

相变与临界现象研究中蒙特卡洛方法介绍

郭文安

北京师范大学物理系

2016 广州

参考文献:

- H.W.J. Blöte, Lecture notes: Nonlocal Monte carlo Methods
- A. W. Sandvik, <http://physics.bu.edu/~sandvik>

提纲

多体系统与统计物理基础

Ising Model 的Monte Carlo 模拟

1. 简单抽样
2. 重要性抽样
3. 模拟结果与分析, 有限尺寸标度(finite-size scaling)
4. cluster算法
5. Wang-Landau algorithm

Quantum Monte Carlo simulations

1. Path integral representation
2. SSE QMC method
 - 2(a). Heisenberg model
 - 2(b). Bose-Hubbard (XXZ) model
3. Projector Quantum Monte Carlo methods

提纲

多体系统与统计物理基础

Ising Model 的Monte Carlo 模拟

1. 简单抽样
2. 重要性抽样
3. 模拟结果与分析, 有限尺寸标度(finite-size scaling)
4. cluster算法
5. Wang-Landau algorithm

Quantum Monte Carlo simulations

1. Path integral representation
2. SSE QMC method
 - 2(a). Heisenberg model
 - 2(b). Bose-Hubbard (XXZ) model
3. Projector Quantum Monte Carlo methods

多粒子系统

▶ 微观状态

- 经典系统，可认为任意力学量与 H 对易，每个微观状态 Γ 有确定能量 $E(\Gamma) = H(\Gamma)$

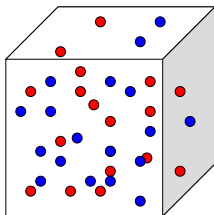
● 平衡态配分函数(partition function)

$$Z = \sum_{\Gamma} e^{-E(\Gamma)/k_B T} = \sum_{\Gamma} W(\Gamma)$$

每个微观状态出现的几率由温度和该状态的能量决定: 正则分布(**Canonical distribution**)

$$p_{\text{eq}}(\Gamma) = W(\Gamma)/Z$$

物理量的统计期望值: $\langle A \rangle = \sum_{\Gamma} A(\Gamma) \frac{W(\Gamma)}{Z}$



多粒子系统

▶ 微观状态

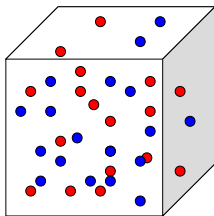
- 量子系统，哈密顿量的本征态不容易知道

● 平衡态配分函数

$$Z = \text{Tr} e^{-H/k_B T} = \sum_{\alpha} \langle \alpha | e^{-\beta H} | \alpha \rangle$$

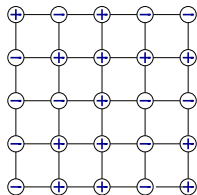
$|\alpha\rangle$ 是任意正交完备基矢
物理量的统计期望值

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



Ising model (经典)

描写单轴铁磁体（可以在任意晶格上，以二维正方晶格为例）



$$H = -J \sum_{\langle i,j \rangle} s_i s_j - B \sum_k s_k; \quad s_k = \pm 1$$

微观状态 $\Gamma = (s_1, s_2, \dots, s_N)$.

总磁矩 $M(\Gamma) = \sum_k s_k$

- $J > 0$, 铁磁(ferromagnetic)
- $J < 0$, 反铁磁(antiferromagnetic)

当系统处于热平衡

$$\langle M \rangle = \sum_{\Gamma} M(\Gamma) p_{eq}(\Gamma)$$

- ▶ 在磁性和相变理论中非常重要
- ▶ 也是其它统计物理问题的有效模型：格气(lattice gas), 合金, 原子在表面的吸附问题等

提纲

多体系统与统计物理基础

Ising Model 的Monte Carlo 模拟

1. 简单抽样
2. 重要性抽样
3. 模拟结果与分析, 有限尺寸标度(finite-size scaling)
4. cluster算法
5. Wang-Landau algorithm

Quantum Monte Carlo simulations

1. Path integral representation
2. SSE QMC method
 - 2(a). Heisenberg model
 - 2(b). Bose-Hubbard (XXZ) model
3. Projector Quantum Monte Carlo methods

为什么要利用蒙特卡洛方法?

Monte Carlo simulation: 直接求和或积分不可能或不容易时的办法

以Ising model 为例, 考虑任意物理量 A , 计算它的统计平均

$$\langle A \rangle = \sum_{\Gamma} A(\Gamma)p(\Gamma)$$

直接求和不现实: 2^N 个微观状态 (位形)

1. 简单抽样(simple sampling)

简单想法：任意从 2^N 个位形中**等概率**任意选 M 个位形来近似

$$\langle A \rangle \approx A_M = \sum_l^M A(\Gamma_l) \frac{e^{-E(\Gamma_l)/k_B T}}{\sum_l e^{-E(\Gamma_l)/k_B T}}$$

称为简单抽样. 但是

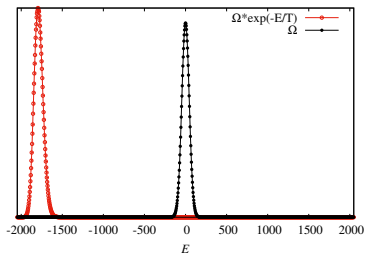
一般情况下，如果系统够大，温度不是非常高，这种抽样过程效率非常低

- 以计算内能为例

$$U \equiv \langle H \rangle = \sum_{\Gamma} H(\Gamma)p(\Gamma) = \sum_E E \frac{\Omega(E)e^{-E/k_B T}}{Z} = \sum_E E \frac{W(E)}{Z}$$

$$Z = \sum_E \Omega(E)e^{-E/k_B T} = \sum_E W(E),$$

$\Omega(E)$ 为能量为 E 的态的数目, $W(E) = \Omega(E)e^{-E/k_B T}$ 是权重



$L = 32$ 2D Ising, $T = 2.0$

- $W(E)$ 在 \bar{E} 处形成非常尖锐的峰值
- 等概率选 Γ 实际是按 $\Omega(E)$ 权重选 E . $\Omega(E)$ 峰随系统的增大也变得尖锐, 但峰值不在 \bar{E} .
- 极端情况: $T \rightarrow 0$. 只有两个位形对内能有贡献, 但是产生出这两个位形的几率是 $2/2^N$
- 除非抽取非常大量 ($\approx 2^N$) 的样本, 否则误差巨大

2. 重要性抽样(Importance sampling)

[Metropolis, Rusenbluth, Rosenbluth, Teller, and Teller, Phys.Rev.1953]

- 构造一个**随机过程**，得到一系列微观状态 $\Gamma_1, \Gamma_2, \dots, \Gamma_M$.

Γ_{t+1} 由 Γ_t 按照一定的跃迁几率(transition prob.) $T(\Gamma_{t+1}, \Gamma_t)$ 得到

当 $M \rightarrow \infty$, 任一给定位形 Γ 出现的**频率** $\frac{N(\Gamma)}{M} = \frac{e^{-E(\Gamma)/k_B T}}{Z}$.

- 实现按位形的**正则分布概率**来抽取位形，而不是在相空间等概率地抽取位形

$$\langle A \rangle \approx A_M = \sum_{\Gamma} \frac{N(\Gamma)}{M} A(\Gamma) = \frac{1}{M} \sum_l^M A(\Gamma_l)$$

怎样实现这样的序列？

要实现这种序列，关键是选取 $T(\Gamma', \Gamma)$ ：更新 Γ 得到 Γ' 的几率

- 满足归一化条件 $\sum_{\Gamma'} T(\Gamma', \Gamma) = 1$

定义 $P_t(\Gamma)$ 为第 t 个微观状态为 Γ 的几率，我们有

$$P_{t+1}(\Gamma') = \sum_{\Gamma} T(\Gamma', \Gamma) P_t(\Gamma)$$

此为主方程。这样的过程称为**Markov过程**。

写成矢量形式

$$\vec{P}_t = \mathbf{T} \cdot \vec{P}_{t-1} = \mathbf{T}^t \cdot \vec{P}_0$$

可以预期

$$t \rightarrow \infty, \quad P_t(\Gamma) \rightarrow P_{eq}(\Gamma)$$

我们的目标:对于任意初始几率分布 $P_0(\Gamma)$

$$P_{eq}(\Gamma) = \frac{e^{-E(\Gamma)/k_B T}}{Z},$$

意味着 \vec{P}_{eq} 是 \mathbf{T} 的本征矢:

$$\vec{P}_{eq} = \mathbf{T} \cdot \vec{P}_{eq}$$

可以证明满足两个条件即可实现

- ▶ 各态历经(ergodic): 任意一个微观状态可以通过一系列的跃迁来达到(矩阵不可约)
- ▶ 细致平衡(detailed balancing)

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

即从 Γ 流到 Γ' 的几率等于反过来从 Γ' 流到 Γ 的

- 由于 \mathbf{T} 非负, 不可约, 其最大本征值 $|\lambda_1|$ 的右本征矢所有元素为正, 其余本征矢元素有正负(正交性)
- 利用细致平衡和归一性, \vec{P}_{eq} 就是最大本征矢, 本征值为1:

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

- 由于 \mathbf{T} 非负, 不可约, 其最大本征值 $|\lambda_1|$ 的右本征矢所有元素为正, 其余本征矢元素有正负(正交性)
- 利用细致平衡和归一性, \vec{P}_{eq} 就是最大本征矢, 本征值为1:

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

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

需要多长时间实现正则分布?

对本征值排序 $\lambda_1 = 1 > |\lambda_2| \geq |\lambda_3| \geq \dots$

记 λ_i 对应本征矢为 \vec{R}_i ; 展开 \vec{P}_0

$$\vec{P}_0 = \sum_i c_i \vec{R}_i$$

t 步之后

$$\begin{aligned} \vec{P}_t &= \sum_i c_i \mathbf{T}^t \cdot \vec{R}_i = \sum_i c_i \lambda_i^t \vec{R}_i = c_1 \vec{R}_1 + \sum_{i=2} c_i \lambda_i^t \vec{R}_i \\ &= c_1 \vec{R}_1 + c_2 e^{-t/\tau_2} \vec{R}_2 + \dots \end{aligned}$$

- $1/\tau_2 = -\ln \lambda_2$: 平衡关联时间 (equilibrium correlation time)

$$\vec{P}_t \rightarrow \vec{P}_{eq}, \text{ 当 } t \gg \tau_2!$$

实现细致平衡

我们的目标是实现

$$P_{eq}(\Gamma) = \frac{e^{-E(\Gamma)/T}}{Z} = \frac{W(\Gamma)}{Z}$$

根据细致平衡

$$T(\Gamma', \Gamma)W(\Gamma) = T(\Gamma, \Gamma')W(\Gamma')$$

跃迁几率可以分两步完成

$$T(\Gamma', \Gamma) = P_{\text{accept}}(\Gamma', \Gamma)P_{\text{attempt}}(\Gamma', \Gamma)$$

- $P_{\text{attempt}}(\Gamma', \Gamma)$: 从各种可能的新位形里选择 Γ' 的几率
- $P_{\text{accept}}(\Gamma', \Gamma)$: 接受新位形 Γ' 的几率

实现细致平衡：两种常见选择

保证细致平衡的两种常见方式

Metropolis:

$$P_{\text{accept}}(\Gamma', \Gamma) = \min\left[\frac{P_{\text{attempt}}(\Gamma, \Gamma')W(\Gamma')}{P_{\text{attempt}}(\Gamma', \Gamma)W(\Gamma)}, 1\right]$$

Heat bath:

$$P_{\text{accept}}(\Gamma', \Gamma) = \frac{P_{\text{attempt}}(\Gamma, \Gamma')W(\Gamma')}{P_{\text{attempt}}(\Gamma, \Gamma')W(\Gamma') + P_{\text{attempt}}(\Gamma', \Gamma)W(\Gamma)}$$

实现细致平衡：两种常见选择

保证细致平衡的两种常见方式

- local update Algorithm (局域更新算法):

从 N 个自旋中挑选一个自旋尝试翻转:

$P_{\text{attempt}}(\Gamma', \Gamma) = P_{\text{attempt}}(\Gamma, \Gamma') = 1/N$, 此时上式化简为

- Metropolis:

$$P_{\text{accept}}(\Gamma', \Gamma) = \min\left[\frac{W(\Gamma')}{W(\Gamma)}, 1\right]$$

- Heat bath:

$$P_{\text{accept}}(\Gamma', \Gamma) = \frac{W(\Gamma')}{W(\Gamma') + W(\Gamma)}$$

这就是通常所谓单自旋蒙特卡罗模拟算法

超越细致平衡

- 要实现重要性抽样，细致平衡是充分的，但并不是必要的

参考文献：

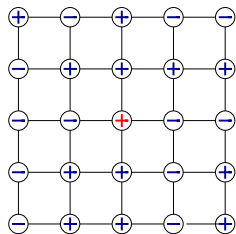
H. Suwa, and S. Todo, Phys. Rev. Lett. 105, 120603 (2010)

Markov Chain Monte Carlo Method without Detailed Balance

Metropolis algorithm: 以Ising model为例

算法:

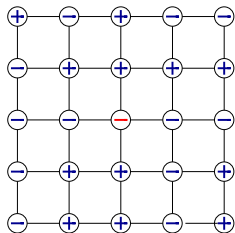
- 1 在位形 $\Gamma = (s_1, s_2, \dots, s_i, \dots, s_N)$ 中任选一个自旋, 比如 s_i .



Metropolis algorithm: 以Ising model为例

算法:

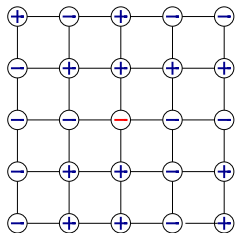
- 1 在位形 $\Gamma = (s_1, s_2, \dots, s_i, \dots, s_N)$ 中任选一个自旋, 比如 s_i .
- 2 翻转它, $\Gamma' = (s_1, s_2, \dots, -s_i, \dots, s_N)$. 这就是按 $p_{\text{attempt}} = \frac{1}{N}$ 的几率选择一个新位形



Metropolis algorithm: 以Ising model为例

算法:

- 1 在位形 $\Gamma = (s_1, s_2, \dots, s_i, \dots, s_N)$ 中任选一个自旋, 比如 s_i .
- 2 翻转它, $\Gamma' = (s_1, s_2, \dots, -s_i, \dots, s_N)$. 这就是按 $p_{\text{attempt}} = \frac{1}{N}$ 的几率选择一个新位形
- 3 计算 $\Delta E = E(\Gamma') - E(\Gamma)$, 得到
$$\frac{W(\Gamma')}{W(\Gamma)} = e^{-\Delta E/k_B T}$$



Metropolis algorithm: 以Ising model为例

算法:

1 在位形 $\Gamma = (s_1, s_2, \dots, s_i, \dots, s_N)$ 中任选一个自旋, 比如 s_i .

2 翻转它, $\Gamma' = (s_1, s_2, \dots, -s_i, \dots, s_N)$. 这就是按 $p_{\text{attempt}} = \frac{1}{N}$ 的几率选择一个新位形

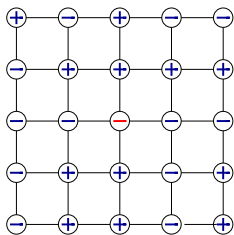
3 计算 $\Delta E = E(\Gamma') - E(\Gamma)$, 得到

$$\frac{W(\Gamma')}{W(\Gamma)} = e^{-\Delta E/k_B T}$$

4 设定 $P_{\text{accept}}(\Gamma', \Gamma) = \min\left[\frac{W(\Gamma')}{W(\Gamma)}, 1\right]$:

▶ 如果 $\Delta E \leq 0$, 接受 Γ' ;

▶ 如果 $\Delta E > 0$, 按几率 $e^{-\Delta E/k_B T}$ 接受 Γ' ; 否则拒绝 Γ' (或者说取 $\Gamma' = \Gamma$).



Metropolis algorithm: 以Ising model为例

算法:

1 在位形 $\Gamma = (s_1, s_2, \dots, s_i, \dots, s_N)$ 中任选一个自旋, 比如 s_i .

2 翻转它, $\Gamma' = (s_1, s_2, \dots, -s_i, \dots, s_N)$. 这就是按 $p_{\text{attempt}} = \frac{1}{N}$ 的几率选择一个新位形

3 计算 $\Delta E = E(\Gamma') - E(\Gamma)$, 得到

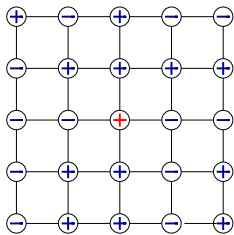
$$\frac{W(\Gamma')}{W(\Gamma)} = e^{-\Delta E/k_B T}$$

4 设定 $P_{\text{accept}}(\Gamma', \Gamma) = \min\left[\frac{W(\Gamma')}{W(\Gamma)}, 1\right]$:

▶ 如果 $\Delta E \leq 0$, 接受 Γ' ;

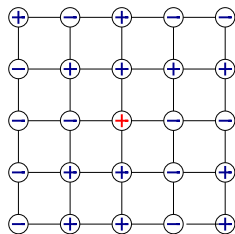
▶ 如果 $\Delta E > 0$, 按几率 $e^{-\Delta E/k_B T}$ 接受 Γ' ; 否则拒绝 Γ' (或者说取 $\Gamma' = \Gamma$).

5 回到1, 循环到足够的位形产生.



Metropolis algorithm: 以Ising model为例

算法:



- 1 在位形 $\Gamma = (s_1, s_2, \dots, s_i, \dots, s_N)$ 中任选一个自旋, 比如 s_i .
- 2 翻转它, $\Gamma' = (s_1, s_2, \dots, -s_i, \dots, s_N)$. 这就是按 $p_{\text{attempt}} = \frac{1}{N}$ 的几率选择一个新位形
- 3 计算 $\Delta E = E(\Gamma') - E(\Gamma)$, 得到 $\frac{W(\Gamma')}{W(\Gamma)} = e^{-\Delta E/k_B T}$
- 4 设定 $P_{\text{accept}}(\Gamma', \Gamma) = \min[\frac{W(\Gamma')}{W(\Gamma)}, 1]$:
 - ▶ 如果 $\Delta E \leq 0$, 接受 Γ' ;
 - ▶ 如果 $\Delta E > 0$, 按几率 $e^{-\Delta E/k_B T}$ 接受 Γ' ; 否则拒绝 Γ' (或者说取 $\Gamma' = \Gamma$).
- 5 回到1, 循环到足够的位形产生.

▶ P_{accept} 也可以选为热浴(Heat Bath)法:

$$P_{\text{accept}}(\Gamma', \Gamma) = \frac{e^{-E(\Gamma')/k_B T}}{e^{-E(\Gamma')/k_B T} + e^{-E(\Gamma)/k_B T}} = \frac{1}{1 + e^{\Delta E/k_B T}}$$

- MC 时间: 一个Monte Carlo 步(1 Monte Carlo step)指进行完 N 次自旋翻转尝试(平均每个自旋尝试一次翻转)
- 在实现平衡后开始测量观测量(确保 $t > \tau_2$)
- Bining: 把 M 步MC叫作一个bin, 最终计算bin平均和统计误差
- 程序流程
 - ▶ 生成任意的一个初始态
 - ▶ 执行一定的蒙特卡罗步(实现平衡)
 - ▶ 运行一定数目的bins
 - ▶ 每个bin包含 M 步
 - ▶ 在每一MC步(或者很少几步)后测量物理量
 - ▶ 完成每一个bin后保存bin平均
 - ▶ 计算最后的平均和统计误差

show

提纲

多体系统与统计物理基础

Ising Model 的Monte Carlo 模拟

1. 简单抽样
2. 重要性抽样
3. 模拟结果与分析, 有限尺寸标度(finite-size scaling)
4. cluster算法
5. Wang-Landau algorithm

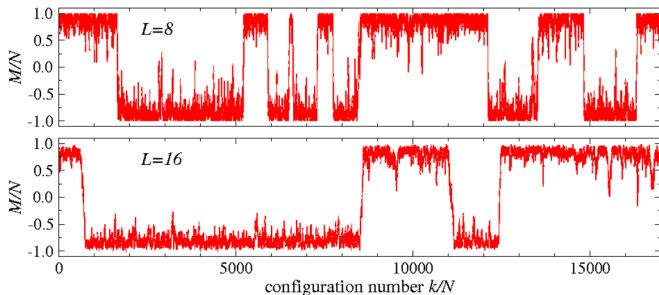
Quantum Monte Carlo simulations

1. Path integral representation
2. SSE QMC method
 - 2(a). Heisenberg model
 - 2(b). Bose-Hubbard (XXZ) model
3. Projector Quantum Monte Carlo methods

磁化强度的时间演化: 对称性的破缺

$$T/k_B T = 2.2 < T_c$$

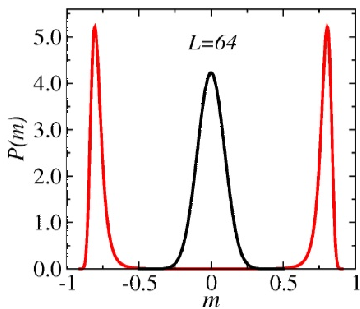
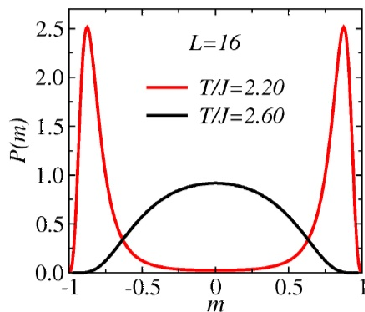
- $\langle m \rangle = 0$, 但是 m 反转所需时间随尺寸增加
- 对于足够大的系统, 上下对称发生**破缺**!



- 磁化强度及其分布依赖于温度和尺寸（有限尺寸效应）

- ▶ $T > T_c$ 在 $m = 0$ 附近形成单峰
- ▶ $T < T_d$ 在 $\pm \bar{m}$ 形成对称双峰

- ▶ m 分布



- **对称破缺：** 模拟中大尺寸系统只能测到一个峰，非常小的几率在两峰间跃迁

测量物理观测量

- 磁化强度为序参量(order parameter)

$$m = \frac{M}{N} = \frac{\sum_i^N s_i}{N}$$

但是它的统计平均（期望值）在有限系统里等于零，因为对称性只在热力学极限下真正破缺

如何在有限尺寸下区分对称破缺相与对称相？

模拟中我们计算 $\langle |m| \rangle$ ，或者 $\langle m^2 \rangle$

测量物理观测量

- 磁化率

$$H = -J \sum_{\langle i,j \rangle} s_i s_j - B \sum_k s_k; \quad s_k = \pm 1$$

$$\begin{aligned} \chi &= \frac{\partial m}{\partial B} = \frac{1}{N} \frac{\partial}{\partial B} \sum_{\Gamma} \frac{M e^{-\beta H}}{Z} \\ &= \frac{\beta}{N} \left(\frac{\sum_{\Gamma} M^2 e^{-\beta H}}{Z} - \left(\frac{\sum_{\Gamma} M e^{-\beta H}}{Z} \right)^2 \right) \\ &= \frac{1}{k_B T N} (\langle M^2 \rangle - \langle M \rangle^2) \end{aligned}$$

即正比于磁矩的涨落

考虑到有限尺寸下 $\langle M \rangle = N \langle m \rangle = 0$, 模拟中我们计算

$$\chi = \frac{N}{k_B T} \langle m^2 \rangle$$

这就是MC中磁化率的estimator

测量物理观测量

- 比热

$$c = \frac{1}{N} \frac{dU}{dT} = \frac{1}{N} \frac{d}{dT} \frac{\sum_{\Gamma} E(\Gamma) e^{-E(\Gamma)/T}}{Z} = \frac{1}{NT^2} (\langle E^2 \rangle - \langle E \rangle^2)$$

表示成能量密度的函数

$$c = \frac{N}{T^2} (\langle e^2 \rangle - \langle e \rangle^2)$$

这就是MC中比热的estimator

计算统计平均和误差

具体计算：考虑 M 个 “bins”，每个包含 n 个测量位形

由于测量位形出现几率（频率）正比于正则分布，有

$$A_i = \frac{1}{n} \sum_l^n A(\Gamma_l) \approx \langle A \rangle.$$

A_i 是第 i 个 bin 的测量平均值

计算统计平均和误差

我们有统计独立的平均 $A_i, i = 1, \dots, M$, 计算它们的平均值

$$\bar{A} = \frac{1}{M} \sum_i^M A_i$$

A_i 的标准偏差为

$$\sigma' = \sqrt{\frac{1}{M} \sum_i^M (A_i - \bar{A})^2} = \sqrt{(\bar{A}^2 - \bar{A}^2)}$$

我们关心的是 \bar{A} 的标准偏差: 误差; 基于中心极限定理

$$\sigma = \frac{\sigma'}{\sqrt{M-1}}$$

成立条件: 要保证 A_i 统计独立, 即每个bin需要足够长

什么叫足够长? 我们要考虑 ‘自关联时间(auto-correlation length)’

程序实现

- Main program

```
open(1,file='ising.in',status='old')
read(1,*)L,temp, initsteps, bins, binsteps
close(1)
call initialize(temp)
do i=1,initsteps
    call mcstep
enddo
do j=1,bins
    call cleandata
    do i=1,binsteps
        call mcstep
        call measure
    enddo
    call writebindata(n,binsteps)
enddo
```

系统变量可以用Module 描述

Module systemvariables

integer :: L ! 尺寸

integer :: N ! 自旋数 $N = L * L$

real(8) :: pflip(-4:4) ! 事先计算的
接受几率

integer, allocatable :: spin(:) ! 自
旋数组, 可调

end module systemvariables

程序实现

- Initialization tasks

```
subroutine initialize(temp)
```

```
use systemvariables
```

```
 $N = L * L$ 
```

```
do  $i = -4, 4, 2$ 
```

```
  pflip( $i$ ) = exp( $-i * 2./temp$ )
```

```
enddo
```

```
call init_random(seed)
```

```
allocate(spin(0:N-1))
```

```
do  $i=0, N-1$ 
```

```
  call random_number(r)      ! spins are labeled  $s = 0, \dots, N - 1$ 
```

```
  spin( $i$ ) = 2 * int(2. * r) - 1 ! spin( $s$ ) 是  $s$  的状态 =  $\pm 1$ 
```

```
enddo
```

! 计算接受概率数组:

4个紧邻自旋的状态有:

4 正, 3正1负, 2正2负, 1正3负, 4负.

$s_1 + s_2 + s_3 + s_4 = 4, 2, 0, -2, -4$

如果被选中自旋为 s , 那么

令 $i = s(s_1 + s_2 + s_3 + s_4)$,

$\Delta E = 2 * i$

程序实现

- Performs one Monte Carlo step

```
subroutine mcstep()
```

```
do i=1,n
```

```
call random_number(r)
```

```
s=int(r*n) ! 随机挑选一个自旋s
```

```
x=mod(s,L); y=s/L !计算横纵坐标
```

```
s1=spin(mod(x+1,L)+y*L) ! 计算4个最近邻, 可事先计算保存在数组中
```

```
s2=spin(mod(x-1+L,L)+y*L)
```

```
s3=spin(x+mod(y+1,L)*L)
```

```
s4=spin(x+mod(y-1+L,L)*L)
```

```
call random_number(r)
```

```
if (r<=pflip(spin(s)*(s1+s2+s3+s4))) spin(s)=-spin(s)
```

```
! 根据随机数r和跃迁几率pflip( $i = \Delta E/2$ )决定是否接受  $s \rightarrow -s$ 
```

```
enddo
```

程序实现

- Measures observables (energy, magnetization, and their squares)

subroutine measure

```
real(8) :: enrg1,enrg2,magn1,magn2
```

```
common/measurments/enrg1,enrg2,magn1,magn2
```

```
e=0; m=0
```

```
do s=0,n-1
```

```
x=mod(s,L); y=s/L
```

```
e=e-spin(s)*(spin(mod(x+1,L)+y*L))
```

```
e=e-spin(s)*(spin(x+mod(y+1,L)*L))
```

```
enddo
```

```
m=sum(spin)
```

```
enrg1=enrg1+dbble(e) !累计能量
```

```
enrg2=enrg2+dbble(e)**2 ! 累计能量平方
```

```
magn1=magn1+abs(m) ! 累计磁化强度
```

```
magn2=magn2+dbble(m)**2 ! 累计磁化强度平方
```

程序实现

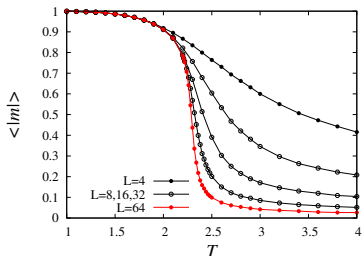
- Writes bin averages to a file

```
subroutine writebindata(n,steps)
real(8) :: enrg1,enrg2,magn1,magn2
common/measurements/enrg1,enrg2,magn1,magn2 !公用块, 可用module代替
open(1,file='bindata.dat',position='append')
!以下计算统计平均
enrg1=enrg1/(dble(steps)*dble(n))
enrg2=enrg2/(dble(steps)*dble(n)**2)
magn1=magn1/(dble(steps)*dble(n))
magn2=magn2/(dble(steps)*dble(n)**2)
write(1,'(4f18.12)')enrg1,enrg2,magn1,magn2
close(1)
```

- These bin data should be processed (giving final averages and statistical errors) with a separate program.

自发磁化，有限尺寸行为

- Ising model的自发磁化

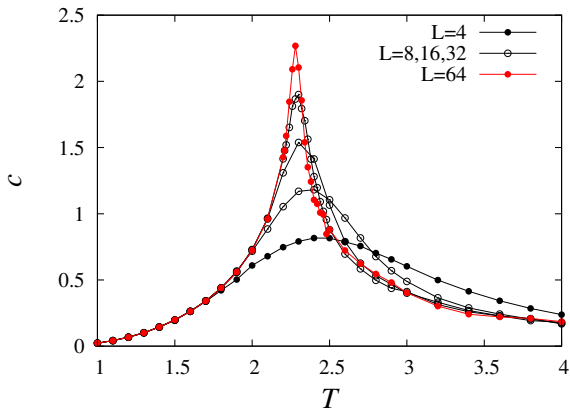


热力学极限下理论公式($T_c = 2.269..$)

$$\langle m \rangle = \left(1 - \frac{1}{\sinh^4(2/T)}\right)^{1/8}, \quad T \leq T_c$$

$$\langle m \rangle = 0, \quad T > T_c$$

比热



$$c \propto |T - T_c|^\alpha$$

临界指数的计算: 临界指数 β

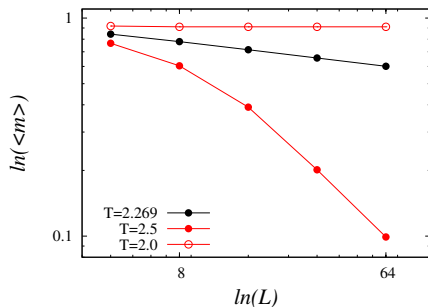
热力学极限下

$$\langle m \rangle \propto (T_c - T)^\beta \propto \xi^{-\beta/\nu}, \quad \beta = \frac{1}{8} \quad T < T_c$$

有限系统, 在 $T = T_c$, 可以理解为用 L 替代 ξ

$$\langle |m| \rangle \propto L^{-\beta/\nu} + \dots$$

- 利用这一关系, 在 T_c 模拟一系列尺寸系统, 我们可以定出 β
(作对数图: $\ln |m|$ vs. $\ln L$)



临界点的确定: Binder ratio

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

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

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

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

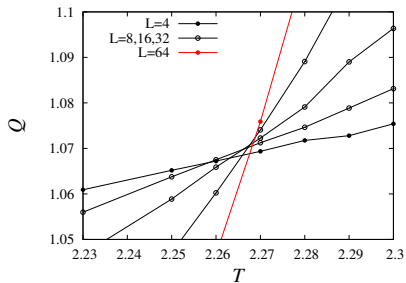
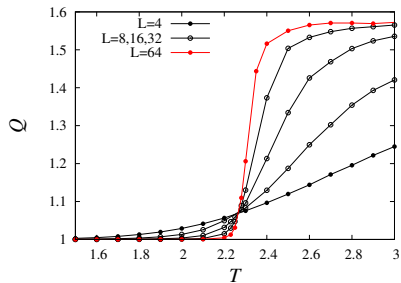
$$\langle m^2 \rangle \propto L^{\gamma/\nu-d} \quad \langle |m| \rangle \propto L^{\gamma/(2\nu)-d/2}$$

Q 在临界点与尺寸无关!

$Q \rightarrow 1$ 当 $T \rightarrow 0$; $Q \rightarrow \text{constant}$ 当 $T \rightarrow \infty$, 由于 m 的高斯分布

临界点的确定: Binder ratio

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



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

自关联时间

设某个物理观测量在第 t MC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

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

Integrated 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)

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

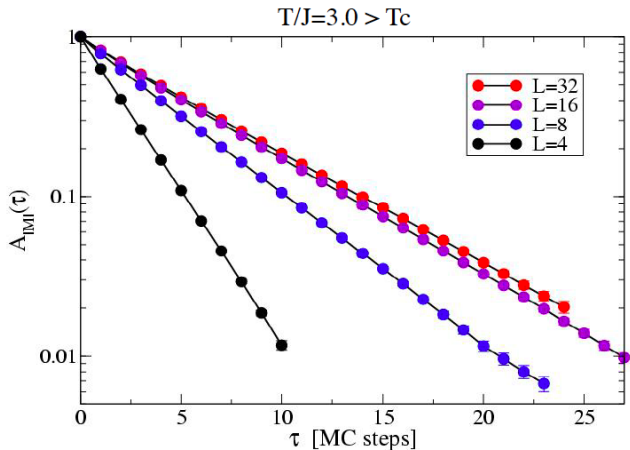
- 对于处于临界点 T_c 的有限尺寸系统, Q 为序参量

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

z 动力学临界指数, 由实现平衡的动力学, 亦即算法, 决定!

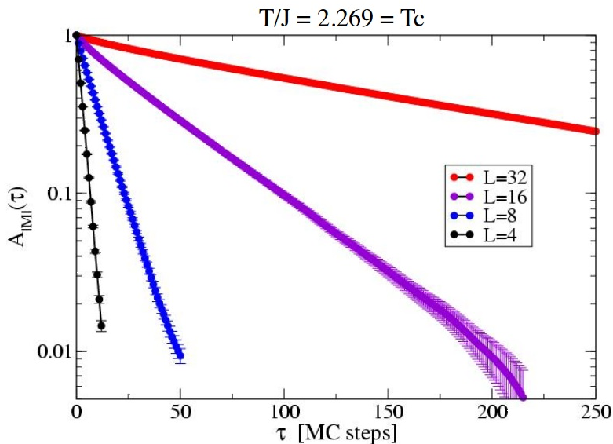
Metropolis算法的弱点, 考虑 $A_{|M|}(\Delta t)$

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

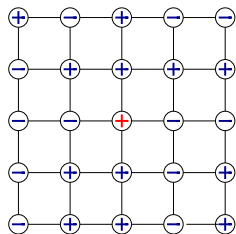
- 需要经过正比于 L^{d+z} 次操作后才可以得到一个独立的位形

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

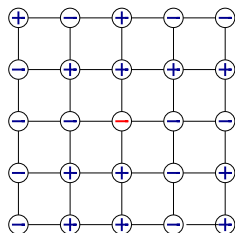


Cluster algorithm: Wolff algorithm

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: $s'_i = -s_i$

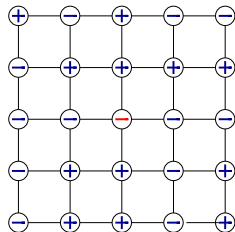


Cluster algorithm: Wolff algorithm

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 $1 - e^{-2K}$, $K = J/k_B T$ 完成
 - i 翻转 $s_k \rightarrow s'_k = -s_k$.
 - ii 将 k 记录到地址列表('堆栈(stack)')

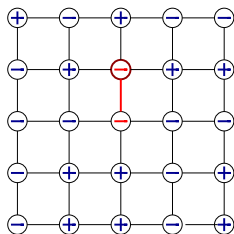


Cluster algorithm: Wolff algorithm

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 $1 - e^{-2K}$, $K = J/k_B T$ 完成
 - i 翻转 $s_k \rightarrow s'_k = -s_k$.
 - ii 将 k 记录到地址列表('堆栈(stack)')

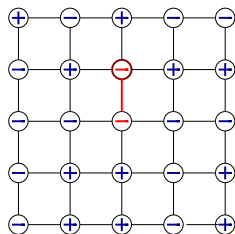


Cluster algorithm: Wolff algorithm

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 $1 - e^{-2K}$, $K = J/k_B T$ 完成
 - i 翻转 $s_k \rightarrow s'_k = -s_k$.
 - ii 将 k 记录到地址列表('堆栈(stack)')
 - ▶ 如果 $s_k = s'_i$ 什么也不作

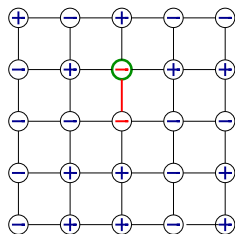


Cluster algorithm: Wolff algorithm

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 $1 - e^{-2K}$, $K = J/k_B T$ 完成
 - i 翻转 $s_k \rightarrow s'_k = -s_k$.
 - 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

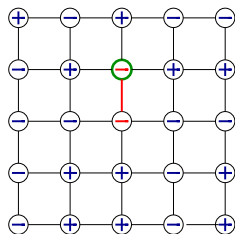


Cluster algorithm: Wolff algorithm

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 $1 - e^{-2K}$, $K = J/k_B T$ 完成
 - i 翻转 $s_k \rightarrow s'_k = -s_k$.
 - 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

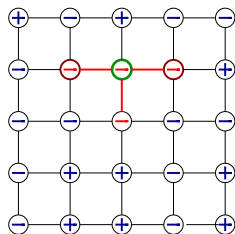


Cluster algorithm: Wolff algorithm

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 $1 - e^{-2K}$, $K = J/k_B T$ 完成
 - i 翻转 $s_k \rightarrow s'_k = -s_k$.
 - 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

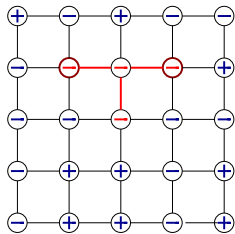


Cluster algorithm: Wolff algorithm

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 $1 - e^{-2K}$, $K = J/k_B T$ 完成
 - i 翻转 $s_k \rightarrow s'_k = -s_k$.
 - 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

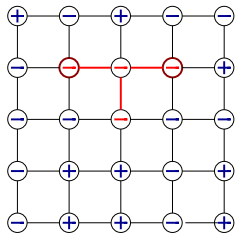


Cluster algorithm: Wolff algorithm

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 $1 - e^{-2K}$, $K = J/k_B T$ 完成
 - i 翻转 $s_k \rightarrow s'_k = -s_k$.
 - 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

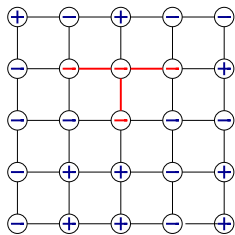


Cluster algorithm: Wolff algorithm

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 $1 - e^{-2K}$, $K = J/k_B T$ 完成
 - i 翻转 $s_k \rightarrow s'_k = -s_k$.
 - 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



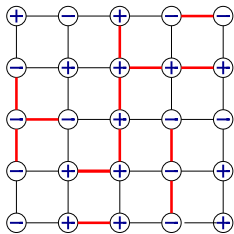
程序实现

```
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
```

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 $1 - e^{-2K}$ 加棒连接, 一个自旋位形等价于一个 ‘棒位形 + 集团符号’

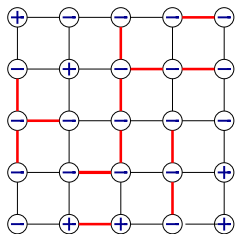


设棒位形有 M 个集团: $N_+ - N_- = \sum_{k=1}^M n_k \text{sig}_k$ 可以证明:

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 $1 - e^{-2K}$ 加棒连接, 一个自旋位形等价于一个 ‘棒位形 + 集团符号’

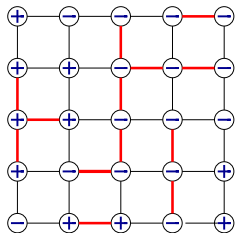


设棒位形有 M 个集团: $N_+ - N_- = \sum_{k=1}^M n_k \text{sig}_k$ 可以证明:

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 $1 - e^{-2K}$ 加棒连接, 一个自旋位形等价于一个 ‘棒位形 + 集团符号’

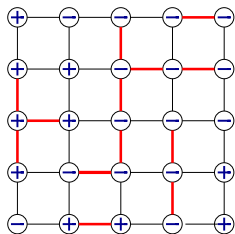


设棒位形有 M 个集团: $N_+ - N_- = \sum_{k=1}^M n_k \text{sig}_k$ 可以证明:

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 $1 - e^{-2K}$ 加棒连接, 一个自旋位形等价于一个 ‘棒位形 + 集团符号’



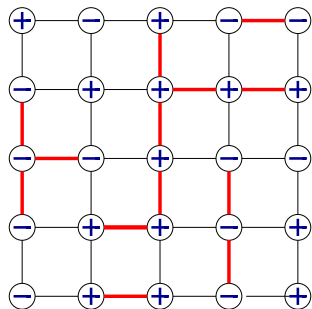
设棒位形有 M 个集团: $N_+ - N_- = \sum_{k=1}^M n_k \text{sig}_k$ 可以证明:

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

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

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

反之, 计算从 Γ' 到 Γ 的几率

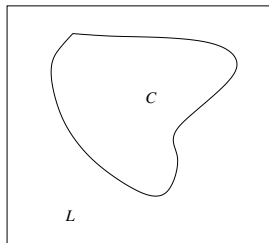
$$T(\Gamma, \Gamma') = e^{-2Kn_a} P(C, \Gamma') \frac{1}{N}.$$

注意到内部连接几率不变

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

因此

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



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,

$$P_{\text{eq}}(\Gamma)/P_{\text{eq}}(\Gamma') = e^{-(E(\Gamma)-E(\Gamma'))/k_B T} = e^{2(n_p-n_a)K}$$

因此

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

魔术的秘诀在哪里？

配分函数

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

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 $\Gamma = \{S, B\} \rightarrow \Gamma' = \{S', B\}$
- 由于上页实际证明了 $W(S, B) = W(S', B)$ ，所以这个 $S \rightarrow S'$ 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！

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]

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.

Wang-Landau algorithm

- 目标: 计算模型的能级简并度(或态密度) $\Omega(E)$
 - ▶ 进而直接计算自由能(任意温度):

$$f = -\frac{k_B T}{N} \ln Z = -\frac{k_B T}{N} \ln \sum_E \Omega(E) e^{-E/k_B T}$$

- ▶ 某些物理量, 比如内能

$$U \equiv \langle H \rangle = \sum_E E \frac{\Omega(E) e^{-E/k_B T}}{Z}$$

一般的MC回顾

位形 Γ 出现的频率 $N(\Gamma)/N$ 正比于正则分布

$$\frac{N(\Gamma)}{N} \sim \exp\left(-\frac{E(\Gamma)}{T}\right) = W(\Gamma)$$

这是通过细致平衡和各态历经来保证的:

- 从 Γ 到 Γ' 的跃迁几率 $T(\Gamma', \Gamma)$

- 满足

$$T(\Gamma', \Gamma)W(\Gamma) = T(\Gamma, \Gamma')W(\Gamma')$$

- Metropolis 算法

$$T(\Gamma', \Gamma) = \min\left[1, \frac{W(\Gamma')}{W(\Gamma)}\right]$$

类似地，我们可以构造一个随机过程，让产生的位形 Γ 出现的频率

$$\frac{N(\Gamma)}{N} \sim \frac{1}{\Omega(E(\Gamma))},$$

即反比于所在能级的态数，通过统计这个频率就可以知道 $\Omega(E)$!

貌似可以通过细致平衡实现（加各态历经）

$$T(\Gamma', \Gamma) \frac{1}{\Omega(\Gamma)} = T(\Gamma, \Gamma') \frac{1}{\Omega(\Gamma')}$$

只要设定 $T(\Gamma', \Gamma)$ 为

$$T(\Gamma', \Gamma) = \min\left[1, \frac{\Omega(E(\Gamma))}{\Omega(E(\Gamma'))}\right]$$

问题是我们并不事先知道 $\Omega(E)$ ，因此上式看来无法给出！

解决办法

- 统计 ‘生产’ 出来的位形出现在每个能级的次数(histgram) $H(E)$
- 如果

$$H(E) = \frac{N(\Gamma)}{N} \times \Omega(E(\Gamma)) = \text{constant}$$

那么 $N(\Gamma)/N \sim \Omega(E(\Gamma))$

- 在实现了这样的频率的前提下，我们统计产生的位形出现在某个能级的次数 $H(E)$ ，应该就是 ‘平的 ‘：即每个能级被访问次数基本一样

让程序自己收敛到 ‘正确’ 的 $\Omega(E)$

算法

- 1 先假定 $\Omega(E) = 1, H(E) = 0$, 让随机序列跑起来。
- 2 从 Γ 试探一个新的位形 Γ' , 根据现有的 $\Omega(E)$ 按

$$T(\Gamma', \Gamma) = \min\left[1, \frac{\Omega(E(\Gamma'))}{\Omega(E(\Gamma))}\right]$$

接受或拒绝。

- 3 根据第2步之后位形的能量 E 进行更新（不论是跃迁成功在新位形，还是跃迁被拒绝回到老位形）

$$\begin{aligned} H(E) &\rightarrow H(E) + 1 \\ \Omega(E) &\rightarrow f \times \Omega(E) \end{aligned}$$

f 是预设的一个常数, 通常取 $f = 2.718... = e$

- 4 重复第2, 3步直到 $H(E)$ 足够‘平’. 清空 $H(E) = 0$. 令 $f \rightarrow \sqrt{f}$, 重新定标 $\Omega(E)$ （见最后说明）, 回到第2步, 继续。
- 5 [2,3,4]循环直到 f 接近1: 根据你设定的要求, 比如 $1 + 1.d - 8$, 退出循环, 完成计算。

说明

- 第4步, ‘足够平’, 通常是检查每个 $H(E)$ 要大于平均数的0.9倍(这个凭经验).

- 怎么估计平均数?

假设我们生产了 N 个位形, 平均数就是 N/N_E , 其中 N_E 就是能级总数。对Ising model, 基本就是自旋数 N_s : 从 $-2N_s$ 到 $2N_s$, 间隔4。
(但是能量不会等于 $-2N_s + 4, 2N_s - 4$.)

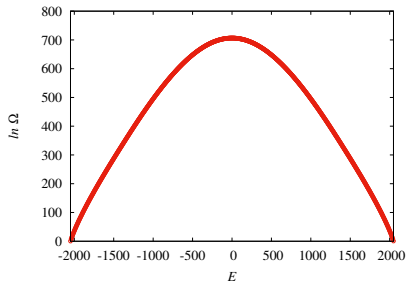
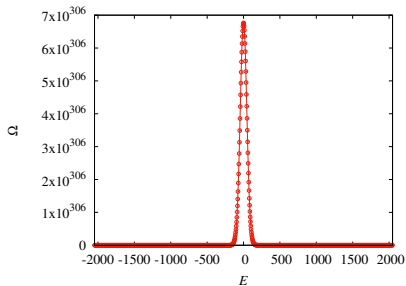
- 由于 $\Omega(E)$ 的量级差别非常大, 通常程序里我们计算 $g(E) = \ln \Omega(E)$.
- 由于MC只能确定频率正比于 $1/\Omega(E)$. 要知道 $\Omega(E)$ 的真正大小, 需要定标: 以Ising model为例, $\Omega(2N_s) = 2$.

我们看到, 整个算法的核心是: 如果频率 $N(\Gamma)/N$ 偏高, 那么随机过程就会通过会通过增大 $\Omega(E)$ 调整它。最终实现细致平衡

$$T(\Gamma', \Gamma) \frac{1}{\Omega(\Gamma)} = T(\Gamma, \Gamma') \frac{1}{\Omega(\Gamma')}$$

判断标准是 $H(E)$ 为常数.

$L = 32$ 二维正方晶格Ising model能级简并度模拟结果



提纲

多体系统与统计物理基础

Ising Model 的Monte Carlo 模拟

1. 简单抽样
2. 重要性抽样
3. 模拟结果与分析, 有限尺寸标度(finite-size scaling)
4. cluster算法
5. Wang-Landau algorithm

Quantum Monte Carlo simulations

1. Path integral representation
2. SSE QMC method
 - 2(a). Heisenberg model
 - 2(b). Bose-Hubbard (XXZ) model
3. Projector Quantum Monte Carlo methods

基本想法

Thermal expectation value of a quantum system

$$\langle A \rangle = \frac{1}{Z} \text{Tr}\{Ae^{-\beta H}\}, \quad Z = \text{Tr}e^{-\beta H}$$

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

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

Choose a basis and insert complete sets of states

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

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 $\Delta_\tau \rightarrow 0$ can be taken.

Example: hard-core bosons

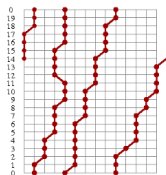
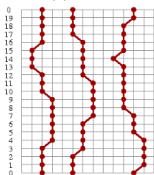
$$H = - \sum_{\langle i,j \rangle} K_{ij} = - \sum_{\langle i,j \rangle} (a_j^\dagger a_i + a_i^\dagger a_j) \quad n_i = a_i^\dagger a_i \in \{0, 1\}$$

Equivalent to $S = 1/2$ XY model

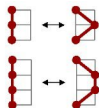
$$H = -2 \sum_{\langle i,j \rangle} (S_i^x S_j^x + S_i^y S_j^y) = - \sum_{\langle i,j \rangle} (S_i^+ S_j^- + S_i^- S_j^+), \quad S^z = \pm \frac{1}{2}$$

"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



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

Expectation values

目标是计算 $\langle A \rangle$, 可写为

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

我们把它写成适合MC importance sampling

$$\langle A \rangle = \frac{\sum_{\{\alpha\}} A(\{\alpha\}) W(\{\alpha\})}{\sum_{\{\alpha\}} W(\{\alpha\})} = \langle A(\{\alpha\}) \rangle_W \quad W(\{\alpha\}) = \mathbf{weight}; A(\{\alpha\}) = \mathbf{estimator}$$

▶ 对角算符(物理量): 粒子数, s_z ,

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

Expectation values

目标是计算 $\langle A \rangle$, 可写为

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

我们把它写成适合MC importance sampling

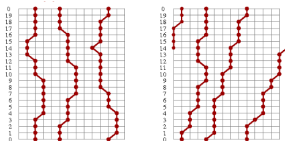
$$\langle A \rangle = \frac{\sum_{\{\alpha\}} A(\{\alpha\}) W(\{\alpha\})}{\sum_{\{\alpha\}} W(\{\alpha\})} = \langle A(\{\alpha\}) \rangle_W$$

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

▶ 非对角算符: Kinetic energy $K = -\sum_{\langle i,j \rangle} K_{ij}$, Quantum Mechanics!

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

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



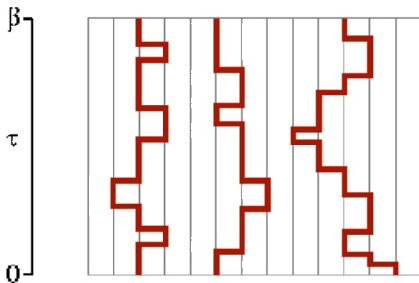
Average over all slices: count number of kinetic jumps

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

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

The continuous time limit

$\Delta_\tau \rightarrow 0$: 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 $e^{-\beta H} = \sum_n^\infty \frac{(-\beta)^n}{n!} H^n$

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

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

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

For any model, $H e^{-\beta H} = \sum_n^\infty \frac{(-\beta)^n}{n!} H^{n+1}$

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

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

$$Z = \sum_{\Gamma} W(\Gamma), \quad \Gamma \text{ 由 } \{\alpha\}_n \text{ 決定, } \langle H \rangle = \frac{\sum_{\Gamma} \frac{n}{\beta} W(\Gamma)}{\sum_{\Gamma} W(\Gamma)} = \frac{\langle n \rangle}{\beta}$$

Series expansion representation

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

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

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

similarly,

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

能量正比于 N , 所以 $\langle n \rangle \approx N\beta$; 热容量正比于 N , $\sigma_n \approx \sqrt{N\beta}$, n 的分布很窄! So, fix length of expansion: cut-off at L and fill in with unit operators $H_0 = 1$

Fixed-length scheme

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

$$H_0 = 1$$

$$n = 5, H_4 H_7 H_1 H_6 H_2$$

→

$$L = 8, H_4 H_0 H_7 H_1 H_0 H_6 H_2 H_0$$

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

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

$H = \sum_i H_i$, 通常 H_i 是 ‘棒算符’ .

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

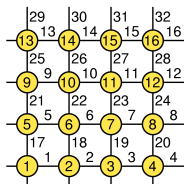
Write H as a bond sum for arbitrary lattice

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

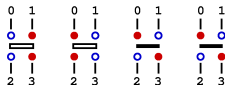
Diagonal(1) and off-diagonal (2) bond operators

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

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



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



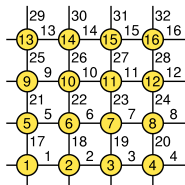
Four Non-zero matrix elements= 1/2

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

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., 重要!

$$\text{Configuration } \Gamma: |\alpha\rangle, S_n: a(p) = 1, 2; b(p) = 1, \dots, 2N$$

位形由 ‘初始态’ $|\alpha\rangle$ 和算符序列 S_n 决定

For fixed-length scheme, introduce $H_{0,0} = 1$,
 $a(p) = 0, 1, 2; b(p) = 0, 1, \dots, 2N$

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

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

$i = 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8$
 $\text{spin}[i] = +1 \ +1 \ -1 \ -1 \ +1 \ -1 \ +1 \ -1$

	p	$a(p)$	$b(p)$	opstring[p]
	1	1	2	4
	2	0	0	0
	3	2	4	9
	4	2	6	13
	5	1	3	6
	6	0	0	0
	7	0	0	0
	8	1	2	4
	9	2	6	13
	10	0	0	0
	11	2	4	9
	12	1	7	14

SSE: an effective discrete representation

- n_2 is even for bipartite lattice!
- $s(p)$ = operator-index string
 - ▶ $s(p) = 0$, if $a(p) = 0$
 - ▶ $s(p) = 2 * b(p) + a(p) - 1$
 - ▶ diagonal $s(p) = \text{even}$
 - ▶ off-diagonal $s(p) = \text{odd}$
- $\sigma(i)$ = spin state, $i = 1, \dots, N$
 - ▶ only one has to be stored

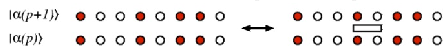
Monte Carlo updating scheme

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

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

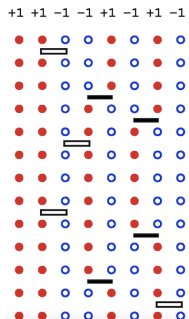
$$P_{accept} = \min\left[\frac{W(\alpha', S'_L)}{W(\alpha, S_L)} \frac{P_{select}(\alpha', S'_L)}{P_{select}(\alpha, S_L)}, 1\right]$$

- 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) 需要 ‘传播’ 自旋状态: flipping spins when off-diag op.



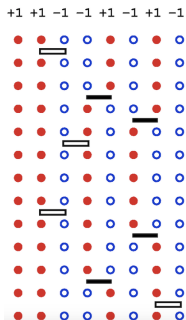
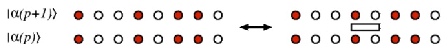
Monte Carlo updating scheme

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

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

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

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



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

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

$n \rightarrow$ current power,

• $n \rightarrow n+1$ ($a=0 \rightarrow 1$)

• $n \rightarrow n-1$ ($a=1 \rightarrow 0$)

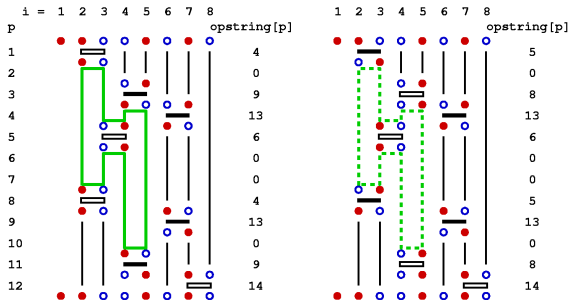
acceptance prob. $P_{accept}([0, 0] \rightarrow [1, b]) = \min\left[\frac{\beta N_b}{2(L-n)}, 1\right]$

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

圈演化, loop update

- 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(Γ) \iff (Γ') 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

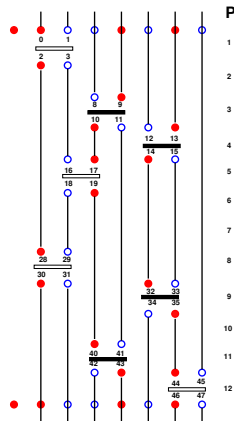
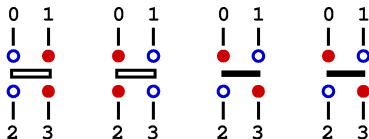
$$W(\Gamma = \{\alpha, S_L\}) = W(\Gamma' = \{\alpha', S'_L\}) = \left(\frac{\beta}{2}\right)^n \frac{(L-n)!}{L!}$$



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	$X(v)$
0	30	30	0
1	31	31	1
2	28	28	2
3	16	16	3
8	42	42	8
9	43	43	9
10	17	17	10
11	41	41	11

$X(v)$: vertex list

▶ operator at p

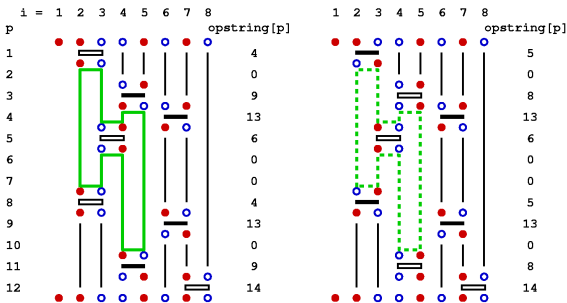
$$v = 4(p - 1) + l, l = 0, 1, 2, 3$$

▶ 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

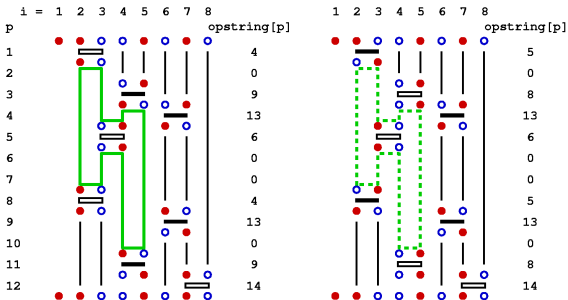
- ▶ a given loop is only constructed once
- ▶ $X(v) = -2$: flipped loop
- ▶ $X(v) = -1$: not flipped loop

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

- 这类似Swendsen-Wang cluster 算法
- 也可以随机构造一个圈，‘反转’它，即类Wolff算法

Operator-loop update

Many spins and operators can be changed simultaneously



```
do  $v_0 = 0$  to  $4L - 1$  step 2
if ( $X(v_0) < 0$ ) cycle
 $v = v_0$ 
if ( $\text{random}[0, 1] < 1/2$ ) then
traverse the loop; for all  $v$  in loop, set  $X(v) = -1$ 
else
traverse the loop; for all  $v$  in loop, set  $X(v) = -2$ 
flip the operators in the loop
endif
enddo
flip:  $s(v/4) = \text{ieor}(s(v/4), 1)$ 

traverse:  $v_2 = \text{ieor}(v, 1)$ 
```

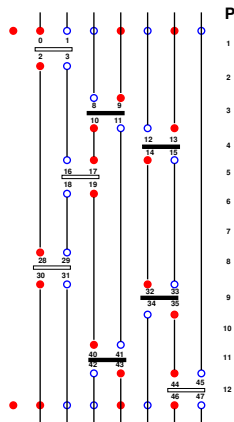
constructing all loops, flip with probability $1/2$; save the changed $s(p)$

- 这类似Swendsen-Wang cluster 算法
- 也可以随机构造一个圈，‘反转’它，即类Wolff算法

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

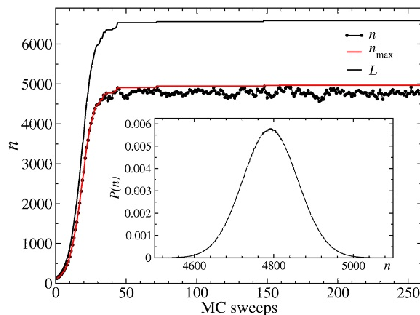
Determination of the cut-off L

Adjust during equilibration

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

● Example: 16×16 system, $\beta = 16$

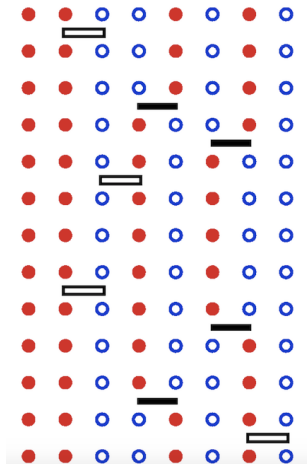
- evolution of L
- n distribution after equilibration



important estimators

- A SSE configuration

+1 +1 -1 -1 +1 -1 +1 -1



- energy estimator : number of operators,
 $H_c = -n/\beta$
- spin stiffness estimator : winding number fluctuations

$$\rho_s = \frac{\langle W_\alpha^2 \rangle}{L^{d-2}\beta}$$

- staggered magnetization
 $m_{sz} = \sum_i (-1)^{i_x+i_y} s_{iz}/N$

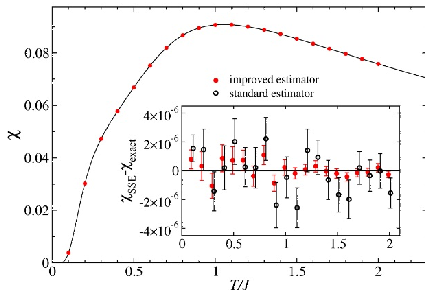
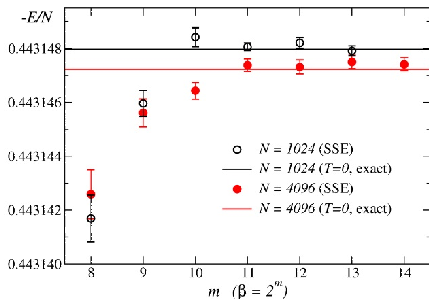
Compare with exact results

Susceptibility of the 4×4 lattice

与严格对角化结果对比

- ▶ SSE results from 10^{10} sweeps

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



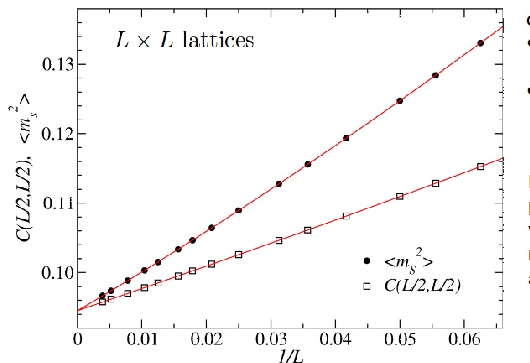
Beta Ansatz; Energy for long chains

- ▶ 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

$N \rightarrow \infty$

▶ $C(L/2, L/2)$ the spin correlation function at the longest distance in a finite system

程序下载

Anders Sandvik 提供了

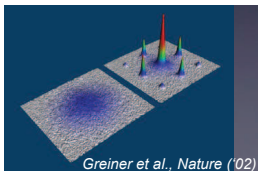
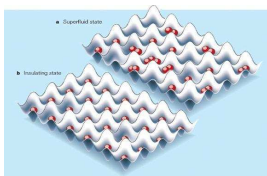
A basic SSE program (Fortran 90) for the 2D Heisenberg model

<http://physics.bu.edu/~sandvik/programs/index.html>

SSE for Bose-Hubbard model

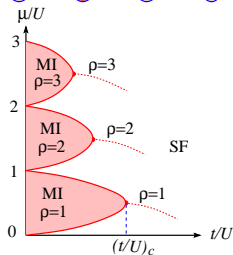
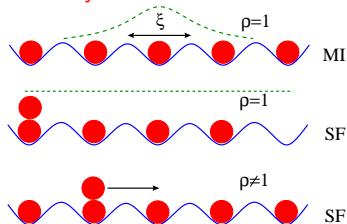
$$H = -t \sum_{\langle ij \rangle} (b_i^\dagger b_j + b_j^\dagger b_i) + \frac{U}{2} \sum_{i=1}^N n_i(n_i - 1) - \mu \sum_{i=1}^N n_i,$$

two ground states SF and MI



- ▶ MI: integer filling, insulating, gaped
- ▶ SF: any filling fraction, gapless

two ways from MI to SF



SSE for Bose-Hubbard model

Bose-Hubbard model

$$H = -t \sum_{\langle ij \rangle} (a_i^+ a_j + a_i a_j^+) + \frac{U}{2} \sum_i n_i (n_i - 1) - \mu \sum_i n_i$$

描述光晶格上玻色冷原子气的量子相变：超流，莫特绝缘

- 占据数表象

$$|\alpha\rangle = |n_1, n_2, \dots, n_N\rangle, \quad \text{需要截断: } n_i = 0, \dots, n_{max}$$

- 哈密顿写成棒算符求和: $H = -\sum_{b=1}^{N_b} \sum_{a=1,2} H_{a,b}$

$$H_{1,b} = \frac{\mu}{z} (n_{i(b)} + n_{j(b)}) - \frac{U}{2z} (n_{i(b)}(n_{i(b)} - 1) + n_{j(b)}(n_{j(b)} - 1)) + C_{add}$$

$$H_{2,b} = t(a_{i(b)}^+ a_{j(b)} + a_{i(b)} a_{j(b)}^+)$$

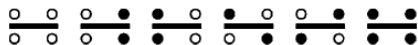
Stochastic Series Expansion

配分函数可作SSE

$$\begin{aligned} Z &= \sum_{\alpha} \sum_{S_M} \frac{\beta^n (M-n)!}{M!} \prod_{p=1}^M \langle \alpha(p) | H_{a_p, b_p} | \alpha(p-1) \rangle \\ &= \sum_{\alpha} \sum_{S_M} W(\alpha, S_M) \end{aligned}$$

但是矩阵元可以不同于Heisenberg model

- 以‘硬核’模型为例($n_i = 0, 1$)



$$\langle 00 | H_{1,b} | 00 \rangle, \langle 11 | H_{1,b} | 11 \rangle, \langle 01 | H_{1,b} | 01 \rangle, \langle 10 | H_{1,b} | 10 \rangle$$

$$\langle 01 | H_{2,b} | 10 \rangle, \langle 10 | H_{2,b} | 01 \rangle$$

对角演化

- 对角演化的操作对象是 S_M 中的单位算符 $[0, 0]$ 和对角算符 $[1, b]$ ，它不改变格点位形 $|\alpha\rangle$ ，但改变棒算符的个数

1. $([0, 0]_p \rightarrow [1, b]_p)$

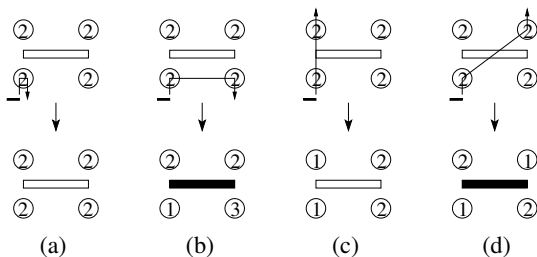


2. $([1, b]_p \rightarrow [0, 0]_p)$



有向圈演化, directed loop updates

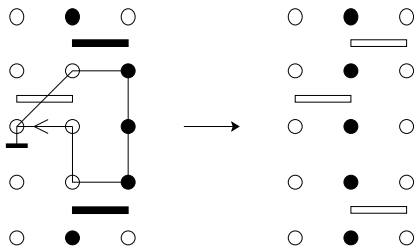
- 实现 $[1, b] \leftrightarrow [2, b]$



圈头携带消灭算符, 通过顶角的四种方式:

(a)是反弹通行, (b)是拐弯通行, (c)是直线通行, (d)是跳跃通行.

有向圈演化, directed loop updates



- 黑色短棒上方的第一个端口是圈头行走的起点
- 出发时圈头携带的操作算符为 a^+
- 右图是演化后的新位形
- ▶ 设 $P(i \rightarrow j)$ 是圈头从 i 进, 从 j 出的几率, 根据细致平衡, 遵从有向圈方程
- ▶ 它的解可以是 heat-bath, 但最好采用避免反弹的 solution B.

Syljuasen and Sandvik, PRE 66, 046701(2002); Syljuasen, PRE 67, 046701 (2003).

Projector Quantum Monte Carlo method

Liang, PRB 1990; Sandvik, PRL 2005

For ground state calculations

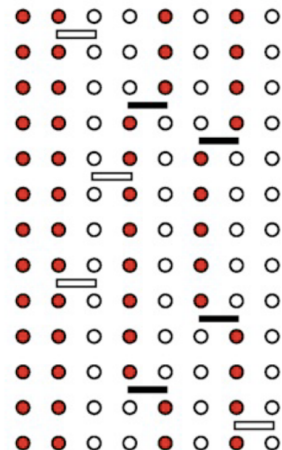
$$\langle A \rangle = \frac{\langle \Psi_0 | U(\tau) A U(\tau) | \Psi_0 \rangle}{\langle \Psi_0 | U(\tau) U(\tau) | \Psi_0 \rangle} \rightarrow \frac{\sum_c A_c W_c}{\sum_c W_c}$$

- A_c is the estimator of A .
- $U(\tau) = H^\tau$ or $U(\tau) = \exp(-\tau H)$

SSE Quantum Monte Carlo method

- A SSE configuration

-1 +1 -1 -1 +1 -1 +1 +1



- S_z basis
- diagonal and loop updates
- observables and estimators
 - energy estimator : number of operators, $H_c = -n/\beta$
 - spin stiffness estimator : winding number fluctuations

$$\rho_s = \frac{\langle W_\alpha^2 \rangle}{L^{d-2}\beta}$$

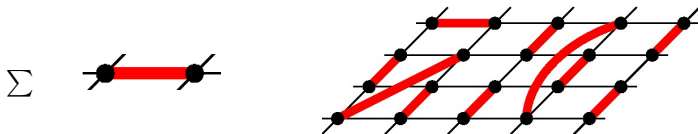
- staggered magnetization

$$m_{sz} = \sum_i (-1)^{i_x+i_y} s_{iz} / N$$

Projector Quantum Monte Carlo method

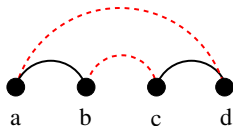
- using VB basis (in the singlet sector $S = 0$)

$$|\Psi\rangle = \sum_v f_v |v\rangle, \quad |v\rangle = |(a_1, b_1) \cdots (a_{N/2}, b_{N/2})\rangle$$



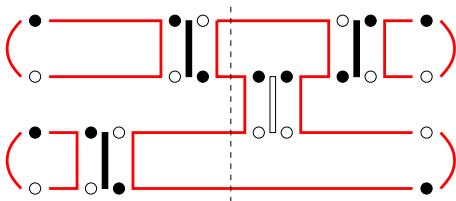
Projector Quantum Monte Carlo method

- take $U(\tau) = \exp(-\tau H)$, SSE representation $\rightarrow Z = \sum_c W_c$
- in the combined basis, loop update algorithm is applied [Sandvik and Evertz, PRB 2010](#)



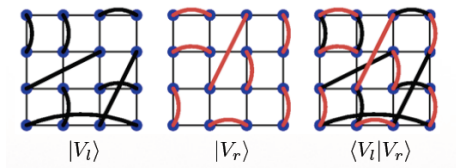
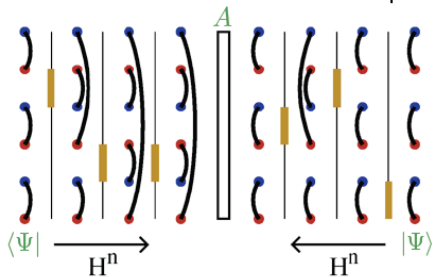
$$P_{ab}|(a, b)(c, d)\rangle = |(a, b)(c, d)\rangle$$

$$P_{bc}|(a, b)(c, d)\rangle = \frac{1}{2}|(a, d)(c, b)\rangle$$



Expectation values

- energy estimator: $H_c = -n/2\tau$ (for $U(\tau) = \exp(-\tau H)$)
- correlation functions computed using **transition graphs**



$$\langle \mathbf{S}_i \cdot \mathbf{S}_j \rangle = \begin{cases} 0, & (i)_L(j)_L \\ \frac{3}{4}\phi_{ij}, & (i,j)_L, \end{cases}$$

$\phi_{ij} = \pm 1$, i, j on the same/different sublattice

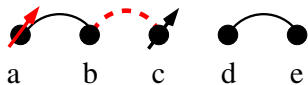
Beach and Sandvik, Nucl. Phys. B 750, 142(2006)

- spin stiffness can also be calculated Liu, Guo, and Sanvik, to be published

Extend valence-bond basis to total spin $S > 0$ states

Tang and Sandvik PRL 2011, Banerjee and Damle JSTAT 2010

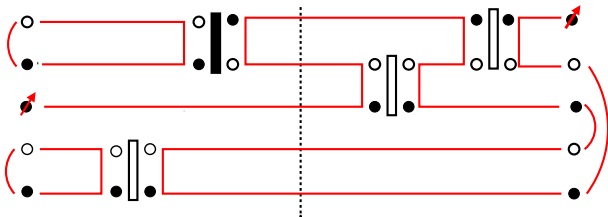
Consider $S_z = S$ sector; $2S$ unpaired "up" spins



$$P_{ab}|(a, b), (\uparrow_c), \dots\rangle = \frac{1}{2}|(a, b), (\uparrow_c), \dots\rangle$$

$$P_{bc}|(a, b), (\uparrow_c), \dots\rangle = \frac{1}{2}|(\uparrow_a), (b, c), \dots\rangle$$

- In the combined basis, loop updates are applied



study spinon bound states and unbinding

transition graphs extended to $S > 0$

- for even N spins:

$S = 0$



$\langle V_\beta | V_\alpha \rangle$, 2 loops

$S = 1$



$\langle V_\beta(j, l) | V_\alpha(i, k) \rangle$,

1

loop, 2 strings

- for odd N spins:

$S = 1/2$



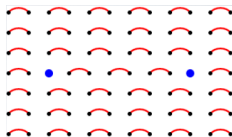
$\langle V_\beta(j) | V_\alpha(i) \rangle$,

1 loop, 1 string

- transition graph has $2S$ open strings:
one spinon \rightarrow one string

The two-spinon distance in the $J-Q_2$ model

- a $S = 1$ state in 2 D



A QMC transition graph representing $\langle \psi_L | \psi_R \rangle$ of $S = 1$ states

- two strings (spinons) in a background of loops formed by valence bonds.
- two strings represent two spinons in bound state



更详细的介绍:

<http://ddl.escience.cn/f/zXHh>

程序下载

<https://physics.bu.edu/~sandvik/huairou16/>

Thank you!