算法讲解--Monte Carlo

时亚洲

2020.9.10

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots, \quad x \in (-\infty, +\infty)$$

武漢紡織大學 WUHAN TEXTILE UNIVERSITY

蒙特卡洛(Monte Carlo)

长3.5km

1.9平方公里

4万人口

布丰(Buffon)投针

法国博物学家、作家
《自然史》

1777年，Buffon投针计算π

步骤：

1. 白纸一张，画间距为d的平行线；
2. 取长度为l ($l<d$)的针,随机投掷；
3. 计算针与直线相交的概率P；
4. 圆周率$\pi=2l/Pd$;

[0,π] 按Δθ等分

针与线夹角为θ的概率

$p_1 = \frac{\Delta\theta}{\pi}$ ①

针与线相交的概率(夹角为θ)

$p_2 = \frac{l'}{d} = \frac{l\sin\theta}{d}$

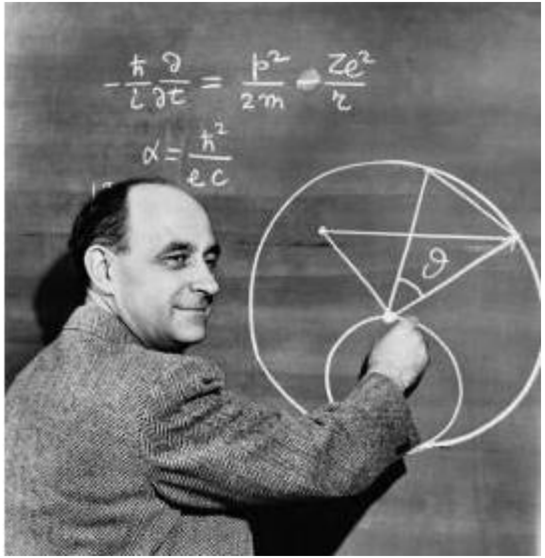
针与线以任意夹角相交的概率

$$P = \sum_{\theta=0}^{\pi} P_1 \cdot P_2 = \int_0^{\pi} \frac{l}{d\pi}\sin\theta \, d\theta = \frac{2l}{d\pi} \Rightarrow \pi = \frac{2l}{P\cdot d}$$

蕴含思想：模拟均匀分布，通过计算概率得到相关量的值
这就是*Monte Carlo*思想！！！

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots, \quad x \in (-\infty, +\infty)$$

费米&费米阿克(Fermiac)



美籍意大利人
物理学家
1938获诺贝尔物理奖
原子能之父
李政道的老师
杨振宁的合作者

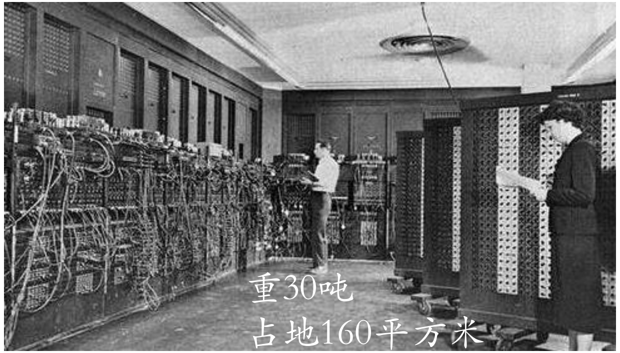
1930年左右，模拟中子在核装置中随机扩算运动

Fermiac机器在二维平面移动，随机选取碰到的中子，测量其运动方向和速度，进而得到中子的运动情况。

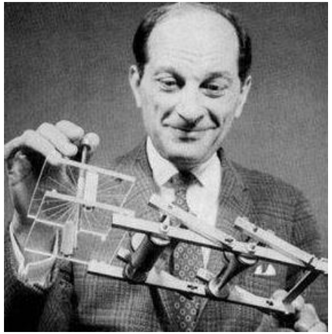
--蒙特卡洛的经典雏形！

$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots,\quad x\in(-\infty,+\infty)$$

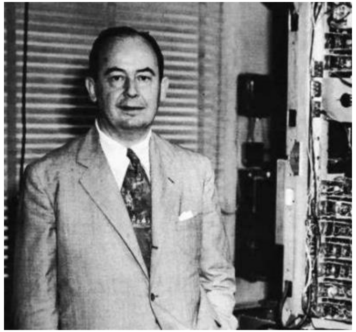
Monte Carlo诞生



1945年，第一台计算机(ENIAC)诞生
Electronic Numerical Integrator & Calculator



S. M. 乌拉姆
美国数学家



冯·诺依曼
数学家&计算机科学家

梅特罗波利斯
物理学家&计算机科学家

曼哈顿原子弹计划

←求解高维玻尔兹曼方程(高维偏微积分方程)←计算中子在原子弹内扩散和增殖问题

提出随机抽样的逆变换算法　　**给出模拟的具体技术方案**　　**编程计算实施方案**

→ **转变成相等形式的随机问题，并通过计算机模拟进行求解**

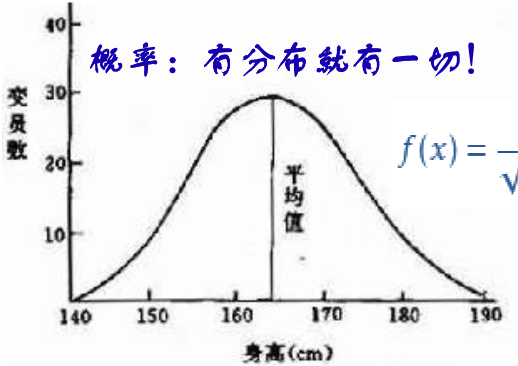
Monte Carlo

乌拉姆曾提及有亲戚在蒙特卡洛，梅特罗波利斯提议不妨用这个富有随机(赌城)、浪漫、美丽的城市命名该方法

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots, \quad x \in (-\infty, +\infty)$$

MC实质：随机抽样

- 为什么要抽样？



$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

现实抽样&统计推断

$$E(g(X))=\int_{-\infty}^{+\infty} f(x)g(x)\,dx$$

这积分一定可求解吗？

求解定积分

不妨抽取样本X，然后计算g(X)，进而算术平均

$$E(g(X))=\frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

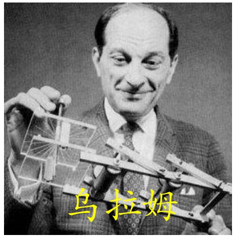
如何抽取服从特定分布的样本呢？

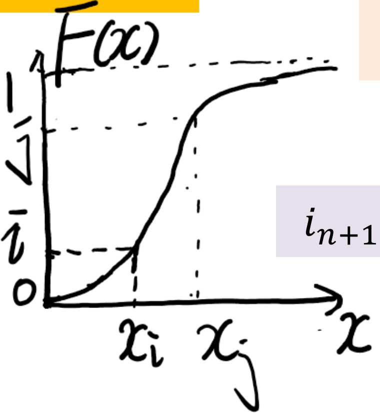
- 随机抽样

累积分布函数

$$F(x)=\int_{-\infty}^{x} f(t)\,dt$$

这个积分就一定可求解吗？





取$i \sim U(0,1)$ $\Longrightarrow$ $x_i = F^{-1}(i)$

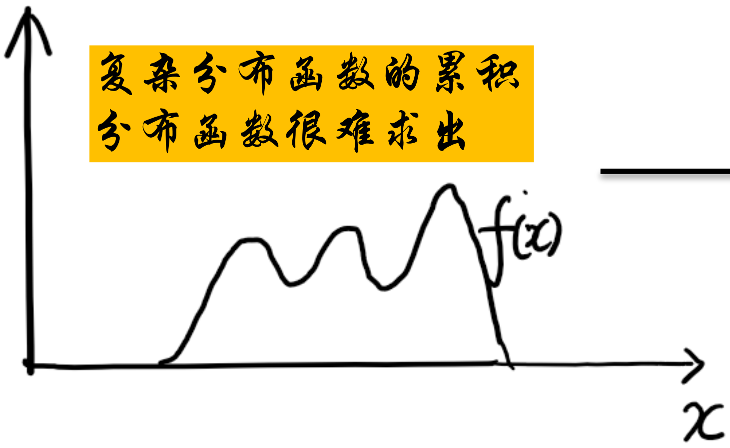
$$i_{n+1}=(a i_{n-1}+c)(mod\ M)$$

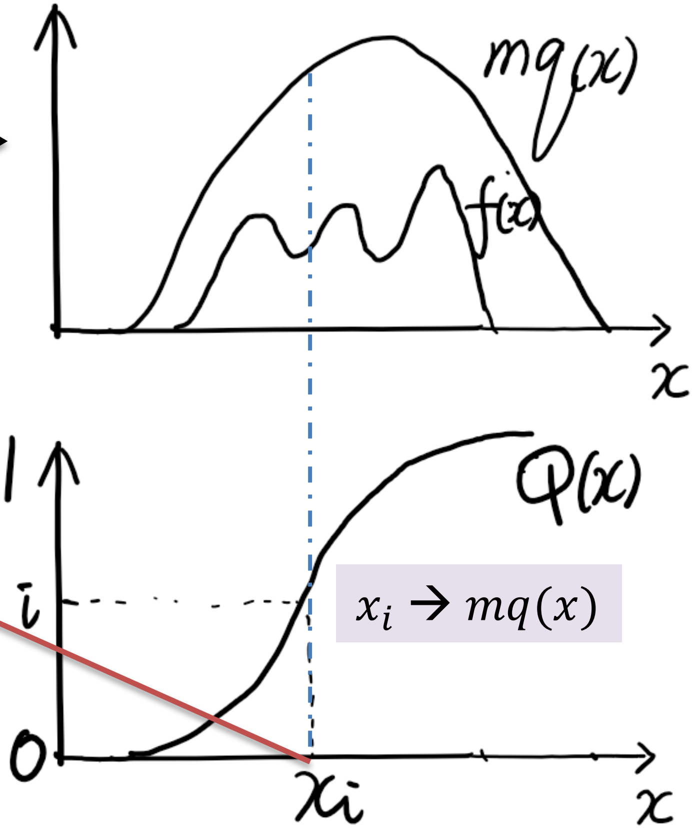
rand()/(RAND_MAX+1.) 随机数

$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots,\quad x\in(-\infty,+\infty)$$

取舍采样rejection sampling

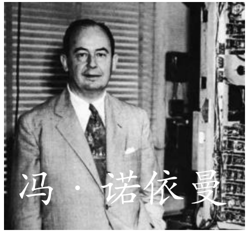
复杂分布函数的累积分布函数很难求出





$g(x)$的 CDF 可求

$f(x) \le mq(x)$



以概率$\mathbf{p}=\frac{f(x)}{mq(x)}$接受

以概率$1-p$拒绝

$x_i \to mq(x)$

$i \sim U(0,1)$

若$i<p$，则接受x_i

否则重新抽样

$X \sim f(x)$

合适的$q(x)$

那有那么好找吗？

高维不适用

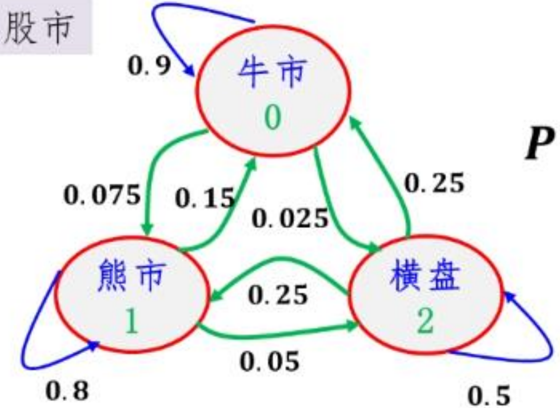
$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots,\quad x\in(-\infty,+\infty)$$

马尔科夫链MC采样

假设状态序列为$\cdots x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2}, \cdots,$

$$P(x_{t+1}|\cdots, x_{t-2}, x_{t-1}, x_t) = P(x_{t+1}|x_t)$$

股市



$$\begin{array}{cccc} & 牛市 & 熊市 & 横盘 \\ P = & \multicolumn{3}{c}{\begin{pmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{pmatrix}} \end{array}$$

状态转移矩阵

初始概率分布&状态转移矩阵P→稳定的概率分布

$\pi^0 \to \pi^1 \to \pi^2 \to \cdots \to \pi^t \to \pi^{t+1} \to \cdots \to \pi^m \to \cdots$

$\pi(x)$

$\pi P = \pi$

隐马尔科夫模型

https://www.bilibili.com/video/BV13C4y1W7iB

敢问如何找到这个P呢？

$\pi(i)P(i,j) = \pi(j)P(j,i)$ 细致平衡条件

⇓

$$\sum_{i=1}^{\infty}\pi(i)P(i,j) = \sum_{i=1}^{\infty}\pi(j)P(j,i) = \pi(j)\sum_{i=1}^{\infty}P(j,i) = \pi(j)$$

对任意Q $\pi(i)Q(i,j) \neq \pi(j)Q(j,i)$

随机取矩阵Q,引入α,满足：

$$\pi(i)Q(i,j)\boxed{\alpha(i,j)} = \pi(j)Q(j,i)\boxed{\alpha(j,i)}$$

$$e^x = \sum_{n=0}^{\infty}\frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots, \quad x \in (-\infty, +\infty)$$

Metropolis算法

平稳分布：$\pi P=\pi$

随机矩阵Q,引入α,满足：

$$\pi(i)Q(i,j)\alpha(i,j)=\pi(j)Q(j,i)\alpha(j,i)$$

$$\alpha(i,j)=\pi(j)Q(j,i)$$

$$\alpha(j,i)=\pi(i)Q(i,j)$$

接受概率

概率 [0,1]

$$P(i,j)=Q(i,j)\alpha(i,j)$$

通过随机转移矩阵Q进行采样，但是样本被保留下来继续抽样的概率为α

$$\alpha(i,j)=min\{\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)},1\}$$

Metropolis-Hastings采样

$$\pi(i)Q(i,j)\cdot min\left(\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)},1\right)=min\left(\pi(j)Q(j,i),\pi(i)Q(i,j)\right)$$

$$=\pi(j)Q(j,i)\cdot min\left(1,\frac{\pi(i)Q(i,j)}{\pi(j)Q(j,i)}\right)=\pi(j)Q(j,i)\alpha(j,i)$$

Los Ala

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

$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots,\quad x\in(-\infty,+\infty)$$

WUHAN TEXTILE UNIVERSITY

Metropolis MC

步骤：

- 给定任意的转移矩阵Q、平稳分布$\pi(x)$
- t=0随机产生一个初始状态x_0
- 从条件概率分布$Q(x|x_0)$中采样x^*
- 从均匀分布产生$u \sim U(0,1)$
- 若$u < \alpha(x_0, x^*) = min(\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)}, 1)$,接受$x^*$ →t=1, $x_1 = x^*$
- 否则拒绝该次采样，$t = 1, x_1 = x_0$
- 继续以上步骤，直到t>T时，达到平衡
- t>T之后的所有接受样本即需要的平稳分布样本

那么，
实际怎么用呢?

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots, \quad x \in (-\infty, +\infty)$$

参考文献

太过理论
内容太充实

https://www.cnblogs.com/pinard/p/6625739.html

MCMC(一)蒙特卡罗方法

MCMC(二)马尔科夫链

MCMC(三)MCMC采样和M-H采样

MCMC(四)Gibbs采样

https://www.bilibili.com/video/BV1qp411R76y?from=search&seid=9283693584074865546

$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots,\quad x\in(-\infty,+\infty)$$

下节内容预告

- Monte Carlo 实现与应用

✓ 圆周率π的求解

✓ 定积分求解实例

✓ 简单物理模型展示MCMC的运用

$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots,\quad x\in(-\infty,+\infty)$$

武漢紡織大學 WUHAN TEXTILE UNIVERSITY

$e^{i\pi}+1=0$ Euler's Identity

算法讲解--Monte Carlo

应用篇-求解定积分

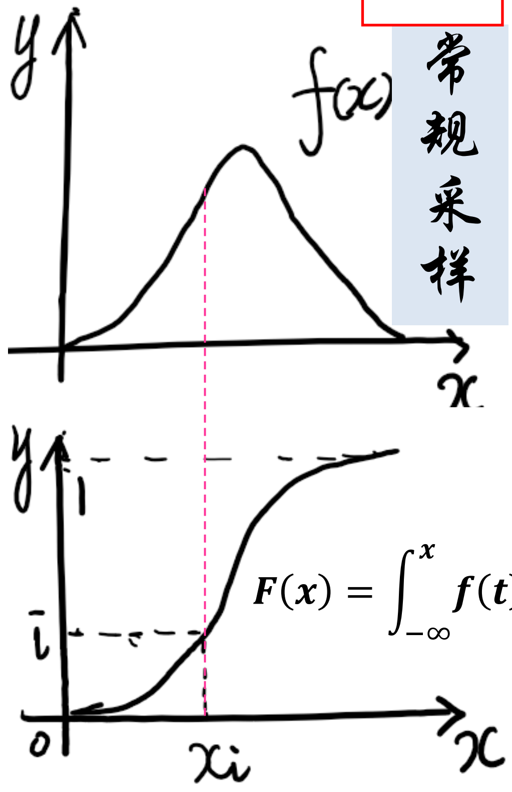
时亚洲

2020.10.6

$$e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots,\quad x\in(-\infty,+\infty)$$

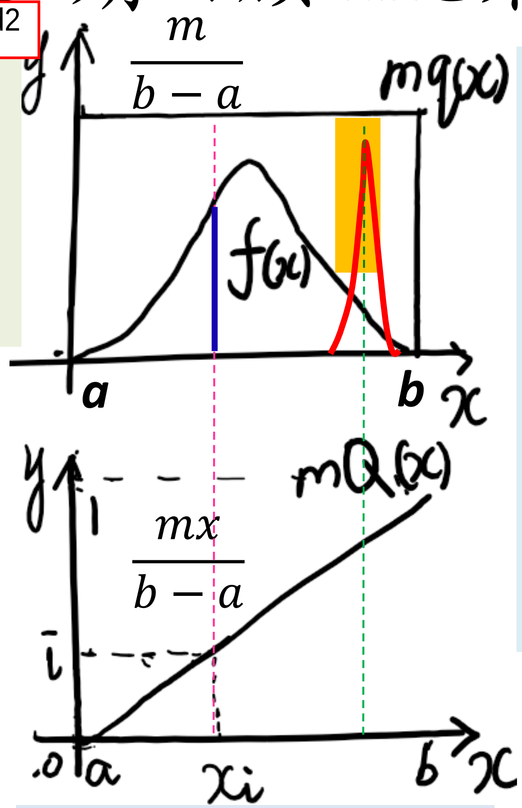
上期回顾-MC采样

method1 常规采样



取$i \sim U(0,1)$ ⇨ $x_i = F^{-1}(i)$

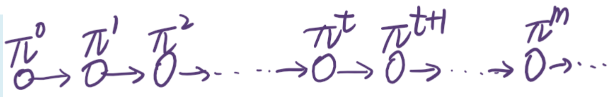
method2 拒绝采样



以概率$\mathbf{p} = \frac{f(x)}{mq(x)}$接受

$u \sim U(0,1), u < p$ √

马尔科夫链MC采样



$\boldsymbol{\pi(x)}$ ⇐ $\pi P = \pi$

⇑

细致平衡 $\pi(i)P(i,j) = \pi(j)P(j,i)$

⇓

$\pi(i)Q(i,j)\alpha(i,j) = \pi(j)Q(j,i)\alpha(j,i)$

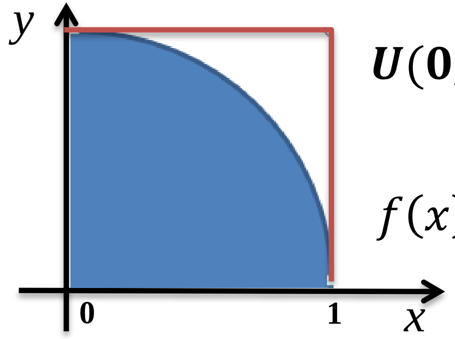
⇓

$P(i,j) = Q(i,j)\alpha(i,j)$

$\alpha(i,j) = min\{\frac{\pi(j)Q(j,i)}{\pi(i)Q(i,j)}, 1\}$

以随机转移矩阵Q采样，以概率α接受，$u \sim U(0,1), u < \alpha$ √

计算圆周率/定积分



$U(0,1)$ **rejection sampling**

$f(x) = \sqrt{1-x^2}$

R语言

```
PI.i<-function(N) { #抽样次数
  count<-0
  for (i in 1:N) {
    x<-runif(1,0,1) #产生随机数x
    y<-runif(1,0,1) #产生随机数y
    if(x^2+y^2<=1) count<-count+1
  }
  return(4*count/N) #返回π值
}
```

```
> PI.i(1000)
[1] 3.172
```

```
> PI.i(100000)
[1] 3.142
```

```
> PI.i(10000000)
[1] 3.141902
```

$$S = \frac{1}{4}\pi r^2 = \int_0^1 f(x)dx \Rightarrow \boxed{\pi = 4\int_0^1 f(x)dx}$$

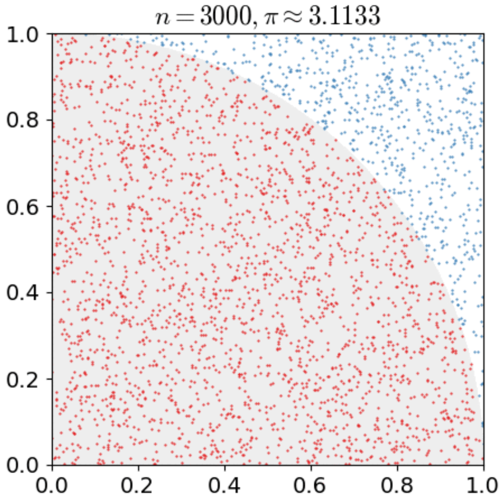
$$\int_0^1 f(x)dx = \int_0^1 dx \int_0^{f(x)} dy = P(Y \le f(X))$$

$X \sim U(0,1), Y \sim U(0,1)$, 计算$y \le f(x)$的概率(频率)即可

C: rand()/(RAND_MAX+1.) $x^2 + y^2 \le 1$

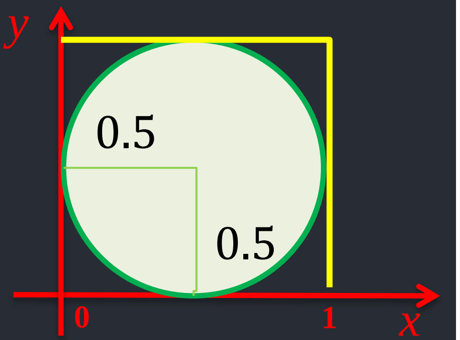
R: runif(1,0,1)

Python: random.random()

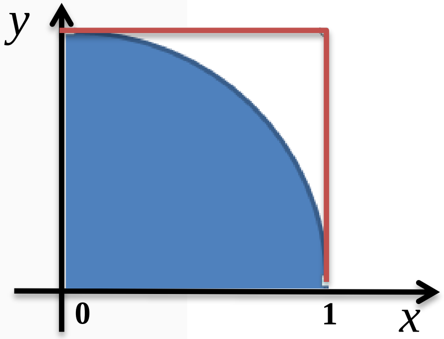


C语言&Python实现

```c
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<math.h>

//Monte Carlo simulation of Pi
void main()
{
    double x, y, pi;
    int n, k;
    srand((unsigned)time(NULL)); //播种随机数
    for (n = 1, k = 0; n <= 200000; n++) //将20万个点随机撒在范围x (
    {
        x = rand() / (double)RAND_MAX;
        y = rand() / (double)RAND_MAX;
        if (sqrt((x - 0.5)*(x - 0.5) + (y - 0.5)*(y - 0.5)) < 0.5)
        {
            k++;
        }
    }
    pi = k / (0.25 * 200000);
    printf("the pi is %7.5lf\n",pi);  //根据圆的面积公式近似模拟出pi
```



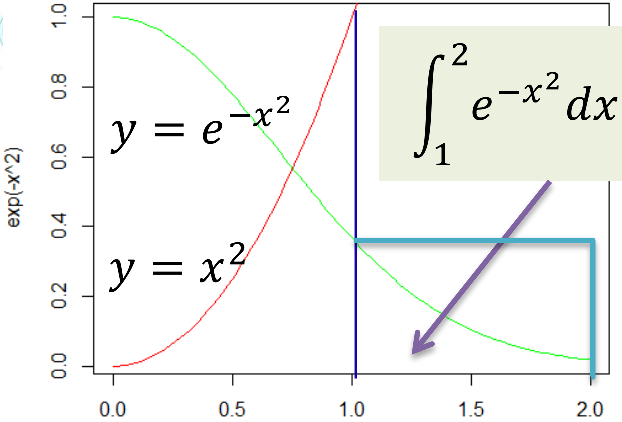

```python
#Created by Lu Zhan

import random
import math


def monteCarlo(N):
    i = 0
    count = 0
    while i <= N:
        x = random.random()
        y = random.random()
        if pow(x, 2) + pow(y, 2) < 1:
            count += 1
        i += 1
    pi = 4 * count / N
    print(pi)

monteCarlo(1000000)
```



https://blog.csdn.net/qq_23927381/article/details/89353015 **利用C语言进行蒙特卡罗模拟圆周率**

https://blog.csdn.net/luzhan66/article/details/82822576 **蒙特卡洛方法及计算圆周率(Python实现)**

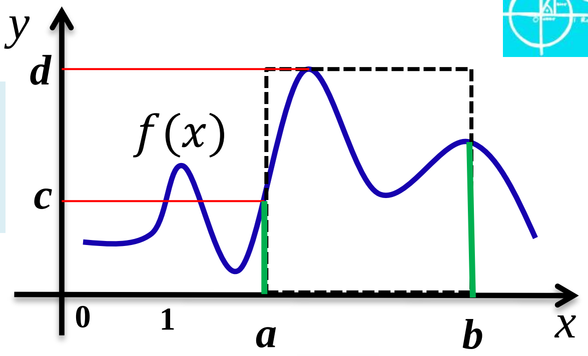
定积分



$$\int_1^2 e^{-x^2} dx = ?$$

$$S = S_0 P$$

$$\int_a^b f(x)dx = ?$$




```
s.f<-function(N) {
  x<-runif(N,0,1)
  y<-runif(N,0,1)
  n<-length(x[which(y<x^2)]) #计算落入区域的点数
  return(n/N) #计算概率
}
```

$$\int_0^1 x^2 dx = \frac{1}{3}$$

```
> s.f(1000)
[1] 0.363
```

```
> s.f(1000000)
[1] 0.333504
```

```
s.f<-function(N) {
  x<-runif(N,0,1)
  y<-runif(N,0,1)
  n<-length(x[which(y<exp(-x^2))]) #计算落入区域的点数
  return(n/N) #计算概率
}
```

$$\int_0^1 e^{-x^2} dx = ?$$

```
> s.f(1000000)
[1] 0.745933
```

```
f<-function(x){
  f(x)
}
s.f<-function(N){
  x<-runif(N,a,b)
  c<-min(f(x))
  d<-max(f(x))
  y<-runif(N,0,d)
  n<-length(x[which(y<f(x))])
  s.j<-(b-a)*d
  return(s.j*n/N)
}
s.f(100000)
```

x^2

a=2
b=5

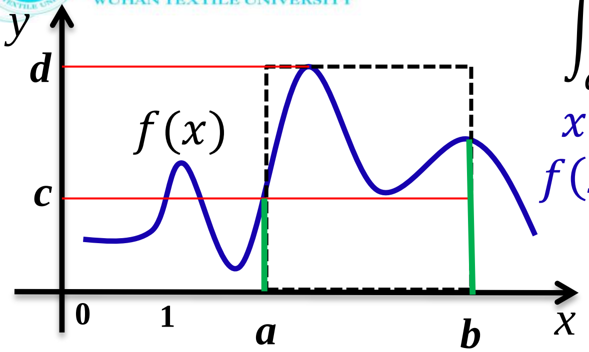
```
> s.f(100000)
[1] 38.9789
```

直接投点法

$$\int_2^5 x^2 dx = 39$$

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots, \quad x \in (-\infty, +\infty)$$

武汉纺织大学
WUHAN TEXTILE UNIVERSITY



$$\int_a^b f(x)dx$$

$x \in [a, b]$

$f(x) \in [c, d]$

思考：
$f(x)$正负均有，
如何做？

令$x = (b-a)t + a$, $t \in [0,1]$, $dx = (b-a)dt$

$$\int_a^b f(x)dx = (b-a)\int_0^1 f[(b-a)t+a]dt$$

$$= (b-a)\int_0^1 \{f[(b-a)t+a] - c + c\}dt$$

$$= (b-a)(d-c)\int_0^1 \frac{f[(b-a)t+a]-c}{d-c}dt + (b-a)c$$

$$0 < g(t) < 1$$

$$= (b-a)(d-c)\int_0^1 g(t)dt + (b-a)c$$

```
a=2
b=5
f<-function(x){
  x^2
}
s.f<-function(N){
  x<-runif(N,a,b)
  c<-min(f(x))
  d<-max(f(x))
  s.j<-(b-a)*d
  t<-runif(N,0,1)
  g<-function(x){
    (f((b-a)*t+a)-c)/(d-c)
  }
  y<-runif(N,0,1)
  n<-length(t[which(y<g(t))])
  return((b-a)*(d-c)*n/N+c*(b-a))
}
```

$$\int_2^5 x^2 dx = 39$$

```
> s.f(100000)
[1] 38.85619
```

变换积分法

```
> s.f(100000)
[1] 38.9789
```

直接投点法

$$e^x = \sum_{n=0}^{\infty}\frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots, \quad x \in (-\infty, +\infty)$$

定积分-重要性采样

$$\int_a^b f(x)dx = \int_a^b \frac{f(x)}{g(x)} g(x)dx$$

若$g(x)$是某个概率密度分布函数

令$Z = \dfrac{f(x)}{g(x)}$

$$\int_a^b g(x)dx = 1$$

$$= E(Z) = \frac{1}{N}\sum_{i=1}^{N} Z_i = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{g(x_i)}$$

$$E(g(X)) = \int_{-\infty}^{+\infty} f(x)g(x)\,dx = \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

$$g(X) \sim U(a,b) = \frac{1}{b-a}$$

$$g(x) = cx$$
$$g(x) = cx^2$$
$$\cdots$$

$$= \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$$

期望法

```
a=2
b=5
f<-function(x){
  x^2
}
s.f<-function(N){
  x<-runif(N,a,b)
  return((b-a)*mean(f(x)))
}
```

```
> s.f(100000)
[1] 38.85619
```

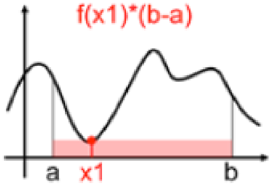
直接投点法

```
> s.f(100000)
[1] 38.9789
```

变换积分法

```
> s.f(100000)
[1] 38.98561
```

期望积分法



积分中值

练习：

$$\int_a^b \frac{sinx}{x} dx$$

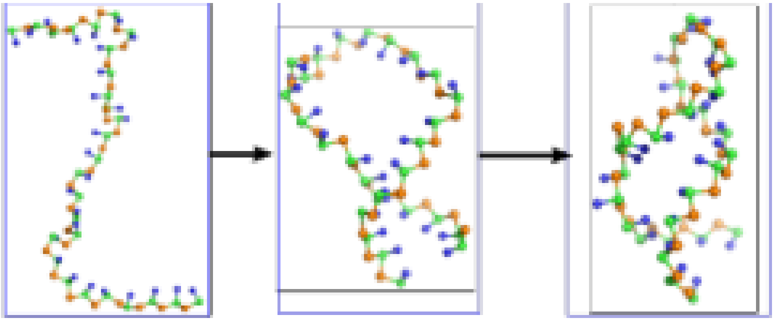
```
a=2
b=5
f<-function(x){
  x^2
}
g<-function(x){
  3/117*x^2
}
s.f<-function(N){
  x<-runif(N,a,b)
  return(mean(f(x)/g(x)))
}
```

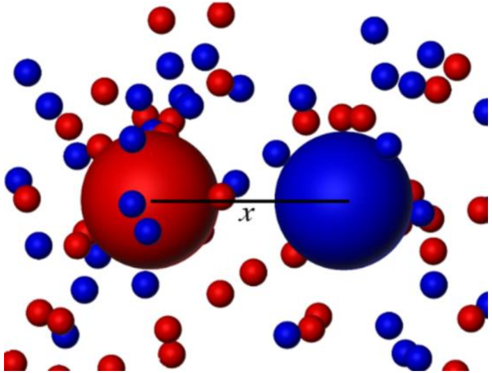
```
> s.f(100000)
[1] 39.08443
```

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots, \quad x \in (-\infty, +\infty)$$

未来预告

- 二重积分的Monte Carlo数值计算 $\theta = \iint_D f(x,y)\,dx\,dy$
- 吉布斯采样原理
- 多球体系(或Ising模型)的Monte Carlo模拟
- 链状分子的Monte Carlo模拟





$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \cdots, \quad x \in (-\infty, +\infty)$$