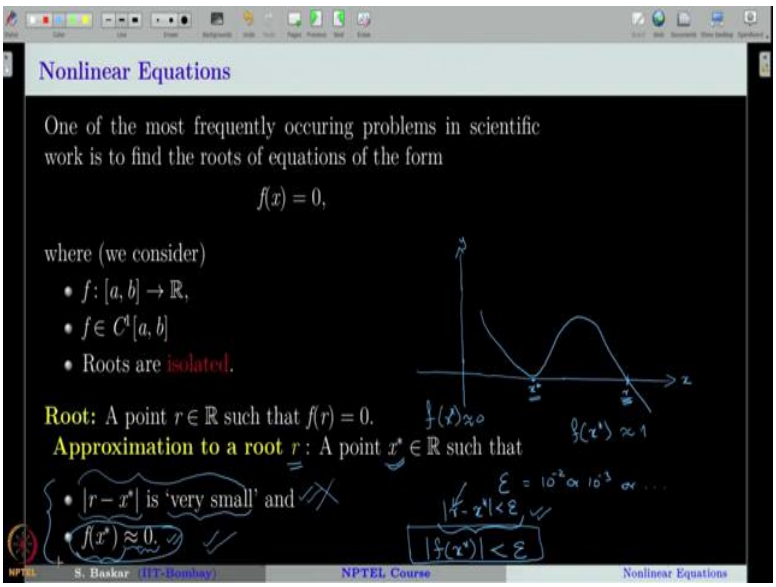
Numerical Analysis
Prof. S. Baskar
Department of Mathematics
Indian Institute of Technology – Bombay

Lecture – 27
Nonlinear Equations: Overview

In this lecture, we will start a new chapter on Nonlinear Equations. In this we will today discuss a very basic and very interesting method called bisection method. Before going into the method, let us define our problem.

(Refer Slide Time: 00:38)



And then we will see some important overviews of the methods that we are going to discuss in this chapter. One of the most frequently occurring problems in scientific work is to find the roots of equations of the form $f(x) = 0$. Where f is a given function. In our lectures, we will take the function f to be defined on a closed interval $[a, b]$ and it is a real valued function. We will also assume that f is a C^1 function, defined on $[a, b]$.

What is mean by C^1 function? f should be differentiable at all the points in the interval a, b and it is derivative, should also be a continuous function in the interval a, b. This we will always assume throughout this chapter and also we will assume that all the roots of the equation $f(x) = 0$ or isolated roots. What is mean by isolated? Suppose you have a function $f(x)$ whose graph looks like this say for instance this is the graph of the function f.

Then the point of intersection of the graph of the function *f* with the *x*-axis say it is denoted by r is called a root of the equation $f(x) = 0$. When will we say that this root is isolated, when we can find a small neighbourhood of this root *r* such that there is no other root of the equation exists in this interval. Then we will say that r is an isolated root. So, we will always assume that our equation has isolated roots.

For instance, the function whose graph looks like this has two isolated roots r_1 and r_2. And you take another function, something like this and then it touches *x*-axis and stays there for some time and then goes. Now, you take any point here it is a root of this equation. Now, can you find a neighbourhood of *r* in which there is no other root lies? no. Because the graph of this function touches the *x*-axis and then stays in an interval and then again it goes up.

Therefore, you take any small interval, however small it is, it will always have many other roots of the function. So, this *r* is not an isolated root. So now, I hope you understood what is mean by an isolated root? And of course, I have already defined what is mean by a root of this equation? Now, our interest in this chapter is to devise methods that can give us approximation to the root of the equation.

What is mean by an approximation to a root? Well, to call a point say let us call it as x^*. If you want to call x^* as an approximation to a root then we need to check two conditions one is $|r - x^*|$ should be very small. And when you plug in that x^* into your function, the value that comes out should be very close to 0. How small it should be and how close it should be these are all depends on once own interest.

But generally, once you fix, what is something that is tolerable to you, with respect to that tolerance level. You will see whether your x^* can be considered as an approximation to your root or not. Generally, what we will do is we will take a small number ϵ. We will call it as the tolerance parameter. It can be something like 10^{-2} or 10^{-3} or something like that small number.

Such that we will check whether, $r - x^*$ is less than ϵ. Of course, when you are working with numerical methods this kind of conditions are not possible because we do not know what is *r*? That is why we are going for an approximation. Therefore, practically this is not possible for

us to check. Nevertheless, this is a criteria just theoretically to accept some number x^* as an approximation.

Well, the next condition is that $f(x^*)$ should be very close to 0. Maybe we can also check $f(x^*)$ is you can take this as modulus is less than ϵ can also be checked. Now, the question is why are we having these two conditions? Why cannot we just impose this alone? Generally, we will be tempted to put only this condition because that itself intuitively tells us that x^* is very close to the root.

But sometimes what happens is, the graph of the function may be like this and then it increases very rapidly in a small neighbourhood of the root *r*. Now, you may be capturing your x^*, pretty close to *r* but it is value, may be very large. This may be something like say 1 or something like that. In that case, even though x^* is very close to *r*, your function value may be very large.

And therefore, there is no point in declaring x^* as an approximation to *r* when $f(x^*)$ is something like order of 1. So that does not seem to be nice. Therefore, this first condition alone is not enough for us to declare that x^* is an approximation. So, we also need x^* to be in such a way that the function value is almost 0. Therefore, we also need this condition.

Now, you may ask why cannot we take only this condition and not check this condition. Well that also has some drawbacks. Let us take another example where the graph of the function goes like this. It goes very close to *x*-axis and then goes back and then comes and hits the *x*-axis somewhere here. And therefore, by definition, the root of this function equation is *r* which is here.

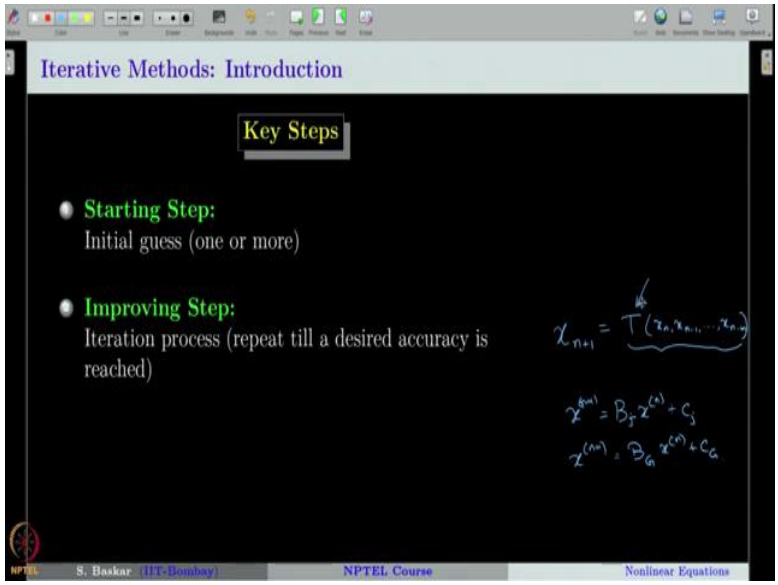
Now, if you impose only the second condition that is, if you only impose $f(x^*)$ is approximately equal to 0 or if you are equivalently imposing this condition when you are working on a computer. Then you may tend to capture a point which is here to be an approximate route for your equation. Because this distance may be less than ϵ and therefore, you may wrongly capture a point very near to this as an approximation to the root.

Whereas, the actual route is far away from your point x^*. Therefore, to declare that a point x^* is an approximation to your root, you have to theoretically at least impose these two conditions.

In our course, whenever we say that a point x^* is an approximation to a root of the equation, $f(x) = 0$ it means $|r - x^*|$ is something very small and $f(x^*)$ is also very small.

In this chapter, we are interested in developing iterative methods to capture an isolated root of our equation. Now, what is mean by iterative methods? Well, by this time we know what is mean by an iterative method. Because we have already introduced some iterative methods in linear systems, as well as in the section on computing Eigen values and Eigen vectors. Therefore, we know what is mean by iterative methods? But still we will repeat it once again.

(Refer Slide Time: 10:28)



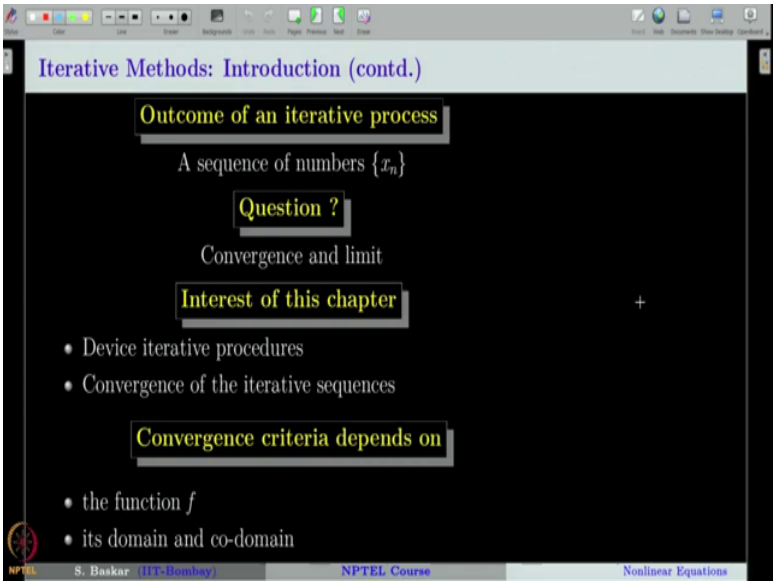
The key idea is to start with an initial guess sometimes the initial guess is just one point or sometimes it may be more than one point also. In Jacobi method and Gauss-Seidel method and all we started with only one point. But here in nonlinear equations you will see that there are some methods where we will start our initial guess with two points. Therefore, we will say that the initial guess may be with one point or more than one point.

And once you have this then you have a procedure that is some formula will be given where you will have x_{n+1} given in terms of x_n, x_{n-1} and so on up to some x_{n-m}. So, if it is only one point that we take as initial guess then *T* will only depend on x_n. If you are taking two points then it will depend on the previous iteration and previous to previous iteration and so on. But we will have a formula. Different formula leads to different methods.

Jacobi method had a formula which was written as $B_J X^{(n)} + c_J$. That is what we called as $x^{(n+1)}$ in the case of Jacobi method. And similarly, in the Gauss-Seidel method we have another

formula that was denoted by $B_G x^{(n)} + c_G$. Similarly, different formulas for T will lead to different methods. In this chapter, we will try to derive different formulas that can generate sequences that are further seen to be converging to an isolated root of our equation.

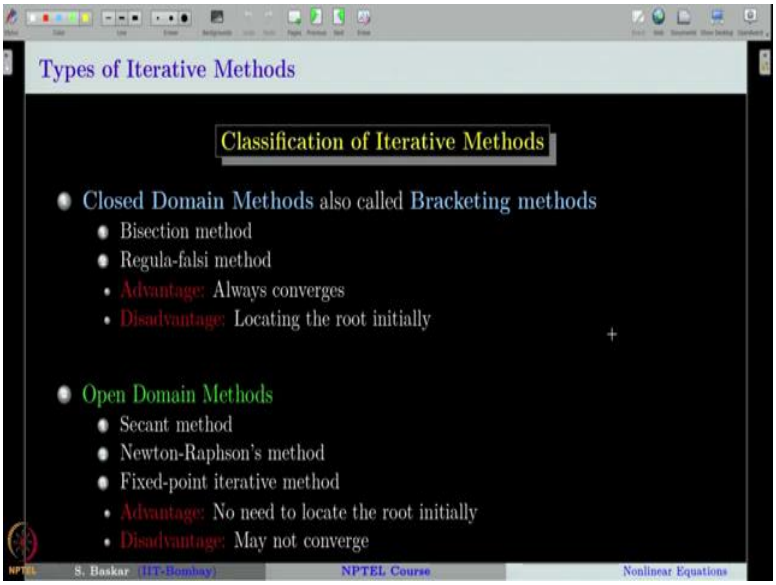
(Refer Slide Time: 12:33)



Well, therefore, what is the outcome of an iterative process? The outcome of an iterative process in general will be a sequence of numbers. And then the question is, will this sequence converge? And if so, what can you tell about the limit of this sequence? That is can we say that the limit of this sequence is an isolated root of our equation. These are the questions that we have to answer.

So, our task is to develop methods and then answer the convergence questions. So that is our interest in this chapter. That is, we have to devise the procedures, iterative procedures and then do the convergence analysis for these procedures. Generally, the convergence criteria depend on mainly two facts. One is, of course, the function that defines our equation $f(x) = 0$. And also, the convergence criteria depend on the domain in which we are interested in searching for a root and also it is co-domain.

(Refer Slide Time: 13:49)



Well, let us classify the iterative methods that we are going to learn in this chapter. The first one is called the Bracketing methods. Sometimes they are also called closed domain methods. You should not confuse the word closed domain with closed interval or something. It is just a name, it is better to use this word Bracketing methods. What this Bracketing methods does? Let us see.

In Bracketing methods, we will learn two methods, one is the Bisection method and another one is the Regula-Falsi method. The advantages of the Bracketing methods are that they always converge that the sequence generated by these methods will always be a convergent sequence. That is the advantage of these methods. But the disadvantage of these methods is that to start this method, you have to locate a root of this equation.

What is mean by that? As an input of this method, you have to give an interval say $[a_0, b_0]$ such that there exist an $r \in [a_0, b_0]$ such that $f(r) = 0$. That is, you have to give an interval in which at least there is one root of your equation. This is generally very difficult for you to find you can only find it by trial and error or plot the graph of the function. And see where it intersects the *x*-axis and then give an interval around that.

These are something which is not very easy, especially you cannot automize this idea in such a way that a computer can capture the root and give an interval around that. So, this mostly has to be done manually and also it is done mostly by trial and error. In that way, the Bracketing methods are not very commonly used because of this disadvantage. And therefore, it is always preferred to go for unbracketing methods or open domain methods.

In open domain methods we will study three types of methods. One is the Secant method another one is the well-known Newton-Raphson's method and Fixed-point iteration method. Again, the advantage of open domain methods is that you do not need to locate the root of the equation in order to start the procedure. You can just start with any initial guess chosen arbitrarily just like what we did with the Jacobi method and the Gauss Seidel method.

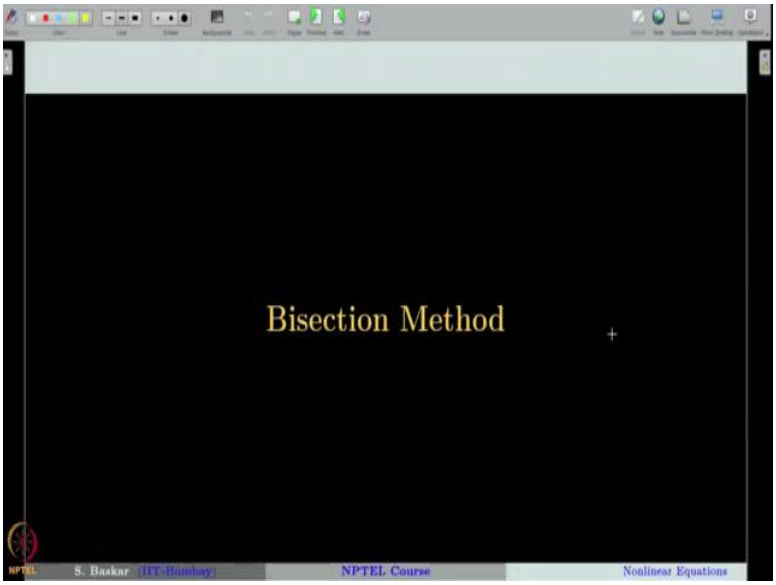
In those methods, we never chosen the initial guess with any property. They come as arbitrary vectors. Similarly, in all these three methods, we do not need to choose the initial guess with any properties imposed on them. You can just start with arbitrary initial guesses but the disadvantage is that the sequence that they generate may or may not converge. That is the problem.

Therefore, there is always an advantage and a disadvantage in both these methods. One needs to choose the methods according to their needs. Suppose, in certain problems you may know an approximate location of your route or in certain problems you want to capture that root with certain properties. Say, for instance, you may be given an equation $f(x) =$ and you want to find the smallest positive root of the equation.

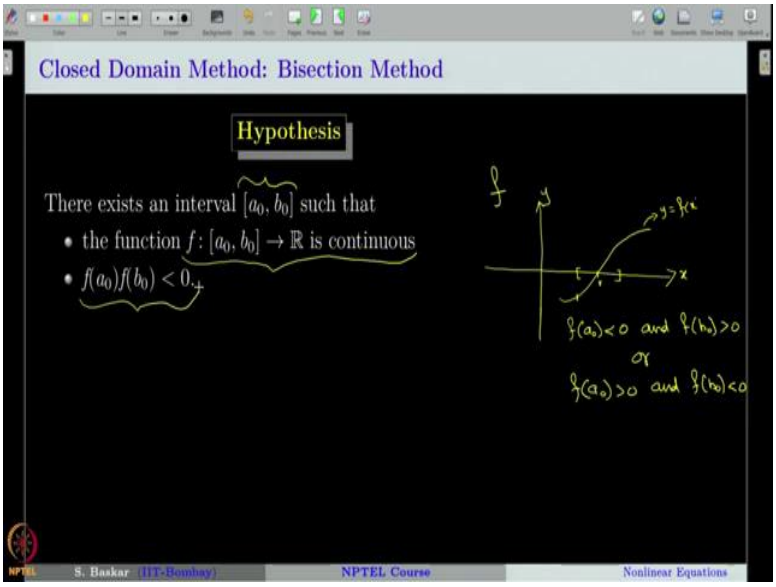
Suppose that is given to you, you may choose the initial guess as 0 and some very large number. And then once you choose that then your bracketing methods will surely converge that is what the advantages they may converge slowly but certainly they will converge. Whereas in the open domain method that confidence is not there, you may land up getting a diverging or oscillating sequence.

So, these are the disadvantages and advantages of these methods. While learning, we will come to know more about these methods, their advantages and disadvantages. So that we can efficiently use them in our problems as per our need. With this small introduction to a general setup of the iterative methods for nonlinear equations, now, let us go on with some specific methods as I told we will be introducing two methods in the bracketing type of methods and three methods, we will discuss in the open domain methods.

(Refer Slide Time: 19:15)



(Refer Slide Time: 19:17)



Let us start our discussion with Bisection methods. Bisection method is a very interesting and geometrically intuitive method. We always prefer to introduce the concepts with something which is geometrically intuitive. That is why we took Bisection method as our first method. As I told you have to first locate an interval in which there is at least one root of the equation $f(x) = 0$.

So, your input to this method is, of course, the function f that you have to give. And then you should also give an interval $[a_0, b_0]$ such that there is at least one root in this interval. Well, I will not always say isolated root because we will always work with those equation which has isolated roots. So, by the word root, I always mean isolated root in this chapter.

So, how to get this interval, is the question? Suppose the graph of the function is given like this. This is your $y = f(x)$ graph and now, I want to find an interval that has this root say *r*. Now, how will you find? If your function *f* is a continuous function again, if you recall, we have put this condition at the beginning of this lecture itself that we will always assume *f* to be continuous.

In fact, we will assume little more also. We will assume that if f' exists and even if f' is continuous that is what we will assume. But to define the problem and define the method you just need to have *f* to be a continuous function, assume that *f* is defined in the interval $[a_0, b_0]$. And how we capture that $[a_0, b_0]$? That they are found in such a way that $f(a_0) < 0$ and $f(b_0) > 0$.

Or you can also have the other way around that is $f(a_0) > 0$ and $f(b_0) < 0$. Any of these two conditions should hold. This is equivalent to saying that the product of $f(a_0)f(b_0) < 0$. Therefore, we have to search for two real numbers a_0 and b_0 such that $f(a_0)f(b_0) < 0$.

Generally, there is no automatic way of finding a_0 and b_0, especially on a computer. One has to do with trial and error either manually or even on a computer you may do it with different trial and errors. So that makes the algorithm much more costly if you go for coding this idea or one has to manually find it and feed it to your code. So that is the main disadvantage of the Bisection method.

Also note that these two conditions are the sufficient conditions for the convergence of the Bisection method. In fact, these are the sufficient conditions for the Regula-Falsi method also that is to say that under these two conditions, these two methods always converge. Well, Bisection method is a very nice intuitively clear and also geometrically interesting method. And we will discuss this method in our next lecture. Thank you for your attention.