Lecture 9. Multilevel sampling and KMC

Tiejun Li[1,2]

[1]School of Mathematical Sciences (SMS),
&
[2]Center for Machine Learning Research (CMLR),
Peking University,
Beijing 100871,
P.R. China
tieli@pku.edu.cn

Office: No. 1 Science Building, Room 1376E

Table of Contents

Swendsen-Wang algorithm

Simulated tempering

Parallel tempering

Kinetic Monte Carlo

Gibbs sampling is not always efficient

- For the numerical computation of the Ising model, a commonly used approach is the Gibbs sampling to flip a single site at each step.

[1] R.H. Swendsen and J.S. Wang, Phys. Rev. Lett. 58 (1987), 86-88.

Gibbs sampling is not always efficient

- For the numerical computation of the Ising model, a commonly used approach is the Gibbs sampling to flip a single site at each step.
- However, the correlation length tends to infinity when the temperature $T \to T_c$. In this case, the single-flip proposal is usually rejected due to the low temperature.

[1] R.H. Swendsen and J.S. Wang, Phys. Rev. Lett. 58 (1987), 86-88.

Gibbs sampling is not always efficient

- For the numerical computation of the Ising model, a commonly used approach is the Gibbs sampling to flip a single site at each step.
- However, the correlation length tends to infinity when the temperature $T \to T_c$. In this case, the single-flip proposal is usually rejected due to the low temperature.
- Swendsen and Wang introduced a powerful clustering algorithm which together with an implementation modification by Wolff, almost completely eliminates the critical slowing down. [1]

[1] R.H. Swendsen and J.S. Wang, Phys. Rev. Lett. 58 (1987), 86-88.

Ising Lattice Configuration at Different Temperature

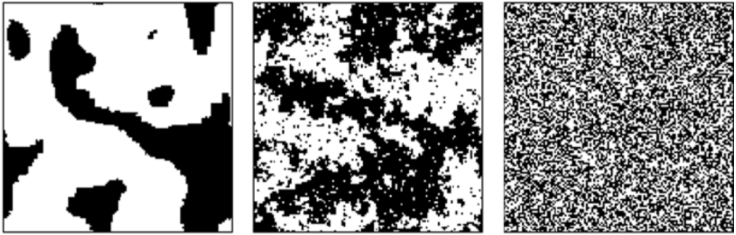

Figure: Ising lattice configuration at different temperature. Leftmost: $T \ll T_c$, Middle: $T = T_c$, Rightmost: $T \gg T_c$

Swendsen-Wang algorithm

- Below explanation to Swendsen-Wang algorithm is from data augmentation viewpoint by Higdon. [2]

[2] D.M. Higdon, J. Amer. Stat. Assoc. 93 (1998), 585-595.

Swendsen-Wang algorithm

- Below explanation to Swendsen-Wang algorithm is from data augmentation viewpoint by Higdon. [2]
- Only consider the case $h = 0$. We have the Gibbs distribution for Ising model

$$\begin{aligned}\pi(\boldsymbol{x}) &\propto \exp\left\{\beta J \sum_{<i,j>} x_i x_j\right\} \\ &\propto \prod_{<i,j>} \exp\left\{\beta J(1 + x_i x_j)\right\}.\end{aligned}$$

[2] D.M. Higdon, J. Amer. Stat. Assoc. 93 (1998), 585-595.

Swendsen-Wang algorithm

- Below explanation to Swendsen-Wang algorithm is from data augmentation viewpoint by Higdon. [2]
- Only consider the case $h = 0$. We have the Gibbs distribution for Ising model

$$\begin{aligned} \pi(\boldsymbol{x}) &\propto \exp\left\{\beta J \sum_{<i,j>} x_i x_j\right\} \\ &\propto \prod_{<i,j>} \exp\left\{\beta J(1 + x_i x_j)\right\}. \end{aligned}$$

- Note that $1 + x_i x_j$ is equal to either 0 or 2. Hence if we introduce an auxiliary variable $\boldsymbol{u}$ on each edge such that

$$\pi(\boldsymbol{x}, \boldsymbol{u}) \propto \prod_{<i,j>} I\Big[0 \le u_{ij} \le \exp\{\beta J(1 + x_i x_j)\}\Big].$$

Then the marginal distribution of $\boldsymbol{x}$ is the Gibbs distribution.

[2] D.M. Higdon, J. Amer. Stat. Assoc. 93 (1998), 585-595.

Swendsen-Wang algorithm: Conditional sampling

Recall that

$$\pi(\boldsymbol{x}, \boldsymbol{u}) \propto \prod_{<i,j>} I\Big[0 \leq u_{ij} \leq \exp\{\beta J(1 + x_i x_j)\}\Big].$$

Under this joint distribution:

- The conditional distribution $\boldsymbol{u}|\boldsymbol{x}$ is a product of uniform distributions with ranges depending on two neighboring spins.

Swendsen-Wang algorithm: Conditional sampling

Recall that

$$\pi(\boldsymbol{x}, \boldsymbol{u}) \propto \prod_{<i,j>} I\Big[0 \leq u_{ij} \leq \exp\{\beta J(1 + x_i x_j)\}\Big].$$

Under this joint distribution:

- The conditional distribution $\boldsymbol{u}|\boldsymbol{x}$ is a product of uniform distributions with ranges depending on two neighboring spins.
- Conversely, the conditional distribution $\boldsymbol{x}|\boldsymbol{u}$ is: if $u_{ij} > 1$, then $x_i = x_j$; otherwise there is no constraint on x_i's.

Swendsen-Wang algorithm: Conditional sampling

Recall that

$$\pi(\boldsymbol{x}, \boldsymbol{u}) \propto \prod_{<i,j>} I\Big[0 \le u_{ij} \le \exp\{\beta J(1 + x_i x_j)\}\Big].$$

Under this joint distribution:

- The conditional distribution $\boldsymbol{u}|\boldsymbol{x}$ is a product of uniform distributions with ranges depending on two neighboring spins.
- Conversely, the conditional distribution $\boldsymbol{x}|\boldsymbol{u}$ is: if $u_{ij} > 1$, then $x_i = x_j$; otherwise there is no constraint on x_i's.
- Thus $\boldsymbol{u}$ affects $\boldsymbol{x}$ only through the event $I[u_{ij} > 1]$.Based on the configuration $\boldsymbol{u}$, we cluster those lattice sites according to whether they have a mutual bond $(u_{ij} > 1)$.

Swendsen-Wang algorithm: Conditional sampling

Recall that

$$\pi(\boldsymbol{x}, \boldsymbol{u}) \propto \prod_{<i,j>} I\Big[0 \leq u_{ij} \leq \exp\{\beta J(1 + x_i x_j)\}\Big].$$

Under this joint distribution:

- The conditional distribution $\boldsymbol{u}|\boldsymbol{x}$ is a product of uniform distributions with ranges depending on two neighboring spins.
- Conversely, the conditional distribution $\boldsymbol{x}|\boldsymbol{u}$ is: if $u_{ij} > 1$, then $x_i = x_j$; otherwise there is no constraint on x_i's.
- Thus $\boldsymbol{u}$ affects $\boldsymbol{x}$ only through the event $I[u_{ij} > 1]$.Based on the configuration $\boldsymbol{u}$, we cluster those lattice sites according to whether they have a mutual bond ($u_{ij} > 1$).

Swendsen-Wang algorithm: Configurations

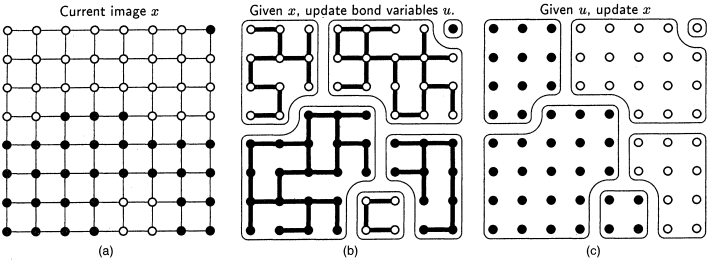

Figure 2. The Swendsen–Wang Algorithm for the Ising Model on the 8 × 8 Lattice. (a) Initial image x and Markov random field graph for $\pi(x)$. (b) Given the current image x, the bond variables u are generated uniformly over the interval $(0, e^{\beta I[x_i = x_j]})$. If $u_{ij} > 1$ (marked by the thick lines), x_i is constrained to equal x_j. These constraints partition the image into clusters of like-colored sites. Clusters induced by this realization of $u|x$ are outlined. The Markov random field graph for $x|u$ differs from that of x, marginally; the auxiliary vector u strengthens the dependence between some neighboring sites, while completely removing it from others. (c) Given the bond variables u, x is now a coarse image of independent clusters. Because there is no external field in this example, each cluster is recolored black or white with probability .5.

Swendsen-Wang Algorithm

Swendsen-Wang Algorithm (Alternating conditional sampling):

- Step 1: Sample $\boldsymbol{u}|\boldsymbol{x}$. For a given configuration of the spins, form the bond variable by giving every edge of the lattice $< i, j >$, between two "like spins" $(x_i = x_j)$ a bond value of 1 with probability $1 - \exp(-2\beta J)$, and a bond value of 0 otherwise.

Further extension to Wolff's modification can be referred to [3].

[3] U. Wolff, Phys. Rev. Lett. 62 (1989), 361-364.

Swendsen-Wang Algorithm

Swendsen-Wang Algorithm (Alternating conditional sampling):

- Step 1: Sample $\boldsymbol{u}|\boldsymbol{x}$. For a given configuration of the spins, form the bond variable by giving every edge of the lattice $< i, j >$, between two "like spins" $(x_i = x_j)$ a bond value of 1 with probability $1 - \exp(-2\beta J)$, and a bond value of 0 otherwise.
- Step 2: Sample $\boldsymbol{x}|\boldsymbol{u}$. Conditional on the bond variable $\boldsymbol{u}$, update the spin variable $\boldsymbol{x}$ by drawing from $\pi(\boldsymbol{x}|\boldsymbol{u})$, which is uniform on all compatible spin configurations; that is, clusters are produced by connecting neighboring sites with bond value 1. Each cluster is flipped with probability 0.5.

Further extension to Wolff's modification can be referred to [3].

[3] U. Wolff, Phys. Rev. Lett. 62 (1989), 361-364.

Table of Contents

Swendsen-Wang algorithm

Simulated tempering

Parallel tempering

Kinetic Monte Carlo

Simulated tempering

Problem: To sample the distribution

$$p(x) \propto \exp\Big(-\frac{U(x)}{T}\Big)$$

or compute the ensemble average with the type

$$\langle H\rangle = \int H(x)\frac{1}{Z}\exp\Big(-\frac{U(x)}{T}\Big)dx,$$

- one usually apply the Metropolis-Hastings MCMC algorithm.

Simulated tempering

Problem: To sample the distribution

$$p(x) \propto \exp\Big(-\frac{U(x)}{T}\Big)$$

or compute the ensemble average with the type

$$\langle H \rangle = \int H(x)\frac{1}{Z}\exp\Big(-\frac{U(x)}{T}\Big)dx,$$

- one usually apply the Metropolis-Hastings MCMC algorithm.
- But when the temperature T is very low, that is, we have many high peaks in the pdf $p(x)$, which may cause the acceptance probability small thus decrease the mixing.

Tempering idea

Strategy: In order to let a MCMC scheme move more freely in the sate space, Marinari and Parasi proposed a data augmentation strategy to increase the mixing, which is called simulated tempering. [4]

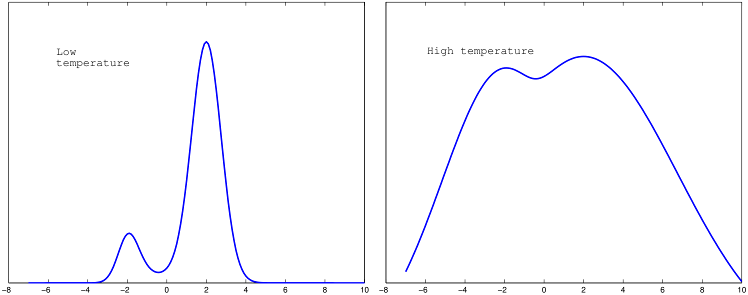

Figure: Sketch of the Gibbs distribution at low and high temperature.

[4] E. Marinari and G. Parisi, Europhys. Lett. 19 (1992), 451-458.

Simulated tempering

- Algorithmically, their basic idea is to extend the state space $x \in \mathcal{X}$ into $(x, i) \in \mathcal{X} \times I$ and perform conditional sampling in this extended space.

Simulated tempering

- Algorithmically, their basic idea is to extend the state space $x \in \mathcal{X}$ into $(x, i) \in \mathcal{X} \times I$ and perform conditional sampling in this extended space.
- Physically, to approach the low temperature case, they consider the pdf at the heated temperature, which can give high acceptance ratio for traversing the state space $\mathcal{X}$, and then jump in the different ensembles.

Simulated tempering

- Algorithmically, their basic idea is to extend the state space $x \in \mathcal{X}$ into $(x, i) \in \mathcal{X} \times I$ and perform conditional sampling in this extended space.
- Physically, to approach the low temperature case, they consider the pdf at the heated temperature, which can give high acceptance ratio for traversing the state space $\mathcal{X}$, and then jump in the different ensembles.
- Mathematically, choose

$$I = \{1, 2, \ldots, L\}, \quad T_1 < T_2 < \ldots < T_L$$

and $T_1 = T$, $T_L = T_{\mathsf{high}}$. Then we ask the stationary distribution in the extended space as

$$\pi_{\mathsf{st}}(x, i) \propto \pi_i \exp\Big(-\frac{U(x)}{T_i}\Big),$$

where π_i is called pseudopriors which is set up *a priori*.

Simulated tempering

From this form, we know the conditional distribution

$$f(x|i) \propto \exp\Big(-\frac{U(x)}{T_i}\Big)$$

which is the standard Gibbs distribution. The marginal distribution

$$f(i) \propto \int \pi_i \exp\Big(-\frac{U(x)}{T_i}\Big)dx = \pi_i Z_i.$$

To make the transition in different ensembles more uniformly, the best choice for the parameter $\pi_i \propto 1/Z_i$. But in the computations, it is not feasible and only updated with the time.

Algorithm: Simulated tempering

To do the conditional sampling in the extended space, we list a mixture-type transition kernel here.

Mixture type of the simulated tempering:

- Step 1. With the current state $(x_n, i_n) = (x, i)$, we draw $u \sim \mathcal{U}[0, 1]$.

Algorithm: Simulated tempering

To do the conditional sampling in the extended space, we list a mixture-type transition kernel here.

Mixture type of the simulated tempering:

- Step 1. With the current state $(x_n, i_n) = (x, i)$, we draw $u \sim \mathcal{U}[0, 1]$.
- Step 2. If $u < \alpha_0$, perform state updates. We let $i_{n+1} = i$ and let x_{n+1} be drawn from a MCMC transition $T_i(x, x_{n+1})$ that leaves $f(x|i)$ invariant (this is also Metropolis-Hastings strategy).

Algorithm: Simulated tempering

To do the conditional sampling in the extended space, we list a mixture-type transition kernel here.

Mixture type of the simulated tempering:

- Step 1. With the current state $(x_n, i_n) = (x, i)$, we draw $u \sim \mathcal{U}[0, 1]$.
- Step 2. If $u < \alpha_0$, perform state updates. We let $i_{n+1} = i$ and let x_{n+1} be drawn from a MCMC transition $T_i(x, x_{n+1})$ that leaves $f(x|i)$ invariant (this is also Metropolis-Hastings strategy).
- Step 3. If $u > \alpha_0$, perform temperature switching. We let $x_{n+1} = x$ and propose a level transtion $i \to j$, from a transition function $\alpha(i, j)$, and let $i_{n+1} = j$ with probability

  $$\min\left(1, \frac{\pi_{st}(x, j)\alpha(j, i)}{\pi_{st}(x, i)\alpha(i, j)}\right).$$

  Otherwise let $i_{n+1} = i$.

Strategy — Connection with Random Walk

- A commonly used strategy for $\alpha(i, j)$ is the random walk proposal with reflecting barrier, that is,

$$\alpha(i, i \pm 1) = 1/2, \quad i = 2, \ldots, L-1$$

and $\alpha(1, 2) = \alpha(L, L-1) = 1$.

Strategy — Connection with Random Walk

- A commonly used strategy for $\alpha(i,j)$ is the random walk proposal with reflecting barrier, that is,

  $$\alpha(i, i \pm 1) = 1/2, \quad i = 2, \ldots, L-1$$

  and $\alpha(1,2) = \alpha(L, L-1) = 1$.
- The idea of simulated tempering is further generalized by Liu and Sabatti into the so called “simulated sintering” scheme.

Table of Contents

Swendsen-Wang algorithm

Simulated tempering

Parallel tempering

Kinetic Monte Carlo

Parallel tempering

The parallel tempering is first proposed by Geyer in 1991. [5]

- Instead of augmenting $\mathcal{X}$ into $\mathcal{X} \times I$, Geyer suggested directly dealing with the product space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_L$, where the $\mathcal{X}_i$ are identical copies of $\mathcal{X}$, suppose

$$(x_1, \ldots, x_L) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_L,$$

[5] C.J. Geyer, Markov chain Monte Carlo maximum likelihood, in E. Keramigas (ed.), Interface Foundation, Fairfax, 156-163, 1991.

Parallel tempering

The parallel tempering is first proposed by Geyer in 1991. [5]

- Instead of augmenting $\mathcal{X}$ into $\mathcal{X} \times I$, Geyer suggested directly dealing with the product space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_L$, where the $\mathcal{X}_i$ are identical copies of $\mathcal{X}$, suppose

$$(x_1, \ldots, x_L) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_L,$$

- we define the stationary distribution

$$\pi_{st}(x_1, \ldots, x_L) = \prod_{i \in I} \pi_i(x_i)$$

where $\pi_i(x_i) = 1/Z_i \exp(-U(x_i)/T_i)$ the Gibbs distribution at $T = T_i$.

[5] C.J. Geyer, Markov chain Monte Carlo maximum likelihood, in E. Keramigas (ed.), Interface Foundation, Fairfax, 156-163, 1991.

Parallel tempering

The parallel tempering is first proposed by Geyer in 1991. [5]

- Instead of augmenting $\mathcal{X}$ into $\mathcal{X} \times I$, Geyer suggested directly dealing with the product space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_L$, where the $\mathcal{X}_i$ are identical copies of $\mathcal{X}$, suppose

$$(x_1, \ldots, x_L) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_L,$$

- we define the stationary distribution

$$\pi_{st}(x_1, \ldots, x_L) = \prod_{i \in I} \pi_i(x_i)$$

  where $\pi_i(x_i) = 1/Z_i \exp(-U(x_i)/T_i)$ the Gibbs distribution at $T = T_i$.
- The parallel tempering is run on all of the $\mathcal{X}_i$.

[5] C.J. Geyer, Markov chain Monte Carlo maximum likelihood, in E. Keramigas (ed.), Interface Foundation, Fairfax, 156-163, 1991.

Algorithm: Parallel tempering

An "index swapping" operation is conducted in place of the temperature transition. The algorithm is defined as follows:

- Step 1: Let the current state be $(x_1^{(n)}, \dots, x_L^{(n)})$. Draw $u \sim \mathcal{U}[0,1]$.

Algorithm: Parallel tempering

An "index swapping" operation is conducted in place of the temperature transition. The algorithm is defined as follows:

- Step 1: Let the current state be $(x_1^{(n)}, \ldots, x_L^{(n)})$. Draw $u \sim \mathcal{U}[0,1]$.
- Step 2: If $u \leq \alpha_0$, we conduct the parallel step. That is, we update each $x_i^{(n)}$ to $x_i^{(n+1)}$ via their respective MCMCM scheme.

Algorithm: Parallel tempering

An "index swapping" operation is conducted in place of the temperature transition. The algorithm is defined as follows:

- Step 1: Let the current state be $(x_1^{(n)}, \ldots, x_L^{(n)})$. Draw $u \sim \mathcal{U}[0, 1]$.
- Step 2: If $u \leq \alpha_0$, we conduct the parallel step. That is, we update each $x_i^{(n)}$ to $x_i^{(n+1)}$ via their respective MCMCM scheme.
- Step 3: If $u > \alpha_0$, we conduct the swapping step. That is, we randomly choose a neighboring pair, say i and $i + 1$, and propose "swapping" $x_i^{(n)}$ and $x_{i+1}^{(n)}$. Accept this swap with probability

$$\min\left\{1, \frac{\pi_i(x_{i+1}^{(n)})\pi_{i+1}(x_i^{(n)})}{\pi_i(x_i^{(n)})\pi_{i+1}(x_{i+1}^{(n)})}\right\}.$$

Algorithm: Parallel tempering

Choose a proper number of temperature levels

- In computations, $T_1 < T_2 < \ldots < T_L$, and it is very important to choose a proper number of temperature levels.

Algorithm: Parallel tempering

Choose a proper number of temperature levels

- In computations, $T_1 < T_2 < \ldots < T_L$, and it is very important to choose a proper number of temperature levels.
- A rough guideline is to choose T_i such that

$$\left(\frac{1}{T_i} - \frac{1}{T_{i+1}}\right)|\Delta U| \approx -\log p_a,$$

where $|\Delta U|$ is the typical energy difference (e.g., the mean energy change under the target distribution) and p_a is the lower bound for the acceptance rate. (The temperature difference can not be too large.)

Algorithm: Parallel tempering

Choose a proper number of temperature levels

- In computations, $T_1 < T_2 < \ldots < T_L$, and it is very important to choose a proper number of temperature levels.
- A rough guideline is to choose T_i such that

$$\left(\frac{1}{T_i} - \frac{1}{T_{i+1}}\right)|\Delta U| \approx -\log p_a,$$

  where $|\Delta U|$ is the typical energy difference (e.g., the mean energy change under the target distribution) and p_a is the lower bound for the acceptance rate. (The temperature difference can not be too large.)
- The rationale behind the choice of temperature T_i is to make the acceptance probability is relatively large since

$$\frac{\pi_i(x_{i+1}^{(n)})\pi_{i+1}(x_i^{(n)})}{\pi_i(x_i^{(n)})\pi_{i+1}(x_{i+1}^{(n)})} \sim \exp\left(-\left(\frac{1}{T_i} - \frac{1}{T_{i+1}}\right)\Delta U\right).$$

Table of Contents

Swendsen-Wang algorithm

Simulated tempering

Parallel tempering

Kinetic Monte Carlo

Drawbacks of standard MC

Kinetic Monte Carlo is also called BKL algorithm. [6]
It is widely used in simulating crystal growth.

- Drawbacks of standard MC:
  At the metastable state σ_m, suppose the proposal state is σ', then
  $$r = e^{-\beta \Delta H}, \qquad \Delta H = H(\sigma') - H(\sigma_m).$$
  If $r \ll 1$, rejection occurred very often! The sample sequence will be like
  $$\sigma_m, \sigma_m, \ldots, \sigma_m, \sigma_{new} \ldots.$$
  That's very inefficient!

[6] A.B. Bortz, M.H. Kalos and J.L. Lebowitz, J. Comp. Phys. 17(1975), 10-18.

Drawbacks of standard MC

Kinetic Monte Carlo is also called BKL algorithm. [6]
It is widely used in simulating crystal growth.

- Drawbacks of standard MC:
  At the metastable state σ_m, suppose the proposal state is σ', then

  $$r = e^{-\beta\Delta H}, \qquad \Delta H = H(\sigma') - H(\sigma_m).$$

  If $r \ll 1$, rejection occurred very often! The sample sequence will be like

  $$\sigma_m, \sigma_m, \ldots, \sigma_m, \sigma_{new} \ldots .$$

  That's very inefficient!
- KMC aims to setup a rejection free algorithm.

[6] A.B. Bortz, M.H. Kalos and J.L. Lebowitz, J. Comp. Phys. 17(1975), 10-18.

Kinetic Monte Carlo: Ten-fold way

- Generation of new state:
  Consider 2D Ising model: (ten-fold way in BKL algorithm)
  For a given state σ, there are 10 kinds of flips (single flip proposal):

Class	Spin	Number of spins up (nearest neighbors)
1	$\uparrow$	4
2	$\uparrow$	3
3	$\uparrow$	2
4	$\uparrow$	1
5	$\uparrow$	0
6	$\downarrow$	4
7	$\downarrow$	3
8	$\downarrow$	2
9	$\downarrow$	1
10	$\downarrow$	0

Table 1: Classification of spins in the 10-fold way

Kinetic Monte Carlo

There are 10 kinds of flipping probability

$$P_j = \min(1, \exp(-\beta \Delta H_j)),\ j = 1, \ldots, 10.$$

Kinetic Monte Carlo

There are 10 kinds of flipping probability

$$P_j = \min(1, \exp(-\beta \Delta H_j)),\ j = 1, \ldots, 10.$$

Suppose there are n_j sites at the jth class $j = 1, \ldots, 10.$ Define

$$Q_i = \sum_{j=1}^{i} n_j P_j, \quad i = 1, \ldots, 10,$$

Kinetic Monte Carlo

There are 10 kinds of flipping probability

$$P_j = \min(1, \exp(-\beta \Delta H_j)),\ j = 1, \dots, 10.$$

Suppose there are n_j sites at the jth class $j = 1, \dots, 10$. Define

$$Q_i = \sum_{j=1}^{i} n_j P_j, \quad i = 1, \dots, 10,$$

BKL Algorithm:

- Step 1. Generate $R \sim \mathcal{U}[0, Q_{10})$;
- Step 2. Identify $Q_{i-1} \leq R < Q_i, \quad (Q_0 = 0)$;
- Step 3. Randomly choose one site to flip in class i.

Kinetic Monte Carlo: Waiting time

Time increment between two flips.

- Suppose on the average there is one attempted flip per lattice site in time τ (physical time), ($\#$ of sites $= N = M^2$) then

$$\frac{Q_{10}}{N} : \text{ Probability of flipping for a spin on a given attempt.}$$

Kinetic Monte Carlo: Waiting time

Time increment between two flips.

- Suppose on the average there is one attempted flip per lattice site in time τ (physical time), (# of sites $= N = M^2$) then

  $$\frac{Q_{10}}{N}: \text{ Probability of flipping for a spin on a given attempt.}$$

- Note that the above procedure has homogenized the successful flipping probability to each site. We have the successful flipping probability for one site in unit time

  $$\frac{Q_{10}}{N} \Big/ \frac{\tau}{N} = \frac{Q_{10}}{\tau}: \text{ Flip one spin unit time.}$$

Kinetic Monte Carlo: Waiting time

- Define $P(\Delta t)$ is the probability that no flip occurs before time Δt has elapsed since the previous flip, then

  $$P(\Delta t) - P(\Delta t + dt) = P(\Delta t) \cdot \frac{Q_{10}}{\tau} dt,$$

  so one has

  $$P(\Delta t) = \exp(-\frac{Q_{10}\Delta t}{\tau}), \quad P(0) = 1.$$

  i.e. the time increment

  $$\Delta t = -\frac{\tau}{Q_{10}} \ln R, \quad R \sim \mathcal{U}[0,1], \ 0 \leq R \leq 1.$$

Mathematical Basis of KMC

- Essence of KMC: A continuous time Q-process with Q-matrix

  $$q_{ij} = 1_{Q_{ij}} A_{ij}$$

  in KMC, where $1_{Q_{ij}}$ is defined as $1_{Q_{ij}} = 1$ if $Q_{ij} > 0$ and $1_{Q_{ij}} = 0$ otherwise. A_{ij} is the acceptance probability P shown above.

Mathematical Basis of KMC

- Essence of KMC: A continuous time Q-process with Q-matrix

  $$q_{ij} = 1_{Q_{ij}} A_{ij}$$

  in KMC, where $1_{Q_{ij}}$ is defined as $1_{Q_{ij}} = 1$ if $Q_{ij} > 0$ and $1_{Q_{ij}} = 0$ otherwise. A_{ij} is the acceptance probability P shown above.
- If one applies KMC to compute the ensemble average, the time increment occurs as a weight for different states.

Mathematical Basis of KMC

- Essence of KMC: A continuous time Q-process with Q-matrix

$$q_{ij} = 1_{Q_{ij}} A_{ij}$$

  in KMC, where $1_{Q_{ij}}$ is defined as $1_{Q_{ij}} = 1$ if $Q_{ij} > 0$ and $1_{Q_{ij}} = 0$ otherwise. A_{ij} is the acceptance probability P shown above.
- If one applies KMC to compute the ensemble average, the time increment occurs as a weight for different states.
- KMC can simulate the non-equilibrium process such as crystal growth, but the connection between the process and the real physics is not clear!