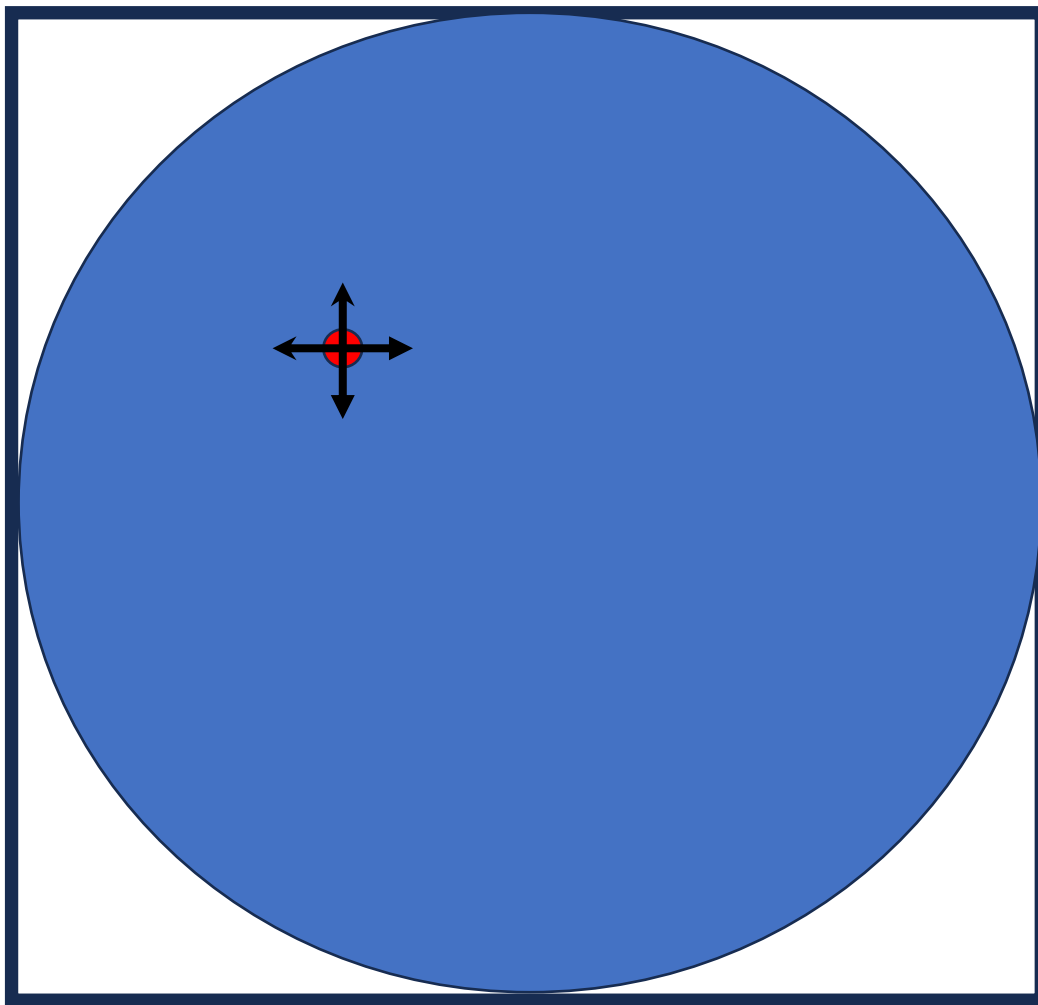


# 蒙特卡洛算法：随机游走

计算物理b

高阳

# 回顾



假如圆的面积过大（圆形球场！）  
或者你的力气太小

采取投石（投点）法

在某个点的位置时，前后投的距离随机分布在 $(-\epsilon, \epsilon)$ ，并且左右投的距离也随机分布在 $(-\epsilon, \epsilon)$ 。前后与左右投是不相关的。此距离比圆和方形的尺寸要小很多。

投好点之后移动到那个点，然后接着投点。

这是一个马尔可夫过程，也即下一步的位置只与当前步相关（虽然当前步与前一步相关）。

# 算法思路与目标

- 从区域内随机某个点出发。
- 按照随机规则往左右投，再往前后投。
- 走到新的位置。如果此位置在圆内则计数器加1.
- 重复N步。
- 用计数器的值除以N，此即为点在圆内的概率。
- 用随机游走的方法抽样出了在正方形内的均匀分布！

# 一维随机游走(1)

- 考虑一个一维的晶格，有一个醉汉从 $x=0$ 处开始行走，每次有 $p$ 的概率向右一格，或者 $q$ 的概率向左一格( $p+q=1$ )。当醉汉行走 $N$ 步时：
- 醉汉位置在第 $k$ 个格点 ( $k$ 的奇偶性一定与 $N$ 相同)：向右的步数为 $\frac{N+k}{2}$ ，向左的步数为 $\frac{N-k}{2}$ ，
- 故概率为 $p(k, N) = 2^{-N} p^{(N+k)/2} q^{(N-k)/2} C_N^{(N+k)/2} = 2^{-N} p^{(N+k)/2} q^{(N-k)/2} \frac{N!}{\frac{N+k}{2}! \frac{N-k}{2}!}$
- 醉汉的平均位置

$$\langle k \rangle = \sum_k k p(k, N) = \sum_k 2^{-N} p^{(N+k)/2} q^{(N-k)/2} k \frac{N!}{\frac{N+k}{2}! \frac{N-k}{2}!} = I(p)$$

$$\text{定义 } 1 = T(p) = \sum_k 2^{-N} p^{(N+k)/2} q^{(N-k)/2} \frac{N!}{\frac{N+k}{2}! \frac{N-k}{2}!}$$

$$0 = \frac{dT}{dp} = \frac{N}{2p} T(p) + \frac{I(p)}{2p} - \frac{N}{2q} T(p) + \frac{I(p)}{2q} \Rightarrow I(p) = N(p - q)$$

## 一维随机游走(2)

- 同法可算平方平均:

$$\frac{dT}{dp} = \frac{N}{2p} T(p) + \frac{I(p)}{2p} - \frac{N}{2q} T(p) + \frac{I(p)}{2q}$$

$$\frac{dI}{dp} = \frac{N}{2p} I(p) + \left( \frac{1}{2p} + \frac{1}{2q} \right) \langle k^2 \rangle - \frac{N}{2q} I(p) = 2N$$

$$\langle k^2 \rangle = 4Npq + (p - q)^2 N^2 \quad \text{方差为 } \langle k^2 \rangle - \langle k \rangle^2 = 4Npq$$

- 左右均匀的情况下

$$\langle k \rangle = 0$$

$$\langle k^2 \rangle - \langle k \rangle^2 = N$$

标准差  $\sqrt{N}$  与扩散运动相同

# 是否均匀抽样?

- 考察极限情况：当抽样次数足够多时，假设有某种“平衡”分布 $p_i$

$$p_i = p p_{i-1} + q p_{i+1}$$

- 采取试探解

$$p_i \propto \lambda^i$$

- 从而有

$$1 = \frac{p}{\lambda} + q\lambda \Rightarrow \lambda_1 = 1, \lambda_2 = \frac{p}{q}$$

- 所以 $p_i \propto 1$ 或者  $p_i \propto \left(\frac{p}{q}\right)^i$
- 若 $p=q$ 显然只有均匀解，否则应该是哪个解呢？
- 从有限 $N$ 来看应该是第二个
- 所以为了获得均匀抽样，必须 $p=q=1/2$

# 与布朗运动的关系

- 考察N足够大的时候的概率密度 ( $p=q=1/2$ 时)

$$p(k, N) = 2^{-N} p^{(N+k)/2} q^{(N-k)/2} \frac{N!}{\frac{N+k}{2}! \frac{N-k}{2}!}$$

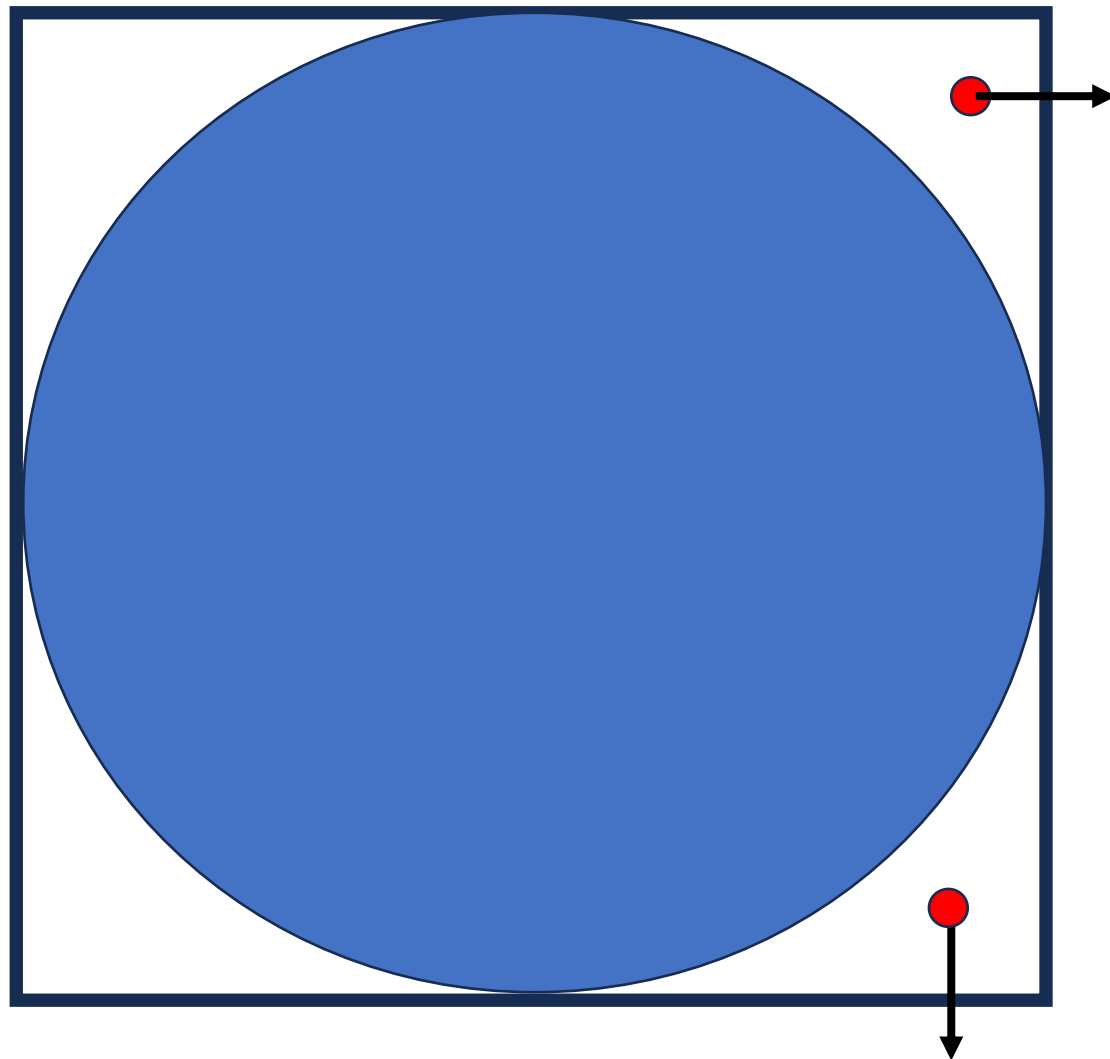
- 利用斯特林公式  $N! \approx \left(\frac{N}{e}\right)^N$

$$p(k, N) = 2^{-N} p^{(N+k)/2} q^{(N-k)/2} \frac{N^N}{\left(\frac{N+k}{2}\right)^{(N+k)/2} \left(\frac{N-k}{2}\right)^{(N-k)/2}}$$

$$\begin{aligned} \ln p(k) &= -N \ln 2 + \frac{N+k}{2} \ln p + \frac{N-k}{2} \ln q + \frac{N+k}{2} \ln \frac{2N}{N+k} + \frac{N-k}{2} \ln \frac{2N}{N-k} \\ &= \frac{N+k}{2} \ln p + \frac{N-k}{2} \ln q - \frac{N+k}{2} \ln \left(1 + \frac{k}{N}\right) - \frac{N-k}{2} \ln \left(1 - \frac{k}{N}\right) \\ &= \frac{N+k}{2} \ln p + \frac{N-k}{2} \ln q - \frac{k^2}{2N} \approx -\frac{k^2}{2N} \end{aligned}$$

- $p(k) \propto e^{-\frac{k^2}{2N}}$  高斯分布。此即为布朗运动

# 边界处理：回顾

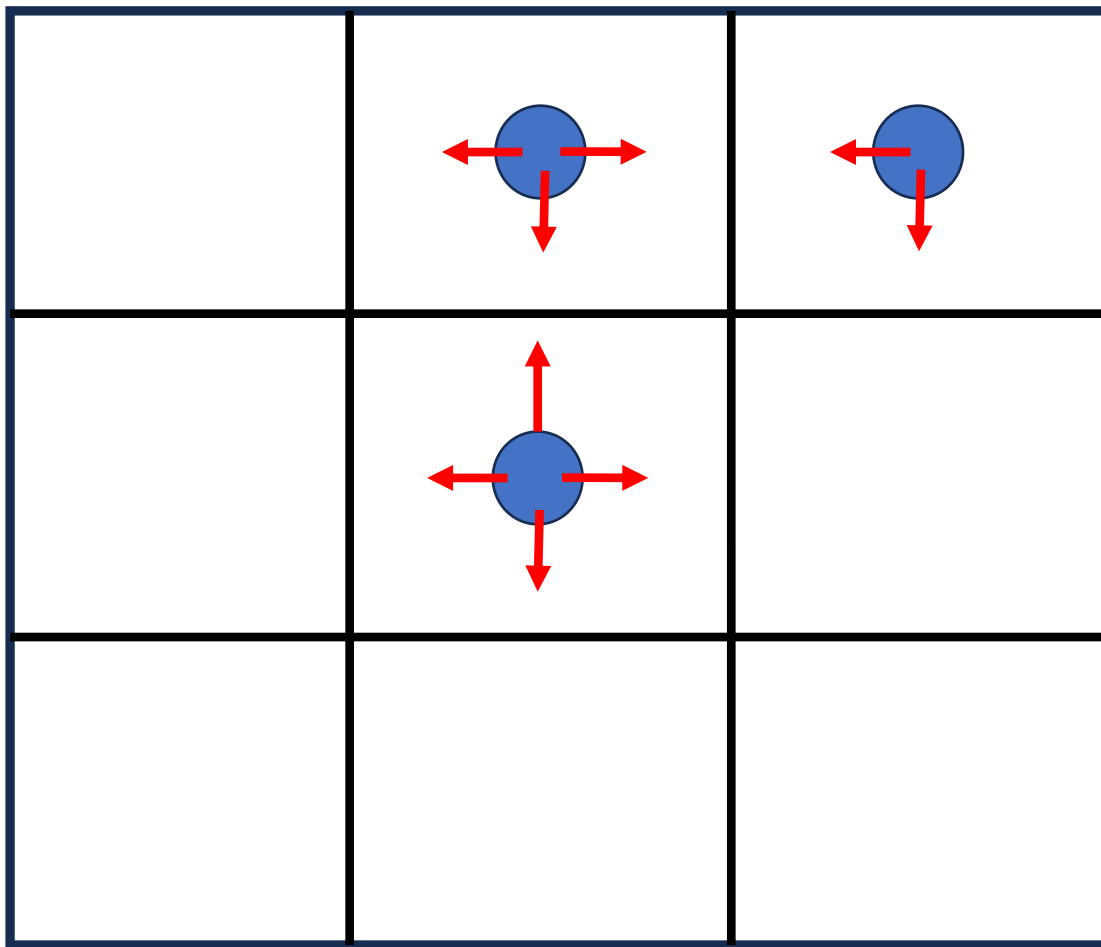


策略3：此投点仍有效，仍记入总投点数（堆石法），但位置不变继续投，直至位置可以变化。

这是Metropolis算法，其本质是细致平衡条件。



# 边界处理：简化



如何选定游走策略，使得当行走步数足够多时，粒子在每个方格的概率相同？也即如何获得一个对离散型均匀分布的抽样？

# 边界处理：细致平衡1

- 对每个方格进行编号，1-9. 我们希望获得在很多次游走之后的稳定分布（平衡分布） $\pi(a)$
- 我们希望定出的是当粒子在一个方格 $a$ 时，其下一步可到的格点（假设相邻）的概率 $p(a \rightarrow b)$
- 简化起见，我们先考虑位于边上的方格 $a$ ，其相邻有三个格点 $b, c, d$ .
- 显然有归一化方程
$$1 = p(a \rightarrow a) + p(a \rightarrow b) + p(a \rightarrow c) + p(a \rightarrow d)$$
- 以及转移方程
$$\pi(a) = \pi(a)p(a \rightarrow a) + \pi(b)p(b \rightarrow a) + \pi(c)p(c \rightarrow a) + \pi(d)p(d \rightarrow a)$$
- 结合二者，我们可获得
$$\pi(a)p(a \rightarrow b) + \pi(a)p(a \rightarrow c) + \pi(a)p(a \rightarrow d) = \pi(b)p(b \rightarrow a) + \pi(c)p(c \rightarrow a) + \pi(d)p(d \rightarrow a)$$
- 细致平衡条件显然可提供一个解
$$\pi(a)p(a \rightarrow b) = \pi(b)p(b \rightarrow a) \quad \pi(a)p(a \rightarrow c) = \pi(c)p(c \rightarrow a) \quad \pi(a)p(a \rightarrow d) = \pi(d)p(d \rightarrow a)$$

## 边界处理：细致平衡2

- 若我们希望均匀分布，则 $p(a \rightarrow b) = p(b \rightarrow a)$ ，即去和回的概率一样
- 这在一维也被验证过
- 上述方法也可处理角落的方格，所得结果相同
- 此结论可给出堆石法的原因  
中心处有4个方向， $\frac{\text{每个概率为1}}{4}$ ，而边处要满足细致平衡，故需 $p(a \rightarrow a) = 1/4$ ，也即在边上的方格有1/4的概率留在原地；同理，角落处的方格有1/2的概率留在原地。这样获得的平衡分布才是均匀分布。

# 边界处理：转移矩阵

- 对方格进行顺序的1-9标号，之后可将所有的过程组成一个矩阵

$$\{p(a \rightarrow b)\} = \begin{bmatrix} p(1 \rightarrow 1) & \cdots & p(9 \rightarrow 1) \\ \vdots & \ddots & \vdots \\ p(1 \rightarrow 9) & \cdots & p(9 \rightarrow 9) \end{bmatrix}$$

$$\{p(a \rightarrow b)\} = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} \end{pmatrix}$$

本征值

-0.5000	0	0	0	0	0	0	0	0
0	-0.0000	0	0	0	0	0	0	0
0	0	-0.0000	0	0	0	0	0	0
0	0	0	0.2500	0	0	0	0	0
0	0	0	0	0.2500	0	0	0	0
0	0	0	0	0	0.5000	0	0	0
0	0	0	0	0	0	0.7500	0	0
0	0	0	0	0	0	0	0.7500	0
0	0	0	0	0	0	0	0	1.0000
0.1667	0.1164	-0.3913	0.1171	-0.3121	0.5000	-0.5236	-0.2432	-0.3333
-0.3333	0.2750	0.5077	0.4096	0.3316	0.0000	-0.1402	-0.3834	-0.3333
0.1667	-0.3913	-0.1164	0.1171	-0.3121	-0.5000	0.2432	-0.5236	-0.3333
-0.3333	-0.5077	0.2750	-0.5267	-0.0195	0.0000	-0.3834	0.1402	-0.3333
0.6667	0.0000	0.0000	-0.2341	0.6242	0.0000	-0.0000	0.0000	-0.3333
-0.3333	0.5077	-0.2750	-0.5267	-0.0195	-0.0000	0.3834	-0.1402	-0.3333
0.1667	0.3913	0.1164	0.1171	-0.3121	-0.5000	-0.2432	0.5236	-0.3333
-0.3333	-0.2750	-0.5077	0.4096	0.3316	-0.0000	0.1402	0.3834	-0.3333
0.1667	-0.1164	0.3913	0.1171	-0.3121	0.5000	0.5236	0.2432	-0.3333

本征矢量

## 边界处理：转移矩阵2

- 本征方程

$$P \psi_i = \lambda_i \psi_i$$

- 对于任意一个初始状态 $\psi_{org}$ ,它可按照本征矢进行分解

$$\psi_{org} = \sum_i a_i \psi_i$$

- 从而，我们可以将转移矩阵多次作用其上来获得演化过程

$$P \psi_{org} = P \sum_i a_i \psi_i = \sum_i a_i \lambda_i \psi_i$$

$$P^n \psi_{org} = P^n \sum_i a_i \psi_i = \sum_i a_i \lambda_i^n \psi_i$$

- 注意到，转移矩阵的本征值最大的为1，其次为3/4，当n很大时，我们有

$$P^n \psi_{org} = \sum_i a_i \lambda_i^n \psi_i \approx a_1 \psi_1 + 0.75^n a_2 \psi_2$$

多次转移之后，分布会以指数趋于均匀分布（平衡分布）

# 引申：Metropolis算法

- 在之前计算圆周率的算法中，构型要么可取要么不可取，对于这种情况，我们已有讨论
- 随机游走抽样可推广至一般情形，也即不同构型有确定的概率分布 $\pi(a)$
- 此时，转移矩阵需满足

$$p(a \rightarrow b) = \min\left(1, \frac{\pi(b)}{\pi(a)}\right)$$

- 证明：

情形	$\pi(a) > \pi(b)$	$\pi(b) > \pi(a)$
$p(a \rightarrow b)$	$\pi(b)/\pi(a)$	1
$\pi(a)p(a \rightarrow b)$	$\pi(b)$	$\pi(a)$
$p(b \rightarrow a)$	1	$\pi(a)/\pi(b)$
$\pi(b)p(b \rightarrow a)$	$\pi(b)$	$\pi(a)$

# 算法步骤

- 1. 选取试探位置,  $x_t = x_n + \eta_n$ , 其中 $\eta_n$ 可为在 $(-\delta, \delta)$ 区间内的随机数。
- 2. 计算 $r = \pi(x_t)/\pi(x_n)$
- 3. 若 $r \geq 1$ , 则应接受此改变, 也即 $x_{n+1} = x_t$ .
- 4. 否则, 产生一个  $(0,1)$  内的随机数 $\xi$ , 若 $\xi < r$ , 则亦接受此改变, 也即 $x_{n+1} = x_t$ 。否则,  $x_{n+1} = x_n$ 。
- 5. 从新的位置出发走下一步, 直到达到预定的总步数。

# 讨论

- 对于简单的正方形区域，做周期性边条件是可以的。但此方法对于一般区域很难扩展；对于平衡分布不均匀的情况，即使在正方形区域，也应按照Metropolis算法做推广。
- 对于原问题，在点到边界外的时候 $\frac{\pi(b)}{\pi(a)} = 0$ , 所以此点需抛弃。在边界内的时候 $\frac{\pi(b)}{\pi(a)} = 1$ （均匀分布），故肯定接受。
- 特别注意：细致平衡仅仅是充分条件，不是必要的！

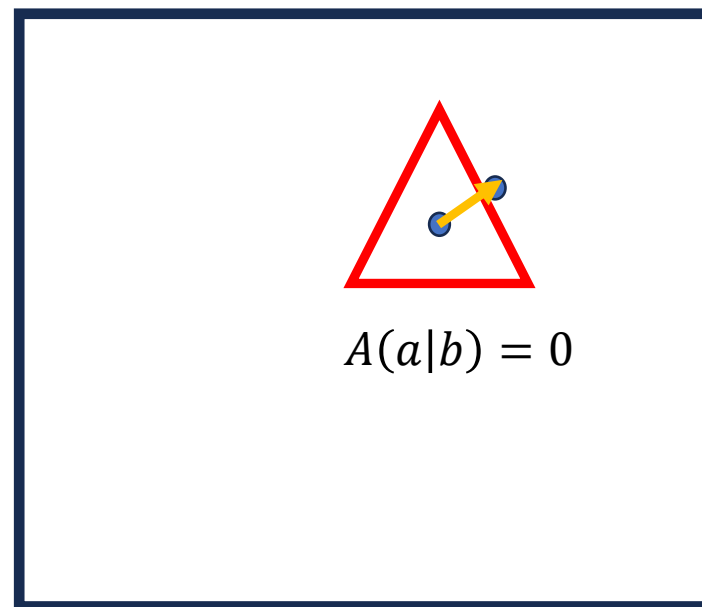
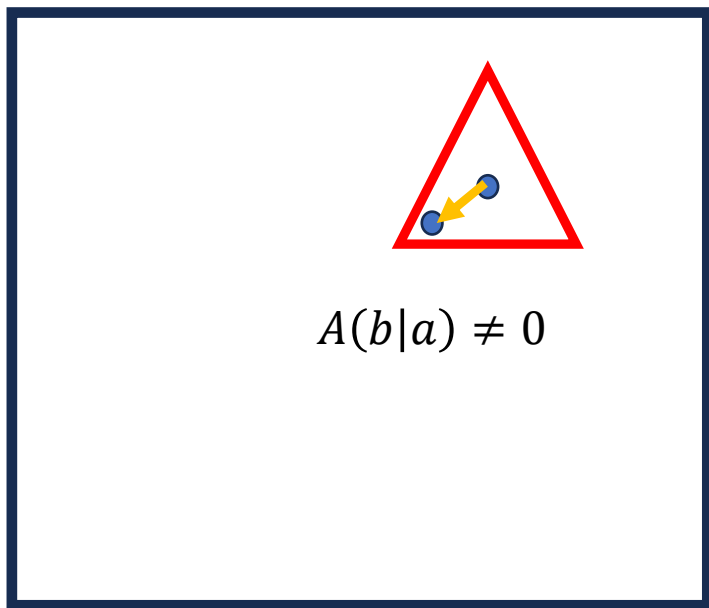


# 先验概率

- 在前面的例子中，从某个点移至下个点时，其移动有范围要求，或者更严谨的说，当粒子位于某个点 $x_0$ 时，其之后的选点满足某个概率分布 $A(x|x_0)$ 。这个概率分布是提前给出而不是后续推导获得的，也即先验概率。
- 先验概率在随机游走抽样中普遍存在。
- 存在先验概率时，Metropolis算法需基于条件概率做进一步修改 $P(a \rightarrow b) = A(b|a) p(a \rightarrow b)$ （转移 = 选择 \* 接受）
- 细致平衡： $\pi(a)P(a \rightarrow b) = \pi(b)P(b \rightarrow a)$
- Metropolis条件：

$$p(a \rightarrow b) = \min \left[ 1, \frac{\pi(b)}{\pi(a)} * \frac{A(a|b)}{A(b|a)} \right]$$

# 先验概率示例：三角形算法



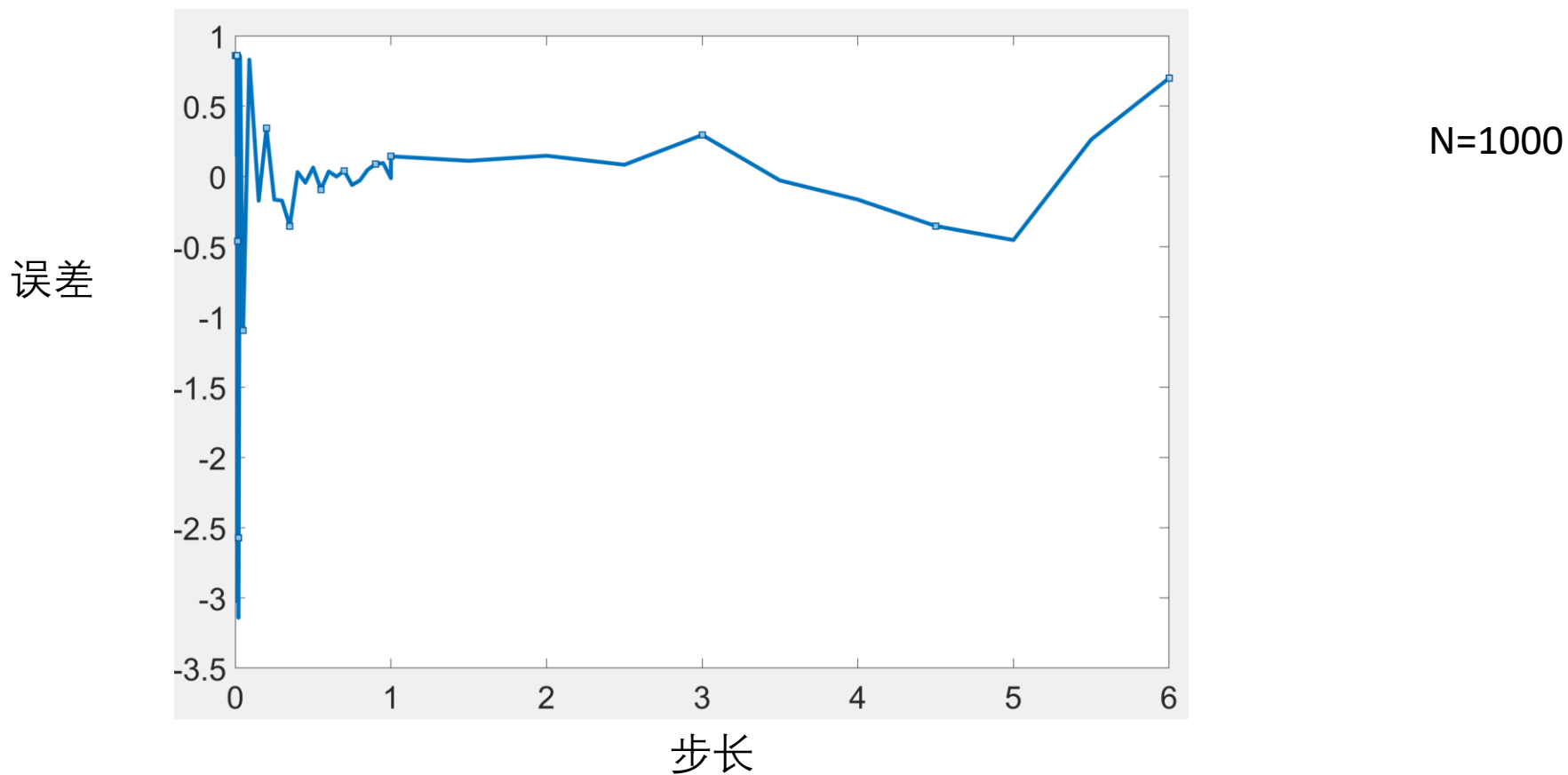
- $p(a \rightarrow b) = \min \left[ 1, \frac{\pi(b)}{\pi(a)} * \frac{A(a|b)}{A(b|a)} \right] = 0$  此位移不可接受!

# 特殊情况

- $p(a \rightarrow b) = \min \left[ 1, \frac{\pi(b)}{\pi(a)} * \frac{A(a|b)}{A(b|a)} \right]$
- 若  $A(a|b) = \pi(a)$ ,  $A(b|a) = \pi(b)$  也即粒子到某个位点的先验概率与当前位点完全无关, 则有  
 $p(a \rightarrow b) = 1$  永远成立
- 此时随机游走抽样完全变为直接抽样 (也即在正方形内随机取点)
- 先验概率最有用的场景: 我们大致可以做直接抽样, 或对于某个子系统大致可以做。此时先验概率的作用大致和微扰处理相似。

# 步长选择

- 步长不可太大，否则投点很容易出界，没有有效的位移点。
- 步长不可太小，否则投点会局域在初始点附近，无法在整个区域均匀分布。
- 步长应保证，大致1/2的位移被接受。



# 应用1：变分蒙特卡罗算法

- 一般的哈密顿量本征值问题

$$\hat{H} \psi = E \psi$$

- 对于基态能量，我们有

$$\langle E \rangle = \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle} \geq E_0$$

- 在多数物理问题中，确定基态起到关键作用。而对于多体问题，由于构型数的指数增长，基态的求解较为困难。
- 根据上式，可采用变分蒙卡，将基态求解化为求参量空间的某个最小值，从而我们从指数型的构型空间退化到线性增长的构型空间。

# 算法描述

- 此算法的核心是计算能量平均值，此处可用随机游走的方法获得。

$$\langle E \rangle = \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle} \geq E_0$$

- 重写能量均值

$$\langle E \rangle = \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle} = \int dx dy dz \rho_\psi E_\psi$$

$$\rho_\psi = \frac{|\psi|^2}{\int dx dy dz |\psi|^2}, \quad E_\psi = \frac{\hat{H}\psi}{\psi}$$

- $\rho_\psi$  可看做带抽样的概率分布。
- 变分法的核心：猜想一个波函数的可能形式  $\psi(\alpha)$ ，其中  $\alpha$  代表一系列可变量。我们则应该在对波函数施加此限制之下，改变  $\alpha$ ，找到对应于最小平均能量的参量值。其对应的波函数则是我们对精确基态波函数的近似。

# 蒙卡算法步骤

- 产生一个初始位置 $x_0$ ，计算 $E_{\psi(x_0)}$
- 在 $(-\delta, \delta)$ 内产生一个随机数 $\eta$ （注意，此步骤是一维情形，在高维要把取值区间相应扩展），因当前位置为 $x_n$ ，下一个位置的试探值为 $x_t = x_n + \eta$
- （Metropolis算法）计算 $\lambda = \frac{|\psi(x_t)|^2}{|\psi(x_n)|^2}$ ，如果 $\lambda > 1$ ，则接受位置改变 $x_{n+1} = x_t$ ；否则，产生一个 $(0,1)$ 内的随机数 $\xi$ ，若 $\xi < \lambda$ ，则接受改变， $x_{n+1} = x_t$ ；再否则， $x_{n+1} = x_n$ 。计算 $E_{\psi(x_{n+1})}$ 。
- 从新位置出发继续游走，直到达到预定的步数。
- 求平均 $\langle E \rangle = 1/N \sum_i E_{\psi(x_i)}$ ，我们应该在参数空间最小化次平均值。
- 注意：上述求平均的过程从原理上只要产生一个初始位置即可。但实际操作时，有可能碰到一个位置周围被多个峰包围从而走不出去的情况，此时可产生多个初始位置分别独立行走来计算均值。

# 总算法步骤

- 选择初始参量值，从而确定初始试探波函数 $\psi(\alpha_0)$ 。
- 利用上述随机游走方法计算此试探波函数下的能量均值 $\langle E_0 \rangle$ 。
- 在参数空间变化一个值，也即将 $\alpha_0$  变为 $\alpha_1$ ，获得新的能量均值 $\langle E_1 \rangle$ ，若 $\langle E_1 \rangle < \langle E_0 \rangle$ ，则接受变化，以新的参量值作为起始替换 $\alpha_0$ 。重复此步，直到能量均值的改变小于某个给定值。
- 注意：参量空间的变化可采用随机游走。此时，应用随机游走产生均匀分布。



## 应用2：统计力学

- 目标：计算平衡态时某物理量的测量值（期望值）；
- 方法：1.选择系综。2.根据系综写出分布函数 $\rho(\hat{H})$ .3.计算平均值

$$\langle A \rangle = \text{Tr}(\hat{A}\rho(\hat{H}))$$

- 注：在明确本征态的情况下，分布函数能直接解析写出，此时只需计算对所有态的求和或积分即可；

在有相互作用的复杂系统，本征态不知，我们只能从每个状态的概率 $f(H)$ 出发，计算配分函数： $Z = \sum f(H)$ 。从而 $\rho = f/Z$

- 由于是求期望，自然可用概率算法。
- 关键：产生满足 $\rho(\hat{H})$ 的状态构型抽样。
- 方法：随机游走+Metropolis算法

# 算法思路 (1)

- 1.选择初始状态 $x_0$ .(注: 按照教材所说, 此状态最好在分布密度较大的区域。若担心初始状态会限于一个区域, 也可产生一个初始状态的合集, 对里面的值分别独立进行游走)
- 2.若游走至第 $n$ 步, 需到第 $n+1$ 步。则首先用随机方法产生一个试探状态 $x_t = x_n + \eta_n$ , 其中 $\eta_n \in [-\delta, \delta]$ 。(注: 这里假设了连续状态, 给出了一个特例。但连续性不是必须的, 只需把握两点: (1) 下个状态在某个范围内取值; (2) 进入这个范围内的状态需要一个预先给定的已知概率, 也即先验概率。)
- 3.计算过渡概率 $w(x_n, x_t)$ , 注意, 可以不是metropolis算法。在Metropolis算法下, 我们有
$$w(x_n, x_t) = \min \left[ 1, \frac{f(H(x_t))}{f(H(x_n))} \right]$$
- 4.产生 $[0,1]$ 内的随机数 $r$

# 算法思路 (1)

- 5. 若 $r \leq w$ , 则接受改变, 也即 $x_{n+1} = x_t$ ; 否则,  $x_{n+1} = x_n$ ;
- 6. 在第 $n+1$ 个构型上计算估计值
$$A_{n+1} = A(x_{n+1})$$
- 7. 回到第2步往下执行, 直到重复 $N$ 次, 共获得 $N$ 个对 $A$ 的估计值。
- 8. 将所有估计值相加除以 $N$ , 此即为 $A$ 的期望值。

# 例子：伊辛模型

- 模型的哈密顿量：

$$H = -\frac{J}{2} \sum_{\langle ij \rangle} S_i S_j - B \sum_i S_i$$

- 每个点的自旋只有两个选择，+1或-1，代表向上或向下。第一项为各向同性交换场，第二项为外磁场引入的塞曼能。 $\langle ij \rangle$ 代表最近邻的两个位点。

- 配分函数：

$$Z = \sum_S e^{-\beta H}$$

- 磁化强度

$$M = \frac{1}{\beta} \frac{\partial \ln Z}{\partial B} = \frac{1}{Z} \sum_S M(S) e^{-\beta H}, \quad M(S) = \sum_i S_i$$

- 通常，我们会关注M，也即 $M(S)$ 的期望值，来看是否有自发磁化（相变），或者关注 $M(S)$ 的涨落，根据涨落-耗散定理，这和磁化率成正比，其发散行为也预示相变。

# 算法思路

- 选择初始构型,  $S_0 = \{s_1, s_2, \dots, s_n\}$ .
- 有了第 $m$ 个位型 $S_m$ 后, 需游走获得第 $m+1$ 个位型。方法: 产生一个1到 $n$ 之间的整数随机数 $i$ , 将 $s_i$ 翻转, 也即 $s_i = -s_i$ 。记此构型为试探构型 $S_t$ 。
- 根据哈密顿量计算能量差 $\Delta E = E(S_t) - E(S_m)$ 。
- 若 $\Delta E \leq 0$ , 则 $S_{m+1} = S_t$ 。
- 否则, 产生一个 $[0,1]$ 内的随机数 $\xi$ , 若 $\xi \leq e^{-\beta\Delta E}$  则 $S_{m+1} = S_t$ ; 否则 $S_{m+1} = S_m$ 。
- 在每一步监控 $\Delta E$ 的值, 若在若干步内 $\Delta E$ 的均值小于某个预设值, 则我们可说系统已到平衡态。
- 在达到平衡态之后, 继续游走 $L$ 步。在每一步计算 $M(S_i)$ . 最后获得
$$\langle M \rangle = \frac{1}{L} \sum_{i \in Equal} M(S_i)$$
- 由于位型增长过快, 体系可能整体大小不大。为消除边界效应, 可采用周期边条件。

## 应用3： 偏微分方程求解

- 目标： 在区域D中求解如下泊松方程

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = q(x, y), \quad \phi|_{\partial D} = F(s), s \in \partial D$$

- 首先进行离散化处理,考虑方形区域, 并且将此区域进行横纵等距分割, 相邻格点距离为h, 则微分可用差分代替(每个格点有四个近邻)。

$$\nabla^2 \phi|_R = \frac{1}{h^2} (\phi_1 + \phi_2 + \phi_3 + \phi_4 - 4\phi_R) = q(R)$$

$$\phi_R = \frac{1}{4} (\phi_1 + \phi_2 + \phi_3 + \phi_4 - h^2 q(R))$$

- 我们的目标是计算格点上的函数值。进一步假设边界正好也对应到分割线。若总共有N个格点, 则我们可将所有格点的函数值组成一个N维的列向量, 则泊松方程加上边界条件可写为如下矩阵形式

$$\phi = P\phi + A$$

# 形式解析

- 离散化方程  $\phi = P\phi + A$
- 1. 对于矩阵  $P$ : 考虑  $p_{ij}$  其中  $i$  不在边界上, 则  $p_{ij} = 1/4$  或  $0$ : 前者对应于  $j$  是  $i$  的相邻点; 后者对应其它情形。
- 2. 对于矩阵  $P$ : 考虑  $p_{ij}$  其中  $i$  在边界上, 则  $p_{ij} = 0$ , 也即边界上是简单赋值。
- 3. 对于向量  $A$ : 考虑  $A_i$  其中  $i$  不在边界上, 则  $A_i = -\frac{h^2}{4} q(R_i)$
- 4. 对于向量  $A$ : 考虑  $A_i$  其中  $i$  在边界上, 则  $A_i = F(R_i)$
- 5. 上式的待定函数有如下形式解:

$$\phi = (I - P)^{-1}A = IA + PA + P^2A + \dots$$

# 与随机游走的关联

- 考虑如下形式解：

$$\phi = (I - P)^{-1}A = IA + PA + P^2A + \dots$$

- 抽取某个项

$$\phi_i = P_{ij}P_{jj_1}P_{j_1j_2} \cdots P_{j_{n-1}j_n}A_{j_n}$$

它实际上代表某个从*i*到*j<sub>n</sub>*的路径而 $P_{ij}P_{jj_1}P_{j_1j_2} \cdots P_{j_{n-1}j_n}$ 则“大约”是此路径的出现概率。

原因：对于P矩阵，其每行元素的和必是1(除非在边界)：这是某种转移矩阵！

性质：若*i*和*j<sub>n</sub>*都不在边界点上，则中间的路径点不能包含边界点。从P的性质的直接观察证明。

- 通过随机游走可抽样出满足此概率的路径集合。方法：定义第*a*个格点的停止概率为 $P_a = 1 - \sum_b P_{ab}$ 。显然 $P_a = 0$ 或1,前者对应*a*在区域内后者对应于*a*在边界。将*P<sub>a</sub>*纳入转移矩阵。从*i*点出发，根据转移矩阵决定的概率进行游走，若下一步的位置和当前位置相同，则停止；此时正好获得一条路径。注意到根据此方法，停止必发生于边界点。



## 与随机游走的关联 (2)

- 考虑所有这种路径的集合，并计算如下表达式

$$\langle \phi_R \rangle = \sum_{\Gamma} A(\Gamma) = \sum_{\Gamma} \sum_{r_i \in \Gamma} A(r_i)$$

- 交换求和次序

$$\langle \phi_R \rangle = \sum_{\Gamma} \sum_{r_i \in \Gamma} A(r_i) = \sum_{r_i} \sum_{\Gamma, s.t. r_i \in \Gamma} A(r_i)$$

- 所有的路径都以R为起点，故第二个求和显然具有之前的形式

$$\sum_{\Gamma, s.t. r_i \in \Gamma} A(r_i) = [(I - P)^{-1} A]_{Rr_i}$$

- 所以，在游走路径足够多时，我们可说

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

# 算法思路

- 1. 网格分划，离散化偏导数，如之前所示。网格分划应包含边界为其中的网格线。初始化游走路径数目 $Nw$ 。
  - 2. 将区域内部的网格点编号，记编号为 $i$ ，其范围为1到 $N$ ， $N$ 为总内部格点数。
  - 3. 考察第 $i$ 个格点，将其作为出发点。从 $i$ 格点开始随机游走，每次有 $1/4$ 的概率走至相邻的格点中的一个（可用塔式抽样决定是哪个格点），直到边界处停止，完成一个路径。在每个路径点处计算 $A$ 的值，其值在内部为 $-\frac{h^2}{4}q(r_i)$ ，在边界为 $F(r_i)$ 。将所有值加和构成对 $\phi_R$ 的一个无偏估计。
  - 4. 重复游走过程，直到获得 $Nw$ 条游走路径，将所有无偏估计加和并除以 $Nw$ ，此即为 $\phi_i$ ，也即待定函数在第 $i$ 个格点的值。
  - 5. 重复第3,4步，直到遍历所有格点。
- 注：由于转移矩阵在内部不停留，此法的收敛性较差，是边界收敛的情况。好的收敛需要在每个格点都有一定概率停留（每个行的和小于1）。

## 应用4：格林函数蒙卡

- 目标：考察扩散方程

$$\frac{\partial \rho}{\partial t} = \alpha \frac{\partial^2 \rho}{\partial x^2}$$

- 上述方程的格林函数为

$$G_0(x, y; t) = \frac{1}{\sqrt{4\pi\alpha t}} e^{-\frac{(x-y)^2}{4\alpha t}}$$

- 当y固定时，G是扩散方程的解。当t趋于0时，G趋于delta函数 $\delta(x - y)$
- 此格林函数可用于计算初态给定时之后任意时刻的函数值

$$\rho(y, t) = \int dx G_0(y, x; t) \rho(x, 0), \quad \int G_0(y, x; t) dx = 1$$

- 所以，为求得函数值，我们只需抽样满足G0分布的位置值。

# 格林函数抽样

- 此格林函数可看做初始值在 $x$ 时的单步游走的概率分布，也即转移概率为

$$T_{\Delta t}(x \rightarrow y) = G_0(y, x; \Delta t)$$

- 故此格林函数也等价于一种随机游走过程，也即郎之万过程，其游走方程由下式给出

$$x(t + \Delta t) = x(t) + \eta \sqrt{\Delta t}$$

其中， $\eta$ 满足高斯分布： $f(\eta) = \frac{1}{\sqrt{4\pi\alpha}} e^{-\frac{\eta^2}{4\alpha}}$

- 证明：在初始为 $x$ ，下一时刻在 $y$ 到 $y+dy$ 的概率为

$$p(y)dy = f\left(\frac{y-x}{\sqrt{\Delta t}}\right)d\eta = f\left(\frac{y-x}{\sqrt{\Delta t}}\right)\frac{1}{\sqrt{\Delta t}}dy = G_0(y, x; \Delta t)dy$$

# 离散对应

- 注意上述郎之万过程是连续的，只有时间尺度离散。
- 我们也可考虑实空间的离散化对应：考虑一个一维的晶格点阵，相邻位点距离相同，为 $a$ 。时间离散间隔为 $h$ 。在每个晶格点上，粒子左移一格和右移一格的概率均为 $\beta$ ，则粒子留在原位的概率为 $1 - 2\beta$ 。
- 我们关心粒子在时刻 $t$ 位于第 $i$ 个格点的概率 $\rho(x_i, t)$ ，则
$$\rho(x_i, t) = \beta\rho(x_i + a, t - h) + \beta\rho(x_i - a, t - h) + (1 - 2\beta)\rho(x_i, t - h)$$
- 做等价变形

$$\rho(x_i, t) - \rho(x_i, t - h) = \beta(\rho(x_i + a, t - h) + \rho(x_i - a, t - h) - 2\rho(x_i, t - h))$$

$$\frac{\partial \rho}{\partial t} = \beta a^2 \frac{\partial^2 \rho}{\partial x^2}$$

确实变为了扩散方程！

连续比离散更好，因为省掉了离散趋于连续的极限过程。

# 一般情形(1)

- 一般的扩散方程

$$\frac{\partial \rho}{\partial t} = L\rho = -\left(\frac{p^2}{2} + V\right)\rho = \left(\frac{1}{2}\frac{\partial^2}{\partial x^2} - V\right)\rho$$

- 格林函数

$$G_0(y, x; t) = \langle x | e^{tL} | y \rangle$$

- 与演化子和编时算符类似
- 由于动能部分和势能部分不对易，上述格林函数很难求解。
- 做离散化处理，考虑在一个小时间间隔 $\tau$ 后的格林函数。利用Baker-Hausdorff公式，我们有

$$e^{-t(T+V) - \frac{1}{2}t^2[T, V] + O(t^3)} = e^{-tT} e^{-tV}$$

$$\text{故 } e^{tL} = e^{-tT} e^{-tV} + O(t^2)$$

## 一般情形(2)

- 动能部分可简单算出

$$G_{kin} = \langle x | e^{-\tau T} | y \rangle = \sum_{p_1 p_2} \langle x | p_1 \rangle \langle p_1 | e^{-\tau T} | p_2 \rangle \langle p_2 | y \rangle = \frac{1}{\sqrt{2\pi\tau}} e^{-\frac{(y-x)^2}{2\tau}}$$

- 总格林函数为

$$G_0 = G_{kin} e^{-\tau V(y)}$$

- 势函数部分破坏了归一性，无法与随机游走对应。

- 定义新的格林函数

$$G = G_0 e^{\tau E_T} \text{ 也即多乘一个常数以恢复归一性。}$$

- 则新格林函数满足

$$\frac{\partial G}{\partial t} = \left( \frac{1}{2} \frac{\partial^2}{\partial x^2} - V + E_T \right) G$$

- 其平衡分布对应一个本征值问题：

$$\left( -\frac{1}{2} \frac{\partial^2}{\partial x^2} + V \right) G(x) = E_T G(x)$$

# 算法执行

- 这里我们只把重点放在随机游走上，也即利用随机游走获得满足格林函数分布的位置。暂时不考虑具体要计算哪个物理量，这应根据实际问题添加。
- 首先，产生一个初始的位置集合  $X_0 = \{x_1, x_2, \dots, x_n\}$ . 我们将从每个位点出发进行独立游走。对于起始于第  $i$  个位置的游走过程，当粒子的当前位置  $x(t)$  已知时，其下一个位置按照郎之万过程，根据扩散部分（动能部分）的格林函数获得

$$x(t + \Delta t) = x(t) + \eta \sqrt{\Delta t}$$

$$\text{其中 } f(\eta) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\eta^2}{2}} \quad \left(\alpha = \frac{1}{2}\right)$$

- 对于新的位点，计算相应物理量的值，此时这个值作为对物理量的平均值的贡献需额外乘以一个由势能部分决定的权重因子

$$w = e^{-\Delta t [V(x(t+\Delta t)) - E_T]}$$

- 对每条路径进行游走，直到步长达到预设值。
- 将所有路径的所有位点的贡献加和，从而得到所需的期望值。
- 注：此方法效率较低。因为可能有不少粒子会跑到权重很小的区域（它们与其他区域从动能角度看无本质不同），而此时他们需要同样多的时间（与其他粒子相比）移出此区域。



# 消长算法

- 我们只需考虑行走至某个新格点之后的处理。原算法是在算物理量之后附加某个势能部分决定的权重因子，而消长算法则给出此消彼长的变化。
- 计算：在新格点处计算  $q = e^{-\Delta t[V(R_{new}) - E_T]}$
- 消：若  $q < 1$ ，则此粒子的随机游走有  $q$  的概率保持（也即接着进行），而有  $1 - q$  的概率直接消亡。注意：若消亡，总粒子数会减少。
- 长：若  $q > 1$ ，则此粒子有  $q - [q]$  的概率孕育  $[q]$  个新粒子，而有  $1 + [q] - q$  的概率孕育  $[q] - 1$  个新粒子。然后对于现在存在的所有粒子，从其当前位置开始游走。注意：此时，总粒子数增多。
- 计算物理量的估计值时，需要用消长之后的粒子来计算。也就是说，若消亡则当前位置无贡献，若生长，则当前位置的值需要乘以生长之后的粒子数。

# 确定 $E_T$

- 我们无法直接求解本征问题确定 $E_T$ :若本征问题可解, 扩散问题已经能解出。在长时间演化后,  $E_T$ 会趋于基态能量 (之后讲)。
- 根据消长算法, 在每一步的粒子数是可能变化的。但实际上, 不管是增多还是减少都不好。若增多, 则计算负担会增大。若减小, 则统计涨落会增大。
- 可取策略, 在每一步都调整 $E_T$  的值, 使得总粒子数不变。
- 调整 $E_T$  的的方法: 若希望总粒子数为 $M$ , 而在某一步根据消长算法变化后粒子数为 $M'$ , 则

$$E_T \rightarrow E_T + \epsilon \ln \frac{M}{M'}$$

其中 $\epsilon$ 为一个预设的小量。

若 $M' > M$ ,  $E_T$  decrease,  $q$  decrease,    *annihilate more!*

若 $M' < M$ ,  $E_T$  increase,  $q$  increase,    *annihilate less or breed more!*

# 归一化原理

- 数学核心：平衡分布可分解成两个部分，即 $f(x) = g(x)h(x)$ ； $g(x) = G_{kin}$ 也即由动能决定的扩散部分的分布，它已经归一化了；而 $h(x) = e^{-\Delta t[V(x)-E_T]}$ 由势能决定
- 若将消长过程统一，算法是每次有 $h(x) - [h(x)]$ 的概率孕育 $[h(x)]$ 个新粒子，而有 $1 + [h(x)] - h(x)$ 的概率孕育 $[h(x)] - 1$ 个新粒子。

- 计算粒子总数平均值

$$\begin{aligned}\langle N \rangle &= N + \int dx \, g(x) (h(x) - [h(x)]) [h(x)] \\ &\quad + \int dx \, g(x) (1 + [h(x)] - h(x)) ([h(x)] - 1) \\ &= N + \int dx \, g(x) [h(x)] - \int dx \, g(x) (1 + [h(x)] - h(x)) \\ &= N + \int dx \, g(x) (h(x) - 1) \\ &= N\end{aligned}$$

扩散概率      生长概率      生长数

- 最后一个等号成立需要：（1） $f(x)$ 归一化，也即 $E_T$ 达到正确值；（2） $g(x)$ 归一化，这个已经被保证了。

# 算法原理

- 数学核心：某个量的平均值

$$\langle O \rangle = \int dx f(x) O(x)$$

- 若将消长过程统一，算法是每次有  $h(x) - [h(x)]$  的概率孕育  $[h(x)]$  个新粒子，而有  $1 + [h(x)] - h(x)$  的概率孕育  $[h(x)] - 1$  个新粒子。

- 计算平均值

$$\begin{aligned} \langle O \rangle &= \int dx g(x) (h(x) - [h(x)]) ([h(x)] + 1) O(x) \\ &\quad + \int dx g(x) (1 + [h(x)] - h(x)) [h(x)] O(x) \\ &= \int dx g(x) [h(x)] O(x) + \int dx g(x) (-[h(x)] + h(x)) O(x) \\ &= \int dx g(x) h(x) O(x) \end{aligned}$$

- 确实和所需的期望值相同。

# 本征值问题

- 演化算子投影：

$$e^{-t(H-E_T)} = \sum_n e^{-t(E_n-E_T)} |\phi_n\rangle\langle\phi_n|$$

- 格林函数为

$$G = \langle x | e^{-t(H-E_T)} | y \rangle = \sum_n e^{-t(E_n-E_T)} \langle x | \phi_n \rangle \langle \phi_n | y \rangle$$

- 当演化时间足够长时， $e^{-t(E_n-E_T)}$ 中最大的对应于基态能量，其他的相比于此会指数衰减。
- 故：长时间演化下的格林函数=基态投影
- 由此，可设计算法获得基态能量和波函数。

# 推广

- Feynmann-Kac公式：考虑如下偏微分方程

$$\frac{\partial u}{\partial t} + \mu \frac{\partial u}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 u}{\partial x^2} - Vu + f = 0$$

- 其中， $\mu$ ， $\sigma$ ， $V$ ， $f$  这四个系数都与位置和时间相关。

$x$ 为实数， $t \in [0, T]$ , 边条件为 $u(x, T) = \psi(x)$

- 则 $u$ 可用如下方式获得

$$u(x, t) = E^Q \left[ e^{-\int_t^T V(X_\tau, \tau) d\tau} \psi(X_T) + \int_t^T e^{-\int_t^s V(X_s, s) ds} f(X_\tau, \tau) d\tau \mid_{X_t=x} \right]$$

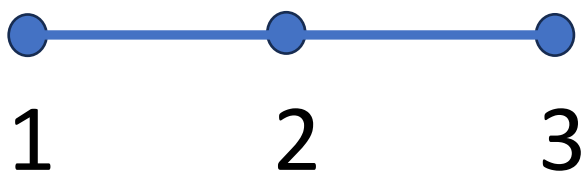
- 其中， $X$ 的值由下述随机过程决定

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t$$

而 $W_t$ 代表布朗运动。

# 作业

## 1.三点问题:



假设我们需使得粒子在这三个点上的概率分别为 0.2, 0.5, 0.3. 请设计随机游走过程加以实现, 其中, 每个粒子最多只能运动到和它当前位置直接相连的格点。(1) 请给出算法步骤; (2) 编写相应程序; (3) 用程序游走  $N$  步 (设  $N=1000$ ), 统计不同格点出现的频数 (画出频数直方图即可)。

# 作业

1. 教材第二章17题，给出具体的算法步骤（只需算法步骤，不需写代码）。