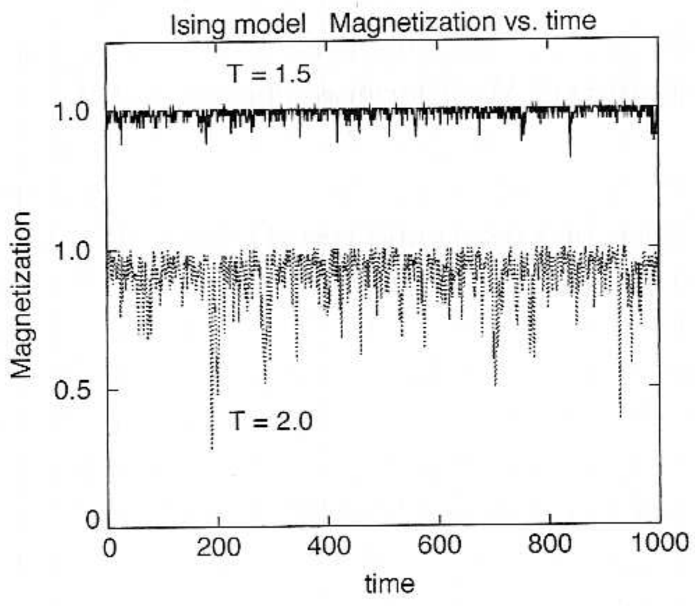
$$C_v = \frac{\partial \langle E \rangle}{\partial T} = \frac{\partial}{\partial T}\left(\frac{\mathrm{Tr}(e^{-H/T}E)}{\mathrm{Tr}(e^{-H/T})}\right) \tag{9}$$

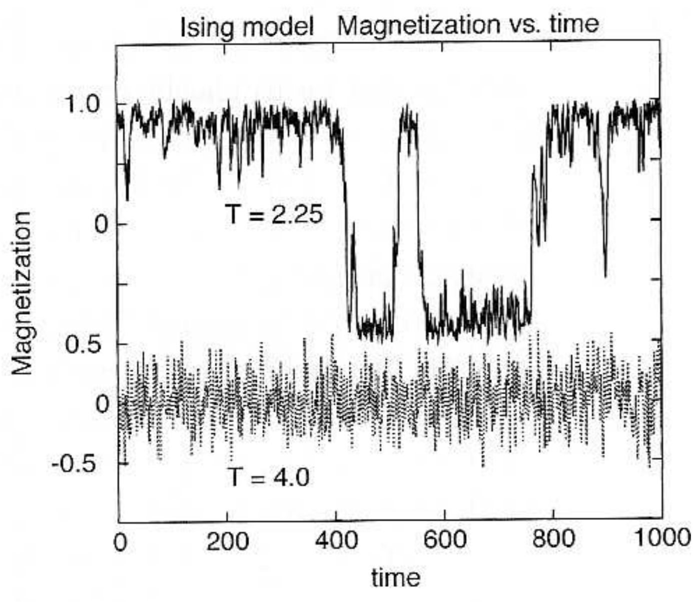
and similar equation exists for χ.

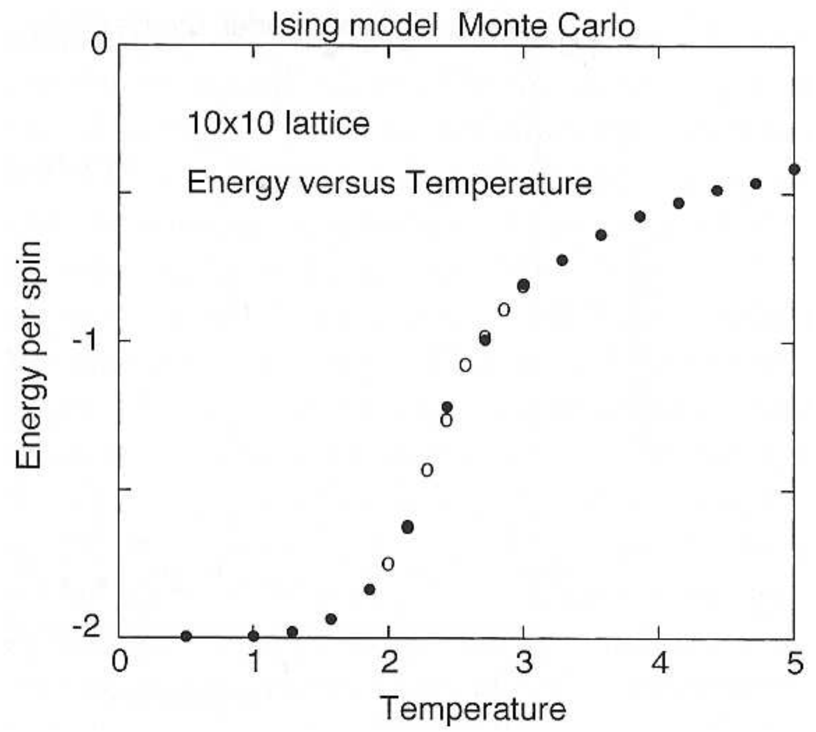
Metropolis

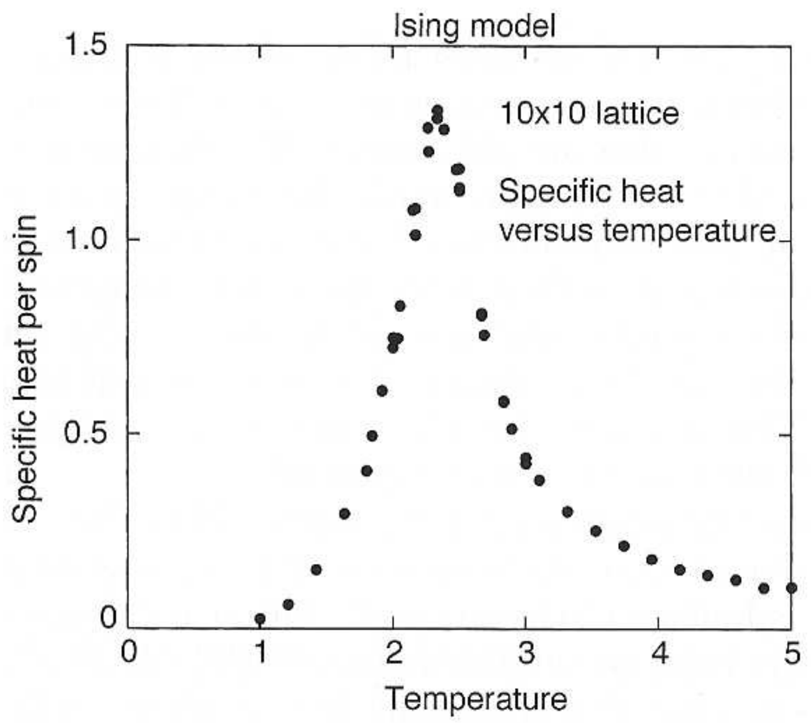



Ising model Monte Carlo
10 x 10 lattice
Magnetization
Temperature
1.0
0.5
0
0
1
2
3
4
5

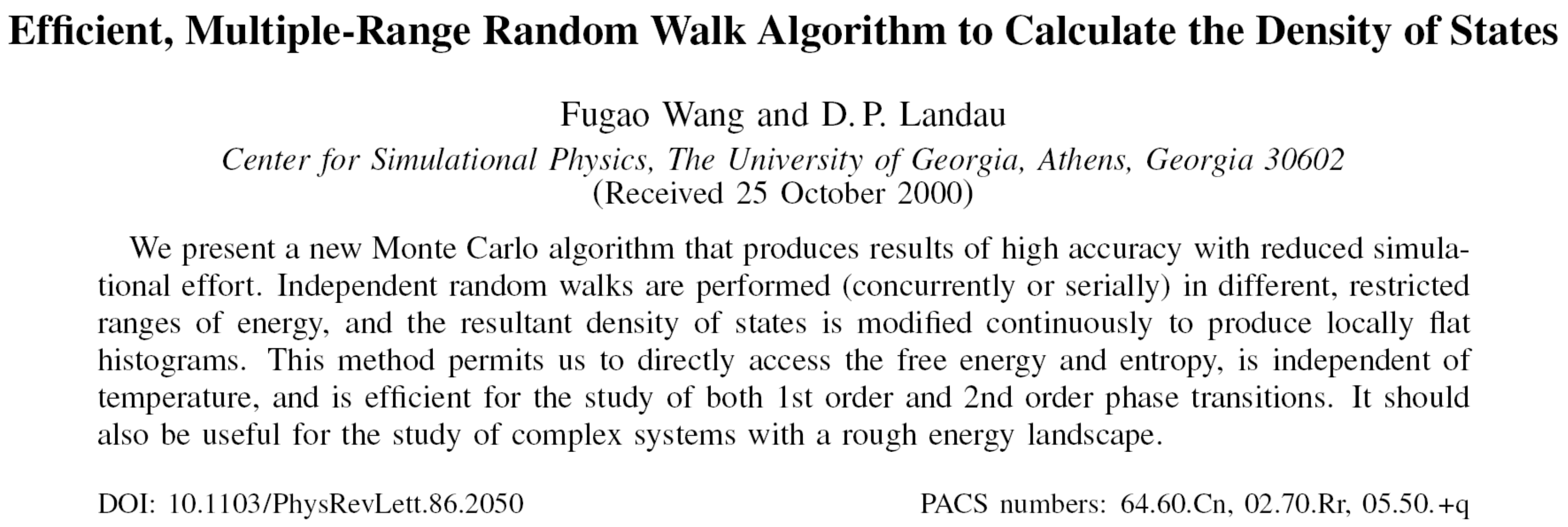

Ising model Magnetization vs. time
T = 1.5
T = 2.0
Magnetization
time
1.0
1.0
0.5
0
0
200
400
600
800
1000


Ising model Magnetization vs. time
T = 2.25
T = 4.0
Magnetization
time
1.0
0
0.5
0
-0.5
0
200
400
600
800
1000


Ising model Monte Carlo
10x10 lattice
Energy versus Temperature
Energy per spin
Temperature
0
-1
-2
0
1
2
3
4
5


Ising model
10x10 lattice
Specific heat versus temperature
Specific heat per spin
Temperature
1.5
1.0
0.5
0
0
1
2
3
4
5

Wang-Landau Method



Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States

Fugao Wang and D. P. Landau

Center for Simulational Physics, The University of Georgia, Athens, Georgia 30602

(Received 25 October 2000)

We present a new Monte Carlo algorithm that produces results of high accuracy with reduced simulational effort. Independent random walks are performed (concurrently or serially) in different, restricted ranges of energy, and the resultant density of states is modified continuously to produce locally flat histograms. This method permits us to directly access the free energy and entropy, is independent of temperature, and is efficient for the study of both 1st order and 2nd order phase transitions. It should also be useful for the study of complex systems with a rough energy landscape.

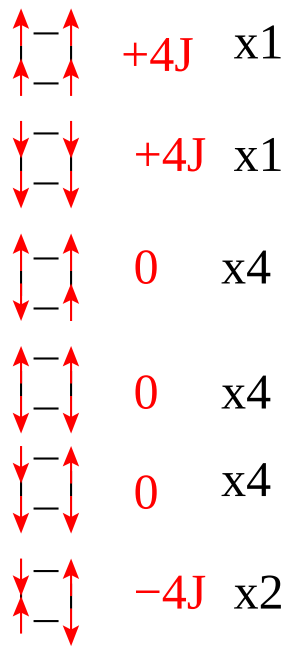
DOI: 10.1103/PhysRevLett.86.2050 PACS numbers: 64.60.Cn, 02.70.Rr, 05.50.+q

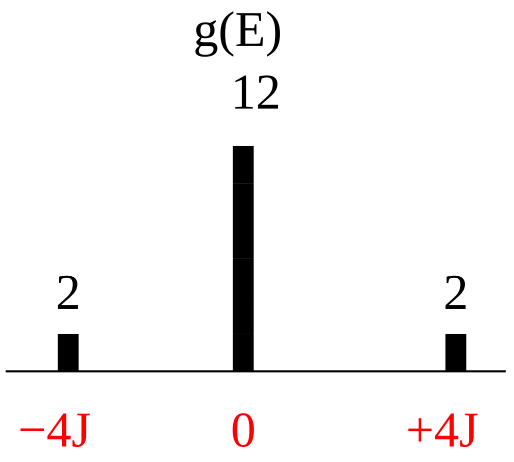
The algorithm is very useful for studing the phase transition phenomena because it does not suffer from the critical slowing down and gives direct access to thermodynamic potential and its derivatives

Metropolis

Consider the 4-site Ising model.

H=J S_iS_j





Many energy levels have very high degeneracy and some are only twice-degenerate.

In case of periodic boundary condistions, the allowed energy levels of the ising model are: $-2JN$, $-2J(N-4)$, $-2J(N-6)$,$-2J(N-8)$ ... $2JN$. The number of energy levels is of the order of N while the number of all possible states is 2^N. Therefore many states have to be heavily degenerate. In Markov-chain simulation at certain low temperature T, most of states visited have the same energy close to $\langle E\rangle$ (the probability for a state is $e^{-E/T}$).

The classical MC simulation generates canonical distribution at a given temperature $P(X_E) \propto g(E)e^{-E/T}$. The idea of Wang-Landau is to estimate the many-body density of states $g(E)$ directly. The temperature is not required for the simulation and by analyzing $g(E)$ one can study thermodynamics at *any* temperature.

If the density of state $g(E)$ is known, the free energy is simply given by

$$F(T) = -k_B T \log(Z) = -k_B T \log \sum_E g(E) e^{-E/T} \tag{10}$$

In the Wang-Landau algorithm the random walk in energy space is performed and the probability to visit a state with energy E is taken to be $P \propto \frac{1}{g(E)}$. The resulting histogram $H(E) \propto g(E)P(E) \propto const$ is therefore flat in the simulation.

In our example above, a simple random walk would visit a state with $E = 0$ six-times more often than the other two states since there are six-times more states at $E = 0$. If we assign probability for the state at $E = -4J$ to be proportional to $1/2$, at $E = 0$ to $1/12$ and at $E = 4J$ to be $1/2$, we will visit all three states on average the same number of times. Hence the histogram of visits is flat.

Metropolis

Let us sketch of the algorithm which generates the density of states $g(E)$.

The transition probability is taken to be

$$P(E_1 \to E_2) = min\left[\frac{g(E_1)}{g(E_2)}, 1\right]. \tag{11}$$

We always accept moves with smaller density of states but we often reject moves that would bring us to the state with high density of states.

- Start with $g(E) = 1$ for all energies of the model.
- Try to flip a random spin and accept the move according to probability (11).
- Every time we accept a move, the density of states for the current energy is updated by multiplying the existing value $g(E)$ by a modification factor $f > 1$, i.e.,
  $g(E) \to g(E)f$.
  This modification factor serves like a temperature in the simulated annealing algorithm. At the beginning it is taken to be relatively large $f \sim e^1 = 2.71828$ and later it is reduced to unity. Because of this large factor, the density of states will grow very fast for highly degenerate states and we will quickly visit all possible energies. However, the density of states obtained by large f is not very accurate. The error is equal to $\log(f)$.

Metropolis

- The simulation at certain modification factor f is running for so long that the histogram becomes flat, i.e., each energy level was visited equal number of times. Here equal means that the deviation is within certain bound, like 80-90%.
- At that point, the histogram is reset to zero $H(E) = 0$ and a new random walk is started with finer modification factor (like $f_1 = \sqrt{f_0}$).
- The simulation is again running so long that the histogram becomes flat. After that, the histogram is again reset and modification factor is reduced.
- When the modification factor f becomes sufficiently close to unity, the density of states does not change anymore. **If we are able to obtain a flat histogram using converged $g(E)$ in Eq. (11), the density of states $g(E)$ is obviously the true density of states of the model.**

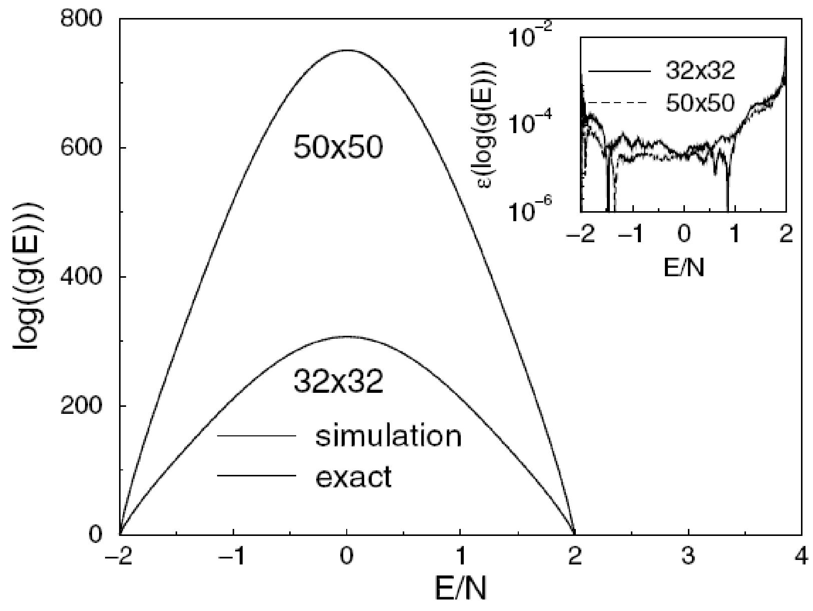


FIG. 1. Comparison of the density of states obtained by our algorithm for 2D Ising model and the exact results calculated by the method in Ref. [13]. Relative errors $\varepsilon(\log(g(E)))$ are shown in the inset.

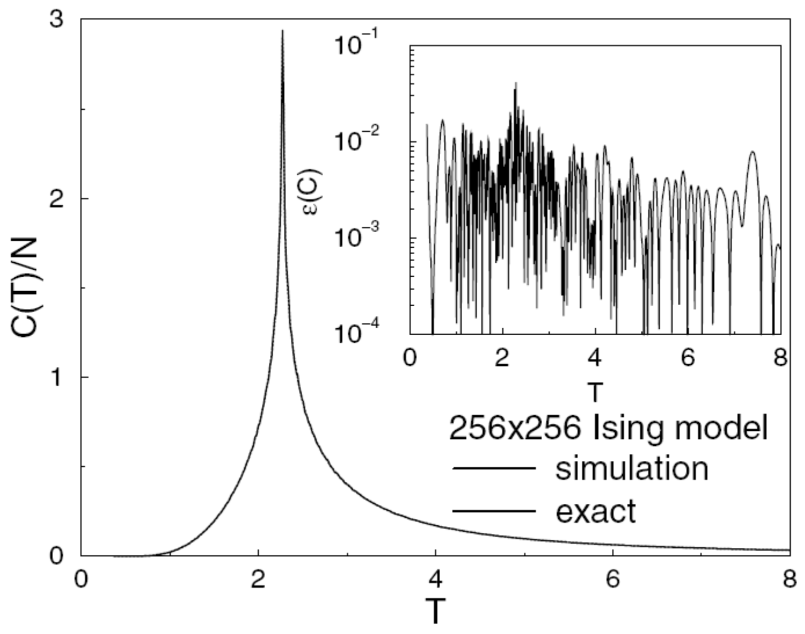


FIG. 3. Specific heat for the 2D Ising model on a 256×256 lattice in a wide temperature region. The relative error $\epsilon(C)$ is shown in the inset in the figure.

Simulated annealing

Is a technique which uses Matropolis to find a global minimum of a function of many variables. It is typically used for large scale problems, especially ones where a desired global minimum is hidden among many poorer local minima.

The class of problems known as NP-complete problems, whose computation time for an exact solution increases with N as $\exp(const.N)$, were basically unsolvable until recently.

Only in 1983, the method was invented (Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. 1983, Science, vol. **220**, 671-680 (1083).) which effectively solved most of these problems for practical purposes.

Simulated annealing does not ”solve” them exactly, but in most cases we just need reasonable good solution even if its not the one with exacly the lowest total energy.

The archtype example of NP-complete problems, the traveling salesman problem, is easy to solve with this method.

Other very important application is in designing complex integrated circuits: The arrangement of several hundred thousand circuit elements on a tiny silicon substrate is optimized so as to minimize interference among their connecting wires.

The simulated annealing uses Metropolis algorithm with slowly decreasing temperature in order that system relaxes into its ”ground state”.

The steps consis of

- Pick initial configuration X and an initial temperature T_0. T_0 should be much higher than the changes in function f to be minimized when typical Monte Carlo step is taken.
- Loop through decreasing temperature T_i
  - Equilibrate at T_i using Metropolis with selected, allowed elementary changes in the system. The system is equilibrated when the rate of change of f averaged over some number of Monte Carlo steps is small.
  - Measure the termal average of f. If f does not decrease at this temperature T_i compared to previous temperature T_{i-1}, exit the loop.
  - Decrease T_i to T_{i+1}. For example, reduces the effective temperature by 10%.

The major difficulty (art) in implementation of the algorithm is that there is no obvious analogy for the temperature T with respect to a free parameter in the combinatorial problem. Furthermore, avoidance of entrainment in local minima (quenching) is dependent on the ”annealing schedule”, the choice of initial temperature, how many iterations are performed at each temperature, and how much the temperature is decremented at each step as cooling proceeds.

Example: The Traveling Salesman problem

The seller visits N cities ($i = 0...N-1$) with given positions $\mathbf{R}_i$, returning finally to his or her city of origin. *Each city* is to be visited *only once*, and the route is to be made as *short* as possible. This problem belongs to a class known as *NP-complete* problems, whose computation time for an exact solution increases with N exponentially.

The traveling salesman problem also belongs to a class of minimization problems for which the objective function E has many local minima. In practical cases, it is often enough to be able to choose from these a minimum which, even if not absolute, cannot be significantly improved upon.

The annealing method manages to achieve this, while limiting its calculations to scale as a small power of N.

Metropolis

As a problem in simulated annealing, the traveling salesman problem is handled as follows:

1. *Configuration:* The cities are numbered $i = 0...N-1$ and each has coordinate $\mathbf{R}_i$. A configuration is a permutation of the number $0...N-1$, interpreted as the order in which the cities are visited.
2. *Rearrangements:* An efficient set of moves are:
   - **(a)** A section of path is removed and then replaced with the same cities running in the opposite order; or
   - **(b)** a section of path is removed and then replaced in between two cities on another, randomly chosen, part of the path.
3. *Objective Function:* In the simplest form of the problem, E is taken just as the total length of journey

The above two mentioned moves are hardest to implement. The following plot explains them in an example

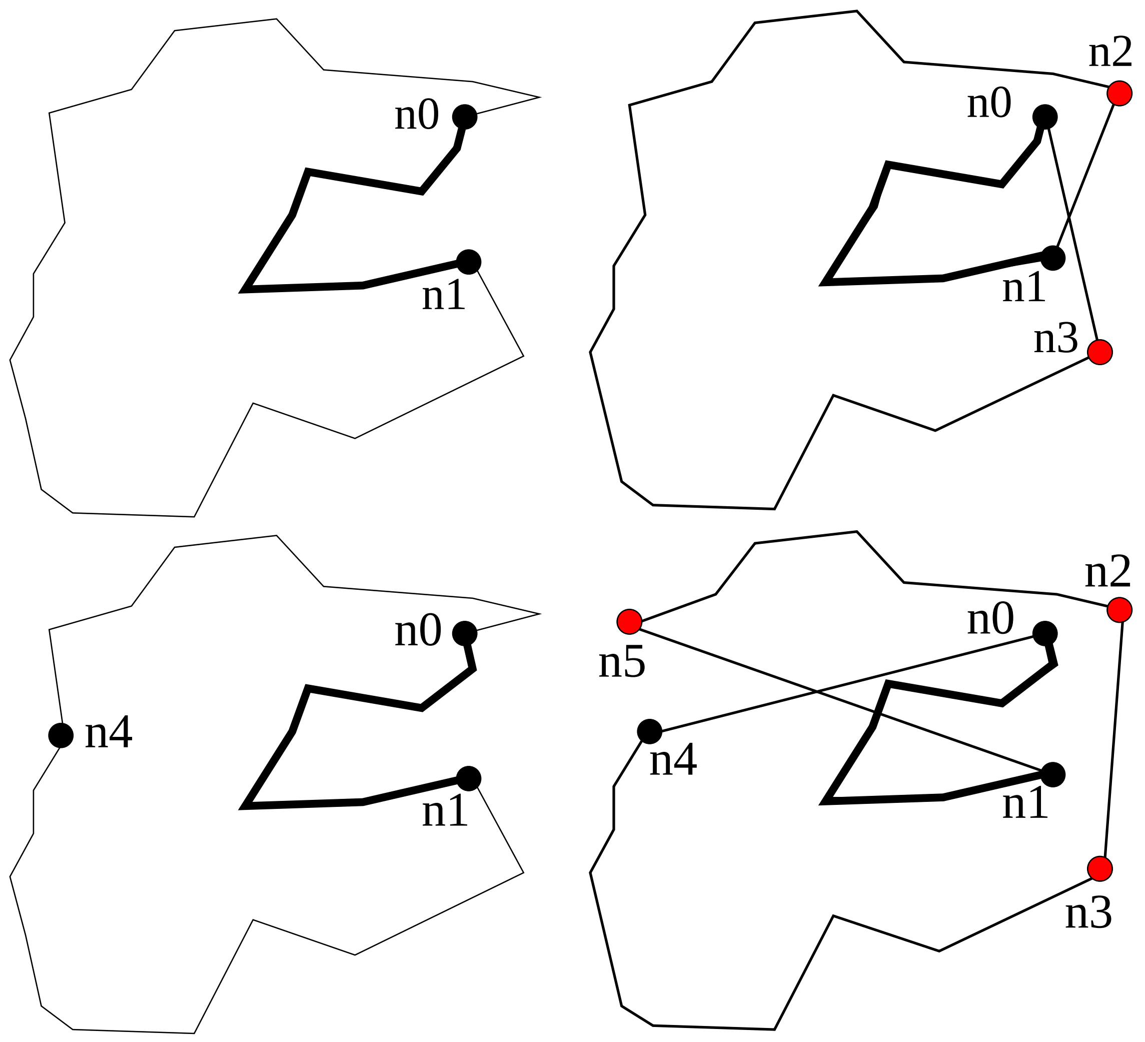


Figure 1: top: reverse move, bottom: transpose move

For details of implementation, check the example source code.

Metropolis

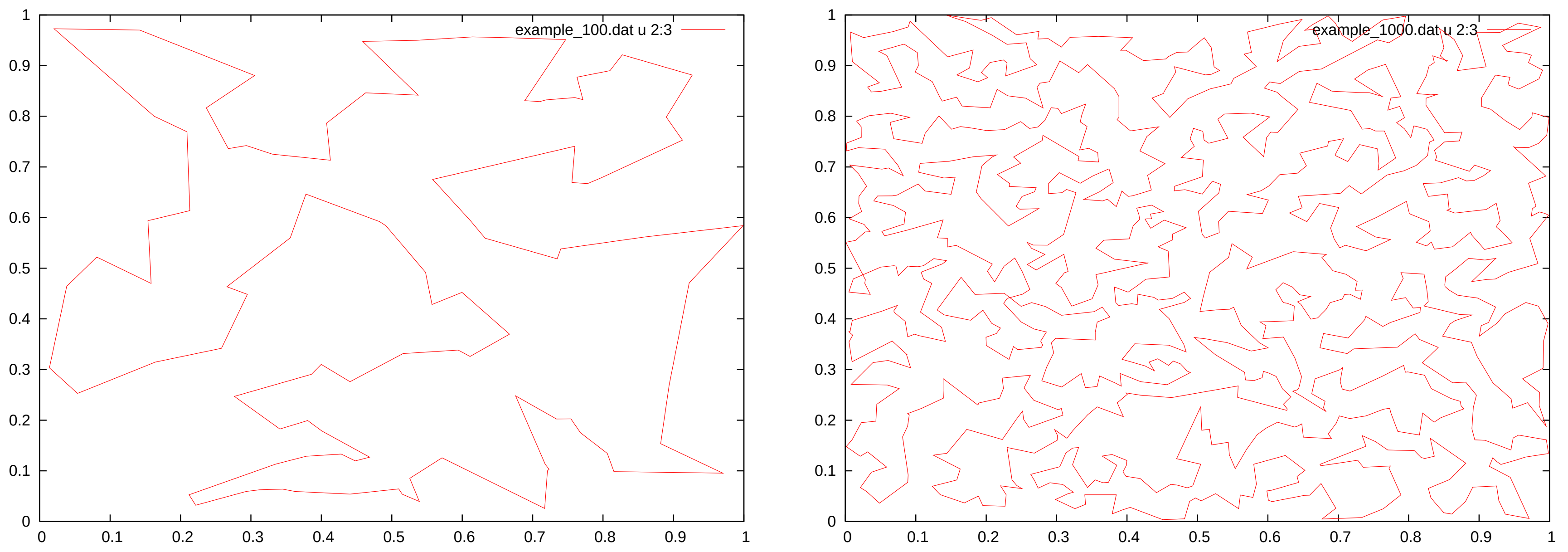


Figure 2: Example of the Traveling Salesman problem with 100 and 1000 cities redistributed randomly on a square between $([0,1],[0,1])$.

Homework 1

- Finish the ising model simulation.
- Finish the Wang-Landau algorithm for Ising model. Compute energy and specific heat for a 10×10 and 20×20 lattice as a function of temperature.