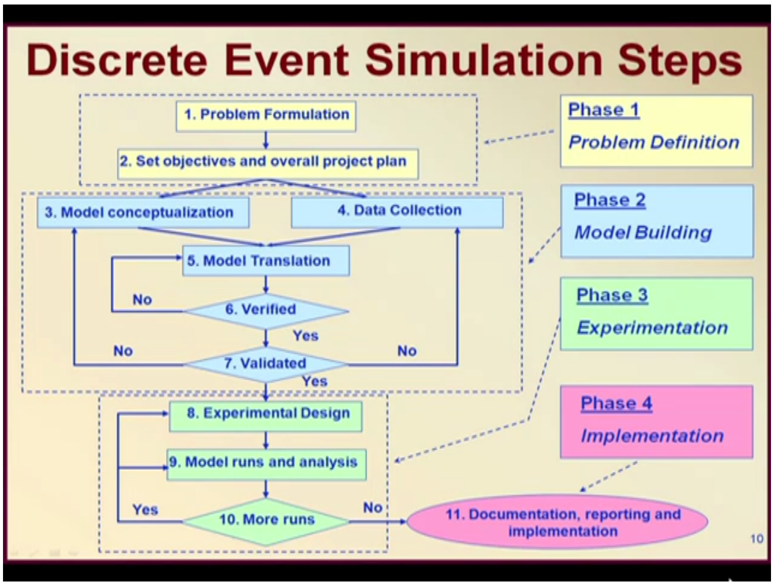
**Course on Decision Modeling**
**Professor Biswajit Mahanty**
**Department of Industrial and System Engineering**
**Indian Institute of Technology Kharagpur**
**Lecture 22**
**Module 5**
**Discrete-Event and Monte-Carlo Simulation**

In our last class we have got a general introduction on the simulation and we have seen the there are different type of simulation the Discrete-Event Simulation, the Monte-Carlo Simulation, Continuous Simulation all of these, out of these our main topic will be Discrete-Event Simulation. We spent maximum time on that particular simulation. So we have already introduced how to the broad ideas of Discrete-Event Simulation where events do occur.

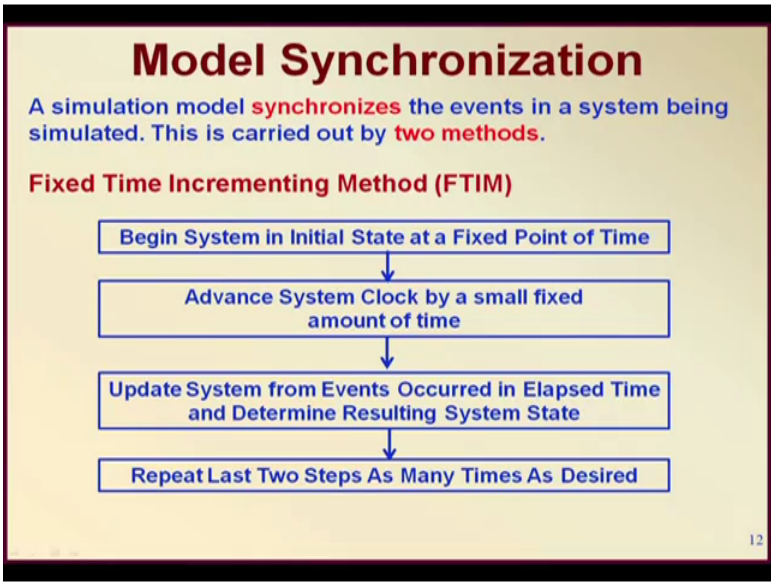
So we will continue that discussion and towards the end we shall see some examples of Monte-Carlo Simulation and thereafter on subsequent lectures we will continue the Discrete-Event Simulation discussions.

(Refer Slide Time: 1:20)



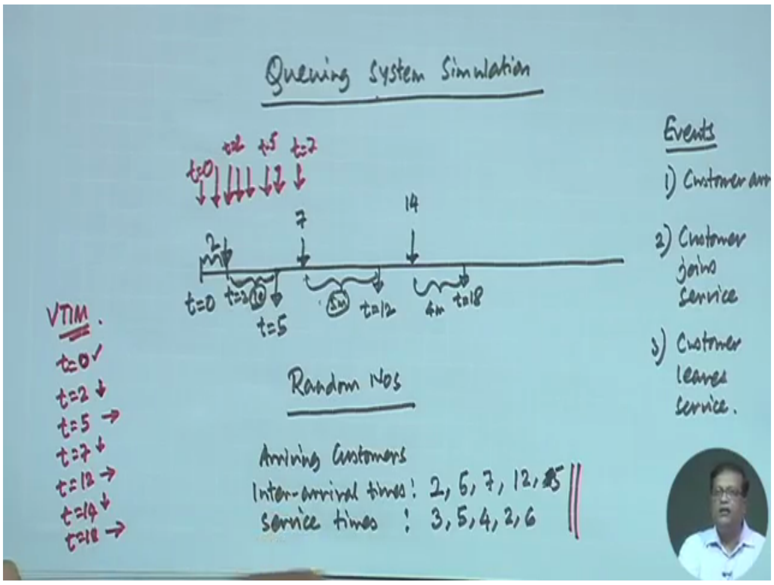

So if you really let me in fact go back to our previous discussions that in our previous class we were towards the end we were discussing the steps of Discrete-Event Simulation. What we found that in the beginning we have problem formulation, after that model building including of conceptualization, data collection, translation, verification and validation. Then comes the experimental design model runs etc and finally implementation.

(Refer Slide Time: 2:04)




Now apart from this conceptualization and things like that when it comes to actual experimentation then there are basically two broad Synchronization method which must be understood. One is called the fixed time incrementing method or so called FTIM, the other is called a variable time incrementing method or VTIM. Let me explain exactly what we mean by them, right.

(Refer Slide Time: 2:32)




So let us say that we have a particular queuing system simulation we are what we are trying to do is we are doing a queuing system simulation, right. So in a queuing system simulation so you see this is the time dimension, this is t equal to 0. Now what really happens you see what are some events there are three events number first event is customer arrive arrival as the customer arrival arrives customer can do two things the customer can actually join a queue or a customer can actually join the service directly, is it alright. These are the two things that the event that can happen.

Now the second event that happens the customer joins service that could be a second event and the third event customer leaves service, right. Now obviously there is a set of random numbers, what are random numbers? How we do random number experimentation? How we generate arrival? How we generate service, etc we shall discuss later. But let us say what we have found out the customers are arriving let us say like this arriving customers, right.

So let us say inter-arrival times suppose the inter-arrival times generated are 2, 5, 7, 12, 15, right and we also generate what is known as service times? What are could be some service times? The service times could be arbitrarily say 3, then 5, 4, 2, 6, right. So if these are our inter-arrival obviously inter-arrival times that we have chosen you know may be too many last one is really very high so let us make it only 5 and not 15. So 2, 5, 7, 12 and 5 and service time 3, 5, 4, 2 and 6.

So really what exactly it means that you see the first customer is arrives at t equal to 2, so this is 2 so you see first customer arrives. So after first customer arrives what does the customer does it directly joins the service, right and it takes 3 minutes. So what will happen the first customer to be serviced will be at t equal to 5, is it not. So you see at during this time there is one customer who is in service is it not one customer is in service during this period.

But when the second customer is arriving let us say second customer is arriving after 5 more minutes so it is 5 so this is where the second customer arrives at t equal to 7, how much time is required for service? 5 minutes, right. So these particular customer will leave at here that is t equal to 12 look so this is the 5 minutes so this is where the second customer get serviced that is about at this point of time there is one customer.

So again the next customer is arriving after 7 more minutes so if 7 it has come the next customer is coming at 14, so you can understand and this customer is again taking 4 minutes of service because during this time there is nobody in the system. So here the next one will become 14, so here you know this 18 t equal to 18 this is the 4 minutes.

So like this the system will go on and you know what is really happening we are able to generate the arrival the customer is coming and second customer is coming, third customer is coming, first customer is getting service after you know certain 5 minutes, the second 3 minutes, second customer is getting service after 5 minutes, next customer is getting service after 4 minutes so all of these things are happening actually all these things can be done by inter-arrival time generation 2, 5, 7, 12, 5 and service time after that we can do you know the detailed processing about queuing and things of that sort.

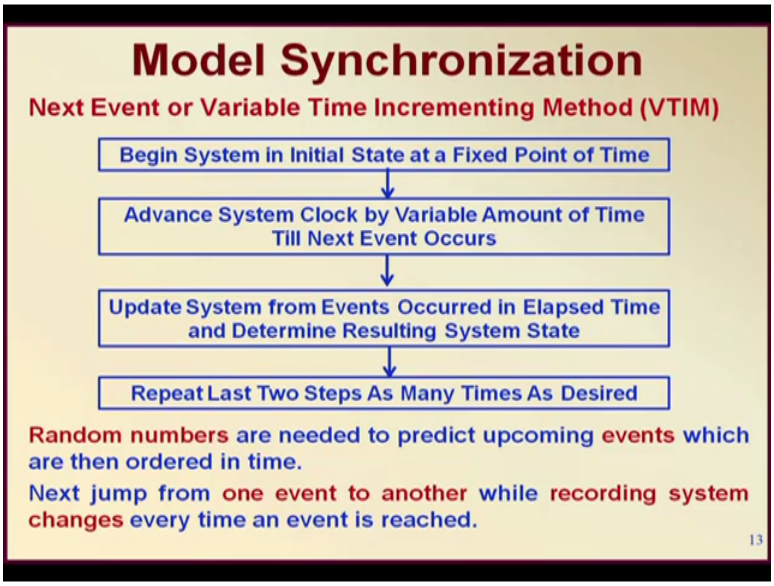
But what is interesting to note here is that you know how do we track track the system? One one method could be we track at every minute, every minute we track that is you see this is t equal to 0 we see nothing happens, then t equal to 1 nothing happens, t equal to 2 suddenly what we find one customer has arrived so note that yes one customer has arrived, t equal to 3 nothing happens, 4 nothing, t equal to 5 suddenly I find yes that customer goes out of service, right so this is t equal to 5, then again you see t equal to 6 nothing happens, t equal to 7 another customer has now arrived so look so like this if you keep tracking the system you know that is called our Fixed Time Increment Method.

So probably you can now look at the slide what is fix time increment method begin the system in initial state you know please look at the slide that is begin system in initial state at fixed point of time, advance system clock by a small fixed amount of time and update system from events occurred in elapsed time and determine results in system state and repeat the last two steps as many times as desired. So this is called the Fixed Time Increment Method, right.

So what is happening the system clock is advanced by a fixed amount of time every time and look whether anything has happen during that one minute time, is it alright. Some event can occur either a customer can arrive or customer can join queue in the example we found nobody joins queue as such because arrival times are too distant, is it alright. But service times on the other hand that is are not so high rather low so you know queues were not formed during that time.

So only two events were generated one is customer arrives and customer goes out of service, is it alright anyhow so if we see every minute and account the system that is called Fixed Time Increment Method.

(Refer Slide Time: 11:30)



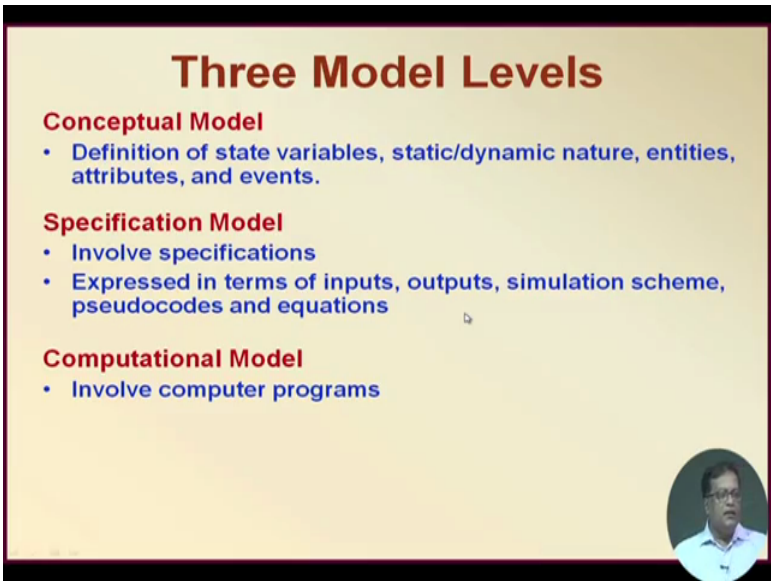

Let us look at the other method which is called the Variable Time Increment Method. What happens here what happens here is look at this you know the diagram that we have drawn you see at t equal to 0 system starts, upto t equal to 2 nothing happens so we have already generated mental this plot you know in our mind or may be somewhere and we actually know that things are happening at t equal to 0, at t equal to 2, at t equal to 5, at t equal to 7, at t equal to 12 and t equal to 14 and t equal to 18, right so things are happening only on these times, right t equal to 0 system starts, t equal to 2 first customer arrives, right t equal to 5 first customer goes out, t equal to 7 second customer arrives, t equal to 12 second customer goes out, t equal to 14 third customer arrives, t equal to 18 third customer goes out.

So this is what is happening in Variable Time Increment Method we are really changing the system time based on the events occur. How do you know when the events are occurring? Basically through random numbers we have already generated the these inter-arrival times and service times and we have created this map already and therefore we know and therefore we increment the system only at these specified times.

So now look at the slide you can see that the next event or Variable Time Increment Method begin system in initial state at a fixed point of time, advance system clock by variable amount of time till next event occurs, update system from events occurred in elapsed time and determine results in system state and repeat the last two steps as many times as desired, alright. So random numbers are needed to predict upcoming events which are then ordered in time. Next jump from one event to another while recording system changes every time an event is reached.

So you see a you are now getting a broad idea of how Discrete-Event Simulation is carried out, right. First make a conceptual model, generate random numbers, generate events and then follow either Fixed Time Increment Method or Variable Time Increment Method to really update the records, right and finally from there make meaningful what you call conclusions for implementation we will give queuing examples on Discrete-Event Simulation in subsequent classes.

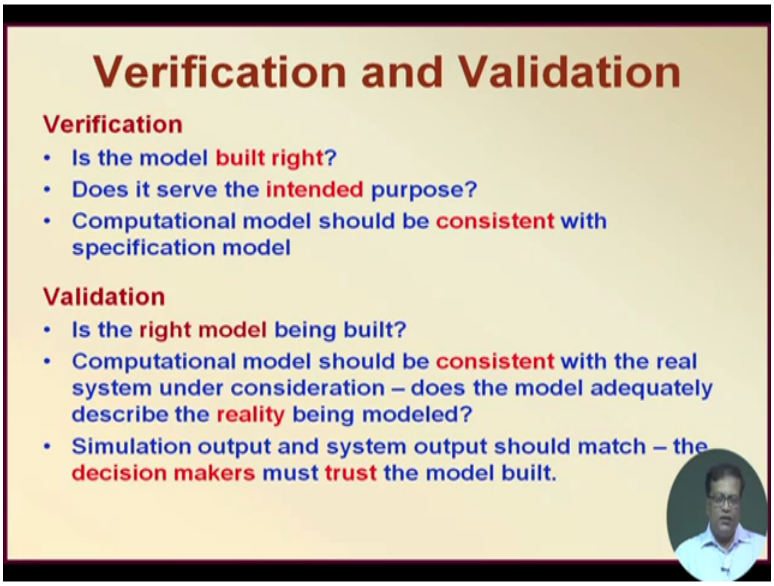
(Refer Slide Time: 14:25)




So basically there are three broad model levels one is called a conceptual model which is like you know in a mentally you generate that these kind of events are occurring, this is happening, this is flowing, these are the entities, these are their attributes, these are the state variables and put them into a conceptual map so that is a conceptual model. Then next one you actually specify and while specifying it has to be expressed in terms of inputs, outputs, simulation scheme, pseudocodes and equations so all of those.

So if you are talking about queuing the kind of queuing parameters have to be established, what is coming? You know you are now becoming more exact and generating and all the relevant requirements which are required to make a computational model, right that is what is specification model and when all of these things are put in final form a computer program that is called a computational model. So these are the three model levels.

(Refer Slide Time: 15:41)

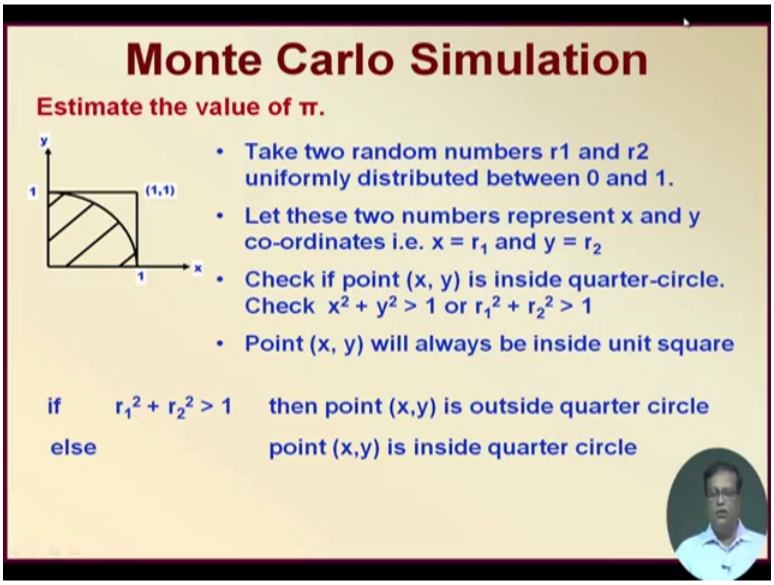



You see and before we proceed we must verify the model and also validate the model. What is verifying? Verifying means are we following all the procedures appropriately and properly that is, is the model built right? Means is the model building process is done with all the right procedures if that is so then that is called verification. In simple words the computational model should be consistent with the specification model, right means whatever specification are given is a computational model consistent with it, if it is so then we say the model is verified.

And finally the validation, is the right model being built? Computational model should be consistent with the real system, right. So some times what happens we built the model it is all the specification (())(16:38) but finally the real question it does not address because in the queuing model we are talking about our real question was that is the service adequate or we have to augment the service.

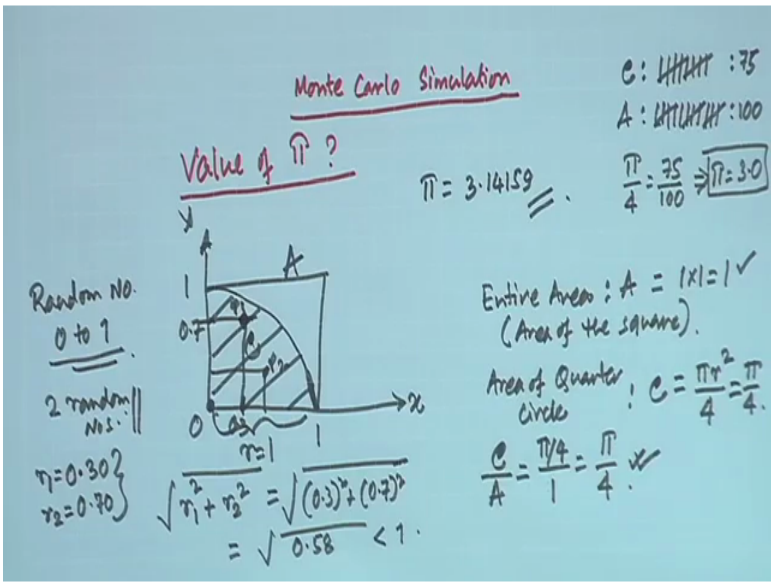
Suppose we do not get answer to that question, right we generate lot of data we simulate, we have done everything may be correctly but we are unable to really find out whether the service is adequate or inadequate. Then the purpose is not served, the objective is not fulfilled. So although the model is verified it follows the specifications but it is not validated. So validation also requires talk to the decision makers really see the decision makers who are finally making the decisions about this particular system are they happy with the model, are they really seeing the system output and the simulation output they are matching and truss could be built into the model, right so that is about verification and validation. So not only just built a model verify the model and then validate the model before you actually start experimenting with it.

(Refer Slide Time: 18:02)




Now with that we start with the we take a little departure from the Discrete-Event Simulation model and now let us look a class of model which is known as the Monte-Carlo Simulation model, right we will not discuss much about Monte-Carlo Simulation model our main discussion will be on what is known as Discrete-Event Simulation model only but since Monte-Carlo Simulation is one of the very very important class of models and lot of interesting questions can be answered let us also look at it in todays class.

(Refer Slide Time: 18:43)



So what is really happening in Monte-Carlo Simulation in Monte-Carlo Simulation we are let us take an example suppose we need to find out value of pi, how do we estimate value of pi, right. Everyone has told you that value of pi is something like 3.14159 approximately so this value you know but suppose you are told that estimate this value. You see very interesting method can be constructed where supposing this is x direction and this is y direction and we have what is known as an unit square so this is 0, this is 1 and this is 1 this is a unit square.

Now within the unit square you take a quarter circle, right you take a quarter circle and then hatch this quarter circle area, right. So let us call the entire area entire area as A, so what is A? This is A the entire area that is the square area of the square and area of quarter circle let us say C, so what is C? This hatched portion is C. So what mathematics tells what is the area of A, area of A equal to 1 into 1 equal to 1 because this side is 1, this side is 1, so 1 into 1 1. So area A is equal to 1. What is the area of C? It is pi r square by 4, but what is the r? You see what is the radius? So this is radius r equal to 1, r equal to 1 so C equal to pi by 4 C equal to pi by 4.

So what is the C by A? C by A will be pi by 4 by 1 equal to pi by 4, so C by A the area C divided by area A is nothing but pi by 4, or pi is to 4 pi is to 4. So this is the conceptual part is done. Now what we do we take we know what are some we will discuss that later that random number is a number between 0 to 1 and we take 2 random numbers arbitrarily we generate 2 random numbers and let us say these random numbers are r1 equal to 0.3 arbitrarily and r2 equal to 0.7, right.

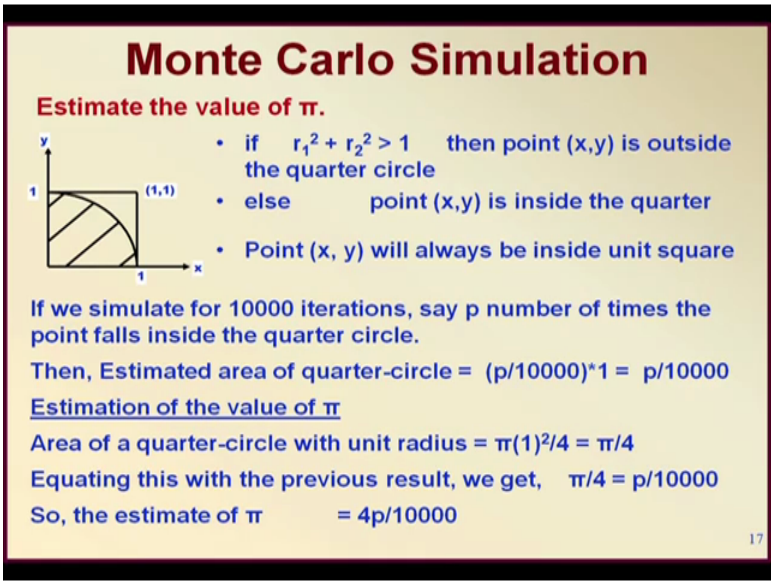
So what will happen suppose we find out the r1 square plus r2 square and then root over, what will be that value? You see this value will now become 0.3 whole square plus 0.7 whole square, how much is that? Equal to 0.09 0.09, 0.49 so 0.58, now root over 0.58. Now you see root over 0.58 is it greater than 1 or less than 1? Very simple it is less than 1. So if it is less than 1 then you see where will it fall? So you see if we plot that number so this is 1, this is somewhere it will be 0.3, somewhere here it will be 0.7 so you see this point will fall somewhere here. So this is the point that we have generated by r1 0.3 and r2 0.7 and this point is within C, right. So if you generate 1 2 random numbers so you see 2 random numbers are generated and that random number is represented by this point P1 and we see that this particular point P1 is within the quarter circle C so that is one point.

So supposing we are tallying how are many are obtained in C? How many are obtained in C? So in this tally we put a 1 under C. Suppose the next point is P2 again we generate so this is another point again it has fallen within C so another like this we keep counting, right so 4, 5, 6, 7, 8, 9 so keep doing experiment. So you see 10 again may be A also are going up, right. How many points are within A? You see there will be some points which will be within C, some points which will be not within C, so but then within A.

So if we keep counting obviously we know that this point is all these points will be within A, so whatever points we have generated they are all within A but not all will be under C, right. So finally suppose under C we have generated 75 points 75 points and within A we have generated 100 points. So you see since C by A is pi by 4, so here we get pi by 4 equal to 75 by 100 so where an estimate of pi will be 300, so it will be 3.0, so an estimate of pi equal to 3.0 could actually be obtained, alright.

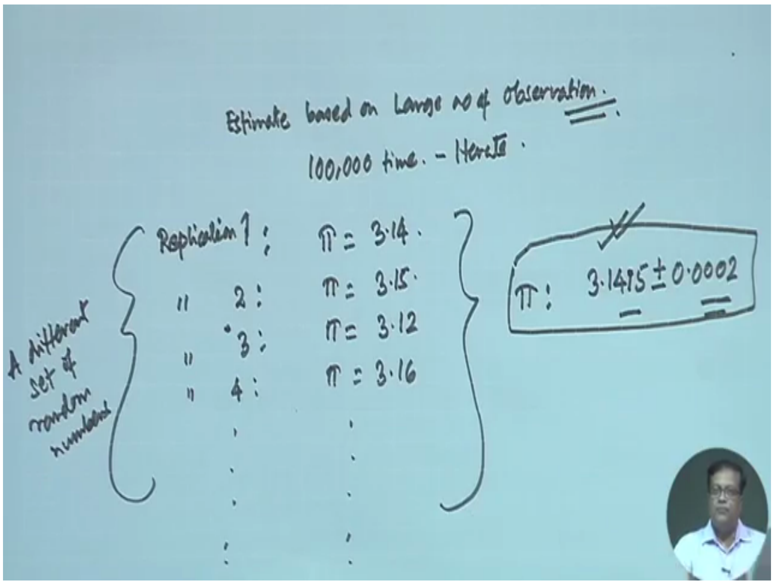
Now question is that it is only 3 it is not 3.14 then what really happens to our calculations? Let us look at it once more in this particular slide. So what has happen take two random numbers uniformly distributed let the two numbers represent x and y co-ordinates. Check if point x, y is inside the quarter circle point x, y will always be inside unit square, right. So if this is greater than 1 then it will be outside otherwise that will be inside.

(Refer Slide Time: 26:26)




So that experiment if you keep doing and if we simulate for 10000 iterations, say p number of times the point falls inside the quarter circle, then estimated area of quarter circle will be p by 10000 star 1 equal to p by 10000, is it alright. So estimation of value of pi, area of quarter circle with unit radius pi square so pi by 4, so pi by 4 equal to p by 10000 or estimate of pi will be 4p by 10000. So that is how a simple Monte-Carlo Simulation can be done and value of pi can be estimated.

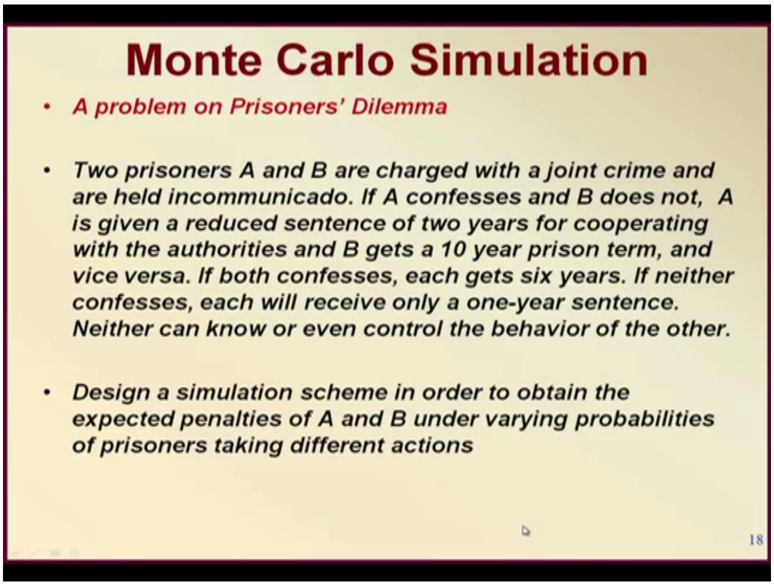
(Refer Slide Time: 27:27)




So important question out here is how do you know that value that we are getting is accurate, right you see to get accuracy two things had to be done number one number one is estimate number of observations, right estimate based on first of all large number of observations, what is large? Is 1000 large or 10000 large? Probably no for such kind of systems may be experiment for very for very large number of times may be even 100000 times, right. So may be iterate 100000 time, right so iterate.

So if you iterate for say 100000 time then probably you estimate let us say value of pi equal to 3.14, but then that is only replication 1 that is only first time. Now do it again with another set of random numbers so you can call it replication 2, suppose at that time you get pi equal to 3.15 do it again, replication 3 may be at this time you get a pi equal to 3.12 do it again may be you will get 3.16. So like this do it for a number of replications by at each replication a different set of random numbers a different set of random numbers are to be used and while using a different set of random numbers you know we can actually predict the value of pi suppose we might get the value of pi comes out to be 3.1415 suppose it has come like this plus minus 0.0002 may be for a large number of experiment we might get you know an estimate which will be a mean value and a variation, right based on standard deviations. So these are the things that we actually get for what is known as Monte-Carlo Simulation.

(Refer Slide Time: 30:11)




Another example of Monte-Carlo Simulation let us look. So this is called a problem of Prisoners Dilemma. You see two prisoners A and B are charged with a joint crime and are held incommunicado. If A confesses and B does not, A is given a reduced sentence of two years for cooperating with the authorities and B gets a 10 year prison term. If neither confesses you know and vice versa. If both confesses each get 6 years. If neither confesses each will receive only one year sentence, neither can know or even control the behaviour of the other.

Design a simulation scheme in order to obtain expected penalties of A and B under various varying probabilities of prisoners taking different actions, right. So this could be another problem where you can probably use Monte-Carlo Simulation and you know try to get answers of it, right. So please work this out and we shall discuss not in the next class but later on while we discuss queuing problems, right so thank you very much.