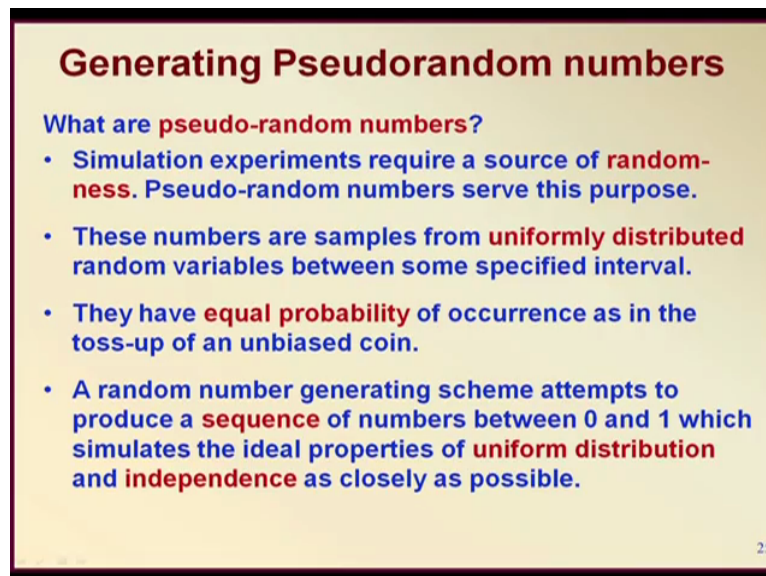


Course on Decision Modeling
Professor Biswajit Mahanty
Department of Industrial and Systems Engineering
Indian Institute of Technology Kharagpur
Module 05
Lecture No. 23
Pseudorandom Numbers

In this particular lecturer let us start with pseudorandom numbers. We have already seen that in order to carry out discrete event simulation random numbers play a very important role right, and all these experiments that we are generating the events with the help of a random numbers and therefore it is very important that generation of random numbers also follows you know very good process. So that is you know we are going to discuss in this particular lecture, that is about pseudorandom numbers.

(Refer Slide Time: 01:03)



Generating Pseudorandom numbers

What are pseudo-random numbers?

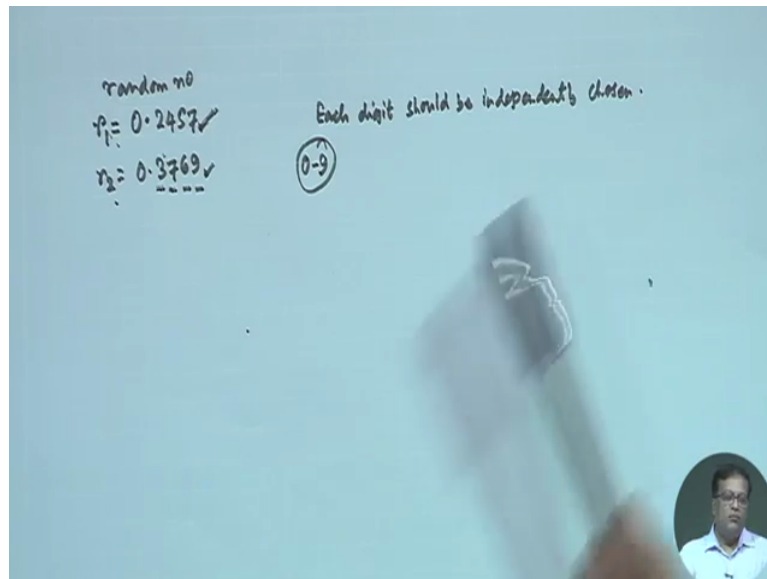
- Simulation experiments require a source of **randomness**. Pseudo-random numbers serve this purpose.
- These numbers are samples from **uniformly distributed** random variables between some specified interval.
- They have **equal probability** of occurrence as in the toss-up of an unbiased coin.
- A random number generating scheme attempts to produce a **sequence** of numbers between 0 and 1 which simulates the ideal properties of **uniform distribution** and **independence** as closely as possible.

25

So first of all what are pseudorandom numbers? simulation experiments require a source of randomness right and pseudorandom numbers serves this purpose. First of all these numbers are samples from uniformly distributed random variables between some specified interval, right. They have equal probability of occurrence as in the toss-up of an unbiased coin.

So if we get a random number say between 0 and 1 something like 0.25, so these 0.25 could have been you know there are 2 digits here 2 and 5 it could be happening any digit between 0 to 9 all with equal probability. That is about 2 digit accuracy, we can have 3 digit, 4 digit as my digit accuracy as the case may be, is it alright.

(Refer Slide Time: 02:29)




So a random number generating schemes attempts to produce a sequence of numbers between 0 and 1 which simulates the ideal properties of uniform distribution and independence as close as possible, right. So when you generate a particular random number let us write down random number arbitrarily, supposing we have random number r_1 0.2457 four digit accuracy. So this 0.2457 now you see there are 4 digits here 2457, this 2457 first of all you know each digit should be independently chosen.

Suppose another random number r_2 we get and these r_2 is let us say arbitrarily 0.3769, so you see this r_1 and r_2 you know this r_1 and r_2 there should be absolutely no relation between them, right. That means if the first number is 0.2457, 2nd number need not be 0.3769 it could be any number and in this 4 places there is equally likely that any number, and any digit between 0 to 9 could have come, that is about that you know uniform distribution and independence that must be maintained about random numbers.

(Refer Slide Time: 04:12)

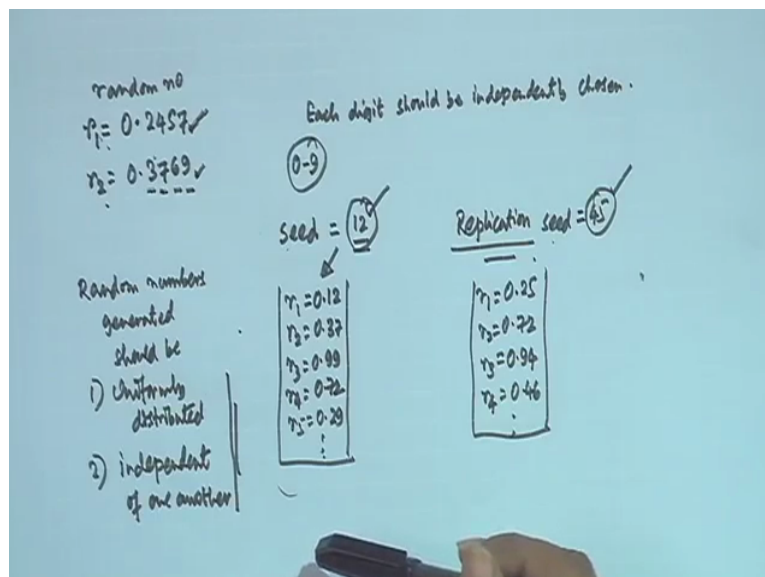
Generating Pseudorandom numbers

- What is “pseudo” about pseudorandom numbers?
- Pseudorandom numbers are generated from a fast and **deterministic** method of generating a sequence of numbers that have the property of being **random**.
- **Pseudo** is used because the numbers are obtained from the use of a **random number generator**.
- Although pseudo means false, false random numbers are **not** generated by such a number generator.
- Here “pseudo” implies that generation of random numbers by a **known method** removes the potential for true randomness. If the method is known, the set of random numbers can then be **replicated**.



But then we are using the word pseudo, why we are calling it pseudo? Basically you see the pseudo does not mean it is false, it does not mean that what we are getting out of the computers through the generator is something that is wrong or false. The fact is it actually comes from a deterministic process that is called a congruential generator, right.

(Refer Slide Time: 04:59)



random no
 $r_1 = 0.2457$
 $r_2 = 0.3769$

Each digit should be independently chosen.
0-9

Seed = 12

Replication seed = 45

Random numbers generated should be

- 1) Uniformly distributed
- 2) independent of one another

$r_1 = 0.12$	$r_1 = 0.25$
$r_2 = 0.37$	$r_2 = 0.72$
$r_3 = 0.99$	$r_3 = 0.94$
$r_4 = 0.72$	$r_4 = 0.46$
$r_5 = 0.29$	
⋮	

So because it comes from a known process method, so you see there is a issue there, what is at issue? Issue is that you know it comes from a particular thing and there is something called a seed, a random number seed. So supposing we give a particular seed let us say arbitrarily let us say seed is 12, so as long as you provide the seed supposing we get a set of 10 random

numbers maybe I am just writing small random number so say $r_1 = 0.12$, $r_2 = 0.37$, $r_3 = 0.99$, $r_4 = 0.72$, $r_5 = 0.29$ etc.

So you see as long as this particular seed 12 is given same random numbers will be generated by the generator, because it comes from a deterministic process, it is got a both good thing and a bad thing. See what is good thing, good thing is that you know if suppose we have done an experiment and we want to reproduce that experiment whether that in order to validate whether what we did is correct or wrong the advantage part is as long as we keep this seed we get the same random numbers once again and the bad thing is that they are all predictable in a sense, because you know once a person know seed all the numbers will be exactly same, right.

So every time you do the experiment you can exactly same results, what do we do to really create replication, because I said at replication means the random number experiment should be generated again. So in order to get a replication we must change the seed, so this time we cannot use the same seed so not 12, we use suppose 45. Moment we use another seed then we get a completely different set of random numbers maybe $r_1 = 0.25$, $r_2 = 0.72$, $r_3 = 0.94$, $r_4 = 0.46$ etc. etc.

so you see we get another set of random numbers when we change the seed, is it alright. Why this is happening? Why when we change the seed we get exactly the same random number sequence you know when we have a particular seed? It happens because they come from a deterministic process. What is that process? We shall see that now.

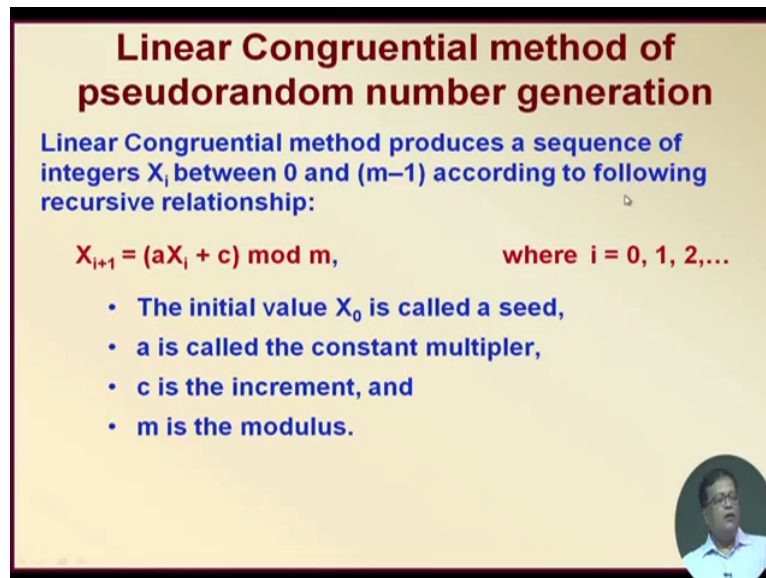
So that is why this is called a pseudorandom number, right. But as far as the 2 properties that the random number generated should be number-one uniformly distributed and number 2 independent, they should be independent of one another, right. So they are independent they are also uniformly distributed, so no issue there, so this properties are fully satisfied and therefore they can be used in experiment no issue about that.

But is be time the seed is same, same random numbers will come out, right that point must be remembered, is it alright. So sometimes you know really speaking in order to generate truly random number which nobody can guess the seed is also randomised and that randomized process is known to anyone, is it alright.

Suppose if you can relate it with your fingerprint, so if you can relate it with your fingerprint and since everybody's fingerprint is different so as long as you will come, you will generate

the same set of random numbers, right. These are very big bearing into things like encryption and maybe real-world experiment, anyhow we will come to that later on.

(Refer Slide Time: 10:00)



Linear Congruential method of pseudorandom number generation

Linear Congruential method produces a sequence of integers X_i between 0 and $(m-1)$ according to following recursive relationship:

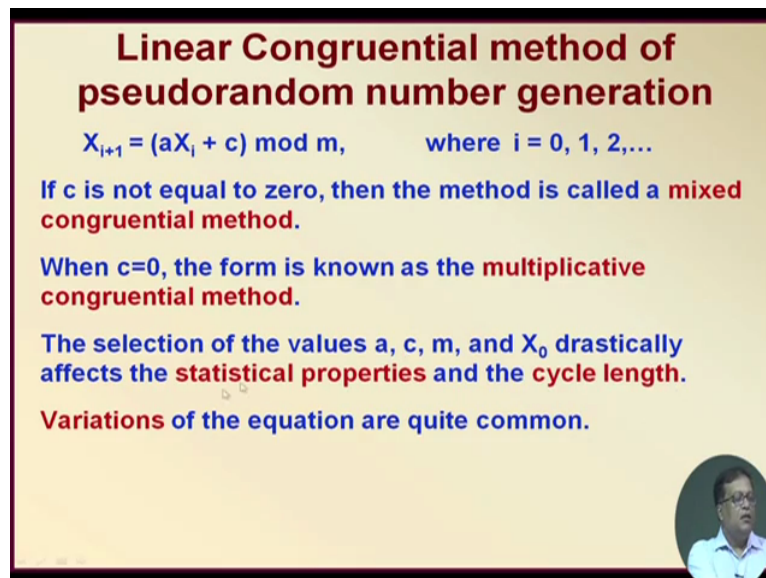
$$X_{i+1} = (aX_i + c) \bmod m, \quad \text{where } i = 0, 1, 2, \dots$$

- The initial value X_0 is called a seed,
- a is called the constant multiplier,
- c is the increment, and
- m is the modulus.

Here let us see how we actually generate a random number, one method can be linear congruential method of pseudorandom number generation. So you see there is a formula given here, X_{i+1} equal to $aX_i + c \bmod m$, where i equal 0, 1, 2. So this kind of generation of random number really follows this mathematical equation, what is that? You know X_0 is a seed, a is called the constant multiplier, c is called the increment and m is called the modulus.

So what exactly we do? We multiply the initial value that is seed X_0 into a , I mean X_0 that is the first number multiplied by a constant multiplier add constant c or increment and then take modulo m , modulo m means divided by n and take the remainder and therefore you can understand that value will always be less than m , is it not. What should be the value of a , c , X_0 and M ? We shall discuss later, but first let us see what exactly this really does.

(Refer Slide Time: 11:28)



Linear Congruential method of pseudorandom number generation


$$X_{i+1} = (aX_i + c) \bmod m, \quad \text{where } i = 0, 1, 2, \dots$$

If c is not equal to zero, then the method is called a **mixed congruential method**.

When $c=0$, the form is known as the **multiplicative congruential method**.

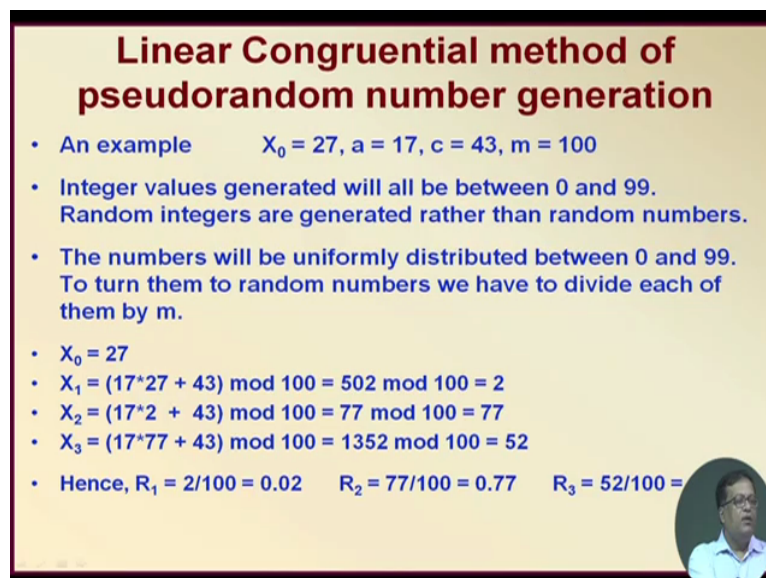
The selection of the values a , c , m , and X_0 drastically affects the **statistical properties** and the **cycle length**.

Variations of the equation are quite common.




So if c is not equal to 0, then the method is called a mixed congruential method, right. So what is a mixed congruential method, where c has a distinct value and a and X_i and m they also have distinct values that is called a mixed congruential method. When c equal to 0 then this form is called a multiplicative congruential method right, because the formula simply becomes $aX_i \bmod m$, is it alright $X_i \bmod m$. The selection of the values a , c , m and X_0 drastically affects the statistical properties and the cycle length and variations of the equations are quite common. So lot of variations are actually available.

(Refer Slide Time: 12:21)



Linear Congruential method of pseudorandom number generation

- An example $X_0 = 27, a = 17, c = 43, m = 100$
- Integer values generated will all be between 0 and 99. Random integers are generated rather than random numbers.
- The numbers will be uniformly distributed between 0 and 99. To turn them to random numbers we have to divide each of them by m .
- $X_0 = 27$
- $X_1 = (17 \cdot 27 + 43) \bmod 100 = 502 \bmod 100 = 2$
- $X_2 = (17 \cdot 2 + 43) \bmod 100 = 77 \bmod 100 = 77$
- $X_3 = (17 \cdot 77 + 43) \bmod 100 = 1352 \bmod 100 = 52$
- Hence, $R_1 = 2/100 = 0.02$ $R_2 = 77/100 = 0.77$ $R_3 = 52/100 =$



Let us look at example of linear congruential method of pseudorandom generation, X_0 equal to 27, a equal to 17, c equal to 43 and m equal to 100. So this will generate this is just an

example integer values will be generated will be all between 0 and 99. So what is the first number 27 because X_0 is 27, 2nd number 17 multiplied by 27 plus 43, mod 100 so it will become $502 \bmod 100$, so divided remainder is 2 so remainder that will be the number.


3rd number 17 into 2 plus 43 you know mod 100, why 2 because last number was 2 and then $77 \bmod 100$ we got 77. Next 17 into 77 plus 43 mod 100 it will come to 52. So 2, 77, 52, 3 numbers are generated divided by 100 then we get 0.02, 0.77, 0.52, so these will be our random numbers. So you see this is a deterministic process, because it is a deterministic process we call it pseudorandom number but all the random numbers that we generated they are actually actual properties of the uniformly distributed and independence amongst each other, so those properties are all satisfied.

(Refer Slide Time: 14:00)

Multiplicative Congruential method of pseudorandom number generation

- Multiplicative congruential generator consists of computing the following recurrent relation:


$$X_{i+1} = (a * X_i) \bmod m, \quad \text{where } i = 0, 1, 2, \dots$$
- Where
- X_i is the i^{th} pseudo-random number,
- a is a constant multiplier, and,
- $\bmod m$ denotes computation of the remainder (less than m) after repeated division of $a * X_i$ by m .
- This remainder is then set equal to the next pseudo-random number X_{i+1} .
- The process starts with an initial seed value of X_0



Multiplicative Congruential method of pseudorandom number generation

Choice of m , X_0 , and a

- The resulting number coming out of the modulus arithmetic, can only be one of $0, 1, 2, 3, \dots, m-1$. Thus eventually the series would repeat itself. To prevent this, m should be one more than the largest integer that can be held in one word of computer being used. For word length of 32 bits, m is $(2^{31} - 1) + 1$.
- The seed X_0 must be relatively prime to m . As m is a power of 2, any odd positive integer for X_0 would do.
- The constant multiplier a should also be relatively prime to m – i.e. it should also be an odd integer. A good choice of a is $8 * k \pm 3$.

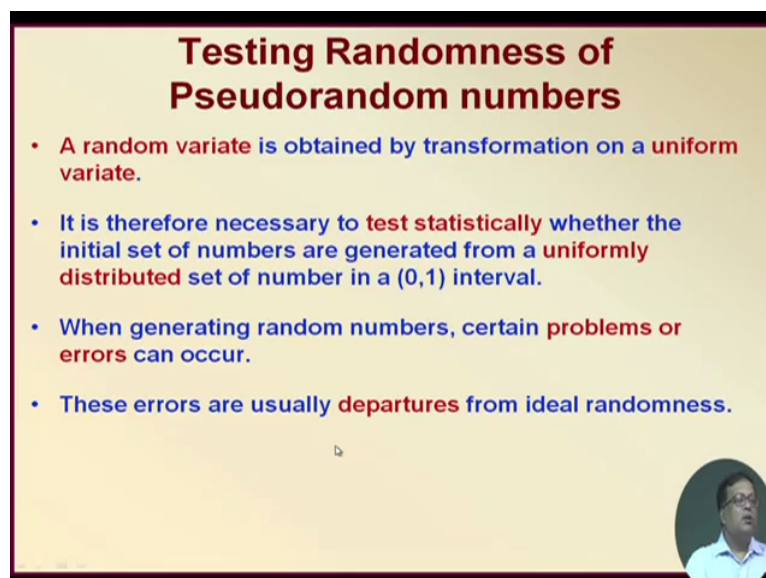


Now multiplicative congruential method as already discussed that is $X_i + 1$ equal to $a X_i$ mod m , right which is also rather popular a is a constant multiplier and this remainder is then set equal to the next pseudorandom numbers process starts with initial seed value X_0 . So you see how this choice of m , X_0 and a really comes in? M could be a very large number because that is very important otherwise you see supposed m is small like in our previous examples that was as I said just an example m should be very large, how much large? As large as possible.

So if it is a 32-bit computer, word length is allowed in a computer then that number could be $2^{31} + 1$. That means large number of you know very large number of random numbers can be generated, otherwise what will happen? The sequence will be rather small. So m should be very large this is first requirement. 2nd the seed X_0 must be relatively prime to m , right as m is the power of 2 any odd positive integer for X_0 would do.

So must remember it should be relatively prime to m and m is of power of 2 so any odd positive integer for X_0 would do, right. And the constant multiplier a should also be relatively prime to m that is it should also be an odd integer a good choice of a is $8k + 3$. So k you can put 1, 2, 3, etc. suppose you put k equal to 1 then a will be 11 or maybe 5 something like that, so these are some choice of m , X_0 and a , right.

(Refer Slide Time: 16:11)



Testing Randomness of Pseudorandom numbers

- A random variate is obtained by transformation on a uniform variate.
- It is therefore necessary to test statistically whether the initial set of numbers are generated from a uniformly distributed set of number in a (0,1) interval.
- When generating random numbers, certain problems or errors can occur.
- These errors are usually departures from ideal randomness.

© 2011 Pearson Education, Inc. All rights reserved.

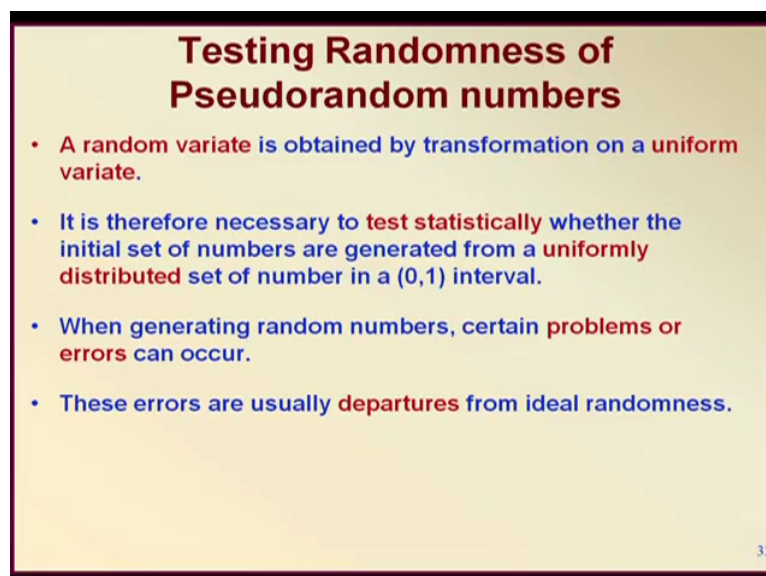
So in order to test randomness of pseudorandom numbers you know what is very important is that the random variate is obtained by transformation of a uniform variate and it is therefore necessary to test statistically whether the initial set of numbers are generated from a

uniformly distributed numbers 0, 1. Actually 2 things are here, the first thing that we are talking we have talked about here is called a random variate you see not all numbers are uniformly distributed. In queuing example please recall we said that arrival is Poisson or service time is exponential.

So look here if arrival is Poisson and random numbers you get is uniformly distributed it will not do. If we take a random number and then we translate this to arrival what will happen? You will be definitely getting a set of arrivals and that will be random in a sense but then they will be uniformly distributed, right. The events all arrivals will be equally likely, but we do not want it, we want the arrival should be Poisson distributed, distribution should be Poisson or service time should be exponential.

How to do that? You have to convert the uniform distribution to a set of what is known as random variates. There are methods to do it and we shall discuss at an appropriate time how to convert particular number which may be uniformly distributed random numbers to a set of random variates which maybe exponential distribution or maybe Poisson distribution on any other distribution during normal, erlang, etcetera etcetera, right.

(Refer Slide Time: 18:29)



Testing Randomness of Pseudorandom numbers

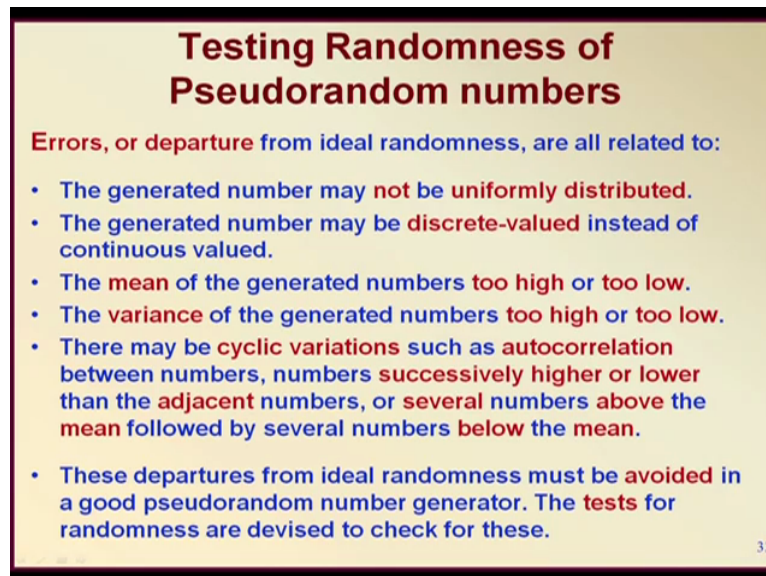
- A random variate is obtained by transformation on a uniform variate.
- It is therefore necessary to test statistically whether the initial set of numbers are generated from a uniformly distributed set of number in a (0,1) interval.
- When generating random numbers, certain problems or errors can occur.
- These errors are usually departures from ideal randomness.

32

We shall discuss it later point of time, but even before that we should test statistically it should be possible that the set of initial numbers which are generated whether they are uniformly distributed or not, right. In order to have confidence in our experiments that is also required sometimes that the random numbers if we are generating weather those random numbers are truly uniformly distributed and they are independent of each other. So that

statistical test has to be done and to really have confidence in the set of random numbers that we are making use of, right, if while generating random numbers certain problems or errors can occur and these errors are usually departures from ideal randomness, alright.

(Refer Slide Time: 19:17)



Testing Randomness of Pseudorandom numbers

Errors, or departure from ideal randomness, are all related to:

- The generated number may **not be uniformly distributed**.
- The generated number may be **discrete-valued** instead of continuous valued.
- The **mean** of the generated numbers **too high or too low**.
- The **variance** of the generated numbers **too high or too low**.
- There may be **cyclic variations** such as **autocorrelation** between numbers, numbers **successively higher or lower** than the **adjacent** numbers, or **several numbers above the mean** followed by **several numbers below the mean**.
- These departures from ideal randomness must be **avoided** in a good pseudorandom number generator. The **tests** for randomness are devised to check for these.

33

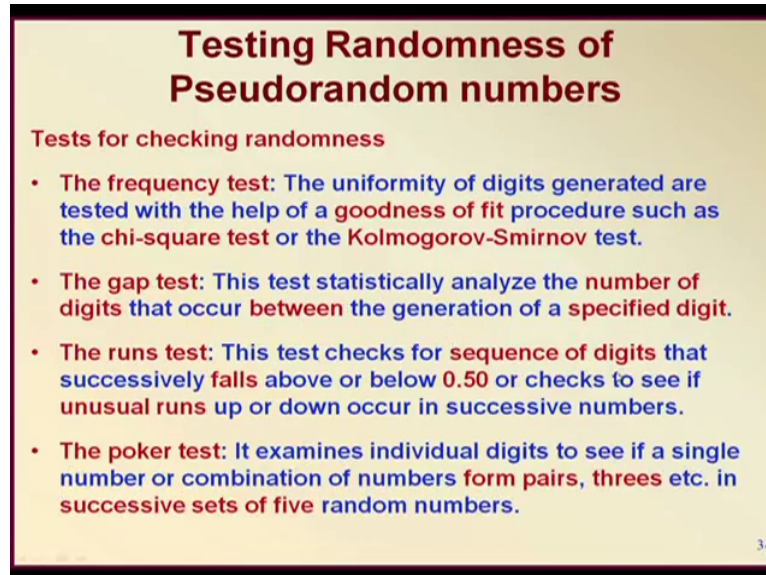
So errors or departure from ideal randomness are related that generated number may not be uniformly distributed, right, the generated number may be discrete valued instead of continuous valued. See sometimes there may be random but certain numbers are not occurring only very set of discrete values are coming up like 0.2, 0.3 but 0.27, 0.29 they are not coming up this could be a problem.

The mean of the generated numbers are too high or too low, right, suppose we have generated 100 numbers, right, out of these 100 numbers how many times 2 should come, right. Suppose out of 100 numbers 1 has 20 times, 2 has come 10 times, 3 has come only 4 times, what will be there mean, right. So you have to see the mean of the generated numbers should not be too high or too low, the variance of the generated numbers again should not too high or too low, is it not.

There may be cyclic variations such as auto correlations between numbers, so you see sometimes those numbers may be auto correlated even that is not good, right. Number successively higher or lower than the adjacent numbers or several numbers above the mean followed by several numbers below the mean all of these essentially tells that the random numbers that we have generated they are not truly uniformly distributed, if such departures

from ideal randomness occurs. A good pseudorandom number generator should avoid such departures from ideal randomness and that is why we carry out some test for randomness.

(Refer Slide Time: 21:18)



Testing Randomness of Pseudorandom numbers

Tests for checking randomness

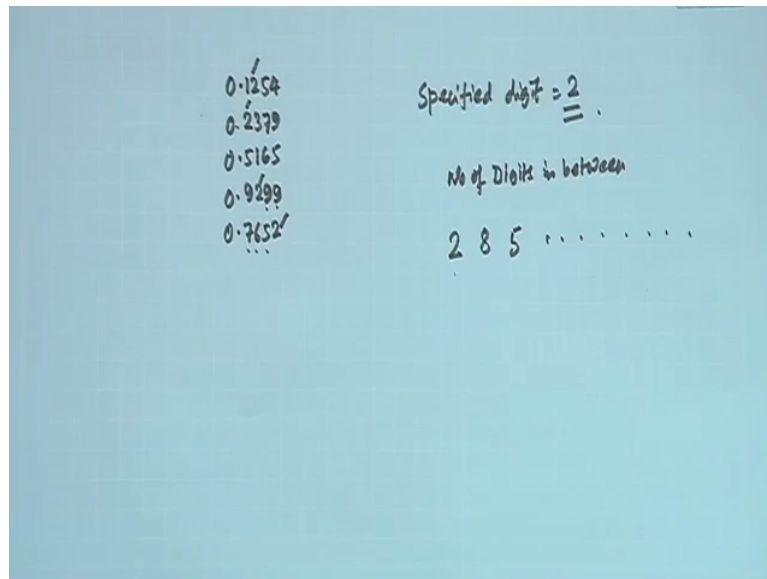
- **The frequency test:** The uniformity of digits generated are tested with the help of a goodness of fit procedure such as the chi-square test or the Kolmogorov-Smirnov test.
- **The gap test:** This test statistically analyze the number of digits that occur between the generation of a specified digit.
- **The runs test:** This test checks for sequence of digits that successively falls above or below 0.50 or checks to see if unusual runs up or down occur in successive numbers.
- **The poker test:** It examines individual digits to see if a single number or combination of numbers form pairs, threes etc. in successive sets of five random numbers.

34

So in our next slide we shall see these test. So apart from many test that could be possible some tests are discussed here 4 of them are here in this page. The first one is called the frequency test, what is frequency test? The uniformity of digits generated are tested with the help of goodness of fit procedures such as a chi-square test or the Kolomogorov-Smirnov test, right.

So what we do? We try to find out that those you know the digits that we have generated what is their frequencies. The frequency test we will discuss little more little later at this point let us only know that we are trying to test the goodness of fit by looking into the uniformity of the numbers. The 2nd one is called the gap test, this test statistically analyse the number of digits that occurs between the generation of a specified digit, right.

(Refer Slide Time: 22:38)



So let us look at what exactly means, let us say suppose we have set of random numbers, so let us call some random numbers you see 0.1254, 0.2379, 0.5165, 0.9299. Now let us say okay let us take another number 0.7652 okay. Now let us say we call a specified digit is 2, let us call a specified digit is 2 then you see how many digits are coming as a number between the specified digit if we take them all as a continuous sequence.

You see we get a 2 here, we get a 2 here, we get another 2 here and we get another 2 here, so between this 2 you know how many digits, number of digits in between, so 2 digits 5 and 4. There is an occurrence of 2, there is another occurrence of 2 between that 2, 2 digits occurred. In the next 1, 2, 3, 4, 5, 6, 7, 8, right, so 8 digits have occurred between the next occurrence of 2.

And then between the next occurrence of 2, one, 2, 3, 4, 5, so 5 digits, so like this you know you form the sequence 2, 8, 5, etcetera etcetera, so what the gap test does? Gap test really looks into whether is numbers or digits that are occurring between the specified digit 2 like 2, 8, 5, etcetera, whether these values are random themselves or there is a pattern. If there is a pattern that means that number that we have got our not truly random.

(Refer Slide Time: 25:02)

Testing Randomness of Pseudorandom numbers

Tests for checking randomness

- **The frequency test:** The uniformity of digits generated are tested with the help of a goodness of fit procedure such as the chi-square test or the Kolmogorov-Smirnov test.
- **The gap test:** This test statistically analyze the number of digits that occur between the generation of a specified digit.
- **The runs test:** This test checks for sequence of digits that successively falls above or below 0.50 or checks to see if unusual runs up or down occur in successive numbers.
- **The poker test:** It examines individual digits to see if a single number or combination of numbers form pairs, threes etc. in successive sets of five random numbers.

34

0.1254 }
0.2379 }
0.5165
0.9299
0.7652

Specified digit = 2

No of Digits in between

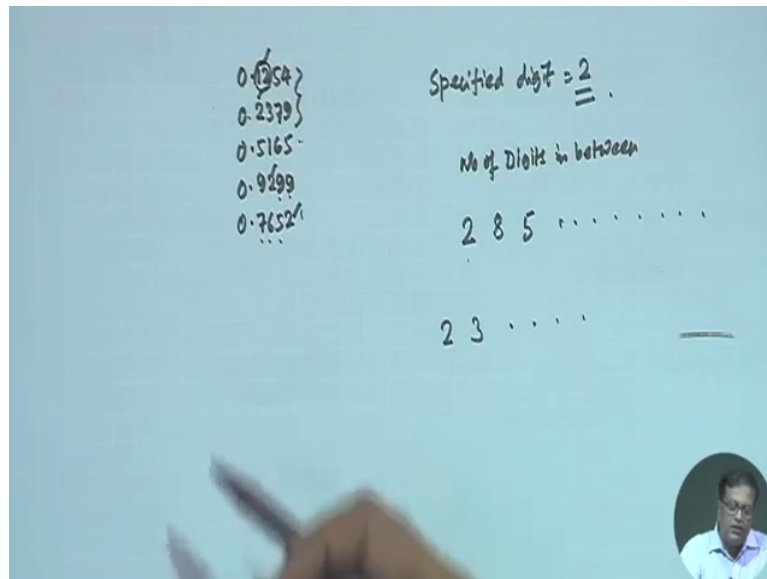
2 8 5

2 3

Then next test is known as the run test, the run test checks the sequence of digits that successively falls above or below 0.5 or checks to see if unusual runs up or down occurs in successive numbers. So how many numbers are successively following above or below 0.5 right. So maybe you see if you look here than there are 2 numbers that are falling below 0.5, then these numbers are all above 0.5, so above 0.5 again 3.

So runs are first 2 then 3 etcetera, right, so 2 numbers below 0.5 and 3 numbers above 0.5. So we can see that whether again these numbers are random themselves. The 4th test is called the poker test, the poker test it examinations individual digits to see if a single number or combination of numbers forms pairs, threes, etcetera in successive sets of you know 5 random numbers.

(Refer Slide Time: 26:20)



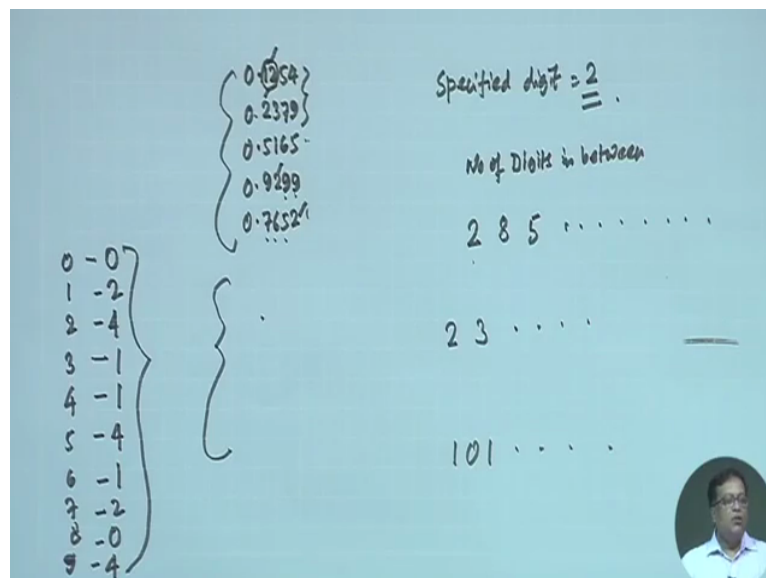

So whether a pair, so suppose we take a pair like 1-2 you see 1-2 if you take it as a pair then you see a pair has come here, right. Now if there is 1-2 again out of all these numbers know we have not found, but maybe if you take 1-2 for a large set of random numbers how many times this 1-2 is coming and you know these 1-2 is coming in successive set of 5 numbers so here is 1, 2, 3, 4, 5 in this 5 numbers there is 1 occurrence. In the next set of 5 numbers maybe there is 0 occurrence, in next set of 5 numbers again there will be 1 occurrence. So you see 1 0 1 etcetera that kind of sequence we shall get for a particular pair and have to see whether they are random in themselves or not, right, that is a poker test.

The 4th one is a serial test that this test takes the randomness of a successive numbers in a sequence, so very simple. The product test checks for independence or co-relation between sequences of random numbers. An autocorrelation test examinations the autocorrelation patterns or relationships between numbers in sequence of random numbers that is whether number is dependent or another number in the sequence, finally the maximum test, the maximum test examinations random numbers sequence to find out abnormally low or high numbers, right.

(Refer Slide Time: 27:53)

The Frequency Test


- Frequency or uniformity test counts **how often** numbers in a given range occur in the sequence to ensure that the numbers are uniformly distributed.
- There should be **no favorite number** and no number should occur more frequently than from expected chance variations.
- In 1000 two-digit numbers, we should expect about 100 numbers from 00 to 09, 100 numbers from 10 to 19 etc. – **non-randomness** in data may be suspected if it is say 30 or 200!
- The uniformity of digits generated are tested with the help of a **goodness of fit** procedure such as the **chi-square test** or the **Kolmogorov-Smirnov test**.
- The pseudo-random numbers must **pass** the frequency test among others.



Specified digit = 2

No. of Digits in between

0 - 0	0.1954	2
1 - 2	0.2379	8
2 - 4	0.5165	5
3 - 1	0.9299	3
4 - 1	0.7652	1
5 - 4		
6 - 1		
7 - 2		
8 - 0		
9 - 4		



So frequency test as we said we will discuss a little bit more it really test how often number in a given range occurs in the sequence to ensure that the numbers are uniformly distributed. So what you do? You keep a count of the different numbers that you have got. So here suppose you have this 5 random numbers we have designed so we can check how many 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, right, how many are generated.

So you can see that out of all these numbers 0 is not there 0 times, 1 has come 1, 2 times, 2 has come 1, 2, 3, 4 times, right. And then 3 has come 1 time, 4 has come again 1 time, so like this we have to make a note 5 has come 1, 2, 3, 4 times, right, 6 has come only 1 time, 7 has come 1, 2 times, 8 is not there at all and 9 is 1, 2, 3, 4 times.

So you see these are the numbers that many times these numbers are generated and we have to see whether these numbers are random or not. Obviously we should not do experiment with just 5 numbers maybe we should take 100 numbers and then see how is this frequency of this 0 to 9 all these numbers and we have to see whether you know by a chi-square test or Kolmogorov-Smirnov test whether goodness of fit really results, right, and the pseudorandom numbers must pass the frequency test among others, right.

So far we have discussed how to generate random numbers, the scheme of discrete event simulation scheme, some examples of how to generate pie or Monte Carlo simulation. In our next classes we shall see more of you know what is known as how to generate random variates and some queuing examples, right. So thank you very much.