东南大學 SOUTHEAST UNIVERSITY

5 随机变量与随机过程

讲 解 人： 王俊波
E-mail: jbwang@seu.edu.cn
Phone: 13770681926
QQ:308322767

主要内容

- 随机数与通信仿真
- 随机变量的产生
  - ✓ MATLAB内置随机变量函数
  - ✓ 随机变量的生成验证
  - ✓ 随机变量的函数
  - ✓ 任意随机变量的生成
- 相关随机变量的生成
  - ✓ 两个相关随机序列的生成
  - ✓ 多个相关随机序列的生成
- 随机过程的谱密度估计

随机数与通信仿真

- 在实际通信系统中，信源发出的信息、信道中的噪声、干扰和衰落等因素都是随机的。要想在准确仿真通信系统，就需要随机数
- “真正的真随机”是一种通过物理过程来生成随机数的设备
  - ✓ 这样的设备通常是基于一些能生成低等级、统计学随机的“噪声”信号的微观现象，如热力学噪声、光电效应和量子现象
  - ✓ 这些物理过程在理论上是完全不可预测的，并且已经得到了实验的证实
  - ✓ 硬件随机数生成器通常每秒只能产生很有限的随机比特，这意味着它是相对较慢的。
- 计算机通过确定性的算法生成随机数，这样生成的序列在本质上不是随机的，只是很好的模仿了随机数的性质(如可以通过统计检验)，称为伪随机数
  - ✓ 伪随机数并不是假随机数，这里的“伪”是有规律的意思
  - ✓ 随机数是由随机种子根据一定的计算方法计算出来的数值
  - ✓ 只要计算方法一定，随机种子一定，那么产生的随机数就不会变
  - ✓ 随机种子通常来自系统时钟，确切地说，是来自计算机主板上的定时/计数器在内存中的记数值，伪随机数也是某种对应映射的产物，这个自变量是系统的时间

蒙特卡洛仿真步骤

- 构造或描述概率关系
  - ✓ 对于本身就具有随机性质的问题，主要是正确描述和模拟这个概率过程
- 从已知分布进行采样，即随机试验
  - ✓ 产生已知概率分布的随机变量（或随机向量）
  - ✓ 均匀分布是最简单的、最基本、最重要的分布
- 建立估计量
  - ✓ 确定一个随机变量，作为所要求的问题的解，我们称它为无偏估计
  - ✓ 建立各种估计量，相当于对模拟实验的结果进行考察和登记，从中得到问题的解
- 统计仿真结果，计算估计量的估计值
- 通常计算量比较大，可以使用重要度采样技术简化仿真

数字通信系统的蒙特卡洛仿真

- 数字通信过程可以建模为
  - ✓ $Y = m + G$
  - ✓ m是某个常数，G为标准正态随机变量
- 假定给某个m值，我们要估计出$Y < 0$的概率
  - ✓ $P(m) = P(Y < 0|m)$
  - ✓ 利用计算机生成标准正态随机变量
  - ✓ 测试每个Y_i，定义$X_i = \begin{cases} 0 & Y_i \geq 0 \\ 1 & Y_i < 0 \end{cases}$
  - ✓ 计算概率的估计为$\hat{P}(m) = \frac{1}{N}\sum_{i=1}^{N} X_i$

- 仿真试验次数
  - ✓ 数量级概念
  - ✓ 至少高2个数量级，可获得10%的相对精度
    - ➢ 相对精度：误码率估计器的标准差除以实际误码率
  - ✓ 10^{-6}以上BER如何仿真？
    - ➢ 如遇到误码率低于10^{-10}的情况，则需进行超过10^{12}次的试验，MC模拟完全不可行
- 从已知分布采样的准确性！！！
  - ✓ 可能造成有偏估计
  - ✓ 严重影响仿真精度
- 计算机只能按人的意志按照一定得程序产生数据，因此必然不可能是纯随机的

伪随机数

- 伪随机数是用确定性的算法计算出来自[0,1]均匀分布的随机数序列
  - ✓ 伪随机数并不真正的随机，但具有类似于随机数的统计特征，如均匀性、独立性等
  - ✓ 在计算机内，数的表示是有一定精度范围的，所以计算机能表示的数是有限多个离散的有理数，不可能有不可数无穷多个点
  - ✓ 计算机完全精确的仿真均匀分布是不可能的，实际应用也没有必要
- 实际上可以取$a_j = j/n$,若能生成$\{0,1,2,\cdots,n-1\}$上的等概率分布，则当n越大，输出a_j越接近[0,1]上均匀分布，但n有一定限度
  - ✓ 例如，$\{0,1,2,\cdots,n-1\}$中的数都可以写成0和1组成的二进制数，若能等概率地生成0、1序列，就可以等概率地输出$\{0,1,2,\cdots,n-1\}$中的随机变量。
- MATLAB中可使用 rand和 randi 函数创建伪随机数序列，使用 rng 函数控制结果的可重复性和选择伪随机数生成算法

连续均匀分布

- rand：函数返回在 0 和 1 之间均匀分布的实数
  - ✓ rand(n)：生成n行n列的0~1均匀分布的随机数；
  - ✓ rand(m,n)：生成m行n列的0~1均匀分布的随机数；
- (b-a)*rand+a：生成a~b上均匀分布的随机数；
  - ✓ 概率密度函数

$$f_X(x)=\frac{1}{b-a},\quad x\in(a,b)$$

  - ✓ 概率分布函数

$$F_X(x)=\frac{x-a}{b-a},\quad x\in(a,b)$$

  - ✓ 均值和方差

$$\mu_X=\frac{b+a}{2}$$

$$\sigma_X^2=\frac{(b-a)^2}{12}$$

离散均匀分布

- randi 函数返回离散均匀分布的整数值
  - ✓ randi(iMax)：生成1:iMax之间均匀分布的随机整数；
  - ✓ randi(iMax,m,n)：生成m行n列的1:iMax之间均匀分布的随机整数；
  - ✓ randi([iMin,iMax],m,n)：生成m行n列的iMin:iMax之间均匀分布的随机整数；
- randi([1,n])：生成一个1,…,n之间的随机整数
  - ✓ 概率质量函数

$$P(X=i)=\frac{1}{n},\quad i=1,2,\cdots,n$$

  - ✓ 均值和方差

$$\mu_X=\frac{n+1}{2}$$

$$\sigma_X^2=\frac{n^2-1}{12}$$

  - ✓ 该分布常用于生产随机信源

MATLAB中的随机数生成控制

- rng函数可控制随机数的生成
  - ✓ rng('default')或rng：将rand、randi 和 randn使用的随机数生成器的设置重置为其默认值（种子为0的梅森旋转生成器），这样会生成相同的随机数序列，控制程序的可重复性；
  - ✓ rng('shuffle') : 根据当前时间为随机数生成器提供种子，生成非可重复随机数；
  - ✓ rng(seed)：非负整数seed为种子，使rand、randi和randn生成可预测的随机数序列；
- MATLAB中的伪随机数来自一个或多个随机数流
  - ✓ 生成随机数数组的最简单方法是直接使用 rand、randn 或 randi。这些函数全部都依赖于同一均匀随机数流，称为全局流
  - ✓ 如果需要对随机数的生成进行更加高级的控制，可以使用 RandStream 类，可以创建与全局流分开使用的其他流，使用它们的 rand、randi 或 randn 方法生成随机数数组。

MATLAB中的随机数生成控制

□ 命令行输入RandStream.list，返回可用随机数生成算法

```
>> RandStream.list

可以使用以下随机数生成器算法:

dsfmt19937:      面向 SIMD 的快速梅森旋转算法，使用梅森质数 2^19937-1
mcg16807:        乘法同余生成器，乘数为 7^5，模为 2^31-1
mlfg6331_64:     乘法 lagged Fibonacci 生成器，lags 为 63 和 31，64 位(支持并行流)
mrg32k3a:        组合多递归生成器(支持并行流)
mt19937ar:       梅森旋转，使用梅森质数 2^19937-1
philox4x32_10:   执行 10 轮的 Philox 4×32 生成器(支持并行流)
shr3cong:        SHR3 移位寄存器生成器与 CONG 线性同余生成器求和
swb2712:         修正的借位减法生成器，lags 为 27 和 12
threefry4x64_20: 执行 20 轮的 Threefry 4×64 生成器(支持并行流)
```

□ 使用方法

- ✓ **获取当前全局随机流使用的算法**
  - ➢ RandStream.getGlobalStream
- ✓ **设置算法：**
  - ➢ RandStream.setGlobalStream(RandStream('mlfg6331_64'));
  - ➢ RandStream.setGlobalStream(RandStream('mlfg6331_64','seed',1));%设置种子
  - ➢ rng('default')：the Mersenne Twister with seed 0.
  - ➢ rng(seed, generator)

- 'twister': Mersenne Twister
- 'simdTwister': SIMD-oriented Fast Mersenne Twister
- 'combRecursive': Combined Multiple Recursive
- 'philox': Philox 4x32 generator with 10 rounds
- 'threefry': Threefry 4x64 generator with 20 rounds
- 'multFibonacci': Multiplicative Lagged Fibonacci
- 'v5uniform': Legacy MATLAB® 5.0 uniform generator
- 'v5normal': Legacy MATLAB 5.0 normal generator
- 'v4': Legacy MATLAB 4.0 generator

□ 注意

- ✓ **rng是简便方法，只有一个全局随机流**
- ✓ **RandStream可以设置多个随机流，便于仿真控制**

随机变量的产生——正态分布

- 正态分布也被称为高斯分布，其概率密度函数为

$$f_X(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x-\mu_X)^2}{2\sigma^2}\right],\quad x\in(-\infty,+\infty)$$

  - ✓ 该分布随机变量常用于描述白高斯随机噪声（AWGN）
- randn函数返回标准正态分布的随机数，常见的使用方法为：
  - ✓ randn(n)：生成n行n列标准正态分布的随机数；
  - ✓ randn(m,n)：生成m行n列标准正态分布的随机数；
  - ✓ randn*var+mu：生成一个均值为mu，标准方差为var的正态分布随机数
  - ✓ sqrt(var/2)*(randn + 1i*randn)：生成一个标准方差为var的复高斯分布随机数
- Q函数
  - ✓ 程序：simQfunc.m

$$Q(x)=\int_x^{\infty}\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{t^2}{2}\right]dt$$

```
function Q=qfunc(x)
Q = 0.5*erfc(x/sqrt(2));
```

```
x=-4:0.1:4;
y=qfunc(x);
plot(x,y);
title('Q函数');
xlabel('x');ylabel('y');grid on
```

小练习

- 考虑X_1~N（1,2）,X_2~N(-2,9),X_3~N(0,4), $Y=X_1+3X_2-2X_3$, 请编写程序，生成Y

```
clear
Nid=1000;
for n=1:Nid
    X1=sqrt(2)*randn+1 ;
    X2=sqrt(9)*randn-2 ;
    X3=sqrt(4)*randn+0 ;
    Y(n)= X1+3*X2-2*X3;
end
mY=sum(Y)/Nid
vY=sum(Y.^2)/Nid-mY^2
```

```
mY =

  -4.7849


vY =

  96.9412
```

伯努利（Bernoulli）分布

- 伯努利分布也被称为0-1分布或两点分布，输出只有两个值的离散随机分布
  - ✓ 伯努利分布主要描述某次随机试验中某事件是否发生的随机现象
- 伯努利分布的概率质量函数为

$$P(x) = p^x (1-p)^{1-x} = \begin{cases} p & x=1 \\ 1-p & x=0 \end{cases}$$

  - ✓ p为事件发生的概率
  - ✓ 伯努利分布的均值和方差分别为

$$\mu_X = p$$

$$\sigma_X^2 = p(1-p)$$

- 伯努利分布可用于二进制数据生成，也可用于描述通信接收二进制数据流中的随机错误位置

◻ MATLAB没有直接提供伯努利分布随机数生成的函数，这里自定义生成函数

```
function X = bernoulliRV(L,p)
%生成参数为 p 的伯努利分布随机数
% 输入参数
% L: 生成的序列长度; p: 参数
% 输出参数
% X: 随机序列
U = rand(1,L);%生成长度为 L 的在(0,1)上均匀分布的随机序列
X = (U<p);
end
```

二项（Binomial）分布

- 二项分布常用于描述n次独立随机试验中某事件发生的次数，而每次试验中该事件发生的概率为

$$P\left(X=x\right)=\begin{pmatrix} n \\ x \end{pmatrix} p^{x}\left(1-p\right)^{n-x}, \quad x=0,1,\cdots,n$$

  - ✓ 二项分布的概率质量函数为 $\begin{pmatrix} n \\ x \end{pmatrix}=\dfrac{n!}{x!\left(n-x\right)!}$
  - ✓ 二项分布的均值和方差分别为 $\mu_X=np$

$$\sigma_X^2=np\left(1-p\right)$$

  - ✓ 二项分布可用于描述通信接收二进制数据流中的错误总数
- 在MATLAB中，二项分布随机变量可使用函数binornd(n,p)或函数random ('Binomial',n,p)生成

指数（Exponential）分布

- 指数分布的概率密度函数为

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0$$

  - ✓ 参数$\lambda > 0$

$$F_X(x) = 1 - e^{-\lambda x}, \quad x > 0$$

  - ✓ 均值和方差

$$\mu_X = \lambda^{-1}$$

$$\sigma_X^2 = \lambda^{-2}$$

  - ✓ 指数分布常用于描述业务的服务时间或信道衰落影响

- 在MATLAB中，指数分布随机变量可使用函数exprnd(λ^{-1})或函数random('Exponential',λ^{-1})生成

泊松（Poisson）分布

- 泊松分布的概率密度函数为

$$P(X=k)=\frac{\lambda^k}{k!}e^{-\lambda},\quad k=0,1,2,\cdots$$

  - ✓ 参数$\lambda>0$
  - ✓ 均值和方差

$$\mu_X=\lambda$$
$$\sigma_X^2=\lambda$$

  - ✓ 泊松分布常用于描述业务的到达过程，如给定时间内服务到达的个数，也可用于一些特殊信道下光子、电子等粒子在给定时间内的发射数目

- 在MATLAB中，泊松分布随机变量可使用函数poissrnd(λ)或函数random('Poisson',λ)生成

对数正态分布

- 对数正态分布的概率密度函数为

$$f_X(x|\mu,\sigma)=\frac{1}{x\sigma\sqrt{2\pi}}\exp\left[-\frac{(\log x-\mu)^2}{2\sigma^2}\right],\quad x\in(0,+\infty)$$

  - ✓ μ和σ分别为对数正态分布的参数
  - ✓ 均值和方差

$$\mu_X=\exp\left(\mu+\sigma^2/2\right)$$

$$\sigma_X^2=\exp\left(2\mu+\sigma^2\right)\left(\exp\left(\sigma^2\right)-1\right)$$

  - ✓ 对数正态分布是随机变量的对数值服从正态分布
  - ✓ 对数正态随机变量常用于描述无线通信中阴影衰落的影响
- 在MATLAB中，对数正态分布随机变量可使用函数lognrnd(μ,σ)或函数random ('Lognormal',μ,σ)生成

瑞利（Rayleigh）分布

- 瑞利分布的概率密度函数为

$$f_X(x) = \frac{x}{b^2}\exp\left[-\frac{x^2}{2b^2}\right], \quad x \in (0, +\infty)$$

  - ✓ 均值和方差

$$\mu_X = b\sqrt{\pi/2}$$

$$\sigma_X^2 = b^2(2 - \pi/2)$$

  - ✓ 瑞利分布常用于无线通信中多径衰落对接收信号包络的影响

- 在MATLAB中，瑞利分布随机变量可使用函数raylrnd(b)或函数random('Rayleigh',b)生成

莱斯（Rician）分布

- 莱斯分布的概率密度函数为

$$f_X(x)=I_0\left(\frac{x\rho}{\sigma^2}\right)\frac{x}{\sigma^2}\exp\left[-\frac{x^2+\rho^2}{2\sigma^2}\right],\quad x\in(0,+\infty)$$

  - ✓ I_0是第一类零阶修正贝塞尔函数， $\rho>0$为非中心参数， σ为大于零参数
  - ✓ 莱斯分布常被用于描述带有直达径的无线多径信道，如空地无线信道
  - ✓ 直达径与散射径上信号能量之比定义为K因子

$$K=\frac{\rho^2}{2\sigma^2}$$

  - ✓ $\rho=0$,莱斯分布变成瑞利分布
- 在MATLAB中，莱斯分布随机变量可使用函数random('Rician',ρ,σ)生成

Nakagami分布

- Nakagami分布的概率密度函数为

$$f_X(x)=\frac{2m^m}{\Gamma(m)\omega^m}x^{2m-1}\exp\left[-\frac{m}{\omega}x^2\right],\quad x\in(0,+\infty)$$

  - ✓ m是形状参数，ω是接收功率
- Nakagami分布常用于描述多径衰落的影响，该分布与实验测试数据有较好的吻合，改变m的值，Nakagami衰落还可以转变为多种衰落模型
  - ✓ $m=0$：Nakagami分布等效为瑞利分布
  - ✓ $m\longrightarrow\infty$：无衰落
- 在MATLAB中，Nakagami分布随机变量可使用函数random('Nakagami',m,ω) 生成。

随机变量的生成验证

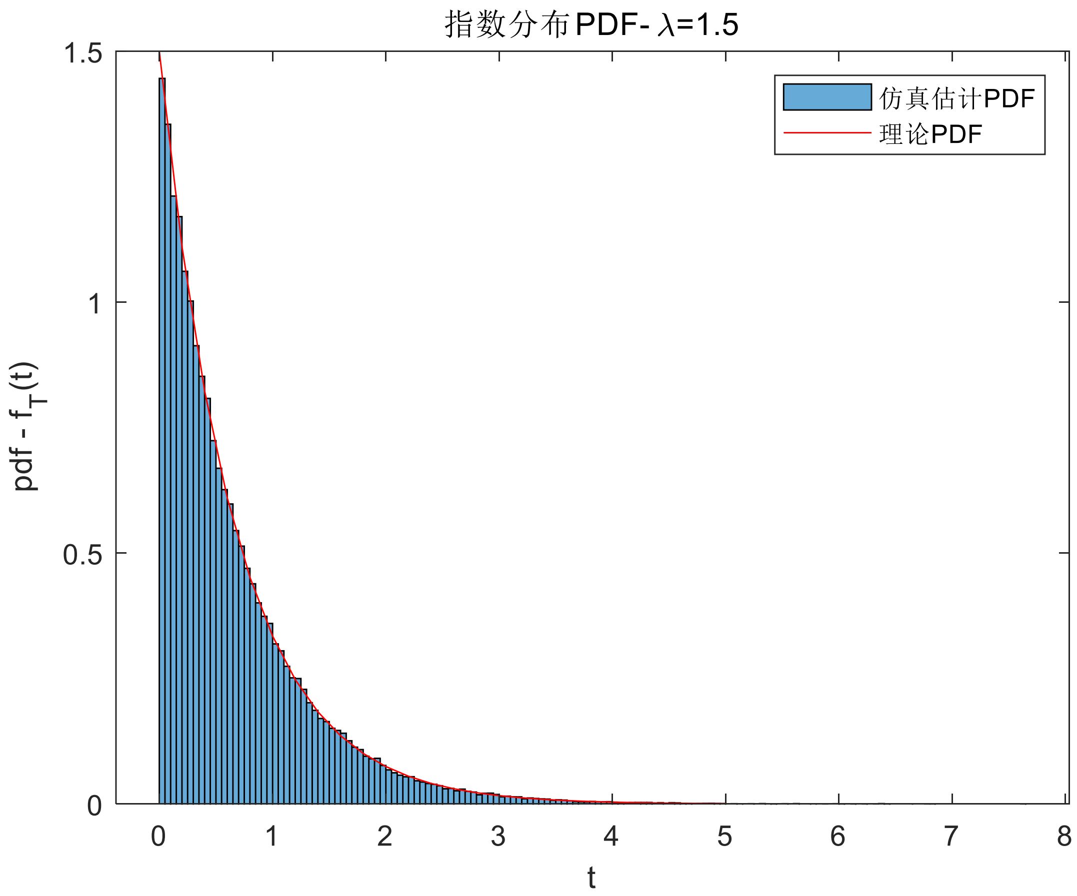
- 在通信系统仿真中，按概率分布或参数要求生成随机变量至关重要。当随机变量生成后，必须有方法能验证生成随机变量是符合设定的分布特性，通常有两种方法
  - ✓ 估计随机变量的均值和方差
  - ✓ 估计随机变量的PDF或PMF
- 将估计出的均值或方差，以及PDF或PMF与设定的分布特性进行比较，若两者之间误差在允许范围内，则生成的随机变量符合要求
- 在MATLAB中，函数mean和var可用于计算生成序列的均值和标准方差；函数histogram可用计算并绘制PDF

示例：随机变量的生成验证

□ 例 生成指数分布的随机变量，并验证其分布特性

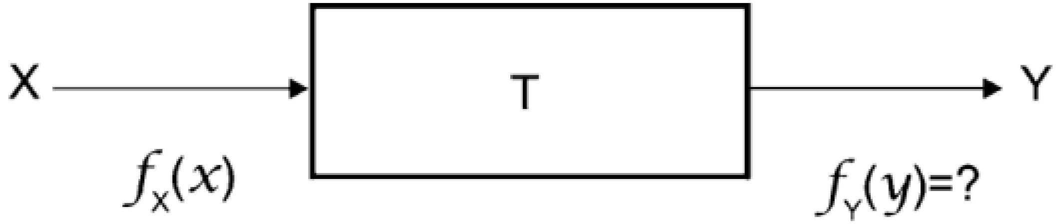
✓ 程序：simRVVerfication.m

```
% 随机变量的验证
close all; clearvars;

lambda=1.5;
L=1e5;
T = exprnd(1/lambda,L,1);%simulating the RV

t_range=0:0.2:5;
T_pdf = lambda*exp(-lambda*t_range); %理论 PDF
histogram(T,'Normalization','pdf'); hold on;%从数据中估计 PDF
plot(t_range,T_pdf,'r');                  % 绘制理论 PDF
xlabel('t');ylabel('pdf - f_T(t)');
title(['指数分布 PDF- \lambda=',num2str(lambda)]);
legend('仿真估计 PDF','理论 PDF');
fprintf('理论: 均值[%2.2d]方差[%2.2d]\n',1/lambda,1/lambda^2)
fprintf('估计: 均值[%2.2d]方差[%2.2d]\n',mean(T),var(T))
```



随机变量的变换

- 随机信号经常通过改变其特性的装置，如果变换前的随机变量的随机特性已知，变换后随机变量的随机特性如何获知？



- 假设变换前后的随机变量分别表示为X和Y，且有如下的函数关系

$$Y = g(X)$$

  - ✓ X可能是连续的随机变量，也可能是离散的随机变量
  - ✓ 函数g可能是单调函数，也可能是非单调函数

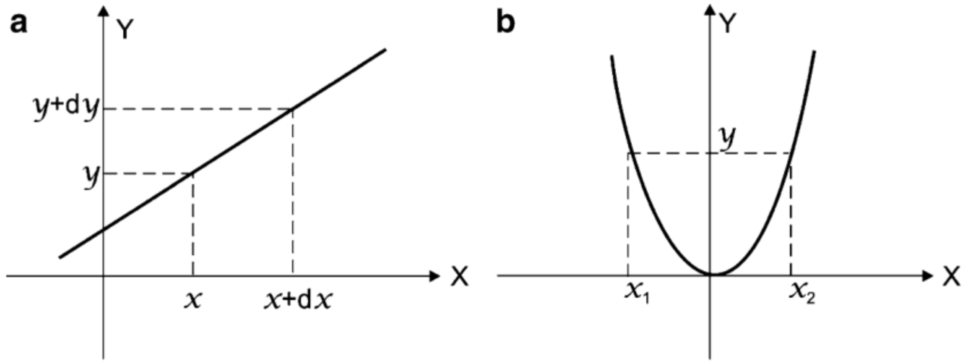
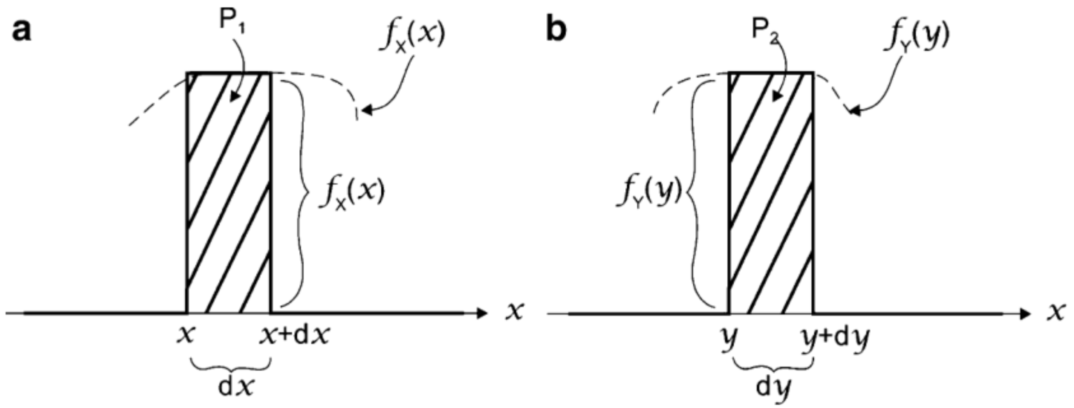


Fig. 2.29 (**a**) Monotone and (**b**) nonmonotone transformation

情况一：X是连续随机变量，g是单调函数



- 由上图可以得到

$$P_1 = P\{x < X \leq x + \mathrm{d}x\} = P_2 = P\{y < Y \leq y + \mathrm{d}y\}. \Rightarrow f_X(x)\mathrm{d}x = f_Y(y)\mathrm{d}y. \Rightarrow f_Y(y) = \frac{f_X(x)}{\mathrm{d}y/\mathrm{d}x}.$$

- 导出的dy/dx既可以是正的，也可以是负的，但密度函数只能是正的，dy/dx加上绝对值符号

$$f_Y(y) = \frac{f_X(x)}{|\mathrm{d}y/\mathrm{d}x|}$$

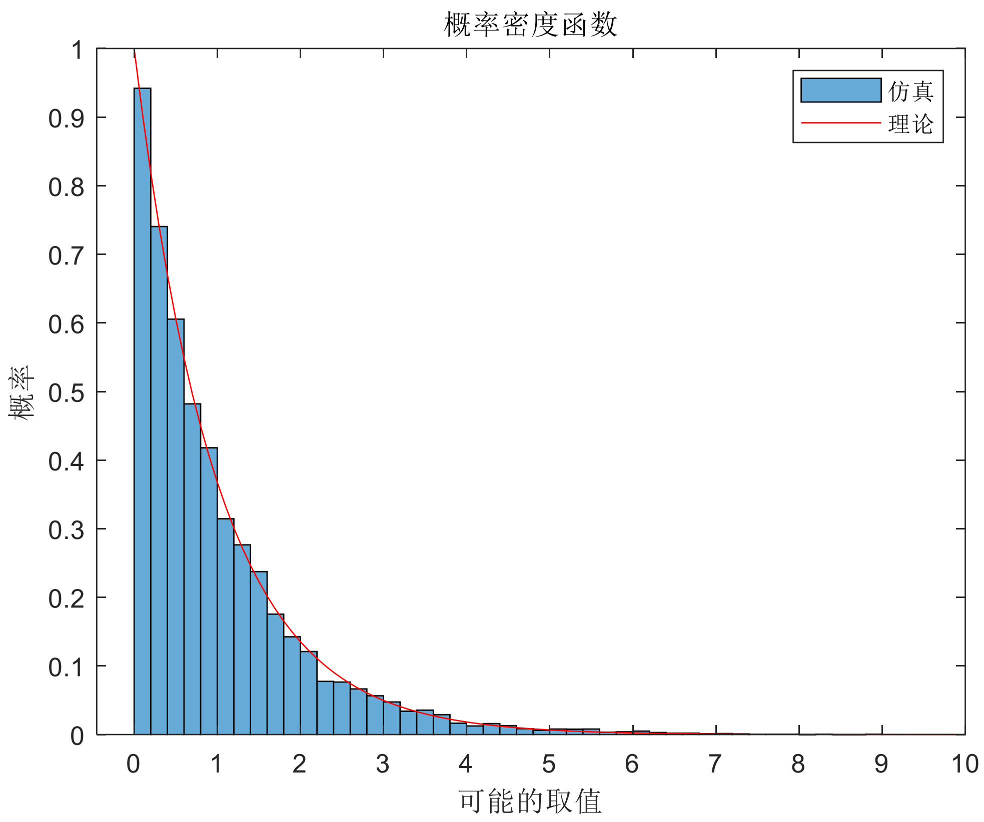
- 将等式右边的x换成y的函数$g^{-1}(y)$

$$f_Y(y) = \left|\frac{f_X(x)}{|\mathrm{d}y/\mathrm{d}x|}\right|_{x=g^{-1}(y)}$$

随机变量的变换示例一

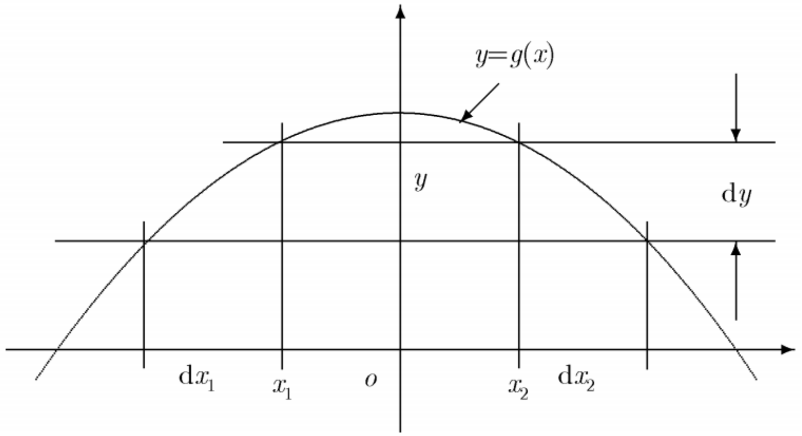
- 例：X是$[0,1]$上的均匀随机变量，$Y=-lnX$，求随机变量Y的概率密度函数
  - ✓ $f_X(x)=1 \;\; x\in[0,1]$
  - ✓ $\left|\left|\frac{dy}{dx}\right|\right|_{x=g^{-1}(y)}=\left|\left|-\frac{1}{x}\right|\right|_{x=e^{-y}}=e^{y}$
  - ✓ $f_Y(y)=\left.\frac{f_X(x)}{\left|\frac{dy}{dx}\right|}\right|_{x=g^{-1}(y)}=\frac{1}{e^{y}}=e^{-y} \;\; y\in[0,+\infty]$
  - ✓ **程序：simRVTrans1.m**

```
% 随机变量的函数(情况 1)
close all; clearvars;

N=1e4;
x=rand(1,N);
y=-log(x);
histogram(y,'Normalization','pdf');

Y = 0:0.1:10;
fY_theory =exp(-Y); %理论 PDF
hold on; plot(Y,fY_theory,'r-');        %绘制理论 PDF
title('概率密度函数');legend('仿真','理论');
xlabel('可能的取值');ylabel('概率');
```



情况二： X是连续随机变量，g是非单调函数

- 对非单调变换$Y = g(X)$存在输入随机变量的多个值只对应于输出随机变量的一个值



$$P\{y < Y \le y + \mathrm{d}y\} = P\{x_1 < X \le x_1 + \mathrm{d}x\} + P\{x_2 - \mathrm{d}x < X \le x_2\}.$$

$$\Rightarrow \quad f_Y(y)\mathrm{d}y = f_X(x_1)\mathrm{d}x + f_X(x_2)\mathrm{d}x$$

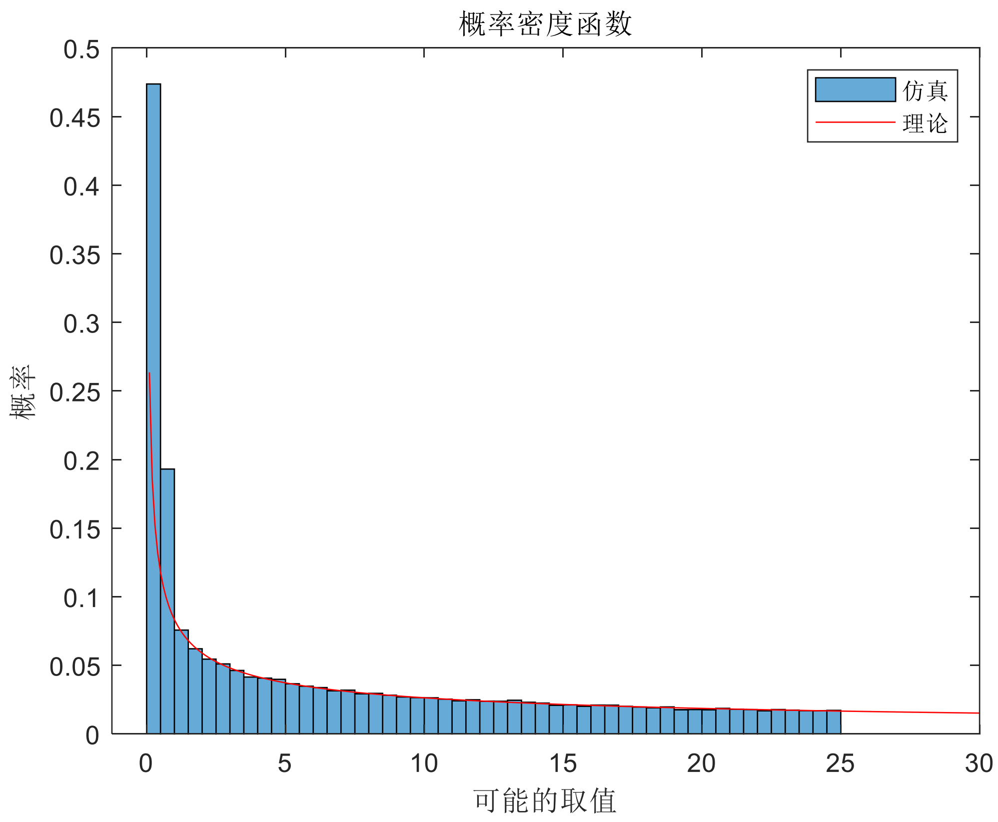
$$\Rightarrow \quad f_Y(y) = \left.\frac{f_X(x_1)}{\left|\frac{\mathrm{d}y}{\mathrm{d}x}(x_1)\right|} + \frac{f_X(x_2)}{\left|\frac{\mathrm{d}y}{\mathrm{d}x}(x_2)\right|}\right|_{\substack{x_1 = g^{-1}(y) \\ x_2 = g^{-1}(y)}}$$

$$\Rightarrow \quad f_Y(y) = \left.\sum_{i=1}^{N} \frac{f_X(x_i)}{\left|\frac{\mathrm{d}y}{\mathrm{d}x}(x_i)\right|}\right|_{x_i = g^{-1}(y), i=1,\ldots,N}$$

随机变量的变换示例二

- **例：**X**是**$[-1,5]$**上的均匀随机变量，**$Y=X^2$**, 求随机变量Y的概率密度函数**
  - ✓ $f_X(x)=1/6 \quad x\in[-1,5]$
  - ✓ $x\in[-1,0]$**时，** $\left|\left|\frac{dy}{dx}\right|\right|_{x=g^{-1}(y)}=-2x|_{x=-\sqrt{y}}=2\sqrt{y}$
  - ✓ $x\in[0,5]$**时，** $\left|\left|\frac{dy}{dx}\right|\right|_{x=g^{-1}(y)}=2x|_{x=\sqrt{y}}=2\sqrt{y}$
  - ✓ $f_Y(y)=\frac{f_X(x)}{\left|\frac{dy}{dx}\right|}\Bigg|_{x=g^{-1}(y)}=\frac{\frac{1}{6}}{2\sqrt{y}}=\frac{1}{12\sqrt{y}} \quad y\in[0,+\infty]$
  - ✓ **simRVTrans2.m**

```
% 随机变量的函数(情况 2)
close all; clearvars;

N=100000;
x=6*rand(1,N)-1;
y=x.^2;
histogram(y,'Normalization','pdf');

Y = 0:0.1:30;
fY_theory =1./(12*sqrt(Y));        %理论 PDF
hold on; plot(Y,fY_theory,'r-');  %绘制理论 PDF
title('概率密度函数');legend('仿真','理论');
xlabel('可能的取值');ylabel('概率');
```



情况三：X是离散随机变量

- 给定离散随机变量

$$f_X(x) = \sum_i P\{x_i\}\delta(x - x_i),$$

$$F_X(x) = \sum_i P\{x_i\}u(x - x_i).$$

- 变换函数为$Y = g(X)$
  - ✓ 若g是单调函数，X和Y之间有唯一的对应关系 $y_i = g(x_i)$

$$\Rightarrow \quad P\{y_i\} = P\{x_i\} \quad \Rightarrow \quad f_Y(y) = \sum_i P\{y_i\}\delta(y - y_i),$$

$$F_Y(y) = \sum_i P\{y_i\}u(y - y_i),$$

  - ✓ 若g是非单调函数，随机变量X的N个值对应于随机变量Y的一个值

$$Y = y_i' \quad \text{if} \quad X = x_1,\ \text{or}\ X = x_2,\ \text{or} \ldots X = x_N. \quad \Rightarrow \quad P\{y_i'\} = \sum_{j=1}^{N} P\{x_j\}.$$

任意随机变量的生成

- 随机数的种类无穷无尽
  - ✓ 满足随机数的性质即可
  - ✓ 有名称的随机数非常有限
    - ➢ 均匀分布，高斯分布，指数分布，瑞丽分布，莱斯分布...
- 方法一：解析变换法（反演法、逆分布函数法）
  - ✓ 对任意随机变量x，设其概率密度函数(PDF)为$f(x)$，其概率分布函数(CDF)为$F(x)=\int_{-\infty}^{x} f(x)dx$，只生成均匀分布的另一随机变量$U$，用$F(x)$的反函数可计算得到$x=F^{-1}(U)$，则$x$一定服从$f(x)$分布
    - ➢ 生成[0,1]上的均匀分布随机变量U
    - ➢ $x=F^{-1}(\theta)$，则随机变量x为所需随机数
  - ✓ 简单且高效，是第一选择
  - ✓ 需要给出概率分布函数反函数的解析表达式

示例：解析变换法

- 生成一个指数分布随机变量

$$f_X(x) = \lambda e^{-\lambda x}, \qquad x \geqslant 0$$

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

- 令$U = F(x)$,可以得到

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

- 由于$1-U$还是$[0,1]$上均匀分布的随机变量，上面式子可以改写为

$$X = -\ln(U)/\lambda$$

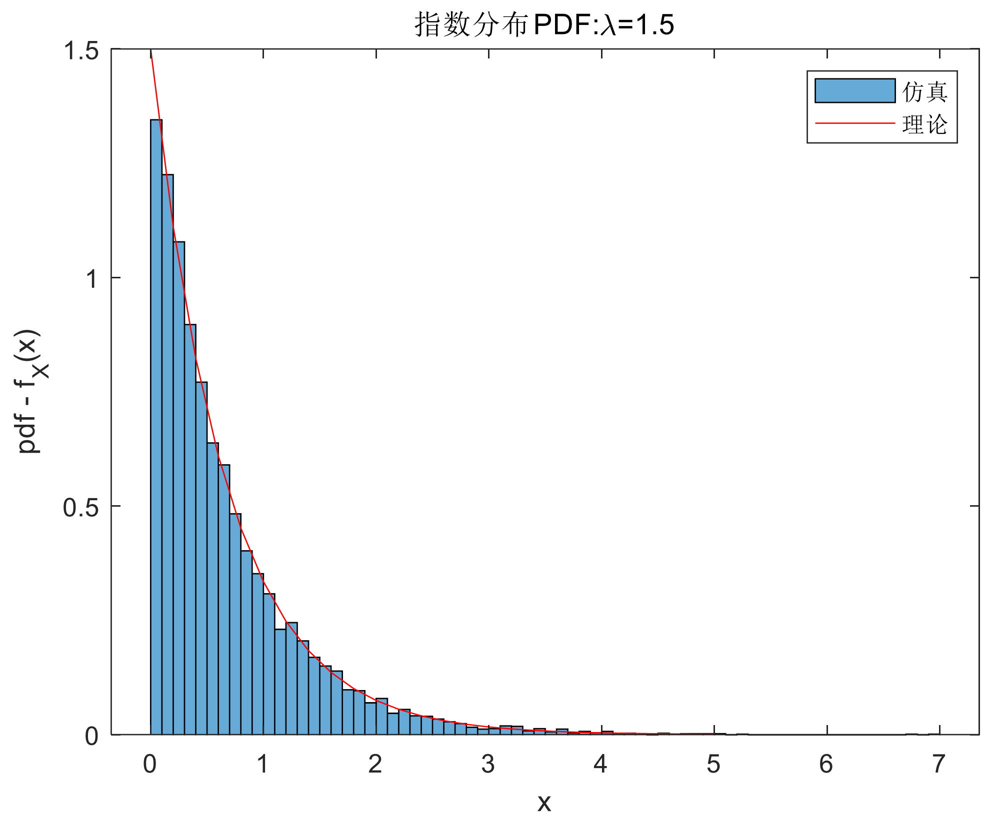
```
function x = expRV(lambda,L)
%变换法生成指数分布随机变量
%lambda - 指数分布的参数, L - 数据长度
u = rand(1,L); %均匀分布随机变量
x = -1/lambda*(log(1-u));
end
```

示例：解析变换法

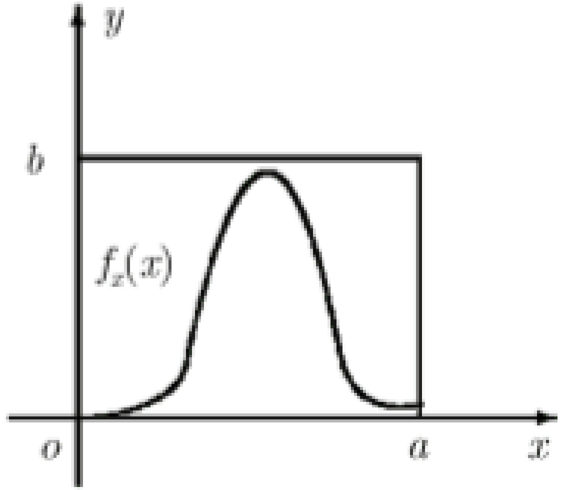
例：验证指数分布随机变量

✓ exExpRv.m

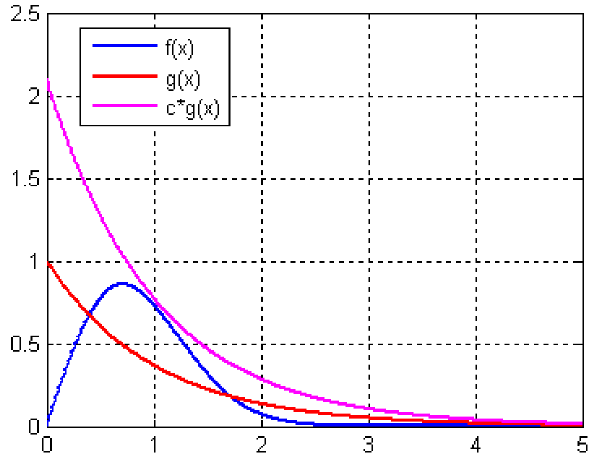
```
lambda=1.5;
L=1e4;
x = expRV(lambda,L);%生成随机变量
t_range=0:0.2:5;
T_pdf = lambda*exp(-lambda*t_range); %理论 PDF
histogram(x,'Normalization','pdf'); hold on;%从数据中估计 PDF
plot(t_range,T_pdf,'r'); %绘制仿真估计 PDF
xlabel('x');ylabel('pdf - f_X(x)');
title(['指数分布 PDF:\lambda=',num2str(lambda)]);
legend('仿真','理论');

function x = expRV(lambda,L)
%变换法生成指数分布随机变量
%lambda - 指数分布的参数, L - 数据长度
u = rand(1,L); %均匀分布随机变量
x = -1/lambda*(log(1-u));
end
```



舍选法



- 舍选法又叫接受/拒绝方法、拒绝法
- 情况1：定义域有限
  - ✓ 不易求概率分布函数的反函数，应用面广
  - ✓ $f(x)$仅在$[x_{min}, x_{max}]$上非零，$f(x)$的值域仅在$[y_{min}, y_{max}]$上非零
    - ➢ 生成一个$[x_{min}, x_{max}]$上均匀分布随机数X
    - ➢ 生成一个$[y_{min}, y_{max}]$上均匀分布随机数Y
    - ➢ 若$Y \leq f(X)$，则返回X，否则重复第一步
- 情况2：定义域无限
  - ✓ 假定我们可以生成概率密度函数为$g(x)$的随机变量，假定存在常数c满足 $f(x)/g(x) \leq c \ \forall x$
    - ➢ 生成均匀分布的随机数u
    - ➢ 生成具有密度$g(x)$的随机数v
    - ➢ 若$cu \leq f(v)/g(v)$,则接受v为随机数输出；反之，返回重新生成



示例：舍选法

- 用舍选法生成如下概率分布的随机变量

$$f(x)=xe^{-\frac{x^2}{2}}\ x>0$$

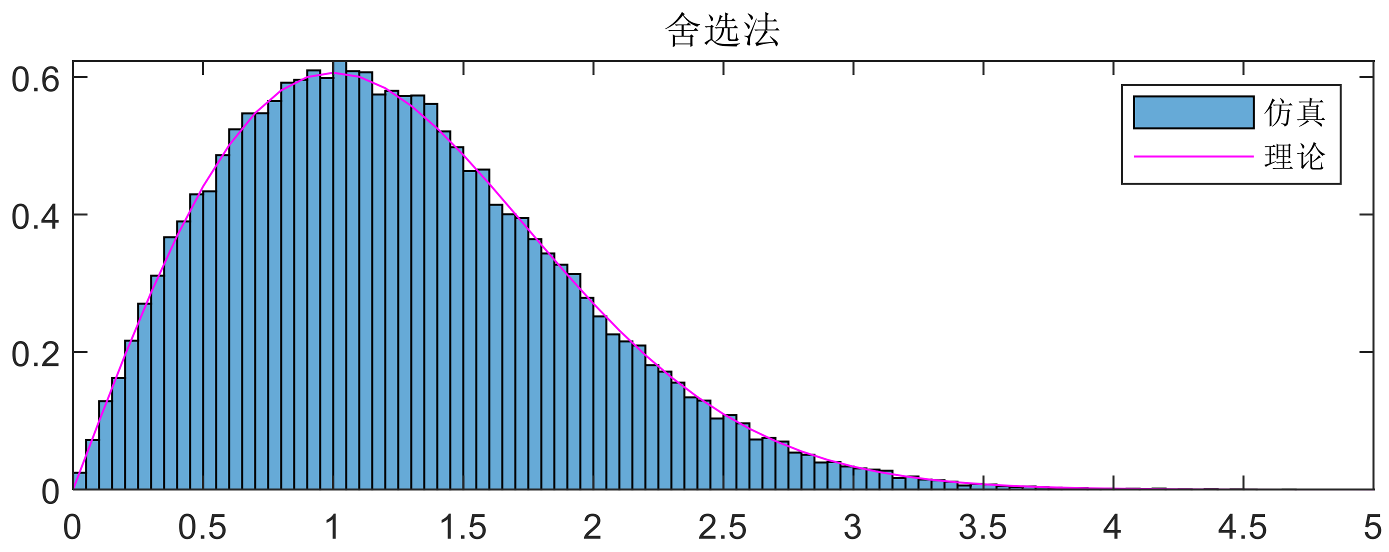
- ✓ 注意：是参数为1的瑞丽(Rayleigh)分布
- ✓ simARMethod.m

```
n=1e5;%生成数量
f = @(x)x.*exp(-(x.^2)/2);
g = @(x)exp(-x);
grnd = @()exprnd(1);
X = ARMethod(f,g,grnd,2.2,n,1);

t=0:0.1:10;
t_pdf=f(t);
Y = raylrnd(1,n,1);%MATLAB 内置的瑞利分布生成随机变量
subplot(211);histogram(X,'Normalization','pdf');
hold on; plot(t,t_pdf,'m-');hold off;title('舍选法')
xlim([0,5]);legend('仿真','理论');
subplot(212);histogram(Y,'Normalization','pdf');
hold on; plot(t,t_pdf,'m-');hold off;title('MATLAB 内置函数')
xlim([0,5]);legend('仿真','理论');
```

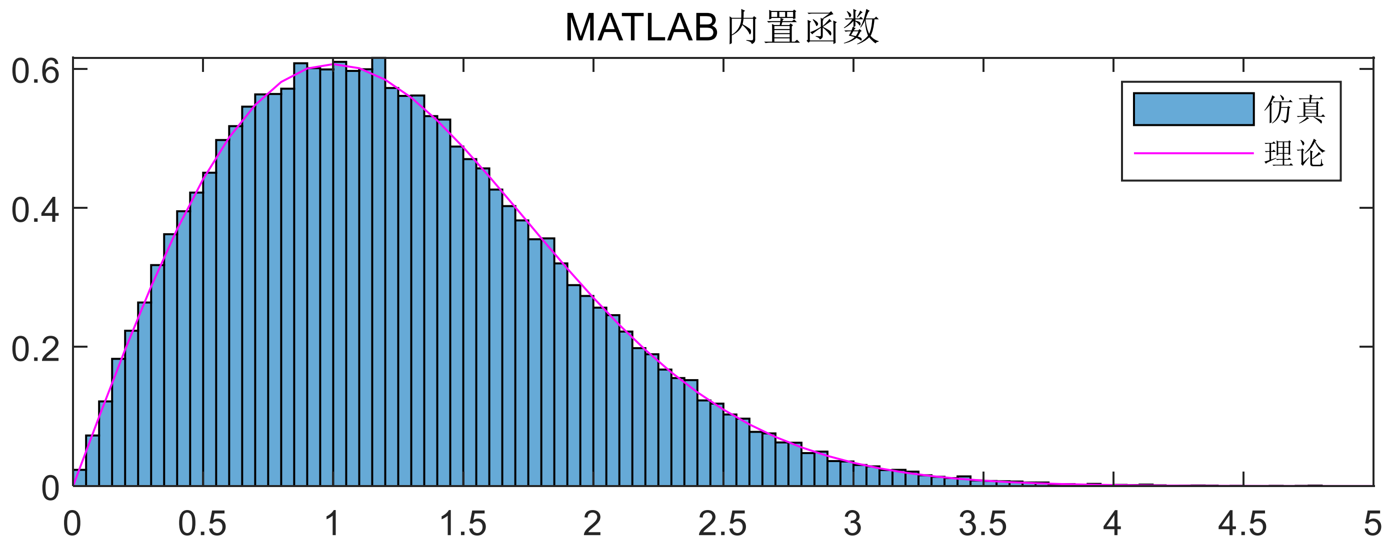
```
function X = ARMethod(f,g,grnd,c,m,n)
%舍选法实现
% f: f(x)函数定义; g: g(x)函数定义; grnd: 概率密度函数为 g(x)的随机数生成器
% m, n: 随机数维度
    X = zeros(m,n); % 预分配存储空间
    for i = 1:m*n
        accept = false;
        while accept == false
            u = rand(); v = grnd();
            if c*u <= f(v)/g(v)
               X(i) = v;
               accept = true;
            end
        end
    end
end
```

- ➢ 生成均匀分布的随机数u
- ➢ 生成具有密度$g(x)$的随机数v
- ➢ 若$cu \le f(v)/g(v)$,则接受v为随机数输出；反之，返回重新生成

示例：舍选法



舍选法
仿真
理论
0
0.2
0.4
0.6
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5



MATLAB内置函数
仿真
理论
0
0.2
0.4
0.6
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

相关随机变量的生成

- 相关系数是研究两个随机序列之间线性相关程度的量
- 给定两个序列X和Y，他们的相关系数定义为

$$\rho(X,Y)=\frac{\mathrm{E}\left\{\left(X-\mu_X\right)\left(Y-\mu_Y\right)\right\}}{\sigma_X\sigma_Y}$$

  - ✓ μ_X 和μ_Y分别是序列X和Y的均值， σ_X 和σ_Y 分别是序列X和Y的标准方差
- 两个序列X和Y的相关矩阵

$$C=\begin{bmatrix}\rho(X,X) & \rho(X,Y)\\ \rho(Y,X) & \rho(Y,Y)\end{bmatrix}$$

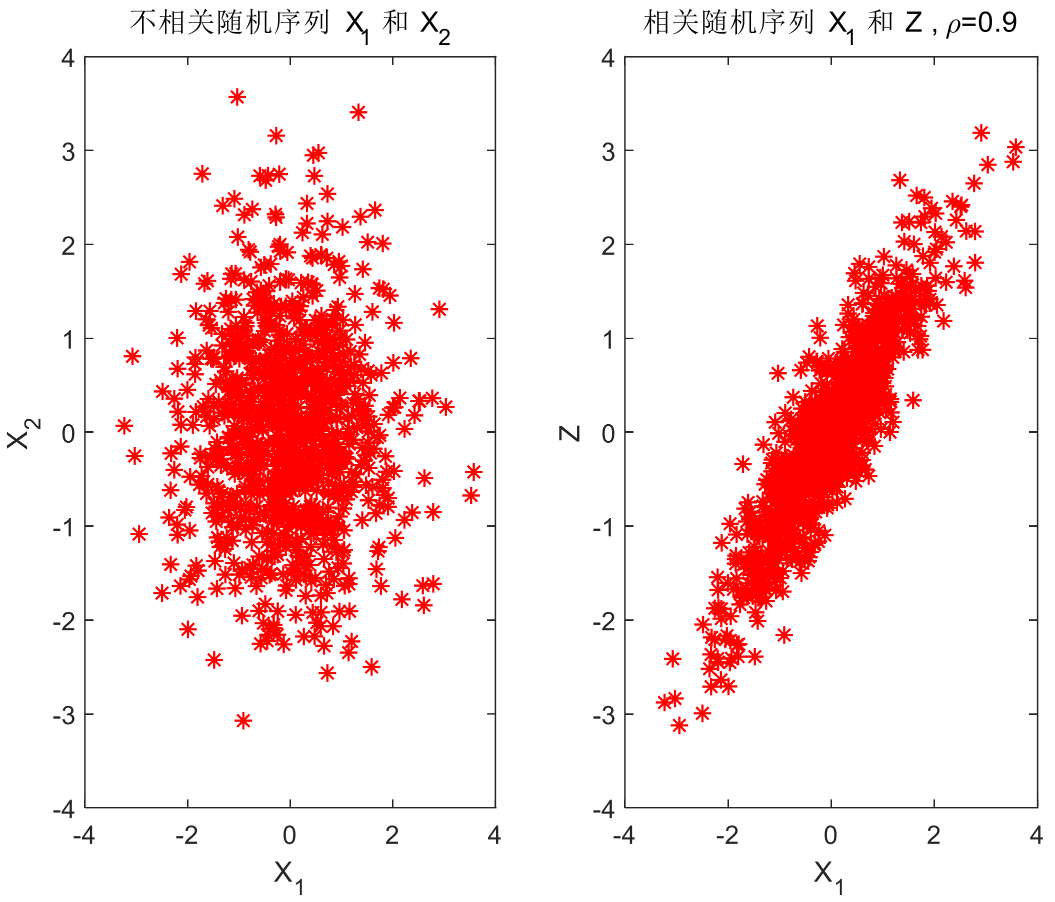
  - ✓ 由于序列与自己完全相关，可改写为

$$C=\begin{bmatrix}1 & \rho(X,Y)\\ \rho(Y,X) & 1\end{bmatrix}$$

生成两个相关随机序列

- 给定相关系数ρ，生成两个相关随机序列
  - ✓ 第一步：按概率分布生成两个不相关的随机序列
  - ✓ 第二步：生成所需的相关序列 $Z=\rho X_1+\sqrt{1-\rho^2}X_2$
  - ✓ 得到的序列Z相对于X_1具有相关性ρ
- 例：生成两个相关高斯随机序列
  - ✓ simCorrelatedRvTwoSeq.m

```
N=1000;          %序列长度
x1=randn(1,N); %正态分布随机序列1
x2=randn(1,N); %正态分布随机序列2
subplot(1,2,1); plot(x1,x2,'r*');
title('不相关随机序列 X_1 和 X_2');
xlabel('X_1'); ylabel('X_2')
axis([-4,4,-4,4])

rho=0.9;
z=rho*x1+sqrt(1-rho^2)*x2;
subplot(1,2,2); plot(x1,z,'r*');
title(['相关随机序列 X_1 和 Z , \rho=',num2str(rho)]);
xlabel('X_1'); ylabel('Z');
axis([-4,4,-4,4])
```



多个相关随机变量序列的生成

- 给定一个相关矩阵，生成多个相关随机变量序列，使用Cholesky分解
  - ✓ Cholesky分解法又称平方根法
  - ✓ 相关矩阵为对称正定阵，表示成具有形式 $C = U^T U = LL^T$ 的分解，其中U和L分别为上三角阵和下三角阵
  - ✓ 要求矩阵的所有特征值必须大于零，故分解的下三角的对角元也是大于零的
- 相关矩阵：N*N矩阵，定义N个变量之间的相关性
  - ✓ 对角线元素（变量与自身的相关性）总是等于1
  - ✓ 第i行j例或第j行i列元素为第i个和第j个变量的相关系数
  - ✓ 例：给定三个变量X、Y和Z
    - ➢ X和Y的相关系数为0.5
    - ➢ X和Z的相关系数为0.3
    - ➢ Y和Z的相关系数为0.2

$$
\begin{array}{cc}
 & \begin{array}{ccc} X & Y & Z \end{array} \\
C = & \begin{bmatrix} 1 & 0.5 & 0.3 \\ 0.5 & 1 & 0.2 \\ 0.3 & 0.2 & 1 \end{bmatrix}
\end{array}
\begin{array}{c} X \\ Y \\ Z \end{array}
$$

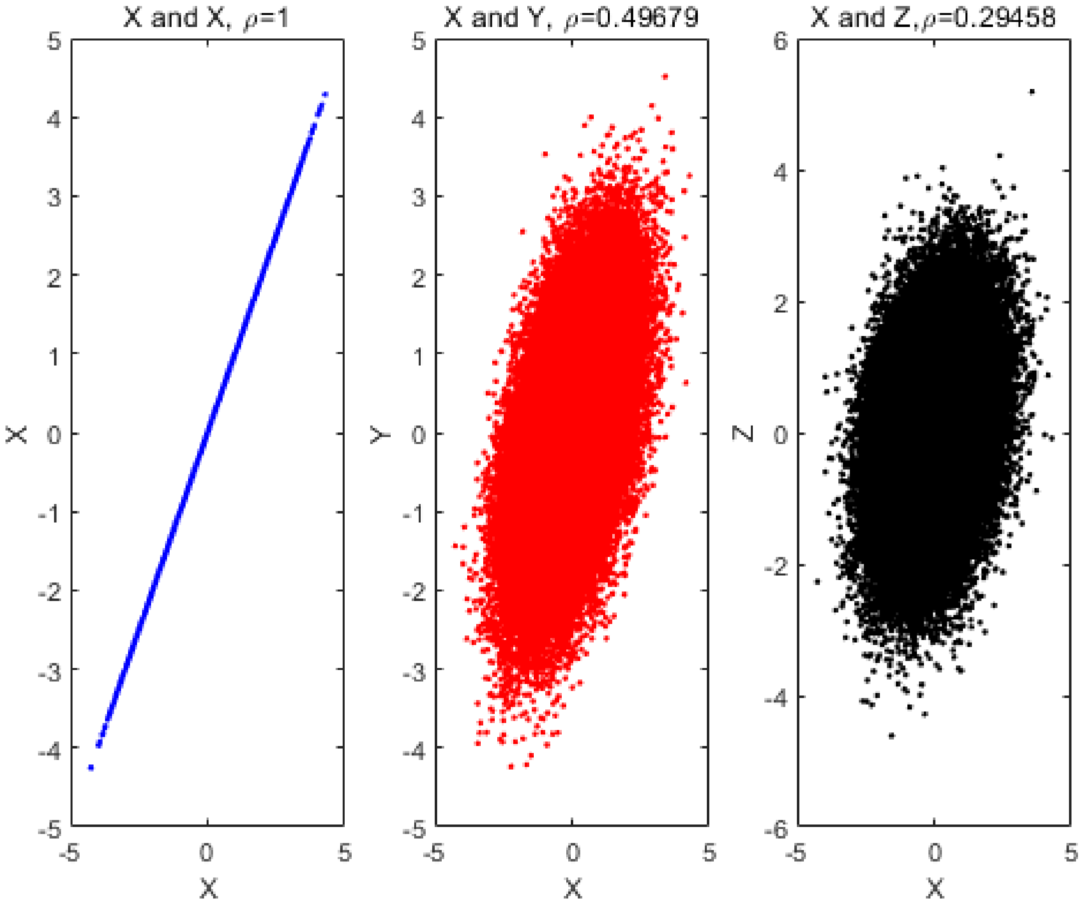
- 相关序列生成：将不相关随机变量与上三角矩阵相乘 $R_c = RU$
  - ✓ R是不相关随机变量
  - ✓ 与上三角或下三角相乘都可以

示例：多个相关随机变量序列的生成

□ 程序：simCorrelatedRvMultiple.m

```
C=[  1  0.5 0.3;
     0.5  1  0.2;
     0.3 0.2  1 ;]; %相关矩阵
U=chol(C);          %Cholesky 分解

N=1e5;          %序列长度
R=randn(N,3); %生成 3 条不相干随机序列
Rc=R*U;         %生成 3 条相关随机序列 X,Y,Z

```

```
%验证
%计算生成的相关序列之间的相关矩阵
coeffMatrixXX=corrcoef(Rc(:,1),Rc(:,1));
coeffMatrixXY=corrcoef(Rc(:,1),Rc(:,2));
coeffMatrixXZ=corrcoef(Rc(:,1),Rc(:,3));

%从相关矩阵中提取出相关系数
coeffXX=coeffMatrixXX(1,2); %序列 X 与序列 X 的相关系数
coeffXY=coeffMatrixXY(1,2); %序列 X 与序列 Y 的相关系数
coeffXZ=coeffMatrixXZ(1,2); %序列 X 与序列 Z 的相关系数
```



随机过程的谱密度

- 当信号的能量集中在一个有限时间区间的时候，尤其是总能量是有限的，就可以计算能量谱密度
- 对某一信号$x(t)$，其频域表示为$X(f)=\int_{-\infty}^{\infty} x(t)e^{-2\pi ift}\mathrm{d}t$
  - ✓ 功率谱密度可定义为

$$S_x(f)=\left|X(f)\right|^2$$

- 对一随机过程$X(t)$, 则其在$[-T,T]$ 上的平均功率为

$$P_T=\lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T}\left|x(t)\right|^2\mathrm{d}t.$$

  - ✓ 功率谱密度可定义为

$$S_X(f)=\lim_{T\to\infty}\frac{1}{2T}\left|\int_{-T}^{T}x(t)e^{-2\pi ift}\mathrm{d}t\right|^2.$$

- 对于宽平稳随机过程，其功率谱密度是其自相关函数的傅里叶变换
- 随机过程$X(t)$的自相关函数定义

$$R_X(t_1,t_2)=E\{X(t_1)X(t_2)\}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x_1x_2 f_X(x_1,x_2;t_1,t_2)\mathrm{d}x_1\mathrm{d}x_2$$

  - ✓ $f_X(x_1,x_2;t_1,t_2)$是随机过程$X(t)$的二维概率密度
- 当随机过程$X(t)$为宽平稳随机过程 $\tau=t_2-t_1$

$$R_X(\tau)=R_X(t_1,t_2)$$

- 得到随机过程$X(t)$的功率谱密度

$$S_X(f)=\int_{-\infty}^{\infty}R_X(\tau)e^{-2\pi i f\tau}\mathrm{d}\tau$$

功率谱密度估计——直接法

- 直接法也称为周期图法，它是直接对有限个样本数据进行傅里叶变换，利用得到的结果计算得到功率谱。样本数据越长，直接法获得的分辨率越高。
- 假定有限长随机信号序列为，则其功率谱密度可估计为

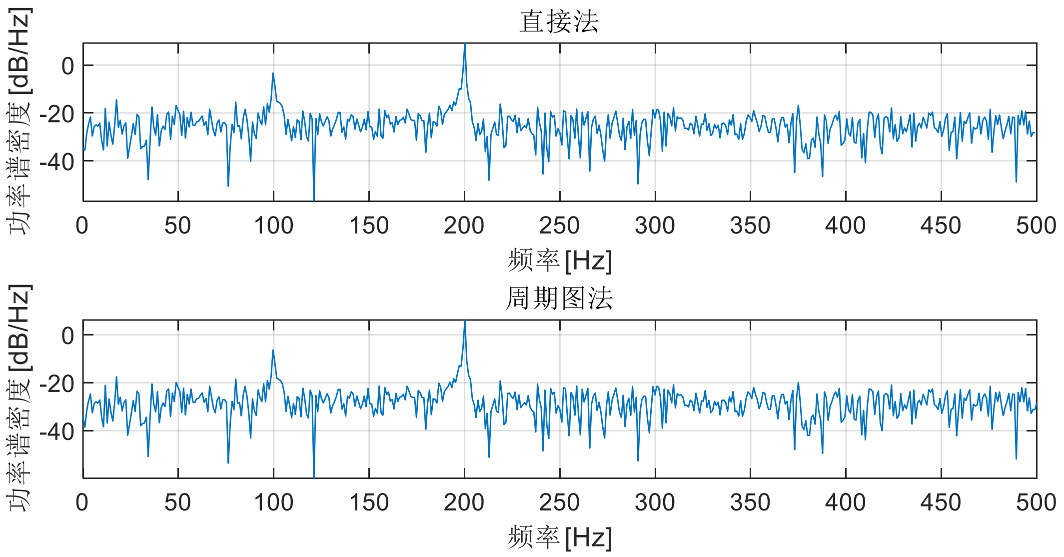
$$S_X(k)=\left|X_k\right|^2 \quad k=0,1,\cdots,N-1$$

- MATLAB提供了periodogram、pwelch等多个函数实现随机过程的功率谱密度估计
- 例：估计随机过程 $x(t)=\cos(2\pi f_1 t)+3\cos(2\pi f_2 t)+n$ 的功率谱密度，其中f1=50,f2=100，n是单位方差的零均值高斯噪声。

示例：功率谱估计——直接法

◻ 程序：simPSD.m

```
Fs = 1000;
nfft = 1024;  %fft采样点数

%产生序列
n = 0:1/Fs:1;
xn = cos(2*pi*100*n) + 3*cos(2*pi*200*n)+(randn(size(n)));
subplot(2,1,1);plot(xn);title('加噪信号');xlim([0 1000]);grid on
% 随机信号的频谱
subplot(2,1,2)
[Xf, f] = SpectrumViewer(xn, Fs, 'onesided');
Y = abs(Xf);
plot(f,Y,'linewidth',2);
grid; xlabel('频率/Hz');  ylabel('幅值')

figure
%FFT直接平方
Y2=Y.^2;
subplot(3,1,1);plot(f,10*log10(Y2));title('直接法');grid on
xlabel('频率[Hz]');ylabel('功率谱密度[dB/Hz]')

%周期图法
window = boxcar(length(xn));  %矩形窗
[psd1,f] = periodogram(xn,window,nfft,Fs);
subplot(3,1,2);plot(f,10*log10(psd1));title('周期图法');grid on
xlabel('频率[Hz]');ylabel('功率谱密度[dB/Hz]')
```



功率谱密度估计——间接法

- 间接法则是先对有限个样本数据进行自相关估计，再进行傅里叶变换，最后得到功率谱
  - ✓ 一个平稳随机过程$X(t)$在频域中是用它的功率谱$S_x(f)$来表征的，功率谱谁随机过程自相关函数的傅立叶变换

$$S_x(f) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j2\pi f\tau}\, dt$$

  - ✓ 反过来，一个平稳随机过程的自相关函数可由其功率谱的傅立叶逆变换得到

$$R_x(\tau) = \int_{-\infty}^{\infty} S_x(f) e^{j2\pi f\tau}\, df$$

  - ✓ 序列的自相关

$$R(k) = \frac{E[(X_i - \mu_i)(X_{i+k} - \mu_{i+k})]}{\sigma^2}$$

E .... 期望值。

X_i .... 在t(i)时的随机变量值。

μ_i .... 在t(i)时的预期值。

X_{i+k} .... 在t(i+k)时的随机变量值。

μ_{i+k} .... 在t(i+k)时的预期值。

σ^2 .... 为方差。

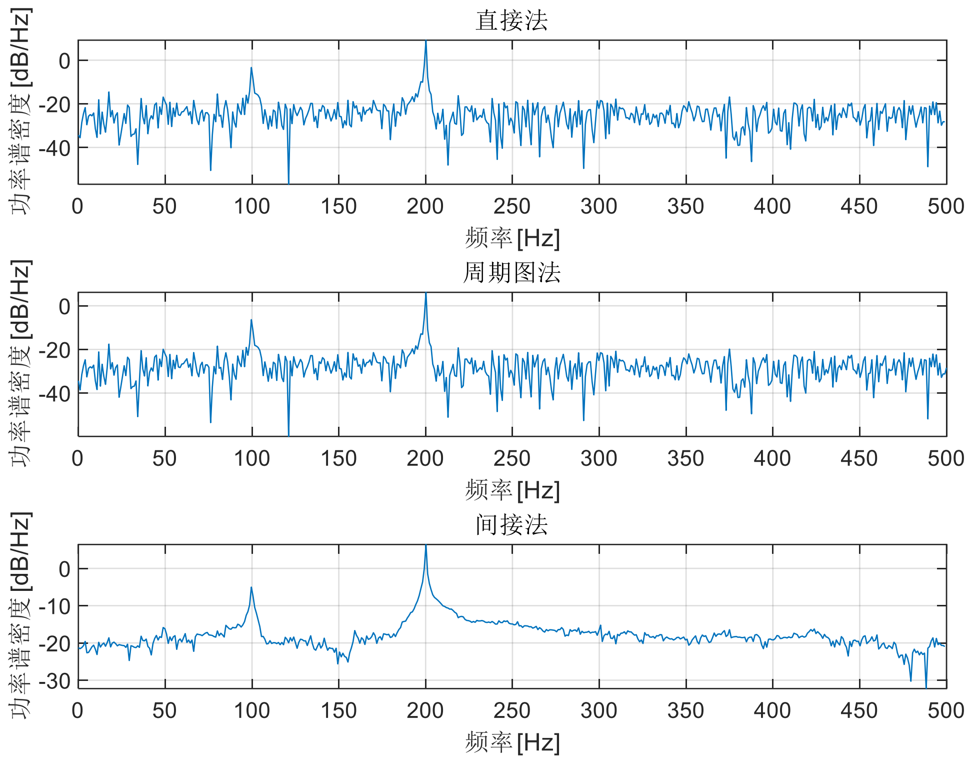
$$\hat{R}_x(m) = \frac{1}{N-m}\sum_{n=1}^{N-m} X_n X_{n+m}, \qquad m = 0, 1, \ldots, M$$

$$= \frac{1}{N-|m|}\sum_{n=|m|}^{N} X_n X_{n+m}, \qquad m = -1, -2, \ldots, -M$$

示例：功率谱密度估计——间接法

▫ 程序：simPSD.m

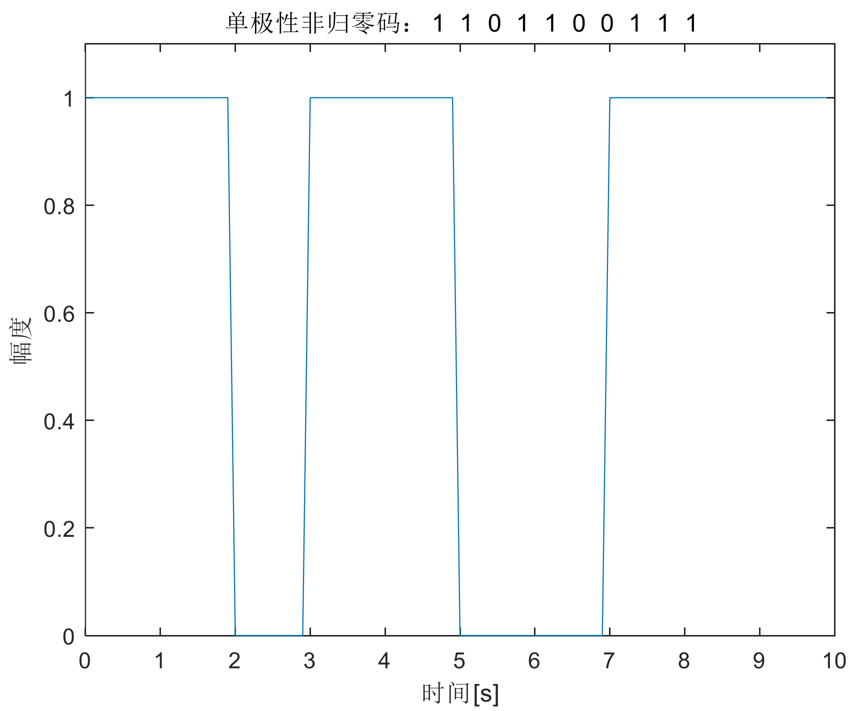
```
%周期图法
window = boxcar(length(xn));  %矩形窗
[psd1,f] = periodogram(xn,window,nfft,Fs);
subplot(3,1,2);plot(f,10*log10(psd1));title('周期图法');grid on
xlabel('频率[Hz]');ylabel('功率谱密度[dB/Hz]')

%自相关法
cxn = xcorr(xn,'unbiased');  %计算自相关函数
CXk = fft(cxn,nfft)/nfft;
psd2 = abs(CXk)*2;           %单边谱，*2
index = 0:round(nfft/2-1);
k = index*Fs/nfft;
psd2 = (psd2(index+1));
subplot(313)
plot(k,10*log10(psd2));title('间接法');grid on
xlabel('频率[Hz]');ylabel('功率谱密度[dB/Hz]')
```



▫ 注意：方法2的噪声明显小于方法1

课后作业

- 考虑X_1~N（2,2）,X_2~N(-2,5)， $Y=X_1+2X_2$ 请编写程序，生成1000个Y，并验证随机数的性质符合要求（均值、方差、直方图等）
- 绘制单极性非归零码的波形与功率谱（理论和仿真数值计算结果）
  - ✓ 程序框架：hw3.m
  - ✓ 注意：Ts=1



$$P_s(f)=\frac{1}{4}\delta(f)+\frac{T_s}{4}Sa^2\left(\frac{2\pi fT_s}{2}\right)$$

$$g(t)=\begin{cases}1 & |t|\le\frac{T_s}{2}\\ 0 & \text{其他}\end{cases}\leftrightarrow G(f)=T_sSa\left(\frac{2\pi fT_s}{2}\right)$$

Sa函数可以用sinc函数实现，注意sinc的定义

$$\operatorname{sinc} t=\begin{cases}\frac{\sin \pi t}{\pi t} & t\neq 0,\\ 1 & t=0.\end{cases}$$

有问题，随便问！