统计物理与复杂系统暑期学校2015

郭文安

北京师范大学物理系

2015 年兰州大学

参考文献:

- H.W.J. Blöte, Lecture notes: Nonlocal Monte carlo Methods
- A. Sandvik, Computational Studies of Quantum Spin Systems, AIP Conf. Proc. 1297, 135 (2010);

提纲

临界点的确定：Binder ratio

Binder Ratio : $Q = \frac{\langle m^2 \rangle}{\langle |m| \rangle^2}$, 或者, $Q = \frac{\langle m^4 \rangle}{\langle m^2 \rangle^2}$

$$\chi = \frac{N}{T}(\langle m^2 \rangle - \langle m \rangle^2) \propto t^{-\gamma}$$

有限尺寸标度(finite-size scaling, 相变点以L代替$\xi \propto t^{-\nu}$)

$$\chi(L, T_c) = \frac{N}{T}(\langle m^2 \rangle - \langle m \rangle^2) \propto L^{\gamma/\nu}$$

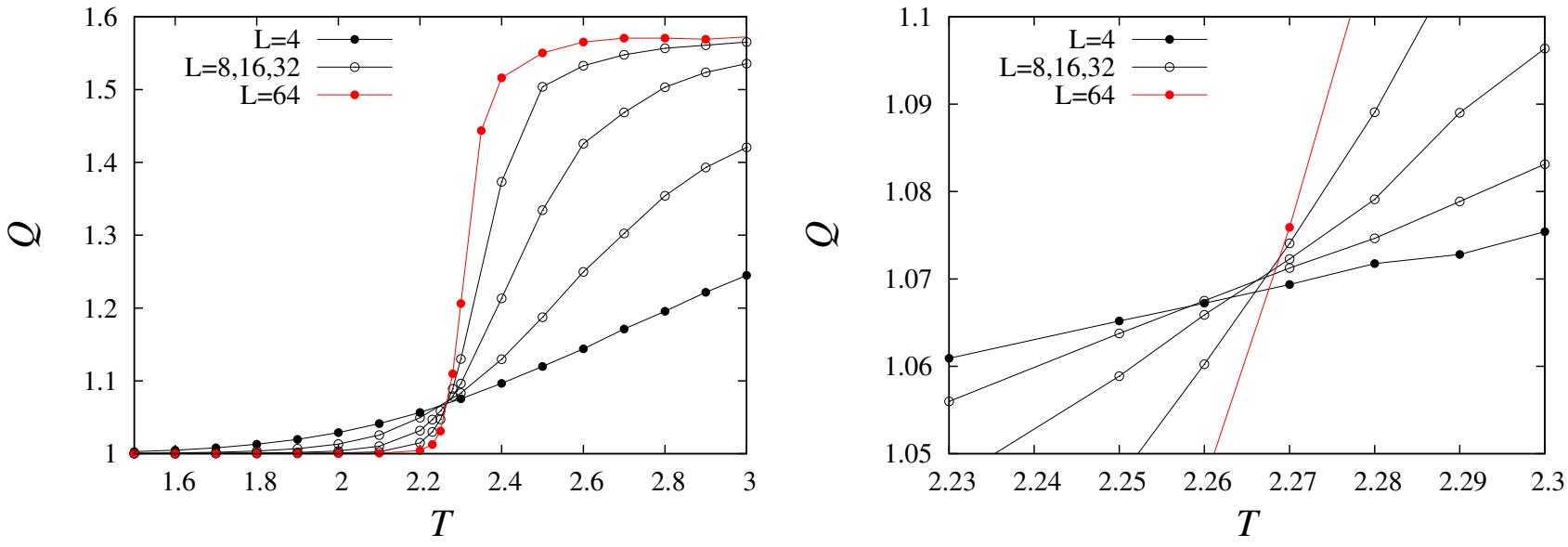
$$\langle m^2 \rangle \propto L^{\gamma/\nu - d} \qquad \langle |m| \rangle \propto L^{\gamma/(2\nu) - d/2}$$

Q 在临界点与尺寸无关!

$Q \to 1$ 当$T \to 0$; $Q \to constant$ 当$T \to \infty$, 由于m的高斯分布

临界点的确定：Binder ratio

Binder Ratio : $Q = \frac{\langle m^2 \rangle}{\langle |m| \rangle^2}$



- 交点并不完全重合，由于标度修正：非关涉场，RG理论
- $L \to \infty$, 交点是真正的临界点
- 可以根据$L, 2L$交点外推到$L \to \infty$

自关联时间

设某个物理观测量在第tMC步的值为$Q(t)$

自关联函数测量它需要花多长‘时间’与之前数值独立?

$$A_Q(\Delta t) = \frac{\langle Q(t+\Delta t)Q(t)\rangle - \langle Q(t)\rangle^2}{\langle Q(t)^2\rangle - \langle Q(t)\rangle^2}$$

一般满足

$$A_Q(\Delta t) \sim \mathrm{e}^{-\Delta t/\tau},\ \tau \text{ 自关联时间}$$

Integerated autocorrelation time (决定独立样本数$\approx n/(2\tau_{int})$)

$$\tau_{int} = \frac{1}{2} + \sum_{\Delta t=1}^{\infty} A_Q(\Delta t) \approx \tau$$

- 临界慢化(critical slowing down)

  $\tau_{int} \propto \xi^z \to \infty$, 热力学极限下，当$T \to T_c$

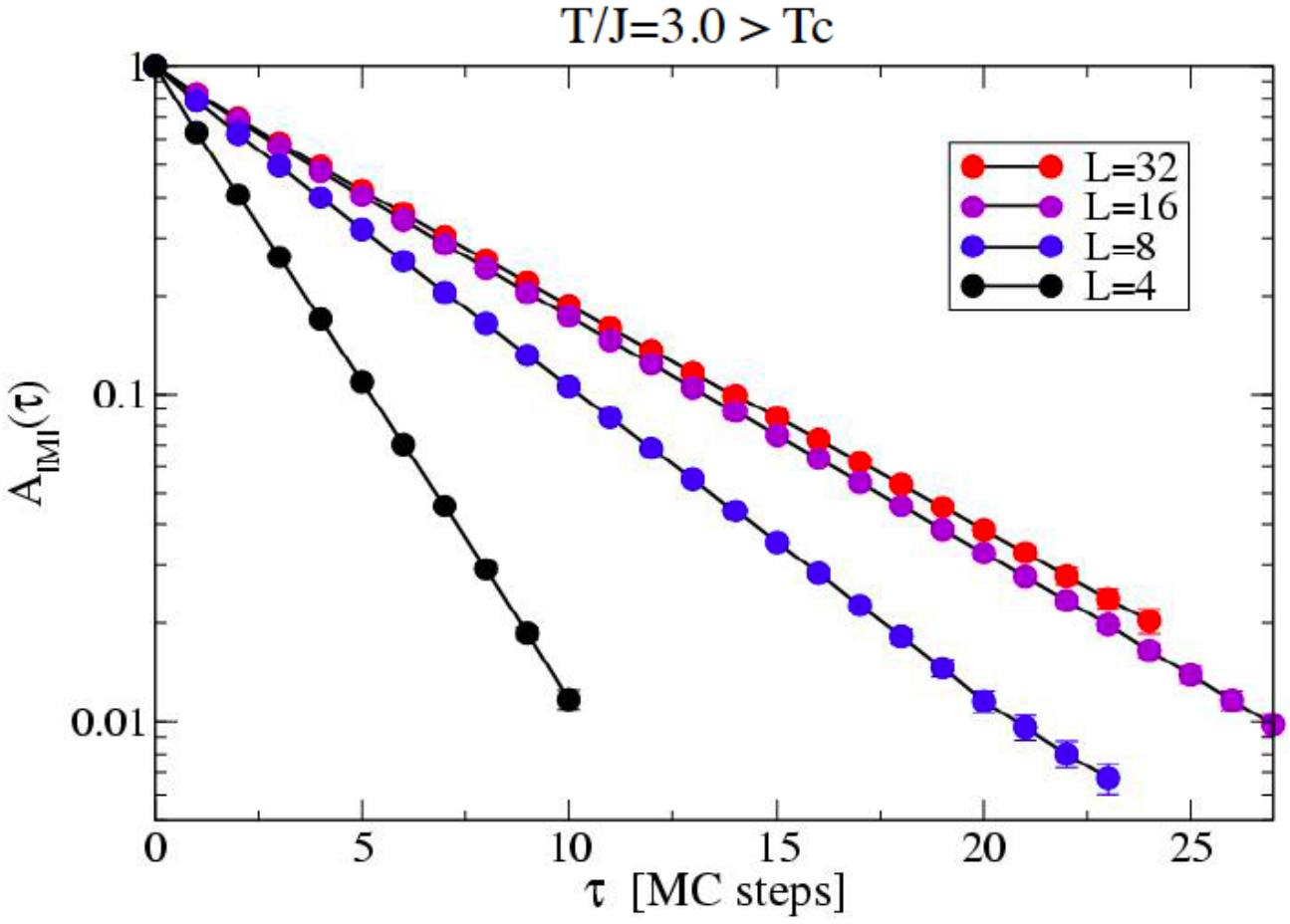
- 对于处于临界点T_c的有限尺寸系统, Q为序参量

  $$\tau_{int} \sim L^z,$$

z动力学临界指数，由实现平衡的动力学, 亦即算法，决定!

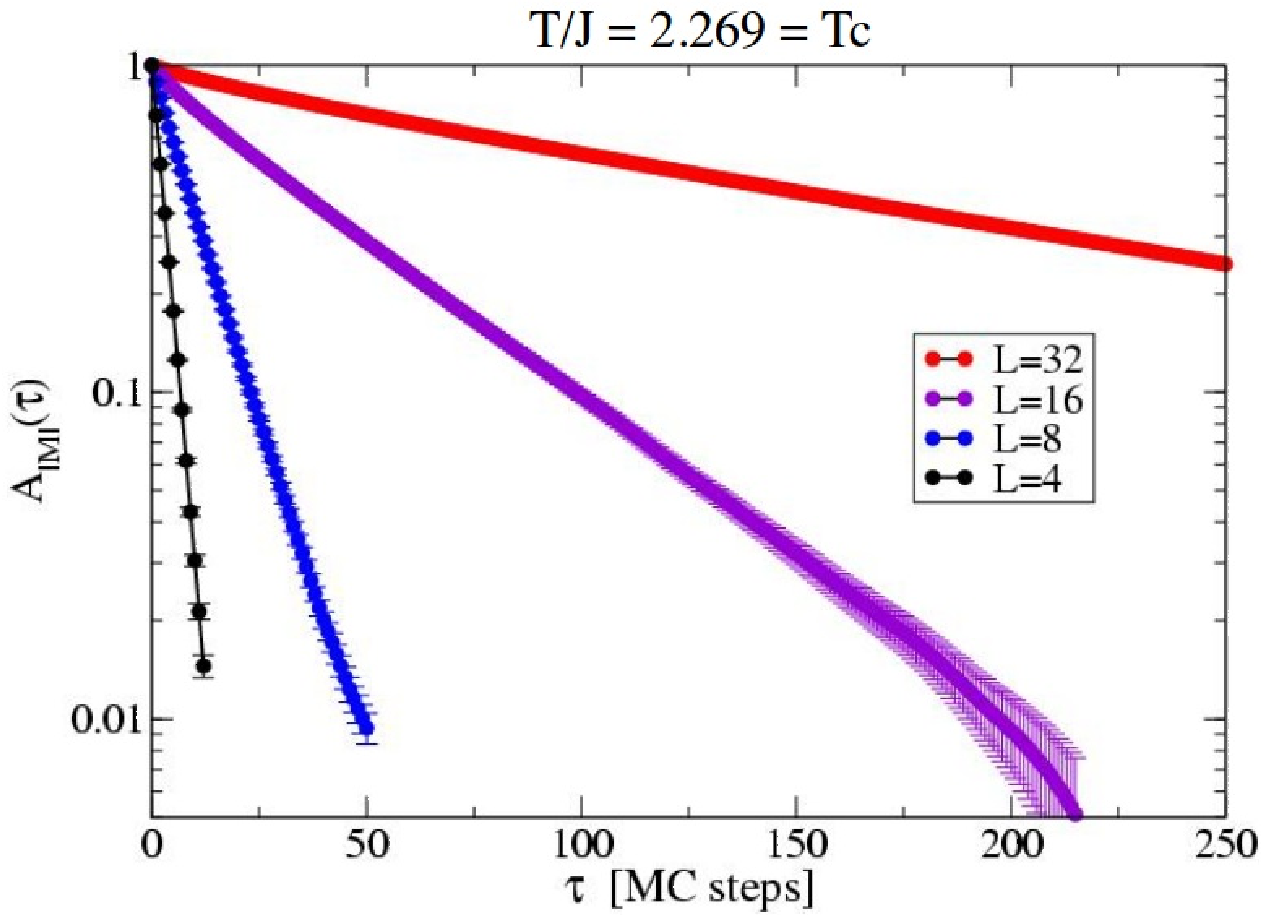
Metropolis算法的弱点, 考虑$A_{|M|}(\Delta t)$

当温度高于T_c



关联时间迅速收敛到与尺寸无关的值

Metropolis算法的弱点, 考虑$A_{|M|}(\Delta t)$



$z \approx 2.2$: 对于局域算法(local algorithm), 由于每次尝试的Γ'与Γ相差很小(local 算法) **自关联时间**很长

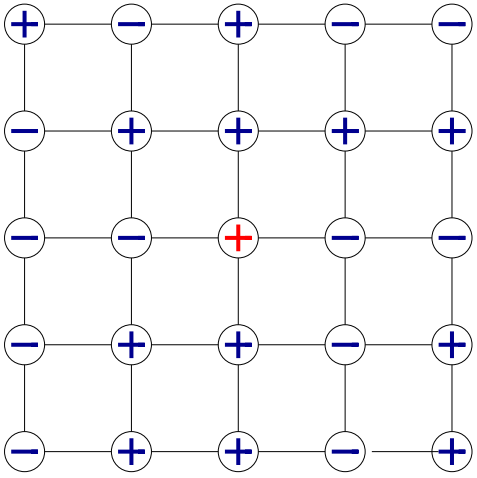
- 需要经过正比于L^{d+z}次操作后才可以得到一个独立的位形

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

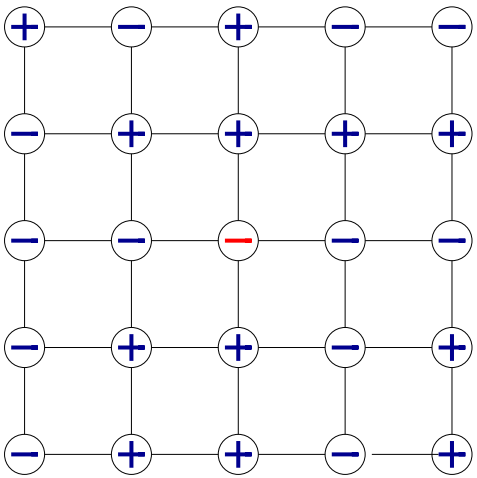
步骤:

1. 任意选取一个自旋i



Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

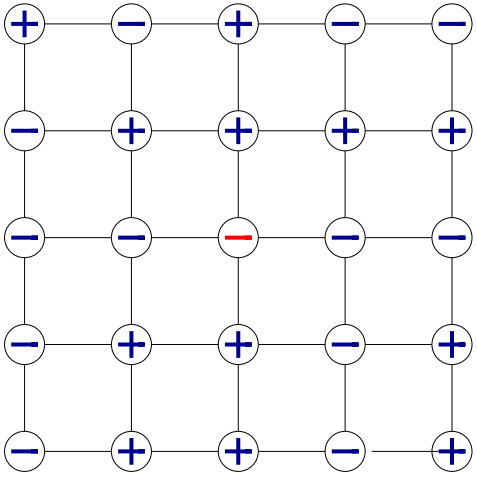


步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

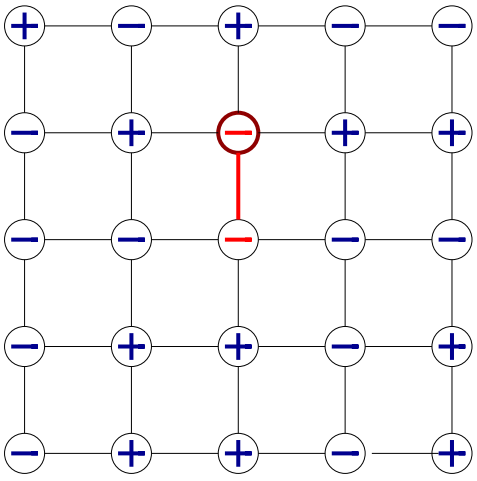


步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$
3. 对i的所有最近邻k作如下操作:
   - 如果$s_k = s_i$ 则以几率$1 - \mathrm{e}^{-2K}$, $K = J/k_B T$完成
     i. 翻转$s_k \to s_k' = -s_k$.
     ii. 将k记录到地址列表('堆栈(stack)')

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

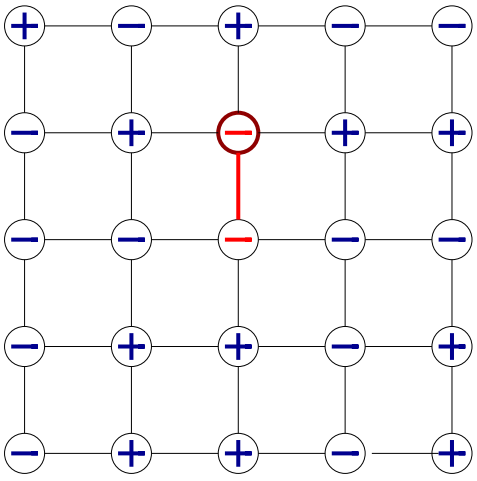


步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$
3. 对i的所有最近邻k作如下操作:
   - 如果$s_k = s_i$ 则以几率$1 - \mathrm{e}^{-2K}$, $K = J/k_B T$完成
     i. 翻转$s_k \to s_k' = -s_k$.
     ii. 将k记录到地址列表('堆栈(stack)')

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

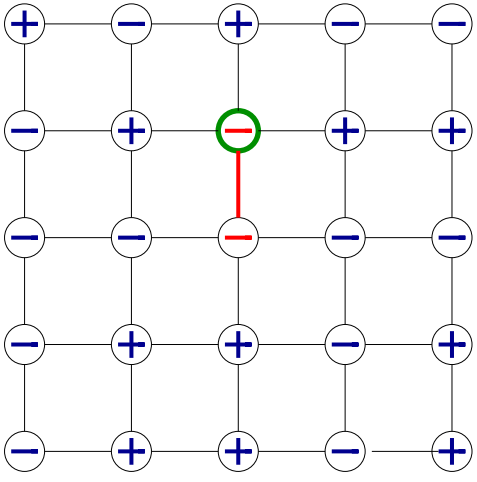


步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$
3. 对i的所有最近邻k作如下操作:
   - 如果$s_k = s_i$ 则以几率$1 - \mathrm{e}^{-2K}$, $K = J/k_B T$完成
     i. 翻转$s_k \to s_k' = -s_k$.
     ii. 将k记录到地址列表('堆栈(stack)')
   - 如果$s_k = s_i'$ 什么也不作

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

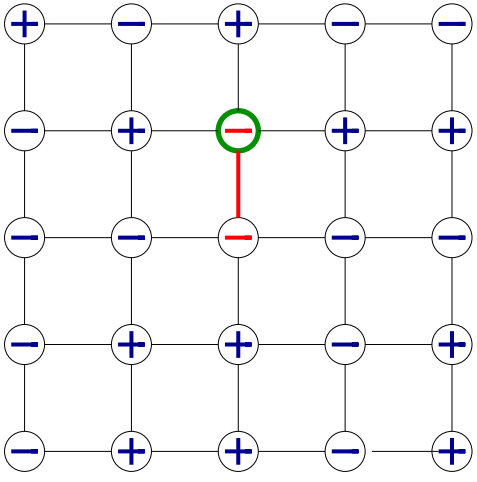


步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$
3. 对i的所有最近邻k作如下操作:
   - 如果$s_k = s_i$ 则以几率$1 - \mathrm{e}^{-2K}$, $K = J/k_B T$完成
     i. 翻转$s_k \to s_k' = -s_k$.
     ii. 将k记录到地址列表('堆栈(stack)')
   - 如果$s_k = s_i'$ 什么也不作
4. 从堆栈中读取一个地址l

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

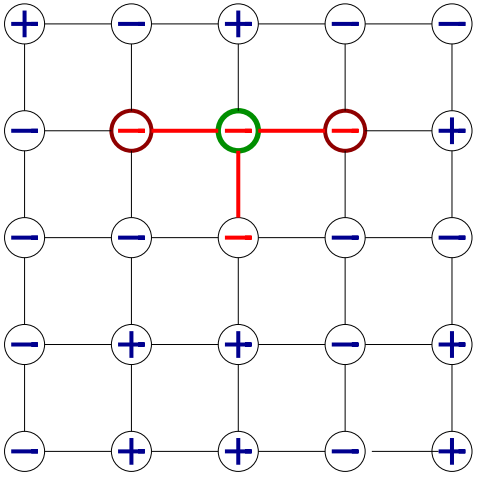


步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$
3. 对i的所有最近邻k作如下操作:
   - 如果$s_k = s_i$ 则以几率$1 - \mathrm{e}^{-2K}$, $K = J/k_B T$完成
     i. 翻转$s_k \to s_k' = -s_k$.
     ii. 将k记录到地址列表('堆栈(stack)')
   - 如果$s_k = s_i'$ 什么也不作
4. 从堆栈中读取一个地址l
5. 对l执行第3步

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

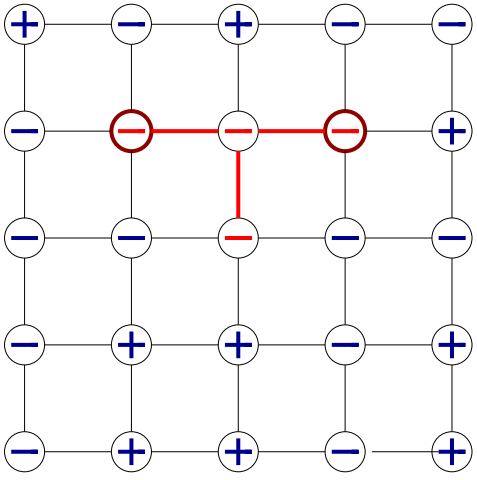


步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$
3. 对i的所有最近邻k作如下操作:
   - 如果$s_k = s_i$ 则以几率$1 - \mathrm{e}^{-2K}$, $K = J/k_B T$完成
     i. 翻转$s_k \to s_k' = -s_k$.
     ii. 将k记录到地址列表('堆栈(stack)')
   - 如果$s_k = s_i'$ 什么也不作
4. 从堆栈中读取一个地址l
5. 对l执行第3步

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

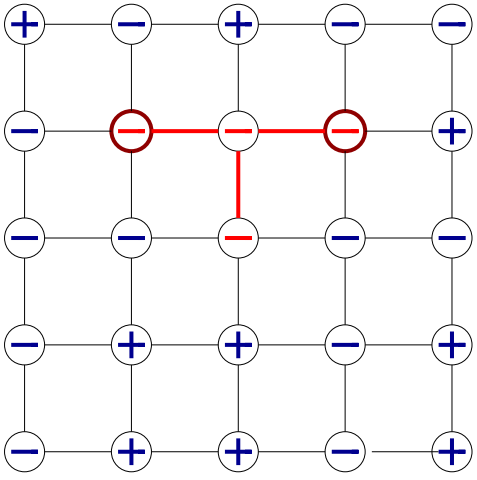


步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$
3. 对i的所有最近邻k作如下操作:
   - 如果$s_k = s_i$ 则以几率$1 - \mathrm{e}^{-2K}$, $K = J/k_B T$完成
     i. 翻转$s_k \to s_k' = -s_k$.
     ii. 将k记录到地址列表('堆栈(stack)')
   - 如果$s_k = s_i'$ 什么也不作
4. 从堆栈中读取一个地址l
5. 对l执行第3步
6. 从堆栈中删除l

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次

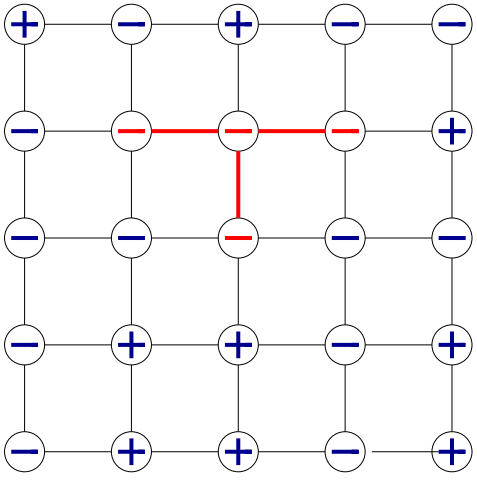


步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$
3. 对i的所有最近邻k作如下操作:
   - 如果$s_k = s_i$ 则以几率$1 - \mathrm{e}^{-2K}$, $K = J/k_B T$完成
     i. 翻转$s_k \to s_k' = -s_k$.
     ii. 将k记录到地址列表('堆栈(stack)')
   - 如果$s_k = s_i'$ 什么也不作
4. 从堆栈中读取一个地址l
5. 对l执行第3步
6. 从堆栈中删除l
7. 重复4-6, 直到堆栈为空

Cluster algorithm: Wolff algorithm

在一步里翻转一个集团，而不是一个一个地翻$N = L^d$次



步骤:

1. 任意选取一个自旋i
2. 翻转之: $s_i' = -s_i$
3. 对i的所有最近邻k作如下操作:
   - 如果$s_k = s_i$ 则以几率$1 - \mathrm{e}^{-2K}$, $K = J/k_BT$完成
     i. 翻转$s_k \to s_k' = -s_k$.
     ii. 将k记录到地址列表('堆栈(stack)')
   - 如果$s_k = s_i'$ 什么也不作
4. 从堆栈中读取一个地址l
5. 对l执行第3步
6. 从堆栈中删除l
7. 重复4-6, 直到堆栈为空

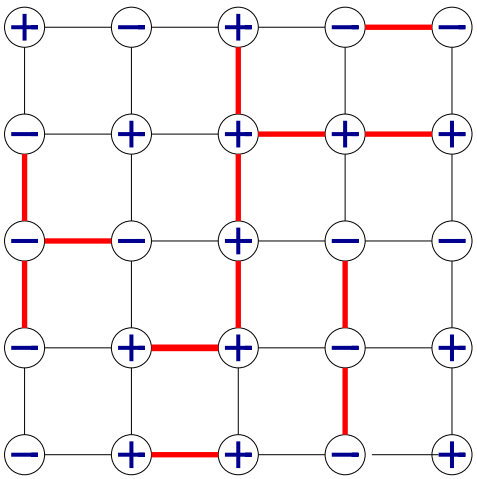
程序实现

```
s=rn()*n+1 !! rn()是随机数，随机选择一个自旋s (位置)
icsp=spin(s) !! icsp 是自旋s原来的状态
spin(s)=-icsp !! 翻转s
nstack=0 !! 堆栈里的成员数和当前成员标号
js=s  !! 当前考虑的自旋（位置）
104 continue
do 107 inb=1,4
   ks=nbor(inb,js) !! js 的4个邻居位置，事先算好保存在数组nbor中
   if ((spin(ks).eq.icsp).and.(rn().lt.bp)) then  !! bp 是几率1 − e^{−2K}
      nstack=nstack+1 !! ks 放入堆栈，成员数加一
       istn(nstack)=ks  !! ks 是第nstack个成员，保存在数组istn中
       spin(ks)=-icsp !! 翻转ks
   endif
107 continue  !! 完成对s 的4个邻居的操作
if (nstack.eq.0) goto 110  !! 堆栈里的成员处理完后，结束Wolff步骤
js=istn(nstack)  !! 从堆栈中读取一个成员
nstack=nstack-1  !! 待操作的成员数减少一个
goto 104  !! 取处理当前成员
110 continue
```

为什么高效?

我们知道磁化率$\chi = \frac{\langle (N_+ - N_-)^2 \rangle}{k_B T N}$, 即正比于磁矩的涨落

对同号自旋按几率$1 - \mathrm{e}^{-2K}$加棒连接，一个自旋位形等价于一个‘棒位形+集团符号’

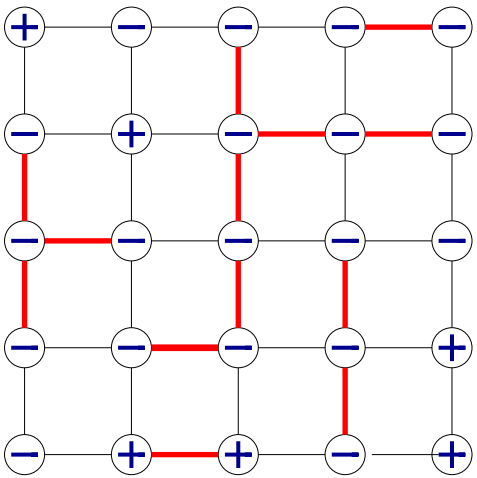


设棒位形有M个集团$N_+ - N_- = \sum_{k=1}^{M} n_k \mathrm{sig}_k$
可以证明:

为什么高效?

我们知道磁化率$\chi = \frac{\langle (N_+ - N_-)^2 \rangle}{k_B T N}$, 即正比于磁矩的涨落

对同号自旋按几率$1 - \mathrm{e}^{-2K}$加棒连接，一个自旋位形等价于一个‘棒位形+集团符号’

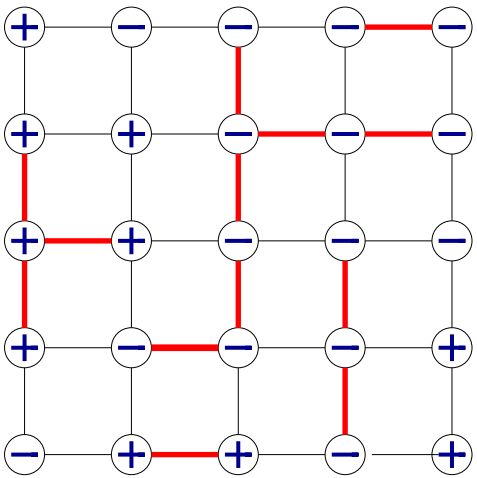


设棒位形有M个集团$N_+ - N_- = \sum_{k=1}^{M} n_k \mathrm{sig}_k$
可以证明:

为什么高效?

我们知道磁化率$\chi = \frac{\langle (N_+ - N_-)^2 \rangle}{k_B T N}$, 即正比于磁矩的涨落

对同号自旋按几率$1 - \mathrm{e}^{-2K}$加棒连接，一个自旋位形等价于一个‘棒位形+集团符号’

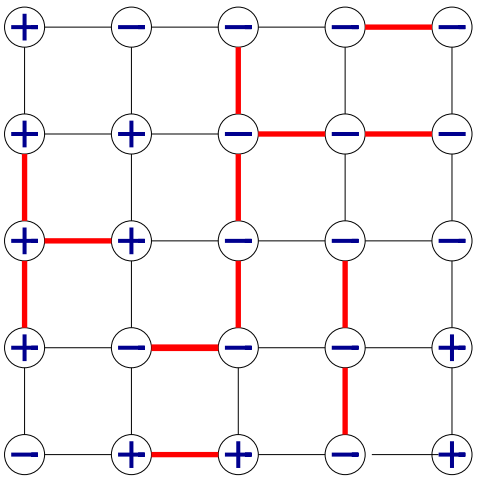


设棒位形有M个集团$N_+ - N_- = \sum_{k=1}^{M} n_k \mathrm{sig}_k$
可以证明:

为什么高效?

我们知道磁化率$\chi = \frac{\langle (N_+ - N_-)^2 \rangle}{k_B T N}$, 即正比于磁矩的涨落

对同号自旋按几率$1 - \mathrm{e}^{-2K}$加棒连接，一个自旋位形等价于一个‘棒位形+集团符号’

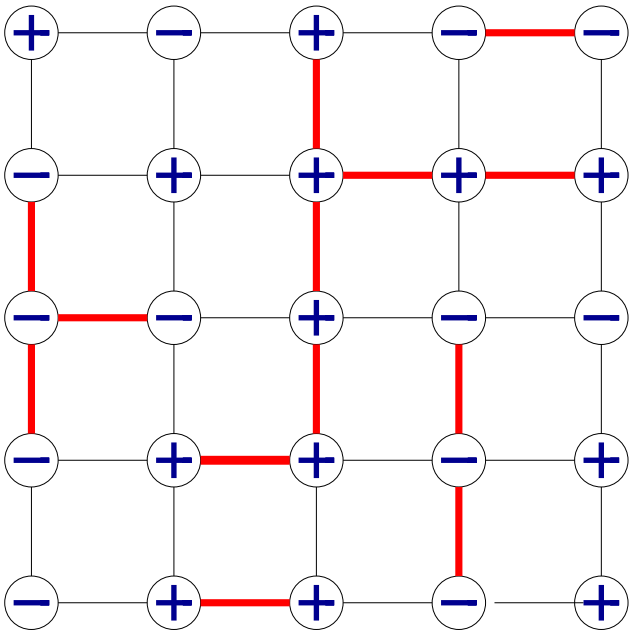


设棒位形有M个集团$N_+ - N_- = \sum_{k=1}^{M} n_k \mathrm{sig}_k$
可以证明:

$$\chi = \frac{1}{N}\langle \sum_k n_k^2 \rangle_B = \langle \sum_k \frac{n_k}{N} n_k \rangle_B$$

Wolff算法随机选取一个自旋，落在第k个集团的几率正是n_k/N,
所以平均Wolff集团的大小等于磁化率χ!
或者说Wolff集团的大小是磁化率的*estimator*!

为什么高效?



在临界点附近，我们知道$\chi \propto t^{-\gamma} = \xi^{\gamma/\nu}$
对于有限尺寸系统，以L代替ξ

$$\chi \propto L^{\gamma/\nu}$$

对于2D Ising: $\gamma = 7/4, \nu = 1$ 集团大小随尺寸L发散!
一个Wolff集团翻转就可以在很大程度上改变位形!
动力学临界指数非常小!

细致平衡的证明

定义$P(C,\Gamma)$为形成C **内部连接棒**位形的几率. 设边界棒(一端属于C, 另一端不属于)的数目为$n = n_a + n_p$, n_a是两端自旋反号的棒数，n_p是同号的棒数.

将同号自旋排除在C之外的几率：e^{-2Kn_p}, 反号自旋以几率1排除

将C翻转得到新位形, 整个过程的几率

$$T(\Gamma',\Gamma) = \mathrm{e}^{-2Kn_p} P(C,\Gamma)\frac{1}{N}.$$

反之，计算从Γ'到Γ的几率

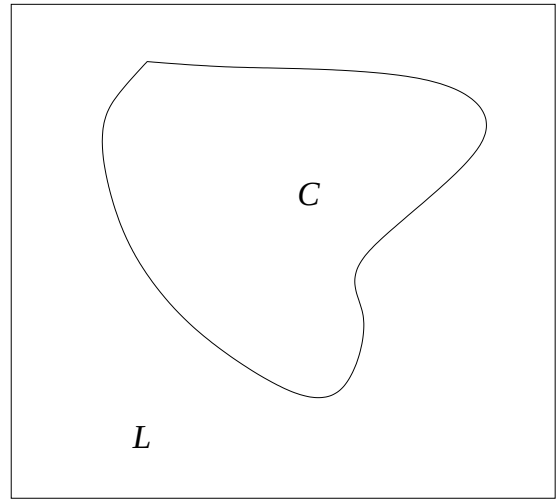
$$T(\Gamma,\Gamma') = \mathrm{e}^{-2Kn_a} P(C,\Gamma')\frac{1}{N}.$$

注意到内部连接几率不变

$$P(C,\Gamma) = P(C,\Gamma').$$

因此

$$T(\Gamma',\Gamma) = \mathrm{e}^{2K(n_a-n_p)} T(\Gamma,\Gamma').$$



C表示Wolff集团, 是晶格L的子集

细致平衡的证明

由于集团的翻转只改变边界的相互作用能，

$$P_{\mathrm{eq}}(\Gamma)/P_{\mathrm{eq}}(\Gamma') = \mathrm{e}^{-(E(\Gamma)-E(\Gamma'))/k_BT} = \mathrm{e}^{2(n_p-n_a)K}$$

因此

$$P_{\mathrm{eq}}(\Gamma)T(\Gamma',\Gamma) = P_{\mathrm{eq}}(\Gamma')T(\Gamma,\Gamma')$$

魔术的秘诀在哪里?

配分函数

$$Z = \sum_S W(S), \qquad S\text{是自旋位形}$$

连接棒之后

$$Z = \sum_{S,B} W(S)P(B|S) \equiv \sum_{S,B} W(S,B)$$

这里利用了从S生成棒位形B的几率$P(B|S)$是归一的:

$$\sum_B P(B|S) = 1$$

位形空间多了一个维度B，而配分函数不变!

- 集团算法把$\Gamma = \{S, B\} \to \Gamma' = \{S', B\}$
- 由于上页实际证明了$W(S,B) = W(S',B)$, 所以这个$S \to S'$的跃迁几率是$T(\Gamma', \Gamma) = 1$.
- 哲学就是“高一维”的位形空间里两个等几率的位形，可以轻松互相跃迁，而其自旋位形相差很大!

好比开通了‘空中航线‘

Cluster algorithm: Swendsen-Wang algorithm

与Wolff算法非常接近

1. 从自旋位形开始
2. 构造集团：与Wolff 算法一样, 但是找到所有集团
3. 以1/2几率翻转集团
4. 回到[1]

Swendsen-Wang

Swendsen-Wang

Swendsen-Wang

概述

The Metropolis algorithm can be used for any system

- Critical slowing down can be serious
- The dynamics can be slow also in non-critical systems, e.g., glassy system.

Cluster algorithms

- other versions， e.g., Geometry cluster
- not working for
  - Magnets with frustration: clusters span the whole system before reaching critical point
  - Magnets in external magnetic fields
  - Most systems of particles in continuous space, no way to construct clusters in general.

提纲

基本想法

Thermal expectation value of a quantum system

$$\langle A \rangle = \frac{1}{Z} \mathrm{Tr}\{A \mathrm{e}^{-\beta H}\}, \qquad Z = \mathrm{Tr} \mathrm{e}^{-\beta H}$$

We don't know the eigenvalues and eigenstates of H.

We wish to write

$$\langle A \rangle = \sum_{\Gamma} A(\Gamma) \frac{W(\Gamma)}{Z}$$

where

$$Z = \sum_{\Gamma} W(\Gamma)$$

Then we can apply the basic idea of Monte Carlo: importance sampling

$$\langle A \rangle = \frac{1}{M} \sum_{l=1}^{M} A(\Gamma_l)$$

Path integral representation

Path integrals in quantum statistical mechanics

"Time slicing" of the partition function

$$Z = \mathrm{Tr}\{\mathrm{e}^{-\beta H}\} = \mathrm{Tr}\{\prod_{l=1}^{L} \mathrm{e}^{-\Delta_\tau H}\},\ \Delta_\tau = \beta/L$$

Choose a basis and insert complete sets of states

$$Z = \sum_{\alpha_0}\sum_{\alpha_1}\cdots\sum_{\alpha_{L-1}} \langle\alpha_0|\mathrm{e}^{-\Delta_\tau H}|\alpha_{L-1}\rangle\cdots\langle\alpha_2|\mathrm{e}^{-\Delta_\tau H}|\alpha_1\rangle\langle\alpha_1|\mathrm{e}^{-\Delta_\tau H}|\alpha_0\rangle$$

Approximation;

$$Z \approx \sum_{\{\alpha\}} \langle\alpha_0|1-\Delta_\tau H|\alpha_{L-1}\rangle\cdots\langle\alpha_2|1-\Delta_\tau H|\alpha_1\rangle\langle\alpha_1|1-\Delta_\tau H|\alpha_0\rangle$$

Leads to error $\propto \Delta_\tau^2$. Limit $\Delta_\tau \to 0$ can be taken.

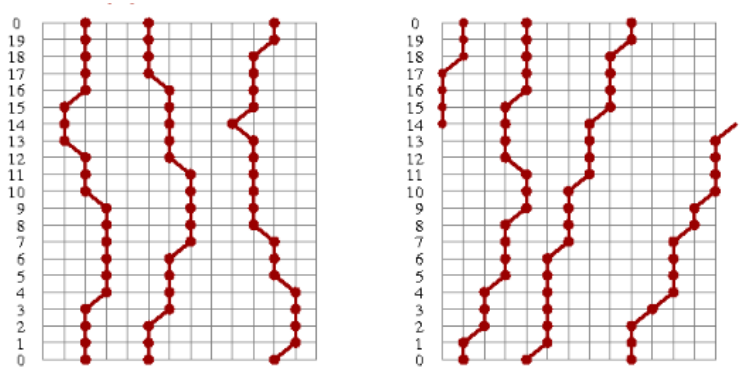
Example: hard-core bosons

$$H = -\sum_{\langle i,j\rangle} K_{ij} = -\sum_{\langle i,j\rangle} (a_j^\dagger a_i + a_i^\dagger a_j) \qquad n_i = a_i^\dagger a_i \in \{0,1\}$$

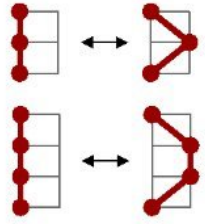
Equivalent to $S = 1/2$ XY model

$$H = -2\sum_{\langle i,j\rangle} (S_i^x S_j^x + S_i^y S_j^y) = -\sum_{\langle i,j\rangle} (S_i^+ S_j^- + S_i^- S_j^+), \qquad S^z = \pm\frac{1}{2}$$

"World line" representation of
$Z \approx \sum_{\{\alpha\}} \langle\alpha_0|1-\Delta_\tau H|\alpha_{L-1}\rangle \cdots \langle\alpha_2|1-\Delta_\tau H|\alpha_1\rangle\langle\alpha_1|1-\Delta_\tau H|\alpha_0\rangle$



World line moves for MC updating



$$Z = \sum_{\{\alpha\}} W(\{\alpha\}), \quad W(\{\alpha\}) = \Delta_\tau^{n_K}, \qquad n_K = \text{number of "jumps"}$$

Expectation values

目标是计算$\langle A\rangle$, 可写为

$$\langle A\rangle = \frac{1}{Z}\sum_{\{\alpha\}}\langle\alpha_0|\mathrm{e}^{-\Delta_\tau H}|\alpha_{L-1}\rangle\cdots\langle\alpha_2|\mathrm{e}^{-\Delta_\tau H}|\alpha_1\rangle\langle\alpha_1|\mathrm{e}^{-\Delta_\tau H}A|\alpha_0\rangle$$

我们把它写成适合MC importance sampling

$$\boxed{\langle A\rangle = \frac{\sum_{\{\alpha\}}A(\{\alpha\})W(\{\alpha\})}{\sum_{\{\alpha\}}W(\{\alpha\})} = \langle A(\{\alpha\})\rangle_W}$$

$W(\{\alpha\})=$ **weight;** $A(\{\alpha\})=$ **estimator**

- 对角算符(物理量): 粒子数，s_z,

$$\boxed{A(\{\alpha\}) = \frac{1}{L}\sum_{l=0}^{L-1}A(\alpha_l)}$$

Expectation values

目标是计算$\langle A\rangle$, 可写为

$$\langle A\rangle = \frac{1}{Z}\sum_{\{\alpha\}}\langle\alpha_0|\mathrm{e}^{-\Delta_\tau H}|\alpha_{L-1}\rangle\cdots\langle\alpha_2|\mathrm{e}^{-\Delta_\tau H}|\alpha_1\rangle\langle\alpha_1|\mathrm{e}^{-\Delta_\tau H}A|\alpha_0\rangle$$

我们把它写成适合MC importance sampling

$$\boxed{\langle A\rangle = \frac{\sum_{\{\alpha\}}A(\{\alpha\})W(\{\alpha\})}{\sum_{\{\alpha\}}W(\{\alpha\})} = \langle A(\{\alpha\})\rangle_W}$$

$W(\{\alpha\})=$ **weight;** $A(\{\alpha\})=$ **estimator**

- 非对角算符: Kinetic energy, Quantum Mechanics!

  利用$K\mathrm{e}^{-\Delta_\tau K}\approx K$, 构造一个estimator $A(\{\alpha\})$!

$$\boxed{K_{ij}(\{\alpha\}) = \frac{\langle\alpha_1|K_{ij}|\alpha_0\rangle}{\langle\alpha_1|1-\Delta_\tau K|\alpha_0\rangle}\in\{0,\frac{1}{\Delta_\tau}\}}$$

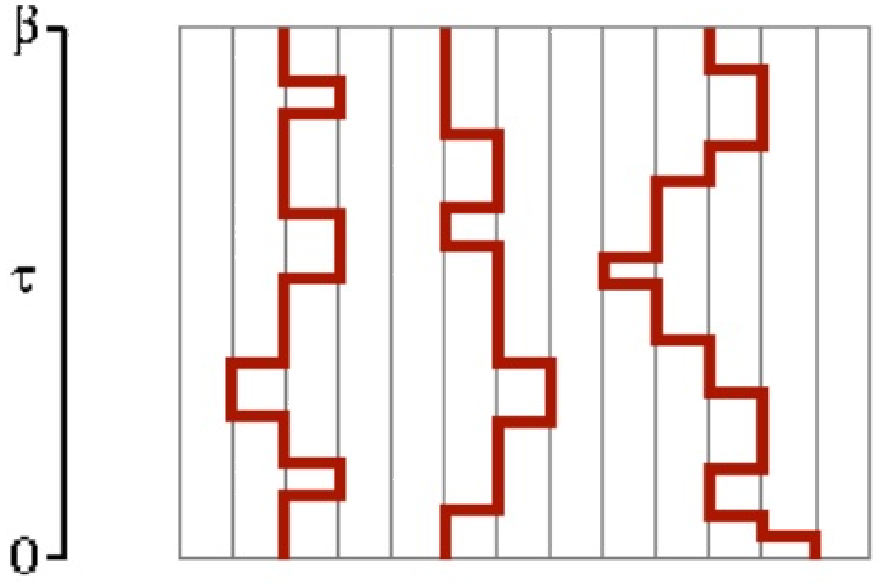
Average over all slices: count number of kinetic jumps

$$\langle K_{ij}\rangle = \frac{n_{ij}}{\beta},\ \langle K\rangle = -\frac{\langle n_K\rangle}{\beta} \qquad \text{利用了}\Delta_\tau=\beta/L$$

$\langle K\rangle\propto N\to\langle n_K\rangle\propto\beta N$: There should be of the order βN

The continuous time limit

$\Delta_\tau \to 0$: number of kinetic jumps remains finite, store events only



Worm updates, Prokofev et al (1996);
loop update, Evertz, Lana, Marcu (1993);

Stochastic Series Expansion

Quantum Monte Carlo method

Series expansion representation

Taylor expansion $\mathrm{e}^{-\beta H} = \sum_n^\infty \frac{(-\beta)^n}{n!} H^n$

$$\boxed{Z = \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \sum_{\{\alpha\}_n} \langle \alpha_0 | H | \alpha_{n-1} \rangle \cdots \langle \alpha_2 | H | \alpha_1 \rangle \langle \alpha_1 | H | \alpha_0 \rangle}$$

Similar to the path integral!

For hard-core bosons, the allowed path weights is $W(\text{path}) = \beta^n / n!$

Series expansion representation

Taylor expansion $\mathrm{e}^{-\beta H} = \sum_n^\infty \frac{(-\beta)^n}{n!} H^n$

$$\boxed{Z = \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \sum_{\{\alpha\}_n} \langle\alpha_0|H|\alpha_{n-1}\rangle \cdots \langle\alpha_2|H|\alpha_1\rangle\langle\alpha_1|H|\alpha_0\rangle}$$

For any model,

$$E = \frac{1}{Z} \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \sum_{\{\alpha\}_{n+1}} \langle\alpha_0|H|\alpha_n\rangle \cdots \langle\alpha_2|H|\alpha_1\rangle\langle\alpha_1|H|\alpha_0\rangle$$

$$= \frac{1}{Z} \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \frac{n}{\beta} \sum_{\{\alpha\}_n} \langle\alpha_0|H|\alpha_{n-1}\rangle \cdots \langle\alpha_2|H|\alpha_1\rangle\langle\alpha_1|H|\alpha_0\rangle$$

$$Z = \sum_{\Gamma} W(\Gamma), \qquad \Gamma \text{ 由} \{\alpha\}_n \text{ 决定}, E = \frac{\sum_\Gamma \frac{n}{\beta} W(\Gamma)}{\sum_\Gamma W(\Gamma)} = \frac{\langle n\rangle}{\beta}$$

Series expansion representation

Taylor expansion $\mathrm{e}^{-\beta H} = \sum_n^\infty \frac{(-\beta)^n}{n!} H^n$

$$\boxed{Z = \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \sum_{\{\alpha\}_n} \langle\alpha_0|H|\alpha_{n-1}\rangle \cdots \langle\alpha_2|H|\alpha_1\rangle\langle\alpha_1|H|\alpha_0\rangle}$$

$$Z = \sum_{\Gamma} W(\Gamma), \qquad \Gamma \text{ 由} \{\alpha\}_n \text{ 决定}, E = \frac{\sum_\Gamma \frac{n}{\beta} W(\Gamma)}{\sum_\Gamma W(\Gamma)} = \frac{\langle n\rangle}{\beta}$$

similarly,

$$C = \langle n^2\rangle - \langle n\rangle^2 - \langle n\rangle$$

Narrow n-distribution with $\langle n\rangle \approx N\beta, \sigma_n \approx \sqrt{N\beta}$

So, fix length of expansion: cut-off at L and fill in with unit operators $H_0 = 1$

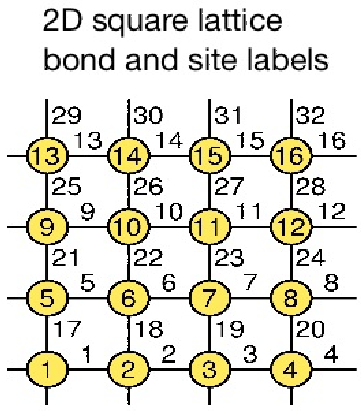
$$\boxed{Z = \sum_{\{\alpha\}_L} \sum_{\{H_i\}} \frac{(-\beta)^n (L-n)!}{L!} \langle\alpha_0|H_{i(L)}|\alpha_{L-1}\rangle \cdots \langle\alpha_2|H_{i(2)}|\alpha_1\rangle\langle\alpha_1|H_{i(1)}|\alpha_0\rangle}$$

n is the number of H_i with $i > 0$ in the sequence of L operators

Stochastic Series expansion(SSE): $s = \frac{1}{2}$ Heisenberg model

Write H as a bond sum for arbitrary lattice

$$H = J \sum_{b=1}^{N_b} \mathbf{S}_{i(b)} \cdot \mathbf{S}_{j(b)}$$



Diagonal(1) and off-diagonal (2) bond operators

$$H_{1,b} = \frac{1}{4} - \mathbf{S}^z_{i(b)} \mathbf{S}^z_{j(b)}$$
$$H_{2,b} = \frac{1}{2}(\mathbf{S}^+_{i(b)} \mathbf{S}^-_{j(b)} + \mathbf{S}^-_{i(b)} \mathbf{S}^+_{j(b)})$$

$$H = -J \sum_{b=1}^{N_b} (H_{1,b} - H_{2,b}) + \frac{J N_b}{4}$$

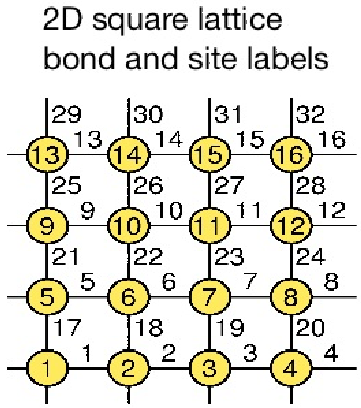
Four Non-zero matrix elements$= 1/2$

$$\langle \uparrow_i \downarrow_j | H_{1,b} | \uparrow_i \downarrow_j \rangle \; \langle \downarrow_i \uparrow_j | H_{2,b} | \uparrow_i \downarrow_j \rangle \; \langle \downarrow_i \uparrow_j | H_{1,b} | \downarrow_i \uparrow_j \rangle \; \langle \uparrow_i \downarrow_j | H_{2,b} | \downarrow_i \uparrow_j \rangle$$

Stochastic Series expansion(SSE): $s = \frac{1}{2}$ Heisenberg model

Write H as a bond sum for arbitrary lattice

$$H = J \sum_{b=1}^{N_b} \mathbf{S}_{i(b)} \cdot \mathbf{S}_{j(b)}$$



Four Non-zero matrix elements$= 1/2$

$$\langle \uparrow_i \downarrow_j |H_{1,b}| \uparrow_i \downarrow_j \rangle \langle \downarrow_i \uparrow_j |H_{2,b}| \uparrow_i \downarrow_j \rangle \langle \downarrow_i \uparrow_j |H_{1,b}| \downarrow_i \uparrow_j \rangle \langle \uparrow_i \downarrow_j |H_{2,b}| \downarrow_i \uparrow_j \rangle$$

Partition function

$$Z = \sum_{\alpha} \sum_{n=0}^{\infty} (-1)^{n_2} \frac{\beta^n}{n!} \sum_{S_n} \langle \alpha | \prod_{p=0}^{n-1} H_{a(p),b(p)} | \alpha \rangle = \sum_{\Gamma} W(\Gamma)$$

$n_2 =$ number of off-diagonal op.

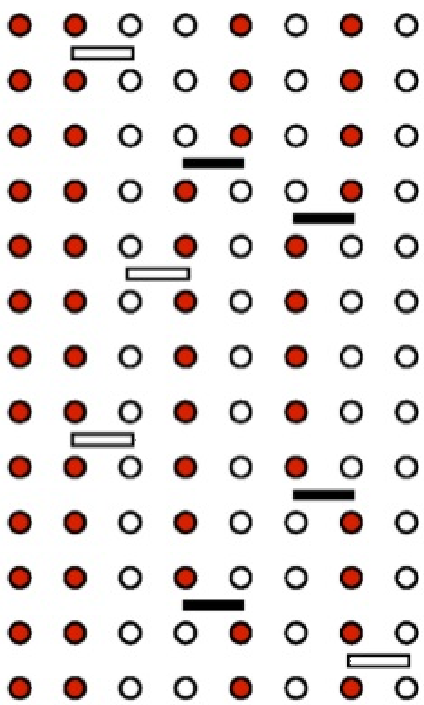
Configuration Γ: $|\alpha\rangle, S_n$: $a(p) = 1, 2; b(p) = b$

For fixed-length scheme

$$Z = \sum_{\alpha} \sum_{S_L} (-1)^{n_2} \frac{\beta^n (L-n)!}{L!} \langle \alpha | \prod_{p=0}^{L-1} H_{a(p),b(p)} | \alpha \rangle$$

State propagate $|\alpha(p)\rangle = \prod_{i=0}^{p-1} H_{a(i),b(i)} |\alpha\rangle$

i	=	1	2	3	4	5	6	7	8
$\sigma(i)$	=	-1	+1	-1	-1	+1	-1	+1	+1



p	$a(p)$	$b(p)$	$s(p)$
11	1	2	4
10	0	0	0
9	2	4	9
8	2	6	13
7	1	3	6
6	0	0	0
5	0	0	0
4	1	2	4
3	2	6	13
2	0	0	0
1	2	4	9
0	1	7	14

- n_2 is even for bipartite lattice!
- $s(p) =$ operator-index string
  - $s(p) = 2 * b(p) + a(p) - 1$
  - diagonal $s(p) = even$
  - off-diagonal $s(p) = odd$
- $\sigma(i) =$ spin state, $i = 1, \ldots, N$
  - only one has to be stored

SSE: an effective discrete representation

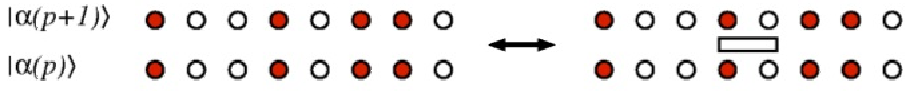
Monte Carlo updating scheme

$$W(\alpha, S_L) = (\frac{\beta}{2})^n \frac{(L-n)!}{L!}$$

Change the configuration; $(\alpha, S_L) \to (\alpha', S'_L)$

$$P_{accept} = \min[\frac{W(\alpha', S'_L)}{W(\alpha, S_L)} \frac{P_{\text{select}}(\alpha', S'_L)}{P_{\text{select}}(\alpha, S_L)}, 1]$$

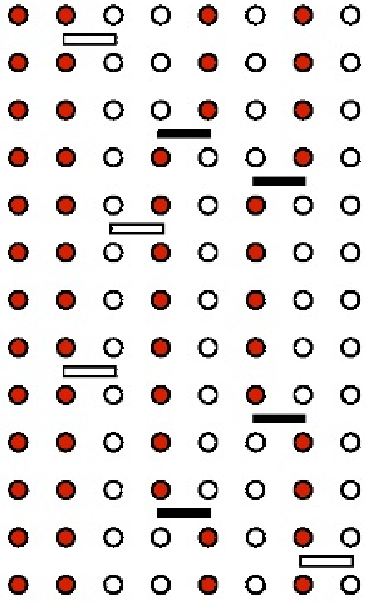
- Diagonal update: $[0,0]_p \leftrightarrow [1,b]_p$



- Attempt at every p, need to know $|\alpha(p)\rangle$

(如果棒两端自旋同号，不能加棒)

generated by flipping spins when off-diag op.



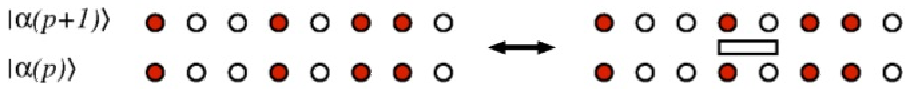
Monte Carlo updating scheme

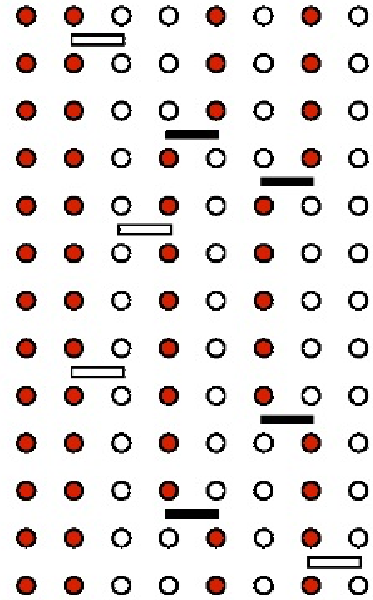
$$W(\alpha, S_L) = (\frac{\beta}{2})^n \frac{(L-n)!}{L!}$$

Change the configuration; $(\alpha, S_L) \to (\alpha', S'_L)$

$$P_{accept} = \min[\frac{W(\alpha', S'_L)}{W(\alpha, S_L)} \frac{P_{\text{select}}(\alpha', S'_L)}{P_{\text{select}}(\alpha, S_L)}, 1]$$

- Diagonal update: $[0,0]_p \leftrightarrow [1,b]_p$





$n \to$ current power,

$P_{select}(a=0 \to a=1) = 1/N_b, P_{select}(a=1 \to a=0) = 1$

$$\frac{W(a=1)}{W(a=0)} = \frac{\beta/2}{L-n}, \qquad \frac{W(a=0)}{W(a=1)} = \frac{L-n+1}{\beta/2},$$

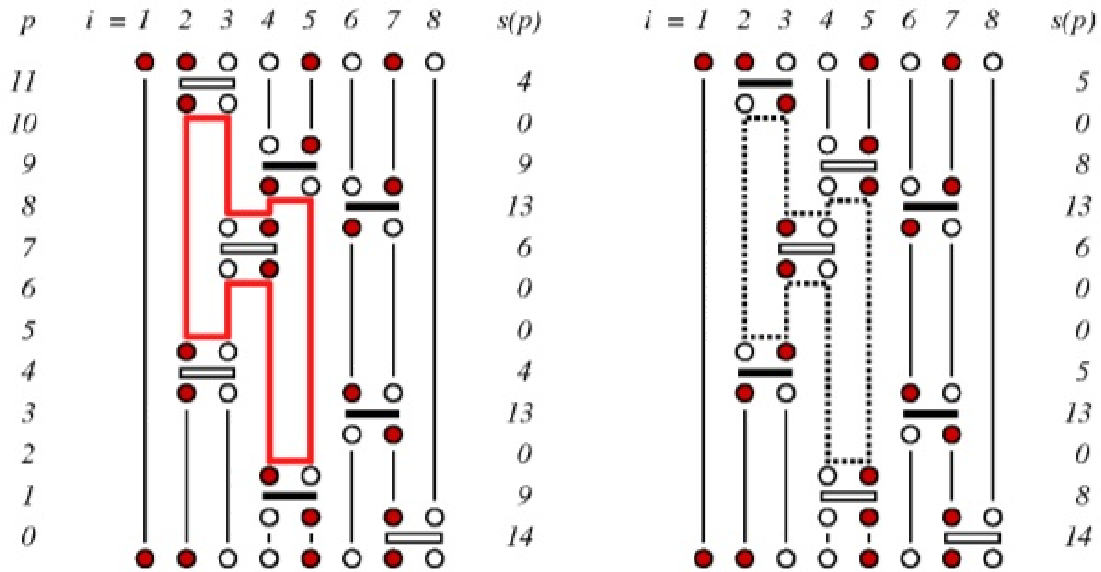
- $n \to n+1$ $(a=0 \to a=1)$
- $n \to n-1$ $(a=1 \to a=0)$

acceptance prob. $P_{accept}([0,0] \to [1,b]) = \min[\frac{\beta N_b}{2(L-n)}, 1]$

$$P_{accept}([1,b] \to [0,0]) = \min[\frac{2(L-n+1)}{\beta N_b}, 1]$$

- 对角演化不会改变非对角算符, 非遍历 !
- 我们需要另外的方式来实现非对角算符$(\Gamma) \Longleftrightarrow (\Gamma')$ 对角算符
- 考虑到这一过程中位形权重不变，可以实现非常简单的集团算法：loop update

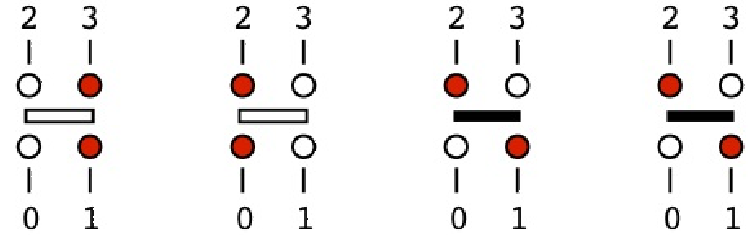
$$W(\Gamma = \{\alpha, S_L\}) = W(\Gamma' = \{\alpha', S'_L\}) = (\frac{\beta}{2})^n \frac{(L-n)!}{L!}$$

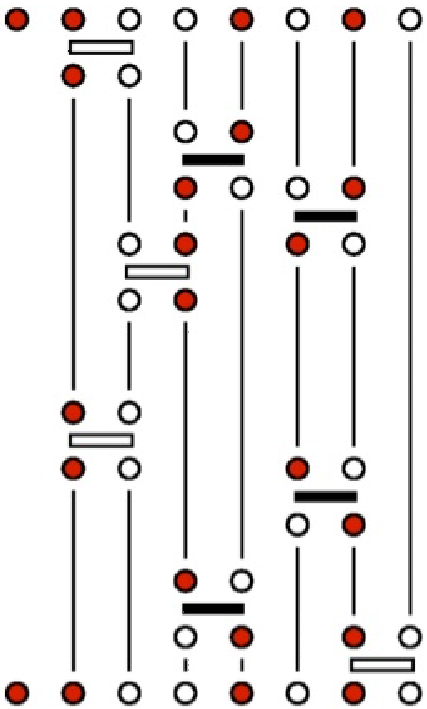


Operator-loop update

Linked vertex storage

Legs of a vertex represents the spin states before and after an operator has acted





p	v X(v)		v X(v)		v X(v)		v X(v)	
11	44	18	45	30	46	16	47	17
10	40	-	41	-	42	-	43	-
9	36	31	37	7	38	4	39	5
8	32	14	31	15	34	12	35	0
7	28	19	29	6	30	45	31	36
6	24	-	25	-	26	-	27	-
5	20	-	21	-	22	-	23	-
4	16	46	17	47	18	44	19	28
3	12	34	13	2	14	32	15	33
2	8	-	9	-	10	-	11	-
1	4	38	5	39	6	29	7	37
0	0	35	1	3	2	13	3	1
	l=0		l=1		l=2		l=3	

$X(\ )$: vertex list

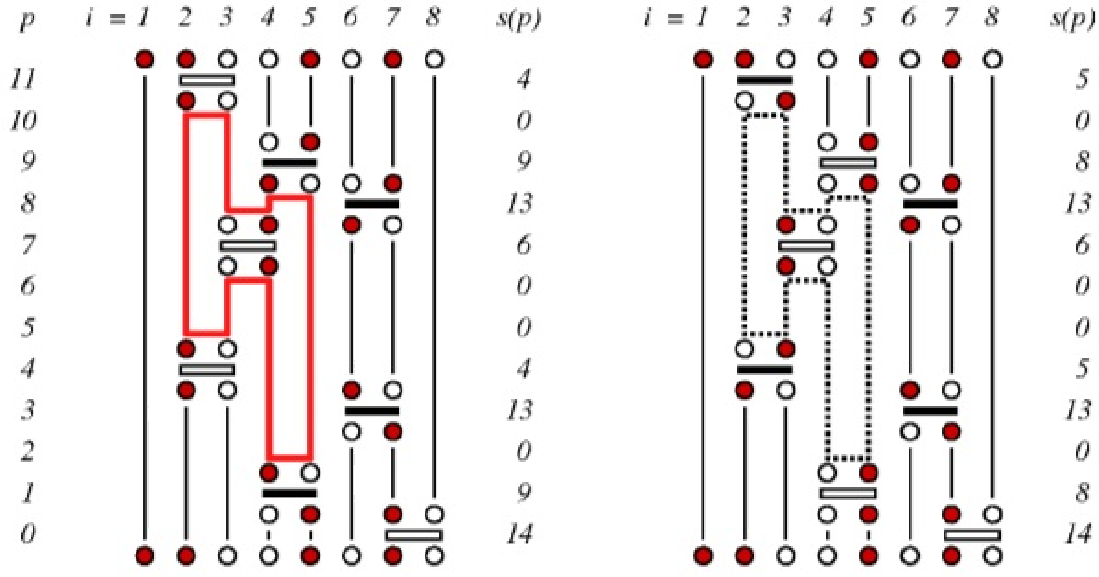
- operator at $p \rightarrow X(v)$ $v = 4p + l, l = 0, 1, 2, 3$
- links to next and previous leg

Spin states between operations are represented by links

network of linked vertices will be used for loop updates of operator strings

Operator-loop update

Many spins and operators can be changed simultaneously



Use link $X(v)$
moving horizontally
corresponds to changing v
even $\leftrightarrow$ odd
Label vertices

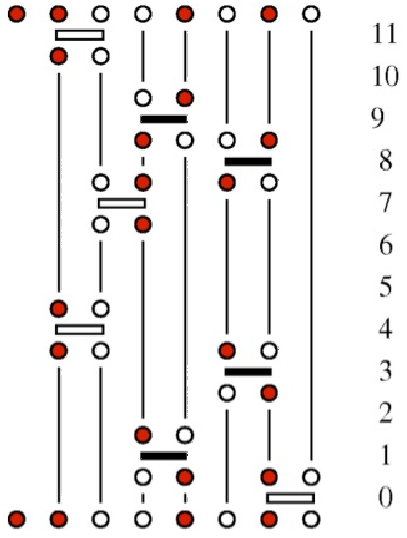
- a given loop is only constructed once
- $X(v) = -2$: flipped loop
- $X(v) = -1$: not flipped loop

constructing all loops, flip with probability $1/2$; save the changed $s(p)$

modify the stored spin state

使用数组跟踪连接每个自旋
的第一条腿和最后一条腿

- $V_{\text{first}}(i)$ = location v of first leg on site i
- $V_{\text{last}}(i)$ = location v of last (currently) leg
- initialize all elements to -1
  - spins with no operators, $V_{\text{first}} = -1$, flipped with probability $1/2$
  - if $X(V_{\text{first}}(i)) = -2$, flip it
  - if $X(V_{\text{first}}(i)) = -1$, no operation



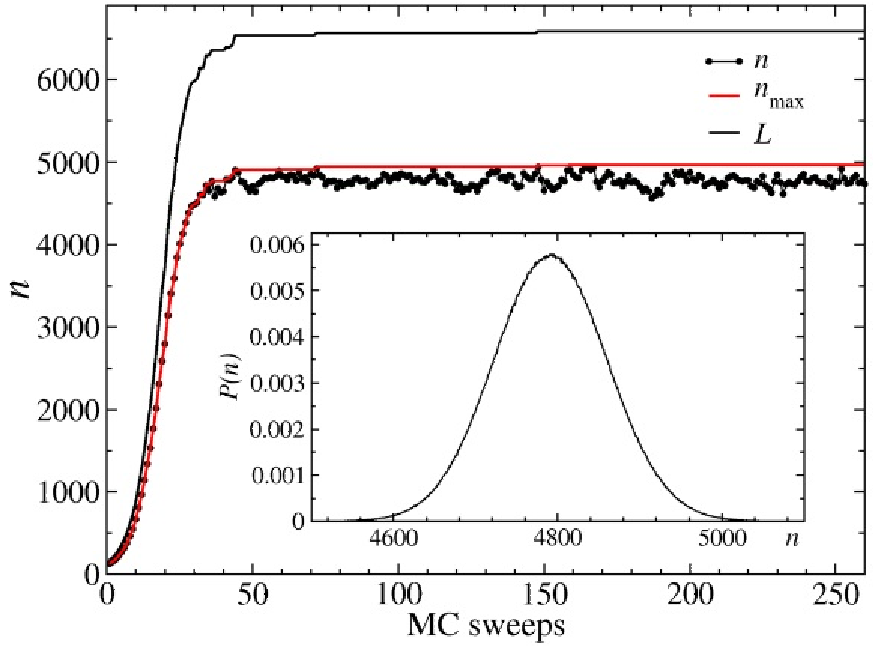
v	X(v)	v	X(v)	v	X(v)	v	X(v)
44	18	45	30	46	16	47	17
40	-	41	-	42	-	43	-
36	31	37	7	38	4	39	5
32	14	31	15	34	12	35	0
28	19	29	6	30	45	31	36
24	-	25	-	26	-	27	-
20	-	21	-	22	-	23	-
16	46	17	47	18	44	19	28
12	34	13	2	14	32	15	33
8	-	9	-	10	-	11	-
4	38	5	39	6	29	7	37
0	35	1	3	2	13	3	1
l=0		l=1		l=2		l=3	

Determination of the cut-off L

Adjust during equilibration

- start with arbitrary (small) n
- Keep track of number of operators n
  - increase L if n is close to current L: $L = \frac{4}{3}n$

- Example: 16×16 system, $\beta = 16$
  - evolution of L
  - n distribution after equilibration



程序下载

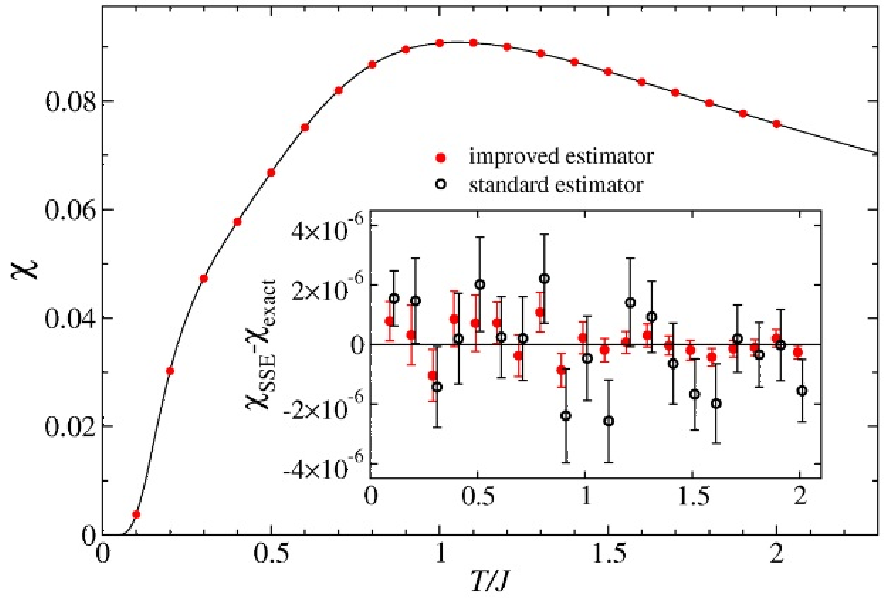
Anders Sandvik 提供了
A basic SSE program (Fortran 90) for the 2D Heisenberg model
http://physics.bu.edu/ sandvik/programs/index.html

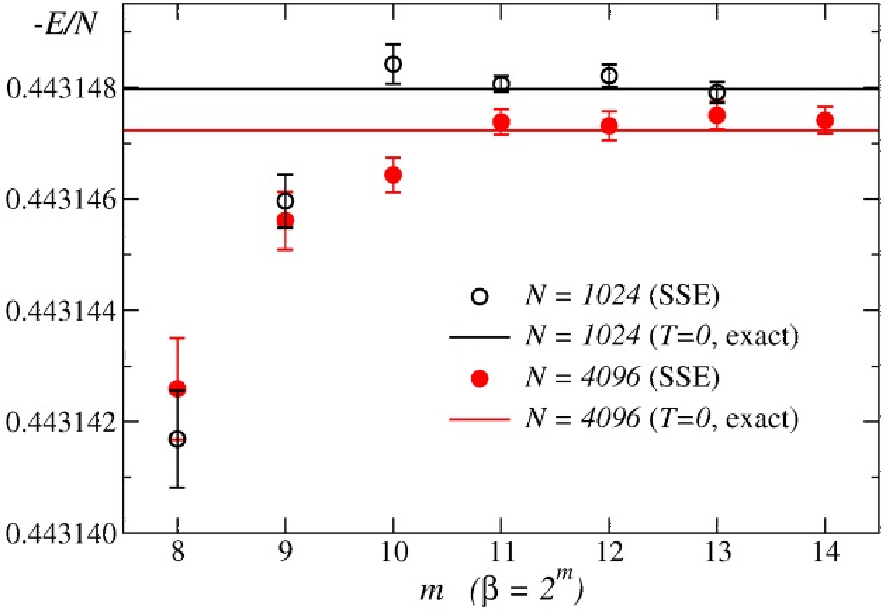
Compare with exact results

Susceptibility of the 4×4 lattice

- SSE results from 10^{10} sweeps

$$\chi = \frac{\beta}{N}\langle M_z^2 \rangle, M_z = \sum S_i^z$$





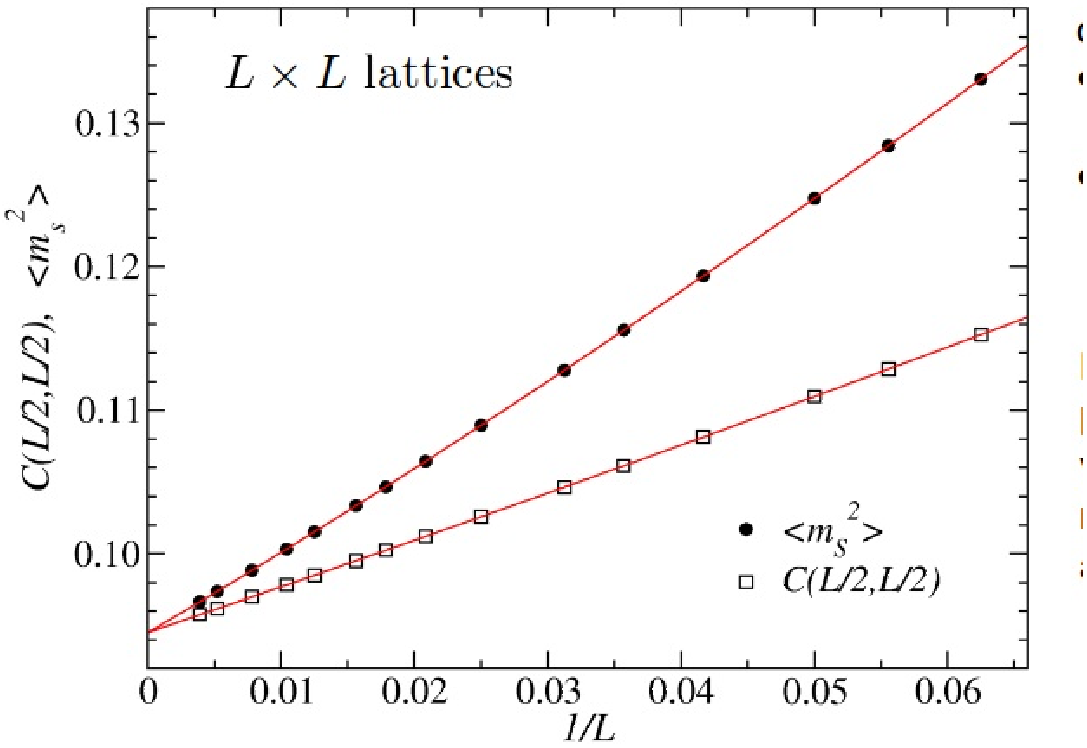
Beta Ansatz; Energy for long chains

- SSE results for 10^6 sweeps
- Beta Ansatz ground state E/N

2D Heisenberg model

Long-range order at $T = 0$; magnetization

- Spin-wave theory prediction: $\mathbf{m_s} = 0.3034$
- SSE gives $\mathbf{m_s} = 0.3074$



- 反铁磁磁化强度 $\mathbf{m_s} = \frac{1}{N}\sum_i (-1)^{x_i+y_i}\mathbf{S}_i$ 长程序 $\langle m_s^2 \rangle > 0$, for $N \to \infty$
- $C(L/2, L/2)$ the spin correlation fuction at the longest distance in a finite system

Thank you!