

模拟物理导论

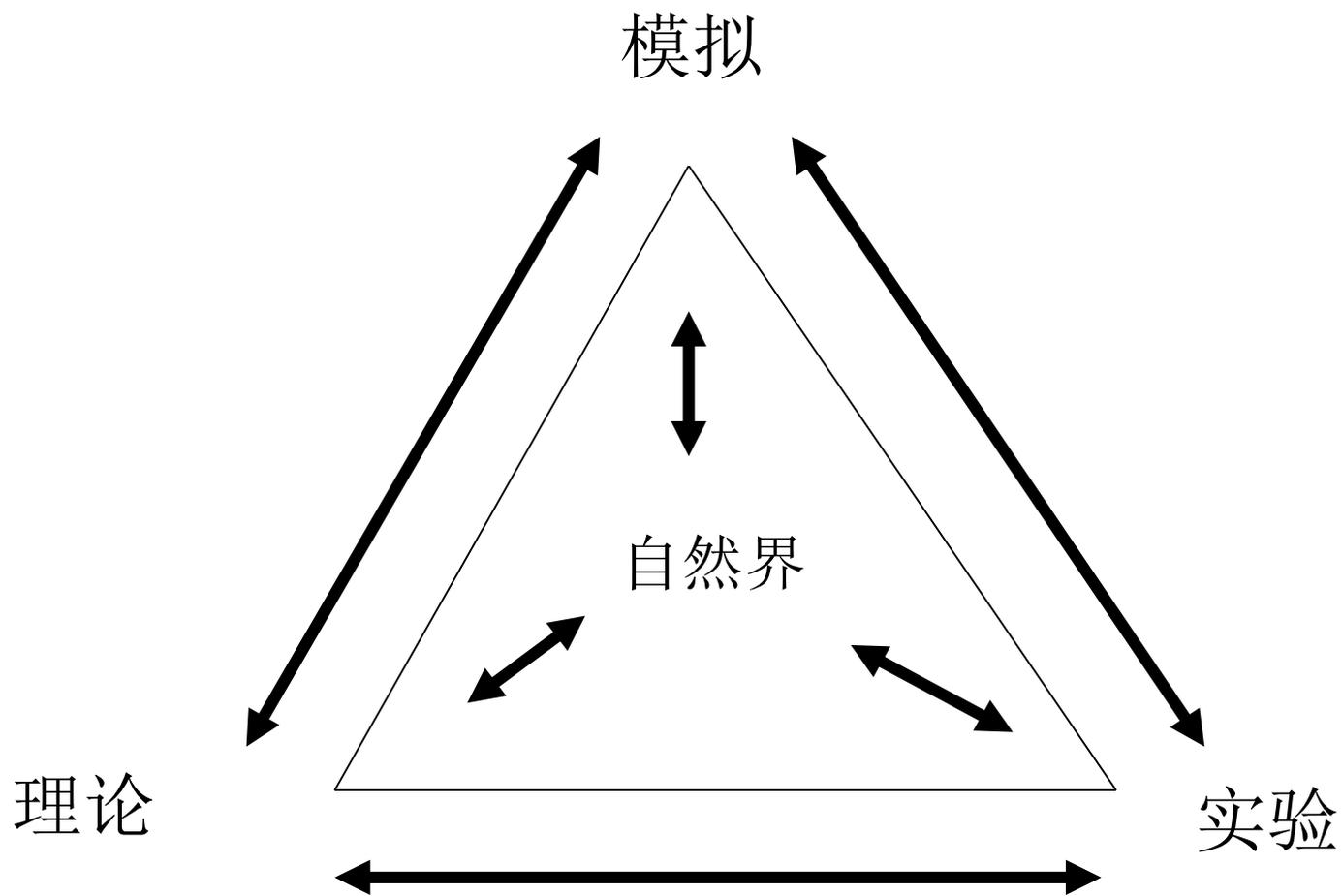
凝聚态物质的数值模拟方法

马红孺

<http://hongruma.net>

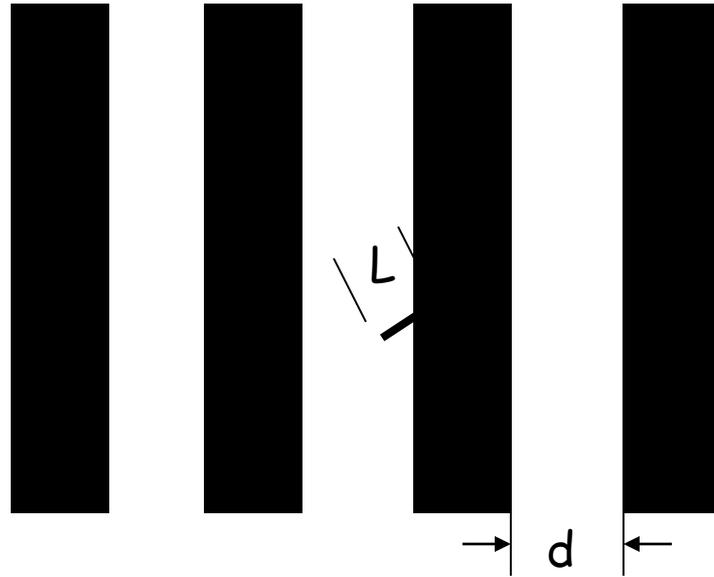
参考书

- Allen and Tildesley, Computer Simulation of Liquids
- Landau and Binder, A Guide to Monte Carlo Simulations in Statistical Physics
- Frenkel and Smit, Understanding Molecular Simulation
- Heermann and Binder, Computer Simulations in Theoretical Physics



Landau 三角形

A little history (the Comte de Buffon needle experiment, AD 1777)



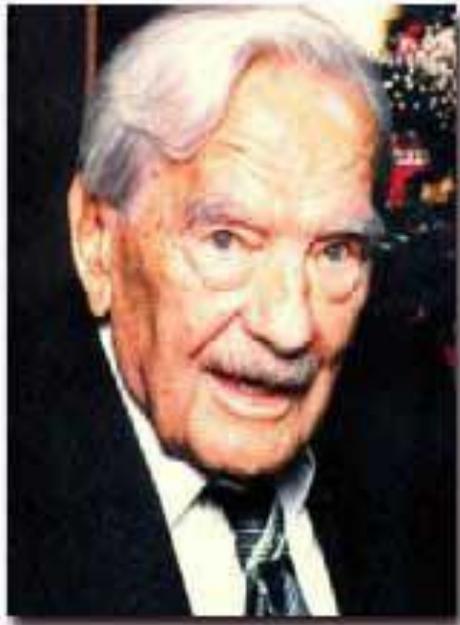
$$p = \frac{2L}{\pi d}$$

Stanislaw Ulam (1909-1984)

S. Ulam is credited as the inventor of Monte Carlo method in 1940s, which solves mathematical problems using statistical sampling.



Nicholas Metropolis (1915-1999)



The algorithm by Metropolis (and A Rosenbluth, M Rosenbluth, A Teller and E Teller, 1953) has been cited as among the top 10 algorithms having the "greatest influence on the development and practice of science and engineering in the 20th century."

The Name of the Game

Metropolis
coined the
name
“Monte
Carlo”, from
its gambling
Casino.



Monte-Carlo, Monaco

Summary of Monte Carlo Method

- Solving mathematical problems (numerical integrations, numerical partial differential equations) by random sampling
- Using random numbers in an essential way
- Simulation of stochastic processes

凝聚态物质的数值模拟方法

主要有三类:

- 1, Monte Carlo 方法,
 - a, 经典统计物理的Monte Carlo 方法,
 - b, 量子Monte Carlo 方法,
- 2, 分子动力学方法,
- 3, 布朗动力学方法.

数值模拟基础:

赝随机数的产生和检验, 微分方程的数值解法.

物理基础:

统计物理学, 凝聚态物理基础知识.

统计物理复习:

理论凝聚态物理的核心问题是:

- 1, 建立理论模型, 写出哈密顿量;
- 2, 利用统计物理计算凝聚态物质的各个物理量.

常用的凝聚态物理模型:

- 1, 格点模型: Ising 模型, Potts模型和Heisenberg 模型及其它们的各种变形(XY, 自旋玻璃,...), Hubbard 模型, 无规行走模型, 逾渗模型, 简谐固体
- 2, 非格点类模型: 经典流体(硬球, LJ, Yukawa,); 量子流体(固体中的电子, 液氦,
- 3, 高分子模型.
- 4, 布朗运动模型.
- 5,

我们不讨论模型的建立, 只讨论如何计算!

经典统计物理:

给定系统的哈密顿量 $H(x_1, x_2, \dots, x_N)$

计算配分函数 $Q = \int dx_1 dx_2 \cdots dx_N \exp[-\beta H(x_1, x_2, \dots, x_N)]$

自由能: $F = -\beta^{-1} \ln Q$

对自由能求导, 得到各个物理量.

内能: $E = -T^2 \frac{\partial \left(\frac{F}{T} \right)}{\partial T}$

熵: $S = - \left(\frac{\partial F}{\partial T} \right)_{V, N}$

问题: 配分函数求不出来!!! (除 1D 和 2D Ising, 理想气体, ..., 很少的几个模型外)

多重积分的计算, Monte Carlo 方法:

由上述第二种表述, 物理量的平均值的计算归结为一个多重积分的计算, 设系统由100个粒子构成, 每个粒子有6个自由度, 所以需要计算600重积分. 现在考虑在每一维取10个点, 总共有 10^{600} 个点. 假设计算机每秒可以计算 10^{12} 个点, 计算这个积分需要 10^{588} 秒!!!!

问题比这个更严重! 如果如上述方式取点, 则积分区域的内点数为 8^{600} , 在总的点中所占比例为 $(8/10)^{600} = 0.714 \times 10^{-58}$, 也就是说, 取的点基本上都在表面上!!!

解决这个问题的方法是Monte Carlo方法.

用Monte Carlo 方法计算积分, 最简单的办法就是在积分区域内随机均匀地取一系列点, 计算被积函数在这些点上的数值并取平均, 然后乘以积分区域的体积. 这种方法称为简单抽样法.

简单抽样法实现起来非常简单, 但有一个问题, 就是把被积函数很小和很大的地方同等对待. 改进这个问题的方法是在重要的区域多取几点. 这种方法称为重要性抽样法.

简单抽样法计算积分 $S = \int_0^1 f(x)dx$

1. 令 $S = 0, N = 0$
2. 产生区间 $[0, 1]$ 上的随机数 x_i ;
3. 计算 $f(x_i)$, $S = S + f(x_i), N = N + 1$;

$$S = S/N$$

简单抽样方法：两个例题

例题一：计算积分 $I = \int_0^1 dx \frac{1}{1+x^4}$ 这个积分的精确值是

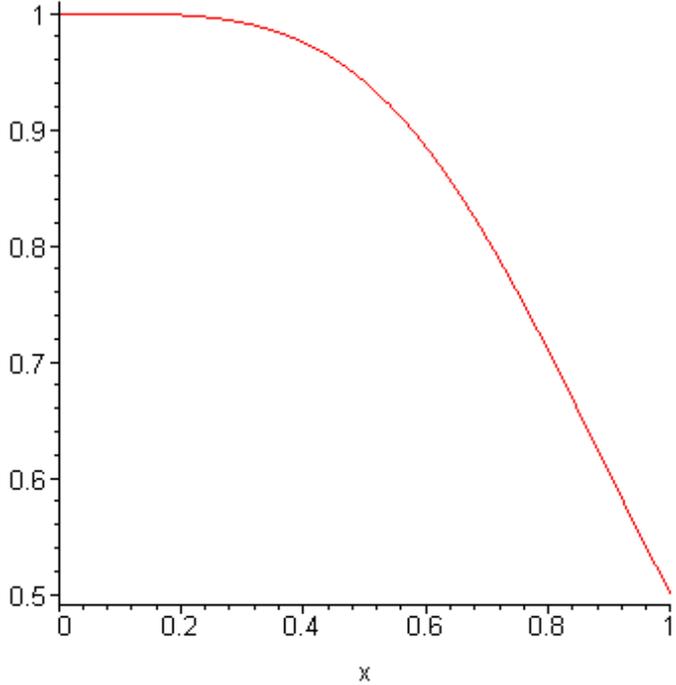
$$I^{\text{exact}} = 0.8669729873399110\dots$$

现在用简单抽样法计算此积分. $I = \langle f \rangle$

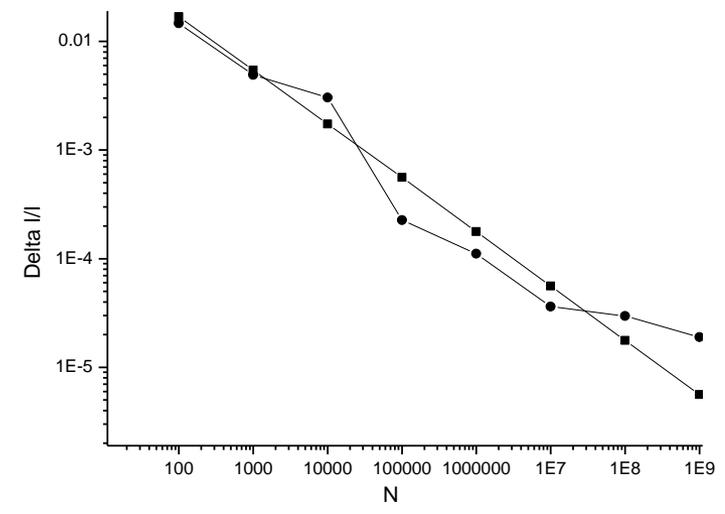
N	I	$\Delta I / I$	$\Delta' I / I$	$1/\sqrt{N}$	$\Delta' I \sqrt{N} / I$
10^2	0.879679	1.47×10^{-2}	1.69×10^{-2}	0.1	1.69×10^{-1}
10^3	0.871238	4.92×10^{-3}	5.45×10^{-3}	3.16×10^{-2}	1.72×10^{-1}
10^4	0.869603	3.03×10^{-3}	1.74×10^{-3}	10^{-2}	1.74×10^{-1}
10^5	0.866777	2.26×10^{-4}	5.61×10^{-4}	3.16×10^{-3}	1.77×10^{-1}
10^6	0.866876	1.11×10^{-4}	1.77×10^{-4}	0.001	1.77×10^{-1}
10^7	0.867004	3.63×10^{-5}	5.60×10^{-5}	3.16×10^{-4}	1.77×10^{-1}
10^8	0.866947	2.97×10^{-5}	1.77×10^{-5}	10^{-4}	1.77×10^{-1}
10^9	0.866956	1.89×10^{-5}	5.60×10^{-6}	3.16×10^{-5}	1.77×10^{-1}

$$\Delta' I / I = \frac{|I - I^{\text{exact}}|}{I^{\text{exact}}}$$

$$\Delta I / I = \frac{\sqrt{\langle f^2 \rangle - I^2}}{I}$$



被积函数



相对误差随样点的变化

$$\frac{\Delta I}{I} \approx \frac{0.177}{\sqrt{N}}$$

例题二:

计算积分

$$I = \int_0^{10} dx x^{68} \exp(-25x)$$

精确结果为

$$I^{\text{exact}} = 0.8641662611747412416$$

用简单抽样法计算得到:

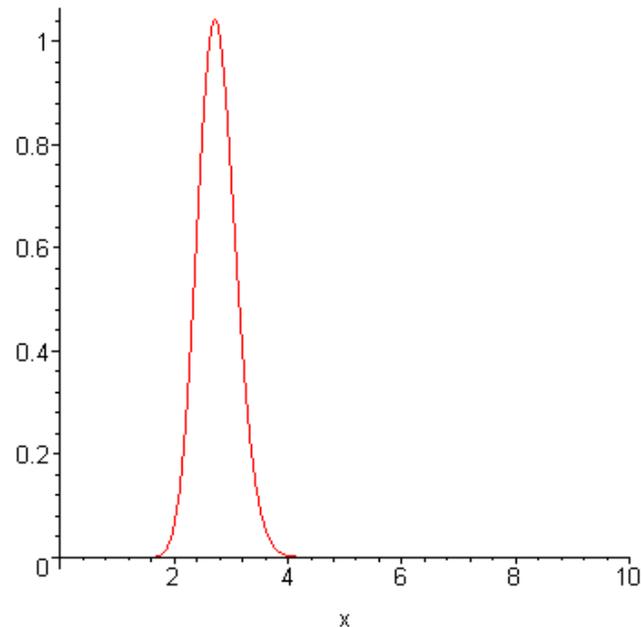
N	I	$\Delta' I / I$	$\Delta I / I$	$1/\sqrt{N}$	$\Delta I \sqrt{N} / I$
10^2	0.52577	0.392	0.287	0.100	2.87
10^3	0.79895	0.755×10^{-1}	0.900×10^{-1}	0.316×10^{-1}	2.85
10^4	0.88543	0.246×10^{-1}	0.269×10^{-1}	0.100×10^{-1}	2.69
10^5	0.87210	0.919×10^{-2}	0.864×10^{-2}	0.316×10^{-2}	2.73
10^6	0.86581	0.190×10^{-2}	0.274×10^{-2}	0.100×10^{-2}	2.74
10^7	0.86485	0.798×10^{-3}	0.868×10^{-3}	0.316×10^{-3}	2.74
10^8	0.86377	0.456×10^{-3}	0.275×10^{-3}	0.100×10^{-3}	2.75
10^9	0.86418	0.218×10^{-4}	0.868×10^{-4}	0.316×10^{-4}	2.75

$$\Delta' I / I = \frac{|I - I^{\text{exact}}|}{I^{\text{exact}}}$$

在这个例子中， $\frac{\Delta I}{I} \approx \frac{2.75}{\sqrt{N}}$

请看右边的函数图像并与例题一的函数图像比较.

由于被积函数主要在区间[2,4]显著,而在其它位置很小,所以在区间[0,10]上均匀取样显然不合适.



从例子可以看出:

$$I = \int_a^b dx f(x) = (b - a) \frac{1}{N} \sum_{i=1}^N f(x_i) = (b - a) \langle f \rangle$$
$$\sigma^2 = \frac{\langle f^2 \rangle - \langle f \rangle^2}{\langle f \rangle^2} \quad \Delta I = (b - a) \frac{\langle f \rangle \sigma}{\sqrt{N}}$$
$$\frac{\Delta I}{I} = \frac{\sigma}{\sqrt{N}}$$

在例题一中，方差比较小，在例题二中，方差比较大！

减小方差：

现在考虑不用均匀取样，而是按照某个分布 $p(x)$ 取样（ $p(x)$ 归一化），积分成为：

$$\begin{aligned} I &= \int_a^b dx f(x) = \int_a^b [f(x)/p(x)] p(x) \\ &= \frac{1}{N} \sum_{i=1}^N [f(x)/p(x)] = \langle f/p \rangle_p \end{aligned}$$

如果取 $p(x) = f(x)/I$ ，则方差为零！！

因 I 待求，我们无法构造出 $p(x)$ 。但我们可以试图去逼近它，因为我们所不知道的是一个归一化常数，分布的形状是知道的。我们可以取一个接近于要求的 $p(x)$ ，例如，可以选

$$p(x) = 0.8920620583 \exp(-2.5(x - 2.8)^2)$$

以减小方差。

如何取样？

试考虑如下算法，构造一个随机过程并产生一系列点 x_i ：

1, 任取 $x_0 \in [a, b]$,

2, 设已经得到 x_i , 选 $y_i = x_i + \delta(\text{ran}() - 0.5)$,
 $\text{ran}()$ 给出 $[0, 1]$ 之间均匀分布的随机数, (δ 为一可调参数)并计算 $p(y)$,

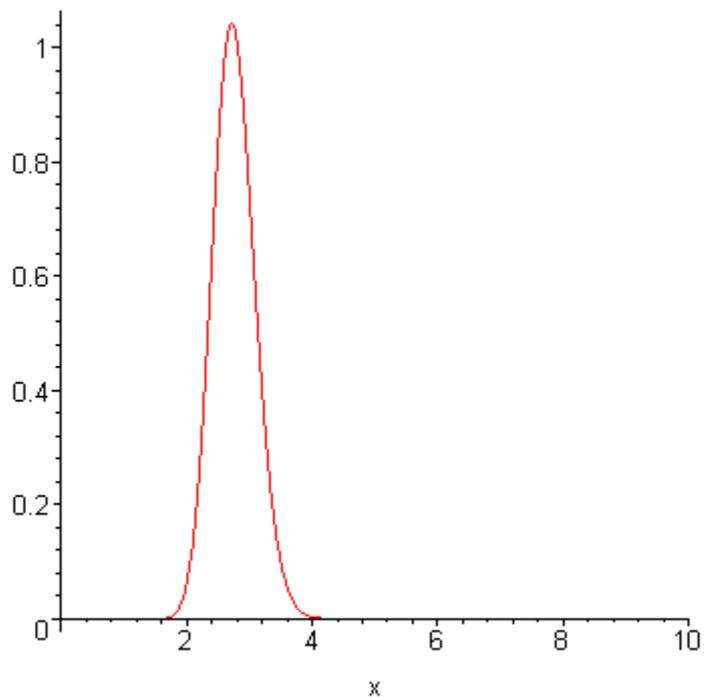
如果 $p(y)/p(x_i) \geq 1$, 则令 $x_{i+1} = y$;

如果 $p(y)/p(x_i) = q < 1$, 在 $[0, 1]$ 区间产生一个随机数 ξ ,

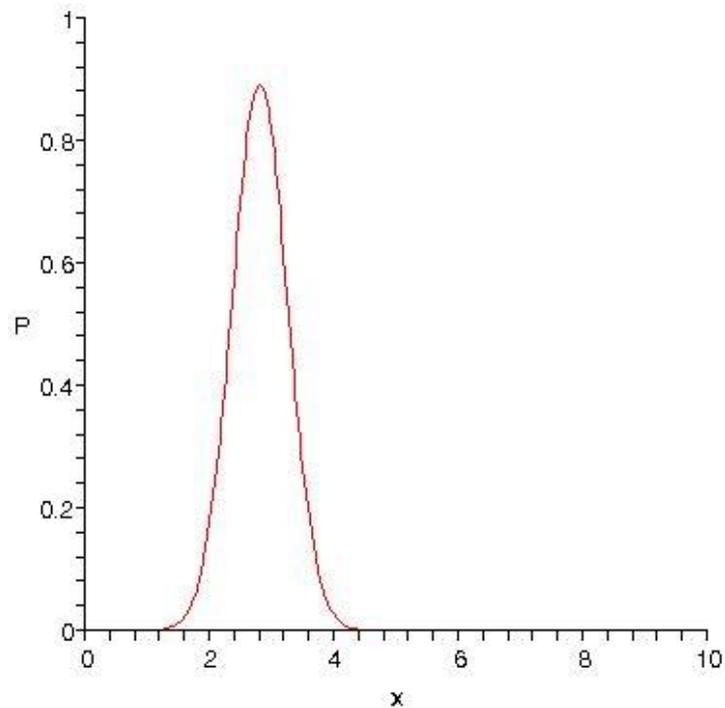
若 $\xi < q$, 则令 $x_{i+1} = y$; 否则, 令 $x_{i+1} = x_i$. 由此可以得到一个序列 $x_i, i = 1, 2, 3, \dots$

利用这样的一系列 $\{x_i\}$, 对于例题二, 计算 f/P 平均值, 得到的结果是:

$$\sigma = 0.37$$



被积函数 $f(x)$



分布 $P(x)$

例题二:

计算积分 $I = \int_0^{10} dx x^{68} \exp(-25x)$ 精确结果为

用重要性抽样法, 按分布 $I^{\text{exact}} = 0.8641662611747412416$

抽样, 计算得到: $P(x) = 0.8920620583 \exp(-2.5(X - 2.8)^2)$

N	I	$\Delta' I / I$	$\Delta I / I$	$1/\sqrt{N}$	$\Delta I \sqrt{N} / I$
10^2	0.8764372	0.142×10^{-1}	0.370×10^{-1}	0.100	0.37
10^3	0.8490170	0.175×10^{-1}	0.120×10^{-1}	0.316×10^{-1}	0.38
10^4	0.8630828	0.125×10^{-2}	0.370×10^{-2}	0.100×10^{-1}	0.37
10^5	0.8603117	0.446×10^{-2}	0.118×10^{-2}	0.316×10^{-2}	0.37
10^6	0.8639793	0.216×10^{-3}	0.367×10^{-3}	0.100×10^{-2}	0.37
10^7	0.8641426	0.273×10^{-4}	0.116×10^{-3}	0.316×10^{-3}	0.37
10^8	0.8642354	0.801×10^{-4}	0.367×10^{-4}	0.100×10^{-3}	0.37
10^9	0.8641638	0.288×10^{-5}	0.116×10^{-4}	0.316×10^{-4}	0.37

随机数发生器:

Monte Carlo模拟离不开随机数, 研究, 寻找好的真随机数发生器是一个永远不会结束的工作.

对真随机数的要求: 尽可能**随机**; **可重复产生**; **移植性好**; **效率高**.

The Roulette and Dice



Mechanical random number
generators

What is a Random Number?

- Follow a definite distribution, usually uniform distribution
- Uncorrelated
- Unpredictable

3

7

0

1

4

Pseudo-Random Numbers

- Truly random numbers can not be generated on a computer.
- Pseudo-random numbers follows a well-defined algorithm, thus predictable and repeatable.
- Have nearly all the properties of true random numbers.

Linear Congruential Generator (LCG)

One of the earliest and fastest algorithms:

$$x_{n+1} = (ax_n + c) \bmod m$$

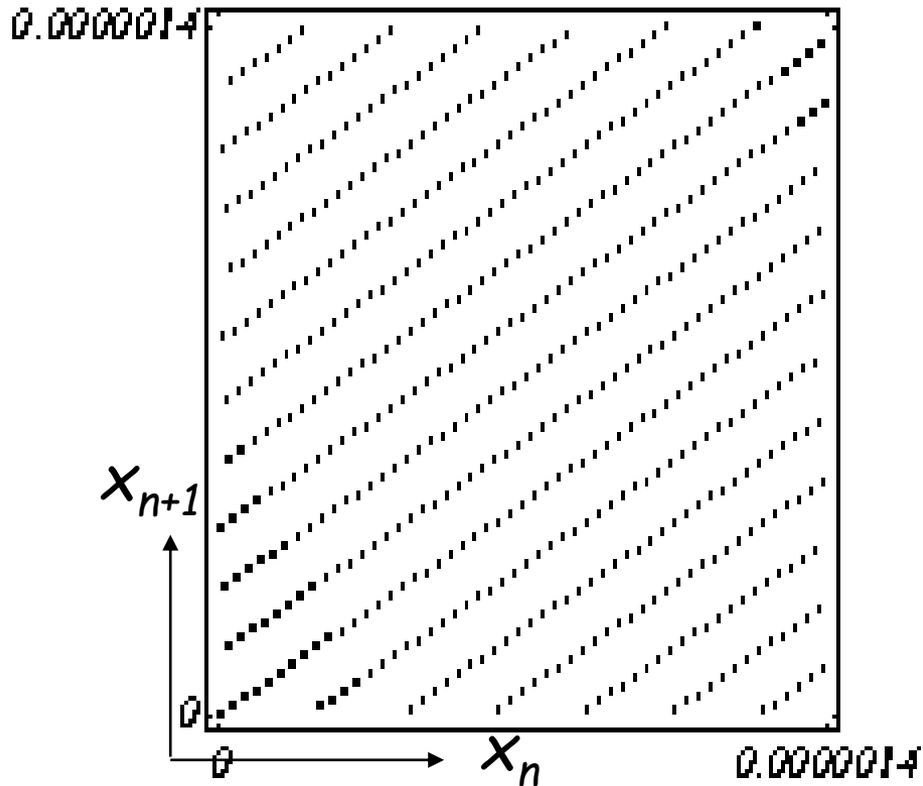
where $0 \leq x_n < m$, m is the modulus, a is a multiplier, c is an increment. All of them are integers. Choice of a , c , m must be done with special care.

Choice of Parameters

Name	m	a (multiplier)	c	period
ANSIC [rand()]	2^{31}	1103515245	12345	2^{31}
Park-Miller NR ran0()	$2^{31}-1$	16807	0	$2^{31}-2$
drand48()	2^{48}	25214903917	11	2^{48}
Hayes 64-bit	2^{64}	6364136223846793005	1	2^{64}

$$(a x + c) \bmod m$$

Short-Coming of LCG



When (x_n, x_{n+1}) pairs are plotted for all n , a lattice structure is shown.

16807法,

`linux=linux*16807`

`if(linux.lt.0) linux=linux+2147483647+1`

产生 $(1, 2^{31})$ 之间的随机数。

Kirkpatrick-Stoll (R250)

假定有250个随机整数. 第251个可以这样得到:

$$N(251) = \text{IEOR}(N(1), N(148))$$

或更一般的

$$N(K) = \text{IEOR}(N(K - 250), N(K - 103))$$

这里 250, 103 是一对特别的数字, 当然还有其它组合. **IEOR** 是 FORTRAN 中 **exclusive OR** 的记号.

实际上只需要存储 250 个整数, 不过为了计算简单, 用 256 个整数.

$$N(\text{IAND}(255, K)) = \text{IEOR}(N(\text{IAND}(255, K-250)), N(\text{IAND}(255, K-103)))$$

附加练习: 写出 R250 随机数发生器的程序.

R250可以写为:

$$x_k = x_{k-250} \oplus x_{k-103}$$

这样一类随机数发生器都可以写为

$$x_k = x_{k-p} \oplus x_{k-q}$$

(p, q) 其它组合有:

(98, 27),	(1279, 216),	(1279, 418),	(2281, 715),
(2281, 915),	(2281, 1029),	(3217, 67),	(3217, 576),
(4423, 271),	(4423, 369),	(4423, 370),	(4423, 649),
(4423, 1393),	(4423, 1419),	(4423, 2098),	(9689, 84),
(9689, 471),	(9689, 1836),	(9689, 2444),	(9689, 4187)

```
#include <stdio.h>
#define AMOL .23283064365386962890625e-9
#define NBIT 32

static int ir[256];
Static int k;

double r250()

{
    int iran_ks;

    k=(k+1)&255;
    ir[k]=ir[(k-250 & 255)] ^ ir[(k-103 & 255)];
    iran_ks=ir[k];
    return iran_ks*AMOL+0.5;

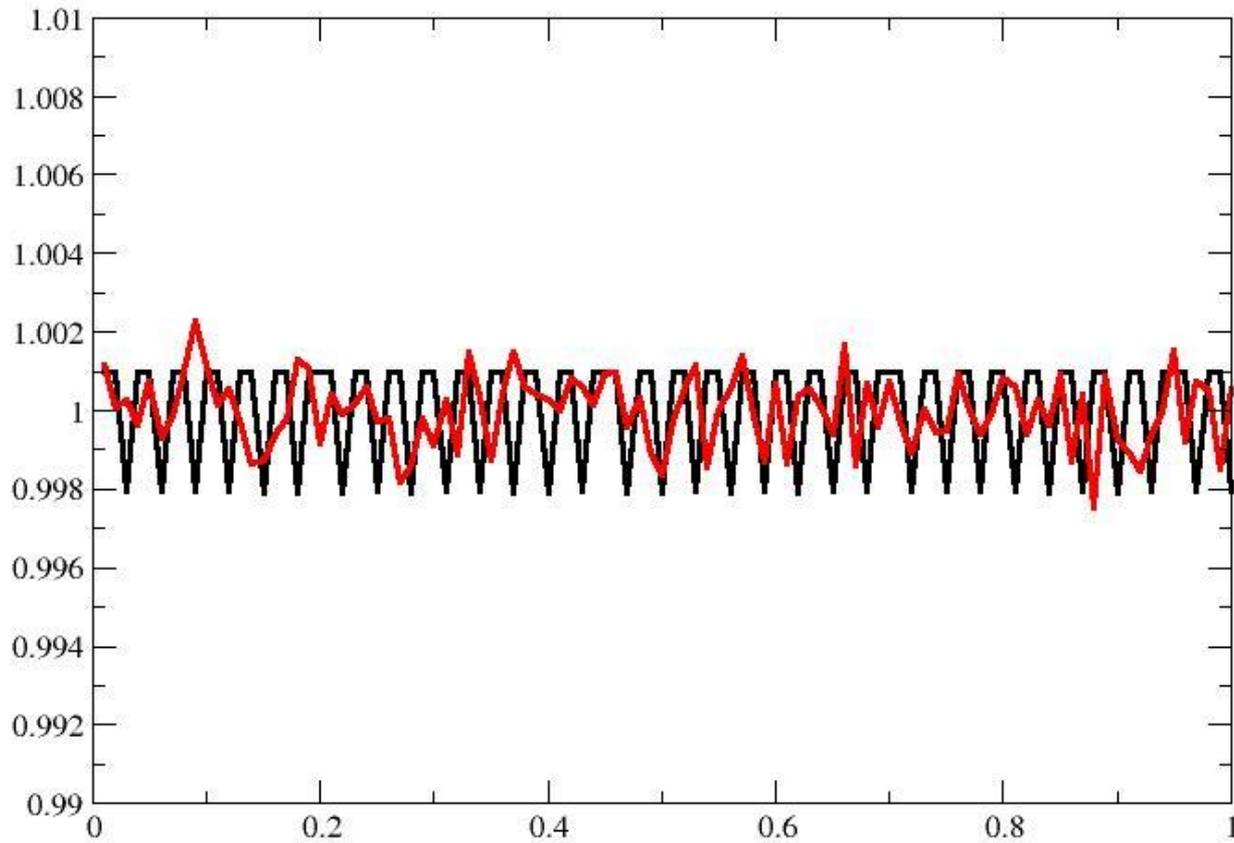
}
```

随机数的随机性检验：

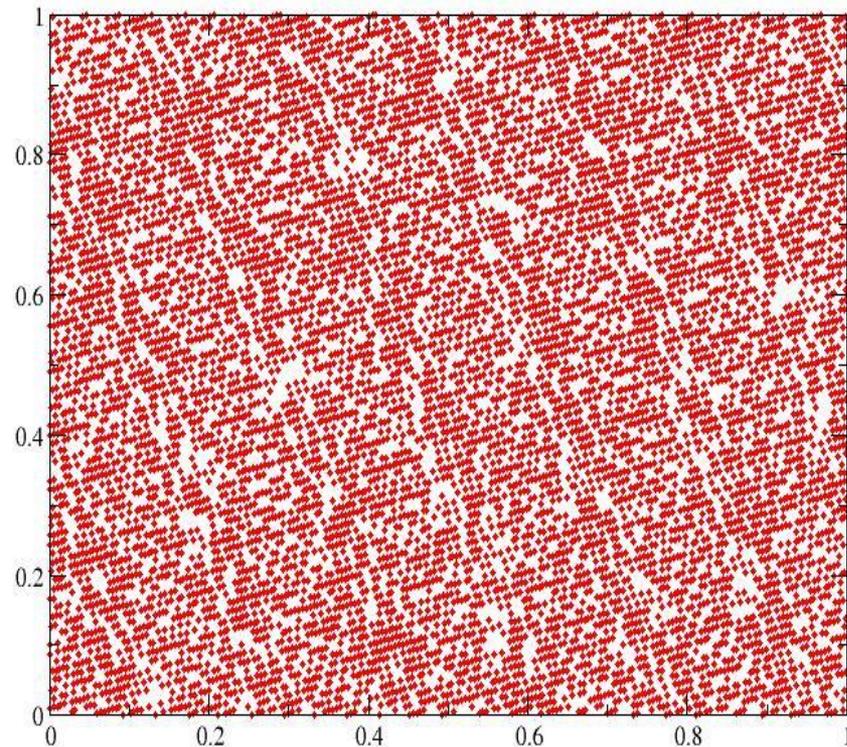
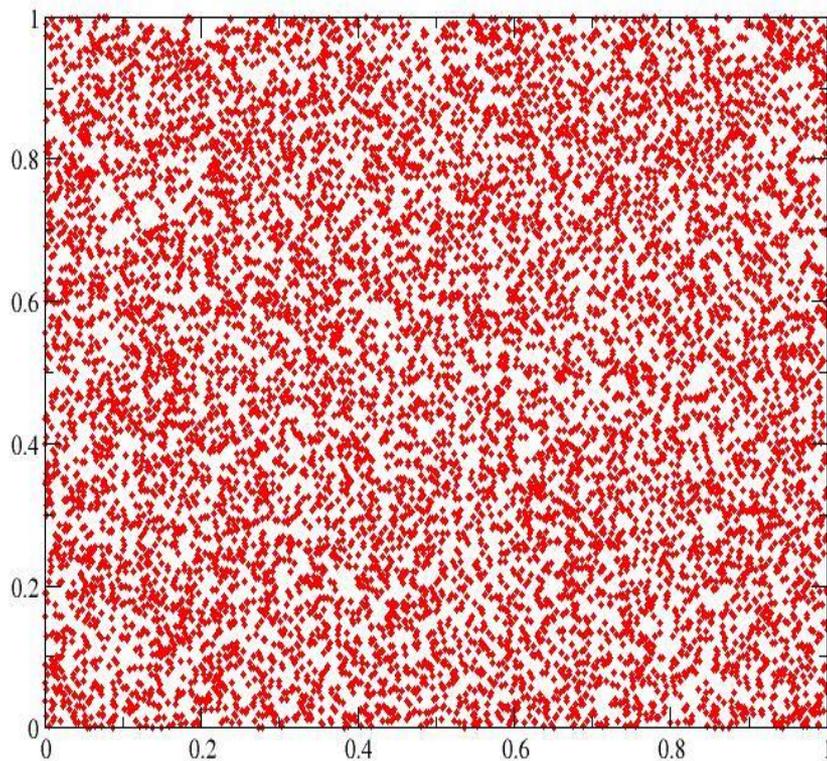
1, 均匀性检验 ([0, 1]区间上的直方图)

2, 相关性检验 (打点观察)

用严格可解的物理模型检验。 如无规行走。



均匀性检验：黑色为同余法的结果，红色为R250的结果。100点， 10^8 个随机数。



相关性检验，左边为R250结果，右边为同余法的结果。10000个点。

无规行走:

为简单起见, 以二维为例. 高分子链的物理模型.

简单无规行走(RW), N 步, 配分函数

$$Q = z^N \quad z \text{ 是最近邻数.}$$

末端距的平均值:

$$\mathbf{R} = \sum_{i=1}^N \mathbf{r}_i \quad \langle \mathbf{R}^2 \rangle = \sum_{i,j=1}^N \langle \mathbf{r}_i \cdot \mathbf{r}_j \rangle = \sum_{i=1}^N \langle \mathbf{r}_i \cdot \mathbf{r}_i \rangle = Na^2$$

a 是单个链段的长度. 这个严格的结果非常有用.

不回头简单无规行走(NRRW), 不容许立即回头, 其余同简单无规行走.
配分函数为

$$Q = (z - 1)^N$$

末端距平方的平均值也与 N 成正比.

计算方法: RW

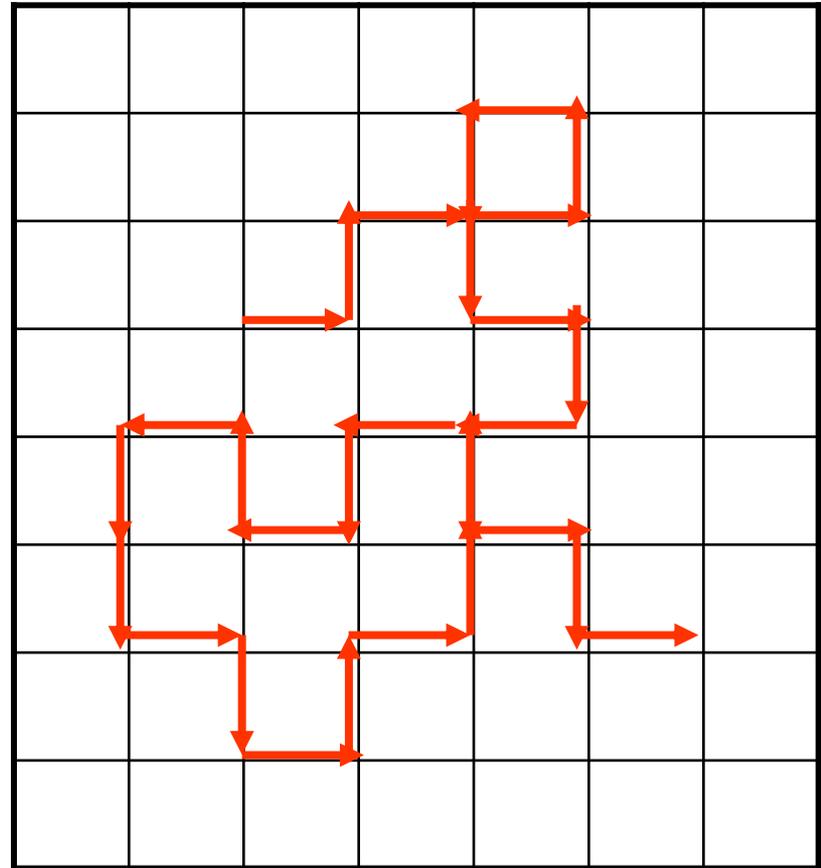
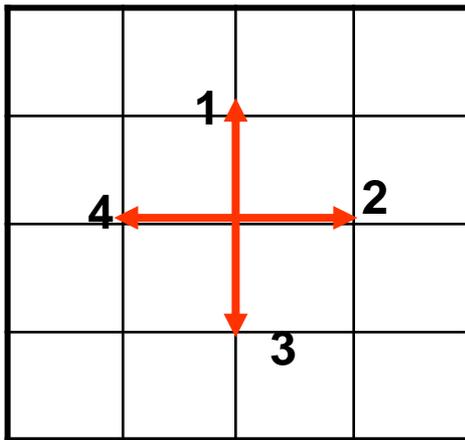
以二维正方格子为例

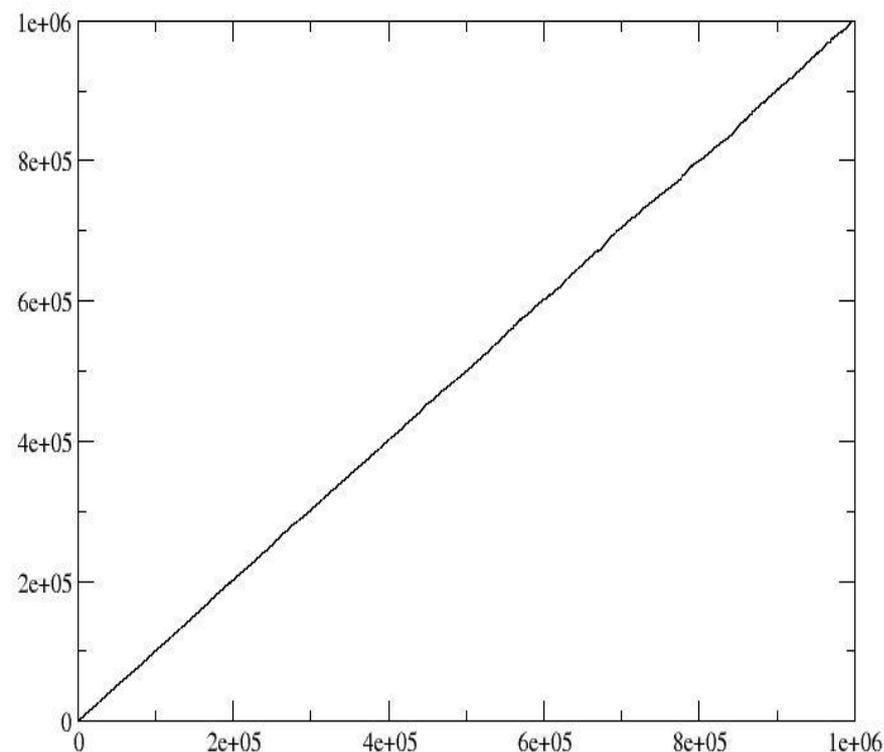
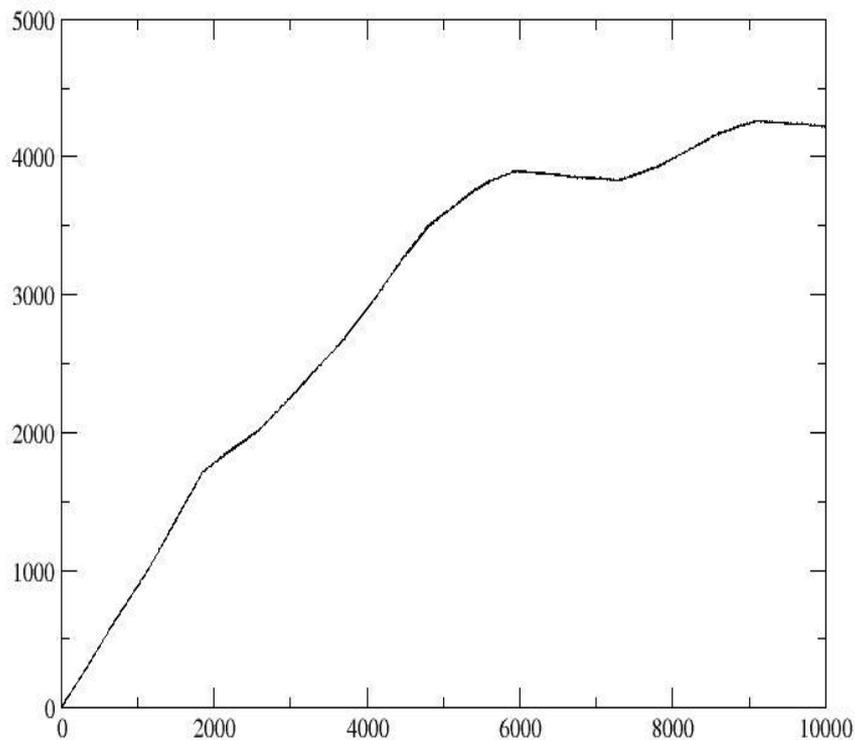
如图, 选定4个方向
产生 $[0,1]$ 区间的随机数 ξ ,

计算 $(int)x = \xi * 4 + 1$

按照 x 确定下一步的行走。

$$v(x) = x$$





无规行走的结果，左边为同余法，右边为R250。
注意两个图的坐标！取10000个样本平均，同余法
用到 10^8 个随机数，R250用到 10^{10} 个随机数。

Monte Carlo 模拟基础

随机变量及其分布

随机变量(以下用 ξ 表示)可分为两类, 一类是离散型随机变量, 它可以取一系列分立值 $x_1, x_2, \dots, x_n, \dots$, 其对应的取某一值的几率为 $p_1, p_2, \dots, p_n, \dots$. p_i 称为的概率分布; 另一类是连续型随机变量, ξ 可连续取值, 设在区间 $[x, x + \delta x]$ 内取值的概率为 $p(x < \xi < x + \delta x)$, 令

$$f(x) = \lim_{\delta \rightarrow 0} \frac{P(x \leq \xi < x + \delta x)}{\delta x}$$

$f(x)$ 称为的分布概率密度, 而处于区间 $[a, b]$ 内的概率由下式给出

$$p(a \leq \xi < b) = \int_a^b f(x) dx$$

几率应归一化，即

$$\sum_{i=1}^{\infty} p_i = 1 \quad \text{对离散型随机变量}$$

$$\int_{-\infty}^{\infty} f(x) dx = 1 \quad \text{对连续型随机变量}$$

定义分布函数

$$F(x) = p(\xi \leq x) = \int_{-\infty}^x f(x) dx$$

代表 ξ 处于 $[-\infty, x]$ 的几率. 显然, $f(x) = \frac{dF}{dx}$

随机变量的数学期望定义为

$$E(\xi) = \sum_i x_i p_i$$

或

$$E(\xi) = \int_{-\infty}^{\infty} x f(x) dx$$

方差定义为

$$D(\xi) = \sum_i (x_i - E(\xi))^2 p_i$$

或

$$D(\xi) = \int_{-\infty}^{\infty} (x - E(\xi))^2 f(x) dx$$

大数定理： 设 $\xi_1, \xi_2, \dots, \xi_n, \dots$ 为一随机变量序列，相互独立，具有同样分布，且 $E(\xi_i) = a$ 存在，则对任意小量 $\varepsilon > 0$ ，有

$$\lim_{n \rightarrow \infty} p \left\{ \left| \frac{1}{n} \sum_{i=1}^n \xi_i - a \right| < \varepsilon \right\} = 1$$

这一定理指出，不论随机变量的分布如何，只要 n 足够大，则算术平均与数学期望值可无限接近，也就是说，算术平均以几率收敛于其数学期望值。

设 $\xi_1, \xi_2, \dots, \xi_n, \dots$ 为一随机变量序列, 相互独立, 具有同样分布, 且 $E(\xi_i) = a, D(\xi_i) = \sigma^2$ 存在, 则当 $n \rightarrow \infty$ 时,

$$P\left(\frac{1}{n} \sum_{i=1}^n \xi_i - a < \frac{X_\alpha \sigma}{\sqrt{n}}\right) \rightarrow \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{X_\alpha} e^{-x^2/2} dx$$

推论

$$P\left(\left|\frac{1}{n} \sum_{i=1}^n \xi_i - a\right| < \frac{X_\alpha \sigma}{\sqrt{n}}\right) \rightarrow \frac{2}{\sqrt{2\pi}} \int_0^{X_\alpha} e^{-x^2/2} dx \quad (A)$$

令:

$$\frac{2}{\sqrt{2\pi}} \int_0^{X_\alpha} e^{-x^2/2} dx = 1 - \alpha$$

$$\left| \frac{1}{n} \sum_{i=1}^n \xi_i - a \right| < \frac{X_\alpha \sigma}{\sqrt{n}}$$

成立的几率为 $1 - \alpha$, $1 - \alpha$ 称为可信水平.

α , $1 - \alpha$ 和 X_α 的数值关系

α	0.5	0.05	0.02	0.01
$1 - \alpha$	0.5	0.95	0.98	0.99
X_α	0.6745	1.9600	2.3263	2.5758

由表可见, 当 $X_\alpha = 2.5758$ 时, (A)成立的几率已经为99%, 也就是说, 该式的可靠性已相当高.

产生高斯分布随机数的一个很好的方法是

$$x = \sum_{i=1}^{12} \xi_i - 6$$

ξ 是 $[0, 1]$ 均匀分布的随机数。

状态空间

- 一个随机变量 X 可以取值的集合为状态空间.
- 例子: 骰子试验中的 $1,2,3,4,5,6$
无规行走中行走者的位置
一组分子的位置和动量
Ising模型的自旋组态.

马尔可夫链Monte Carlo 取样的实现

马尔可夫链Monte Carlo 取样不直接使用 $P(X)$, 而是以某种方式取样, 大量的样本最终符合所需的分布.

马尔可夫链Monte Carlo利用转移矩阵 $W(x \rightarrow y)$, 从当前 x 生成一个状态 y .

马尔可夫链和Monte Carlo

马尔可夫链是一个简单的随机过程，给定转移矩阵 W ，可以得到平衡分布 P 。

马尔可夫链Monte Carlo 是马尔可夫链的计算机实现。在马尔可夫链Monte Carlo中， P 给定，我们需要寻找 W 使得 $P = PW$ 。

马尔可夫链的收敛条件(充分)

各态历经

$$W^n(X \rightarrow X') > 0$$

对所有 $n > n_{\max}$, 所有 X 和 X'

细致平衡

$$P(x_r)W(x_r \rightarrow x_s) = P(x_s)W(x_s \rightarrow x_r)$$

考虑很多个平行的Markov链，在一个给定的某一步，有 N_r 个链处于第 r 个态， N_s 个链处于第 s 个态。于是在下一步从 r 态到 s 态的数目为

$$N_{r \rightarrow s} = N_r W(x_r \rightarrow x_s)$$

从 s 态到 r 态的数目为

$$N_{s \rightarrow r} = N_s W(x_s \rightarrow x_r)$$

从 r 态到 s 态的净转移的数目为

$$\Delta N_{r \rightarrow s} = N_r W(x_s \rightarrow x_r) \left[\frac{W(x_r \rightarrow x_s)}{W(x_s \rightarrow x_r)} - \frac{N_s}{N_r} \right]$$

$$\Delta N_{r \rightarrow s} = N_r W(x_s \rightarrow x_r) \left[\frac{W(x_r \rightarrow x_s)}{w(x_s \rightarrow x_r)} - \frac{N_s}{N_r} \right]$$

若 $W(x_r \rightarrow x_s)$ 满足细致平衡条件，则上式成为

$$\Delta N_{r \rightarrow s} = N_r W(x_s \rightarrow x_r) \left[\frac{P(x_s)}{P(x_r)} - \frac{N_s}{N_r} \right]$$

这是一个十分重要的结果，上式表明，如果二个状态之间不满足分布 P ，则这一Markov过程的演化结果将总是使其趋于满足。这样，就证明了我们的论断。

Metropolis 算法 (1953)

Metropolis 算法取

$$W(x \rightarrow x') = T(x \rightarrow x') \min(1, P(x')/P(x)), \quad x \neq x'$$

T 为一个对称的转移矩阵

$$T(x \rightarrow x') = T(x' \rightarrow x)$$

The Paper (13300 citations from 1988 to 2011)

THE JOURNAL OF CHEMICAL PHYSICS

VOLUME 21, NUMBER 6

JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,

Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER, * *Department of Physics, University of Chicago, Chicago, Illinois*

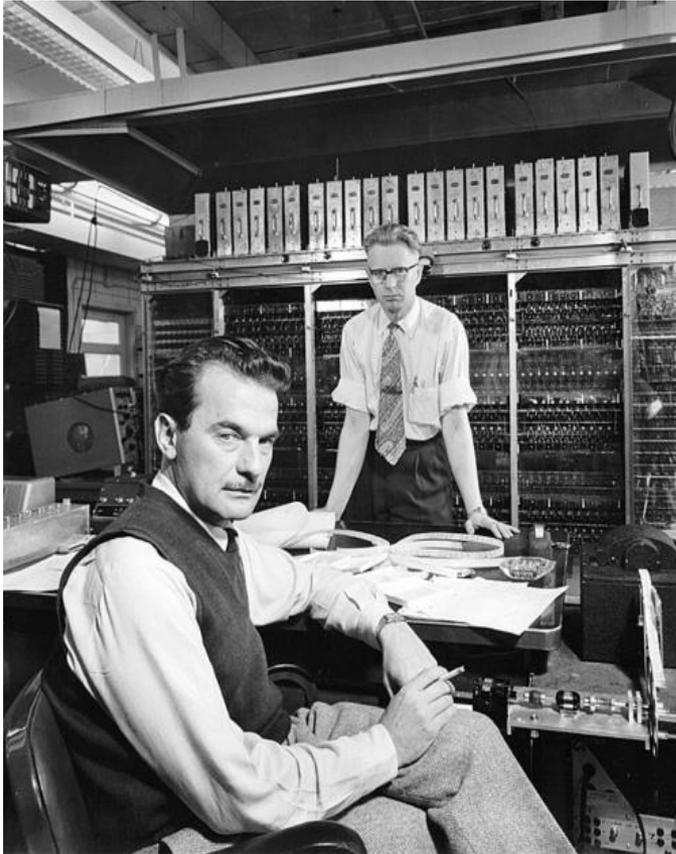
(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

The Calculation

- Number of particles $N = 224$
- Monte Carlo sweep ≈ 60
- Each sweep took 3 minutes on MANIAC
- Each data point took 5 hours

MANIAC the Computer and the Man



Seated is Nick
Metropolis, the
background is
the MANIAC
vacuum tube
computer

正则分布的抽样方法:

选择一个满足细致平衡条件的转移几率;
产生一个Markov链, 丢掉链的前面M个状态;
用其余状态进行物理量的计算.

考虑从r态到s态的转移, 若二状态的能量差为

$$\delta H \equiv H(x_s) - H(x_r)$$

则:

$$\frac{W(x_r \rightarrow x_s)}{W(x_s \rightarrow x_r)} = \exp[-\beta \delta H]$$

当年Metropolis 选择：

$$W(x_r \rightarrow x_s) = \begin{cases} \exp[-\beta\delta H] & \text{如果 } \delta H > 0 \\ 1 & \text{如果 } \delta H \leq 0 \end{cases}$$

目前常用的另一种选择是:

$$W(x_r \rightarrow x_s) = \frac{1}{2} \left[1 - \tanh \left(\frac{\beta \delta H}{2} \right) \right]$$

应当注意的是, w 的选择并不唯一, 只要满足细致平衡条件的要求即可, 但不同的 w 收敛速度往往差别很大, 如何选择合适的 w 以达到尽可能快的收敛速度和尽可能高的计算精度仍然是当前 Monte Carlo 算法研究的前沿课题之一.

分子模型

Molecular systems:

$$H(X) = \sum_{i=1}^N \frac{p_i^2}{2m} + V_N$$

In most cases the interaction part can be approximated by pair interactions:

$$V_N = \frac{1}{2} \sum_{i,j;i \neq j} v(r_i - r_j)$$

One famous example is the Lennard-Jones potential

$$v_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

正则分布:

$$P(E(r, p)) = \frac{e^{-\beta H}}{Q}$$

$$Q = \frac{1}{N!h^{3N}} \int dr dp e^{-\beta H(r, p)}$$

对动量的积分可以算出来

$$Q = \frac{1}{N!\lambda^{3N}} Z_N, \quad \lambda = \frac{h}{\sqrt{2\pi m k T}}$$

分子模型

A very important quantity in statistical mechanics is the pair correlation function $g(\mathbf{r}, \mathbf{r}')$, defined as

$$g(\mathbf{r}, \mathbf{r}') = \frac{V^2}{Z_N} \int_V d^3r_3 d^3r_4 \cdots d^3r_N \exp[-\beta V_N(\mathbf{r}, \mathbf{r}', r_3, \cdots, r_N)],$$

where

$$Z_N = \int_V d^3r_1 d^3r_2 \cdots d^3r_N \exp[-\beta V_N(r_1, r_2, r_3, \cdots, r_N)].$$

It may also be written as

$$g(\mathbf{r}, \mathbf{r}') = \frac{V^2}{N(N-1)} \left\langle \sum_{i,j;i \neq j} \delta(\mathbf{r} - \mathbf{r}_i) \delta(\mathbf{r}' - \mathbf{r}_j) \right\rangle.$$

分子模型

For a homogeneous system the pair correlation function depends only on the distance between r and r' . In this case we denote it as $g(r)$.

The $g(r, r')$ is proportional to the probability that given a particle at point r and find another particle at point r' . At large distances $g(r)$ tends to 1, we may define the total correlation function

$$h(r) = g(r) - 1.$$

The Fourier transform of the above function gives the static structure function (or structure factor)

$$S(k) = 1 + n \int h(r) e^{ik \cdot r} d^3r.$$

分子模型

The structure function is defined as the correlation function of Fourier component of density fluctuations

$$S(k) = \frac{1}{N} \langle \Delta n_{-k} \Delta n_k \rangle$$

The density is defined as $n(\mathbf{r}) = \sum_i^N \delta(\mathbf{r} - \mathbf{r}_i)$
and the density fluctuation is $\Delta n(\mathbf{r}) = \sum_i^N \delta(\mathbf{r} - \mathbf{r}_i) - \frac{N}{V}$
and its Fourier component is: $\Delta n_k = \int d\mathbf{r} \Delta n(\mathbf{r}) e^{-i\mathbf{k} \cdot \mathbf{r}} = \sum_i^N e^{-i\mathbf{k} \cdot \mathbf{r}_i} - N\delta_{\mathbf{k},0}$

$$S(k) = \frac{1}{N} \langle \Delta n_{-k} \Delta n_k \rangle = \frac{1}{N} \sum_{ij; i \neq j} \langle e^{i\mathbf{k} \cdot (\mathbf{r}_i - \mathbf{r}_j)} \rangle + 1 - N\delta_{\mathbf{k},0}$$

分子模型

$$\begin{aligned} S(\mathbf{k}) + N\delta_{\mathbf{k},0} &= \frac{1}{N} \langle \Delta n_{-\mathbf{k}} \Delta n_{\mathbf{k}} \rangle = \frac{1}{N} \sum_{ij; i \neq j} \langle e^{i\mathbf{k} \cdot (\mathbf{r}_i - \mathbf{r}_j)} \rangle + 1 \\ &= \int d\mathbf{r} d\mathbf{r}' e^{i\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}')} \frac{1}{N} \sum_{ij; i \neq j} \langle \delta(\mathbf{r} - \mathbf{r}_i) \delta(\mathbf{r}' - \mathbf{r}_j) \rangle + 1 \\ &= \frac{N-1}{V^2} \int d\mathbf{r} d\mathbf{r}' e^{i\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}')} g(\mathbf{r} - \mathbf{r}') + 1 \\ &= 1 + \left(n - \frac{1}{V} \right) \int d\mathbf{r} g(\mathbf{r}) e^{i\mathbf{k} \cdot \mathbf{r}} \\ &= 1 + \left(n - \frac{1}{V} \right) \int d\mathbf{r} h(\mathbf{r}) e^{i\mathbf{k} \cdot \mathbf{r}} + (N-1)\delta_{\mathbf{k},0} \end{aligned}$$

当体积趋于无限时，**红颜色**的部分可以略去。

分子模型

The structure factor can be measured directly by scattering experiments and can also be calculated by simulations.

Many physical quantities can be expressed in terms of the pair correlation function, for example the energy in NVT ensemble is

$$E = \frac{3}{2}Nk_B T + \frac{N}{2}n \int d^3r v(r) g(r).$$

The pressure is

$$\begin{aligned} \frac{\beta P}{n} &= 1 - \frac{\beta}{3N} \left\langle \sum_{i=1}^N \hat{r}_i \cdot \nabla_i V_N \right\rangle \\ &= 1 - \frac{\beta}{6}n \int d^3r \hat{r} \cdot \nabla v(r) g(r). \end{aligned}$$

分子模型

The compressibility

$$k_B T \left(\frac{\partial n}{\partial p} \right)_T = 1 + n \int d^3 r (g(r) - 1).$$

This expression can be derived from the fluctuations of particle numbers

$$\langle N^2 \rangle - \langle N \rangle^2 = k_B T \left(\frac{\partial \langle N \rangle}{\partial \mu} \right)_{T,V} = -k_B T \frac{\langle N \rangle^2}{V^2} \left(\frac{\partial V}{\partial P} \right)_{T, \langle N \rangle}.$$

Since $n = \frac{\langle N \rangle}{V}$, $dV = -\frac{V^2}{\langle N \rangle} dn$, so

$$\langle N^2 \rangle - \langle N \rangle^2 = \langle N \rangle k_B T \left(\frac{\partial n}{\partial P} \right)_T.$$

分子模型

On the other hand, it can be proved that

$$\langle N^2 \rangle - \langle N \rangle^2 = \langle N \rangle \left(1 + n \int d^3r (g(r) - 1) \right).$$

We have the final result.

The time correlation function is the correlations of two physical quantities at different times,

$$C_{AB}(t, t') = \langle A(t)B(t') \rangle.$$

For systems at equilibrium the time correlation function is a function of the time difference only and can be written as

$$C_{AB}(t) = \langle A(t)B(0) \rangle.$$

分子模型

The velocity auto correlation function of the i th particle is

$$C_{vv}(t) = \langle v_i(t) \cdot v_i(0) \rangle.$$

Which is related to the diffusion constant of the particle.

$$D = \frac{1}{3} \int_0^\infty dt \langle v(t) \cdot v(0) \rangle$$

This can be derived from the definition (we will back to this point)

$$\langle (r_i(t) - r_i(0))^2 \rangle = 6Dt$$

which holds for large t .

分子模型

In general, a transport coefficient is defined in terms of the response of a system to a perturbation.

$$\gamma = \int_0^{\infty} dt \langle \dot{A}(t) \dot{A}(0) \rangle$$

where γ is the transport coefficient, and A is a physical variable appearing in the perturbation Hamiltonian. There is also an Einstein relation associated with this kind of expression

$$\langle (A(t) - A(0))^2 \rangle = 2\gamma t$$

which holds for large t , ($t \gg \tau$, where τ is the relaxation time of \dot{A}).

分子模型

The shear viscosity η is given by

$$\eta = \frac{V}{k_B T} \int_0^\infty dt \langle P_{\alpha\beta}(t) P_{\alpha\beta}(0) \rangle$$

or

$$\frac{V}{k_B T} \langle Q_{\alpha\beta}(t) Q_{\alpha\beta}(0) \rangle = 2\eta t.$$

Here

$$P_{\alpha\beta} = \frac{1}{V} \sum_i \left(\frac{p_{i\alpha} p_{i\beta}}{m_i} + r_{i\alpha} f_{i\beta} \right)$$

$$Q_{\alpha\beta} = \frac{1}{V} \sum_i r_{i\alpha} p_{i\beta}.$$

The negative of $P_{\alpha\beta}$ is often called stress tensor.

位力定理

计算如下平均值

$$\overline{x_i \frac{\partial H}{\partial x_j}}$$

这里 x_i 是 $6N$ 个变量 (q, p) 中的任意一个。在正则分布下

$$\overline{x_i \frac{\partial H}{\partial x_j}} = \frac{1}{Q} \int d\omega x_i \frac{\partial H}{\partial x_j} \exp(-\beta H)$$

其中 $d\omega = \frac{dqdp}{N!h^{3N}}$, 对 x_j 做分部积分,

$$\frac{1}{Q} \int d\omega x_i \frac{\partial H}{\partial x_j} \exp(-\beta H) = -\frac{1}{\beta} \frac{1}{Q} \int d\omega_{(j)} x_i \exp(-\beta H) \Big|_{x_{jL}}^{x_{jU}} + \frac{1}{\beta} \frac{1}{Q} \int d\omega \frac{\partial x_i}{\partial x_j} \exp(-\beta H)$$

其中 $d\omega_{(j)}$ 表示扣除 x_j 以后的积分元。右方的第一项的上下限为 x_j 的上下限, 若 x_j 为位置坐标, 其上下限为容器的两个边, 对应于无限大的势能, 而如果 x_j 是动量坐标, 则其上下限为 $\pm\infty$, 对应于无限大的动能, 从而第一项为0. 第二项中的导数 $\frac{\partial x_i}{\partial x_j} = \delta_{ij}$, 余下的积分恰为配分函数 Q , 于是

$$\overline{x_i \frac{\partial H}{\partial x_j}} = \frac{1}{\beta} \delta_{ij}$$

这样, 我们就得到了一个非常重要的公式

$$\overline{x_i \frac{\partial H}{\partial x_j}} = \delta_{ij} k_B T \tag{1}$$

如果取 $x_i = x_j = q_i$, 由哈密顿运动方程, $\frac{\partial H}{\partial q_i} = -\dot{p}_i$,

$$\overline{q_i \dot{p}_i} = -k_B T$$

如果取 $x_i = x_j = p_i$, 则 $\frac{\partial H}{\partial p_i} = \dot{q}_i$, 于是

$$\overline{p_i \dot{q}_i} = k_B T$$

上式对所有 $3N$ 个 i 求和, 得到

$$\overline{\sum_{i=1}^{3N} p_i \dot{q}_i} = 3N k_B T$$

注意到 $\dot{p}_i = F_i$, 我们得到如下结果

$$\overline{\sum_{i=1}^N \vec{r}_i \cdot \vec{F}_i} = -3N k_B T$$

\vec{F}_i 是第 i 个粒子所受到的力, 对于理想气体, 这个力来自于系统容器的器壁, 可以用压强 P 表示, 在器壁 的 \vec{r} 点的一个小体元, 粒子受到的力为 $-P d\vec{s}$, 于是

$$\overline{\sum_i \vec{r}_i \cdot \vec{F}_i} = - \oint P \vec{r} \cdot d\vec{s} = -P \int \nabla \cdot \vec{r} dV = -3PV$$

即

$$PV = N k_B T$$

这就是理想气体的物态方程。

对于相互作用气体， $\vec{F}_i = \vec{F}_i^{ext} + \sum_j \vec{f}_{ij}$ ，第一项是来自于器壁的外力，第二项是系统中其它粒子对第*i*个粒子的作用。于是

$$\overline{\sum_i \vec{r}_i \cdot \vec{F}_i} = - \oint P \vec{r} \cdot d\vec{s} + \overline{\sum_{ij} \vec{r}_i \cdot \vec{f}_{ij}} = -3PV + \overline{\sum_{ij} \vec{r}_i \cdot \vec{f}_{ij}} = -3Nk_B T$$

$$PV = Nk_B T + \frac{1}{3} \overline{\sum_{ij} \vec{r}_i \cdot \vec{f}_{ij}}$$

右边第二项交换求和指标*i, j*，与原式相加并除以2，注意到 $\vec{f}_{ij} = -\vec{f}_{ji}$ ，上式成为

$$PV = Nk_B T + \frac{1}{6} \overline{\sum_{ij} \vec{r}_{ij} \cdot \vec{f}_{ij}} \quad (1)$$

其中 $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ 。这个公式在Monte Carlo 或分子动力学模拟中常用来计算系统的物态方程。

Monte Carlo 模拟

Monte Carlo simulation of Particle Systems

粒子系统的Monte Carlo 模拟和自旋系统原则上是一样的。

Metropolis 算法为:

1, 随机或顺序选取一个粒子, 其位置矢量为 r_i , 对此粒子做移动

$$x_i \Rightarrow x_i + d_x (\text{ran}() - 0.5)$$

$$y_i \Rightarrow y_i + d_y (\text{ran}() - 0.5)$$

$$z_i \Rightarrow z_i + d_z (\text{ran}() - 0.5)$$

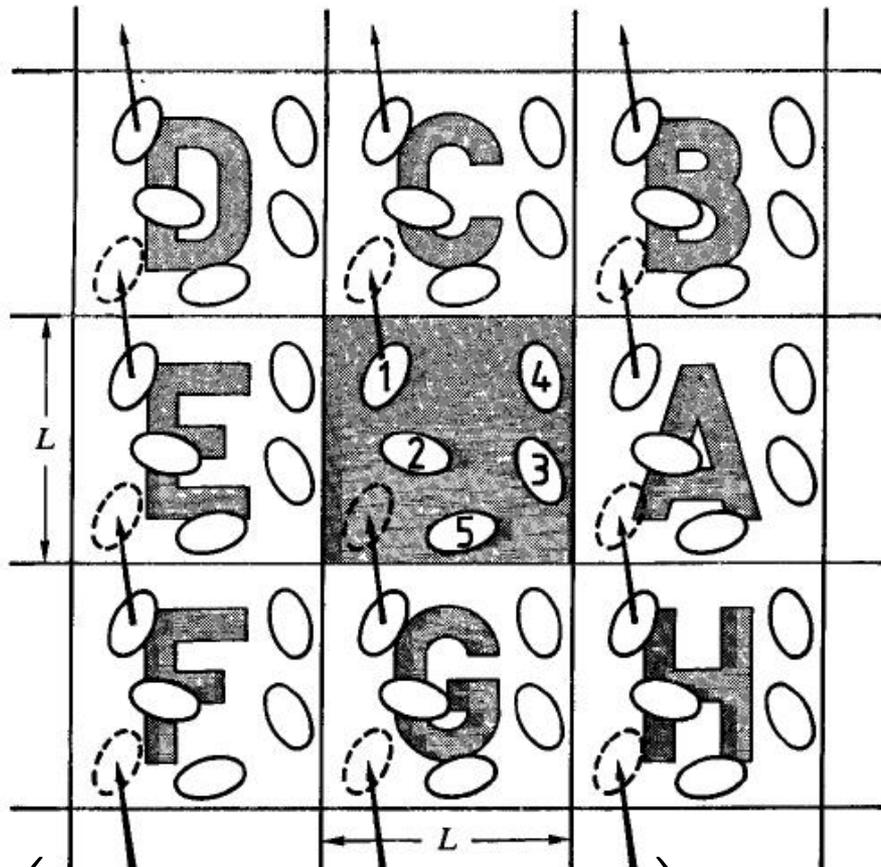
2, 计算前后的能量差, 决定是否接受移动。

3, 在达到平衡后, 收集数据, 计算物理量。

周期性边界条件

设想把模拟的盒子在所有方向周期性排列，实际上计算的是一个无穷大的系统。

相互作用能



$$u_{\text{PBC}}(\mathbf{r}_i - \mathbf{r}_j) = \sum_{\mathbf{n}} u \left(\mathbf{r}_i - \mathbf{r}_j + \sum_{\mu} L_{\mu} n_{\mu} \right)$$

$$U = \frac{1}{2} \sum'_{i,j} u_{\text{PBC}}(\mathbf{r}_i - \mathbf{r}_j)$$

约化单位, ...

对于LJ势,

能量的单位 ϵ

长度的单位: σ

温度的单位: ϵ/k_B

$$u = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

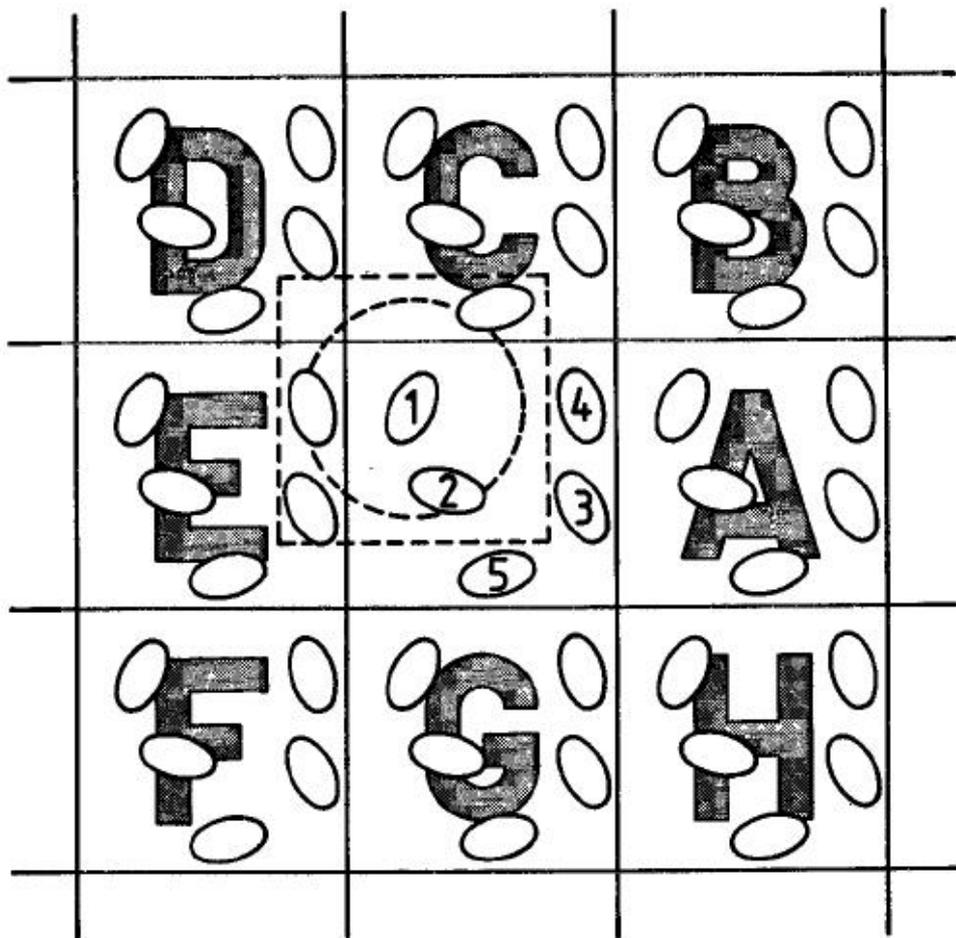
$$u = 4 \left[\left(\frac{1}{r} \right)^{12} - \left(\frac{1}{r} \right)^6 \right]$$

计算能量时的切断

对于短程相互作用，实际计算时，需要对能量的表达式做切断近似，然后考虑尾修正。

切断后，计算对第 i 个粒子的相互作用时，只考虑离开第 i 粒子距离小于切断半径的粒子。

注意包含周期性边界条件后像中的粒子。



切断势定义为：

$$u(r) := \begin{cases} u(r), & r \leq r_c \\ 0, & r > r_c \end{cases}$$

能量的尾修正（平均每个粒子）：

$$u^{\text{tail}} = \frac{1}{2} \int_{r_c}^{\infty} 4\pi r^2 dr \rho u(r)$$

压强的尾修正：

$$P^{\text{tail}} = \frac{1}{6} \rho^2 \int_{r_c}^{\infty} 4\pi r^3 dr \frac{\partial u}{\partial r}$$

对于LJ势， 能量的尾修正（平均每个粒子）：

$$u^{\text{tail}} = \frac{8}{3} \pi \rho \left[\frac{1}{3} \left(\frac{1}{r_c} \right)^9 - \left(\frac{1}{r_c} \right)^3 \right]$$

压强的尾修正：

$$P^{\text{tail}} = \frac{16}{3} \pi \rho \left[\frac{2}{3} \left(\frac{1}{r_c} \right)^9 - \left(\frac{1}{r_c} \right)^3 \right]$$

切断导致的冲击力对于压强的修正：

$$\Delta P^{\text{imp}} = \frac{8\pi}{3} \rho^2 g(r_c) \left[\left(\frac{1}{r_c} \right)^9 - \left(\frac{1}{r_c} \right)^3 \right]$$

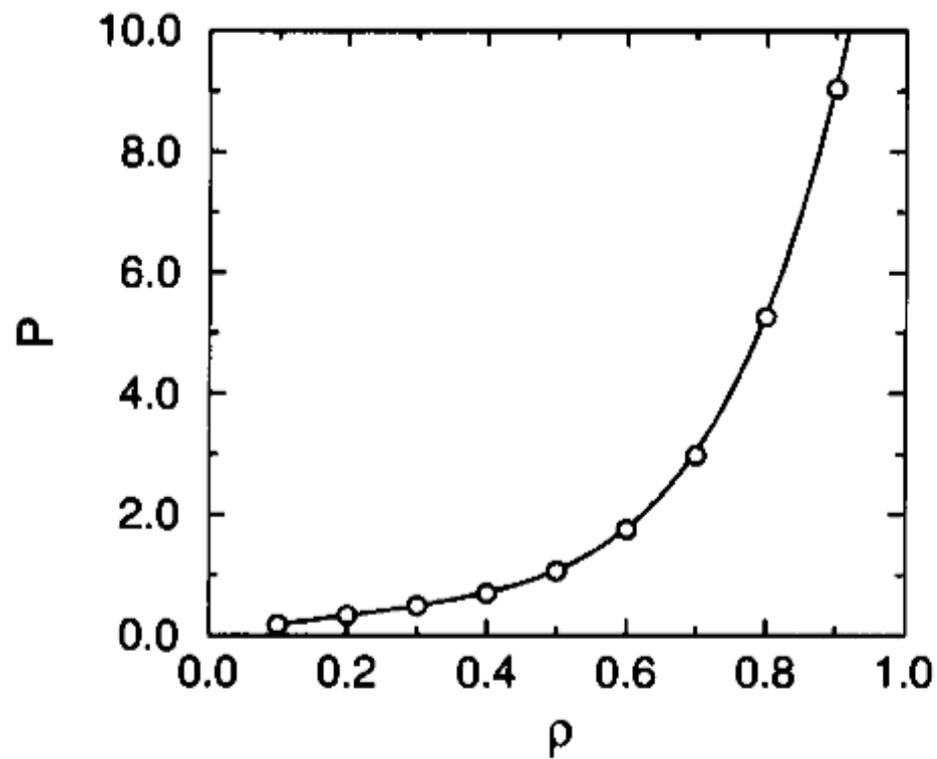
切断势定义为：

$$u(r) := \begin{cases} u(r) - u(r_c), & r \leq r_c \\ 0, & r > r_c \end{cases}$$

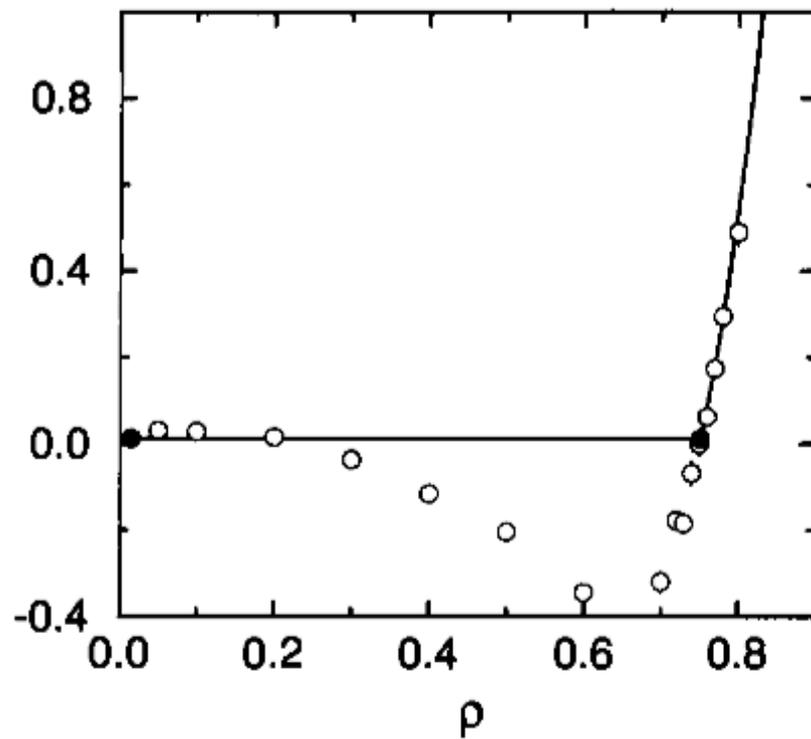
一个算法

- 选取一个粒子，随机移动 Δr
- 计算与此移动相联系的能量变化 δH
- 计算此移动的转移概率 W
- 产生一个在 $[0, 1]$ 之间均匀分布的随机数 ξ
- 如果 $\xi < W$ ，则接受此移动，否则，不移动。
无论何种情况，其结果都作为一新的状态
- 分析该状态，为计算平均值收集数据

LJ 物态方程

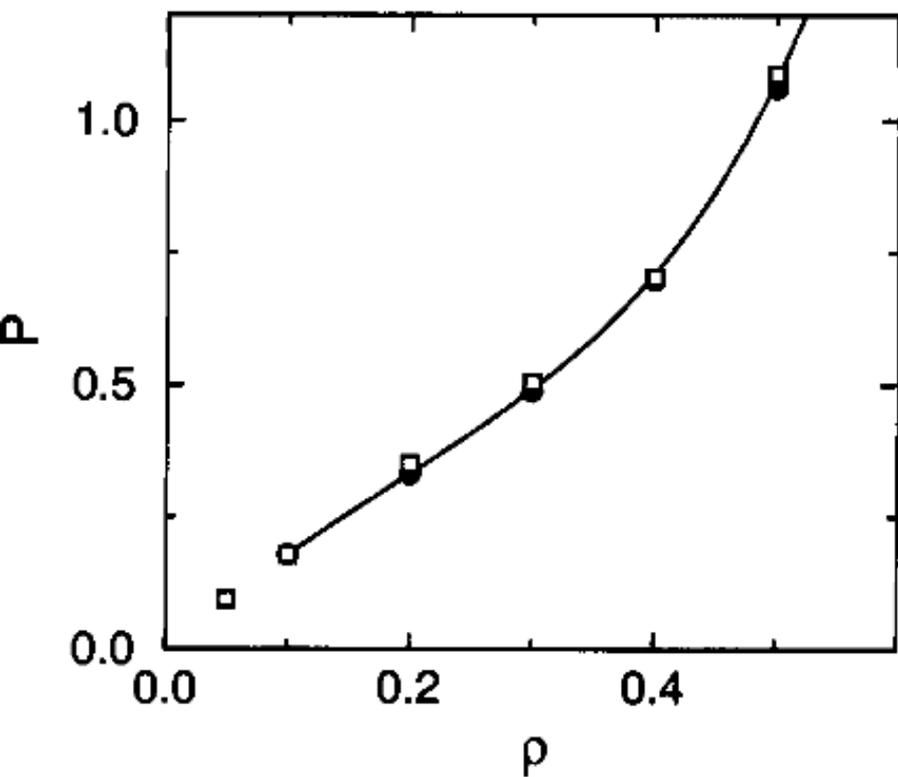


$T=2.0$

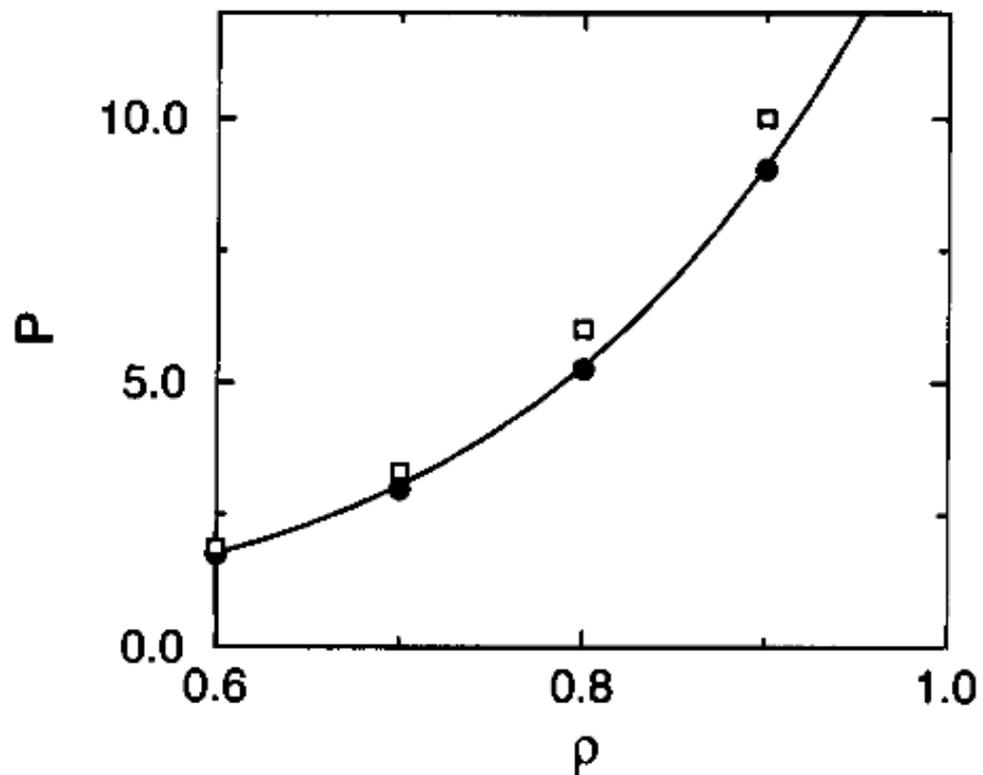


$T=0.9$

LJ 物态方程

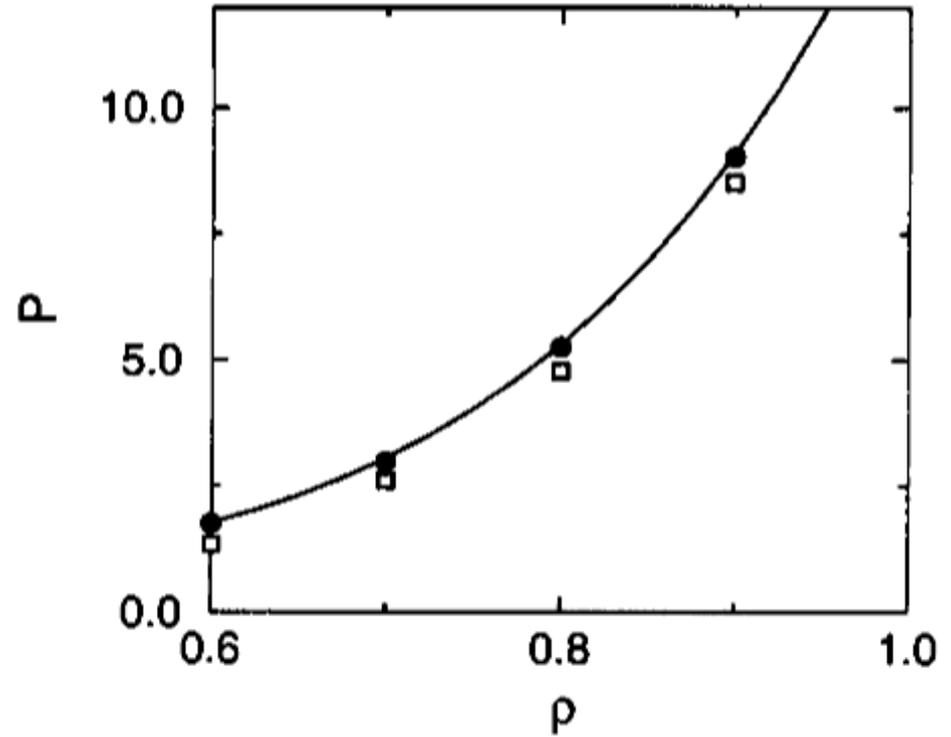
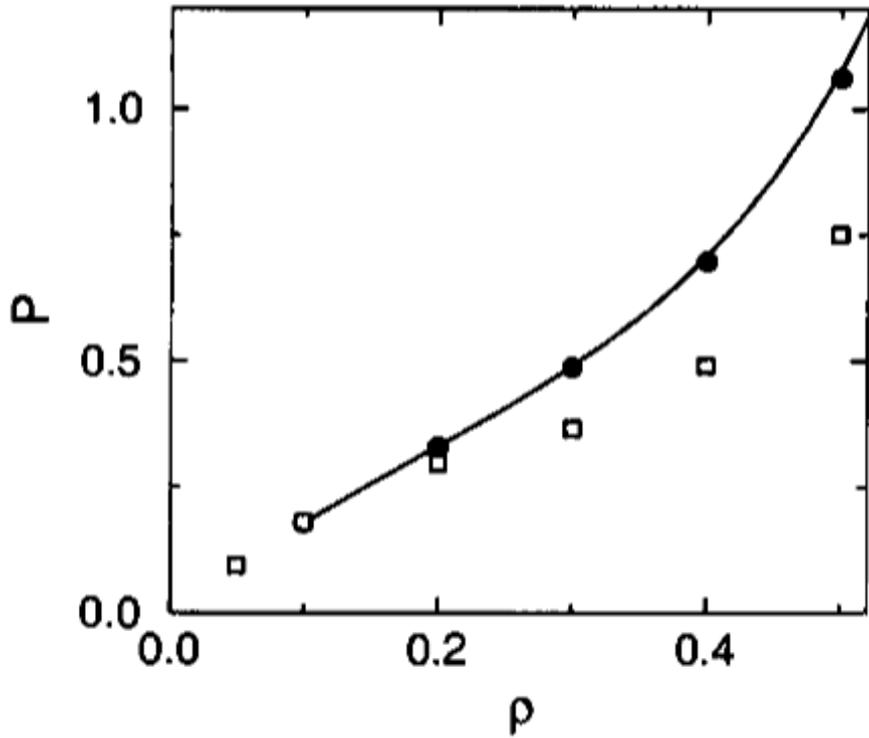


细致平衡条件破坏的后果

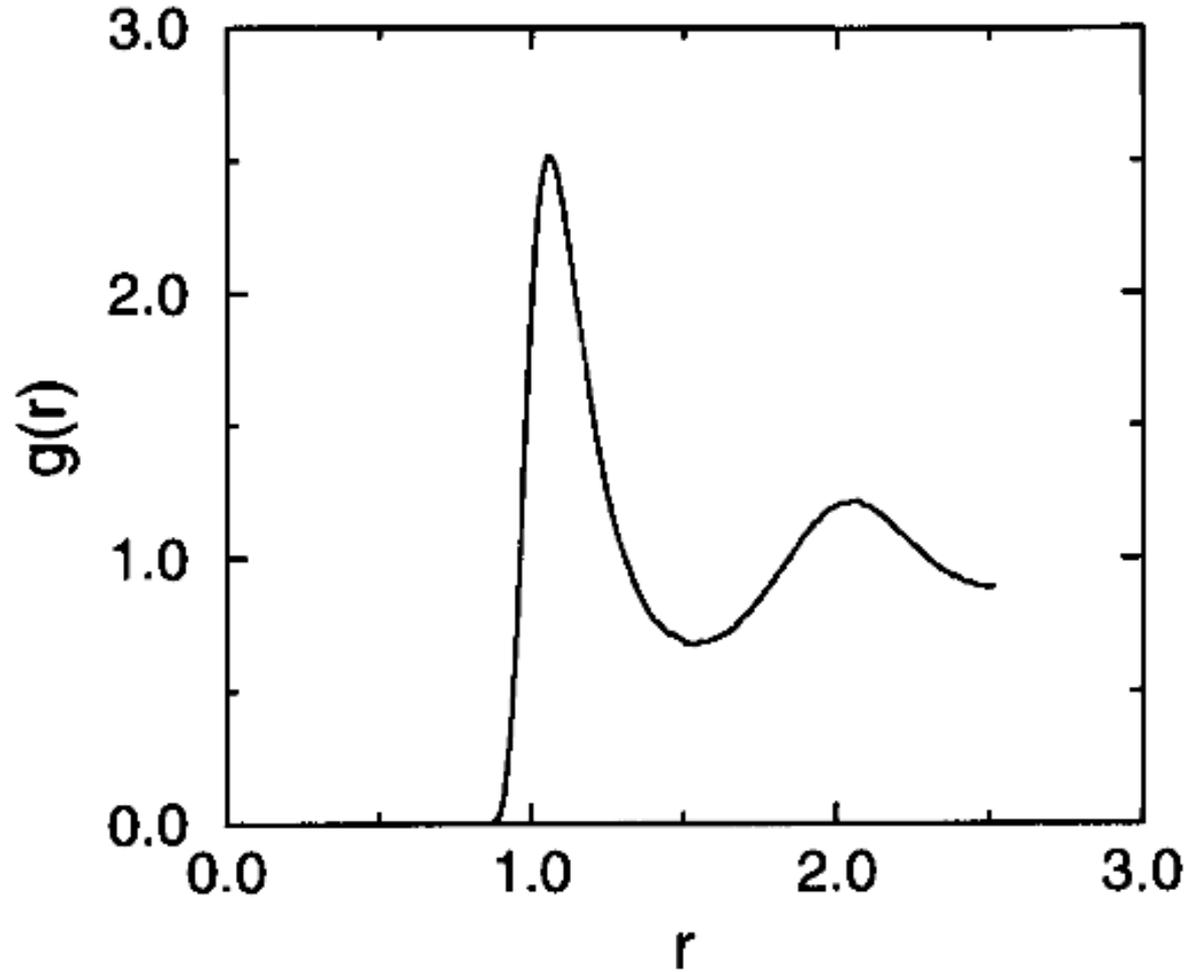


$$\delta x = \delta (\text{ran}(\cdot) - 0.0)$$

LJ 物态方程



细致平衡条件破坏的后果: 不断尝试, 直至移动



径向分布函数: $T = 1.5043$, $\rho = 0.8442$

模拟技术细节

● 选取产生一系列状态的方式.

一般来说, **Markov** 链中的每一个状态与其前一个状态相差应较小, 因为如果两个状态相差过大, 其能量差亦较大, 从而转移几率太小, 计算很容易陷入相空间中初态附近一个很小的子空间内.

模拟技术细节

讨论:

可以顺序选取每个粒子，也可以随机选取。在随机选取时，应使每个粒子平均说来被选的次数相同。通常每个粒子平均都被访问一次称为一个 **Monte Carlo 步(Monte Carlo Step or MCS)**，例如对于N个粒子，可以把N次粒子移动做为一个**MCS**。一次有价值的计算通常需要做几千或几万个**MCS**。有时，为了得到高精度的结果，甚至要作百万**MCS**以上的计算。

模拟技术细节

讨论:

由于每一个状态与其前导状态最多相差一个粒子移动，因而其物理性质具有很强的关联。这样，上述过程的最后一步不必对每次粒子移动都进行，而间隔一个或数个MCS (视问题的关联时间的大小) 进行一次。另外，如在前面已经指出过的，前面若干个MCS应舍弃。

模拟技术细节

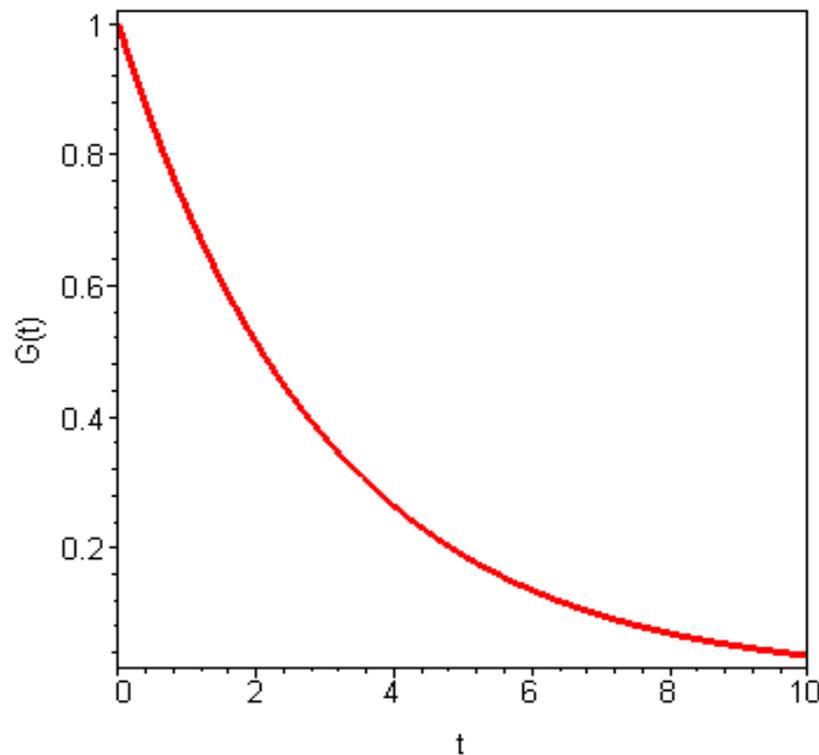
讨论:

每隔多少MCS取一次样的问题是由状态的关联决定的，考虑物理量A，并计算A的关联函数

$$G(t) = \frac{\langle A(t)A(0) \rangle - \langle A \rangle^2}{\langle A^2 \rangle - \langle A \rangle^2}$$

模拟技术细节

$G(t)$ 一般是指指数衰减的, 如图所示. t 的单位是MCS. 由于先后状态的关联, 可以导致误差估计的偏小, 可以证明, 如果 $G(t)$ 的衰减时间为 τ , 则实际误差与计算得到的误差估计差一个因子.



模拟技术细节

$$\langle (\delta A)^2 \rangle = \frac{1}{M} [\langle A^2 \rangle - \langle A \rangle^2] \left(1 + \frac{2\tau}{\delta t} \right)$$

δt 为取样间隔, 只有当 $\delta t \gg \tau$ 时, 修正因子才为 1, 当 $\delta t \sim 2\tau$ 时, 修正因子为 2. τ 一般为 2--3 个 MCS.

技术细节

- 计算能量是非常费时的，可以先算好，用时再查表。

自由能的计算

配分函数，态密度，熵等

自由能的计算

热力学积分方法，利用：

$$P = - \left(\frac{\partial F}{\partial V} \right) \Big|_T \quad E = \left(\frac{\partial \beta F}{\partial \beta} \right) \Big|_V$$

得到：

$$F(T, V_1) = F(T, V_0) - \int_{V_0}^{V_1} P(T, V) dV;$$
$$\frac{F(T_1, V)}{T_1} = \frac{F(T_0, V)}{T_0} - \int_{T_0}^{T_1} \frac{E(T, V)}{T^2} dT.$$

态密度

定义态密度为微观状态数和能量之间的关系，记为

$$\Omega(E)$$

对于分立能级

$$\Omega(E) = \sum_{E(X)=E} 1$$

配分函数可以用态密度表示出来

$$\begin{aligned} Q &= \sum_X e^{-\beta E(X)} = \sum_E \sum_{E(X)=E} e^{-\beta E(X)} \\ &= \sum_E \Omega(E) e^{-\beta E} \end{aligned}$$

如果 $\Omega(E)$ 已经求得，统计物理的问题实际上也就已经求出来了。

Ferrenberg-Swendsen 直方图方法

在给定温度 $T = 1/(k_B\beta)$, 做一个正则系综的模拟, 并做出能量的直方图 $H(E)$

$$H(E) \propto \Omega(E)e^{-\beta E}$$

由此, 态密度可以确定到一个常数因子 (原则上)

$$\Omega(E) \propto H(E)e^{+\beta E}$$

计算能量的矩

原则上，可以利用这个态密度计算任一温度下能量的各阶矩

$$\langle E^n \rangle_{\beta'} = \frac{\int E^n H(E) e^{(\beta-\beta')E} dE}{\int H(E) e^{(\beta-\beta')E} dE}$$

结果

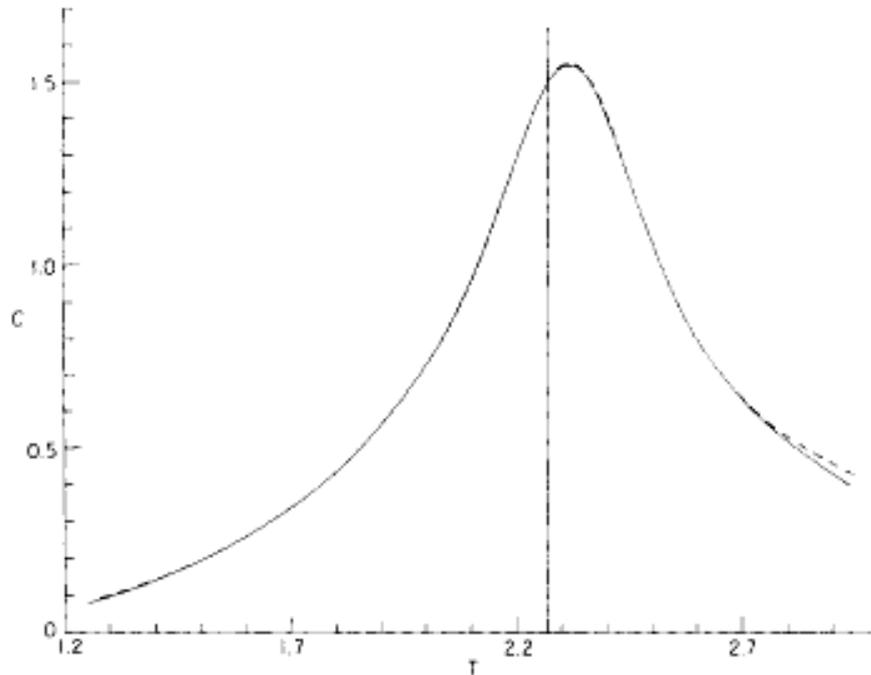
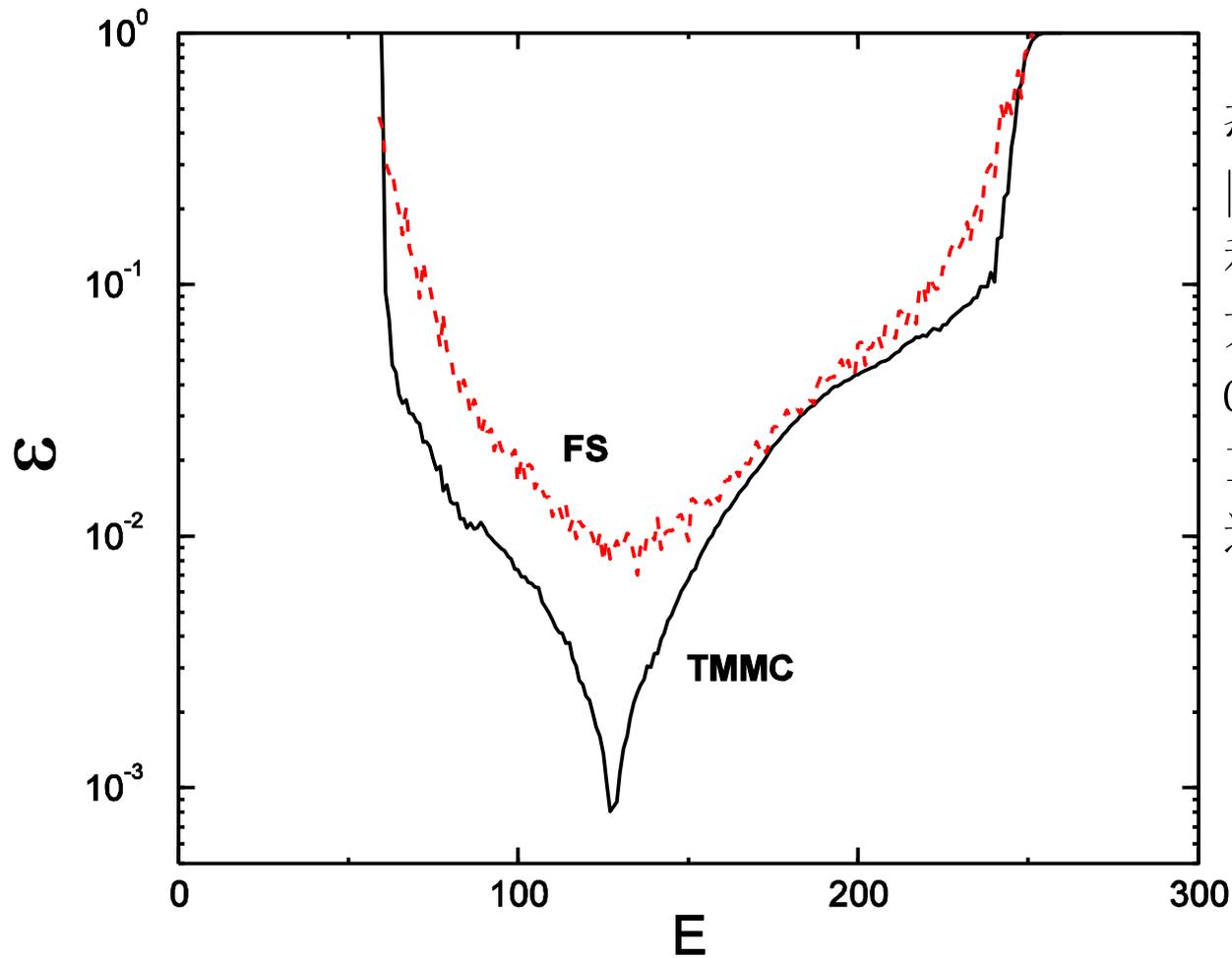


FIG. 1. Plot of specific heat vs T for the 16×16 $d=2$ Ising model. The dashed line is the exact solution (see Ref. 9) while the solid line is the result calculated from the single simulation at $T=T_c$. The location of the simulated temperature is marked with a vertical line.

2D Ising模型在 T_c 附近一次模拟结果，利用直方图方法外推到其它温度。

From Ferrenberg and Swendsen, Phys Rev Lett 61 (1988) 2635.

$\Omega(E)$ 精度, 或成立的范围



态密度的相对误差。

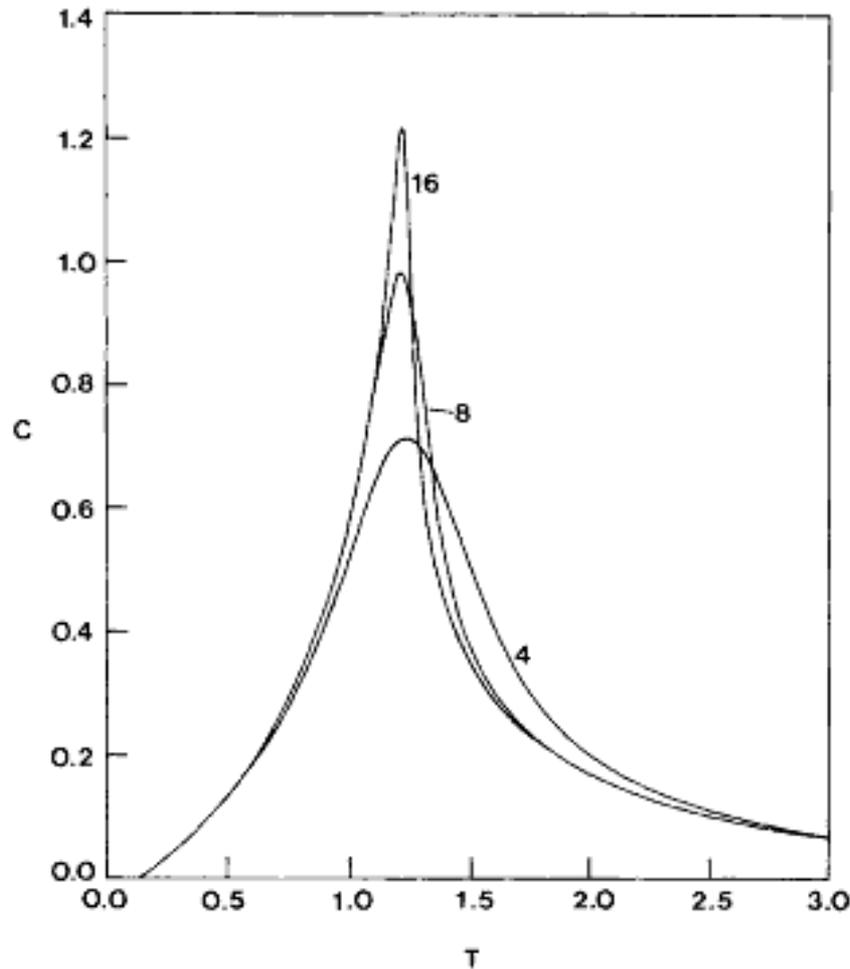
$$|\Omega_{MC}/\Omega_{exact} - 1|$$

利用 Ferrenberg-Swendsen
方法, 转移矩阵 Monte
Carlo, 2D 32×32
Ising 模型, T_C 附
近。

From J S Wang and R
H Swendsen, J Stat
Phys **106** (2002) 245.

多直方图方法

在几个不同的温度 T_i 做模拟
把所得结果， $H_i(E)$ ，合成得到更好的态密度
如何合成？



用多直方图方法计算的3
维3态反铁磁Potts 模型
的比热。

From J S Wang, R H
Swendsen, and R
Kotecký, Phys Rev Lett
63 (1989) 109.

FIG. 1. Specific heat of the $d=3$, $q=3$ Potts antiferromagnet as a function of temperature for $L=4, 8$, and 16 . The continuous functions in both figures were obtained from multiple-histogram analyses (Ref. 27). The run lengths in MC steps per site (and the number of runs used for the multiple-histogram analysis) are, respectively, 3×10^5 (10), 1.5×10^5 (14), and 10^5 (9).

多正则系综

多正则系综定义为使得能量的直方图为常数的系综

$$H(E) = \Omega(E)f(E) = \text{const.}$$

这表明位形的概率分布为

$$P(X) \propto f(E(X)) \propto \frac{1}{\Omega(E(X))}$$

如何做？

- 由概率分布 $f_n(E)$ ，利用Metropolis 算法做模拟，转移概率取为

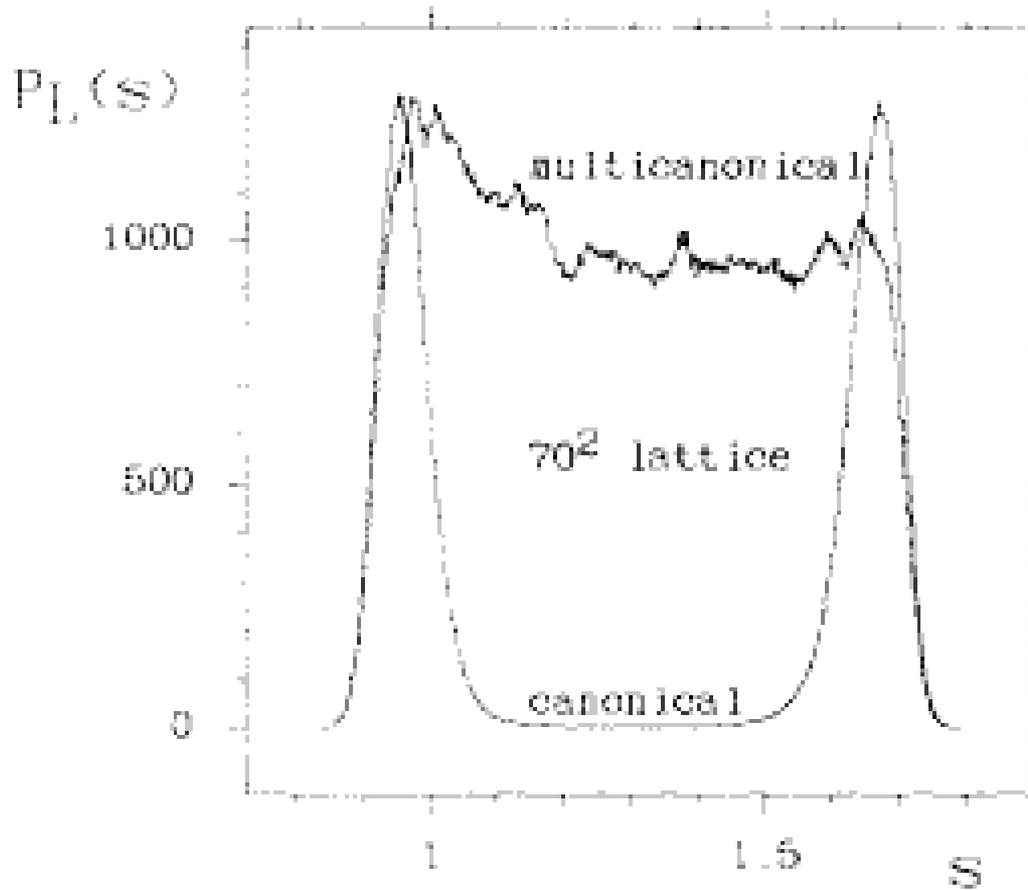
$$W(X \rightarrow X') \propto \min \left(1, \frac{f_n(X')}{f_n(X)} \right)$$

- 计算直方图 $H(E)$
- 由直方图得到新的分布

$$f_{n+1}(E) = \frac{f_n(E)}{H(E)}$$

- 迭代直到 $H(E)$ 变平

多正则， 计算结果



2D, 10态Potts模型
， 多正则

From A B Berg and
T Neuhaus, Phys
Rev Lett 68 (1992)
9.

自由能的计算

王福高-Landau方法, 能量空间的无规行走:

Fugao Wang and D. P. Landau, Phys. Rev. Lett. 86, 2050(2001);
Phys. Rev. E 64, 056101(2001).

自由能的计算

在过程中计算 $\Omega(E)$

以二维Ising模型为例，能量的可能取值为
 $-2NJ, -2NJ + 2J, \dots, 2NJ - 2J, 2NJ$

我们需要计算每个能级的状态数 $\Omega(E)$.

1. 赋初值： 令所有能级 E 的状态数 $\Omega(E)$ 均为1
2. 选定一个已知能量的位型
3. 以随机或顺序的方式翻转每一个自旋。设翻转前系统的能量为 E_1 ， 翻转后系统的能量为 E_2 。
为了使访问能级 E 的概率正比于该能级状态数 $\Omega(E)$ 的倒数， 从能级 E_1 到 E_2 的转移概率为 $\min[\Omega(E_1)/\Omega(E_2), 1]$
4. 按照接受状况更新状态数 $\Omega(E)$ ， 同时计算状态的
能量分布 $H(E)$

即：(1) 倘若当前的状态数是

$$\Omega(E_1) \geq \Omega(E_2)$$

则无条件接受新位型。

同时能级 E_2 的状态数 $\Omega(E_2)$ 乘以一个修正因子 $f(f > 1)$ ，并且能级 E_2 的访问次数 $H(E_2)$ 计数一次。

(2) 倘若当前的状态数是

$$\Omega(E_1) < \Omega(E_2)$$

则按几率 $\Omega(E_1)/\Omega(E_2)$ 接受新位型。如果接受新位型，那么能级 E_2 的状态数 $\Omega(E_2)$ 乘以一个修正因子 f ，并且能级 E_2 的访问次数 $H(E_2)$ 计数一次；如果不接受新位型，那么能级 E_1 的状态数 $\Omega(E_1)$ 乘以一个修正因子 f ，并且能级 E_1 的访问次数 $H(E_1)$ 计数一次。

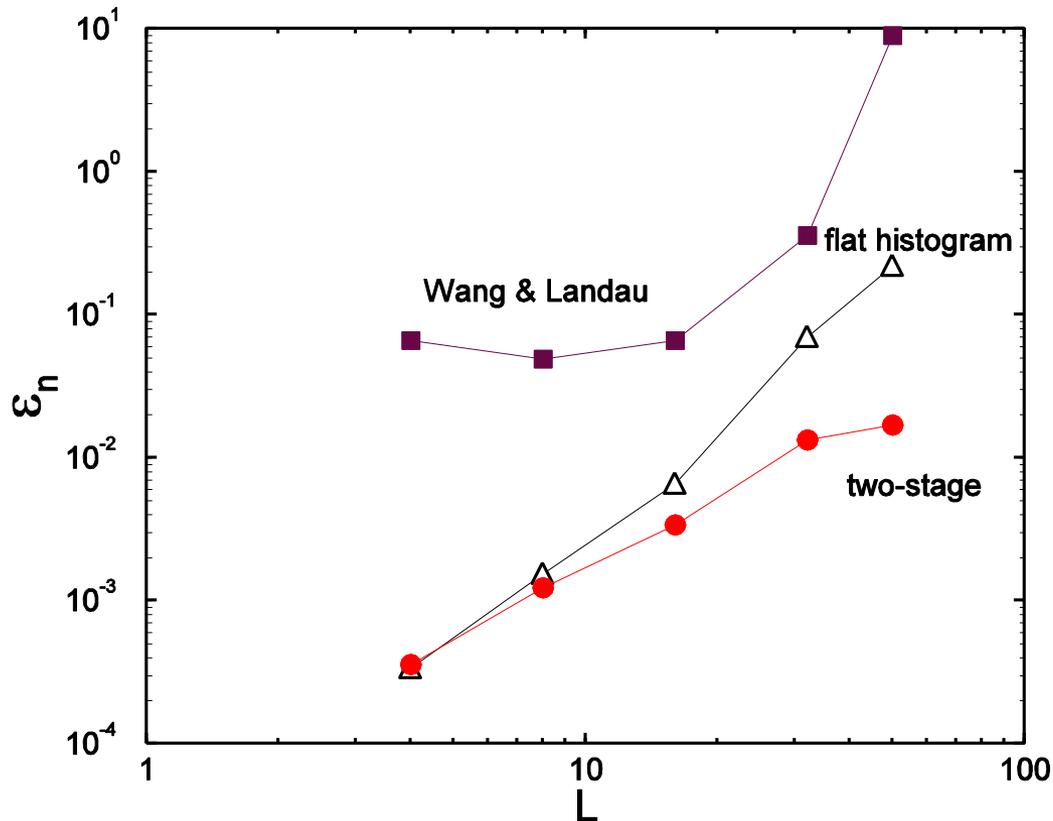
收敛的标志： $H(E) = \text{常数} !$

自由能的计算

其它重要方法:

- 1, Multistage Sampling (McDonald and Singer, 1967, 1969)
- 2, Finite Size method (Mon, 1985)
- 3, Particle-Insertion Method (B. Widom, 1962)
- 4, Density Scaling Monte Carlo (J P Valleau, 1991)
- 5,
- 6,

比较



2D Ising 模型态密度的误差

10^6 MCS.

From J S Wang and R H Swendsen, J Stat Phys 106 (2002) 245.

参考书

- Allen and Tildesley, Computer Simulation of Liquids
- Landau and Binder, A Guide to Monte Carlo Simulations in Statistical Physics
- Frenkel and Smit, Understanding Molecular Simulation
- Heermann and Binder, Computer Simulations in Theoretical Physics

分子动力学模拟

Consider a rectangular volume of $L_1 \times L_2 \times L_3$, with N classical particles put in. The particles are interact with each other. In principle, the interaction includes pair interactions, three body interactions as well as many body interactions. For simplicity we will consider here only pair interactions. In this case each particle feel a force by all other particles and we further assume the force depends only on distances from other particles and for each pair the force directed along the line join the pair of particles. So the force on the i th particle is

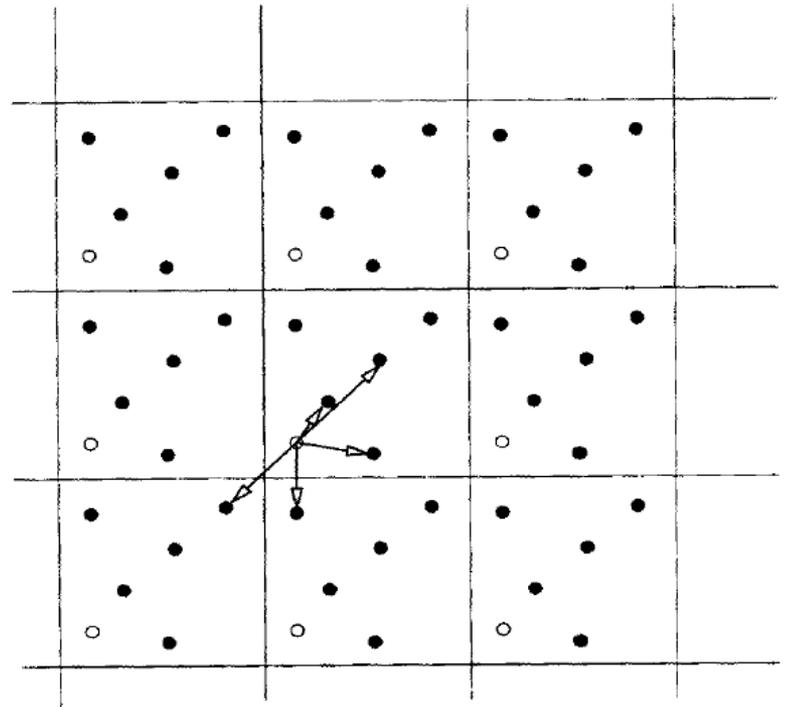
$$F_i = \sum_{j=1, N; j \neq i} F(|r_i - r_j|) \hat{r}_{ij},$$

where \hat{r}_{ij} is an unit vector along $r_i - r_j$.

Periodic boundary condition(PBC)

$$F_{PBC}(\mathbf{r}_i - \mathbf{r}_j) = \sum_{\mathbf{n}} F\left(\mathbf{r}_i - \mathbf{r}_j + \sum_{\mu} L_{\mu} n_{\mu}\right)$$

where L_{μ} are vectors along the edges of the rectangular system volume and the sum over \mathbf{n} is with all integers n_{μ} . Usually this sum is the most time consuming part in a simulation.



General procedure of MD (NVE ensemble)

1. Initialize;
2. Start simulation and let the system reach equilibrium;
3. Continue simulation and store results.

分子动力学模拟

Initialize:

- 1, Specify the number of particles and interaction;
- 2, Setup the simulation box;
- 3, Specify the total energy of the system;
- 4, Assign position and momenta of each particle.
 - a, In many cases we assign particles in a FCC lattice,
If we use cubic unit cell and cube BOX then the number of particles per unit cell is 4, and the total numbers of particles are $4M^3$, $M=1,2,3,\dots$. That is we may simulate systems with total number of particles $N=108, 256, 500, 864, \dots$.
 - b, The velocities of particles are draw from a Maxwell distribution with the specified temperature.
This is accomplished by drawing the three components of the velocity from the Gaussian distribution.

分子动力学模拟

The distribution of the x-component of velocity is

$$P(v_x) \propto \exp \left[-\frac{mv_x^2}{2k_B T} \right].$$

Draw numbers from a Gaussian:

Consider:

$$P(v_x, v_y) \propto \exp \left[-\frac{mv_x^2}{2k_B T} \right] \exp \left[-\frac{mv_y^2}{2k_B T} \right] = \exp \left[-\frac{m(v_x^2 + v_y^2)}{2k_B T} \right].$$

Then

$$P(v_x, v_y) dv_x dv_y = \tilde{P}(v) v dv d\phi,$$

where $v^2 = v_x^2 + v_y^2$ and

$$\tilde{P}(v) \propto \exp \left[-\frac{mv^2}{2k_B T} \right].$$

分子动力学模拟

So the distribution of v_x and v_y may be obtained from v and ϕ .

The distribution of v :

$$P(v) \propto v \exp \left[-\frac{mv^2}{2k_B T} \right]$$

The distribution of ϕ is uniform in the interval $[0, 2\pi]$.

Generate random numbers for a given distribution

For a given distribution $P(y)$ we consider how to get a random number y draw from $P(y)$ from a random number x draw from uniform $[0,1]$, i.e., we are going to find a function $f(x)$, from which for a series of random numbers x distributed uniformly in the interval $[0,1]$, $y = f(x)$ will distributed according to $P(y)$.

分子动力学模拟

Since

$$\int_{-\infty}^y P(y') dy' \equiv F(y) = \int_0^x dx = x,$$

then

$$y = F^{-1}(x)$$

Exponential distribution

$$P(y) = \begin{cases} 0, & y < 0 \\ \lambda e^{-\lambda y}, & y > 0 \end{cases}.$$

$$F(y) = \int_{-\infty}^y P(y') dy' = \begin{cases} 0, & y < 0 \\ 1 - e^{-\lambda y}, & y > 0 \end{cases},$$

$$y = f(x) \equiv F^{-1}(x) = -\frac{1}{\lambda} \ln(1 - x).$$

分子动力学模拟

The distribution of v :

$$P(v) = \frac{m}{k_B T} v \exp \left[-\frac{mv^2}{2k_B T} \right]$$

$$\frac{m}{k_B T} \int_0^v v \exp \left[-\frac{mv^2}{2k_B T} \right] dv = -\exp \left[-\frac{mv^2}{2k_B T} \right] + 1.$$

$$v = \sqrt{-\frac{k_B T}{m} \ln(1-x)}$$

分子动力学模拟

Draw random numbers uniformly distributed in the interval $[0, 2\pi]$.

$$v_x = v \cos \phi$$

$$v_y = v \sin \phi$$

Another method of draw random numbers in the Gaussian distribution is through the following empirical methods.

Consider the distribution

$$\exp \left[-\frac{x^2}{2} \right]$$

分子动力学模拟

According to the central limit theorem, if we draw uniform random numbers r_i in interval $[0,1]$, and define a variable

$$\xi = \frac{\frac{1}{n} \sum_{i=1}^n r_i - \frac{1}{2}}{\sqrt{\frac{1}{12n}}}$$

when $n \gg 1$ the distribution of ξ is the Gaussian distribution

$$\exp \left[-\frac{\xi^2}{2} \right]$$

If we take $n = 12$, we get

$$\xi = \sum_{i=1}^{12} r_i - 6.$$

分子动力学模拟

After the generation of the velocity of each particle, we may shift the velocity so that the total momentum is zero.

The standard Verlet algorithm is the first successful method in history and still wide used today in different forms. It is

$$r(t+h) = 2r(t) - r(t-h) + h^2 F(r(t)) / m$$

$$v(t) = \frac{r(t+h) - r(t-h)}{2h}.$$

To start the integration we need $r(h)$, given by

$$r(h) = r(0) + hv(0) + h^2 F(r(0)) / m.$$

分子动力学模拟

Variations of this method are

$$v(t + h/2) = v(t - h/2) + hF(r(t))$$

$$r(t + h) = r(t) + hv(t + h/2).$$

and

$$r(t + h) = r(t) + hv(t) + h^2F(r(t))$$

$$v(t + h) = v(t) + h \frac{F(r(t + h)) + F(r(t))}{2}.$$

Both of these variations are mathematically equivalent to the original one but more stable under finite precision arithmetic.

分子动力学模拟

The temperature of the system is given by the equal partition theorem, that is the average of kinetic energy of each degree of freedom is half $k_B T$,

$$\frac{3}{2}k_B T_D = \frac{1}{N-1} \sum_{i=1}^N \left\langle \frac{1}{2} m v_i^2 \right\rangle.$$

The $N-1$ is due to the conservation of the total momentum which reduce the degree of freedom by 3.

To reach the desired temperature we may scale the velocity at every few steps of integration

$$v_i(t) \rightarrow \lambda v_i(t)$$

$$\lambda = \sqrt{\frac{(N-1) 3k_B T}{\sum_{i=1}^N m v_i^2}}$$

分子动力学模拟

After the system reach to equilibrium the integration continue in the same method as above without scaling of velocity. The data are stored or accumulated for the calculating of physical properties. The static properties of physical quantity A is given by time average

$$\bar{A} = \frac{1}{n - n_0} \sum_{\nu > n_0}^n A_\nu$$

分子动力学模拟

here A_ν is the value of A at ν th time step. Usually the data stored in each step include:

1, the kinetic energy $\sum_{i=1}^N \frac{1}{2} m v_i^2$

2, the potential energy $U = \sum_{\langle i,j \rangle} v(r_{ij})$

3, the virial $\sum_{\langle i,j \rangle} r_{ij} \frac{\partial v(r_{ij})}{\partial r_{ij}}$.

分子动力学模拟

We also need data to calculate the pair correlation function, this is done by divide the interval $[0, r]$ into sub intervals $[ir, (i + 1)r]$, at each stage of updating, add the number of pairs with separation in the interval $[ir, (i + 1)r]$, to an array $n(i)$ and find the average value after simulation, the pair correlation function is given by

$$g(r) = \frac{2V}{N(N-1)} \frac{\langle n(r) \rangle}{4\pi r^2 \Delta r}$$

分子动力学模拟

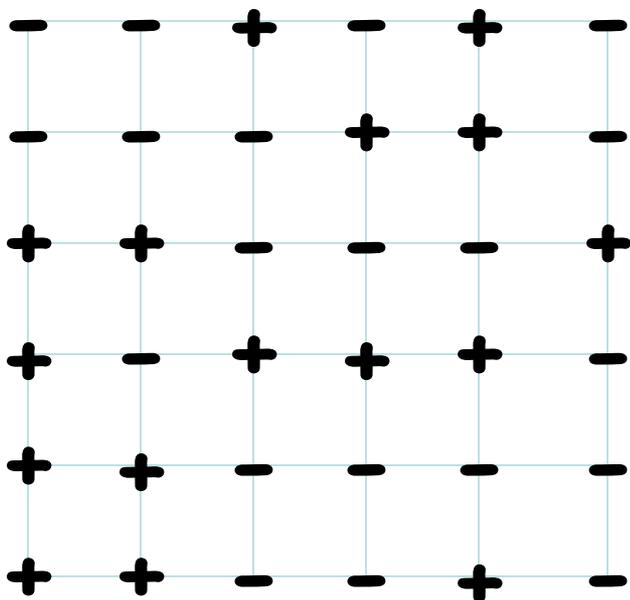
<http://lammps.sandia.gov/>

<http://www.gromacs.org/>

<http://www.ccp5.ac.uk/>

<http://netlib.org/>

The Ising Model



The energy of configuration s is

$$E(s) = -J \sum_{\langle ij \rangle} s_i s_j$$

where i and j run over a lattice, $\langle ij \rangle$ denotes nearest neighbors, $s = \pm 1$

$$s = \{s_1, s_2, \dots, s_i, \dots\}$$

例题, Ising模型的模拟

Ising 模型:

$$H = -J \sum_{nn} s_i s_j - h \sum_i s_i$$

式中J称为交换积分, h为外场, s_i 可取值(1, -1), 称为自旋变量. Ising 模型是最简单的非平庸统计物理模型, 它是由德国物理学家 Lenz 在二十年代提出的, 这一模型可用来描述单轴各向异性磁性系统, 合金等物理体系, 同时也是一个十分有兴趣的理论模型.

例题, Ising模型的模拟

Ising 最早给出了这一模型在一维情况下的严格解, 证明了在一维下这一模型不存在相变. Onsager 于1944 年做出了零场下这一模型在二维空间的严格解并计算了它的相变温度, 比热在相变点的行为等热力学量. 杨振宁在1952 年解出了外场很小时二维空间的 Ising 模型, 求出了序参量的临界行为.

由于对这一模型的很多形为目前了解的比较透彻，因此它经常被用来做为检验各种数值方法或解析近似方法的标准。

1969年，A. E. Ferdinand和M. E. Fisher求得了有限尺寸2D Ising 模型在周期性边界条件下的严格解，成为检验模拟结果的一个有效标准。

感兴趣的物理量

1. 平均能量 $\langle E(s) \rangle$
2. 比热，利用下面的公式：

$$c = \frac{1}{Nk_B T^2} (\langle E^2 \rangle - \langle E \rangle^2)$$

3. 磁化 $\langle M \rangle = \langle |\sum_i s_i| \rangle$

4. 磁化率

$$k_B T \chi = \langle M^2 \rangle - \langle M \rangle^2$$

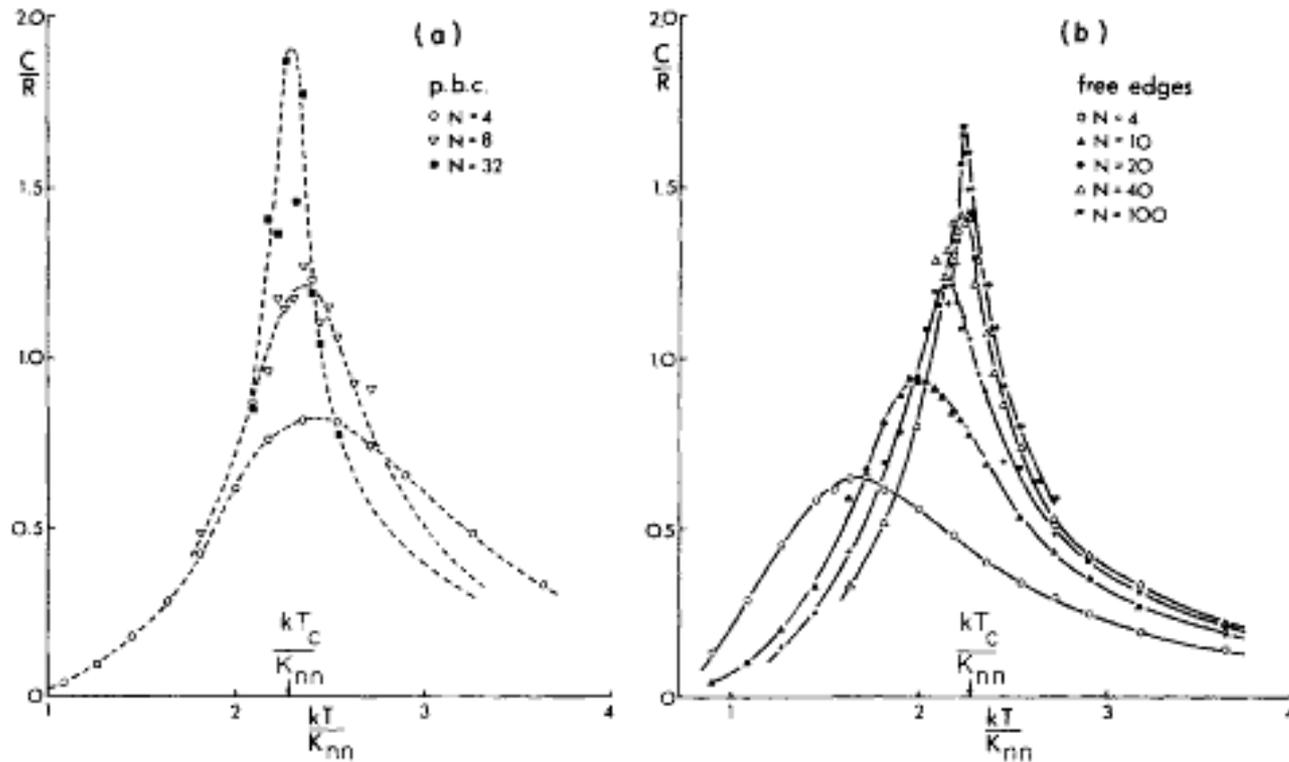
5. Binder 4 阶累积量

$$U = 1 - \frac{\langle M^4 \rangle}{3 \langle M^2 \rangle^2}$$

6. 自旋相关函数 $\langle s_i s_j \rangle$

7. 时间相关函数 $\langle E(t') E(t' + t) \rangle$

2D Ising 模型的比热



From D P Landau, Phys Rev B 13 (1976) 2997.

一个算法:

- 选择一个格点 i , 其自旋将考虑作翻转 $s_i \rightarrow -s_i$.
- 计算与此翻转相联系的能量变化 δH .
- 计算这一翻转的转移几率 W .
- 产生一在 $[0, 1]$ 之间均匀分布的随机数 ξ .
- 如果 $\xi < W$, 则翻转该自旋, 否则, 保持不变. 不论何种情况, 其结果都作为一新的状态.
- 分析该状态, 为计算平均值收集数据.

```

void monte_carlo( )
{
    int i, j, k, e;          /* i is the center site */
    int nn[Z];              /* the name neighbors */

    for(k = 0; k < N; ++k) {
        i = r250( ) * (double) N; /* pick site at random */
        neighbor(i, nn);          /* find neighbors of site i */
        for(e = 0, j = 0; j < Z; ++j) /* go over the neighbors */
            e += s[nn[j]];        /* sum of the neighbor spins */
            e *= 2 * s[i];        /* 2 times the center spin */
            if (e <= 0)           /* when energy change is less */
                s[i] = - s[i];   /* than zero, spin is flipped */
            else if (r250( ) < exp(-e/T)) /* otherwise, it is flipped */
                s[i] = - s[i];   /* with probability less one */
    }
}

```

```

void neighbor(int i, int nn[ ])
{
    int j, r, p, q;

    r = i;
    p = 1 - L;
    q = 1;

    for(j = 0; j < Z; j += 2) {
        nn[j] = (r + 1) % L == 0 ? i + p : i + q;
        nn[j+1] = r % L == 0 ? i - p : i - q;
        r = r/L;
        p *= L;
        q *= L;
    }
}

```

```
void energy(double *e)
{
    int i, j, ie = 0;
    int nn[Z];

    for(i = 0; i < N; ++i) {
        neighbor(i, nn);
        for(j = 0; j < Z; j += 2) /* look at positive direction only */
            ie += s[i]*s[nn[j]];
    }
    assert(ie <= 2*N && ie >= -2*N);
    *e += ie;
}
```

```
void main( )
{
    int i, mc;
    double e = 0;

    for (i = 0; i < N; ++i)    /* initialize, all spin up */
        s[i] = 1;

    for(mc = 0; mc < MCTOT; ++ mc) {
        monte_carlo();
        if( mc >= MCDIS)
            energy(&e);
    }
    printf("<e> = %f\n", e/(MCTOT-MCDIS)/N);
}
```

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>

                /* macro definitions */
#define L 16                /* lattice linear size */
#define N (L*L)            /* total number of spins */
#define Z 4                /* coordination number = 2*d */
#define MCTOT 1000        /* total Monte Carlo steps */
#define MCDIS 500         /* steps discarded in the beginning */

                /* global variables */
int s[N];                /* spin +1 or -1 */
double T = 2.269;        /* temperature */

                /* function prototypes */
void neighbor(int i, int nn[ ]);
void monte_carlo();
void energy(double *);

```

讨论:

关于每一步要翻转的格点 i 的选择, 一般来说可有很多种不同的方法, 最常用的有两种, 一种是**顺序取每一个格点**, 另一种是**随机的选取**. 在随机选取时, 应使每个格点平均说来被访问的次数相同, 通常每个格点被访问一次称为一个 Monte Carlo 步 (Monte Carlo Step or MCS), 一次有价值的计算通常需要做几千或几万个MCS. 有时, 为了得到高精度的结果, 甚至要作百万MCS以上的计算.

由于每一个状态与其前导状态最多相差一个自旋翻转，因而其物理性质具有很强的关联。这样，上述过程的第6步不必对每次自旋都进行，而是每间隔一个或数个MCS (视问题的关联时间的大小) 进行一次。另外，如在前面已经指出过的，前面若干个MCS应舍弃。

计算能量差是最费时的工作，对于Ising 模型，由于能量差只能取很少几个数值，我们可以预先算好存起来以节省计算量。这一技巧不仅适用于Ising 模型，也适用于其它分立变量的模型如 Potts 模型等。

模拟技术细节

为了模拟Ising 模型:

☺ 要确定一个晶格和晶格的尺寸.

例如, 我们可以取一个简单立方格子, 取三个方向的大小均为 L ;

其它取法包括FCC, BCC, 金刚石结构, 六角密堆结构等.

模拟的盒子可以取为立方体, 也可以取其它形状, 如各种多面体; 如果利用周期性边界条件, 则对盒子的形状有限制——盒子的周期性排列应该能够填满整个空间。

☺ 要指定一个初始位形.

在临界温度之下进行计算，取所有自旋沿同一方向为初始位形.

在临界温度之下的平衡位形是有序的，若从一无序位形出发，系统在演变中将形成若干个有序的区域，相邻区域的边界上将出现畴壁。（畴壁具有一定的能量，是一种拓扑性缺陷，在周期性边界条件下很难通过长时间的模拟消除）.

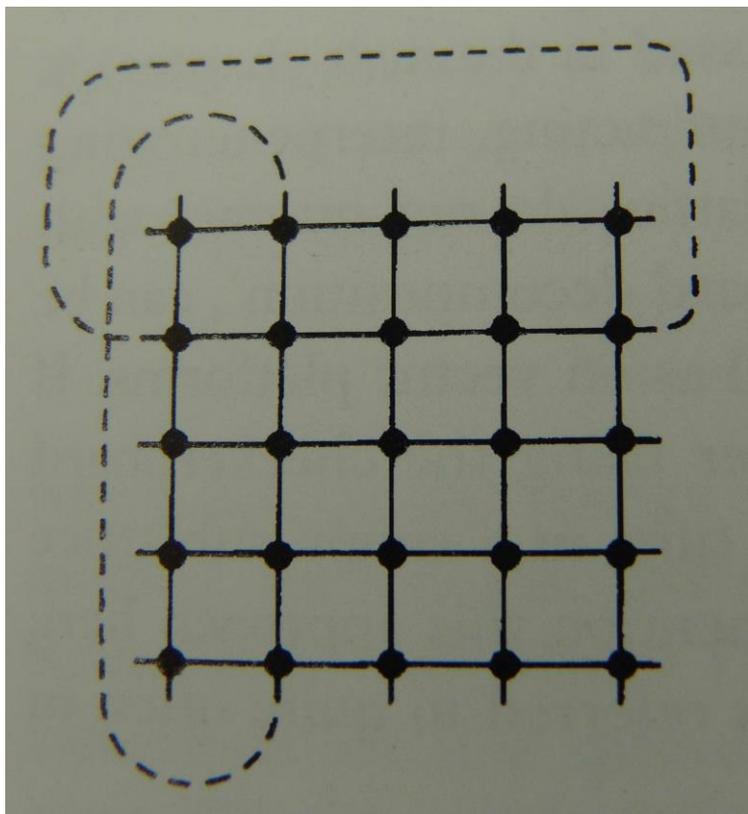
☺ 要指定一个初始位形.

如果计算的目的是为了研究畴壁，则必须从一自旋取向为随机分布的初始位形出发；或直接产生畴壁，研究其性质。

在临界温度之上进行计算，取随机分布的自旋取向为初始位形或取所有自旋沿同一方向为初始位形。

模拟技术细节

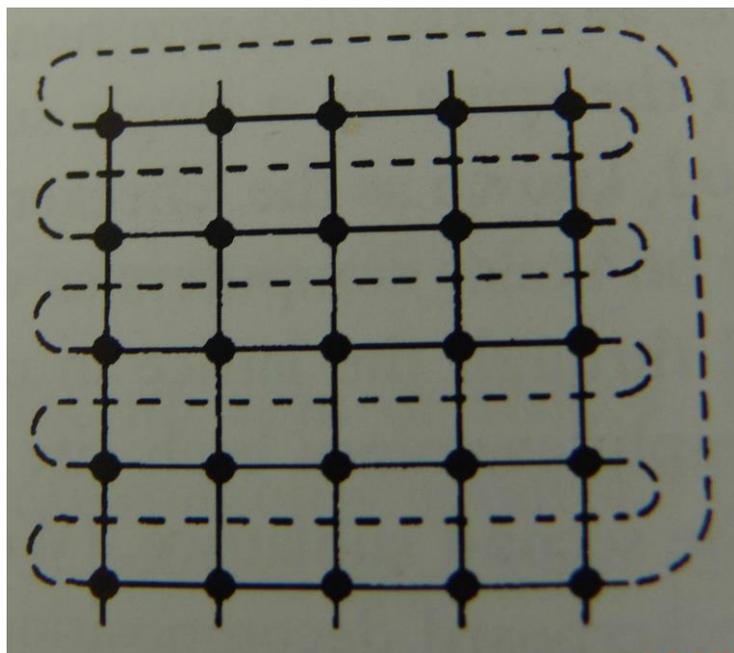
☺ 要确定适当的边界条件



因为计算总是对有限大小的系统进行的, 选择合适的**边界条件**对于得到好的结果是十分重要的, 如果我们感兴趣的是系统的体性质, 则应尽量消除边界的影响, 此时一般取周期性边界条件, 即在每个方向上, 取 $s_{L+1}=s_1$.

模拟技术细节

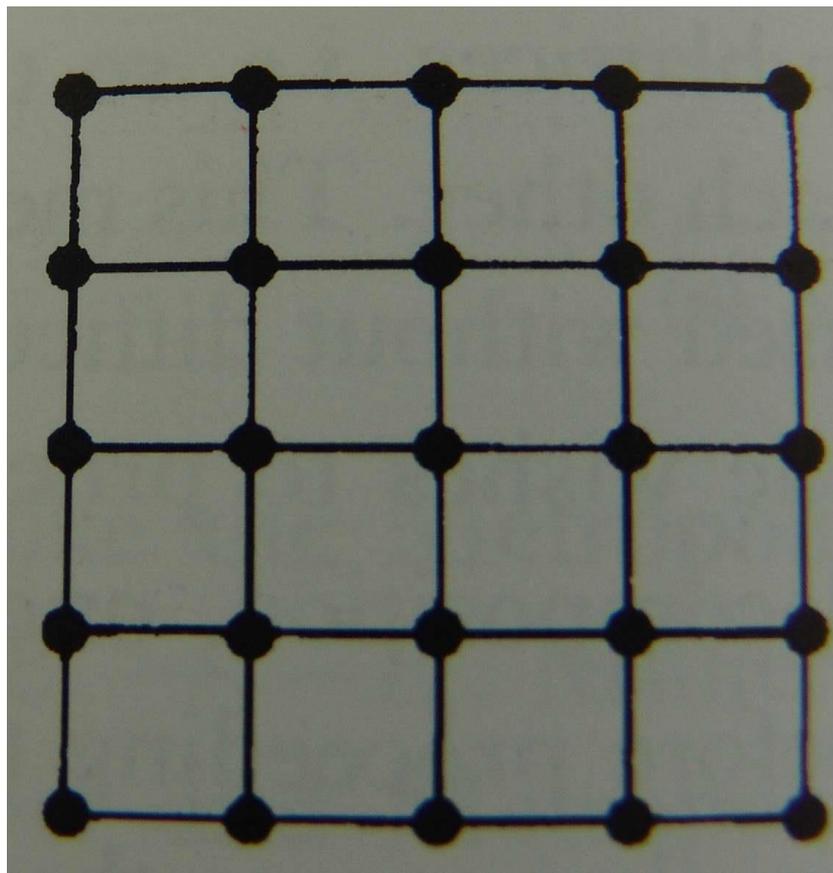
☺ 要确定适当的边界条件



周期性边界条件对于写并行计算的程序不太方便，而并行计算是当前计算机发展的一个重要方向，因此，另一种称为螺旋周期边界条件得到了较多的应用，以二维正方格子为例，这种边界条件是让每一行的最后一个自旋与其下一行的第一个自旋相同。

其它边界条件:

自由边界条件



自由边界条件+周期性边界条件

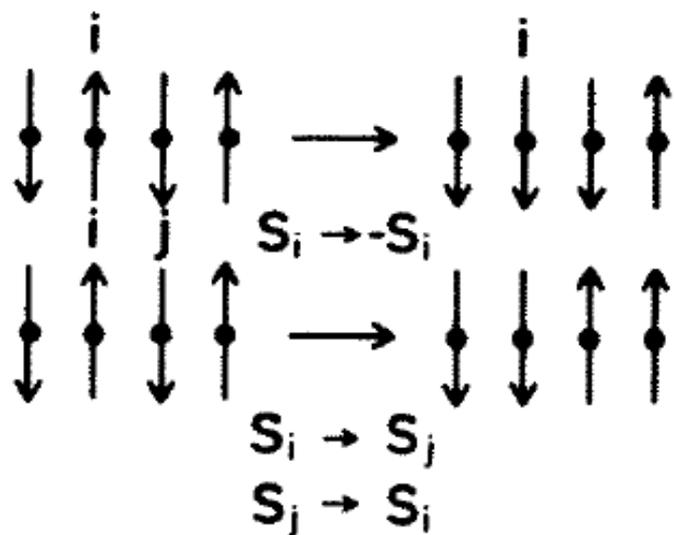
平均场边界条件

固定和部分固定边界条件

反周期边界条件(用于界面的研究)

模拟技术细节

● 选取产生一系列状态的方式.



一般来说, Markov 链中的每一个状态与其前一个状态相差应较小, 因为如果两个状态相差过大, 其能量差亦较大, 从而转移几率太小, 计算很容易陷入相空间中初态附近一个很小的子空间内.

模拟技术细节

● 选取产生一系列状态的方式.

通常有两种作法，一种是一次翻转一个自旋，这是一种不保持总自旋守恒的计算，另一种是每次交换一对相邻自旋，这种计算将保持总自旋守恒.

模拟技术细节

讨论:

在随机选取时, 应使每个格点平均说来被访问的次数相同. 通常格子上每个格点平均都被访问一次称为一个 **Monte Carlo 步**(**Monte Carlo Step or MCS**), 例如对于 $N=L \times L \times L$ 的格子, 可以把自旋翻转 N 次做为一个 **MCS**. 一次有价值的计算通常需要做几千或几万个 **MCS**. 有时, 为了得到高精度的结果, 甚至要作百万 **MCS** 以上的计算.

讨论:

关于每一步要翻转的格点 i 的选择, 一般来说可有很多种不同的方法, 最常用的有两种, 一种是**顺序取每一个格点**, 另一种是**随机的选取**.

讨论:

由于每一个状态与其前导状态最多相差一个自旋翻转，因而其物理性质具有很强的关联。这样，上述过程的第6步不必对每次自旋都进行，而是每间隔一个或数个MCS (视问题的关联时间的大小) 进行一次。另外，如在前面已经指出过的，前面若干个MCS应舍弃。

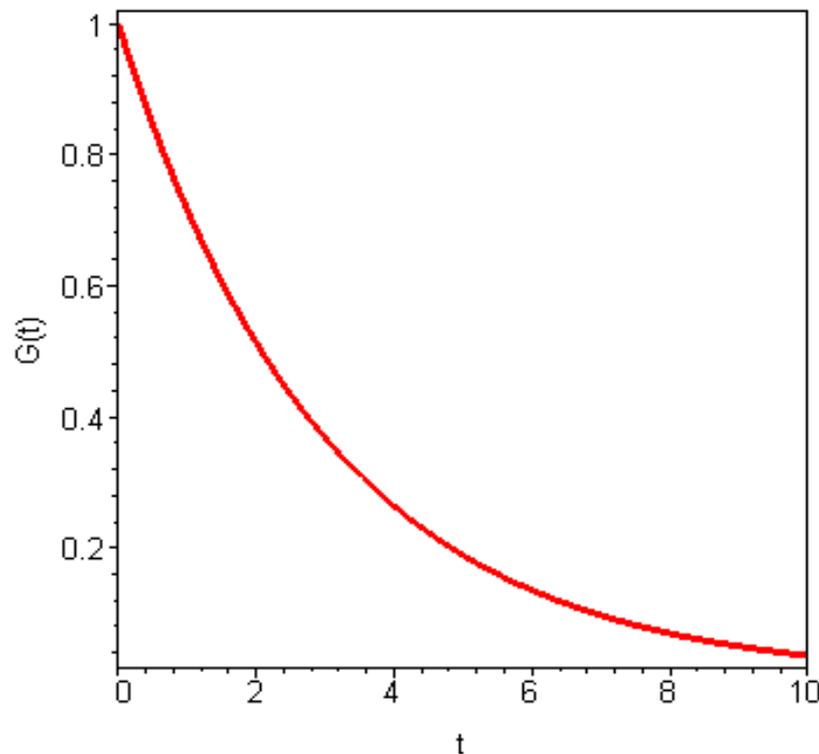
讨论:

每隔多少MCS取一次样的问题是由状态的关联决定的, 考虑物理量A, 并计算A的关联函数

$$G(t) = \frac{\langle A(t)A(0) \rangle - \langle A \rangle^2}{\langle A^2 \rangle - \langle A \rangle^2}$$

模拟技术细节

$G(t)$ 一般是指指数衰减的, 如图所示. t 的单位是MCS. 由于先后状态的关联, 可以导致误差估计的偏小, 可以证明, 如果 $G(t)$ 的衰减时间为 τ , 则实际误差与计算得到的误差估计差一个因子.



模拟技术细节

$$\langle (\delta A)^2 \rangle = \frac{1}{M} [\langle A^2 \rangle - \langle A \rangle^2] \left(1 + \frac{2\tau}{\delta t} \right)$$

δt 为取样间隔, 只有当 $\delta t \gg \tau$ 时, 修正因子才为 1, 当 $\delta t \sim 2\tau$ 时, 修正因子为 2. τ 一般为 2--3 个 MCS.

计算能量差是最费时的工作，对于Ising 模型，由于能量差只能取很少几个数值，我们可以预先算好存起来以节省计算量。这一技巧不仅适用于Ising 模型，也适用于其它分立变量的模型如 Potts 模型等。

例子，对于正方格子上的Ising模型，每个自旋(设为 s_0)有四个近邻，设为 s_1, s_2, s_3, s_4 ，则 s_0 与近邻的相互作用能量为：

$$E_0 = -Js_0(s_1 + s_2 + s_3 + s_4)$$

翻转一个自旋(S_0)可能的能量差为

$$\delta E_0 = 2Js_0(s_1 + s_2 + s_3 + s_4)$$

s_0 是翻转前的自旋值，于是

$$\delta E_0 = 8J, \quad 4J, \quad 0, \quad -4J, \quad -8J$$

求出 E_0 和 $\exp(-\beta E_0)$ 的几个数值
保存在数组中
需要时查表
可以节省很多计算时间.

其它常用的模型：

Potts 模型：

$$H = -J \sum_{\langle ij \rangle} \delta_{\sigma_i \sigma_j}$$

其中的求和对近邻格点对进行。 σ_i 可以取值 $1, 2, \dots, q$ ，当 $q = 2$ 时，对应于Ising模型。对于二维晶格， $q < 5$ 时，系统具有二级相变，当 $q \geq 5$ 时，系统具有一级相变。对于三维晶格，当 $q \geq 3$ 时为一级相变。

其它常用的模型：

在转变的 q 处 ($q = 3(3D)$, $q = 5(2D)$), 是非常弱的一级相变, 通过模拟很难与二级相变区分。Potts模型描述了从二级相变到弱一级相变再到强一级相变的各种情况, 是验证各种理论方法和模拟方法的试验模型。

Potts 模型的模拟方式与Ising模型非常相似。代替自旋的翻转, 可以在 $q - 1$ 个态中随机取一个。

Heisenberg类模型:

$$H = -J \sum_{\langle ij \rangle} S_i \cdot S_j$$

S_i 是一个定义在格点上的单位矢量，一般有三个分量，最早由Heisenberg提出以描述铁磁性(原始的Heisenberg模型中 S_i 是自旋算符)。如果把自旋限制在平面上， S_i 只有二个分量，这样的模型称为XY模型，在二维XY模型中发现了很多非常有趣的物理现象。如果把此模型做一推广，使得 S_i 有 n 个分量，则成为所谓的 n -矢量模型，当 $n \rightarrow \infty$ 时，成为球模型，而当 $n \rightarrow 0$ 时，可以用来描述高分子链的相变等问题。

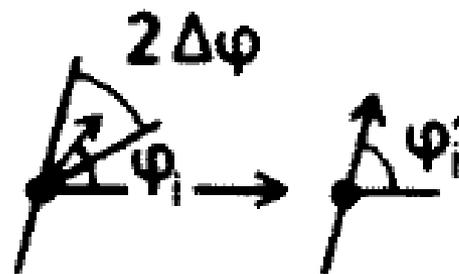
模拟技术细节

XY模型的update方法:

$$\mathcal{H}_{XY} = -J \sum_{\langle i,j \rangle} (S_i^x S_j^x + S_i^y S_j^y) - H_x \sum_i S_i^x \quad (S_i^x)^2 + (S_i^y)^2 = 1$$



$$\varphi_i \rightarrow \varphi'_i = 2\pi\zeta$$



$$\varphi_i \rightarrow \varphi'_i = \varphi_i + \Delta\varphi (2\zeta - 1)$$

还有一些有趣的模型，如

有次近邻的Ising 模型：

$$H = -J_1 \sum_{\langle ij \rangle} S_i S_j - J_2 \sum_{\langle\langle ij \rangle\rangle} S_i S_j$$

自旋玻璃模型：

$$H = - \sum_{\langle ij \rangle} J_{ij} S_i S_j$$

J_{ij} 随机取值。

等等 。