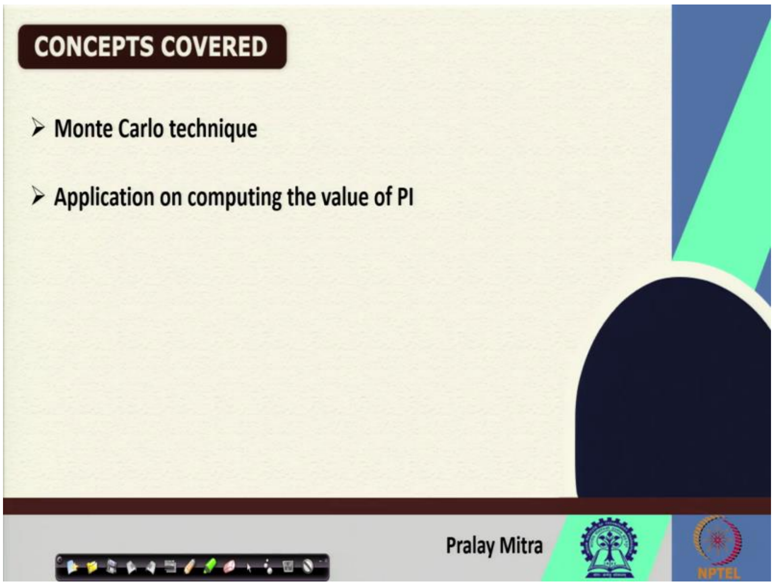
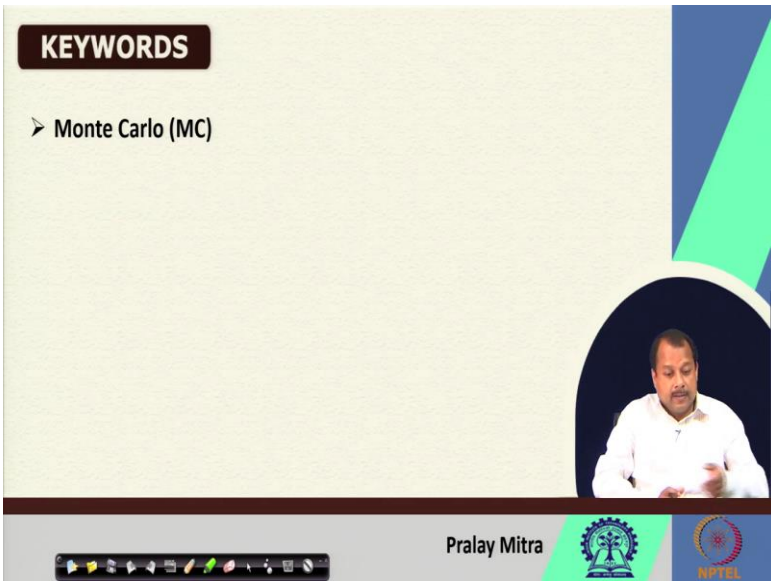
Algorithms for Protein Modelling and Engineering
Profesor Pralay Mitra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture 16
Monte Carlo (MC) Method

Welcome back, so in this week the first lecture or lecture 16 we are planning to discuss Monte Carlo method, so Monte Carlo has a lot of applications and we will start discussing the basic concept of the Monte Carlo.

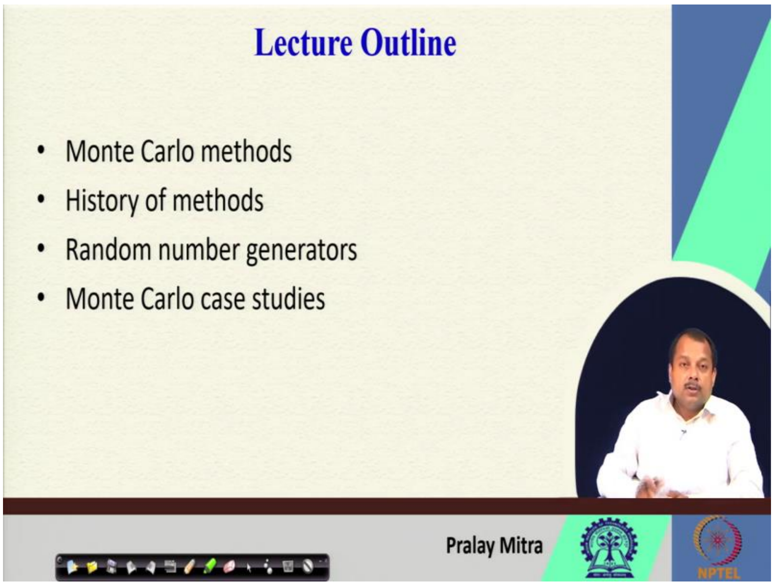
(Refer Slide Time: 00:30)





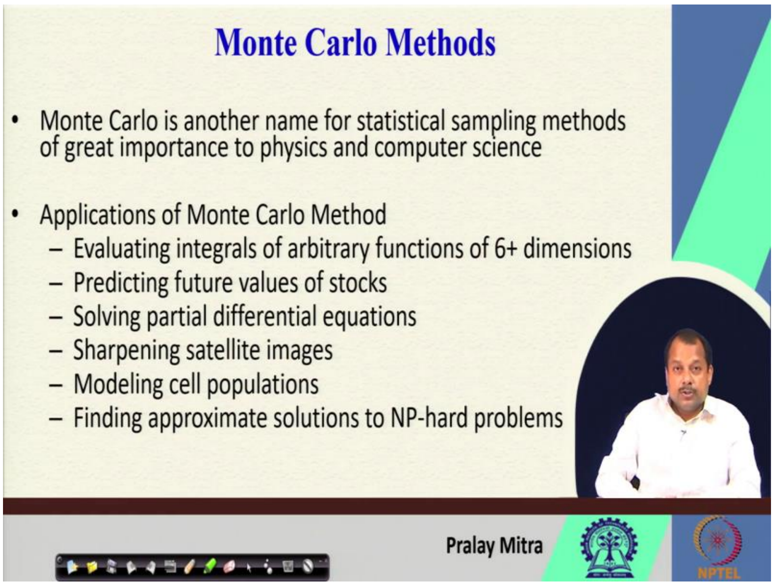
And we will also show its application on computing the value of Pi first followed by a when we are approximating some integral function, then we will move on to our actual protein relative problem, where we will demonstrate in brief the protein folding problem and protein design problem, so the concept that will be covered in this lecture is Monte Carlo technique and application on computing the value of Pi.

(Refer Slide Time: 01:04)



So, as an outline I would like to mention that Monte Carlo method, then I will also introduce to you a little bit about the history of the method and then the random number generator, its role in designing a good Monte Carlo method and finally with some case studies, so in this lecture, we will cover the computing the value of Pi then we will follow other applications and protein related applications also.

(Refer Slide Time: 01:36)

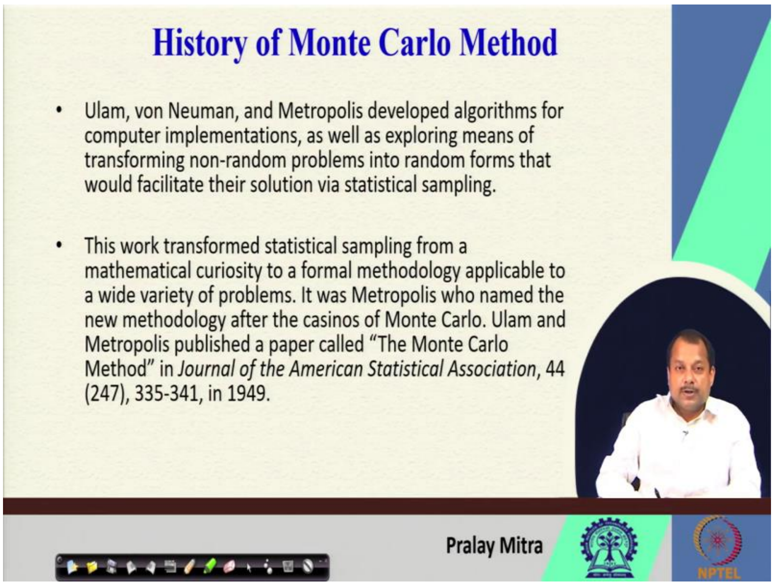


So, Monte Carlo method is another name of statistical sampling methods of great importance to physics and computer science, but by this way, I do not wish to restrict the domain and physics and computer science only, so it has a lot of applications specifically it finds its application in evaluating integral of arbitrary functions.

So, where the dimension is even more than 6 plus, predicting future values of stocks, so you understand that computational finance that is another area of application solving partial differential equations, so computational linear algebra or computational science in general then sharping satellite images, where the application is geographical information systems.

Modelling cell populations, that is another area of population dynamics, finding approximate solution to some NP-hard problems, so for that, usually this non deterministic, this Monte Carlo technique is being used.

(Refer Slide Time: 02:40)



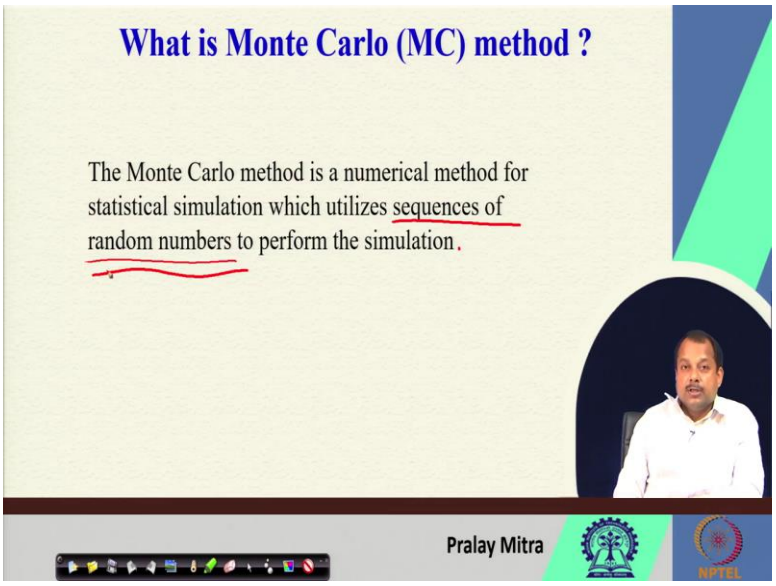
So, to give you a history, so basically, the concept has started a long back since the Bernoulli, then, Boltzmann, so they started to work on the, conceptualizing the idea of the heat, how the heat is being generated, so based upon that gradually progresses and finally, it is Ulam, von Neuman and Metropolis who developed the algorithm for computer exploration, its implementation and exploring the means of transforming non random problems into random forms that would facilitate their solution via statistical sampling.

So, at the core there is some statistical sampling and also we will see that the correctness or goodness of the method will depend upon the statistical sampling technique based upon that sampling, if it is a very random in nature or purely random in nature, then we will get a very good accuracy otherwise, there might be some problem.

The work was transformed into statistical sampling from a mathematical curiosity to a formal methodology applicable to a wide range of problems, it was Metropolis, who named the new methodology after the casinos of Monte Carlo, Ulam and Metropolis published a paper called the Monte Carlo method in the Journal of American Statistical Association long back in 1949.

The method although we will start by discussing the Monte Carlo, but we will shortly see that when we will apply it for our protein related problem, then one flavor or one specific type of the problem we will be considering, that is the metropolis Monte Carlo technique, where the metropolis criteria will be incorporated into the Monte Carlo technique.

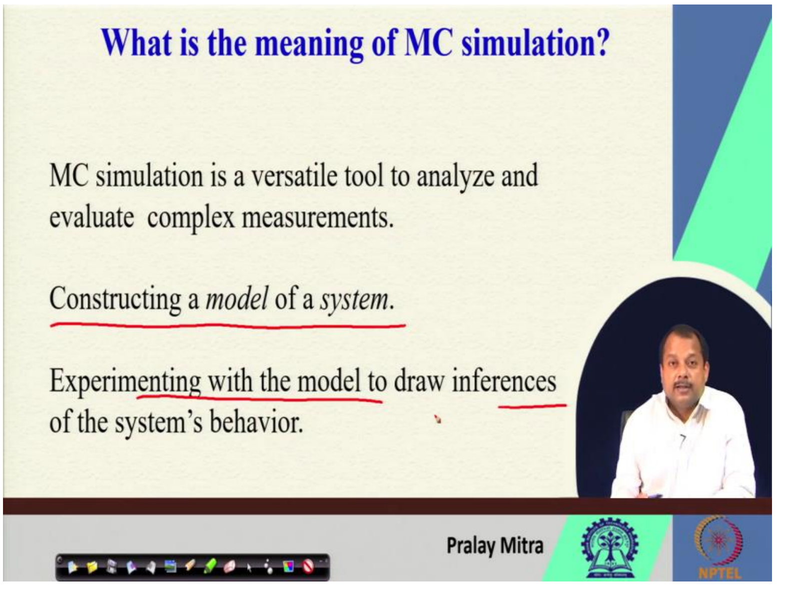
(Refer Slide Time: 04:37)



So, to summarize, what is the Monte Carlo method, the Monte Carlo method is a numerical method for statistical simulation which utilizes sequences of random numbers to perform the simulation. So, let me underline this random number. So, this random number is of great importance and that I will also demonstrate when we will discuss the application on the computing the value of Pi. And of course, it is performing some simulation.

The sequences of random number, when I say the sequences of the random number then I mean that when I generate a number of random numbers then it will be random in nature, so there should not be any pattern observed in the sequences of the random number, so we have to be careful otherwise there will be some bias introduced in the solution.

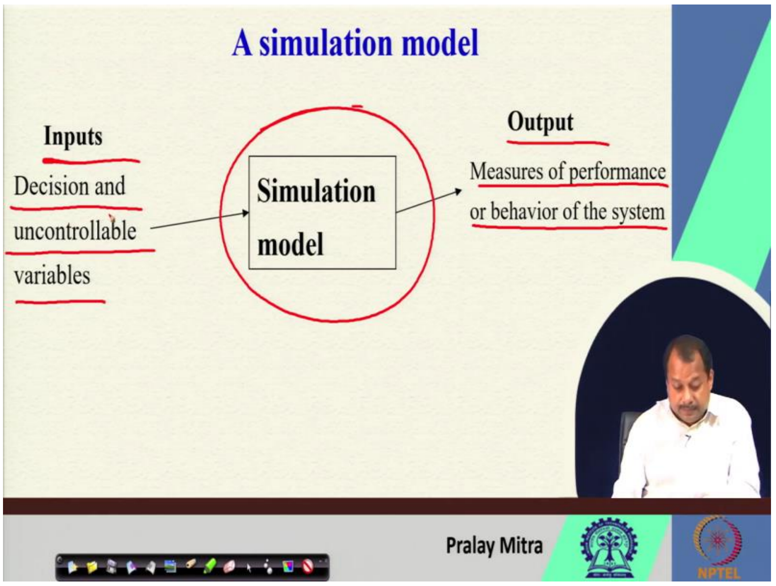
(Refer Slide Time: 05:44)



So, what is the meaning of MC simulation? MC simulation is a versatile tool to analyze and evaluate complex measurements, it construct a model of a system and then experimenting with the model to draw inferences of the system's behaviour, so that is very interesting. So, what it used to do that as the name suggests constructing a model of a system.

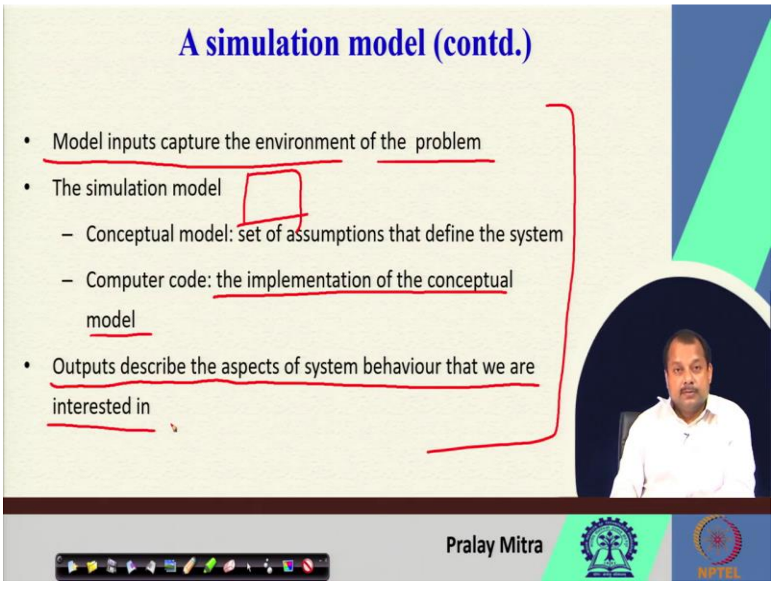
So, first one model is created corresponding to a system then it starts experimenting with the model with a hope that it will give an optimized solution and while doing this process, then sometimes it will get some good results, sometimes it will get some bad result, that good and bad will depend upon the definition of some function or mathematical equations, so that is my inference and based upon the inference, I will draw some conclusion regarding the system's behaviour.

(Refer Slide Time: 06:53)



So, if I wish to present it using the schematic diagram, then the simulation method will look like this, so I have some input the input is decision and uncontrollable variables and output is my measure of performance or behaviour of the system and here it maps the simulation model, so the simulation model maps the input of decision and uncontrollable variables to measures of performance or behaviour of the system.

(Refer Slide Time: 07:40)

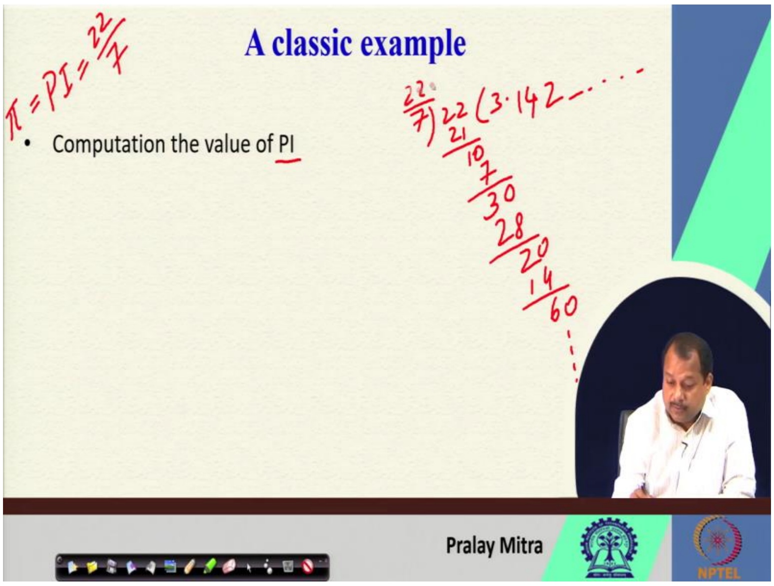


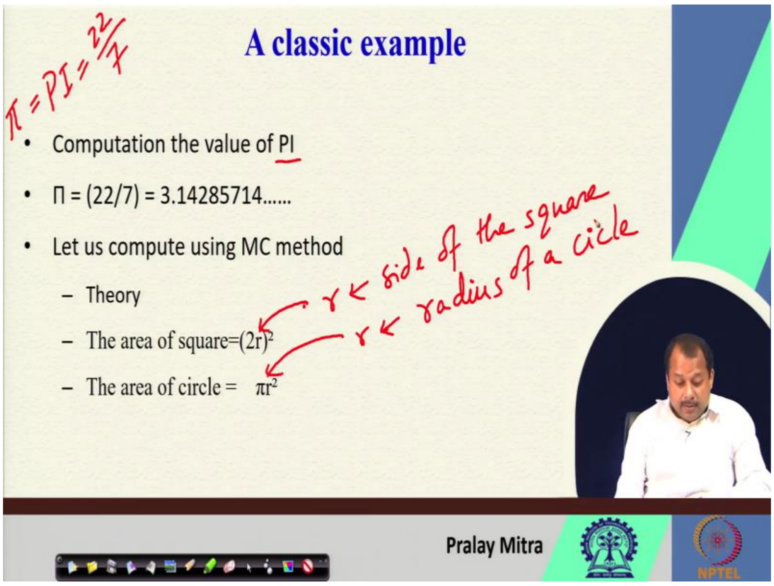
So, the model input capture the environment of the problem, the simulation model, it is the conceptual model set of assumptions that define the system and computer code that implements the conceptual model. And outputs describes the aspects of the system behaviour that we are interested in.

So, this is what we have presented medically in the previous slide, so basically we have model inputs to capture the environment of the problem and there is a black box that is my simulation model, inside that black box there is a conceptual model and the implementation of that model using some programming language, so if you wish to go to the low level implementation then definitely you have to go for some say C, C++ or Fortran otherwise using say Par, Python other languages also you can implement.

But, if you implement the in conceptual model, then after that implementation, so the input will be mapped to the output what is the output it describes the aspect of the system behaviour that we are interested in.

(Refer Slide Time: 09:18)





So, one classical example that I wish to give you is the computation of the value of the Pi. Now, all of us are aware that, what is this Pi? Pi is usually computed as 22 by 7 and we are aware that this Pi is also denoted like this Pi now, we are aware about this fact that if I keep on dividing 22 by 7 then no pattern will occur, so I will keep on dividing say if I divide say his 22 then what I will get 321 then 1 then 0 then 7 then 30 28 then this is 2 and it will keep on going.

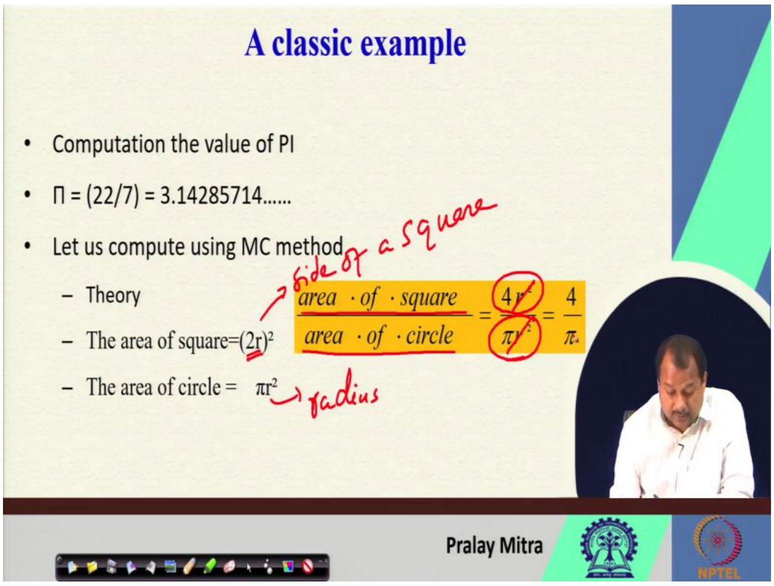
Now, the point is using Monte Carlo simulation technique I wish to compute these values, now you can understand one situation and you can also argue with me that what is the point of using the Monte Carlo technique, I can simply keep on doing the division and that way I can get the values of this Pi.

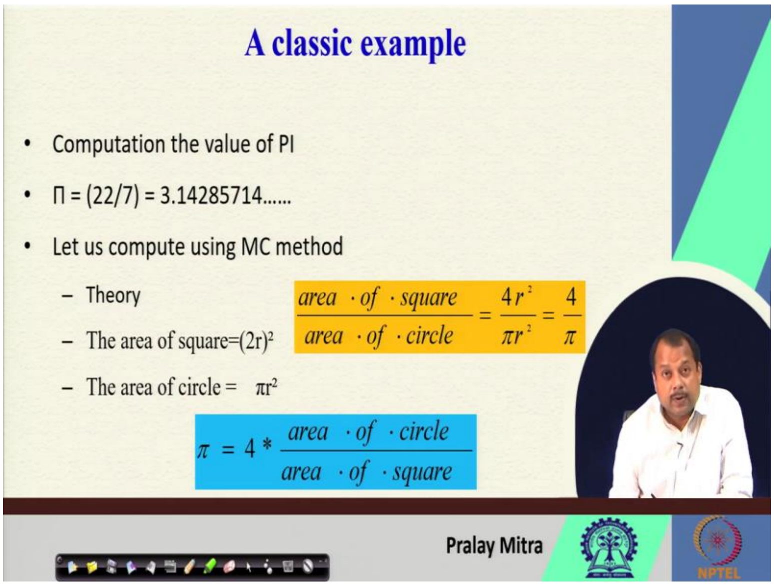
So, absolutely no problem with that one I completely agree with you, but here our intention is to demonstrate the use and application of the Monte Carlo technique that is why we picked this example case and you will be surprised to know that, this is one of the simplest and classical one where people use this Monte Carlo technique even to benchmark the parallel programming like open MP MPI or other parallel programming languages they used to benchmark using this one.

Why, that I will come back shortly, but before that, let me tell you that what is the basic idea behind this, so as I mentioned Pi equals to 22 by 7 which translates to 3.14285714 dot dot dot etc, now let us compute using Monte Carlo method, so what is our core idea or the theory behind this?

So, first of all we all know that the area of a square is 2r square where r is the side of the square then it is 2r square, now we all know the area of a circle is Pi r square where, this r is radius of a circle, so we know this fact.

(Refer Slide Time: 13:00)

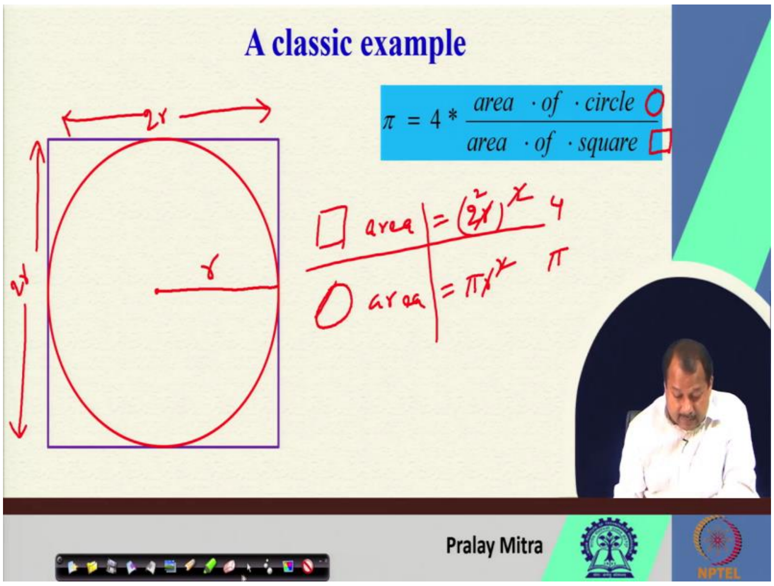




Now, if I say that area of a square divided by area of a circle, then I can write basically 4r squared divided by Pi r square, now this 4r square actually not r is the side, actually 2r is the side, so the area of the square is actually side square, so I was wrong on the last slide, so it is the side square.

Now, side of a square is 2r, so this I kept intentionally the reason I will explain in the next slide, but if I assume that 2r is the side of a square and r is the radius of a circle, then what I can write area of square divided by area of circle equals to 4r square divided by Pi r square and then crossing out this r square I am getting 4 by Pi. Now, Pi equals to 4 multiplied with area of the circle divided by area of the square, so what is this area of the circle and area of the square?

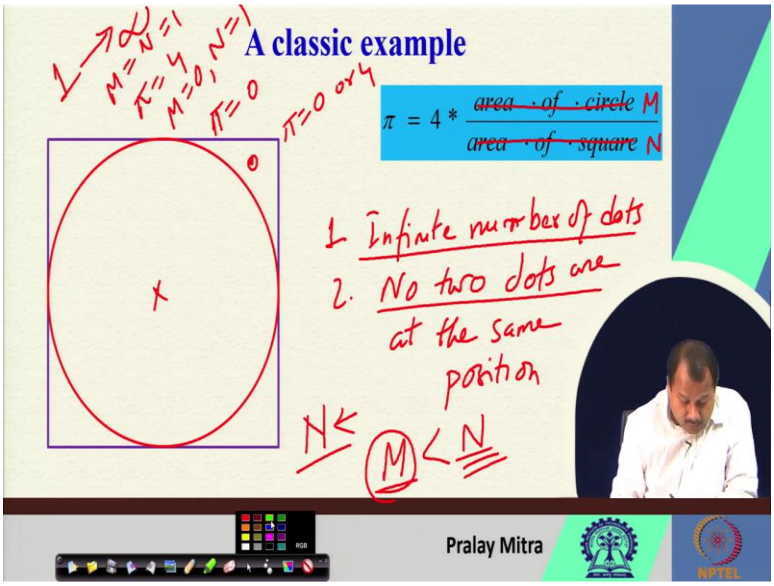
(Refer Slide Time: 14:42)



It says let us assume that, there is a square and there is circle, now if this is my centre then what I can say that, this is my r and if this is my r then this is my 2r, now if this is my 2r this is also 2r because this is a square, so this is r, so now you understand that why I said that area of a square equals to 2r whole square and area of a circle equals to Pi r square.

Now, if I divide this area of the circle and area of the square if I divide this 2 then this r square will cross out this 2 square will evaluate to 4 this is Pi and by doing this one, what I can get actually this Pi equals to 4 multiplied with area of the circle divided by area of the square, now up to this, it is clear, so if it is clear.

(Refer Slide Time: 16:21)



Then let us see how we are actually exploiting the Monte Carlo simulation method, so our proposal is basically, here our proposal is, anyway it will it will go, so here our proposal is that if I have the knowledge of the area of circle and area of the square then I can compute the value of Pi.

Now, in order to compute the area of the circle and area of the square I am not utilizing the radius information, so what I am utilizing that let us assume that I have put in finite number of dots inside this square if I put infinite number of dots inside this square and if I assume that no two dots are at the same position.

So, first of all infinite number of dots and no two dots are at the same position if these two points are valid then you can understand that the number of points which are inside the circle will give you the area of the circle and the number of points or dots which are inside the square will give you the area of the square.

So, the number of dots or points which are inside the square will give you the area of the square and the number of points or dots which are inside the circle will give you the area of the circle. Now, if I assume that infinite number is a very large number and that number I am making as N.

So, if n is such a large number say near to infinity and that number I am generating and while generating then I am making sure that the number will always lie inside the square, if I make sure that the number that I have generated will lie always inside the square then all the numbers that I have generated will actually tell me what is the area of the square.

Look at my assumption, so it says that I am generating N number of points and N is near to infinity and also no two numbers in N or no two dots in N are coinciding or are the same they are all different, which means those numbers are sufficient enough to cover the area or the surface of the square if N such numbers are generated then N will tell me or N is actually the area of the square.

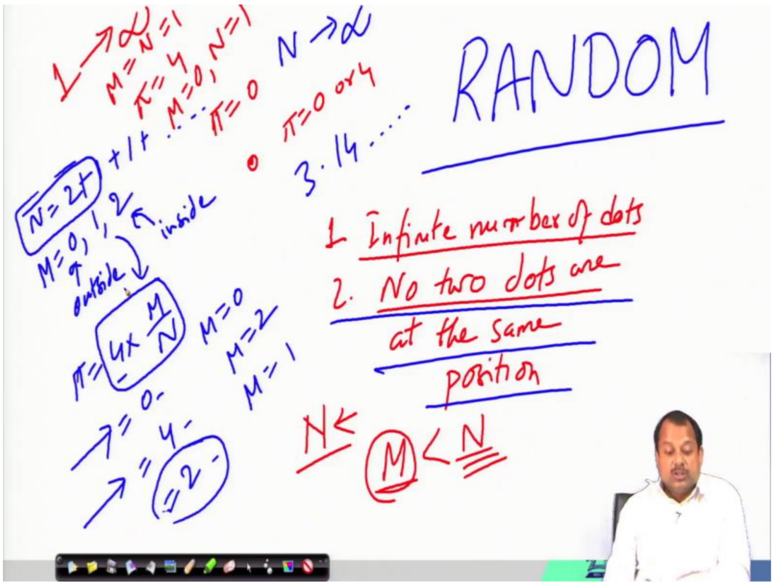
Now, if I make sure that N will always lie inside this square and no two dots or no two numbers are same or they are coinciding then I can ensure that the area of the square is basically N.

Now, the point is out of N, say M will be the number of points which are inside the circle, so I can say M number of points if I am since my assumption of M N is close to infinity then M will also be close to infinity and no two dots are at the same position or coinciding for N it will be then hold true for a subset of N that is M also, so the circle, the area of the circle will be counted by this M, so here are what I can write area of circle is basically M area of square is basically N and that way, if I have M and N then 4 multiplied with M divided by N will give me them value of Pi, very simple thing.

Now, see where the Monte Carlo simulation comes into play it says that N that you can vary from 0 to infinity if I do not assume 0, then say 1 to infinity now, if it is 1 then the point can be inside the circle or outside the circle accordingly, if it is inside the circle then M equals to N equals to 1, so Pi equal to 4, so if M equals to N equals to 1 since I generated only one point and that point I am assuming there is inside the circle, then Pi will be 4 as for this equation.

Now, if M equals to 0, N equals to 1 which means, I generated one point and that point is outside the circle and my assumption is that all the points that I have generated, that point that I have generated is within the square, so which means this must be inside the square, so it is part of N but it is not part of M if it goes outside the circle, so at this position not this one then, M equals to 0 N equals to 1 and using this equation I am getting Pi equals to 0 so, for one point the value of Pi can be 0 or 4 for two points, what is the situation?

(Refer Slide Time: 23:42)



So, let me go to here for two points, what is the situation now, so I believe you remember that equation, so these are not of our interest, now for 2 points, situation is, so N equals to 2 if N equals to 2 M can be 0, 1, 2, three different possibilities because, all the 2 points can be outside of a circle, inside the circle, all 2 points are inside the circle or 1 is outside the circle another is inside the circle.

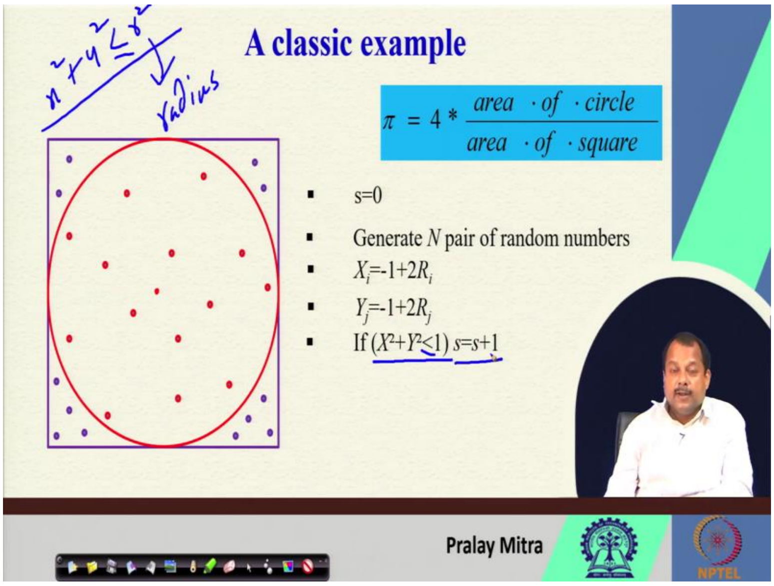
Now, following that one if I compute the value the equation is Pi equals to 4 multiplied with M by N, so if M equals to 0, then it will be 0, M equals to 0, if M equals to 2 then, it will be 2 by 2 which means 4, if M equals to 1 then it will be 1 by 2, so it will be 2, 4 multiplied with 1 by 2, so the value for 2 points can be 0, 4, 2, actually we know that value is 3.14, etc.

So, the amount of error is reducing based upon the number of points I am generating and also when I am generating the number of points, then one situation is that all the points are actually outside the circle, all the points are inside the circle and sometimes they are distributed.

As of now, intentionally I did not use the word random, but in order to make sure this second point no two dots are at the same position you have to make sure that you are generating random number sequences or a sequences of the random points and your random number generation is sufficiently random, so that no two points are same or they are coinciding.

If they are coinciding or there is a bias then basically it will go either this way or this way, but if there is no bias then perhaps, either this way or this way, but if there is a bias there is no bias then there is a possibility to will go here, that way if you keep on this value of N by 1 plus 1 plus 1 this way and if N goes to infinity, then you know the equation is 4 multiplied with M divided by N that way you can have a value which is very close to your, very close to your actual solution.

(Refer Slide Time: 27:13)



The same thing we will be doing now here, so what I am doing I am generating one point and I am assuming it is inside the circle throughout my generation if my assumption is that all the points that I have generated randomly will always be inside the square, so for within this inside the square I need not have to check, only thing I have to check that whether they are inside the circle or not, so that I am going to test and where I am going to test, so one point is inside the circle, the second point is outside the circle, next another point I generated that is inside the circle.

Next, I am generating another point that is inside the circle, I am generating one more circle inside the circle, inside circle, inside circle and then when I finish generation N the number of points that I will generate is given to me or known to me then I see that all the pink circles are actually outside the circle and red circles are inside the this circle.

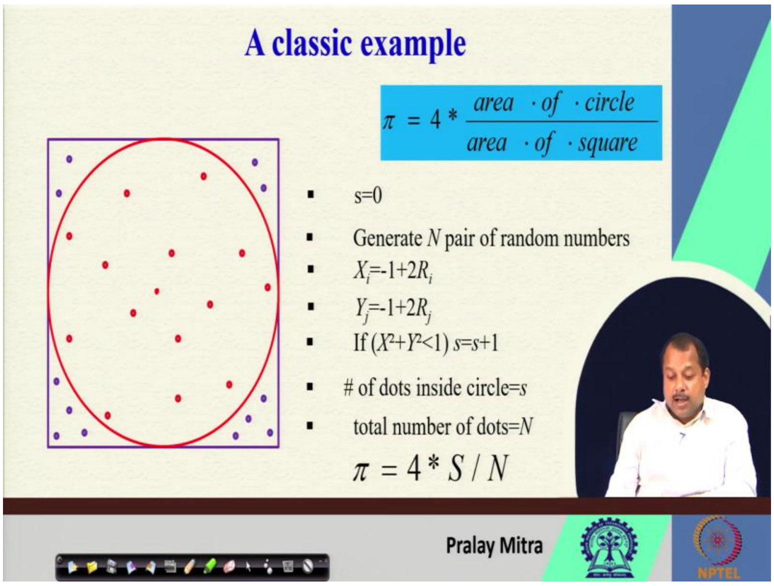
And what is this circle as you understand that this is the largest circle which will be inside the square, so that is why the radius of this circle is actually half of the side of the square or the diameter of the circle is same as the side of the square, so the theory here is initially we will assume s equals to 0, what is s number of red circles or number of circles which are inside the red circle.

Next, generate N pair of random numbers, why pair because, we are working in the 2d, so x and y both coordinate we need that is why pair of random numbers we have to generate it and that is N, so in number of pair of random numbers I am generating since N number of pair of random numbers we are generating, so I can make sure that all the N numbers are inside the square and that way I will have the area of the square in terms of the number of points that I have generated.

Next, xi equals to minus 1 plus 2Ri, so based upon the coordinate system of your square and the circle, next Yj using the same thing minus i plus 2Rj then, if so this is your equation x square plus y square less than 1 actually it is x square plus y square less than r square, where r is the basically radius, so that is the standard equation for the circle.

Now, here you can put less than equals to that is not a problem, so you can put that one but whatever it may be, it should be uniform considering whether you are considering the point on the circle or not, that way if it is inside then you are implementing the value of s.

(Refer Slide Time: 30:33)



So, by this time you know that what will be the area, so the area it will be basically number of dots inside the circle is S and N is the total number of dots, so it will be Pi equals to 4 multiplied with S by N, so that is the equation, so I believe you are able to understand this one.

So, in the next lecture, I will elaborate this using an algorithm, so that you can able to implement and then it will be useful when we will do Monte Carlo simulation or protein folding and protein design or say protein engineering problem, thank you very much.