随机变量与随机过程的产生

目录

1

均匀分布随机变量的产生

均匀分布随机数的产生

蒙特卡洛仿真中的随机采样（1）

- 基于计算机的蒙特卡洛仿真要求在计算机上产生满足一定概率分布的随机数
- 计算机只能按人的意志按照一定的**程序产生数据**，因此必然不可能是纯随机的，只可能是**伪随机的**
- 满足一定条件的**伪随机采样**能够完成仿真评估目标
- 若从已知分布得到的采样**不准确**，则：
  - **可能造成有偏估计**
  - **严重影响仿真精度**

均匀分布随机数的产生

蒙特卡洛仿真中的随机采样（2）

- 传统的随机数产生采用**物理设备、机械方法和电子热噪声**
- 现在多数的随机数据发生器都不基于物理设备，而是在计算机上通过对数的操作实现
  - 运行快，要求的存储空间小
  - 能很容易的**重现给定的随机数**
- 一个好的随机数产生器几乎拥有着真实随机序列的所有**重要的统计特性**
- 但这些序列是由**确定性算法**产生的
- 由于这个原因，这类发生器所产生序列的**随机性常常称为伪随机**

均匀分布随机数的产生

均匀分布随机数产生的重要性

- **均匀分布的随机数**在蒙特卡洛仿真中具有非常重要的地位，是从已知随机分布采样的基础
- 其他分布随机数的产生以均匀分布随机数为基础
- 均匀分布随机数产生方法的优劣决定了**蒙特卡洛仿真从已知分布采样的可信度**，其中产生**[0,1]区间上均匀分布**的随机变量是生成其他随机变量的前提
- 在不少文献中所说的**随机数发生器（RNG）就特指产生[0,1]区间上均匀分布随机变量的算法**

均匀分布随机数的产生

均匀分布随机数产生方法种类

- **线性同余方法**及其变种
  - 乘同余
  - 混合同余
  - 组合
- 非线性同余方法
- 移位寄存器法及其变种
  - Mersenne Twister算法

均匀分布随机数的产生

LCG是由Lehmer在1951年提出的，其是利用数论中的同余运算来产生随机数

线性同余法（1）LCG(linear congruential generator)

- 采用下述公式产生随机数：

$$x_{n+1} = [ax_n + b] \bmod (m)$$

- 其中x_n是第n个随机数，a为乘子，b为增量，m为模数，$0 \le x_0 < m$为随机数源或种子（**seed**），均为非负整数

- 令$u_n := x_n / m$，则可得到$[0,1)$区间上的随机数

均匀分布随机数的产生

线性同余法（2）

$$x_{n+1} = [ax_n + b] \bmod (m)$$

- 若$a!=0, b!=0$，称为混合同余法
- 若$a=1$，$b!=0$，称为加同余法
- 若$a!=0, b=0$，称为乘同余法
- 研究表明，乘同余法优于加同余法，而混合同余法与简单的乘同余法相比并无显著优点
- 乘同余法更常用

均匀分布随机数的产生

线性同余法（3）

- 产生的随机数实际上是伪随机数
- 采用递推算法，后面产生的数会重复前面的数，即实际上是有限的周期数列。无论x_n取何值，其循环顺序相同
- 随机数循环一次称为发生器的一个周期(P)
- 对混合同余法，若$P=m$，则该发生器具有满周期
- 对乘同余法，由于x_i不能取0，因此满周期为m-1
- 受计算机字长的限制，线性同余法的周期一定小于计算机字长

蒙特卡洛仿真通常需要较大的仿真量，如果随机数发生器在仿真时长内发生重复，会对仿真结果可信度造成较大影响，因此**设计乘同余或混合同余发生器时总希望是满周期的**

均匀分布随机数的产生

线性同余法（4）

- 发生器的周期是受a，b，m共同控制的。
- 当且仅当以下特性满足时，混合同余发生器达到最大周期m：
  - b和m互素
  - a-1可以被m的任意素因子整除
  - 如果m是4的倍数，则a-1也必须是4的倍数

均匀分布随机数的产生

线性同余法（7）

- 并不是所有具有满周期的线性同余发生器都是好的随机数发生器
- 良好的随机数发生器必须通过各种随机数测试，以满足**均匀性、独立性、快速性、计算机字长效应**的要求
- 因此，线性同余发生器中的常数选择是一个十分复杂的问题

均匀分布随机数的产生

线性同余法（8）

- 随机数发生器测试比较复杂，不可能任意设计并测试
- 因此有人提出多种“**最小标准**”，最小标准要求：
  - 满周期
  - 通过一组特定的随机性统计测试（不同的测试对应不同的最小标准）
    - 频率测试、块内频率测试、游程测试、累积和测试，等
  - 易于从一台计算机移植到另一台计算机

均匀分布随机数的产生

复合线性同余法

- 在希望获得一些更长周期（并不一定是满周期）的随机数发生器时，改进思想：把具有较短周期的**不同随机数发生器的输出线性合并**
- 两个周期分别为N_1和N_2的随机数发生器合并之后，输出随机数周期为

$$N = lcm(N_1, N_2)$$ lcm：最小公倍数

- 当N_1和N_2互素时，$N = N_1 \times N_2$。
- 用一个随机数发生器控制另一个随机数发生器输出的置乱也可以获得类似的效果

均匀分布随机数的产生

复合线性同余法

$$X(n)=171X(n-1)\bmod 30269$$

$$Y(n)=172Y(n-1)\bmod 30307$$

$$Z(n)=171Z(n-1)\bmod 30323$$

- 三个乘同余发生器复合:

$$U(n)=\{\frac{X(n)}{30269}+\frac{Y(n)}{30307}+\frac{Z(n)}{30323}\}\bmod 1$$

- 周期远大于单个乘同余发生器

均匀分布随机数的产生

线性同余法的其他变种

- 多步递归

$$x_i \equiv (a_1 x_{i-1} + a_2 x_{i-2} + \cdots a_k x_{i-k}) \bmod m$$

- 滞后斐波那契

$$x_i \equiv (x_{i-j} + x_{i-k}) \bmod m$$

- 矩阵同余

$$x_i \equiv (A x_{i-1} + c) \bmod m$$

  - A为矩阵，x_i，x_{i-1}和c为向量

复合多步递归生成器的代表：MRG32k3a

- $X_t=(1403580X_{t-2}-810728X_{t-3})\ mod\ m_1$
- $Y_t=(527612Y_{t-1}-1370589Y_{t-3})\ mod\ m_2$
- 其中：
  - $m_1 = 2^{32}$-209
  - $m_2 = 2^{32}$-22853
- 复合输出：$U_t=\begin{cases}\frac{X_t-Y_t+m_1}{m_1+1} & X_t \le Y_t\\ \frac{X_t-Y_t}{m_1+1} & X_t > Y_t\end{cases}$

周期约为$3*10^{57}$

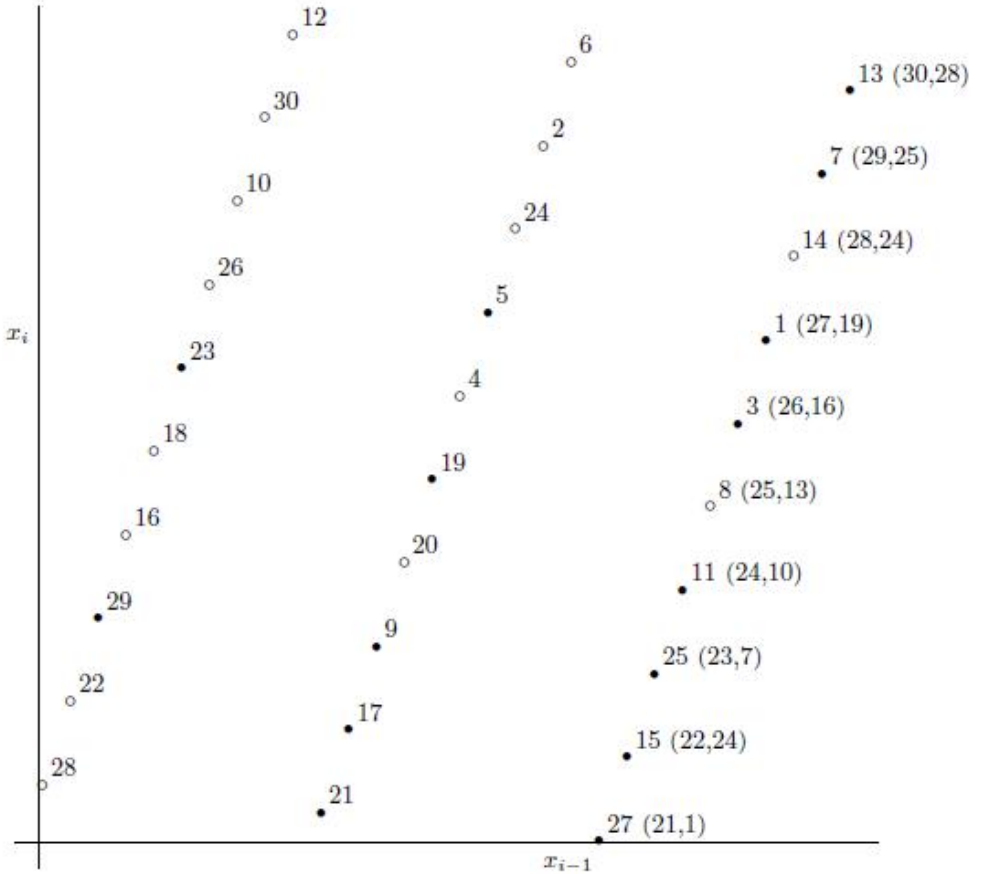
在一些主流软件包中得到应用：Matlab，NS-3，MKL（intel数学函数库）等。

- 例如：Matlab中的RandStream类

均匀分布随机数的产生

线性同余法的缺点

- 线性同余法计算简单，参数选取适当的线性同余随机数发生器能够通过大多数随机数测试方法
- 但是，受到**字长和模数的限制**，线性同余法的**周期一般不可能很长**
- 线性同余法产生伪随机数的另一个重要问题在于：它的**结构性**
  - 总是存在某种晶格结构（Lattice）
  - 参数选择较好的线性同余发生器在高维空间上才能看出晶格结构
- 这种结构性带来的相关性在某些情况下会造成仿真结果的偏差



$x_i = 3x_{i-1} \bmod 31$

例如27(21,1)中：

- 27为随机数序号
- 21为x_{i-1}，1为x_i

均匀分布随机数的产生

非线性同余生成器

➢加法进位法AWC

$$x_i \equiv (x_{i-s} + x_{i-r} + c_i) \bmod m$$

- $c_1 = 0$
- $c_{i+1} = 0$，if $x_{i-s} + x_{i-r} + c_i < m$
  $c_{i+1} = 1$，if $x_{i-s} + x_{i-r} + c_i \geqslant m$

c_i为进位项

- 减法借位法SWB
- 乘法进位法MWC

➢逆同余

$$x_i \equiv \left(ax_{i-1}^{-1} + c\right) \bmod m$$

➢二次同余

$$x_i \equiv \left(dx_{i-1}^2 + ax_{i-1} + c\right) \bmod m$$

均匀分布随机数的产生

FSR方法产生随机数(1)

- FSR：反馈移位寄存器

$$b_i \equiv \left(a_p b_{i-p} + a_{p-1} b_{i-p+1} + \cdots + a_1 b_{i-1}\right) \bmod 2$$

- 多个连续的b看作一个多位二进制数
- 满周期：2^p-1
- 满周期条件：$f(z) = z^p - \left(a_1 z^{p-1} + \cdots + a_{p-1} z + a_p\right)$ 为本原多项式
  - 所有系数的最大公因数为1的多项式

均匀分布随机数的产生

FSR方法的变种

- 多个FSR的组合
- 矩阵形式的FSR
- TGFSR：Twisted Generalized Shift Register

$$x_i = x_{i-p} \oplus Ax_{i-p+q}$$

- 这里：$\oplus$ 为异或。

均匀分布随机数的产生

Mersenne Twister(1)

- Mersenne Twister是迄今为止**最好的随机数发生器之一**
- 被标准C++（C++11）、Python、Matlab等流行的编程工具采用
- 属于FSR产生方法的变种
- MT19937-32和MT19937-64应用最为广泛
- 周期为2^19937-1

均匀分布随机数的产生

Mersenne Twister(2)

➢Mersenne Twister递推式：

$$x_{k+n} := x_{k+m} \oplus \left(x_k^u | x_{k+1}^l\right)A, \quad k = 0,1, \ldots$$

- x_i是w位的字向量
- $\left(x_k^u | x_{k+1}^l\right)$：将$x_k$的前$w$-$r$位与$x_{k+1}$的后$r$位连接，组成一个新的$w$位的字向量

➢Twist：

$$A = R = \begin{pmatrix} 0 & \mathbf{I}_{w-1} \\ a_{w-1} & (a_{w-2}, \ldots a_0) \end{pmatrix}$$

➢A的特殊结构使得tA的计算得以简化：位移运算

$$tA = \begin{cases} t \gg 1 & t_0 = 0 \\ (t \gg 1) \oplus a & t_0 = 1 \end{cases}$$

- $t = t_{w-1}, t_{w-2}, \ldots t_0$是$w$位的字向量
- $a = a_{w-1}, a_{w-2}, \ldots a_0$

均匀分布随机数的产生

Mersenne Twister(2)

- Temper：为改善均匀性进一步引入的运算

$$y := x \oplus (x >> u)$$

$$y := :y \oplus ((y << s) \,\&\, b)$$

$$y := :y \oplus ((y << t) \,\&\, c)$$

$$z := y \oplus (y >> l)$$

均匀分布随机数的产生

Mersenne Twister(3)

➢MT19937-32参数

$(w, n, m, r) = (32, 624, 397, 31)$

$a = 9908B0DF_{16}$

$f = 1812433253$

$(u, d) = (11, FFFFFFFF16)$

$(s, b) = (7, 9D2C5680_{16})$

$(t, c) = (15, EFC60000_{16})$

$l = 18$

- w：长度（以bit为单位）
- n：递归长度
- m：周期参数，用作第三阶段的偏移量
- r：低位掩码/低位要提取的位数
- a：旋转矩阵的参数
- f：初始化梅森旋转链所需参数
- b,c：TGFSR的掩码
- s,t：TGFSR的位移量
- u,d,l：额外梅森旋转所需的掩码和位移量

均匀分布随机数的产生

Mersenne Twister(4)

- Mersenne Twister有以下优点：
  - 周期长
  - 随机性好
    - 623维均匀分布：624维上才有Lattice结构
    - 通过了绝大多数随机性测试，包括一些要求非常苛刻的测试

Matlab中的随机数生成器

- RandStream.list：列出了当使用RandStream或RandStream.create创建随机数流时，可以使用的所有生成器算法。
- 右表给出了可用的生成器算法及其属性。

关键字	生成器	多流和子流支持	全精度近似周期
mt19937ar	梅森旋转 (MATLAB®启动时默认流使用的算法)	否	$2^{19937}-1$
dsfmt19937	面向SIMD的快速梅森旋转算法	否	$2^{19937}-1$
mcg16807	乘法同余生成器	否	$2^{31}-2$
mlfg6331_64	乘法滞后Fibonacci生成器	是	2^{124} (2^{51}个流，长度为2^{72})
mrg32k3a	组合多递归生成器	是	2^{191} (2^{63}个流，长度为2^{127})
philox4x32_10	执行10轮的Philox 4×32 生成器	是	2^{193} (2^{64}个流，长度为2^{129})
threefry4x64_20	执行20轮的Threefry 4×64 生成器	是	2^{514} (2^{256}个流，长度为2^{258})
shr3cong	移位寄存器生成器与线性同余生成器求和	否	2^{64}
swb2712	修正的借位减法生成器	否	2^{1492}

3

产生其他随机变量的一般方法

产生其他随机变量的一般方法

- 在蒙特卡洛仿真中，我们所需要的不仅仅是(0,1)区间上的均匀分布随机变量
- 必须在(0,1)区间上均匀分布随机变量的基础上，产生其他分布的随机变量
  - 反变换法
  - 组合法
  - 舍选法

产生其他随机变量的一般方法

反变换法（1）

- 一种最常用、最直观的随机变量产生方法
- 基本思想：将累积分布函数的反函数作用于均匀分布随机变量，以此产生目标随机变量的方法。

- 由概率论可知，累积分布函数具有以下特点：
  - 单调不减函数
  - 右连续函数
  - 值域为[0,1]

- 如果目标随机变量的累积分布函数具有闭式表达式，都可以得到其反函数，其反函数的定义域为[0,1]区间。

→ 连续随机变量

→ 离散随机变量

反变换法（2）

➢连续随机变量的反变换法

- 如果连续分布随机变量X的累积分布函数为$F_X(x)$，则变量$U=F_X(x)$服从[0,1]区间上的均匀分布
- 由均匀分布的独立随机序列U经过如下反变换，即可获得具有累积分布函数$F_X(x)$的随机变量序列

$$X=F_X^{-1}(U)$$

产生其他随机变量的一般方法

反变换法（3）

➢连续随机变量的反变换法基本流程：

- 用随机数发生器产生在[0,1]区间上均匀分布的、独立的随机变量U。
- 将U代入到反变换$X = F_X^{-1}(U)$中，得到随机变量X。

此方法条件 ⇒ **所需分布的累积分布函数及其反函数具有解析式表达**

产生其他随机变量的一般方法

反变换法（4）

例：医院患者在超声科的服务时间可以建模为指数分布的随机变量。为了更好安排医生排班计划，使患者得到及时治疗，现需要产生指数分布的随机变量作为后续分析。

- 指数分布的累积分布函数

$$F_X(x)=1-e^{-\lambda x} \qquad x\geq 0$$

- 累积分布函数的反函数为

$$F^{-1}(u)=-\frac{1}{\lambda}\ln(1-u)$$

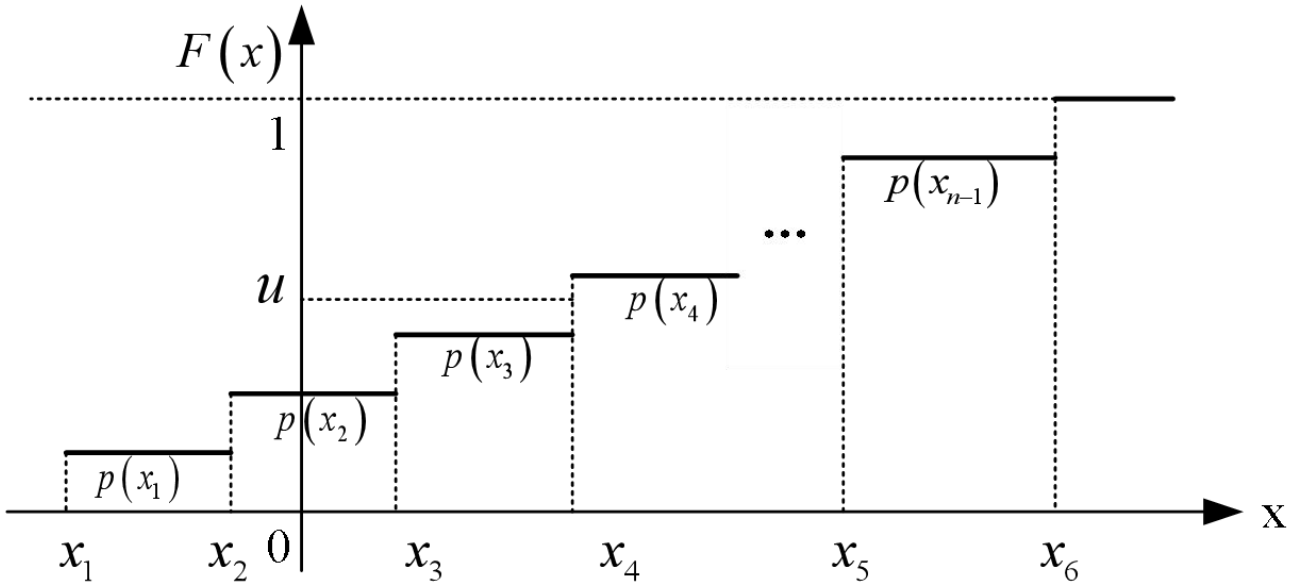
- 指数分布随机变量的产生方法
  - 产生[0,1]区间上均匀分布的随机变量$u'(u'=1-u)$
  - 产生服从参数为λ的指数分布的随机变量 $x=-\frac{1}{\lambda}\ln u'$

产生其他随机变量的一般方法

反变换法（5）

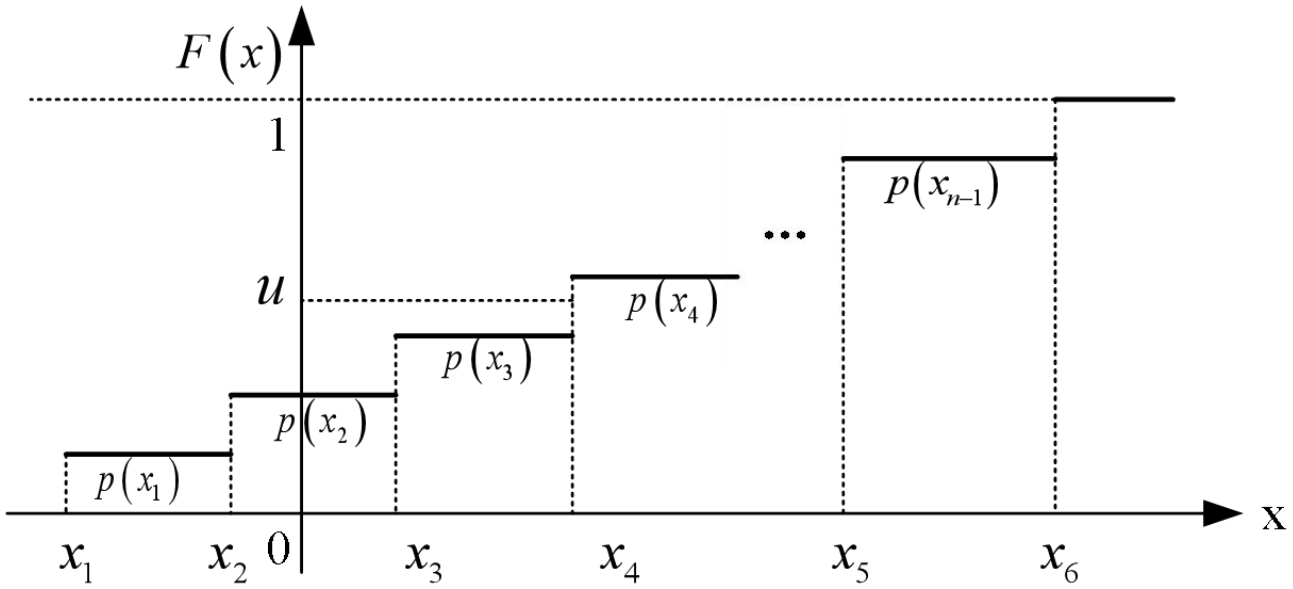
➢离散随机变量的反变换法

- 已知**概率分布**的离散随机变量，适合采用反变换法
- 离散随机变量 x 分别以概率 $p(x_1), p(x_2), \ldots, p(x_n)$ 取值 $x_1, x_2, \ldots, x_n$，其中 $0 < p(x_i) < 1$，且 $\sum p(x_i) = 1$
- 累积分布函数如图



产生其他随机变量的一般方法

反变换法（6）



- 离散随机变量的反变换法基本流程：

- 将 x_i 按照从小到大的顺序排序，得到分布函数子区间的分界点。
- 由[0,1]区间上均匀随机数发生器产生随机变量u。
- 如果$u \le p(x_1)$，则$x = x_1$；若$p(x_1) < u \le p(x_1) + p(x_2)$，则$x = x_2$.....

产生其他随机变量的一般方法

反变换法（7）

优点

- 根据累积分布函数推导，非常直观，便于理解

缺点

- 所需连续分布的累积分布函数必须具有解析式表达
- 累积分布函数的反函数必须具有解析式表达

→ 组合法

舍选法

产生其他随机变量的一般方法

组合法（1）

- 基本思想：当目标连续随机变量x的累积分布函数$F(x)$可以表示为多个其它累积分布函数加权和的形式，

$$F(x) = \sum p_i F_i(x)$$

$$\text{其中 } p_i \geq 0, \sum p_i = 1$$

而这些分布函数$F_i(x)$比目标变量分布函数$F(x)$更容易采样时，适合采用**组合法**。

产生其他随机变量的一般方法

组合法（2）

➢组合法基本流程：

- 产生随机整数I，使$P\{I=i\}=p_i$
- 产生具有分布函数$F_i(x)$的随机变量 x_i
- 令$x=x_i$

确定采用哪一个分布函数来取样，可采用**离散反变换法**来实现

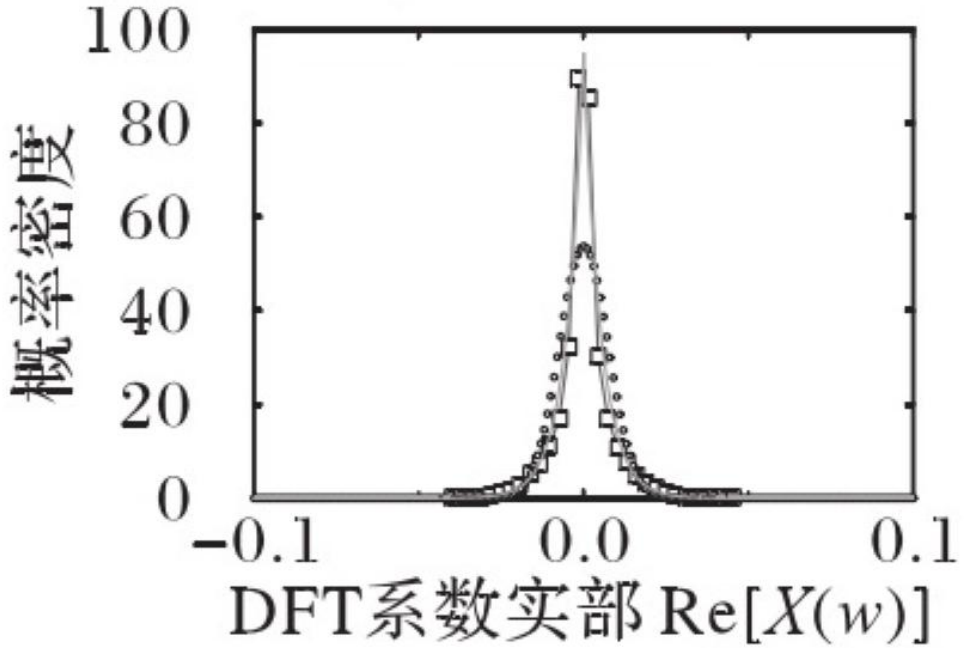
以该分布函数产生随机变量，可采用**连续反变换法**来实现

产生其他随机变量的一般方法

组合法（3）

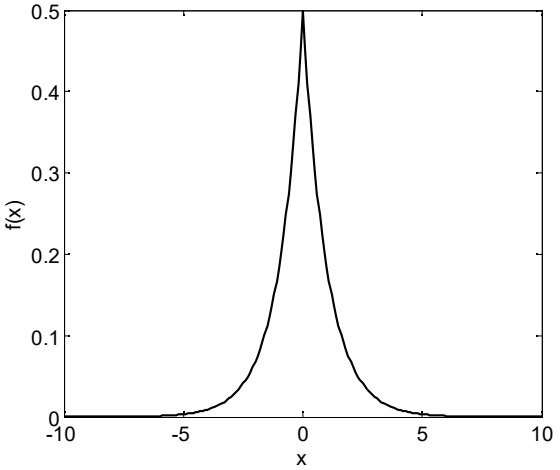
➢例：语音信号的DFT系数满足双指数分布，即概率密度函数 $f(x)=\frac{1}{2\lambda}e^{-\frac{|x|}{\lambda}}$。

为了对语音处理系统进行建模分析，现需要产生满足该分布的随机变量X。



产生其他随机变量的一般方法

组合法（4）



- 假设$\lambda=1$，产生概率密度函数为$f(x)=0.5e^{-|x|}$的随机变量

$$f(x)=0.5e^{-x}u(x)+0.5e^{x}u(-x)$$

- 定义：$f_1(x)=e^{-x}u(x)$ 和$f_2(x)=e^{x}u(-x)$，两者都符合概率密度函数条件
- 可以看做两个新的分布的概率密度函数，而且其累积分布函数及其反函数都非常容易获得，适合用组合法生成

产生其他随机变量的一般方法

u_1：确定采用哪一个分布函数来取样；

u_2：用于产生该分布函数相应的随机变量

组合法（5）

具体步骤：

- 产生(0,1)区间上的均匀随机变量u_1和u_2
- 若$u_1 < 0.5$，则根据$f_1(x)$所对应的累积分布函数采用反变换法得到：

$$x = -\ln u_2$$

- 若$u_1 \geq 0.5$，则根据$f_2(x)$所对应的累积分布函数采用反变换法得到：

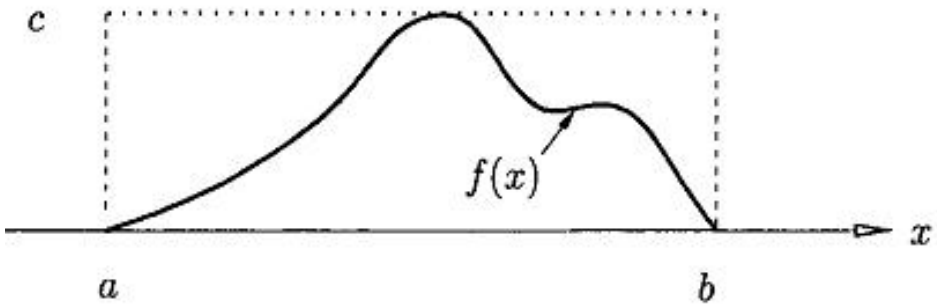
$$x = \ln u_2$$

Matlab 代码示意

```
clc
clear
n=1e7;
u1=rand(1,n);
u2=rand(1,n);
z = zeros(1,n);
for i=1:n
    if u1(i)<0.5
        z(i)=-log(u2(i));
    else
        z(i)=log(u2(i));
    end
end
plot([1:n],z)
```

产生其他随机变量的一般方法

舍选法（1）



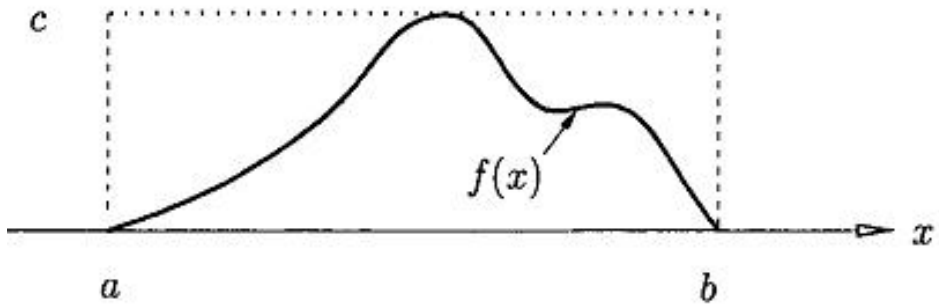
➢舍选法使用场景：目标连续随机变量的累积分布函数难以写成闭式表达式，但是其概率密度函数可以写成闭式表达式形式。

概率密度函数的本质：描述连续随机变量在某个确定取值点附近的可能性。

基本思想：产生随机数u_1，在$f_X(x)$大的地方，保留较多u_1；在$f_X(x)$小的地方，保留较少随机数u_1。使得到的子样本中，u_1的分布满足$f_X(x)$的要求。

产生其他随机变量的一般方法

舍选法（2）



➢舍选法基本流程：

- 产生(a,b)区间内均匀分布的随机变量u_1；
- 产生$(0,c)$区间内均匀分布的随机变量u_2；
- 如果 $u_2 \le f_X(u_1)$ 则 $X = u_1$；否则，回到第(1)步。

$f_X(u_1)$越大，则该不等式成立的概率越大，u_1更容易被保留

缺点：该步骤中，当撒点区域的面积远大于$f_X(x)$所围成面积时，舍选法效率很低

产生其他随机变量的一般方法

舍选法（3）

➢提高舍选法效率的方法：撒点区域往往不设置成矩形区域，而是根据概率密度函数$f_X(x)$的特征，选择一个函数$t(x)$来代替常数c。$t(x)$应满足以下条件：

- $t(x) \geq f_X(x)$
- $\int_{-\infty}^{+\infty} t(x)dx = C < \infty$
- $t(x)$易于进行反变换

产生其他随机变量的一般方法

舍选法（4）

➢令 $r(x)=\frac{1}{C}t(x)$，其中 $C=\int_{-\infty}^{+\infty}t(x)dx$。则 $\int_{-\infty}^{+\infty}r(x)dx=1$，$r(x)$ 可以看做一个概率密度函数。由此，一般的舍选法具体步骤变为：

（1）产生[0,1]区间上的均匀分布随机变量 u_1；

（2）根据 $r(x)$ 产生随机变量 u_2；

（3）检验不等式 $u_1 \le f_X(u_2)/t(u_2)$，如成立，则 $X=u_2(X\sim f_X(x))$，否则返回(1)。

➢舍选法效率的高低与 $t(x)$ 的选择密切相关

$$P\left(u_1 \le \frac{f_X(u_2)}{t(u_2)}\right)=1/C$$

3

高斯分布随机变量的产生

高斯分布随机变量的产生

高斯分布的特殊性

- 在信息网络仿真中会常常碰到高斯随机变量
- 累积分布函数：

$$F(x)=\int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}\sigma}exp\left(-\frac{y^2}{2\sigma^2}\right)dy$$

- **非闭合形式，反变换法不适用**
- 概率密度函数具有闭合形式，舍选法可用，但**效率非常低**
- 需要采用一些**特殊的方法产生高斯分布随机变量**
  - 中心极限定理法
  - Box-Muller法
  - Box-Muller法与舍选法结合

高斯分布随机变量的产生

中心极限定理方法（1）

- 中心极限定理指出，当 $N \to \infty$ 时，N个独立随机变量之和的分布趋近于高斯分布
- 基本步骤：
  - 产生若干个（0,1）区间上的均匀分布随机数U_i
  - $\mathrm{U}_i - \frac{1}{2}$服从（$-\frac{1}{2}$，$\frac{1}{2}$）的均匀分布，方差为$\frac{1}{12}$
  - N个$\mathrm{U}_i - \frac{1}{2}$相加，近似服从均值为均值为0，方差为$\frac{N}{12}$的高斯分布
  - 取$B = \sqrt{\frac{12}{N}}$，则$\mathrm{X} = B\sum_{i=1}^{N}\left(\mathrm{U}_i - \frac{1}{2}\right)$近似服从标准高斯分布
  - 一般，取：$N = 12$

高斯分布随机变量的产生

中心极限定理方法（2）

- 中心极限定理法的**截尾现象**
  - 按上述参数得到的随机变量只能在区间(-6,6)区间上取值
  - 利用中心极限定理方法产生高斯分布随机变量时，实际产生变量的范围总是无法达到**高斯分布的尾部**，这种现象称为截尾
  - 增大N值，可以减小这种影响，却会增大计算量
- 因此，如果采用中心极限定理方法产生高斯分布，应该根据需要合理选取N值，以在**仿真速度和精度**之间作出折中

高斯分布随机变量的产生

Box-Muller方法（1）

- 根据瑞利分布和高斯分布之间的对应关系
  - 如果X和Y为均值为0、方差为σ^2的独立高斯随机变量，那么$R=\sqrt{X^2+Y^2}$服从参数为σ的瑞利分布
  - 定义：把X和Y分别作为复平面的两个坐标，则相角θ服从$[0,2\pi]$上的均匀分布

  - 将该关系反过来，就是Box-Muller方法的原理
  - 由George Box和Mervin Muller在1958年提出

高斯分布随机变量的产生

Box-Muller方法（2）

- 瑞利分布的概率密度函数和累积分布函数

$$f_R(r) = \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right)u(r)$$

$$F_R(r) = \int_0^r \frac{y}{\sigma^2}\exp\left(-\frac{y^2}{2\sigma^2}\right)dy = 1 - \exp\left(-\frac{r^2}{2\sigma^2}\right), r \geq 0$$

- 瑞利分布的累积分布函数具有闭式表达式，可以采用反变换法产生瑞利分布随机变量

高斯分布随机变量的产生

Box-Muller方法（3）

- Box-Muller方法的步骤
  - 产生两个独立的(0,1)区间上均匀分布随机变量u_1和u_2；
  - 根据瑞利分布的累积分布函数，对u_1进行反变换

$$F_R(r) = 1 - \exp\left(-\frac{r^2}{2\sigma^2}\right), r \geq 0$$

  - 产生两个独立的、均值为0方差为1的高斯分布随机变量。

$$X = \sqrt{-2\ln u_1}\cos(2\pi u_2)$$

$$Y = \sqrt{-2\ln u_1}\sin(2\pi u_2)$$

高斯分布随机变量的产生

Box-Muller与舍选法的结合(1)

- 标准的Box-Muller方法需要计算三角函数，计算复杂度较高
- 将Box-Muller和舍选法结合形成另一种方法。

(1)产生两个独立的在[-1,1] 上均匀分布的随机变量U_1和U_2;

(2)令$S = U_1^2 + U_1^2$。如果 $S \geq 1$，舍弃并返回上一步；否则，进入下一步；

(3)
$$X = U_1\sqrt{\frac{-2\ln S}{S}}, \qquad Y = U_2\sqrt{\frac{-2\ln S}{S}}$$

则 X、 Y即为标准高斯分布的随机变量。

高斯分布随机变量的产生

Box-Muller与舍选法的结合(2)

- 也称为极坐标法
- 从运算上，这种方法方法与标准的Box-Muller方法相比，避免了三角函数运算，这样使运算速度加快
- 但是它采用了舍选法，效率有所下降
- 同时，多次调用极坐标算法产生的高斯随机变量对的个数也是随机变量，这会使得仿真程序更加复杂。

其他常见分布的产生方法：连续分布

- $[a,b]$区间上的均匀分布：$X=a+U(b-a)$
- 指数分布：$X=-\ln(U)/\lambda$
- 瑞利分布：$X=\sqrt{-2\sigma^2\ln(U)}$
- 自由度为r的χ^2分布：生成r个独立高斯，求其平方和；
- 对数正态分布：$Y=\exp(X)$, X服从正态分布

其他常见分布的产生方法：离散分布

- 贝努利分布：$B=\begin{cases}1 & U\le p\\ 0 & U>p\end{cases}$
- 二项分布：$X=\sum_{i=1}^{n}B_i$
- 泊松分布

指数分布：独立随机事件的时间间隔，
泊松分布：单位时间内的随机事件数。

  - 方法1：产生指数分布随机数并求和，直到大于1为止，输出个数k；
  - 方法2：产生均匀分布随机数并求积，直到小于exp(-λ)为止，输出个数k；

其他常见分布的产生方法：离散分布

➢几何分布：

- 用来表示贝努利实验中，事件第一次发生时所对应的实验次数；
- 概率分布：$P(X = k) = p(1-p)^{k-1} \quad k = 1,2, \ldots$
- 产生方法：
  - 产生指数分布随机数$Y \sim \exp(\lambda)$，其中$\lambda = -\ln(1-p)$
  - $X = 1 + [Y]$

4

随机过程的产生

随机过程的产生

独立向量的产生

- 产生随机过程实际上就是产生一个离散的样本序列
- 长度为n的样本序列为向量$X=\{x_1,x_2,\ldots,x_n\}$
- 产生满足某种条件的随机过程本质上是：产生联合概率密度函数为$f(x)$或联合概率分布函数为$F(x)$的一组序列；
- 如果要产生的随机过程样本之间独立同分布，那么问题就变为n个独立同分布随机变量的产生问题
  - 反变换
  - 舍选法
  - 组合法
  - ……

随机过程的产生

不独立向量的产生

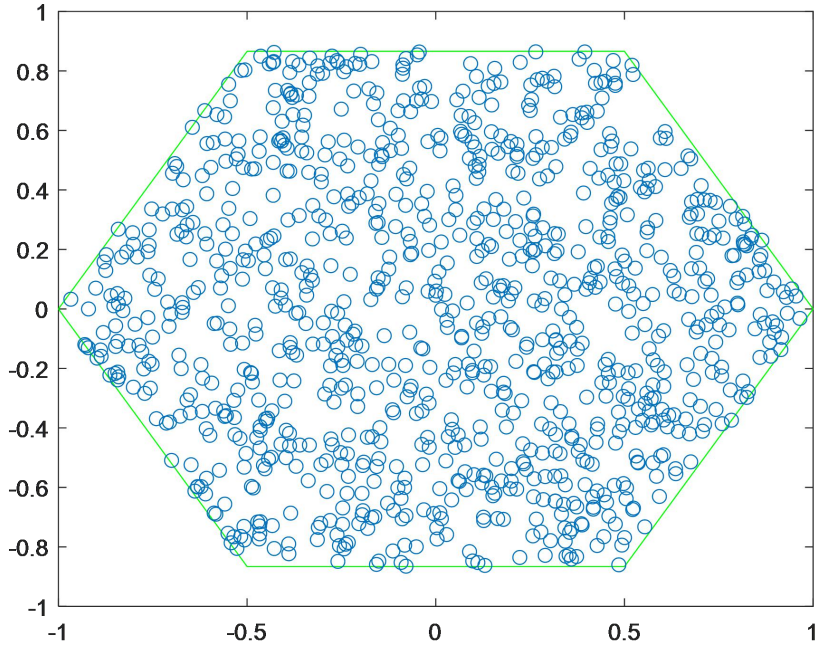
- 已知联合概率密度函数：
  - $f(X)=f(x_1)f(x_2|x_1)f(x_3|x_1x_2)\ldots f(x_n|x_1x_2\ldots x_{n-1})$
- 已知概率密度函数$f(x_1)$，用产生随机变量的方法即可产生x_1；
- 在得到x_1的条件下可以得到$f(x_2|x_1)$，即可用产生随机变量的方法产生x_2；
- 以此类推，即可产生不独立的向量。

随机过程的产生

向量舍选法

- 产生一个满足某种约束条件的向量
- 典型应用：在某一多维图形内撒点
- 方法：产生一个向量，满足约束则接受，不满足则舍去
- 例子：在正六边形区域内撒1000个点

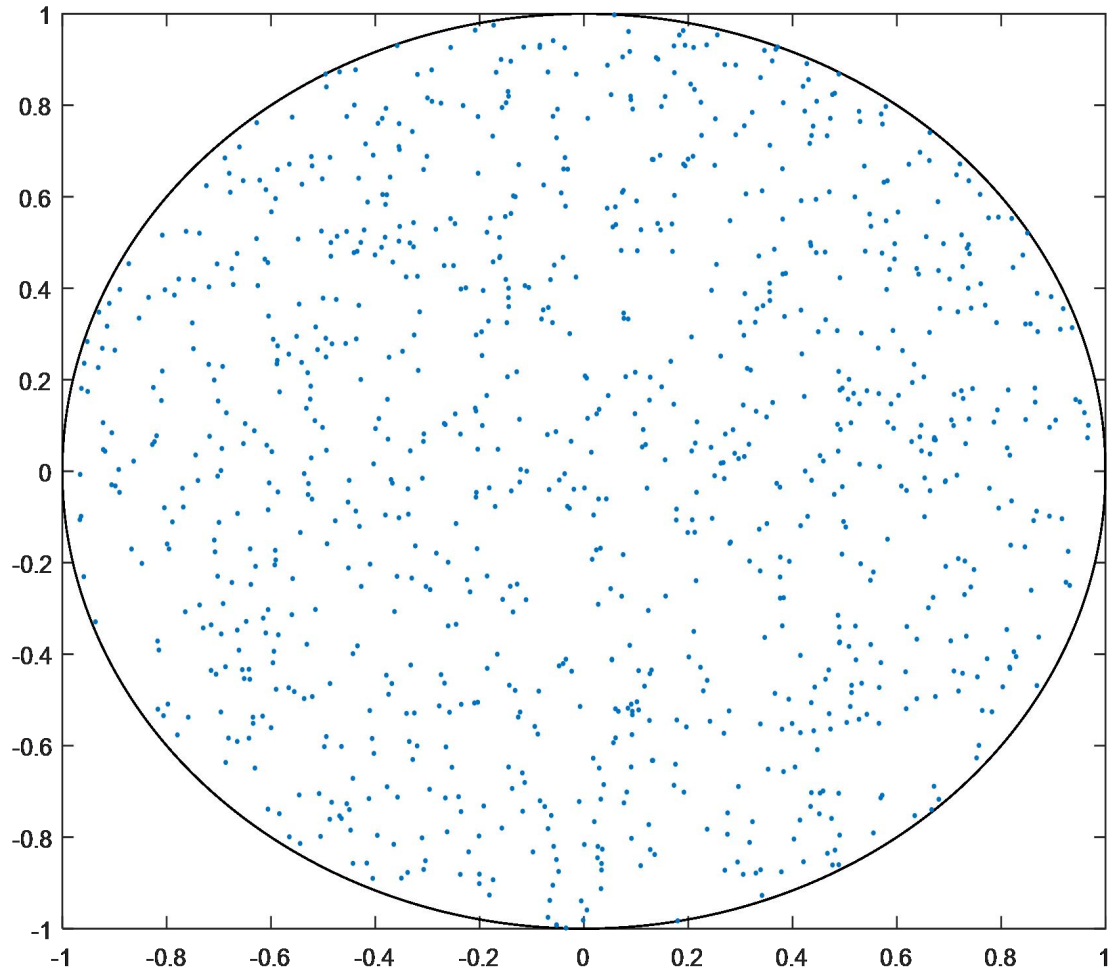
```
N=6;
theta = 0:2*pi/N:2*pi;
D=1;
plot(D*cos(theta),D*sin(theta),'g-');
point=zeros(1000,2);
n=0;
while(n<1000)
    x = 2*D*rand(1,2)-1*D;
  if(abs(x(1))+ abs(x(2))/sqrt(3))<=D&&abs(x(2))<= D*sqrt(3)/2
        n=n+1;
        point(n,:) = x;
    end
end
hold on
scatter(point(:,1),point(:,2))
```



随机过程的产生

Matlab实现单位圆内随机撒点

```
theta=0:0.001:360;
R = 1;
num = 1000;
Circle1=R*cos(theta);
Circle2=R*sin(theta);
plot(Circle1,Circle2,'k')
r=R*sqrt(rand(1,num));
seta=2*pi*rand(1,num);
x=r.*cos(seta);
y=r.*sin(seta);
hold on
plot(x,y,'.');
```



随机过程的产生

泊松过程的产生

- 泊松过程的两种解释：
  - 某一时间内的事件发生次数服从泊松分布；
  - 相邻两次事件发生的时间间隔服从指数分布
- 利用第二种解释，可以产生泊松过程
  - 产生均匀分布随机数
  - 利用反变换法产生指数分布随机数
  - 把指数分布随机数作为时间间隔，可以得到一个时间序列

- 产生λ=1的10个样本的泊松过程

$$X = -\frac{1}{\lambda}\ln U'$$

```
tm = 10;
lambda = 1;
n=1;
t=0;
while(t<tm)
   x=rand(1,1);
   if(n==1)
        tp(n)=-1/lambda*log(x);
   else
        tp(n)=tp(n-1)-1/lambda*log(x);
   end
   t=tp(n);
   n=n+1;
end
```

n为泊松分布的一个样本
时间序列tp为泊松过程的一个样本

马尔科夫链的产生

- 给定初始分布$\pi^{(0)}$和转移矩阵P
- 先根据$\pi^{(0)}$产生X_0;
- For t=1:N
- 　根据转移概率矩阵的第$X_{t\text{-}1}$行所对应的概率分布产生X_t;
- 最终生成的$X_0\ X_1\ X_2 \ldots\ X_n$即为马尔科夫链的一个样本。

THANKS FOR WATCHING