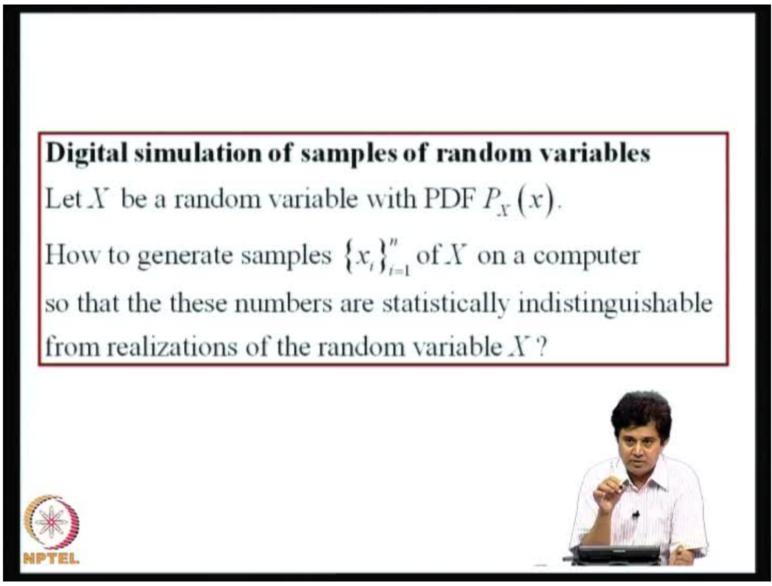
Stochastic Structural Dynamics

Prof. Dr. C. S. Manohar
Department of Civil Engineering
Indian Institute of Science, Bangalore

Module No. # 07
Lecture No. # 27
Monte Carlo Simulation Approach-3

We have been discussing Monte Carlo simulation methods for studying randomly exacted structural systems. So, we reviewed certain principles of statistics in the previous lectures, and today, what we will do is, we will discuss how to simulate random variables on a computer.

(Refer Slide Time: 00:36)



Digital simulation of samples of random variables

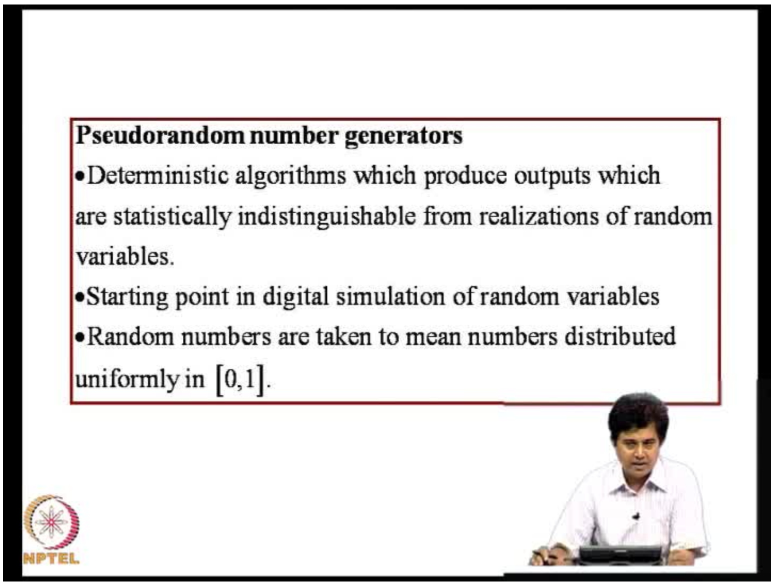
Let X be a random variable with PDF $P_X(x)$.

How to generate samples $\{x_i\}_{i=1}^n$ of X on a computer so that the these numbers are statistically indistinguishable from realizations of the random variable X ?

.

So, let X be a random variable with probability distribution function P x of x. The problem on hand is, how to generate samples of, samples x i say 1 to n of X on a computer, so that these numbers are statistically indistinguishable from realizations of the random variable X. So, that is the problem; this we have to do on a computer, not through a physical device.

(Refer Slide Time: 01:05)

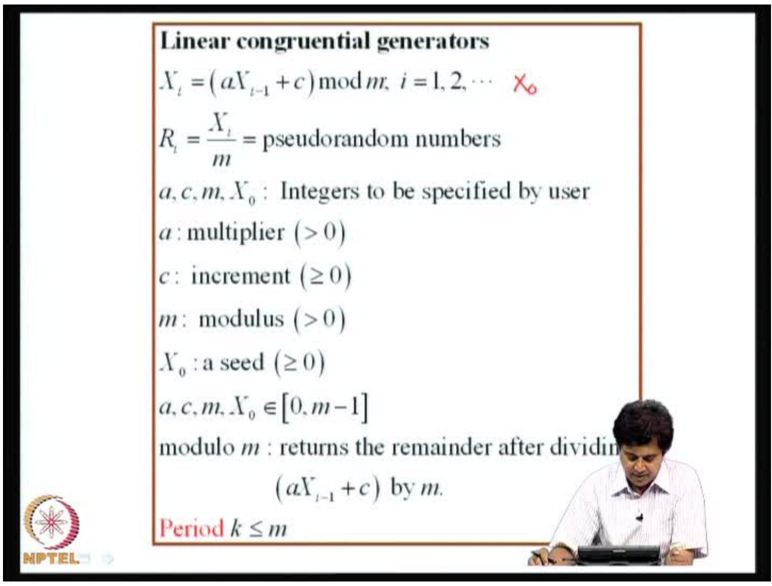


Pseudorandom number generators

- Deterministic algorithms which produce outputs which are statistically indistinguishable from realizations of random variables.
- Starting point in digital simulation of random variables
- Random numbers are taken to mean numbers distributed uniformly in $[0,1]$.

The starting point for this discussion, what are known as pseudorandom generators, these are deterministic algorithms, which produce outputs, which are statistically indistinguishable from realizations of random variables. These random variables are typically, taken to be uniformly distributed between 0 and 1. So, pseudorandom number refers to random numbers, which are uniformly distributed between 0 and 1. So, this is the starting point in digital simulation of random variables; random numbers are taken to mean numbers, which are distributed uniformly 0 to 1.

(Refer Slide Time: 01:57)



Linear congruential generators

$X_i = (aX_{i-1} + c) \bmod m,\ i = 1, 2, \cdots$ X_0

$R_i = \frac{X_i}{m}$ = pseudorandom numbers

a, c, m, X_0 : Integers to be specified by user

a : multiplier (>0)

c : increment (≥ 0)

m : modulus (>0)

X_0 : a seed (≥ 0)

$a, c, m, X_0 \in [0, m-1]$

modulo m : returns the remainder after dividi

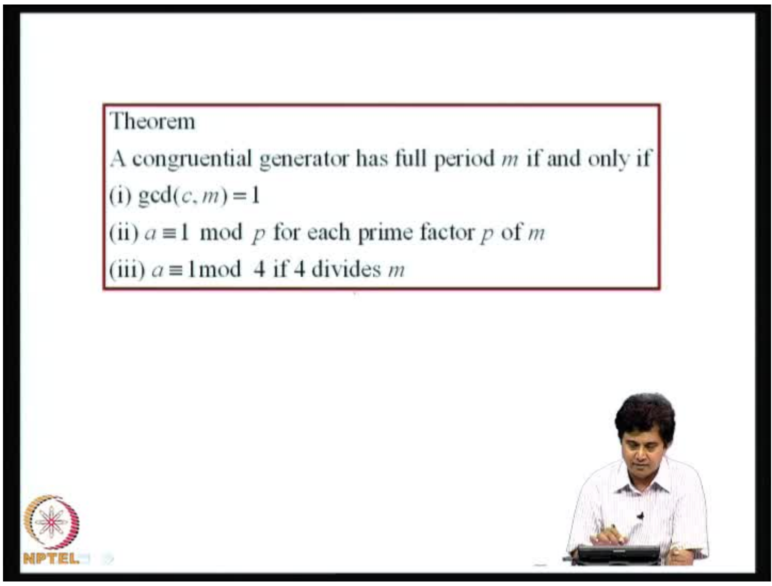
$(aX_{i-1} + c)$ by m.

Period $k \leq m$

So, the first task on hand is how to simulate realizations of uniformly distributed random numbers, which are uniformly distributed in 0 to 1. There are certain deterministic algorithms, which will produce these random numbers and these are known as linear congruential generators. So, the algorithm here is, X i is (a X i minus 1 plus c) modulo m. So, I will explain what these terms are - X i's are generated through this map and R i, which are the pseudorandom numbers, are defined as X i divided by m; these constants here: a, c, m and X naught, it should start with, the, i equal to 0, which is X naught. X naught has to be specified, they are integers to be specified by the user. So, a is known as a multiplier, which is positive; c is known as increment, which is non-negative; m is a modulus, positive; X naught is a seed, greater than or equal to 0.

This operation modulo m means, modulo m returns the remainder after dividing this number (a X i minus c plus c) by m; the remainder is returned as a, the output. Now, these, this is a map whose period is less than or equal to m, that is, for each of this map, there is a quantity known as period, and after, suppose i ranks from 1, 2, 3 and up to the period, then it repeats; the same sequence repeats. This period, further, this algorithm is k and k can be less than or equal to m. So, ideally, we want k to be equal to m, to be, to make best use of this generator and there are theorems in number theory, which states, under what condition a congruential generator has full period m, and it has something to do with choice of a, c and m.

(Refer Slide Time: 03:53)



Theorem

A congruential generator has full period m if and only if

(i) $\gcd(c,m)=1$

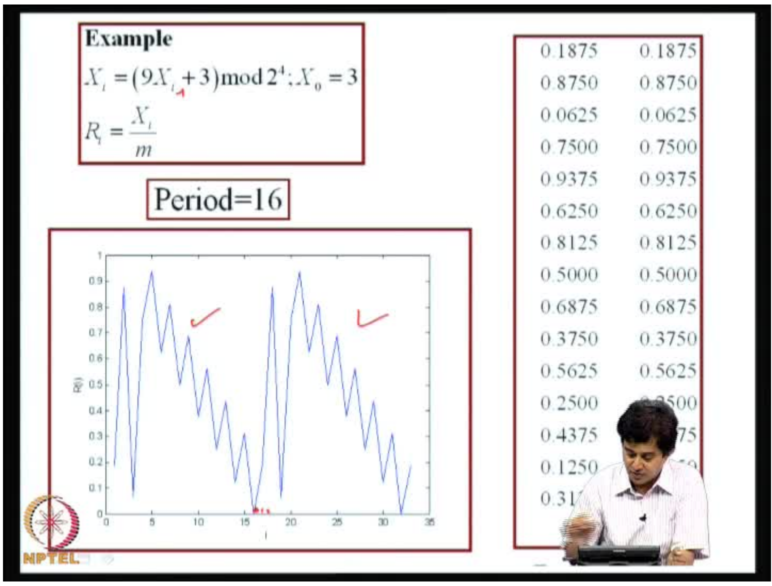
(ii) $a \equiv 1 \mod p$ for each prime factor p of m

(iii) $a \equiv 1 \mod 4$ if 4 divides m

.

So, the gcd of c and m must be equal to 1, and a equal to 1 mod p for each prime factor p of m, and a 1 mod 4 if 4 divides m. So, if these conditions are satisfied, the period will be m. So, we need not worry about the theoretical background behind this because it is a specialist area, which belongs to the domain of computer science and number theory. So, we will use these tools as starting point in our discussion.

(Refer Slide Time: 04:41)



Example

$X_i = (9X_{i-1}+3)\bmod 2^4 : X_0 = 3$

$R_i = \dfrac{X_i}{m}$

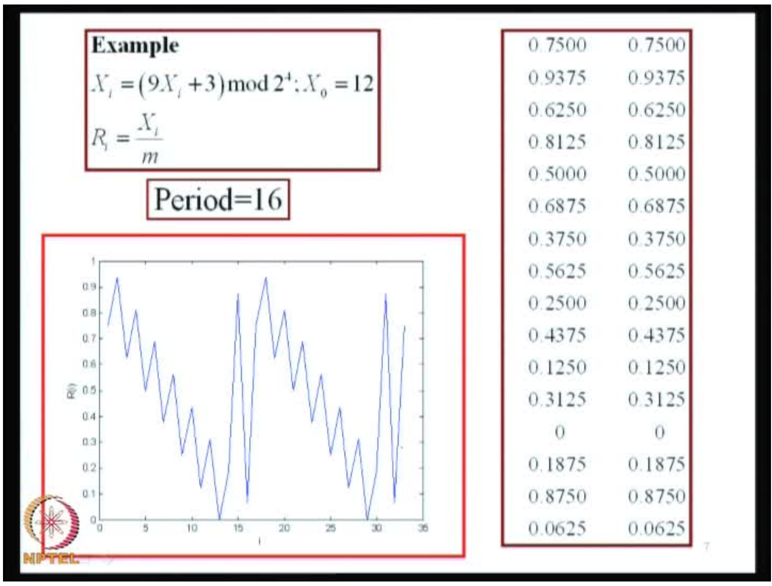
Period=16

0.1875	0.1875
0.8750	0.8750
0.0625	0.0625
0.7500	0.7500
0.9375	0.9375
0.6250	0.6250
0.8125	0.8125
0.5000	0.5000
0.6875	0.6875
0.3750	0.3750
0.5625	0.5625
0.2500	[illegible]500
0.4375	[illegible]75
0.1250	[illegible]
0.31	[illegible]

.

But to understand what we are going to use, it is useful to simulate the output of this map for a few choices of the parameters. So, let us take X i is (9 X i minus 1 plus 3) modulo 2 to the power of 4. So, the period here can be 16 and indeed, it turns out, that after every 16th number, the signal repeats. So, this part of the signal simply, repeats after every 16th number; this is the 16th number and this part repeats. So, there are 16 random numbers that we can generate through this; not very useful for practical applications.

(Refer Slide Time: 05:21)



Example

$X_i = (9X_i + 3) \bmod 2^4 ; X_0 = 12$

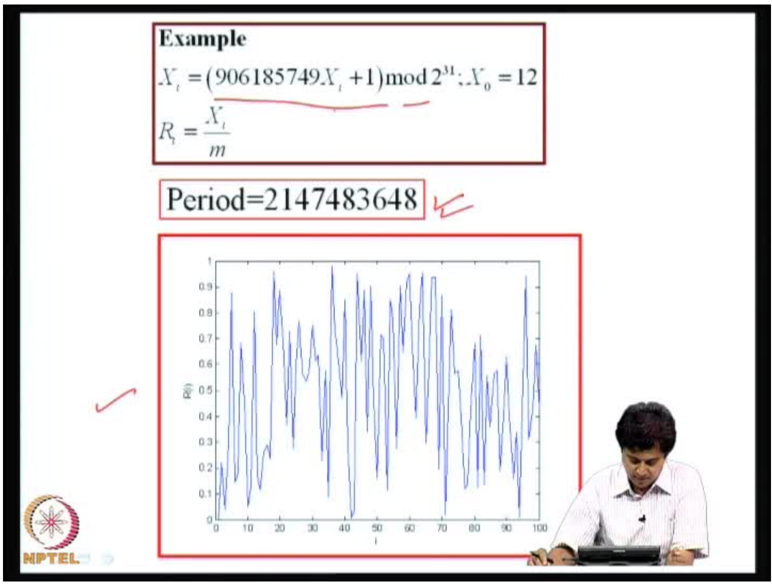
$R_i = \frac{X_i}{m}$

Period=16

0.7500	0.7500
0.9375	0.9375
0.6250	0.6250
0.8125	0.8125
0.5000	0.5000
0.6875	0.6875
0.3750	0.3750
0.5625	0.5625
0.2500	0.2500
0.4375	0.4375
0.1250	0.1250
0.3125	0.3125
0	0
0.1875	0.1875
0.8750	0.8750
0.0625	0.0625

Now, similarly, we make another choice. Again, the period is 16, we are, what we are doing is we are changing the initial value. This initial value is 3 here and this is made as 12 here, and this is known as seed of the random number generator. So, for different seeds, we get different sets of 16 numbers, but that repeats.

(Refer Slide Time: 05:48)



Example

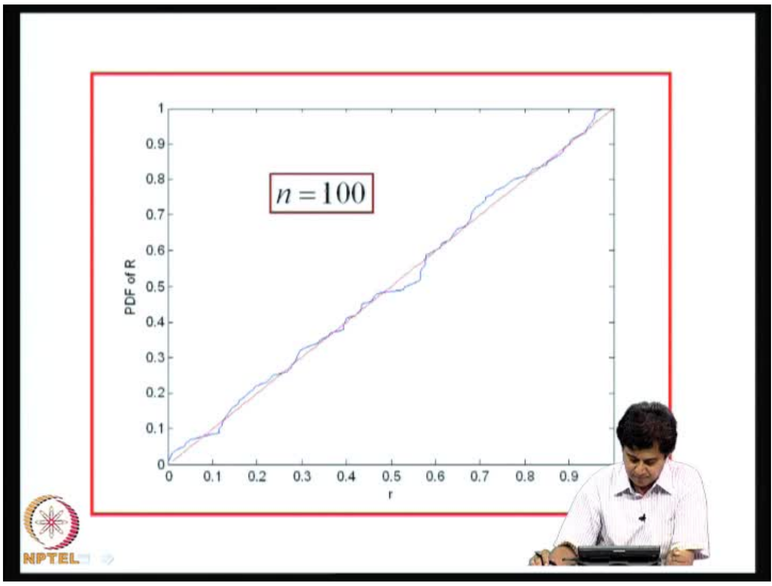
$X_i = (906185749X_i + 1) \bmod 2^{31} ; X_0 = 12$

$R_i = \frac{X_i}{m}$

Period=2147483648

Now, on a computer, typically the choice of a, c and m will read something like this - this m is 2 to the power of 31; and, this a is this large number; and for X naught equal to 12, I have simulated 100 numbers and I have shown the plot here and you can show that the period of this generator is given by this number. So, this is a fairly large number, if your, if your application uses numbers, total number of random numbers that you are using is less than this, you can go for this.

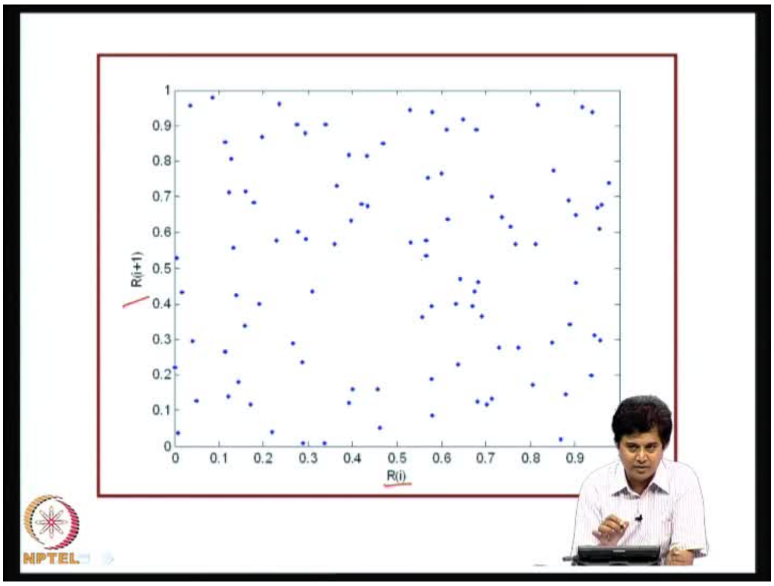
(Refer Slide Time: 06:32)



.

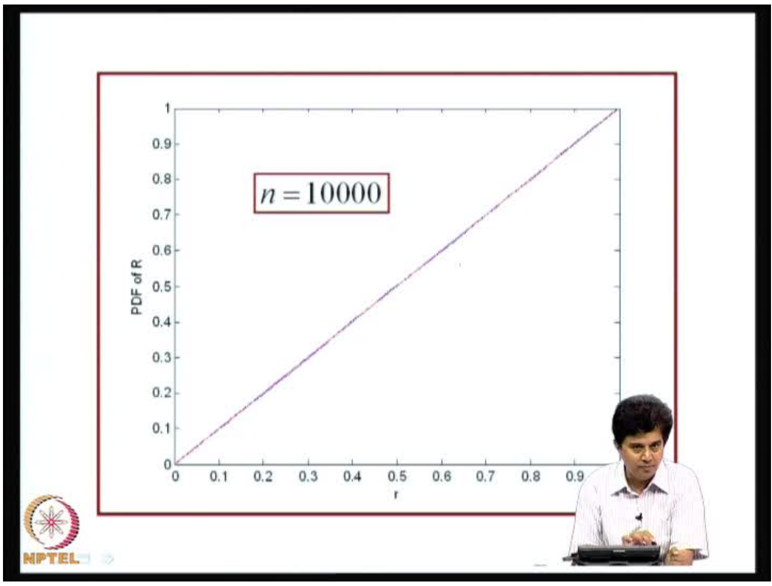
So, if you take 100 such numbers and draw the empirical probability distribution function, if it is uniformly distributed, it should lie along this red line and we can conduct for example, hypothesis test like, call Kolmogorov-Smirnov test and see, whether we can accept the hypothesis, that these numbers are coming from a population, which is uniformly distributed 0 and 1 at some significance level, so that is, whenever hypothesis testing helps us to access the random numbers.

(Refer Slide Time: 07:03)



So, if I plot now these 10 numbers, suppose I am plotting (R i plus 1) versus R i, so our worry will be, is there any order in this sequence of numbers? If it is truly random, there should be no hidden orders. So, here if I plot i plus 1 versus i, this should be randomly distributed in this space. So, similarly, I can do with similar higher dimensional spaces and we never know, whether such an order exist or not, so we have to go on checking.

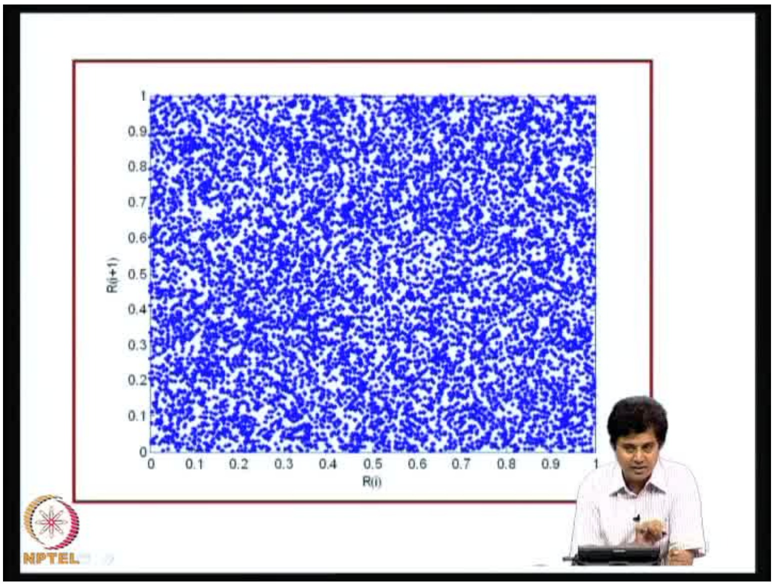
(Refer Slide Time: 07:35)



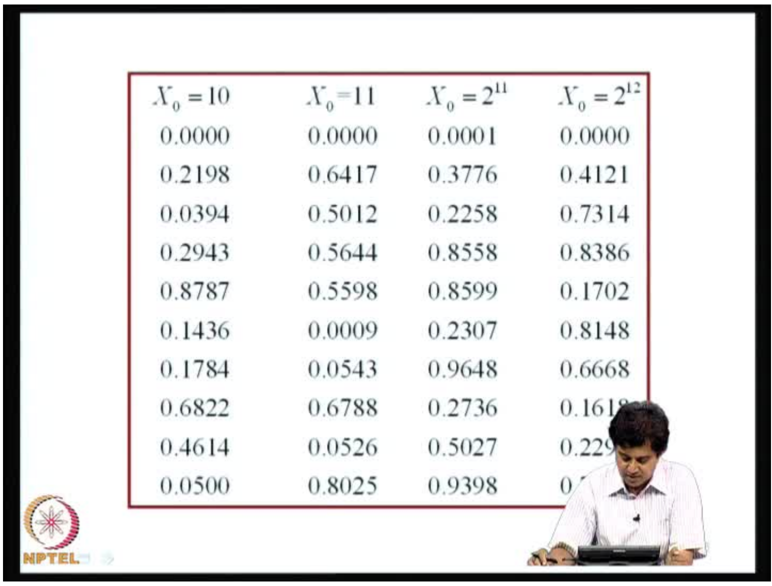
.

So, if I use 10000 these numbers, you can see, that the agreement between uniform, uniform target, uniform distribution and the empirical distribution is pretty close and again, we can do the K-S test and see, whether we can accept the hypothesis, that these numbers originate from population with uniformly distributed probability distribution?

(Refer Slide Time: 07:56)



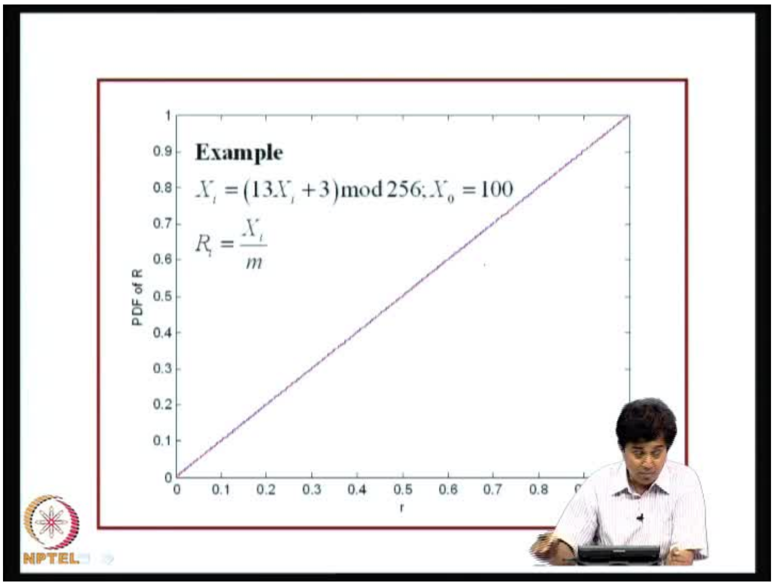
(Refer Slide Time: 08:15)



$X_0 = 10$	$X_0 = 11$	$X_0 = 2^{11}$	$X_0 = 2^{12}$
0.0000	0.0000	0.0001	0.0000
0.2198	0.6417	0.3776	0.4121
0.0394	0.5012	0.2258	0.7314
0.2943	0.5644	0.8558	0.8386
0.8787	0.5598	0.8599	0.1702
0.1436	0.0009	0.2307	0.8148
0.1784	0.0543	0.9648	0.6668
0.6822	0.6788	0.2736	0.1618
0.4614	0.0526	0.5027	0.229
0.0500	0.8025	0.9398	0.

Now, these 10000 numbers are plotted, that is, (R i plus 1) versus R i is plotted and we can see, that this is some kind of random image. I cannot discern any pattern here, so there is no reason to reject them, these numbers as being random. This story is not always so, I will come to that slightly later. Now, I have shown a few random numbers starting with different initial conditions, that is, seed; if I start with 10, I get these numbers, if I start with 11 I get different numbers, and so on and so forth. So, just for illustration, how these numbers behave.

(Refer Slide Time: 08:36)



Example

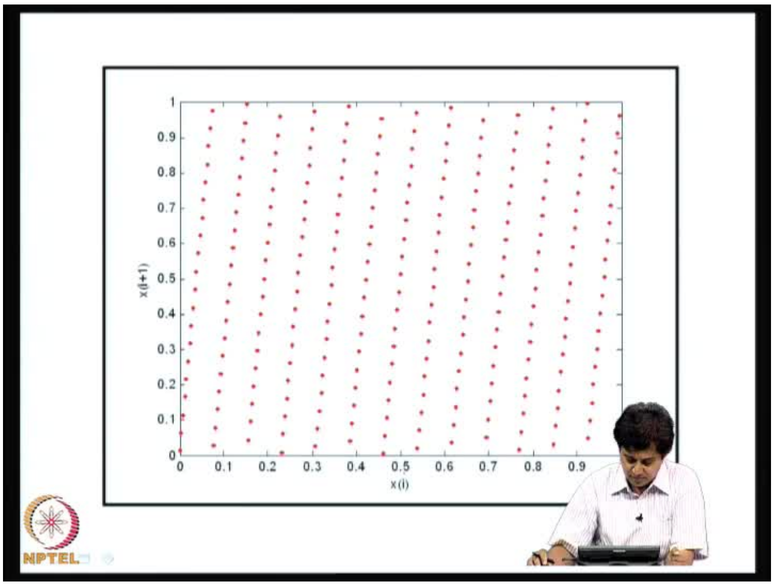
$X_i = (13X_i + 3) \bmod 256; X_0 = 100$

$R_i = \frac{X_i}{m}$

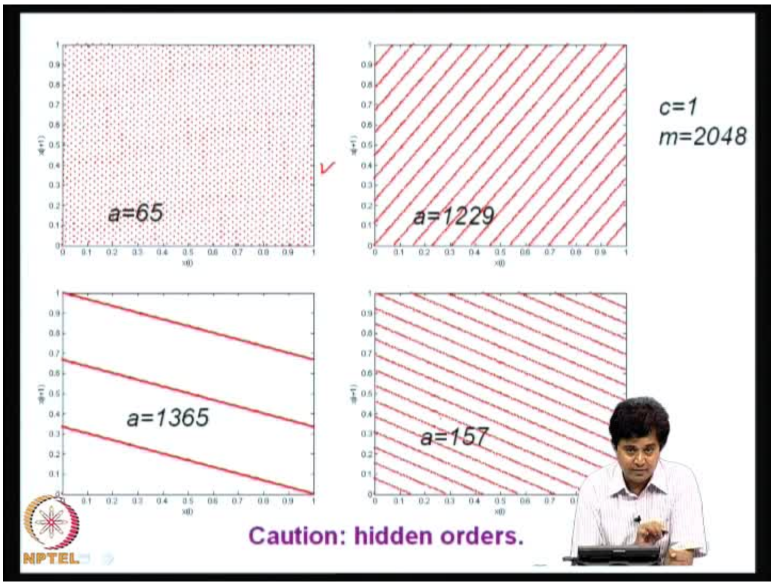
.

Now, if I take m to be 256 and a to be 13 and c to be 3, and generate say, few random numbers and I think I have generated some 10000 numbers, and if I show the empirical probability distribution function, it matches very well with the target distribution.

(Refer Slide Time: 08:56)



(Refer Slide Time: 09:10)

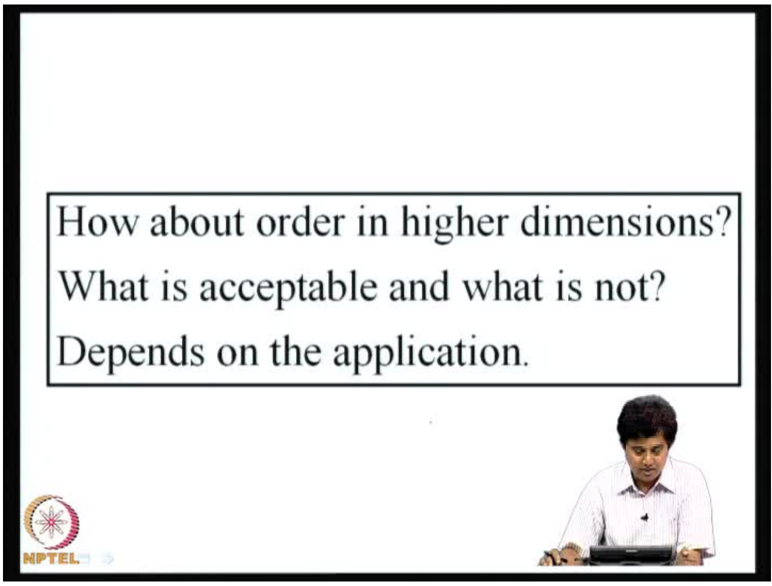


Caution: hidden orders.

.

But if we look at the (i plus 1) versus i plot, there is a structure, so obviously, this is alarming, we cannot accept these as random numbers. More such examples, I have shown the parameters a, c and m, so we can see here, this is some kind, in all these 4 cases, there are quite a good amount of patterns that emerge and this cannot be trusted as random numbers. So, if you make these choices, I mean, if you try generating random numbers, then you have tough task on your hand, you have to perform several test and in the literature, there are several test that are discussed in this context. So, danger lurks everywhere, so you have to be cautious.

(Refer Slide Time: 09:50)

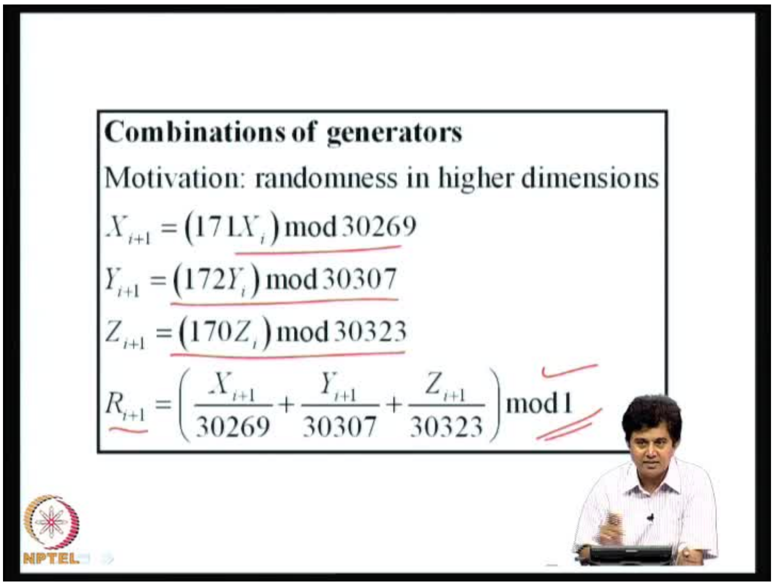


How about order in higher dimensions?
What is acceptable and what is not?
Depends on the application.

.

Now, how about order in higher dimensions? I am plotting (i plus 1) versus i. Suppose, I have plot (i plus 1), (i plus 2) versus (i plus 1) versus i in a 3-dimensional space and they should be randomly distributed. How do I know in 5-dimensional space there are no orders? So, this is a question that is almost impossible to answer, so it depends on the application. Suppose, we are evaluating a 2-dimensional or a 3-dimensional integral using Monte Carlo simulation, you should ensure that at least, in 3-dimensional there are no manifest orders in these numbers. So, this is where one has to be cautious.

(Refer Slide Time: 10:31)



Combinations of generators

Motivation: randomness in higher dimensions

$X_{i+1} = (171X_i) \bmod 30269$

$Y_{i+1} = (172Y_i) \bmod 30307$

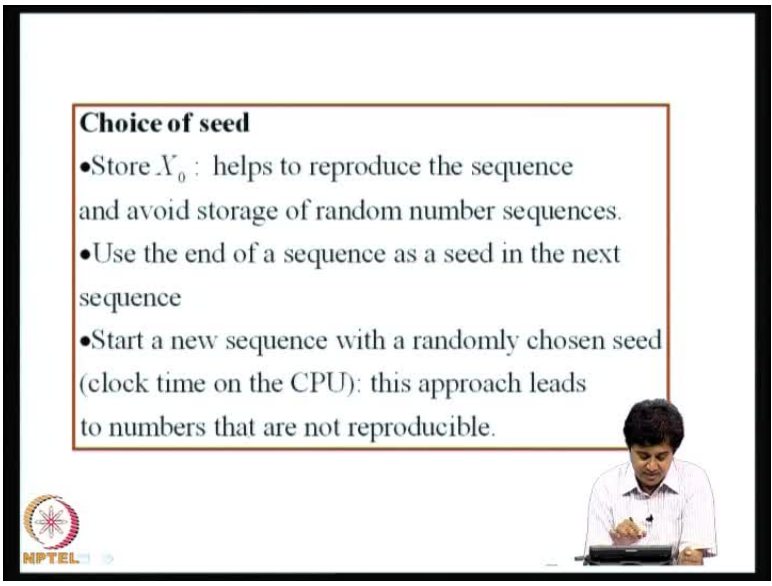
$Z_{i+1} = (170Z_i) \bmod 30323$

$$R_{i+1} = \left(\frac{X_{i+1}}{30269} + \frac{Y_{i+1}}{30307} + \frac{Z_{i+1}}{30323} \right) \bmod 1$$

.

Now, there are other strategies like combinations of generators to deal with to ensure that there exists randomness in higher dimensions. So, there are various algorithms, one of that I will state here. So, you, you generate, you take 3 generators, (X i plus 1) is this, (Y i plus 1) is this, (Z i plus 1) is this and then you define (R i plus 1) as a kind of sum shown here, modulo 1, and this will be, this is supposed to behave well in 3-dimensions, well, in terms of being random.

(Refer Slide Time: 11:10)



Choice of seed

•Store X_0 : helps to reproduce the sequence and avoid storage of random number sequences.

•Use the end of a sequence as a seed in the next sequence

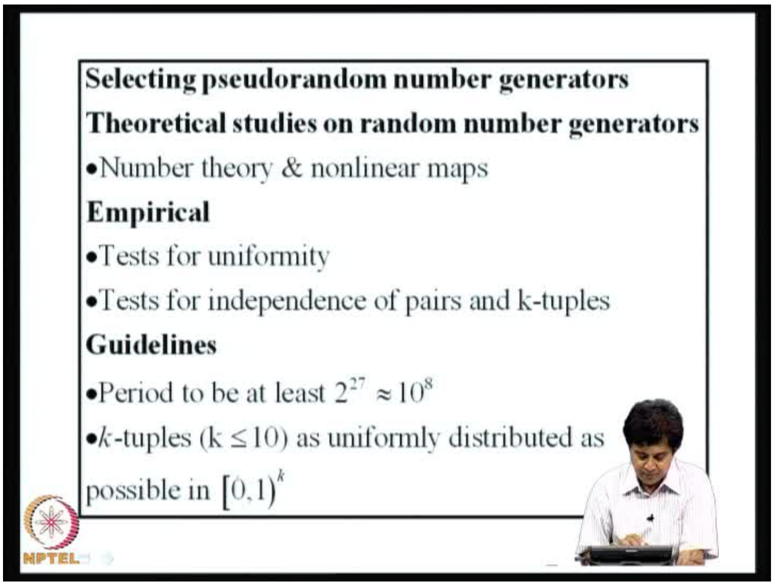
•Start a new sequence with a randomly chosen seed (clock time on the CPU): this approach leads to numbers that are not reproducible.

Now, I talked about seed. So, on a computer, when you are generating random numbers, if you want to debug your program, every time you run the computer, suppose you are generating 10000 random numbers, it is useful to get the same 10000 numbers. One way is to generate them and store and read from the disk, but in the program cell, it is infeasible, if you are really doing a practical problem to store all the random numbers before and draw from them. So, the best way is to start with the same seed and that would ensure, that we get same set of random numbers.

So, this also, this helps us to reproduce the sequence on one hand and also, it, we can avoid storage of random sequences on the computer. Now, if you want to prolong the sequence, you have generated 10000, you need ten more thousand random numbers, you can use the last number in your first sequence as a seed for the next one, so that you continue with same sequence, or you can start a new sequence with a randomly chosen seed. For example, clock time on the CPU or some other numbers, but this approach leads to numbers that are not reproducible. If you do not specify seed yourself, you cannot reproduce the sequence of random numbers.

I am talking about generation of random numbers on computers. There are strategies where the random numbers are drawn from a physical process, for example noise is drawn from atmosphere through a radio, so that noise is digitized and used as random numbers. So, there is no algorithm there and we bank on certain physical mechanism for production of random numbers, but for, for the Monte Carlo application that we are discussing, we bank on random numbers, which are generated on a computer.

(Refer Slide Time: 13:07)



Selecting pseudorandom number generators

Theoretical studies on random number generators

- Number theory & nonlinear maps

Empirical

- Tests for uniformity
- Tests for independence of pairs and k-tuples

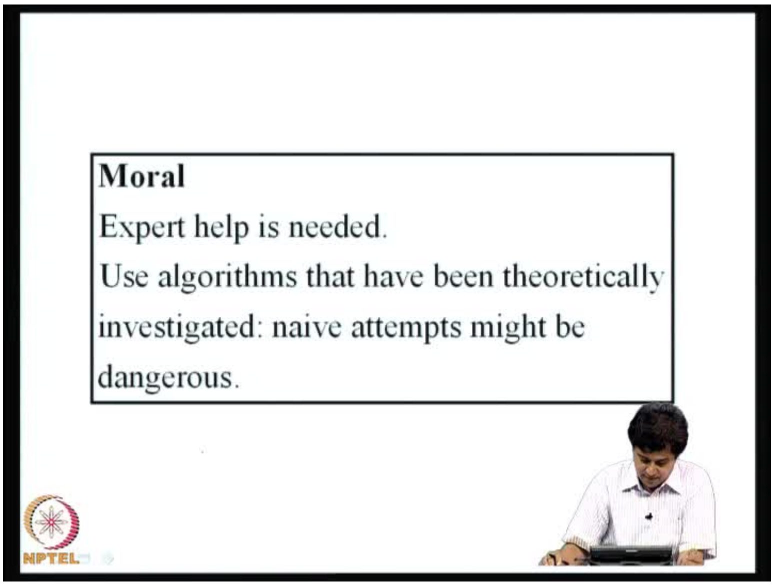
Guidelines

- Period to be at least $2^{27} \approx 10^8$
- k-tuples ($k \leq 10$) as uniformly distributed as possible in $[0,1)^k$

.

So, selecting pseudorandom number generators, you know you need to, I mean this choice is based on theoretical studies on random number generators. It involves number theory and study of non-linear maps, so if you are doing empirically, you have to test for uniformity, test for independence of pairs and k tuples, and so on and so forth. There are some guidelines: period to be at least 2 to the power of 27, that is, about 10 to the power of 8, and k tuples about up to the order of 10, as uniformly distributed as possible in 10 and k dimensional spaces. So, you can do some of these tests using hypothesis testing methods and see whether these numbers pass the, I mean, you can accept these numbers according to the given level of significance.

(Refer Slide Time: 13:58)



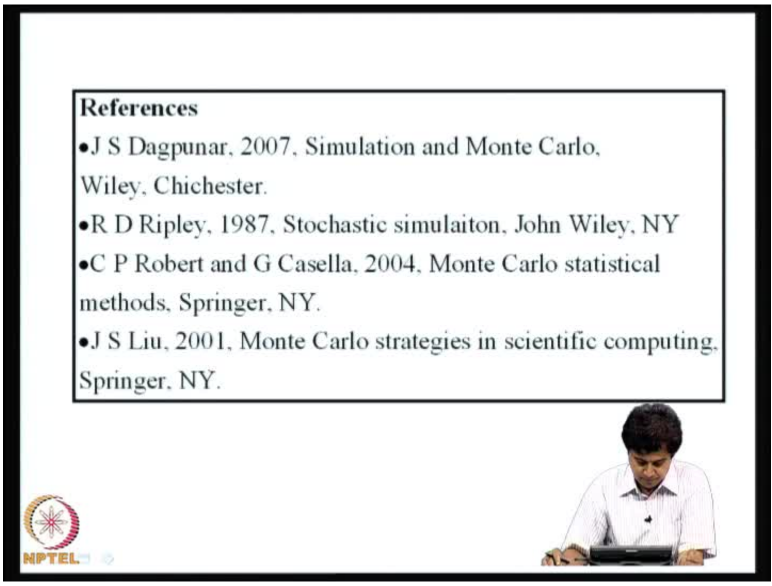
Moral

Expert help is needed.

Use algorithms that have been theoretically investigated: naive attempts might be dangerous.

Now, the moral is, you, if we are using pseudorandom numbers on a computer you need expert help. Most of the professional softwares have random number generators, that is, numbers that are uniformly distributed in 0 and 1, which are based on algorithms, which are theoretically investigated and you are advised to use such algorithms. If you attempt to dissimilate your own random number, you have to be very careful about many aspects of the nature of these random numbers. So, you use what is professionally available and trust; that is the moral.

(Refer Slide Time: 14:36)

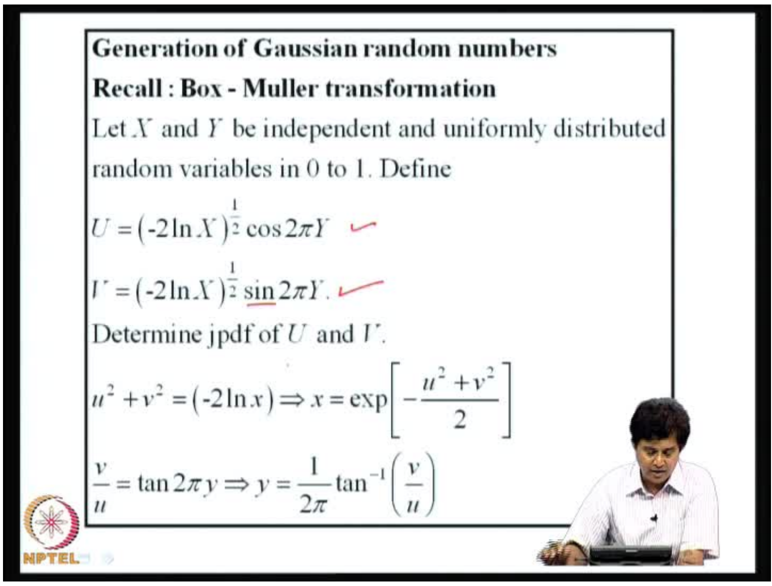


References

- J S Dagpunar, 2007, Simulation and Monte Carlo, Wiley, Chichester.
- R D Ripley, 1987, Stochastic simulaiton, John Wiley, NY
- C P Robert and G Casella, 2004, Monte Carlo statistical methods, Springer, NY.
- J S Liu, 2001, Monte Carlo strategies in scientific computing, Springer, NY.

There are few references I have listed here: book by Dagpunar, Ripley, Robert and Casella and Liu, which discusses not only the details of pseudorandom number generators, but also few of the methods that I am going to discuss in due course.

(Refer Slide Time: 14:56)



Generation of Gaussian random numbers

Recall : Box - Muller transformation

Let X and Y be independent and uniformly distributed random variables in 0 to 1. Define

$$U = (-2\ln X)^{\frac{1}{2}} \cos 2\pi Y$$

$$V = (-2\ln X)^{\frac{1}{2}} \underline{\sin} 2\pi Y.$$

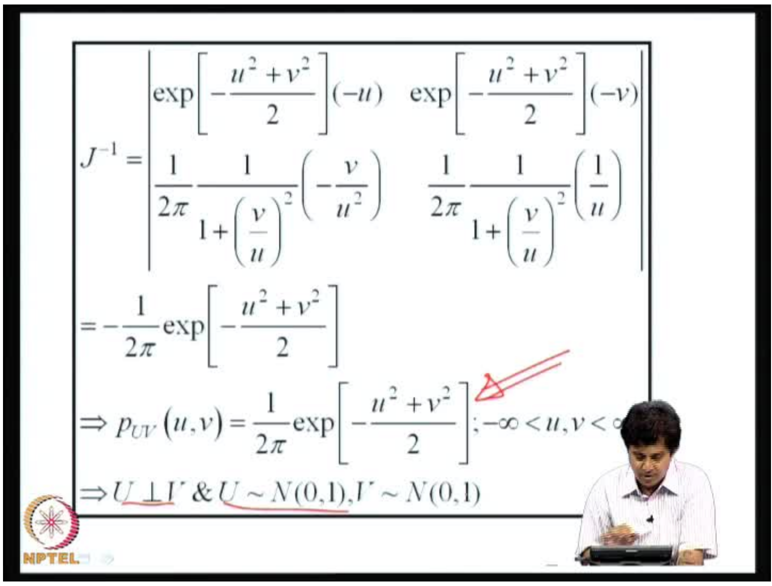
Determine jpdf of U and V.

$$u^2 + v^2 = (-2\ln x) \Rightarrow x = \exp\left[-\frac{u^2+v^2}{2}\right]$$

$$\frac{v}{u} = \tan 2\pi y \Rightarrow y = \frac{1}{2\pi}\tan^{-1}\left(\frac{v}{u}\right)$$

Now, we have generated now uniformly distributed random numbers. How do we generate say, Gaussian random numbers? So, we here, many strategies are possible. We recall, when we studied transformation of random variables, there was an algorithm known as transformation, known as Box-Muller transformation; we will quickly recall what it is. So, let X and Y be independent and uniformly distributed random variables 0 to 1. I define U as minus 2ln X to the power of half cos 2 pi Y and similarly V, instead of cos here, I have sine. The problem is to find the joint probability distribution function of U and V. We have solved this problem and we have shown that U and V are independent and jointly normal, they have 0 mean and unit standard deviation. So, that we have done;

(Refer Slide Time: 15:50)



$$J^{-1}=\begin{vmatrix}\exp\left[-\dfrac{u^2+v^2}{2}\right](-u) & \exp\left[-\dfrac{u^2+v^2}{2}\right](-v)\\ \dfrac{1}{2\pi}\dfrac{1}{1+\left(\dfrac{v}{u}\right)^2}\left(-\dfrac{v}{u^2}\right) & \dfrac{1}{2\pi}\dfrac{1}{1+\left(\dfrac{v}{u}\right)^2}\left(\dfrac{1}{u}\right)\end{vmatrix}$$

$$=-\frac{1}{2\pi}\exp\left[-\frac{u^2+v^2}{2}\right]$$

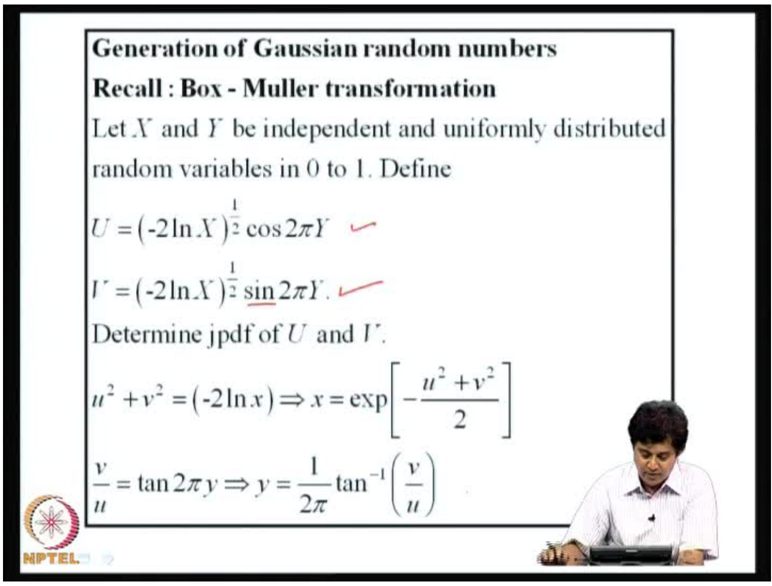
$$\Rightarrow p_{UV}(u,v)=\frac{1}{2\pi}\exp\left[-\frac{u^2+v^2}{2}\right];-\infty<u,v<\infty$$

$$\Rightarrow U\perp V\ \&\ U\sim N(0,1), V\sim N(0,1)$$

.

These are the steps where we find the jacobian and go through this and we get this joint probability density function, that would mean U and V are independent; U is normal with mean 0 and standard deviation 1, V is normal with mean 0 and standard deviation 1.

(Refer Slide Time: 14:56)



Generation of Gaussian random numbers

Recall : Box - Muller transformation

Let X and Y be independent and uniformly distributed random variables in 0 to 1. Define

$$U=(-2\ln X)^{\frac{1}{2}}\cos 2\pi Y$$

$$V=(-2\ln X)^{\frac{1}{2}}\sin 2\pi Y.$$

Determine jpdf of U and V.

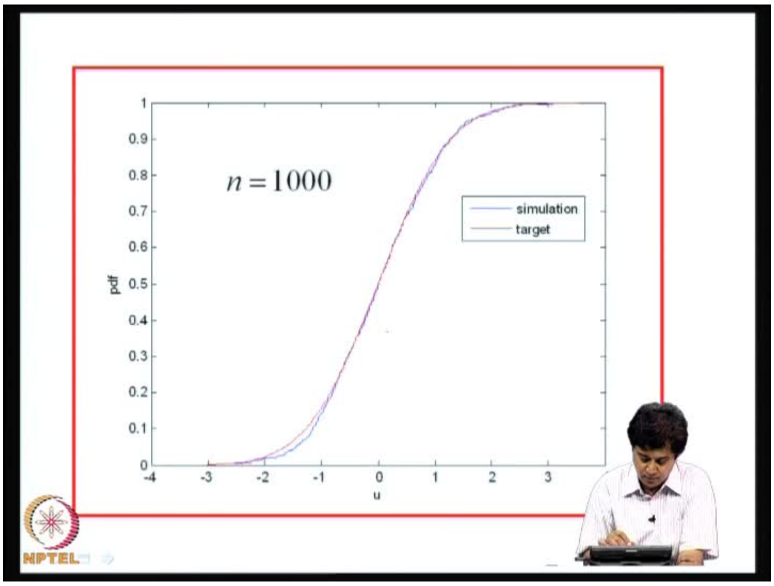
$$u^2+v^2=(-2\ln x)\Rightarrow x=\exp\left[-\frac{u^2+v^2}{2}\right]$$

$$\frac{v}{u}=\tan 2\pi y\Rightarrow y=\frac{1}{2\pi}\tan^{-1}\left(\frac{v}{u}\right)$$

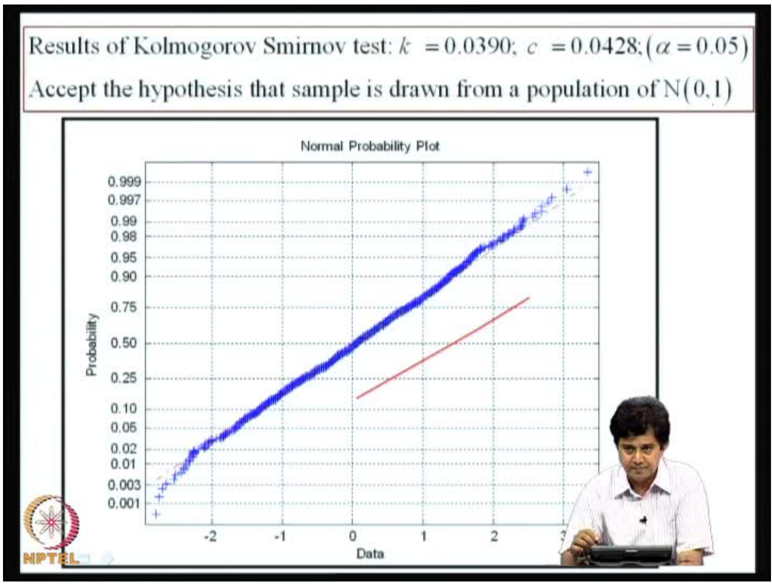
.

So, we could use, now that we know how to generate uniformly distributed random numbers, we can use Box-Muller transformation and generate Gaussian random numbers. So, generation of Gaussian random numbers, once you know how to generate uniform random numbers, is reasonably straight forward. Again, if your random number generated in the, for uniform distributed random numbers is bad, it will get into U and V also, so I have to be careful.

(Refer Slide Time: 16:34)



(Refer Slide Time: 16:56)



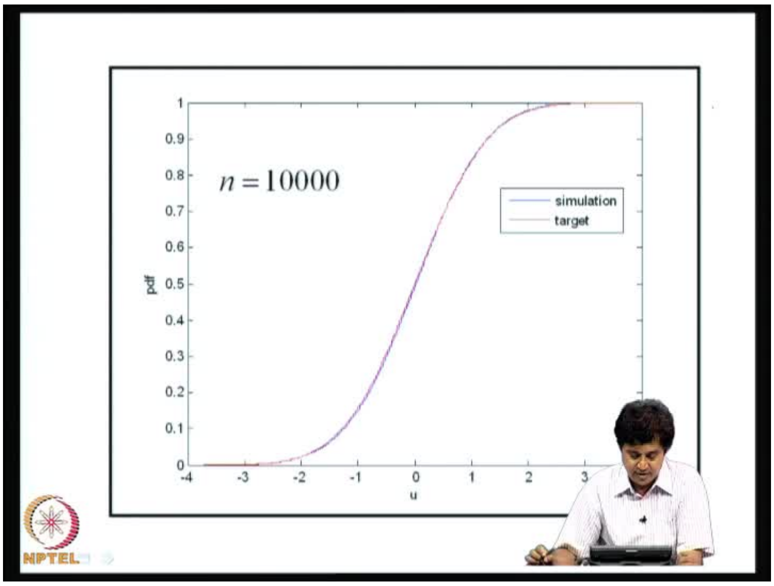
Results of Kolmogorov Smirnov test: $k = 0.0390$; $c = 0.0428$; $(\alpha = 0.05)$

Accept the hypothesis that sample is drawn from a population of $N(0,1)$

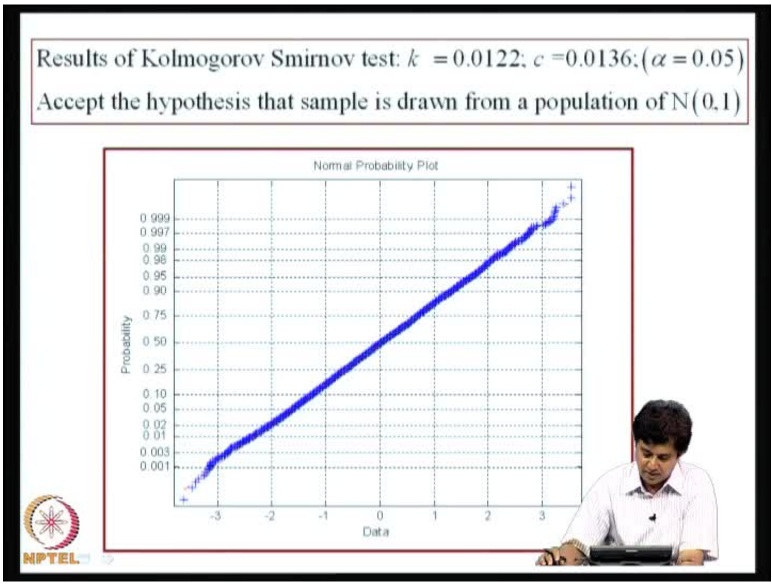
.

So, I have used 1000 numbers and plotted the empirical probability distribution function of, random, Gaussian distributed random numbers; blue line is the simulation, red is the target, and again, we see reasonable agreement between the two. If I use the same results, if I plot on normal probability paper, they look like this and qualitatively we can see that these numbers are lying on a straight line. And I did perform the Kolmogorov-Smirnov test at 5 percent significance level and I reached the conclusion, that we can accept the hypothesis, that sample is drawn from a population of normal random variables with 0 mean, unit standard deviation at 5 percent significance. So, it passes the test.

(Refer Slide Time: 17:29)



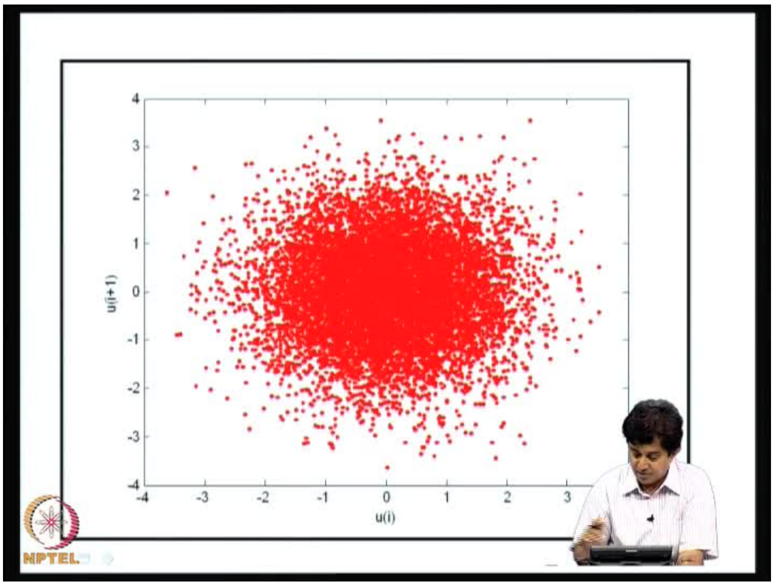
(Refer Slide Time: 17:39)


Results of Kolmogorov Smirnov test: $k = 0.0122$; $c = 0.0136$; $(\alpha = 0.05)$

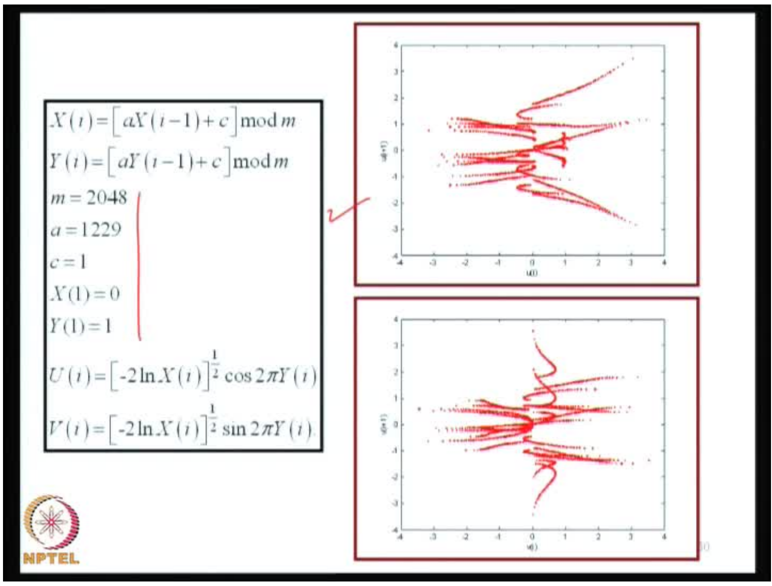
Accept the hypothesis that sample is drawn from a population of $\mathrm{N}(0,1)$

Similarly 10000; the agreement, as we can infer from visual inspection is much better now and if you conduct the test again, plot on the, you know, do the case Kolmogorov-Smirnov test again, we will be in a position to accept the hypothesis, that these numbers are drawn from a standard normal population at 5 percent significance level.

(Refer Slide Time: 17:54)



(Refer Slide Time: 18:09)


$$X(i)=\left[aX(i-1)+c\right]\bmod m$$

$$Y(i)=\left[aY(i-1)+c\right]\bmod m$$

$$m=2048$$

$$a=1229$$

$$c=1$$

$$X(1)=0$$

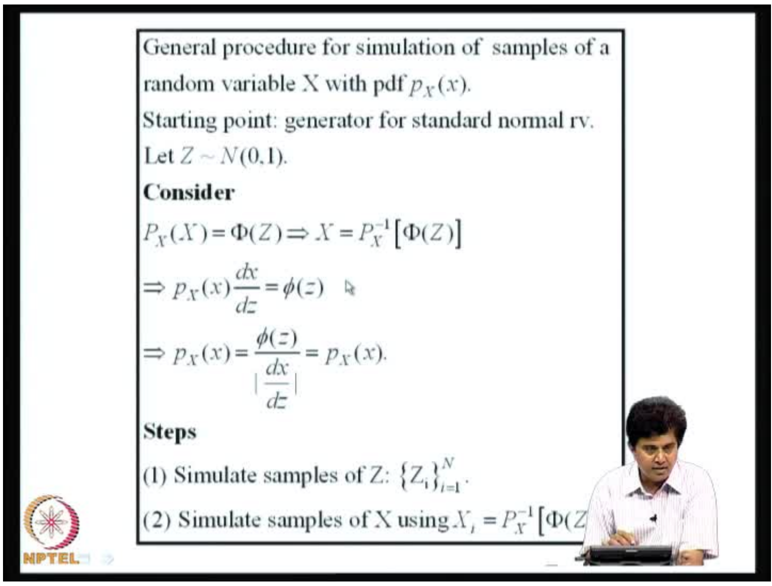
$$Y(1)=1$$

$$U(i)=\left[-2\ln X(i)\right]^{\frac{1}{2}}\cos 2\pi Y(i)$$

$$V(i)=\left[-2\ln X(i)\right]^{\frac{1}{2}}\sin 2\pi Y(i)$$

So, these numbers I have displayed on 2-dimensional map, (U i plus 1) versus U i and we do not recognize any pattern in this, so this seems to be acceptable again. Just a word of caution, I have used the congruential algorithm for generating uniform distributed random numbers X and Y, I have simulated following this prescription and then I have used a Box-Muller transformation, and the results on U and V are shown here. This is (U i plus 1) versus U i, this one and this is (V i plus 1) versus V i. So this has disturbing levels of order, the numbers and if you use these numbers in your modeling works, you are bound to face lots of difficulties.

(Refer Slide Time: 18:48)



General procedure for simulation of samples of a random variable X with pdf $p_X(x)$.

Starting point: generator for standard normal rv.

Let $Z \sim N(0,1)$.

Consider

$$P_X(X) = \Phi(Z) \Rightarrow X = P_X^{-1}\left[\Phi(Z)\right]$$

$$\Rightarrow p_X(x)\frac{dx}{dz} = \phi(z)$$

$$\Rightarrow p_X(x) = \frac{\phi(z)}{\left|\frac{dx}{dz}\right|} = p_X(x).$$

Steps

(1) Simulate samples of Z: $\{Z_i\}_{i=1}^{N}$.

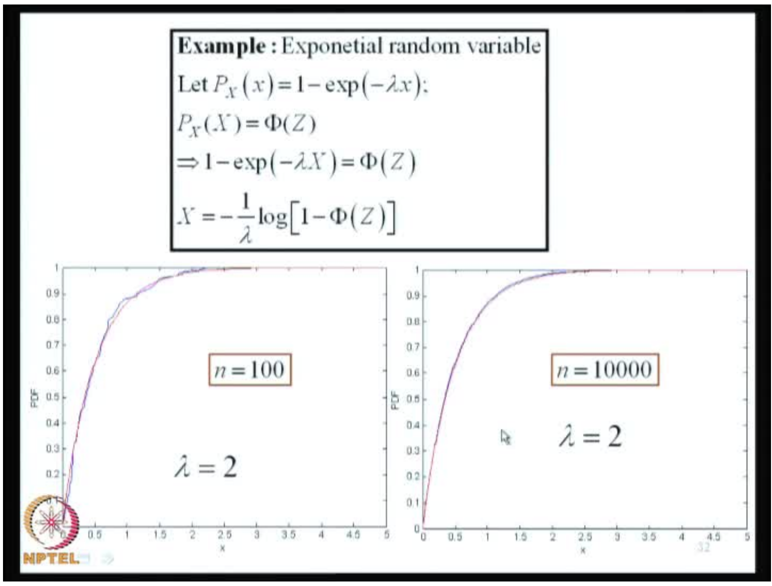
(2) Simulate samples of X using $X_i = P_X^{-1}\left[\Phi(Z\right.$

.

Now, I talked about Gaussian distributed random variables, how about X, which has a given probability density function P X of x. Now, we can take the, as a starting point our ability to generate standard normal random numbers, 0 mean, unit standard deviation random numbers. Now, we consider the transformation P X of x is phi of Z, where capital phi is the probability distribution function of standard normal random variable and X is the target random variable, and X is written as X equal to P X inverse phi of Z. So, one of the prerequisite for applying this method is that you should be able to invert the probability distribution of the target random variable.

Now, if you follow the rules of transformation of random variables, you can show that if you generate X according to this rule, the probability distribution function or density function of X will be identical to the target probability density function. Now, so what are the steps we simulate? Standard normal random variables use this transformation and get samples of X. So, this is very straight forward exercise, provided you know how to invert your probability distribution function.

(Refer Slide Time: 20:03)



Example : Exponetial random variable

Let $P_X(x) = 1 - \exp(-\lambda x)$;

$P_X(X) = \Phi(Z)$

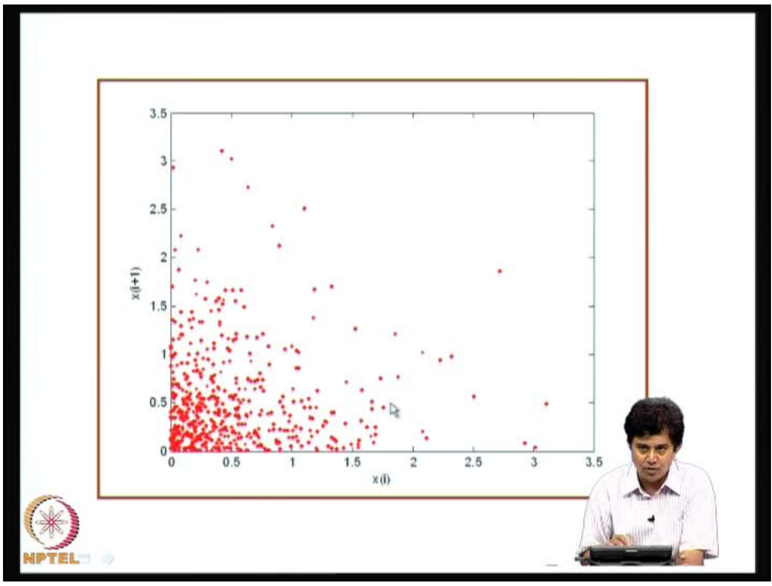
$\Rightarrow 1 - \exp(-\lambda X) = \Phi(Z)$

$X = -\frac{1}{\lambda}\log\left[1 - \Phi(Z)\right]$

.

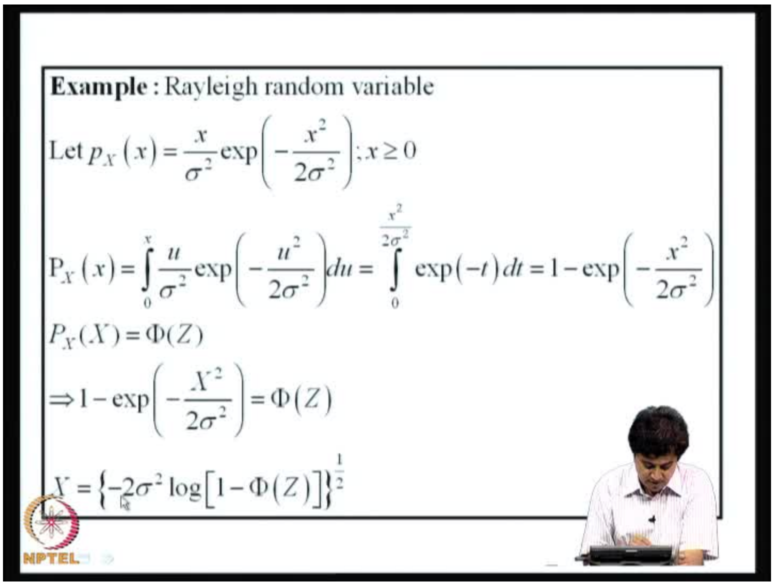
So, we can start with an exponential random variable. The probability distribution function is 1 minus exponential minus lambda x. So, the transformation here is 1 minus exponential minus lambda X, this is random variable X equal to phi of Z and from this if I solve for X, X will be minus 1 by lambda log of (1 minus phi of Z). So, I simulate normal random variables 0, 1 and use the transformation, generate the corresponding sample of X. So, I have done with 100 numbers here for some value of lambda and you can see, the blue line is the, is empirical probability distribution function generated from the samples, and red is a theoretical or the target probability distribution function. And with 10000 samples, the match is lot better as to be expected, and again, we can perform the Kolmogorov-Smirnov test and be sure, that we have succeeded in simulating acceptable random numbers.

(Refer Slide Time: 21:00)



So, this is a map of X (i plus 1) versus X i. Again, I do not see any discernible order in this, so it looks fine.

(Refer Slide Time: 21:11)



Example : Rayleigh random variable

Let $p_X(x) = \frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right); x \geq 0$

$$P_X(x) = \int_0^x \frac{u}{\sigma^2}\exp\left(-\frac{u^2}{2\sigma^2}\right)du = \int_0^{\frac{x^2}{2\sigma^2}} \exp(-t)\,dt = 1-\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

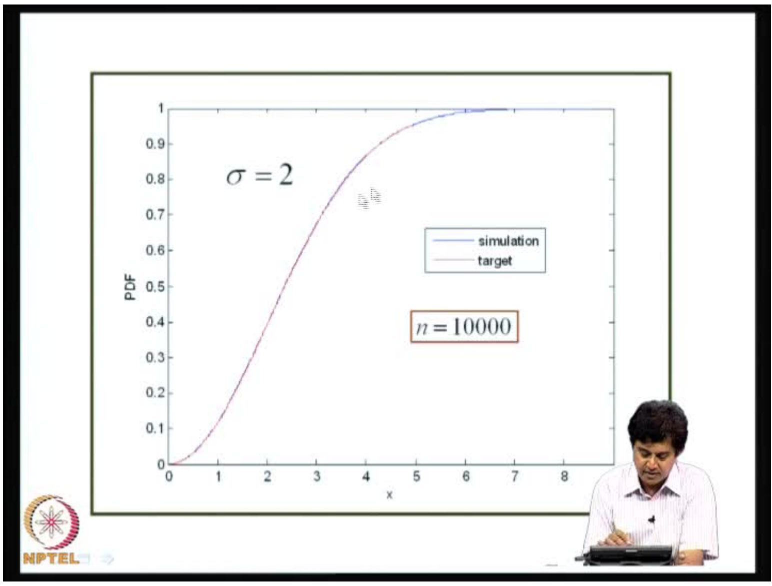
$P_X(X) = \Phi(Z)$

$\Rightarrow 1-\exp\left(-\frac{X^2}{2\sigma^2}\right) = \Phi(Z)$

$X = \left\{-2\sigma^2\log\left[1-\Phi(Z)\right]\right\}^{\frac{1}{2}}$

Another example, how do you simulate Rayleigh random variables? So, p X of x in this case is x by sigma square exponential minus x square by 2 sigma square, where x to transform 0 to infinity, and the distribution function here becomes, you can integrate this and we get 1 minus exponential minus x square by 2 sigma square. So, the transformation we are looking for is 1 minus exponential minus X square by 2 sigma square is equal to phi of Z. So, you invert this and I get X, which is a seed function of Z, as shown here. So, once this is determined, the ball gets rolling, we simulate random numbers Z normal 0 1, apply this transformation, get sample of X.

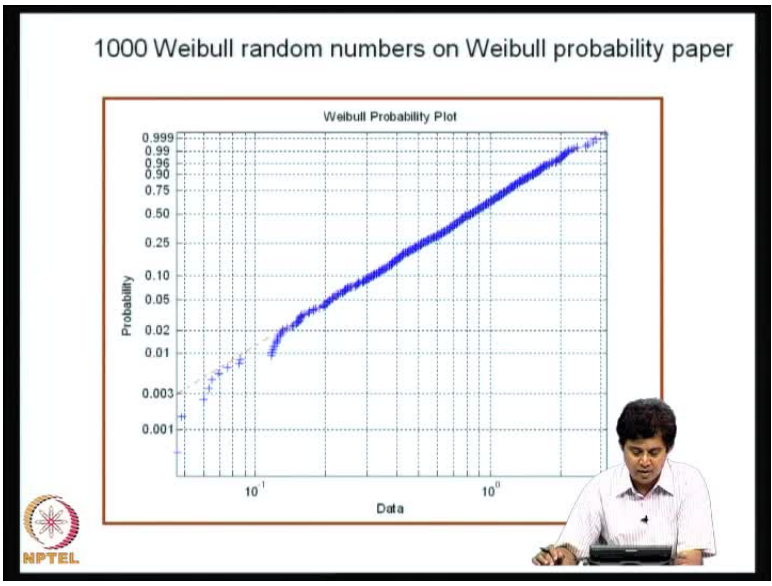
(Refer Slide Time: 21:58)



.

So, here with 10000 samples for sigma equal to 2, I see that the match between simulation, which is blue line and the red, which is the target, is quite good.

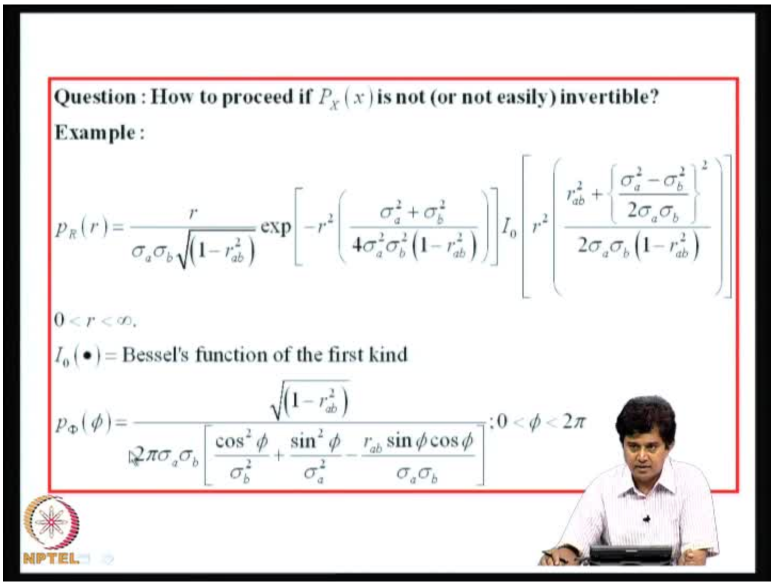
(Refer Slide Time: 22:13)


1000 Weibull random numbers on Weibull probability paper

.

Now, similarly, we can, I have done few more exercises I have generated 1000 Weibull random numbers on and displayed on Weibull probability paper, they look fine.

(Refer Slide Time: 22:26)



Question : How to proceed if $P_X(x)$ is not (or not easily) invertible?

Example :

$$p_R(r)=\frac{r}{\sigma_a\sigma_b\sqrt{\left(1-r_{ab}^2\right)}}\exp\left[-r^2\left(\frac{\sigma_a^2+\sigma_b^2}{4\sigma_a^2\sigma_b^2\left(1-r_{ab}^2\right)}\right)\right]I_0\left[r^2\left(\frac{r_{ab}^2+\left\{\dfrac{\sigma_a^2-\sigma_b^2}{2\sigma_a\sigma_b}\right\}^2}{2\sigma_a\sigma_b\left(1-r_{ab}^2\right)}\right)\right]$$

$0<r<\infty.$

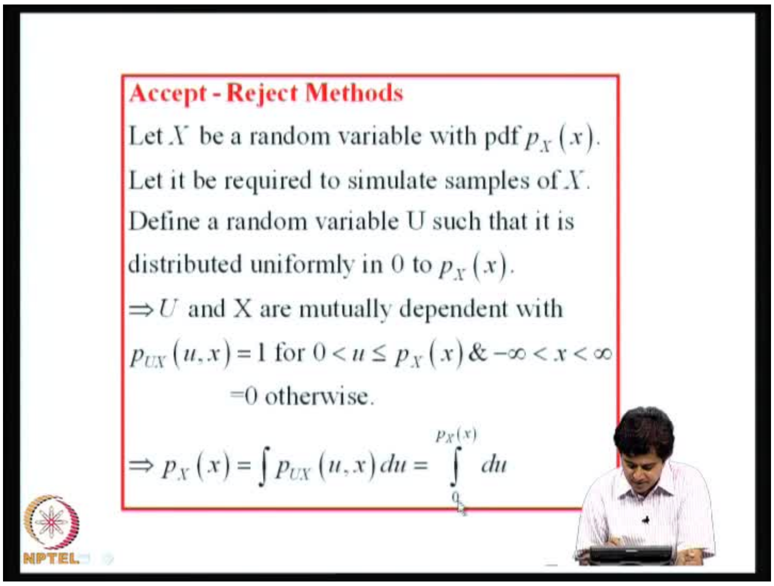
$I_0(\bullet)=$ Bessel's function of the first kind

$$p_\Phi(\phi)=\frac{\sqrt{\left(1-r_{ab}^2\right)}}{2\pi\sigma_a\sigma_b\left[\dfrac{\cos^2\phi}{\sigma_b^2}+\dfrac{\sin^2\phi}{\sigma_a^2}-\dfrac{r_{ab}\sin\phi\cos\phi}{\sigma_a\sigma_b}\right]};0<\phi<2\pi$$

Now, what happens if you are not in a position to invert the probability distribution function? How will you proceed? For example, in our earlier studies on envelopes, I got the probability distribution for an envelope as displayed here. There is Bessels's function here and r sits here and r sits here and r is here, and how do you, this is density function, so you have integrate this with respect to r and find the distribution function and then invert it. So, it is, it is very unlikely that we succeed in this.

Similarly, you take the probability distribution of the phase, you have cos square phi sine square phi sine phi cos phi etcetera, etcetera, and it is very unlikely, that you will be able to first of all find out the probability distribution function and then invert it. So, how do we proceed in such situations? So, the method of transformation of random variables does not seem to, I mean, it is not a promising method for these problems.

(Refer Slide Time: 23:29)



Accept - Reject Methods

Let X be a random variable with pdf $p_X(x)$.

Let it be required to simulate samples of X.

Define a random variable U such that it is distributed uniformly in 0 to $p_X(x)$.

$\Rightarrow U$ and X are mutually dependent with

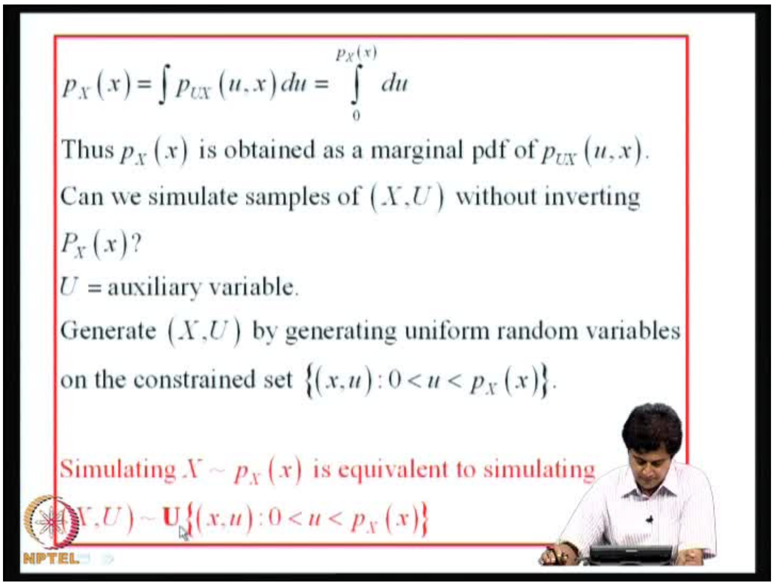
$p_{UX}(u,x) = 1$ for $0 < u \le p_X(x)$ & $-\infty < x < \infty$

$=0$ otherwise.

$$\Rightarrow p_X(x) = \int p_{UX}(u,x)\,du = \int_0^{p_X(x)} du$$

So, there is a method known as accept and reject methods, accept-reject methods. So, the logic of this is as follows, let X be a random variable with pdf p X of x. Let it be required to simulate samples of X. Now, what I do is, I define a random variable U, such that it is distributed uniformly in 0 to p X of x. Now, that would mean U and X are mutually dependent with the joint density function, U of, UX (u, x) is equal to 1, whenever u of, whenever u lies between 0 and p X of x, where x itself takes value from minus infinity to plus infinity, otherwise it is 0. How do you get the marginal density function of X from this? You have to integrate with respect to u, now that is nothing but u is 1 for whenever it takes value from 0 to p X of x. Therefore, it is integral 0 to p X of x du, so p X of x equal to 0 to x p X of x du. It is a very profound result, quite simple to understand, but it has very deep implications.

(Refer Slide Time: 24:33)



$$p_X(x) = \int p_{UX}(u,x)\,du = \int_0^{p_X(x)} du$$

Thus $p_X(x)$ is obtained as a marginal pdf of $p_{UX}(u,x)$.

Can we simulate samples of (X,U) without inverting $P_X(x)$?

U = auxiliary variable.

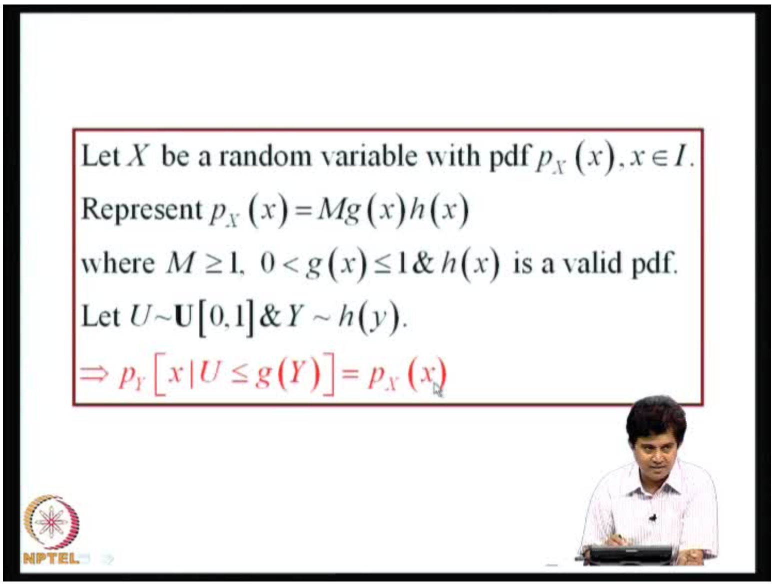
Generate (X,U) by generating uniform random variables on the constrained set $\{(x,u): 0 < u < p_X(x)\}$.

Simulating $X \sim p_X(x)$ is equivalent to simulating $(X,U) \sim \mathbf{U}\{(x,u): 0 < u < p_X(x)\}$

.

So, you, let us look at this again p X of x is 0 to p X of x du. Thus, p X of x is obtained as a marginal probability density function of p UX (u, x). Now, can we simulate samples of X and U without inverting P X of x and by using this logic? Now, this U is known as an auxiliary variable, so the story here is, we generate (X, U) by generating uniform random variables on the constrained set, that is, set of (x, u), such that u takes value between 0 to p X of x. That means, simulating X from p X of x, that is, according to this, probabilistic law is equivalent to simulating these pair (X, U), so that x, the set, constrained set (x, u) is uniformly distributed. This script, u means, it is uniformly distributed.

(Refer Slide Time: 25:38)



Let X be a random variable with pdf $p_X(x), x \in I$.

Represent $p_X(x) = Mg(x)h(x)$

where $M \geq 1$, $0 < g(x) \leq 1$ & $h(x)$ is a valid pdf.

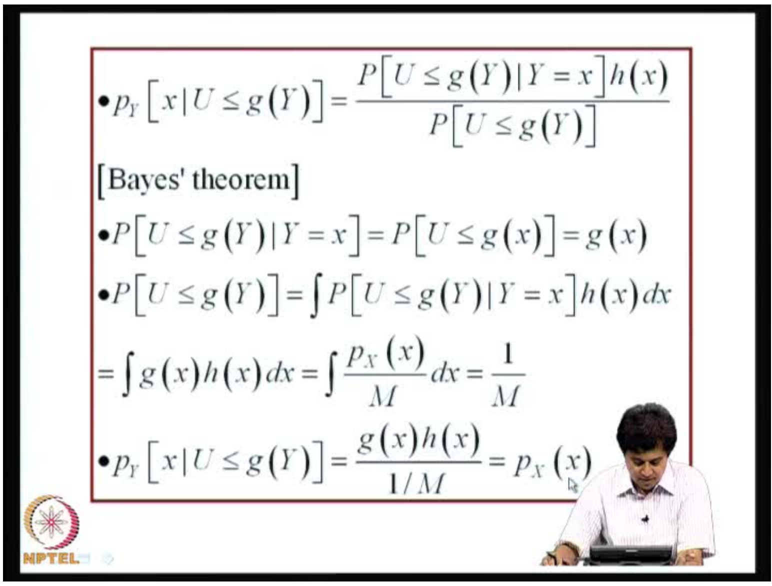
Let $U \sim \mathbf{U}[0,1]$ & $Y \sim h(y)$.

$\Rightarrow p_Y[x \mid U \leq g(Y)] = p_X(x)$

.

Now, how do you translate this idea in to a working algorithm? Now, I will not get in to too many details, I will illustrate this by few examples and the simple explanation. So, let X be a random variable with pdf p X of x and x belong to I and interval I. Now, what we do is we represent p X of x as a number M into g of x into h of x, where M is a number greater than 1 and g of x takes values between 0 and 1 and h of x is a valid probability density function. Now, let U be between 0 and 1 and Y be the random variable with pdf h of y and h of y should be such, that you should be able to sample Y from h of y. Now, p Y of x condition on U less than or equal to g of Y can be shown to be p X of x.

(Refer Slide Time: 26:34)



- $p_Y\left[x \mid U \le g(Y)\right] = \dfrac{P\left[U \le g(Y) \mid Y = x\right] h(x)}{P\left[U \le g(Y)\right]}$

[Bayes' theorem]

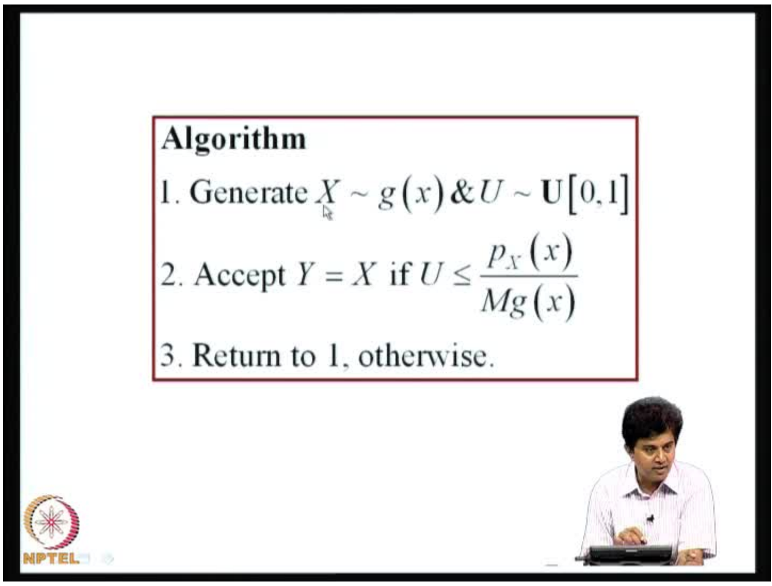
- $P\left[U \le g(Y) \mid Y = x\right] = P\left[U \le g(x)\right] = g(x)$
- $P\left[U \le g(Y)\right] = \int P\left[U \le g(Y) \mid Y = x\right] h(x)\,dx$

$= \int g(x)h(x)\,dx = \int \dfrac{p_X(x)}{M}\,dx = \dfrac{1}{M}$

- $p_Y\left[x \mid U \le g(Y)\right] = \dfrac{g(x)h(x)}{1/M} = p_X(x)$

How do you show that? You use Bayes’ theorem and consider p Y of x condition on U less than or equal to g of Y, by using Bayes’ theorem I can write it like this, I mean, use conditional probability and then rewrite this in a slightly different form. Now, we consider now here, there is one probability U of U less than equal to g of Y condition Y equal to x. Let us consider that here and this is nothing but g of x because U is uniformly distributed; so this is g of x. Similarly, U less than or equal to g of Y is, if you carry out these analysis, you can show, that this is equal to 1 by M. Now, therefore, you substitute this and this into this, we can show, that this conditional probability density function is nothing but p X of x.

(Refer Slide Time: 27:23)



Algorithm

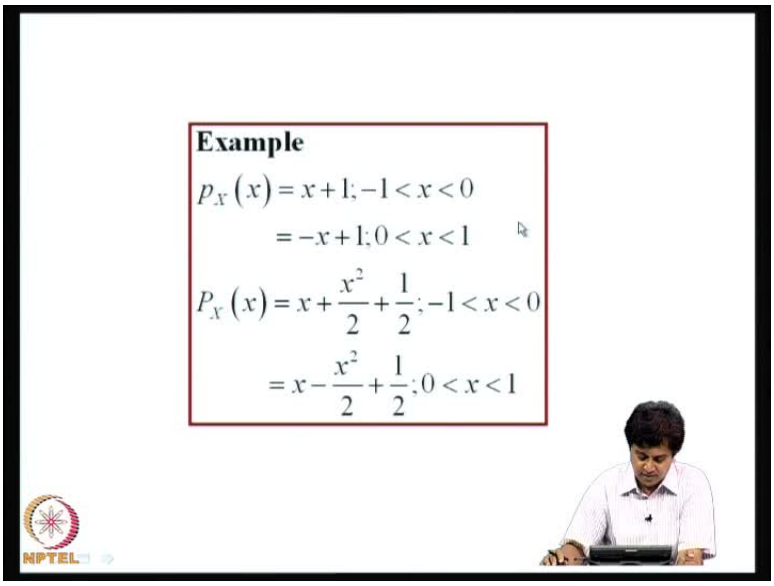
1. Generate $X \sim g(x)$ & $U \sim \mathbf{U}[0,1]$

2. Accept $Y = X$ if $U \leq \dfrac{p_X(x)}{Mg(x)}$

3. Return to 1, otherwise.

So, the algorithm is, so you generate X from g of x and U from 0 to 1 and we accept Y equal to X, if U is less than or equal to p X of x divided by Mg of x, or you go back and do this calculation again.

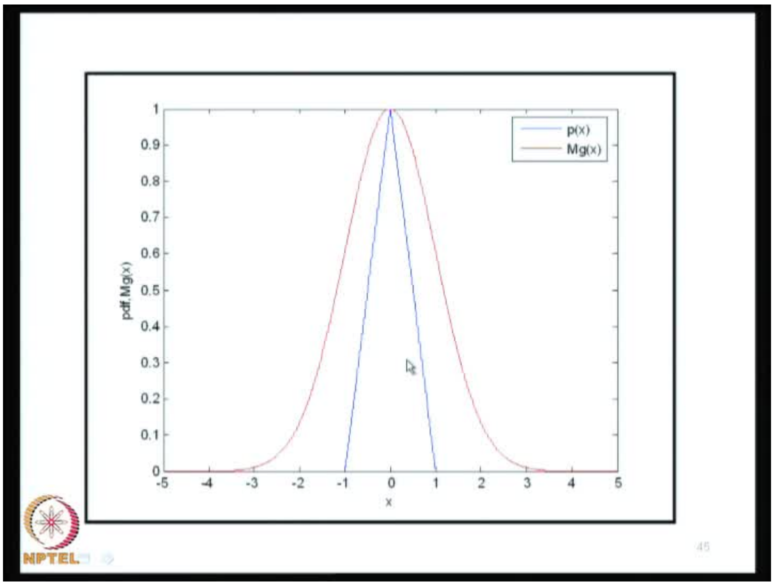
(Refer Slide Time: 27:41)



Example

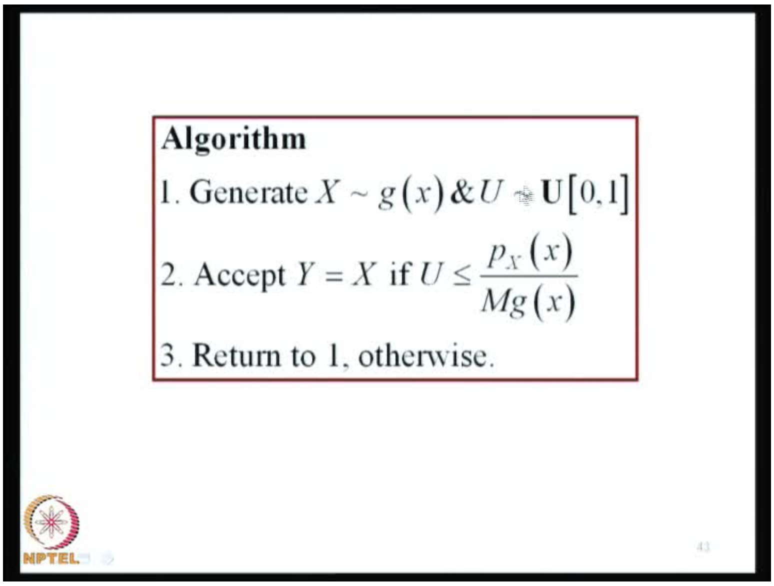
$$p_X(x) = x+1; -1 < x < 0$$
$$= -x+1; 0 < x < 1$$
$$P_X(x) = x + \frac{x^2}{2} + \frac{1}{2}; -1 < x < 0$$
$$= x - \frac{x^2}{2} + \frac{1}{2}; 0 < x < 1$$

(Refer Slide Time: 27:52)



Now, I will give an example. Let us consider a probability density function, which is a triangle between 0 and minus 1 to plus 1, that means, I am looking at this blue line, I want to draw random numbers or generate random numbers whose probability density function matches with this. So, what I do is I construct this red line, which envelopes this density function and that will be my m, this Mg x.

(Refer Slide Time: 27:23)



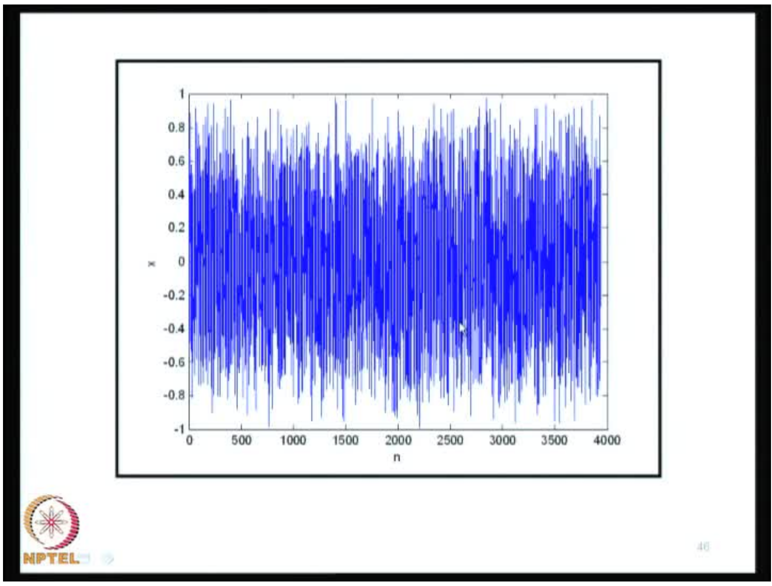
Algorithm

1. Generate $X \sim g(x)$ & $U \sim \mathbf{U}[0,1]$

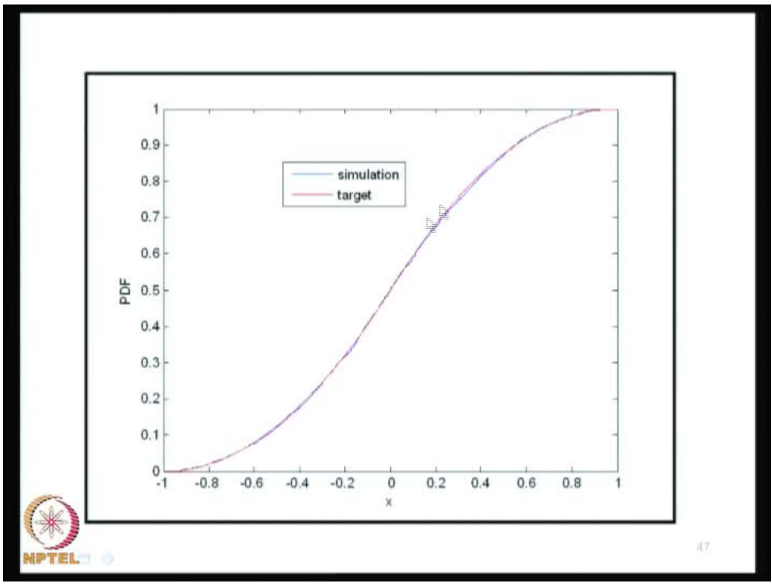
2. Accept $Y = X$ if $U \leq \dfrac{p_X(x)}{Mg(x)}$

3. Return to 1, otherwise.

(Refer Slide Time: 28:23)

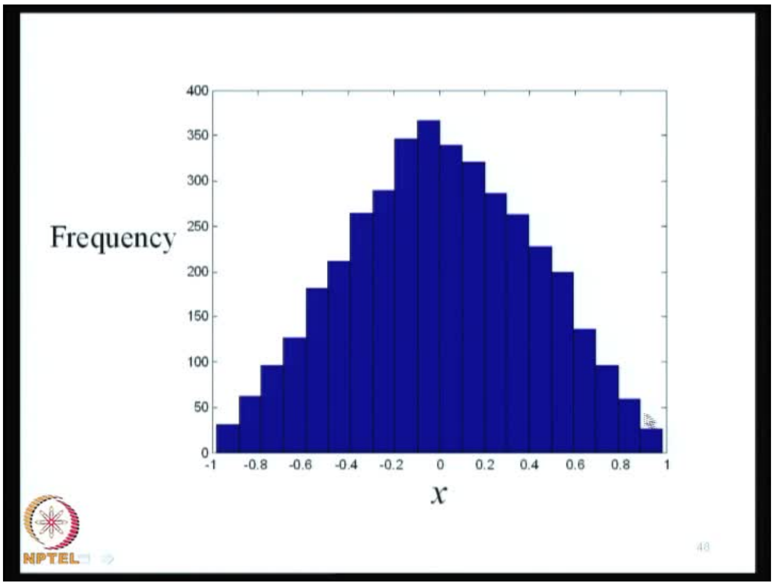


(Refer Slide Time: 28:32)

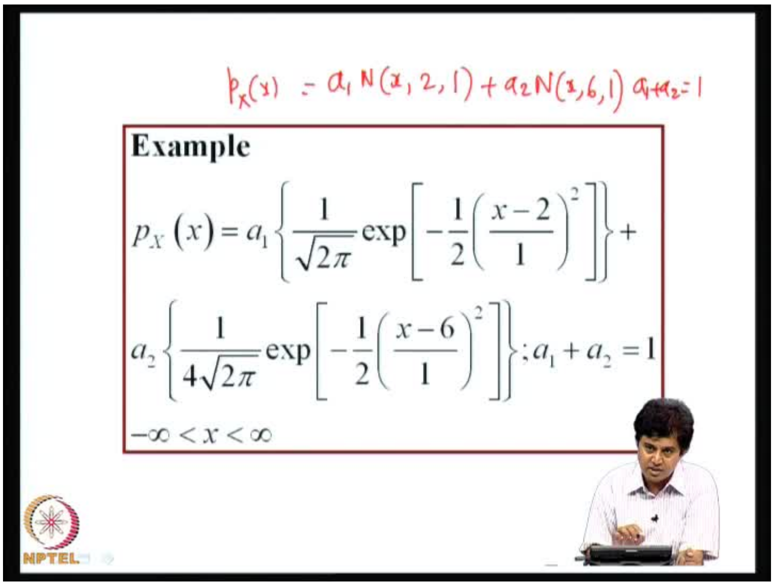


Now, I will follow this algorithm and if I do this exercise I get these numbers. This is characterless, I mean, nothing can be inferred from this except that they are lying between minus 1 and plus 1, but if I draw the probability distribution function, blue is a simulation, red is the target and this again, is not quite revealing. But if you draw the histogram, you can see, that the numbers are along this triangle.

(Refer Slide Time: 28:42)



(Refer Slide Time: 28:47)



$p_X(x) = a_1 N(x, 2, 1) + a_2 N(x, 6, 1) \quad a_1 + a_2 = 1$

Example

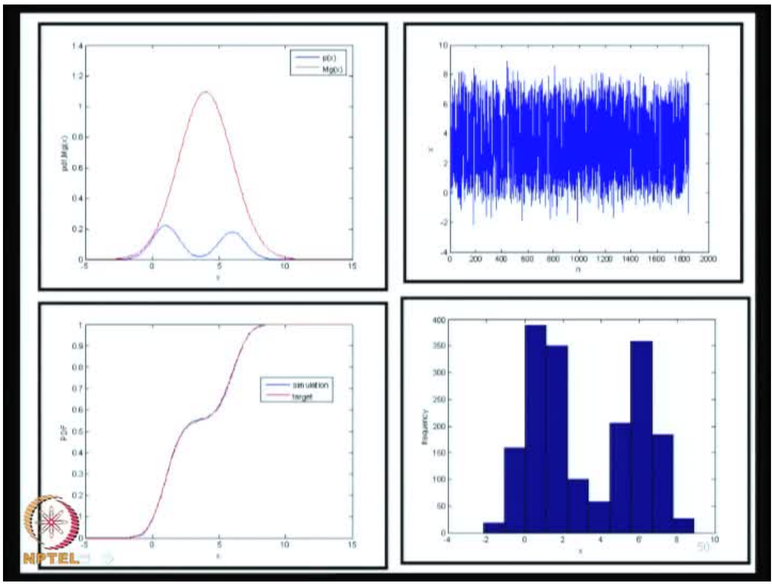
$$p_X(x) = a_1\left\{\frac{1}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{x-2}{1}\right)^2\right]\right\} +$$

$$a_2\left\{\frac{1}{4\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{x-6}{1}\right)^2\right]\right\}; a_1 + a_2 = 1$$

$-\infty < x < \infty$

Another example, if you want to draw random numbers from a pdf, which is made up of say 2, this is a 1 normal pdf x with mean 2 and standard deviation 1 plus a 2 normal x, again mean 6 and 1, this is my p X of x. I (( )) a 1 plus a 2 is 1, you can easily see that inverting this for finding X is quite difficult. You can find out p X of x in terms of the normal probability distribution function, but that itself is not very easy to invert. So, how do you proceed? So, we can use this method.

(Refer Slide Time: 29:32)

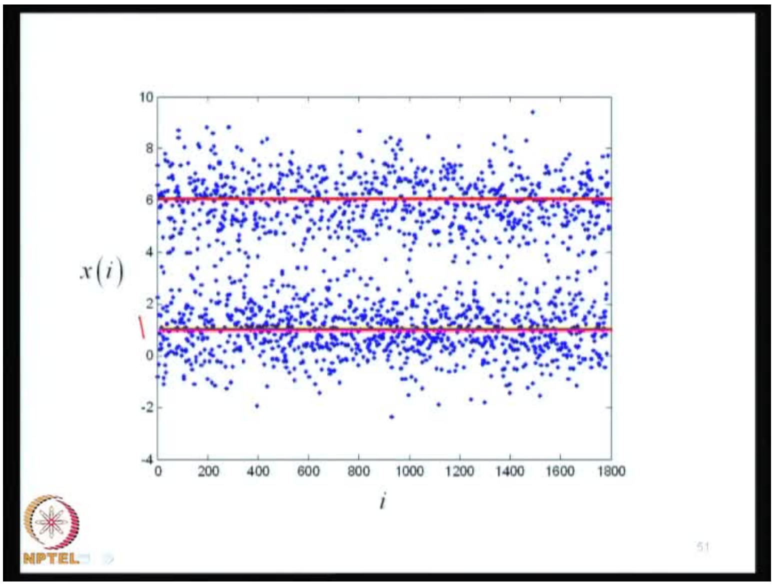


.

So, what we did, what we do is, this blue line is the probability density function, which is the target and I select this M into g of x to be a normal like density function and that m is selected, so that this envelopes the target density function; that is very important because the theory works only under that situation.

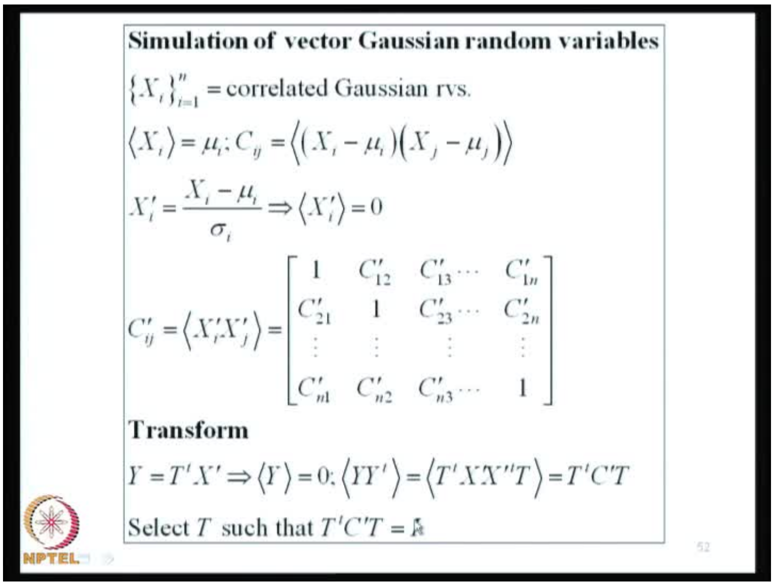
So, this graph shows the samples again, which are not very revealing, but the probability distribution function, you can see, that the agreement between blue line simulation, red line is the target, is quite good here. And this is the histogram, you can easily see, that there are 2 modes and numbers are clustering around the one point here and another point here, this is 1 and this is 6; so, this is 1 and this is 6.

(Refer Slide Time: 30:23)



So, if we plot x of i versus i, you can see that the numbers are clustering around 1 and 6. So, these clusters correspond to the modes that we are seeing here. So, without inverting the probability density function or the probability distribution function, we were able to simulate random numbers. So, this is, this method is applicable when the form of the probability distribution function becomes too complex and we face difficulties in inverting. I have, I have talked about scalar random variables, how about vector random variables?

(Refer Slide Time: 30:57)



Simulation of vector Gaussian random variables

$\{X_i\}_{i=1}^{n}$ = correlated Gaussian rvs.

$\langle X_i \rangle = \mu_i; C_{ij} = \langle (X_i - \mu_i)(X_j - \mu_j) \rangle$

$X_i' = \dfrac{X_i - \mu_i}{\sigma_i} \Rightarrow \langle X_i' \rangle = 0$

$$C_{ij}' = \langle X_i' X_j' \rangle = \begin{bmatrix} 1 & C_{12}' & C_{13}' \cdots & C_{1n}' \\ C_{21}' & 1 & C_{23}' \cdots & C_{2n}' \\ \vdots & \vdots & \vdots & \vdots \\ C_{n1}' & C_{n2}' & C_{n3}' \cdots & 1 \end{bmatrix}$$

Transform

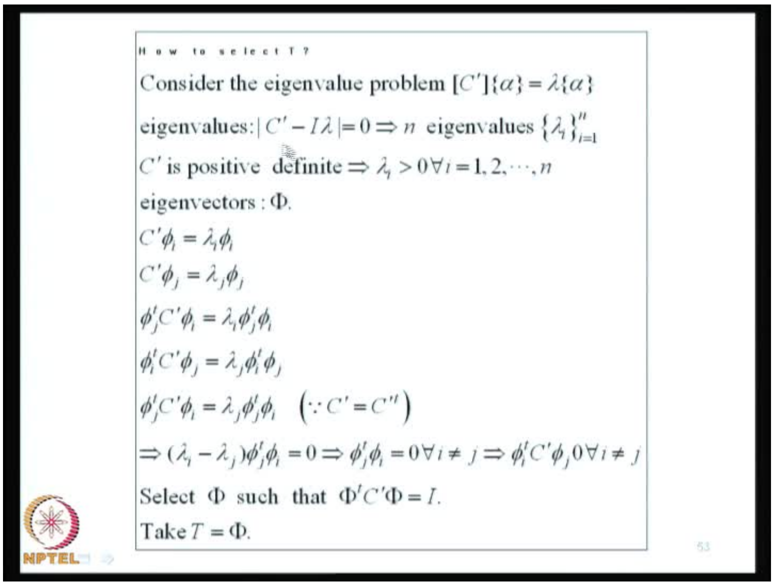
$Y = T'X' \Rightarrow \langle Y \rangle = 0; \langle YY' \rangle = \langle T'X'X''T \rangle = T'C'T$

Select T such that $T'C'T = I$

So, we will start with simulation of vector Gaussian random variables. Suppose, I have a sequence of n random variables X i, which are correlated Gaussian random variables and how to simulate? If they are uncorrelated, I have case of n scalar random variable, each one I can simulate separately and since they are the independent, it does not really matter. But now they are correlated, so mean of X i mu i and the covariance is (X i minus mu i) (X j minus mu j) C ij. Now, what I do is I first remove the mean and make the standard deviation as 1. By using this transformation, X i prime is X i minus mu i divided by sigma i. Obviously, the mean of X i prime is 0 and if you write the covariance of this X i prime, you can see that this would be nothing but the correlation coefficient matrix of X i and along the diagonal we get 1 and this is symmetric matrix.

Now, our objective is to therefore, simulate X i prime whose mean is 0 and correlation covariance matrix is this fully populated square matrix symmetric, etcetera. Now, the strategy we follow is we use a transformation, we propose a transformation Y is T transpose X prime. Now, mean is preserved through this transformation, but covariance get discarded, so the covariance expectation of YY transpose would be T transpose C prime. Though T is yet to be selected and I can always select T, show that T transpose C prime T is a identity matrix. So, how do you get the T matrix? We study the Eigen values of C prime matrix.

(Refer Slide Time: 32:44)



How to select T?

Consider the eigenvalue problem $[C']\{\alpha\} = \lambda\{\alpha\}$

eigenvalues: $|C' - I\lambda| = 0 \Rightarrow n$ eigenvalues $\{\lambda_i\}_{i=1}^{n}$

C' is positive definite $\Rightarrow \lambda_i > 0 \forall i = 1, 2, \cdots, n$

eigenvectors : Φ.

$C'\phi_i = \lambda_i\phi_i$

$C'\phi_j = \lambda_j\phi_j$

$\phi_j^t C'\phi_i = \lambda_i\phi_j^t\phi_i$

$\phi_i^t C'\phi_j = \lambda_j\phi_i^t\phi_j$

$\phi_j^t C'\phi_i = \lambda_j\phi_j^t\phi_i \quad \left(\because C' = C'^t\right)$

$\Rightarrow (\lambda_i - \lambda_j)\phi_j^t\phi_i = 0 \Rightarrow \phi_j^t\phi_i = 0 \forall i \neq j \Rightarrow \phi_i^t C'\phi_j 0 \forall i \neq j$

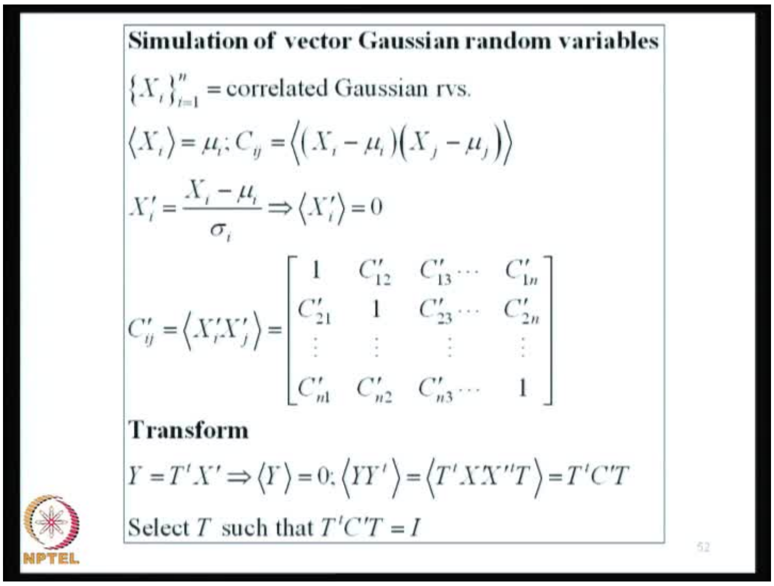
Select Φ such that $\Phi^t C'\Phi = I$.

Take $T = \Phi$.

.

So, we consider the Eigen value problem, C prime alpha equal to lambda alpha and we solve for the n Eigen values by solving this characteristic equation, and corresponding to each one of that, we get an eigenvector. And since, C is positive definite lambda, i will be positive and these Eigen vectors are assembled into a single matrix, which is square matrix. And we have seen in discussion on vibration problems, that these Eigen vectors have orthogonality property and we can show that phi transpose phi i equal to 0 for i not equal to j and phi i transpose C prime phi j equal to 0 for i not equal to j, so that phi transpose C prime phi is I; I is the identity matrix, so that would mean, we can take T to be phi.

(Refer Slide Time: 30:57)



Simulation of vector Gaussian random variables

$\{X_i\}_{i=1}^{n}$ = correlated Gaussian rvs.

$\langle X_i \rangle = \mu_i; C_{ij} = \langle (X_i - \mu_i)(X_j - \mu_j) \rangle$

$X_i' = \dfrac{X_i - \mu_i}{\sigma_i} \Rightarrow \langle X_i' \rangle = 0$

$$C_{ij}' = \langle X_i' X_j' \rangle = \begin{bmatrix} 1 & C_{12}' & C_{13}' \cdots & C_{1n}' \\ C_{21}' & 1 & C_{23}' \cdots & C_{2n}' \\ \vdots & \vdots & \vdots & \vdots \\ C_{n1}' & C_{n2}' & C_{n3}' \cdots & 1 \end{bmatrix}$$

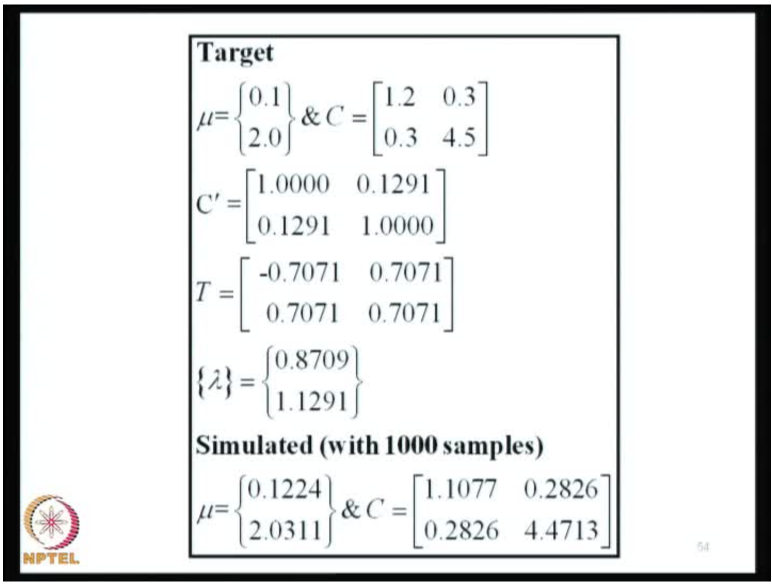
Transform

$Y = T'X' \Rightarrow \langle Y \rangle = 0; \langle YY' \rangle = \langle T'X'X''T \rangle = T'C'T$

Select T such that $T'C'T = I$

So, once I do that, I am able to transform the given random variable. So, there are 2 step transformation - the first one is simple, I remove the mean and divide it by standard deviation, so X i prime has 0 mean and unit standard deviation, and it become non dimensional. So, this is good for us and then I diagonalize the correlation coefficient matrix or the covariance matrix. So, I have find out T by finding the Eigen vector matrix and moment I find that, what I will do is, I will simulate X prime, sorry, simulate Y, which has 0 mean and mutually independent elements. I simulate Y, use this transformation, get X prime and from X prime, I work backwards and get X i. So, the main task here is to diagonalize the covariance matrix (( )) and the building block is our ability to simulate normal random numbers with 0 mean and unit standard, unit standard deviation, for a scalar case.

(Refer Slide Time: 34:32)



Target

$$\mu=\begin{Bmatrix}0.1\\2.0\end{Bmatrix}\&C=\begin{bmatrix}1.2 & 0.3\\0.3 & 4.5\end{bmatrix}$$

$$C'=\begin{bmatrix}1.0000 & 0.1291\\0.1291 & 1.0000\end{bmatrix}$$

$$T=\begin{bmatrix}-0.7071 & 0.7071\\0.7071 & 0.7071\end{bmatrix}$$

$$\{\lambda\}=\begin{Bmatrix}0.8709\\1.1291\end{Bmatrix}$$

Simulated (with 1000 samples)

$$\mu=\begin{Bmatrix}0.1224\\2.0311\end{Bmatrix}\&C=\begin{bmatrix}1.1077 & 0.2826\\0.2826 & 4.4713\end{bmatrix}$$

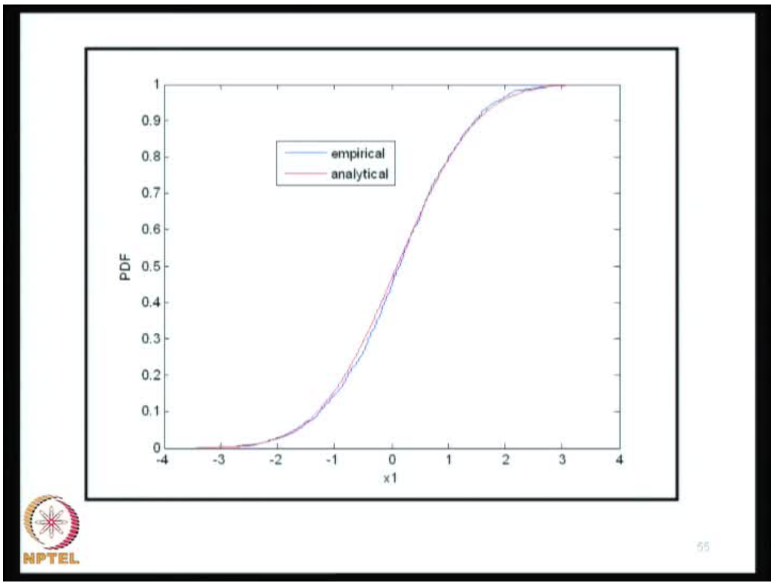
54

.

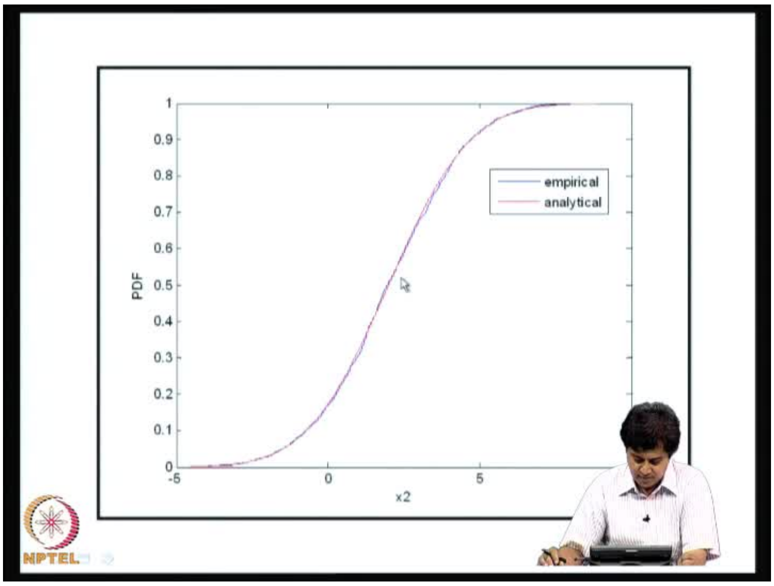
So, we can do this for example, if we have a target mean vector of point 0.1 and 2.0 and covariance: 1.2, 0.3, 0.3, 4.5, I will remove the mean and divide by the standard deviation, I get the covariance matrix is as 11, 0.1291, 0.1291, this is Eigen vector matrix, these are the Eigen values. And then, I have done the simulation in the uncorrelated normal space, use this transformation work backwards and got X, and with thousand samples, I am able to get a mean vector of 0.1224 as against 0.1 and 2.0311 as against 2.0 and covariance of 1.1077 as against 1.2, 0.2826 as against 0.3, 4.47 against 4.5.

Now, again we can do hypothesis test and verify, whether we can accept the hypothesis, that for instance, there marginals are Gaussian, the covariance matrix transform a population whose covariance matrix is this, etcetera. So, for that, of course, you need to know the sampling distribution of covariance, estimates of covariance; that we have not discussed, may be some point in future lectures, I may be able to say few things on that.

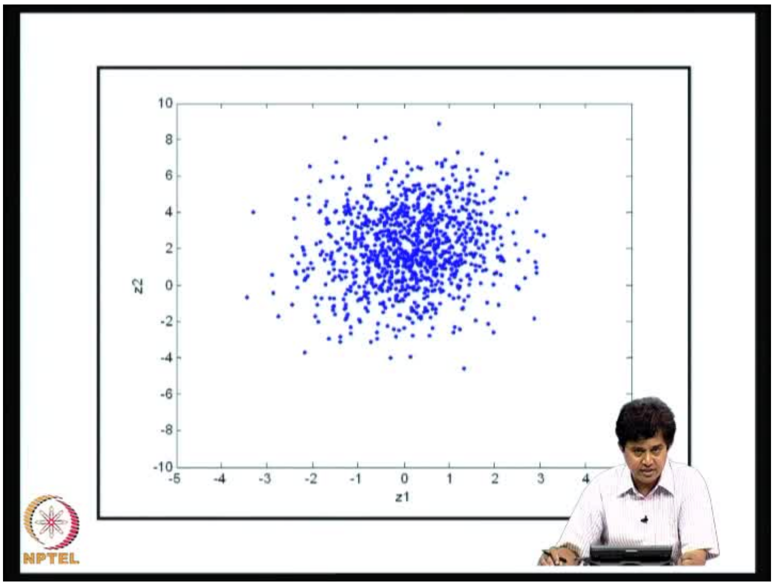
(Refer Slide Time: 35:51)



(Refer Slide Time: 35:59)



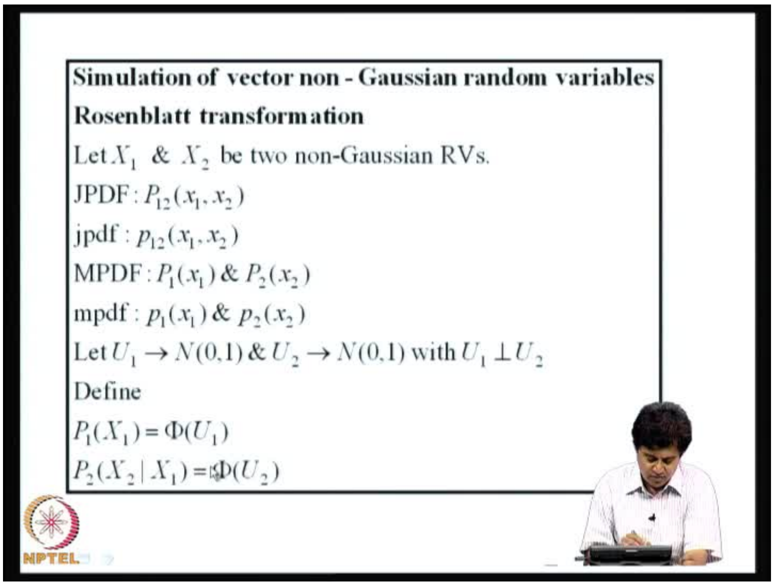
(Refer Slide Time: 36:07)



.

So, this is pdf of X 1; blue is empirical, red is analytical, target. This is for X 2, so in either cases, the comparison between theory and simulation is quite good. And these are the, this is a display of Z 2 versus Z 1; again, we do not see any discernible patterns.

(Refer Slide Time: 36:13)



Simulation of vector non - Gaussian random variables

Rosenblatt transformation

Let X_1 & X_2 be two non-Gaussian RVs.

JPDF : $P_{12}(x_1,x_2)$

jpdf : $p_{12}(x_1,x_2)$

MPDF : $P_1(x_1)$ & $P_2(x_2)$

mpdf : $p_1(x_1)$ & $p_2(x_2)$

Let $U_1 \to N(0,1)$ & $U_2 \to N(0,1)$ with $U_1 \perp U_2$

Define

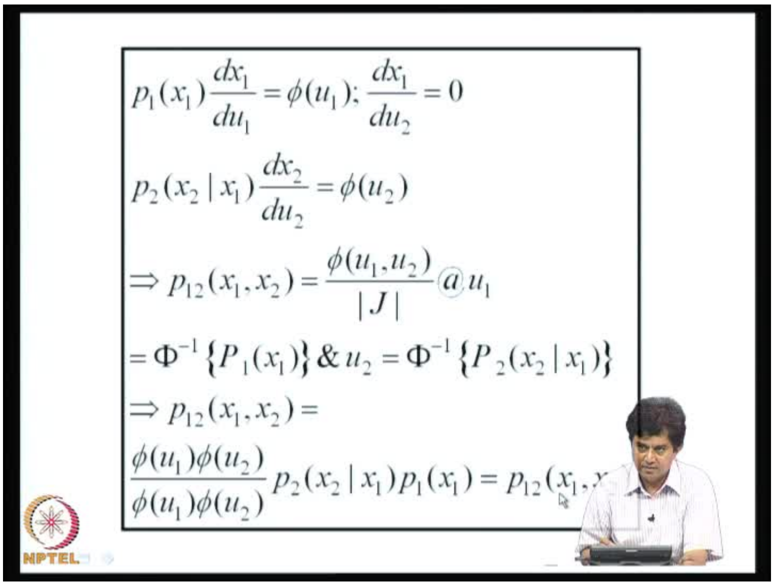
$P_1(X_1)=\Phi(U_1)$

$P_2(X_2 | X_1)=\Phi(U_2)$

.

So, a vector of Gaussian random variables does not pose any major difficulty for us. How about vector of non-Gaussian random variables? So, there are different approaches, one is what is known as Rosenblatt transformation. So, I will illustrate this; this is an extension of method of transformation of pdf’s. So, let X 1 and X 2 be two Gaussian random variables and joint probability distribution function P 1 2 and density function, lower case p 1 2, and marginal pdf's P 1 P 2, marginal density is P 1 P 2, that describes X 1 and X 2.

(Refer Slide Time: 37:15)



$$p_1(x_1)\frac{dx_1}{du_1}=\phi(u_1);\ \frac{dx_1}{du_2}=0$$

$$p_2(x_2\,|\,x_1)\frac{dx_2}{du_2}=\phi(u_2)$$

$$\Rightarrow p_{12}(x_1,x_2)=\frac{\phi(u_1,u_2)}{|J|}\,@\,u_1$$

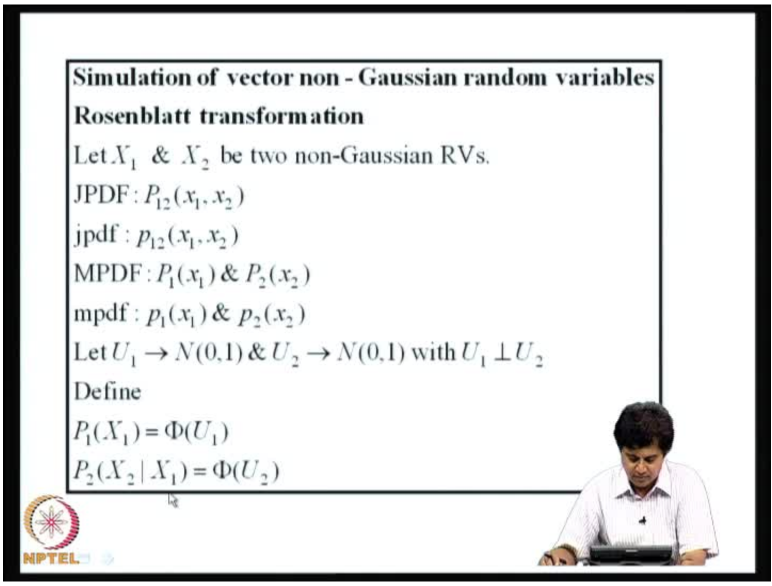
$$=\Phi^{-1}\{P_1(x_1)\}\ \&\ u_2=\Phi^{-1}\{P_2(x_2\,|\,x_1)\}$$

$$\Rightarrow p_{12}(x_1,x_2)=$$

$$\frac{\phi(u_1)\phi(u_2)}{\phi(u_1)\phi(u_2)}p_2(x_2\,|\,x_1)p_1(x_1)=p_{12}(x_1,x$$

Now, let u 1 be distributed normally in 0, with mean 0 and standard deviation 1 and similarly, u 2 normal with mean 0 and 1, and let u 1 and u 2 be independent. So, I begin by defining p 1 x 1 equal to phi u 1 and p 2 of x 2 condition on x 1 is phi u 2. Based on this transformation, we can show that x 1 and x 2 are…, the density function of x 1 and x 2 will be according to the target joint density function. So, to do that, we apply the rules of transformation of random variables and find the jacobian, and we can do this calculation, the details are displayed here, and we get in fact, the joint density between x 1 and x 2 will be indeed, be the target density function.

(Refer Slide Time: 36:13)



Simulation of vector non - Gaussian random variables

Rosenblatt transformation

Let X_1 & X_2 be two non-Gaussian RVs.

JPDF : $P_{12}(x_1, x_2)$

jpdf : $p_{12}(x_1, x_2)$

MPDF : $P_1(x_1)$ & $P_2(x_2)$

mpdf : $p_1(x_1)$ & $p_2(x_2)$

Let $U_1 \to N(0,1)$ & $U_2 \to N(0,1)$ with $U_1 \perp U_2$

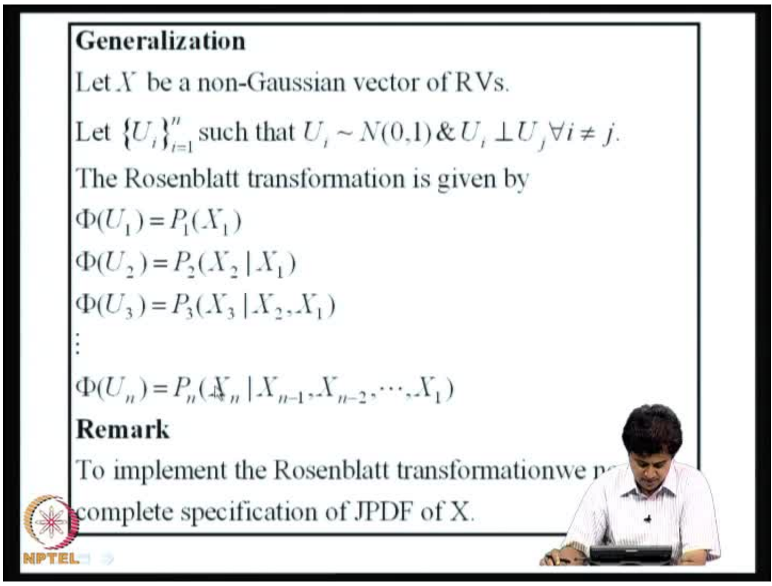
Define

$P_1(X_1) = \Phi(U_1)$

$P_2(X_2 | X_1) = \Phi(U_2)$

So, how do you implement the method? You simulate U 1 and U 2 and you find out X 1, by X 1 equal to P 1 inverse of phi of P 1; then, you find out X 2, X 1 is already simulated for every value of X 1, you come here and find out the inverse of this. So, you will be able to proceed with the implementation.

(Refer Slide Time: 38:06)



Generalization

Let X be a non-Gaussian vector of RVs.

Let $\{U_i\}_{i=1}^{n}$ such that $U_i \sim N(0,1)$ & $U_i \perp U_j \forall i \neq j$.

The Rosenblatt transformation is given by

$\Phi(U_1) = P_1(X_1)$

$\Phi(U_2) = P_2(X_2 | X_1)$

$\Phi(U_3) = P_3(X_3 | X_2, X_1)$

$\vdots$

$\Phi(U_n) = P_n(X_n | X_{n-1}, X_{n-2}, \cdots, X_1)$

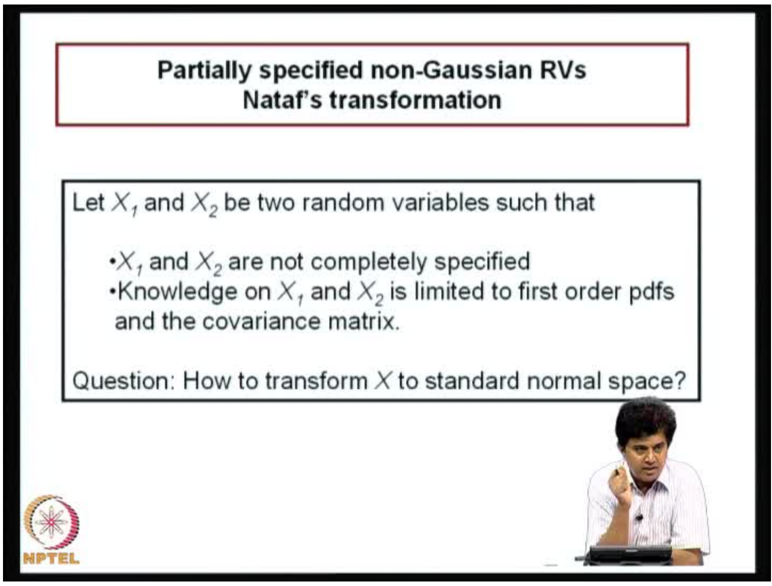
Remark

To implement the Rosenblatt transformationwe n

complete specification of JPDF of X.

Now, this generalization to N random variables is straight forward, but it would involve inversion of increasingly more complicated conditional probability distribution functions. If you can do it, you will able to proceed according to this.

(Refer Slide Time: 38:20)



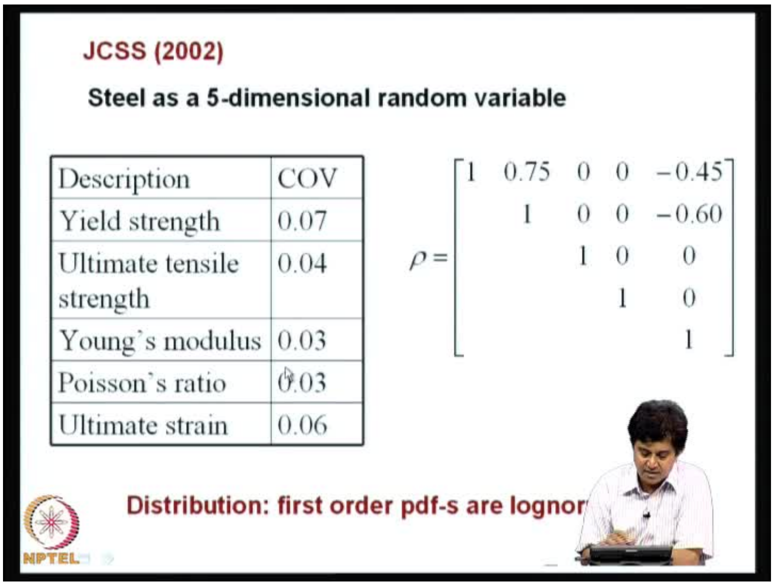
Partially specified non-Gaussian RVs
Nataf's transformation

Let X_1 and X_2 be two random variables such that

•X_1 and X_2 are not completely specified
•Knowledge on X_1 and X_2 is limited to first order pdfs and the covariance matrix.

Question: How to transform X to standard normal space?

(Refer Slide Time: 38:40)



JCSS (2002)

Steel as a 5-dimensional random variable

Description	COV
Yield strength	0.07
Ultimate tensile strength	0.04
Young's modulus	0.03
Poisson's ratio	0.03
Ultimate strain	0.06

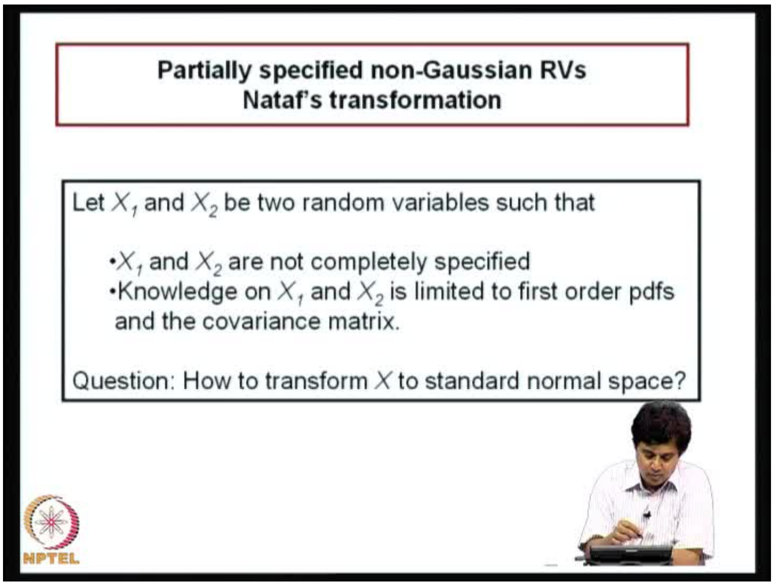
$$\rho = \begin{bmatrix} 1 & 0.75 & 0 & 0 & -0.45 \\ & 1 & 0 & 0 & -0.60 \\ & & 1 & 0 & 0 \\ & & & 1 & 0 \\ & & & & 1 \end{bmatrix}$$

Distribution: first order pdf-s are lognor

Now, most often, we will not be in a position to provide the Nth order joint probability distribution function of random variables in practical applications, what we would be able to provide would be the first order probability distribution function and a covariance matrix. For example, if you see the joint committee on structural safety, there is a document, which has attempted to provide certain canonical distributions for properties of construction material and if you look at the properties of steel according to this document, the five variables of steel as a material, namely: yield, strength, ultimate tensile strength, Young's modulus, Poisson's ratio and ultimate strain, the coefficient of variation is specified. Mean can be taken from your test coupons whatever, but the coefficient of variation is this and a correlation coefficient matrix is given here and also stipulated is a fact, that the first order probability density function for each one of these variables is lognormal.

Now, I want to simulate samples of these 5 random quantities, which have this first order pdf, this coefficient of variation and this correlation coefficient matrix. Obviously, as a set of 5 random variables, the information that is provided here would not lead to a complete specification of 5-dimensional random variables, for that you need a 5-dimensional probability density function, but most often, as I was telling in practical application, the kind of information we will have, will be of this kind.

(Refer Slide Time: 38:20)



Partially specified non-Gaussian RVs
Nataf's transformation

Let X_1 and X_2 be two random variables such that

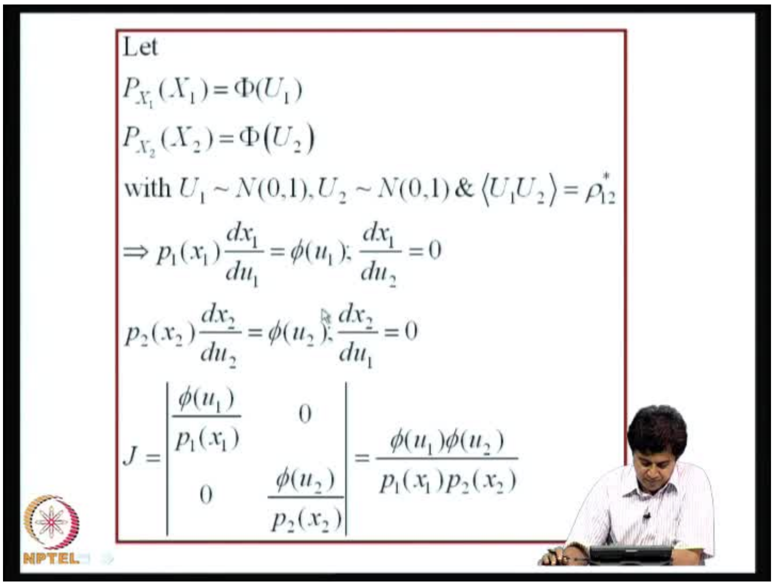
- X_1 and X_2 are not completely specified
- Knowledge on X_1 and X_2 is limited to first order pdfs and the covariance matrix.

Question: How to transform X to standard normal space?

.

So, how do you proceed? So, we start with a case of 2 random variables. Let X 1 and X 2 to be 2 random variables, such that X 1 and X 2 are not completely specified, knowledge on X 1 and X 2 is limited to first order pdfs and the covariance matrix. Now, how do you simulate X 1 and X 2? So, the question is if you transform X 1 and X 2 to standard normal space, you can start with standard normal random variables and transform back to the required X 1, X 2 random variables.

(Refer Slide Time: 40:25)



Let

$P_{X_1}(X_1) = \Phi(U_1)$

$P_{X_2}(X_2) = \Phi(U_2)$

with $U_1 \sim N(0,1), U_2 \sim N(0,1)$ & $\langle U_1 U_2 \rangle = \rho_{12}^*$

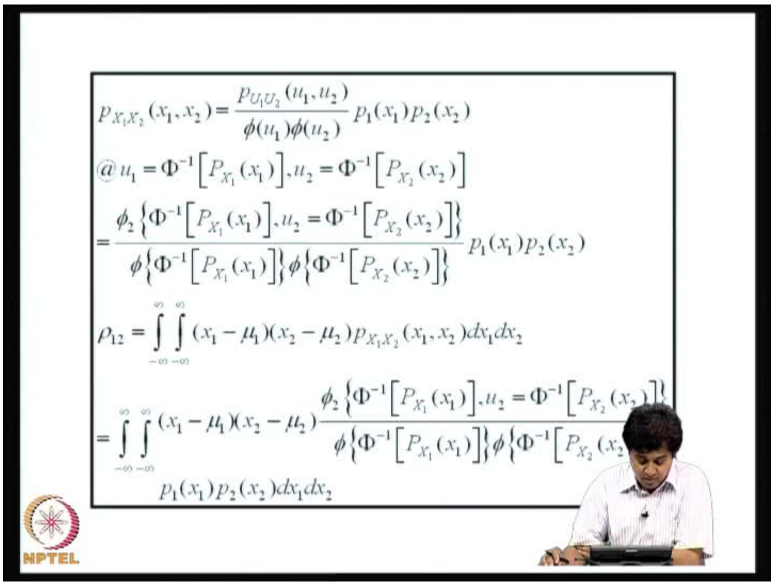
$\Rightarrow p_1(x_1)\dfrac{dx_1}{du_1} = \phi(u_1); \dfrac{dx_1}{du_2} = 0$

$p_2(x_2)\dfrac{dx_2}{du_2} = \phi(u_2); \dfrac{dx_2}{du_1} = 0$

$$J = \begin{vmatrix} \dfrac{\phi(u_1)}{p_1(x_1)} & 0 \\ 0 & \dfrac{\phi(u_2)}{p_2(x_2)} \end{vmatrix} = \frac{\phi(u_1)\phi(u_2)}{p_1(x_1)p_2(x_2)}$$

.

(Refer Slide Time: 41:11)



$$p_{X_1X_2}(x_1,x_2) = \frac{p_{U_1U_2}(u_1,u_2)}{\phi(u_1)\phi(u_2)} p_1(x_1)p_2(x_2)$$

$@ u_1 = \Phi^{-1}[P_{X_1}(x_1)], u_2 = \Phi^{-1}[P_{X_2}(x_2)]$

$$= \frac{\phi_2\{\Phi^{-1}[P_{X_1}(x_1)], u_2 = \Phi^{-1}[P_{X_2}(x_2)]\}}{\phi\{\Phi^{-1}[P_{X_1}(x_1)]\}\phi\{\Phi^{-1}[P_{X_2}(x_2)]\}} p_1(x_1)p_2(x_2)$$

$$\rho_{12} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x_1-\mu_1)(x_2-\mu_2)p_{X_1X_2}(x_1,x_2)dx_1dx_2$$

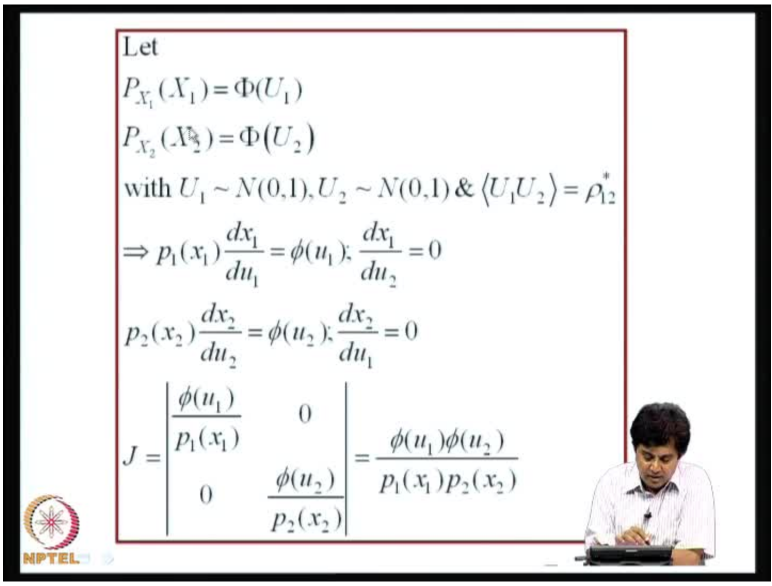
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x_1-\mu_1)(x_2-\mu_2)\frac{\phi_2\{\Phi^{-1}[P_{X_1}(x_1)], u_2 = \Phi^{-1}[P_{X_2}(x_2)]\}}{\phi\{\Phi^{-1}[P_{X_1}(x_1)]\}\phi\{\Phi^{-1}[P_{X_2}(x_2}$$

$p_1(x_1)p_2(x_2)dx_1dx_2$

.

So, here what we do is, we define P X 1 of X 1 is phi of U 1, P X 2 of X 2 is phi of U 2 where U 1 and U 2 are iid normal, mean 0, standard deviation 1; not iid, sorry, they have correlation coefficient rho 1 2 star. They are not iid’s, they have 0 mean and unit standard deviation, but they are correlated, this correlation is unknown, what I know is a correlation between X 1 and X 2. So, I need to develop a method to find correlation between U 1 and U 2, so that the correlation between X 1 and X 2 will meet the specified target value. So, again we go through the rules of transformation of random variables and we can indeed show that the, as far as probability density functions or marginal are concerned, this transformation leads to satisfactory results; there is no problem there.

(Refer Slide Time: 40:25)



Let

$P_{X_1}(X_1)=\Phi(U_1)$

$P_{X_2}(X_2)=\Phi(U_2)$

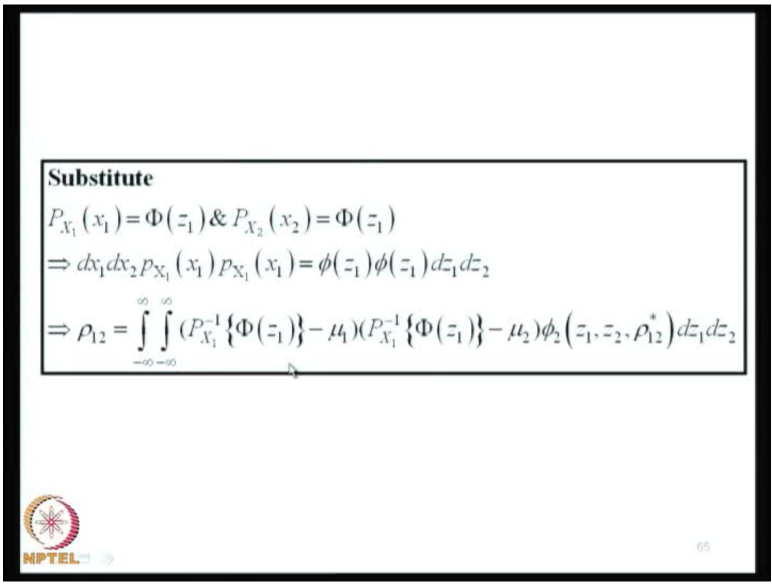
with $U_1 \sim N(0,1), U_2 \sim N(0,1)$ & $\langle U_1U_2\rangle=\rho_{12}^*$

$\Rightarrow p_1(x_1)\dfrac{dx_1}{du_1}=\phi(u_1); \dfrac{dx_1}{du_2}=0$

$p_2(x_2)\dfrac{dx_2}{du_2}=\phi(u_2); \dfrac{dx_2}{du_1}=0$

$$J=\begin{vmatrix}\dfrac{\phi(u_1)}{p_1(x_1)} & 0 \\ 0 & \dfrac{\phi(u_2)}{p_2(x_2)}\end{vmatrix}=\frac{\phi(u_1)\phi(u_2)}{p_1(x_1)p_2(x_2)}$$

(Refer Slide Time: 41:45)



Substitute

$P_{X_1}(x_1)=\Phi(z_1)$ & $P_{X_2}(x_2)=\Phi(z_1)$

$\Rightarrow dx_1dx_2p_{X_1}(x_1)p_{X_1}(x_1)=\phi(z_1)\phi(z_1)dz_1dz_2$

$$\Rightarrow \rho_{12}=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(P_{X_1}^{-1}\{\Phi(z_1)\}-\mu_1)(P_{X_1}^{-1}\{\Phi(z_1)\}-\mu_2)\phi_2(z_1,z_2,\rho_{12}^*)dz_1dz_2$$

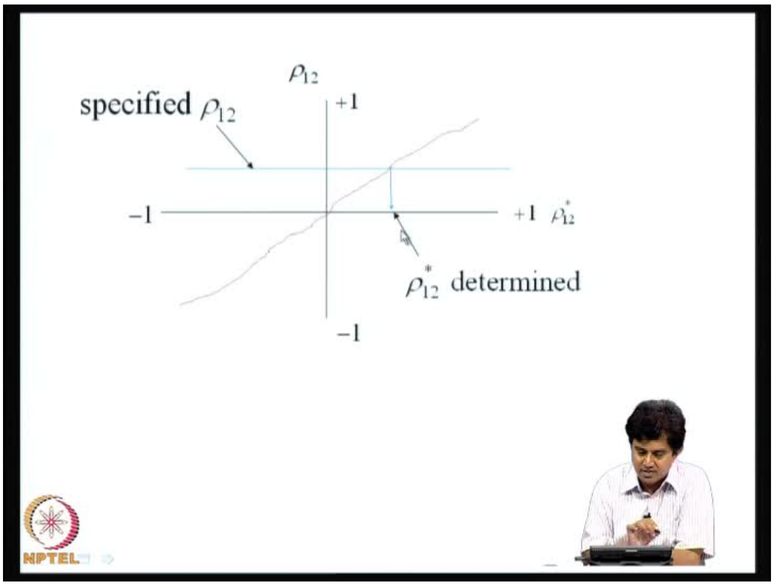
65

But the question of finding rho 1 2 star, that means, we will be, once I implement this transformation, U 1 and U 2 will have this correlation coefficient. So, to simulate U 1 U 2, I need to know this. So, how do we proceed? So, what we do is, we write an expression for rho 1 2 star using this transformation. So, if I do that, I get, it is fairly involved, but one could easily digest this, it is tedious, but not very complicated.

So, if you do this, you will get an equation for rho 1 2, which is given the correlation coefficient between x 1 and x 2 and rho 1 2 star, which is the correlation coefficient between u 1 and u 2, which is not known. So, all other parameters in these equation are known and we need to solve for rho 1 2 star. So, this is an integral equation and it looks quite formidable to solve, but there is very simple strategy to deal with this. So, what we could do is, we know, what we are asked to do is for a given value of rho 1 2, which is on the left hand side, we need to find rho 1 2 star, which is buried inside the integral, but we know that rho 1 2, rho 1 2 star are both bounded between minus 1 and plus 1.

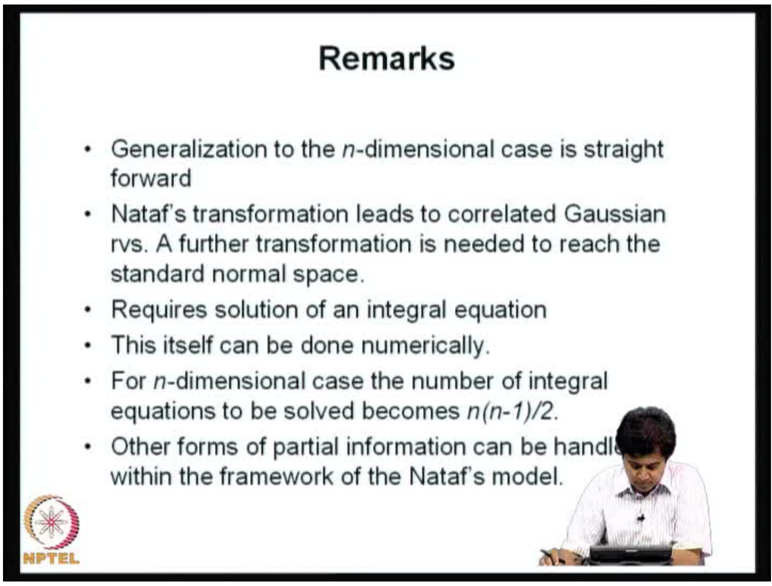
So, what I will do is I will start by assigning values for rho 1 2 star between minus 1 and plus 1 and evaluate this integral and find out, what is rho 1 2 after I do this exercise, for a few complete range of minus 1 to plus 1? It would, if there is a root, it will certainly lead to the one that we want for rho 1 2.

(Refer Slide Time: 42:56)



So, the game is, what we do is, we run the calculation and for every value of rho 1 2 star, we find out rho 1 2 and this is the target value. So, once I do this, I will have the value known in the neighborhood and I can use either better approximation or an interpolation and arrive at the corresponding rho 1 2 star, which is precise. We will have a table of values for rho 1 2 in terms of rho 1 2 star and we can interpolate or do some refined calculations and arrive at this. So, this works quite well in applications.

(Refer Slide Time: 43:31)



Remarks

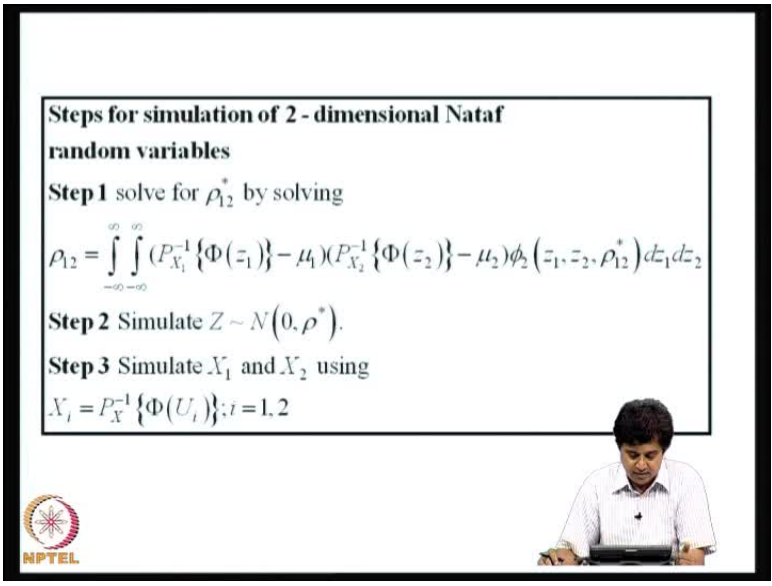
- Generalization to the *n*-dimensional case is straight forward
- Nataf's transformation leads to correlated Gaussian rvs. A further transformation is needed to reach the standard normal space.
- Requires solution of an integral equation
- This itself can be done numerically.
- For *n*-dimensional case the number of integral equations to be solved becomes *n(n-1)/2*.
- Other forms of partial information can be handl within the framework of the Nataf's model.

.

Now, this I have illustrated for two-dimensional case, we can generalize n-dimensional case, so you have to take pairwise. This transformation by the way, is known as Nataf's transformation, so this can be extended to n-dimensional cases, requires a solution of an integral equation. This itself can be done numerically and for n-dimensional, the number of integral equations to be solved becomes n into n minus 1 by 2 and other forms actually, within the framework of this. Suppose, we consider 3 random variables, it may so happen that for a pair of two random variables, I know the joint density functions.

So, what I know, for example could be marginal density of x 1, x 2, x 3, joint density between x 1 and x 3 and covariance matrix between x 1, x 2, x 3. So, we are falling short of complete specification by different way, may we can fall short of the complete specification in many ways, but in each one of the cases the Nataf's model can be used. And whatever is not available can be, I mean, we can simulate, we can transform with the available information the 3 random variables or n random variables to equivalent normal random variable and implement the method.

(Refer Slide Time: 44:51)



Steps for simulation of 2 - dimensional Nataf random variables

Step 1 solve for ρ_{12}^{*} by solving

$$\rho_{12} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (P_{X_1}^{-1}\{\Phi(z_1)\} - \mu_1)(P_{X_2}^{-1}\{\Phi(z_2)\} - \mu_2)\phi_2\left(z_1, z_2, \rho_{12}^{*}\right) dz_1 dz_2$$

Step 2 Simulate $Z \sim N\left(0, \rho^{*}\right)$.

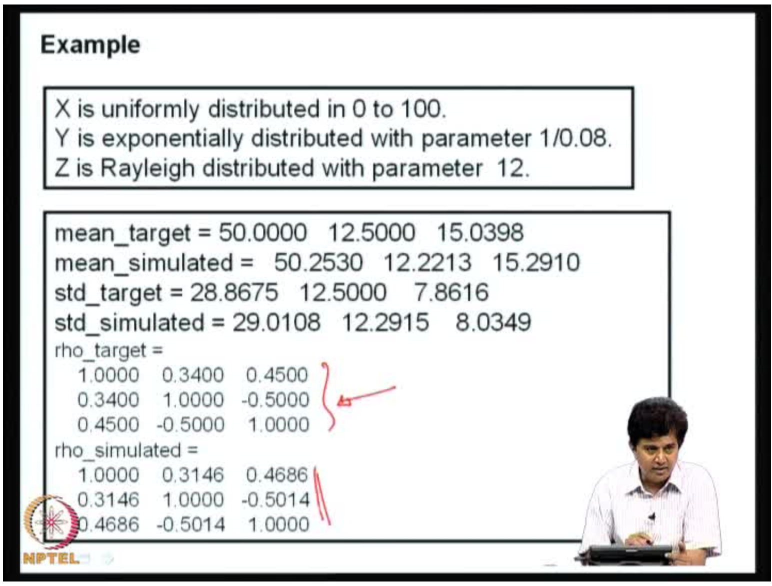
Step 3 Simulate X_1 and X_2 using

$X_i = P_X^{-1}\left\{\Phi\left(U_i\right)\right\}; i = 1, 2$

So, steps for simulation of 2-dimensional Nataf random variables: you solve for rho 1 2 star by solving this, then simulate Z as (0, rho star), then simulate X 1 and X 2 using this transformation. So, to simulate Z as (0, rho star), again we have to find Eigen values analysis of rho star, generate independent normal random variables, apply the transformation and then simulate Z. And once you simulate Z, you have to get into this non-linear transformation and you will be able to get X 1 and X 2.

Now, obviously, this cannot be done with pen and paper, you need to write a computer program for this, then only you will be able to see the working of different steps of this method.

(Refer Slide Time: 45:45)



Example

X is uniformly distributed in 0 to 100.
Y is exponentially distributed with parameter 1/0.08.
Z is Rayleigh distributed with parameter 12.

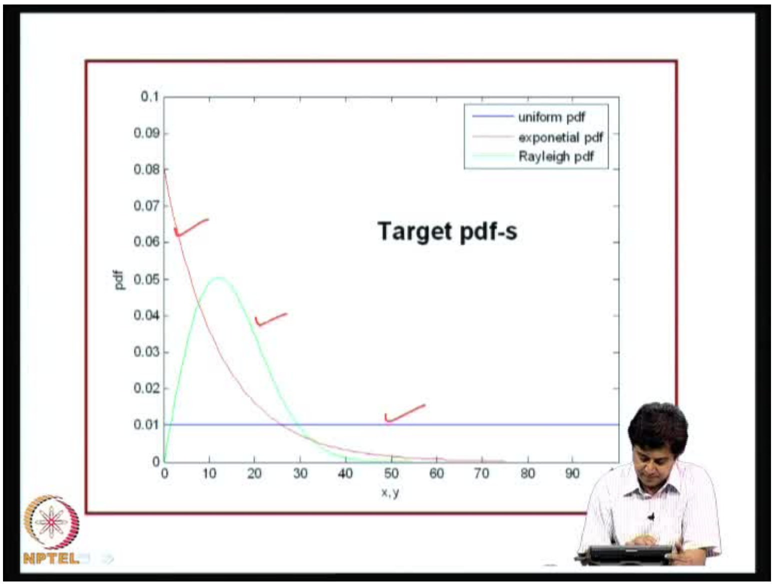
```
mean_target = 50.0000   12.5000   15.0398
mean_simulated =   50.2530   12.2213   15.2910
std_target = 28.8675   12.5000    7.8616
std_simulated = 29.0108   12.2915    8.0349
rho_target =
   1.0000   0.3400   0.4500
   0.3400   1.0000  -0.5000
   0.4500  -0.5000   1.0000
rho_simulated =
   1.0000   0.3146   0.4686
   0.3146   1.0000  -0.5014
   0.4686  -0.5014   1.0000
```

.

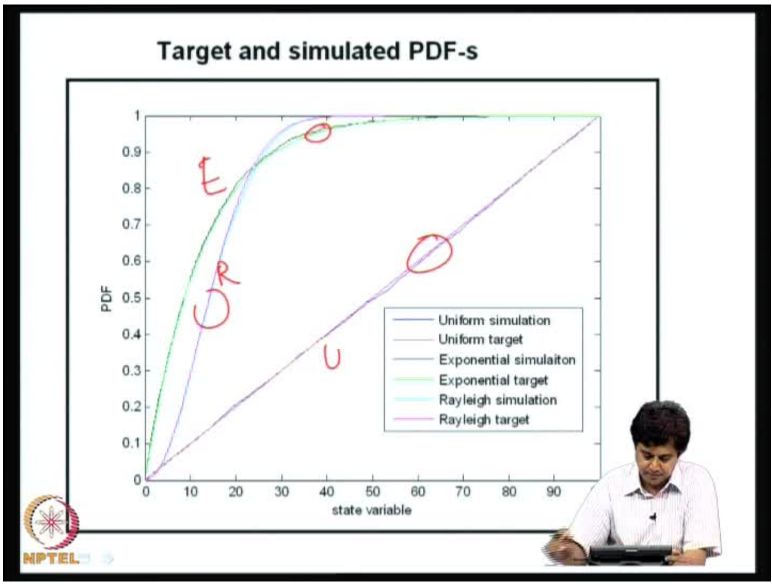
So, let me consider a couple of examples. So, 1st example is X is uniformly distributed between 0 to 100 and Y is exponentially distributed with a parameter 1 divided by point naught 8 and Z is Rayleigh distributed with parameter 12 and the correlation coefficient matrix is displayed here. Now, I want samples of X, Y, Z, so when I simulate I should get uniform distribution for X, exponential distribution for Y and Rayleigh distribution for Z and the correlation coefficient matrix for these 3 random variables should be, should be similar to this. So, we have done this I think with 1000, 5000 samples. Now, the target mean for this is 50, 12.5 and 15.039, that can be calculated from the given values of the system pair. With a simulation, we got a mean of 50.25, 12.22, 15.29 and standard deviation target is 28.86, 12.5, 7.86, etcetera and simulated one is 29 point something, 12.29, 8.0, etcetera.

Again, we can test the hypothesis, whether the samples are drawn from a population with mean 50 or not? We can do a hypothesis test on variance, you can do a chi square test, you can use Gaussian sampling distribution here, you can use chi square distribution, and so on so forth. How about the simulated covariance? What was simulated? This is the target and what was simulated is displayed. There is a broad agreement; there is no reason to suspect that things are wrong.

(Refer Slide Time: 47:38)

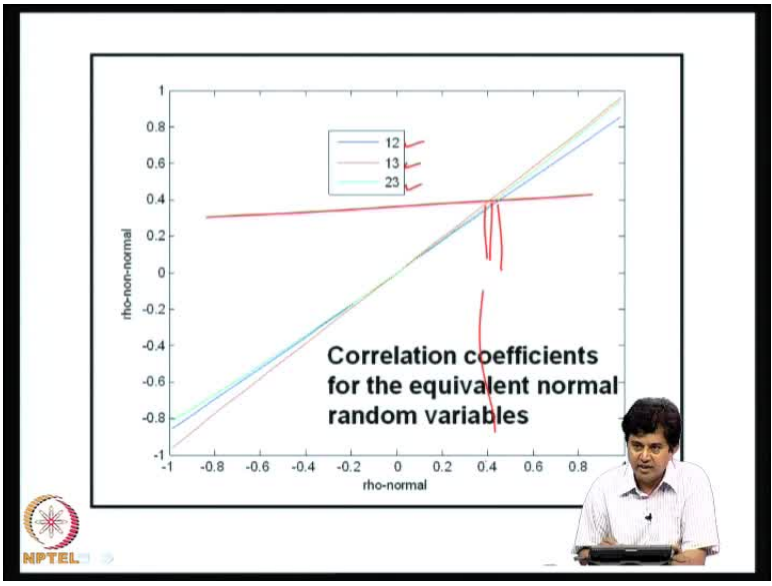

Target pdf-s

(Refer Slide Time: 47:54)


Target and simulated PDF-s

Now, some displays of these results. The 3 target probability density functions are shown here: this blue is a uniform, this is the exponential, this is Rayleigh. The simulated pdfs, so the red and blue is, the simulation results for uniform, the red and blue match well, here, pink and cyan and green and black, the, this is Rayleigh, this is uniform distributed, this is exponential. So, again we can see that the empirical probability distribution function and the target probability distribution function are in good mutual agreement.

(Refer Slide Time: 48:29)

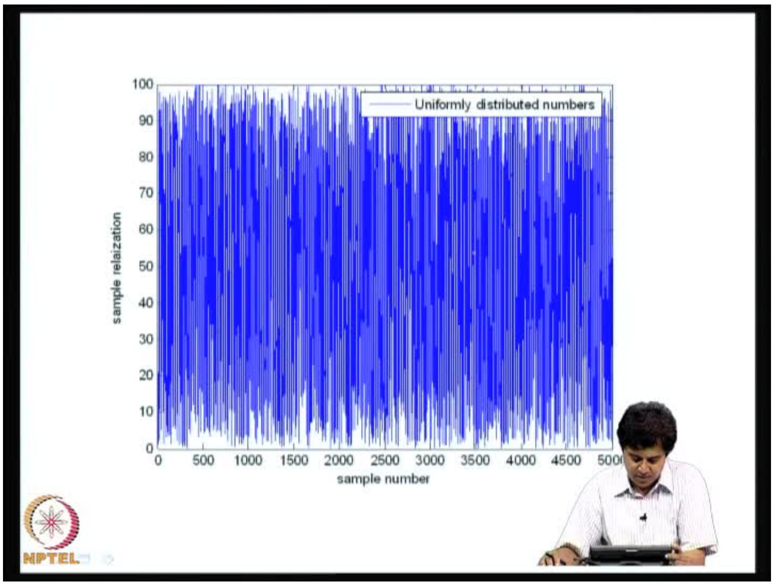


Correlation coefficients for the equivalent normal random variables

.

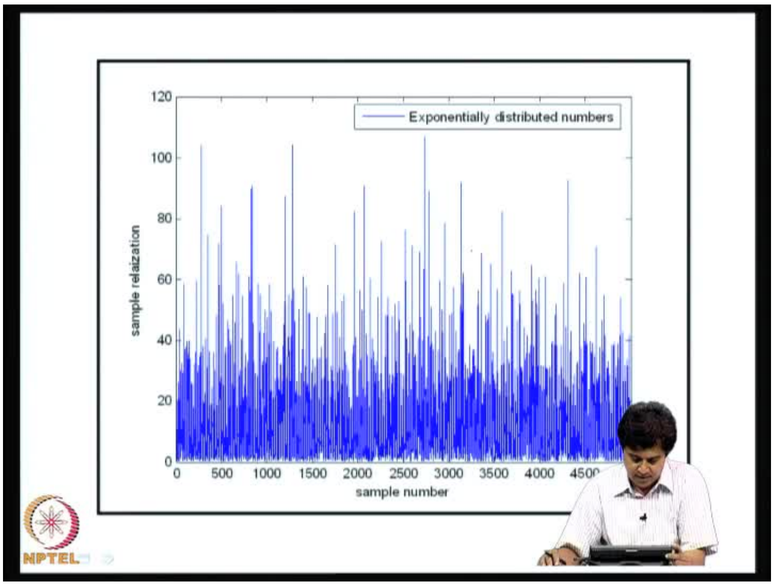
The plot between rho star and rho is shown here. So, this is minus 1 to plus 1 and this minus 1 to plus 1 and this is for 1 2, that is, x 1 x 2, x 1 x 3 and x 2 x 3. So, we can, this is a plot that we get by solving the integral equations for various values of minus 1 to plus 1 for rho star. And this is the rho that we get and we know, that target number here and we can read, of, from here the corresponding rho star. So, this plot has to be generated as a part of the solution to the problem.

(Refer Slide Time: 49:22)

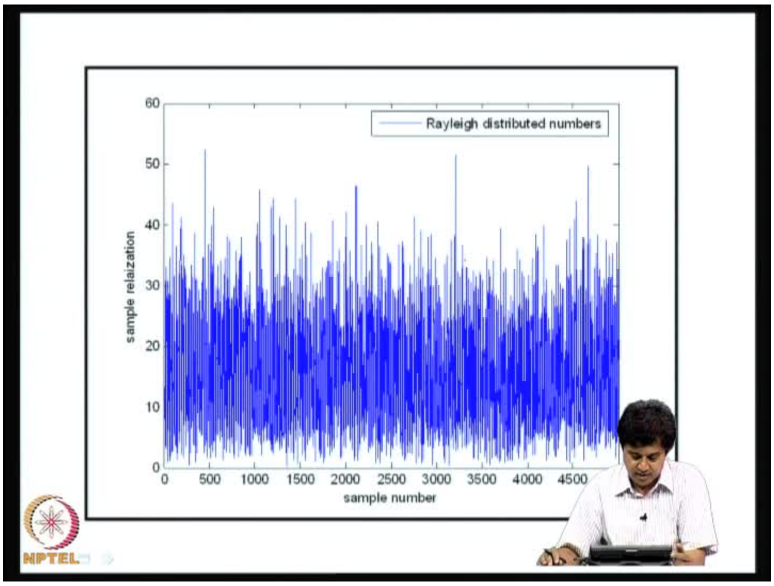


.

(Refer Slide Time: 49:28)

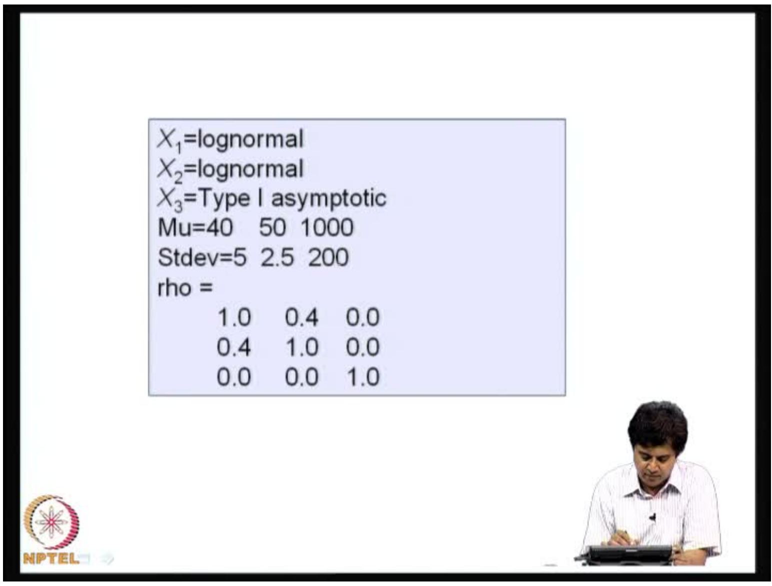


(Refer Slide Time: 49:33)



So, this simulation method works for…, Therefore, this is reasonably complicated problem in simulations and this method works quite well. One more example, this is some displays of samples: this is samples of uniformly distributed numbers, this is exponentially distributed numbers, this is Rayleigh distributed numbers.

(Refer Slide Time: 49:37)



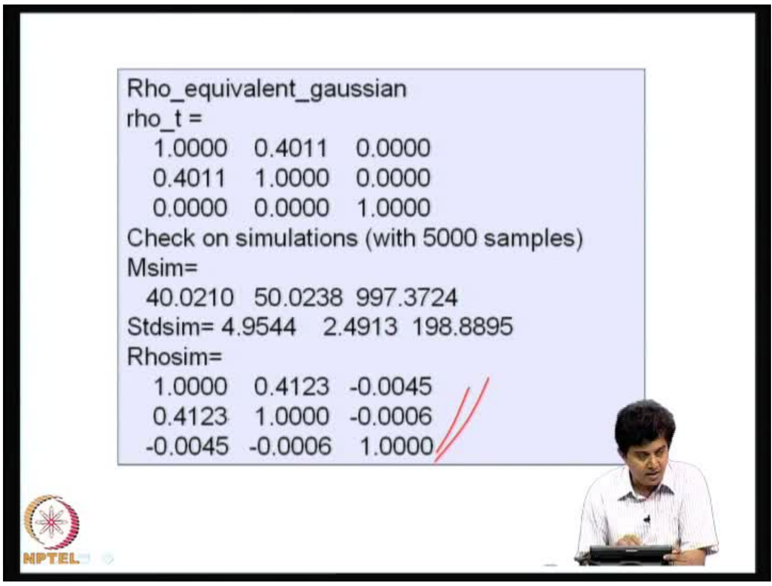
X_1=lognormal
X_2=lognormal
X_3=Type I asymptotic
Mu=40 50 1000
Stdev=5 2.5 200
rho =

```
        1.0    0.4    0.0
        0.4    1.0    0.0
        0.0    0.0    1.0
```

.

I have a one more example: X 1, X 2, X 3. X 1 is lognormal, X 2 is lognormal, X 3 is type one asymptotic and the target mean 40, 50, 1000; standard deviation 5, 2.5, 200 and the correlation coefficient matrix is this.

(Refer Slide Time: 49:56)

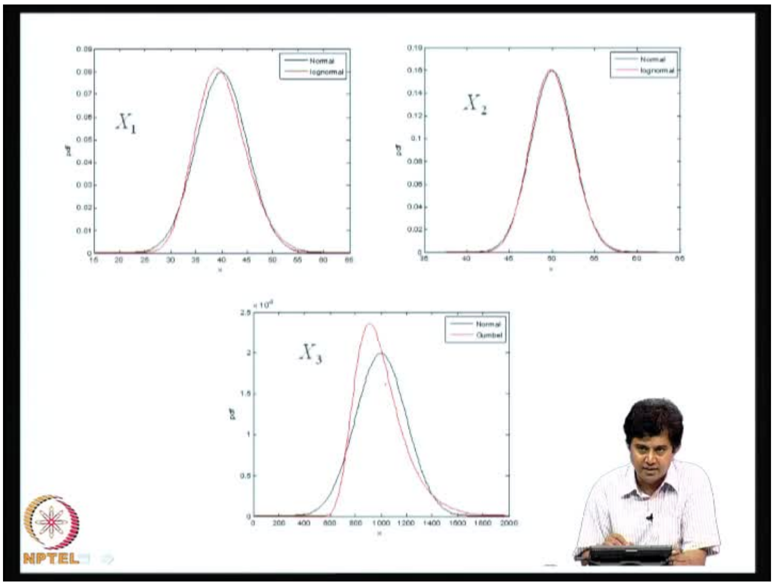


```
Rho_equivalent_gaussian
rho_t =
   1.0000   0.4011   0.0000
   0.4011   1.0000   0.0000
   0.0000   0.0000   1.0000
Check on simulations (with 5000 samples)
Msim=
  40.0210   50.0238  997.3724
Stdsim= 4.9544    2.4913  198.8895
Rhosim=
   1.0000   0.4123   -0.0045
   0.4123   1.0000   -0.0006
  -0.0045  -0.0006    1.0000
```

.

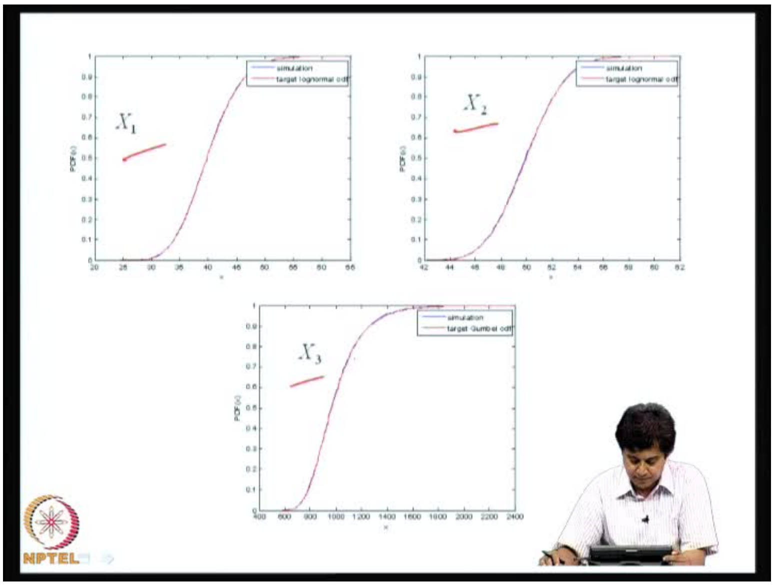
So, we can run through these calculations again. So, this is the equivalent correlation coefficient matrix for the Gaussian random numbers, this is the rho star. Now, we do checks, the mean that we got is 40.02, 50.02 with 5000 samples and 997 and then these things should be compared with 40, 50, 1000.

(Refer Slide Time: 50:31)



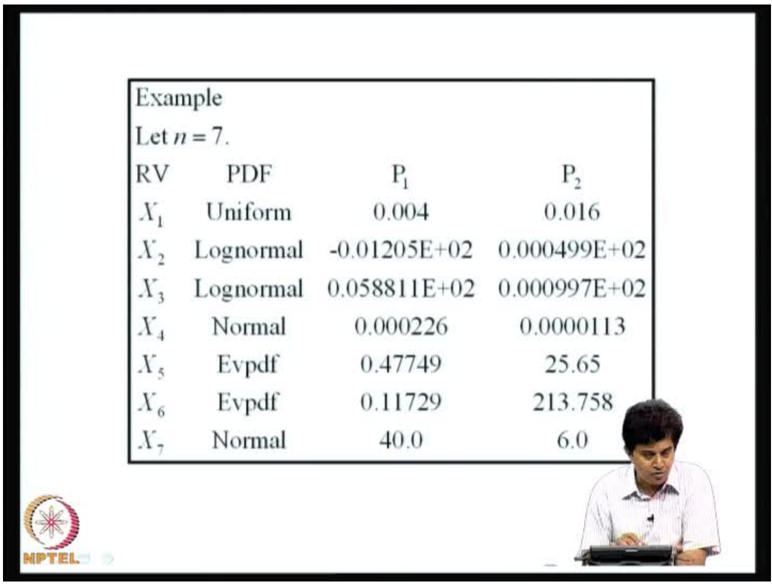
So, this is 40 point something, 50 point something, 997 point something. Similarly, standard deviations 4.95, 2.49, 198, so on and so forth. This is simulated correlation coefficient matrix. So, again, things are, things seems to be alright and these are the 3 target probability density functions X 1, X 2, X 3 to depict how non-normal they are and also shown on this graph, the corresponding plot of a normal random variable, so that difference between the two gives an idea on the non-gaussianity of the corresponding random variable.

(Refer Slide Time: 50:54)



This is simulation results on X 1, X 2 and X 3; again, blue is simulation, red is the target and we can see, that this is quite a good mutual agreement.

(Refer Slide Time: 51:07)

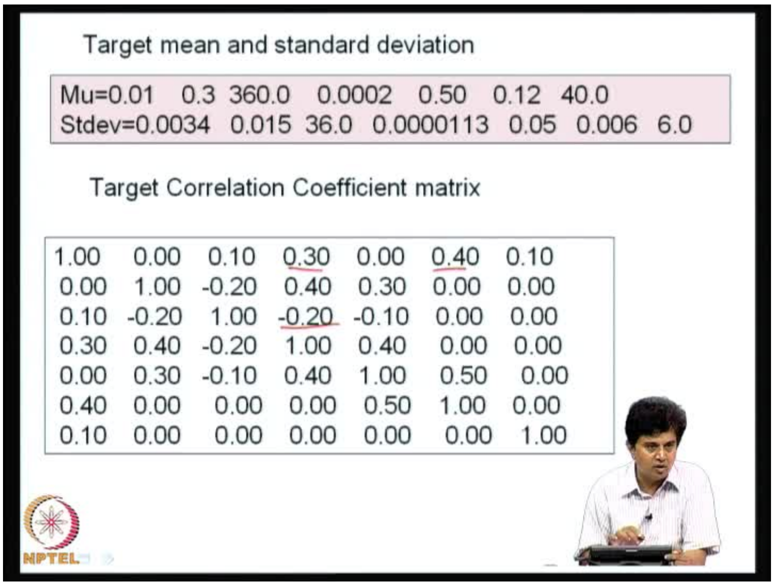


Example

Let $n = 7$.

RV	PDF	P_1	P_2
X_1	Uniform	0.004	0.016
X_2	Lognormal	-0.01205E+02	0.000499E+02
X_3	Lognormal	0.058811E+02	0.000997E+02
X_4	Normal	0.000226	0.0000113
X_5	Evpdf	0.47749	25.65
X_6	Evpdf	0.11729	213.758
X_7	Normal	40.0	6.0

(Refer Slide Time: 51:33)



Target mean and standard deviation

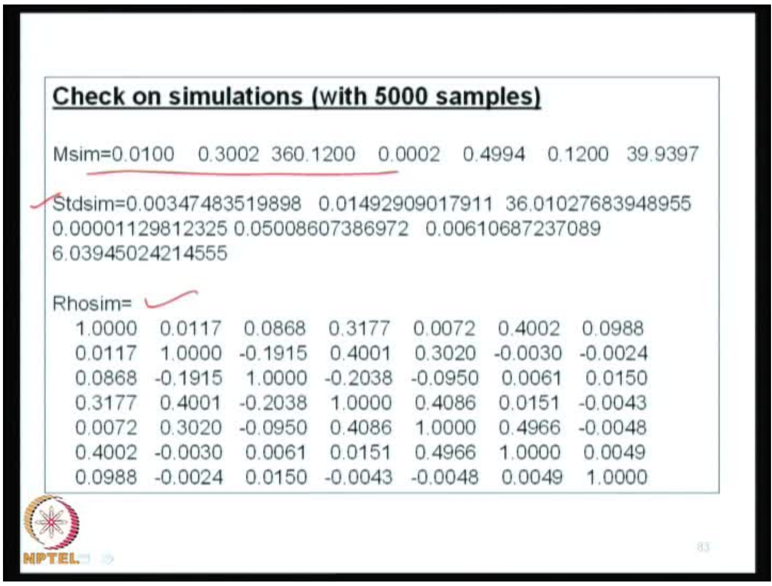
Mu=0.01 0.3 360.0 0.0002 0.50 0.12 40.0
Stdev=0.0034 0.015 36.0 0.0000113 0.05 0.006 6.0

Target Correlation Coefficient matrix

1.00	0.00	0.10	0.30	0.00	0.40	0.10
0.00	1.00	-0.20	0.40	0.30	0.00	0.00
0.10	-0.20	1.00	-0.20	-0.10	0.00	0.00
0.30	0.40	-0.20	1.00	0.40	0.00	0.00
0.00	0.30	-0.10	0.40	1.00	0.50	0.00
0.40	0.00	0.00	0.00	0.50	1.00	0.00
0.10	0.00	0.00	0.00	0.00	0.00	1.00

Now, a slightly more complicated example, there are 7 random variables: X 1 is uniform, X 2 is lognormal, X 3 is lognormal, x 4 is normal, X 5 is extreme value probability density function of type one, X 6 is again extreme value type one, X 7 is normal and P 1 and P 2 are the parameters of the distributions. There are 7 random variables and the, this is the mean and standard deviation of 2 of the 7 random variables, and this is the target correlation coefficient matrix.

(Refer Slide Time: 52:00)



Check on simulations (with 5000 samples)

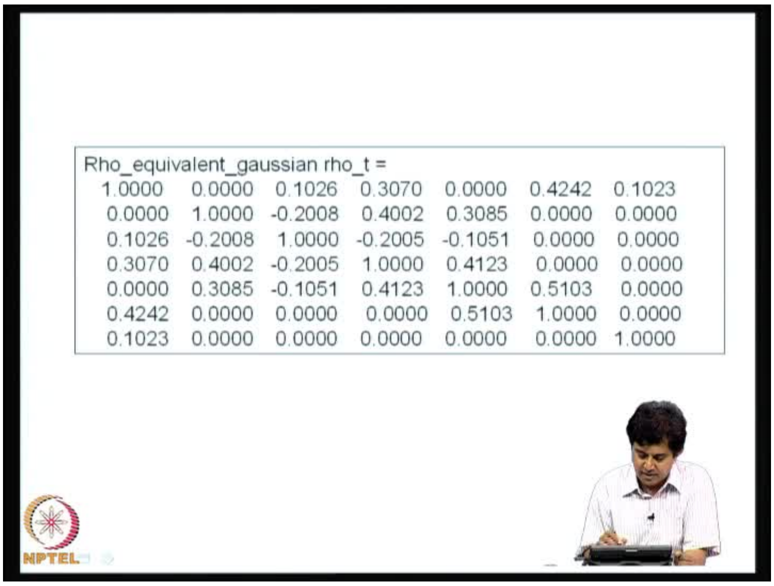
```
Msim=0.0100   0.3002  360.1200   0.0002   0.4994   0.1200   39.9397

Stdsim=0.00347483519898  0.01492909017911  36.01027683948955
0.00001129812325 0.05008607386972  0.00610687237089
6.03945024214555

Rhosim=
   1.0000   0.0117   0.0868   0.3177   0.0072   0.4002   0.0988
   0.0117   1.0000  -0.1915   0.4001   0.3020  -0.0030  -0.0024
   0.0868  -0.1915   1.0000  -0.2038  -0.0950   0.0061   0.0150
   0.3177   0.4001  -0.2038   1.0000   0.4086   0.0151  -0.0043
   0.0072   0.3020  -0.0950   0.4086   1.0000   0.4966  -0.0048
   0.4002  -0.0030   0.0061   0.0151   0.4966   1.0000   0.0049
   0.0988  -0.0024   0.0150  -0.0043  -0.0048   0.0049   1.0000
```

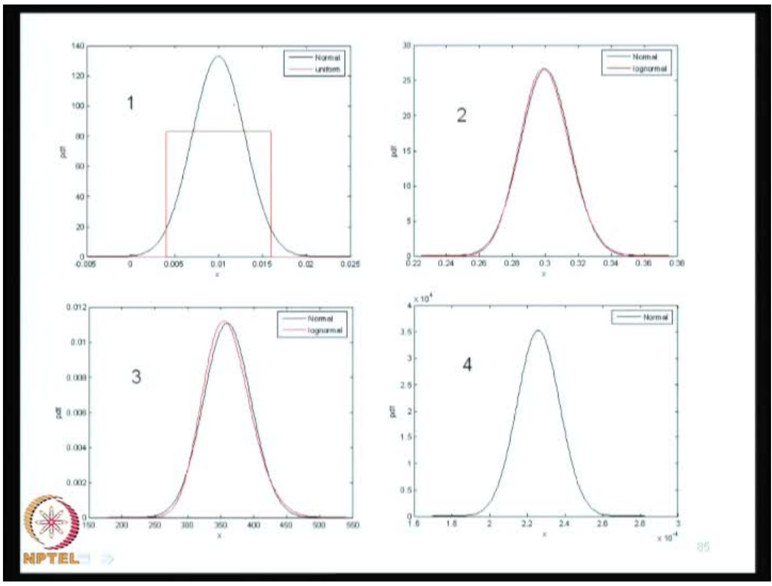
So, it is one diagonal, there are (( )) diagonal terms here, you can see some of them are negative and some of them are positive and along the diagonal we have unity. Now, we need to simulate say, 10000 numbers or 5000 numbers of these vectors of realizations of X with 5000 samples and using the Nataf transformation method, the problem is solved. And we have the simulated value of the mean, standard deviation and the correlation coefficient matrix and these have to be compared with the corresponding target values and again, they are quite good in a good agreement between the target and simulated numbers, can be remains in these numbers. That agreement, again let me emphasize, can be actually, be objectively assessed using hypothesis testing methods, with using appropriate sample distributions and for specified significance levels.

(Refer Slide Time: 52:42)



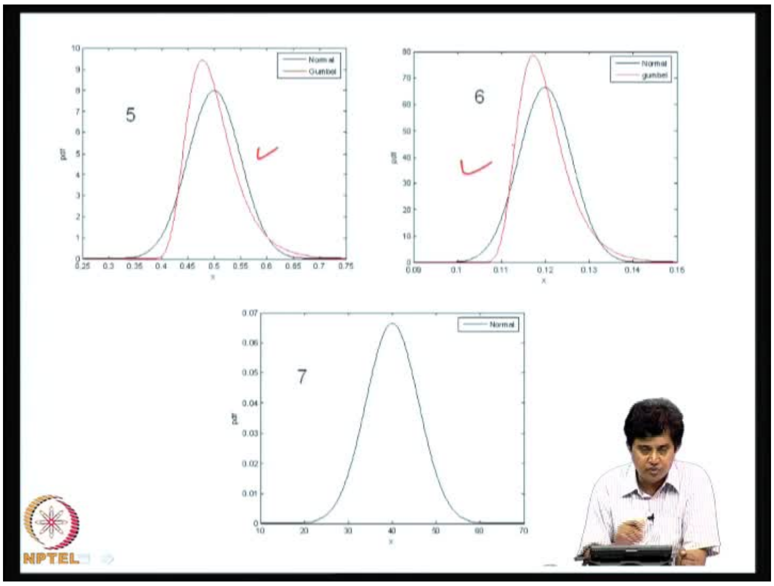
```
Rho_equivalent_gaussian rho_t =
  1.0000   0.0000   0.1026   0.3070   0.0000   0.4242   0.1023
   0.0000   1.0000  -0.2008   0.4002   0.3085   0.0000   0.0000
   0.1026  -0.2008   1.0000  -0.2005  -0.1051   0.0000   0.0000
   0.3070   0.4002  -0.2005   1.0000   0.4123    0.0000   0.0000
   0.0000   0.3085  -0.1051   0.4123   1.0000   0.5103    0.0000
   0.4242   0.0000   0.0000    0.0000   0.5103   1.0000   0.0000
   0.1023   0.0000   0.0000   0.0000   0.0000    0.0000  1.0000
```

(Refer Slide Time: 53:00)

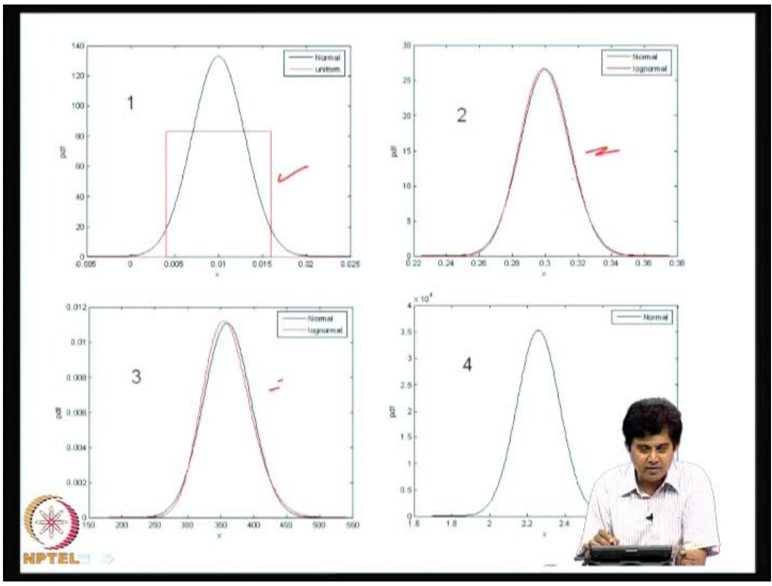


Now, this is an equivalent rho for the Gaussian random variables obtained by solving that n into n minus 1 by 2 number of integral equations that is displayed here, on this of course, there is no verification, this is an intermediate step. So, these are the random variables that we are simulating, the red plots are the target probability density function, black is the standard normal just to, is not the standard normal, the corresponding normal distribution with same mean and same standard deviation, just to give an idea on how much this specified first order distributions depart from the normal distribution.

(Refer Slide Time: 53:22).



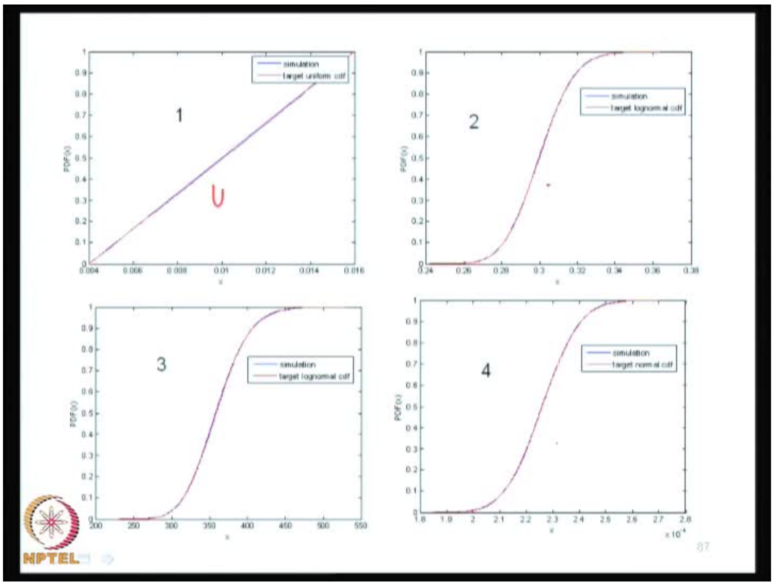
(Refer Slide Time: 53:30)



.

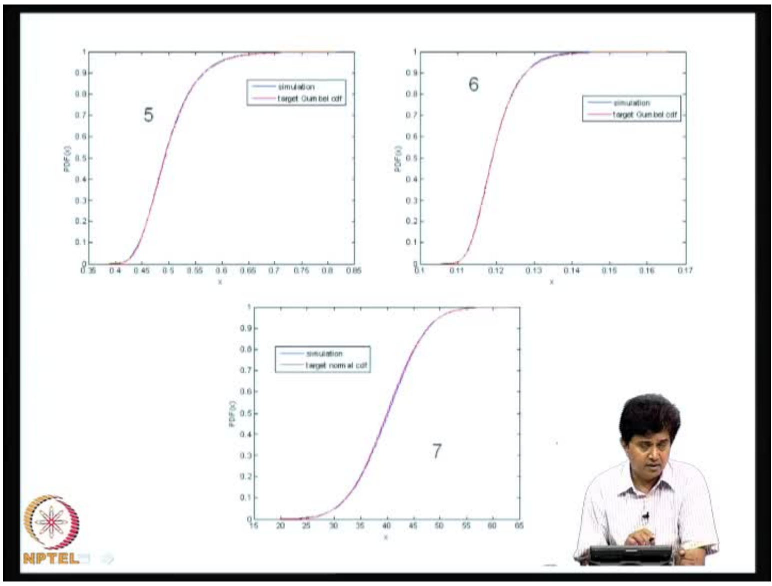
So, there is considerable departure here, here and of course, uniform distribution. There is a, this X 1, there is good departure and whereas, here although the random variable lognormal, the agreement is between normal and non-normal; the differences are not that much.

(Refer Slide Time: 53:41).



.

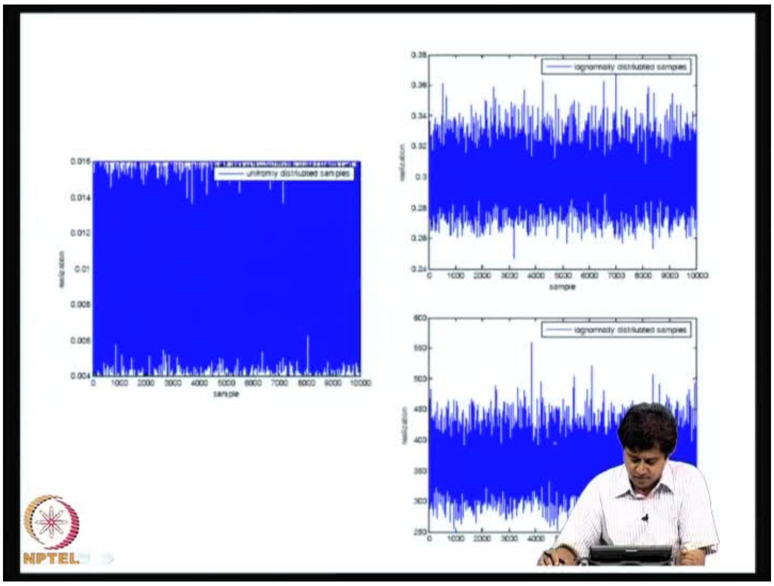
(Refer Slide Time: 53:54)



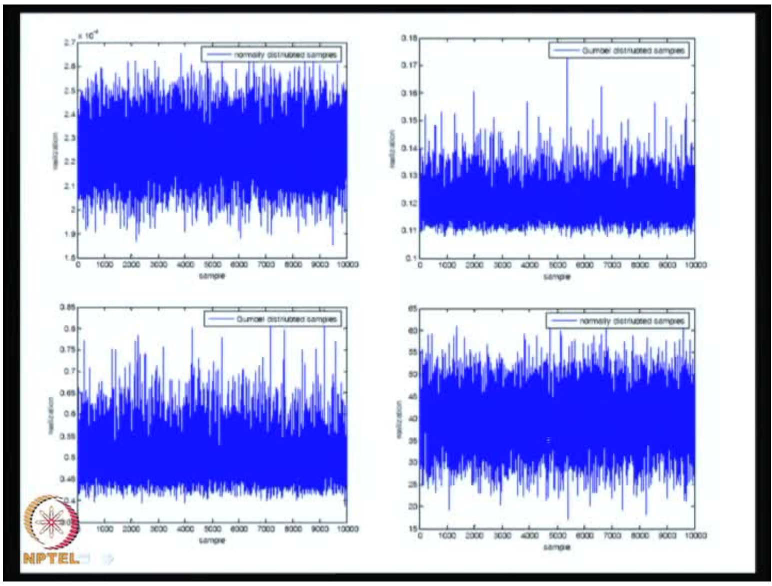
.

So, this simulation results with 5000 samples, blue is simulation, red is a target, this is uniform distribution, this is lognormal, lognormal I think, this is normal and so on, extreme value is normal, so on and so forth. So, again, the agreement is quite satisfactory, so this method seems to work quite well.

(Refer Slide Time: 54:08)



(Refer Slide Time: 54:14)



.

These are realizations; these are characterless, just for, to get a visual field for what we are doing, these are the numbers. And we have reached the end of this lecture. So, to summarize, so for what we have been able to do is to develop methods to simulate samples of scalar Gaussian, non-Gaussian random variables and vector Gaussian and vector non-Gaussian random variables. For vector non-Gaussian random variables, we have 2 methods, that is, the Rosenblatt transformation method and the Nataf transformation method. Rosenblatt method can be used if you have complete specification of the vector non-Gaussian random variables in terms of their joint density functions. The Nataf can be used in case when information is partial.

So, in the next lecture what we will do is, we will extend this discussion to cover cases of simulation of random processes. How to simulate samples of random processes where a probabilistic description, the underline random process is provided? This process can be Gaussian or non-Gaussian, scalar or vector, completely specified or partially specified and let us see how to go about doing that, in the next lecture. So, at this point, we will conclude this lecture.