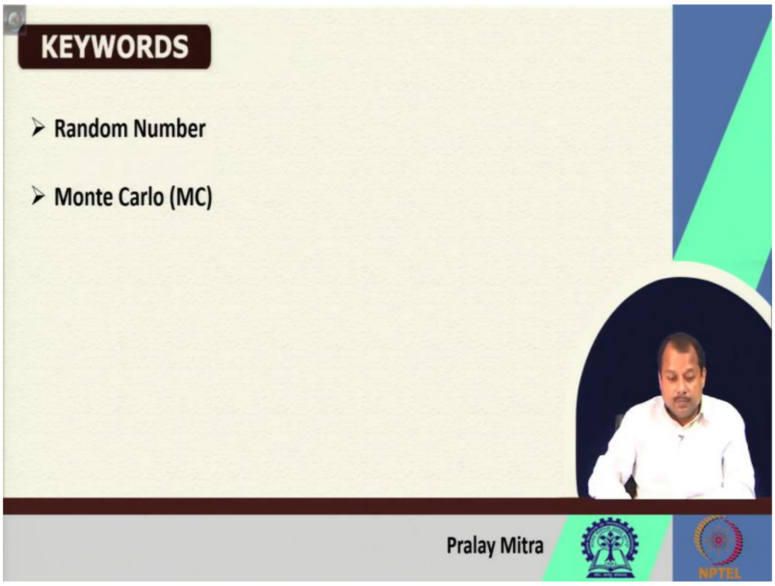
Algorithms for Protein Modelling and Engineering
Professor Pralay Mitra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture 18
Monte Carlo (MC) Method (Contd.)

Welcome back. So, we will continue with the Monte Carlo method. But, in this lecture we will introduce you one example that is the integration computing the integration. So, we will show you the approximation the kind of Monte Carlo techniques we can use the effect of the randomness and then slowly we will move on to our actual problem of say protein folding, etcetera.

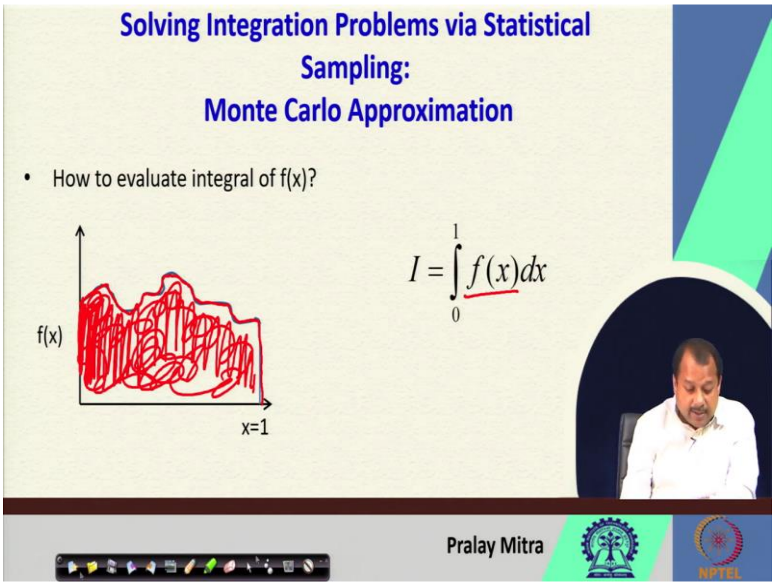
(Refer Slide Time: 00:46)



KEYWORDS

- Random Number
- Monte Carlo (MC)

Pralay Mitra

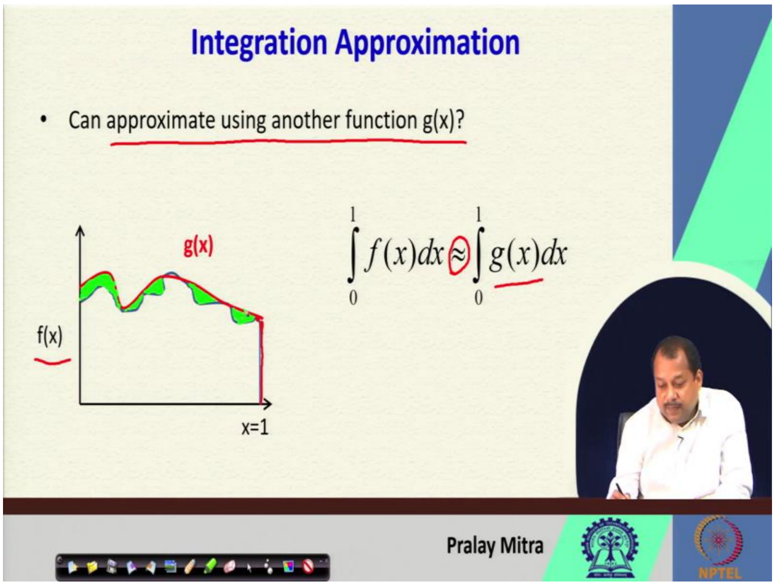


Solving Integration Problems via Statistical Sampling:
Monte Carlo Approximation

- How to evaluate integral of f(x)?

$$I = \int_{0}^{1} f(x)dx$$

Pralay Mitra



Integration Approximation

- Can approximate using another function g(x)?

$$\int_0^1 f(x)dx \approx \int_0^1 g(x)dx$$

So, the application of the Monte Carlo that is our topic which will be covered today. Now, the random number and Monte Carlo is the key word for us. So, we are interested to compute the integration of this equation. As you can see this is a very simple integration and that is why it is from 0 to 1 f x dx. So, this integration I used to calculate and during the calculation I wish to use the Monte Carlo technique.

If I say that this is my equation then pictorially it can be seen as something like this. So, here what I am interested in so, this is my function, so, this border indicates the f x value so, which means that inside this everything is my integration part. So, I have to calculate this area using this integration.

So, now, the primary problem here is that this f x, one way could be that if there exists one f x and that f x is not we are not able to say formulate that one here properly then is it possible that we can say approximate that to some other function gx which is known to us and if we approximate that one, then how will it look like on this diagram.

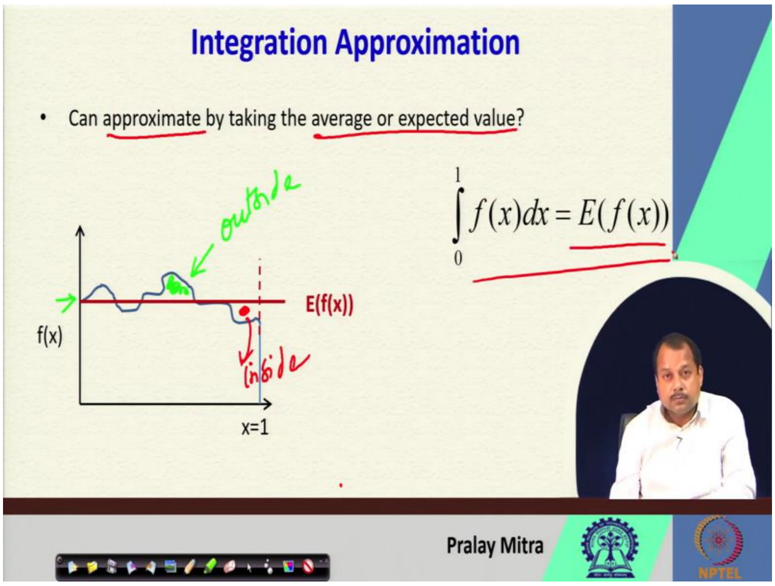
So, something like this that you see the red line and this line of course will end here. So, the same thing I am interested to compute, but I am telling this is an approximation. So, we are interested to approximate to another function gx and we are interested to calculate. When we are doing that one you see that gx is kind of computing the area which is little larger compared to the value of f x or the integration of f x.

So, our motivation of posing this problem like this is that when say we will be working with the protein score function say for a given protein structure then the stability of that protein will be given by some energy function or some score function or we can in general say that a protein will be stable if its energy is within this limit and that way if we compute the energy value of all the stable proteins and if we assume that there is some technique, which is kind of designed by God. So, that each of the energy values for each of the proteins can be calculated correctly.

So, if that is the situation then we can consider that perhaps f x is my function and if I take the integration over that one then I will get one region. So, within which if somebody falls then that is my stable one otherwise it is not. Now, that is as I mentioned is designed by God. So, we are trying to copy the concept of God, not the concept of the God rather what I am interested to say that we use to copy the way God has designed that score function and that score function is gx not f x. And that way you see that it is enclosing little more area compared to f x, it is fine.

But you understand that these green regions the area which although I am drawing as green, but you can consider that as a green region which means that I am not sure that whether the protein will be stable or not in that region because it is not within f x within my approximation, within my approximation, but not within f x. And I agreed that let it be. So, if that is there then whether I can approximate gx also using some Monte Carlo technique is my interest.

(Refer Slide Time: 06:06)



Integration Approximation

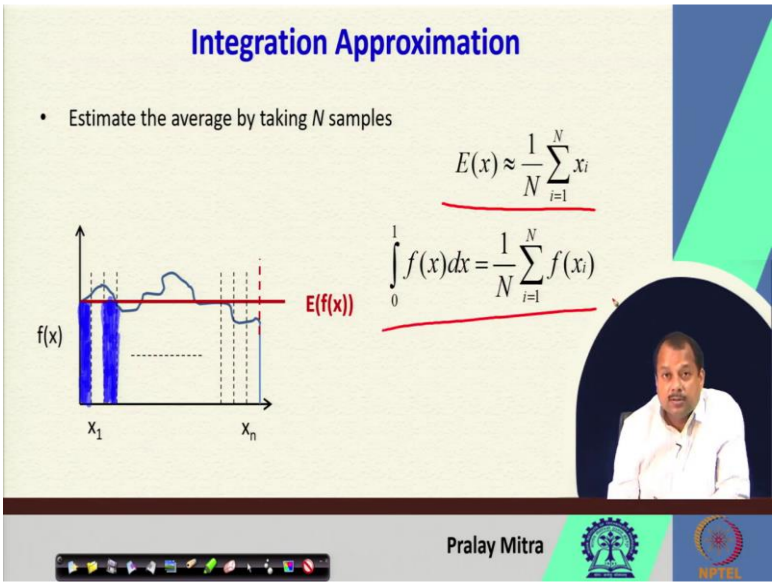
- Can approximate by taking the average or expected value?

$$\int_0^1 f(x)dx = E(f(x))$$

Otherwise, what I may think of that is it possible to approximate by taking the average or expected value? The average or expected value means that integration of 0 f x dx varies from 0 to 1 it will be E f x. So, that E f x will now be plotted on this graph like this red line that you can see I am drawing one green marker here. So, this red line that is my approximation.

And when I am doing this approximation by these approximation you see that some of the cases are which are relevant but is outside here which are not relevant is inside. So, that is because I am so, approximating and the average of the expected values. So, both way, am I am I can do the approximation, one using some another functions say gx, another by taking this E of f x some expected values.

(Refer Slide Time: 07:40)



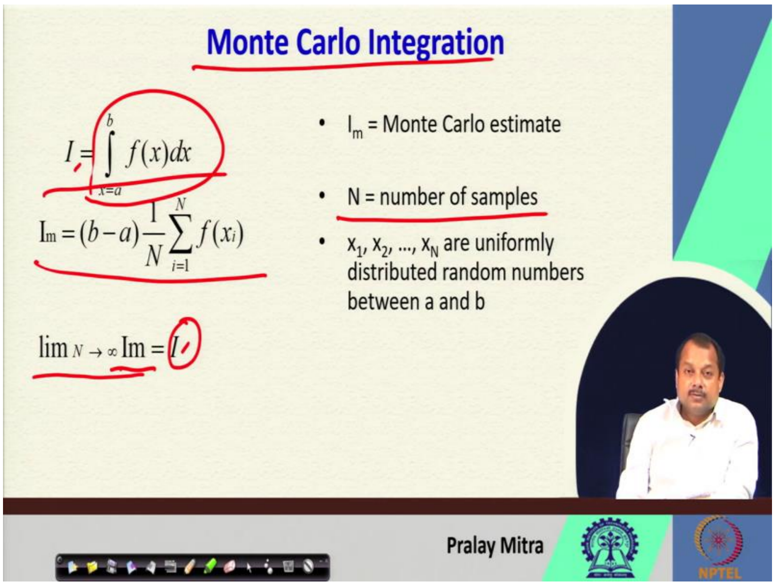
Integration Approximation

- Estimate the average by taking *N* samples

$$E(x) \approx \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\int_0^1 f(x)dx = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Pralay Mitra



Monte Carlo Integration

$$I = \int_{x=a}^{b} f(x)dx$$

$$I_m = (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

$$\lim_{N\to\infty} Im = I$$

- I_m = Monte Carlo estimate
- N = number of samples
- $x_1, x_2, \ldots, x_N$ are uniformly distributed random numbers between a and b

Pralay Mitra

If I do this one, then combining these two concept what I can say estimate the average by taking N samples. So, E x approximately 1 divided by N, summation of i equals to 1 to N xi. So, that how will be interpreted in this particular graph is very simple. X1, one vertical dotted line; X2, another vertical dotted line; X3 another vertical dotted line, dot, dot, dot up to XN.

So, what I am doing now is I am dividing the area into small, small regions and then I am calculating them. So, 0 to 1, f x dx equals to 1 divided by N. So, I am using this equation and the equation from the previous slide where integration of f x dx varies from 0 to 1 equals to approximating to E x. And by writing E x approximately 1 divided by N summation one equals to 1 through N, xi equating this to I am getting this equation.

So, these two things. So, combining these two I am getting this equation, this equation part and it is implementation or it is the significance in this particular context combining these two if I wish to implement then in the context of say Monte Carlo simulation, what it will be?

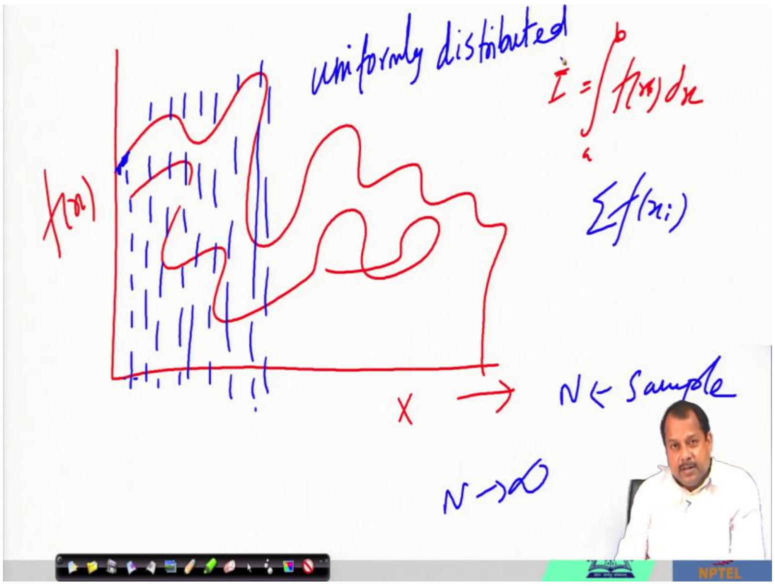
I equals to f x dx where the range varies from x equals to a to b. Now, I m the Monte Carlo estimate equals to b minus a 1 divided by N summation of I equals to 1 through N, f x f xi. Now limit M tends to infinity I m Monte Carlo estimate equals to I. This I is this I. And what is this M? It is Monte Carlo estimate that I mentioned. What is N? Number of samples and X1, X2 and XN are uniformly distributed random numbers between a and b.

So, if it is the situation then I mean that if I am interested to compute the integration of this one then using the Monte Carlo integration I can compute where I need to calculate this one and then limit N tends to infinity will give me I. Now, you remember the situation of our calculating the value of pi in that context the classic example.

So, there was an equation x squared plus y squared less than equals to r square and I mentioned that if the number of points that I generate is close to infinity and the assumption is that no two points coincide, I mean no two points are identical in nature in that the least of the numbers, list of the points, then I can get close to the correct value of pi. So, in this case also you see that the equation is Monte Carlo estimate I m, it goes to b minus a that is the range 1 divided by N summation I equals to 1 through N f x f xi.

Now, I am dividing that one into N parts, so, N number of samples. So, if N goes to infinity, so, infinite number of small parts I create and for each finite part if I compute the value of f x, then I will get I which is the integral part of here, which is the integration.

(Refer Slide Time: 12:36)

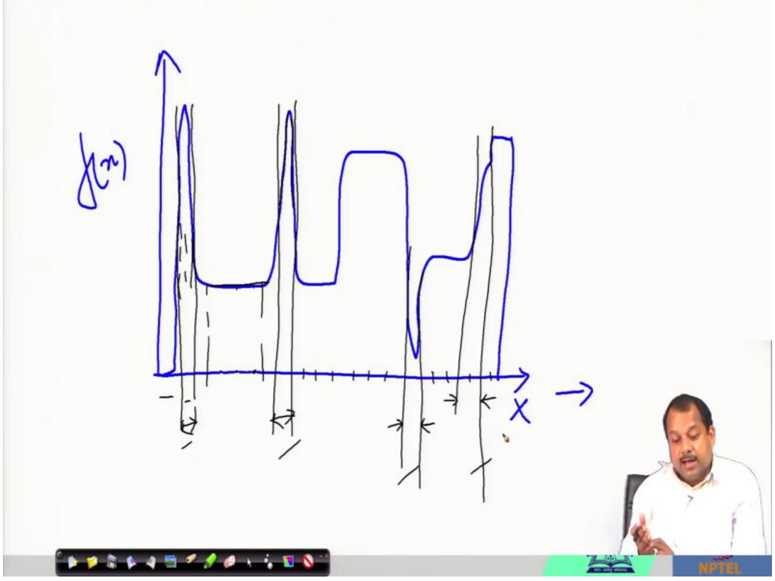


Now, let me give you this one. So, we started with x, f x and there was some function and it says that say integration I need to compute a to b if I say f x dx, if that is my integration or say estimate. Now, I need to compute the area inside this one this area, I need to compute that one. In order to compute that one, I mentioned that you divide this into different samples and for each sample you compute the value f xi, you take the summation for all those samples assuming that there are N number of samples, then you will get the estimate.

Now, if N increases and it means such that if N tends to infinity very large, so small ranges which means, the changes are error which you will be incorporated or the approximation which will take place will be very small in nature. So, overall, the error will be very less. So, if I keep on generating so many cases, this so many cases I am emphasizing, I emphasize this one when I generated the points during pi calculation here also and when I will do protein design or protein folding.

Then also if I generate more and more samples in this case intervals, then I will reach too close to the actual solution. Now, up to this it is fine, absolutely fine. Now, based upon the application time to time, not only the definition of the random number but the distribution of the random number, the word uniformly distributed. It will change time to time, why? Because you will see that when I am calculating the value of pi, I want that uniform distribution on all over the surface of the square.

(Refer Slide Time: 15:33)



It is fine, but what if this is your function? What is the definition of the uniformity here? Now, you see that here in this region there is a steepness, so, it is increasing, decreasing. So, if your distribution is such that, this is one interval say, this is another interval. This is third, this is fourth, fifth, sixth, seventh, eighth, ninth, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty, twenty-one, twenty-two, etcetera. These are your distribution then you see, this is even, so, it is a flat region and in this flat region, if I do these divisions, it is not a problem, even if I have only one region here these are not required.

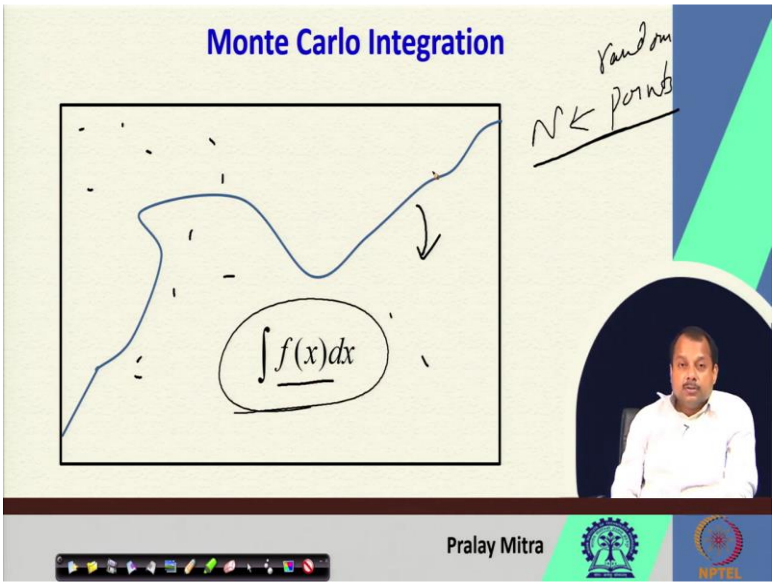
So, only one then also you see that there is not much I am losing in terms of the approximation point of view. So, when I am doing approximation, then also I am not losing much, but in this case, because of the steepness, if I do this, then also I am losing lot. So, what I need I need more and more finer sampling or distribution at this region, at this region, at this region, at this region. So, these are required.

So, based upon the nature of the problem, so, the uniformity or the distribution also has an impact and when I say uniformly distributed that may be say good for one kind of problem like computing the value of pi which was a classical example and very easy one, but for computing so, calculating the integration of this kind of function where at some position there is a very steepness at some position there is a flat.

So, having uniform say distribution or sampling make a lot of difference. So, you have to be careful reason is that the landscape of the protein or the energy funnel, energy landscape of the protein or the folding funnel of the protein is very rugged in nature. So, which one is the good, which one is the bad, it is not always possible to understand. And for some protein one particular nature is there, for another protein another particular nature is there. So, those problems are there.

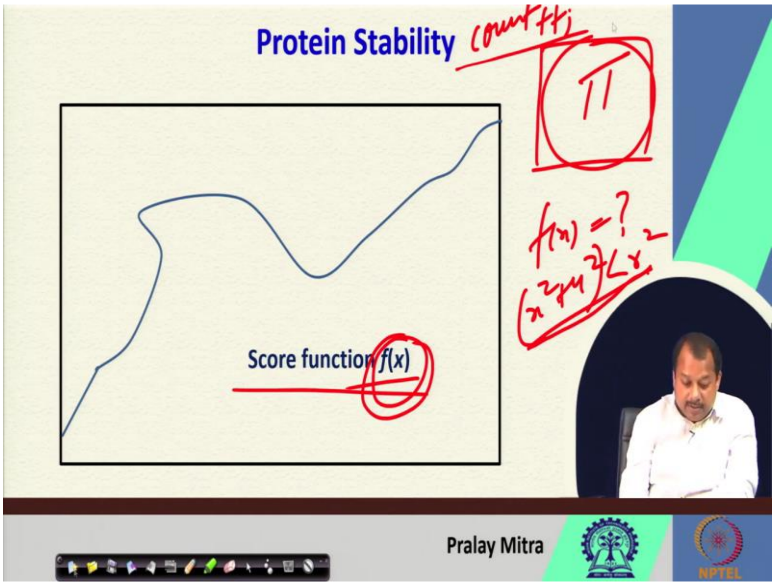
So, we have to build those problems. So, apart from the situation that okay, we are going for Monte Carlo simulation technique and when we are going for the Monte Carlo simulation technique, then there will be at the code there will be a random number generator. So, that random number generator will be very good in nature, there is no, there will be no cycle no pattern and the range is very high. So, even if all those situations are there then also some problems may arise because of this kind of nature of the function.

(Refer Slide Time: 20:20)


Monte Carlo Integration

$\int f(x)dx$

Pralay Mitra


Protein Stability

Score function f(x)

Pralay Mitra

So, in the Monte Carlo integration we as of now have discussed this. So, there is a function say f x dx that we wish to compute the integration of that we wish to estimate. Now, there is a very rough or say curved line that you can see. So, down to that where this is written this so, this region is my region of interest that I need to estimate not this one and for that also you can adopt the same technique that we have discussed for that computing the value of pi.

That within this say rectangle you generate a number of say N number of points of course, random in nature. So, random points you generate within that region. After generating that N random points. So, this will be here everywhere. Now, we have to test, last time it was x square plus y square less than r square in this case, you have to test this function, whether it is inside this or outside this. If it is inside, then you increment otherwise you do not increment that way you can have an estimate also here.

The similar thing if I extend for the protein and assuming the protein stability score function is say f x and the nature is also something like this, then also I have to estimate or say after looking at, after looking at my function, then I have to understand that at which position I am and whether that is an acceptable one or not, because one thing you should remember that in the context of the protein stability when you are say analysing that one.

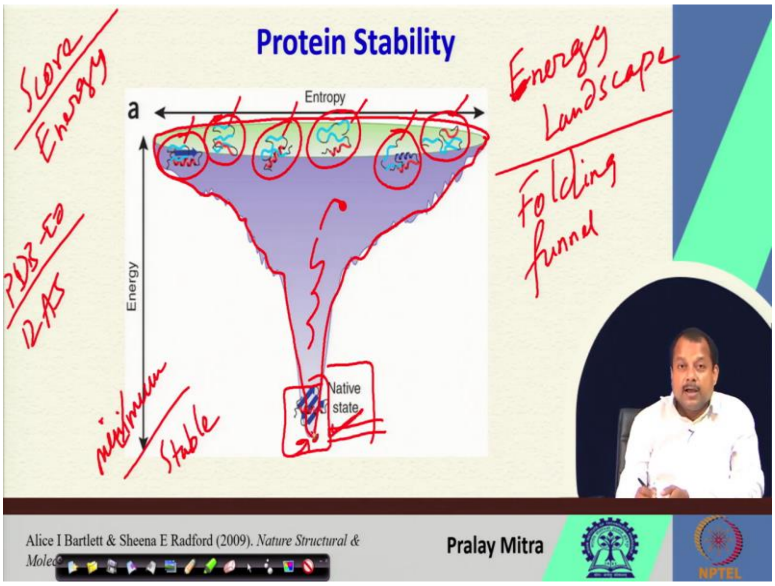
So, what will happen that randomly you will generate one say I should not say solution randomly you generate one instance so in case a protein folding it will be one 3D structure in case of protein design or protein engineering, you will generate one protein sequence. So, one instance you will generate randomly again at the code there is a random number using that random number you are generating one instance. After generating that instance then you have to consider whether it is correct or not.

So, in case of that square and circle your function was f x was what x square plus y square less than r square. It was your equation at that time. Now, in case of this core function when it is a protein stability, it is a concept of a protein stability in this context, your score function is going to be something some score function and when that score function is with you, then you have to test in the context of that score is whether it is a correct or not.

If it is a correct then you accept, if it is not correct then you reject. Again what I was to mean that during the Monte Carlo simulation in the context of a protein folding or protein design, I am generating one random instance, in case of protein folding it will be 3 dimensional structure, in case of protein design it will be a protein sequence then I will check its fitness based upon some stability function something like this f x and something like this x square plus y square less than r square which was in the context of calculating the value of pi using this square and circle concept.

So, once you generate that one, I mean in this case you generate that acceptability of the f x then you can consider whether to accept that or to reject that, if you accept that one, then like that value or pi or computing the value of pi it will be count plus plus, if not, it will be rejected. So, either you will consider that move or generation of that instance or you will not consider that one. So, it is based upon that one. However, in this case the score function or f x that I am giving is kind of a schematic, actually it is not true for the protein folding.

(Refer Slide Time: 25:56)



Protein Stability

Alice I Bartlett & Sheena E Radford (2009). *Nature Structural & Molec*

So let me show you the energy function for protein stability. This figure I have taken from this literature that is mentioned here nature structure and molecular biology published in 2009 this is also called as the folding funnel or the energy landscape.

Detail we will discuss later it will be required for us, but for today for this lecture’s completion, when we are moving from integration of some function to some actual score function of a protein, then you can see the entropy or different proteins structure. So, those are given here. So, these are different structures and there is some score function for the time being let us assume that score function is the ideal one.

So, that score function is being evaluated for the structure and I got some help value in this case is the score value and you should note this score or energy we will use interchangeably. So, both are up similar concept in this protein stability. So, we will use that interchangeably.

So, if we evaluate that function and compute the energy or score of each entropy or different instances or different structures for protein folding problem, then I will get that one and if I plot then what I am getting, I am getting some function like this, you see this function where this region indicates this region just now what is I am in, I am drawing the boundary using the red line.

So, within that one protein decides. So, if I take one particular protein say one particular protein who say PDB Id say 12 AS. So, all the conformations I generated that one I evaluated the energy of all the conformations or all the different instances when I say conformation it means that different orientations or different organizations. So, this is one conformation, this is another conformation and this is another conformation and this is another conformation, but you see that all belongs to only one protein. So, different organization.

So, as if this is one, this is one, this is one, this is one, this one this one this one, so, these different, these are different conformation organizations. So, if I evaluate that one, then the energy of value will lie within this region, which are inside this red boundary, it will be inside this one and the minimum energy is in this way, this is my minimum. So, which means, this is my stable also. And that way, this structure which is at the tip of this region, so that is called as the native state or biologically valid state.

As of now, this much. So, later we will see different variations of this folding funnel or energy landscape, and we will see that it may not be the correct one for always that we will discuss later, but for the time being we can consider that this is the region where all the energy values will lie and this region is the minimum one, the native state will come here.

So, during the Monte Carlo simulation, my job will be randomly generate some structures or conformations and evaluate that conformation. So, that what I can see that whose energy value will be minimum is my guide or is my native state. So, in my iteration process, I need have to accept all what I will do that at any instance say Ith, Ith instance.

So, what is the minimum energy I am getting and corresponding to that what is the conformation of the structure that I know. Now, I generate another structure randomly I evaluate its energy if the energy new energy is less than the Ith energy then I will accept the new one or I will replace the existing Ith one by this new one, its energy and conformation. Otherwise, I will not take that one, that way say I will come down, come down, I will come down, come down, come down, come down.

And with the hope that if I say generate infinite number of possibilities, then I will also generate this one and when I will generate this one, I will evaluate the energy which is minimum and I will keep that one. So, that is it. We will elaborate this more detail in the next lecture and following lectures. Thank you very much.