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

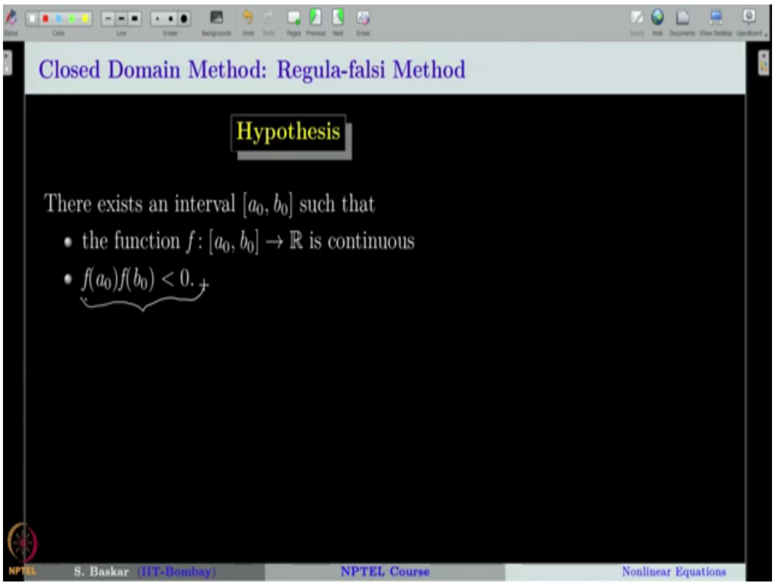
**Lecture – 30**
**Nonlinear Equations: Regula-Falsi and Secant Method**

Hi everybody, we are discussing Numerical Methods for Non-Linear Equations. In this we are discussing bracketing methods. Under this category we are interested in introducing two methods, one is Bisection method and another one is the Regula-Falsi method. In the last class, we have introduced Bisection method. We also did convergence analysis for bisection method and also we quickly gone through how to code Bisection method.

Today, we will start our second method on bracketing methods type that is the Regula-Falsi method. And then we will also pass on to the non-bracketing methods, in that we will discuss the first method called Secant method. Secant method is a slight modification of the Regula-Falsi method.

**(Refer Slide Time: 01:21)**


**(Refer Slide Time: 01:24)**

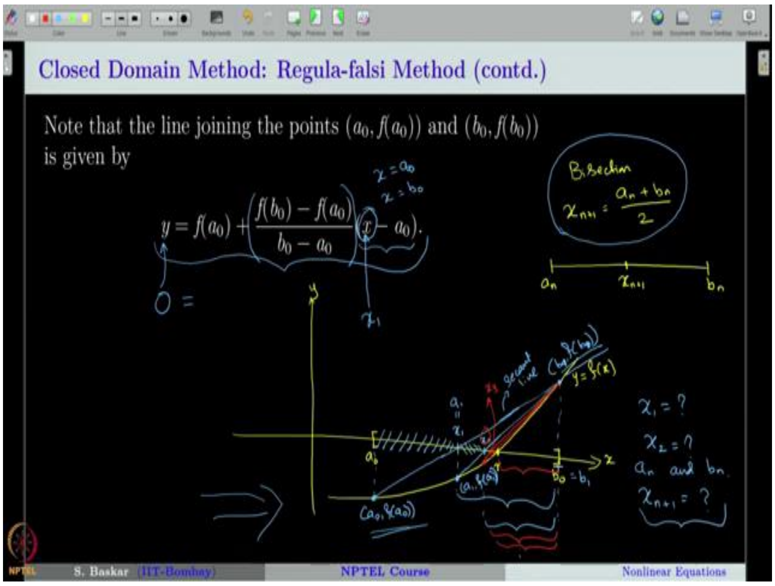



Let us start our discussion on Regula-Falsi method. Regula-Falsi method is a bracketing method. Therefore, if you recall, a bracketing method needs initially an interval to be specified where there is at least one root line. So, this is what we have also seen in the bisection method. Therefore, we need the following two hypothesis for Regula-Falsi method in order to start the method.

These two hypotheses, if you recall, are the same as we had in the bisection method. In fact, Regula-Falsi and bisection method, the underlying algorithm is the same. Only difference between these two methods is in the way how we split an interval in each iteration. Therefore, the hypothesis for Regula-Falsi method goes like this. We have to find an interval $[a_0, b_0]$ such that the function $f$ is defined on the interval $[a_0, b_0]$ and it should be continuous.

What is that function? The function i is actually, defining our equation $f(x) = 0$. And the second condition is that $f(a_0)f(b_0) < 0$. Since $f$ is continuous this condition will ensure that there exists at least one root in the interval $[a_0, b_0]$. We have seen that this is a simple application of the intermediate value theorem.

**(Refer Slide Time: 03:10)**




Once you set your hypothesis then our interest is to see how to split the interval $[a_0, b_0]$ to get an interval $[a_1, b_1]$ whose length is smaller than the interval $[a_0, b_0]$ length. So that is the only way that the Regula-Falsi method will differ from the bisection method. If you recall, in the bisection method, we define $x_{n+1} = \frac{a_n+b_n}{2}$.

That is, you obtained an interval at the *n*th iteration given by $a_n$ and $b_n$ and then in order to get the interval for the *n* + 1 iteration, you have to bisect the interval $[a_n, b_n]$ that is take the midpoint of this interval and define that as $x_{n+1}$ and that is the point where you are breaking the interval $[a_n, b_n]$. Now, the question is, in the Regula-Falsi method how are we breaking the interval $[a_n, b_n]$?

That is, you have to assume that you have obtained an interval $[a_n, b_n]$ at the n*th* iteration. And now, in order to go to the *n* + 1 iteration the basic idea is to split this interval $[a_n, b_n]$ into two parts. Well, to understand the method geometrically, let us consider a function $f(x)$ whose graph is given like this and we are interested in capturing the root or here and for that assume that we have located this root in the interval $[a_0, b_0]$ given like this.

Now, previously what we do is we just bisect this interval but now what we are going to do is you take two points, $(a_0, f(a_0))$ and $(b_0, f(b_0))$ and then draw the secant line joining these two points, that is the idea. Now, take the point $x_1$ as the point of intersection of this secant line with the *x*-axis. So, this is the secant line. Once you locate your $x_1$ then getting the interval $[a_1, b_1]$ is something similar to what we did in the bisection method.

That is you have to see where the root lies, whether the root lies in the first part of the interval or at the second part of the interval. In this particular example, you can see that the root lies in the second part of the interval that is, it lies between $x_1$ and $b_0$. Therefore, we will discard the interval $[a_0, x_1]$ and we will take $a_1$ as $x_1$ and $b_1$ as $b_0$. And now, what we will do? Again, you take this point which is $\big(a_1, f(a_1)\big)$.

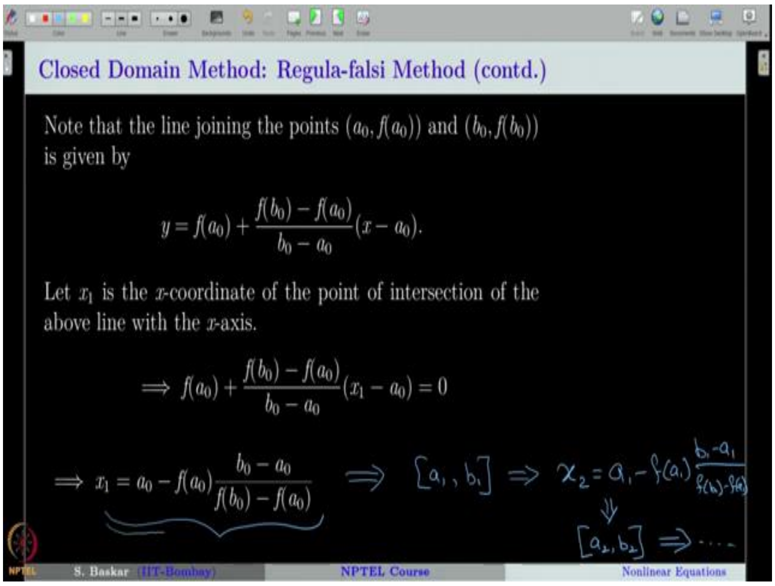
And similarly, the same point $\big(b_1, f(b_1)\big)$ and then again draw the secant line joining these two points and take the point of intersection of this secant line with the *x*-axis as $x_2$. And again, you apply the same idea of locating the root. You will see that the root in this example lies in this interval and it goes on like that. You will discard this part of the interval and take the second part of the interval in this particular example and the iteration sequence goes on like this.

Now, the question is how to derive $x_1, x_2$ and so on. That is how to derive the expression or formula for $x_1$and $x_2$ and so on. In general, how to derive the formula for $x_{n+1}$ given $a_n$ and $b_n$? That is the question. In bisection method it is just a direct idea you just take the midpoint. Therefore, we know how that expression looks like but here we have to get the expression for $x_{n+1}$.

Well, that is not very difficult because we know how to write the equation of the line joining two points $\big(a_0, f(a_0)\big)$ and $\big(b_0, f(b_0)\big)$ is given like this, so this is called this secant line. And the slope of this line is given like this. So, we know this formula from our basic geometry course and therefore, it is not very difficult. In fact, you can directly see by putting $x = a_0$ you will get $y = f(a_0)$.

And putting $x = b_0$ you will get $y = f(b_0)$. Now, what we want? We want to get this $x_1$ in such a way that $y = 0$. So, *y* should be 0 at the point $x = x_1$.

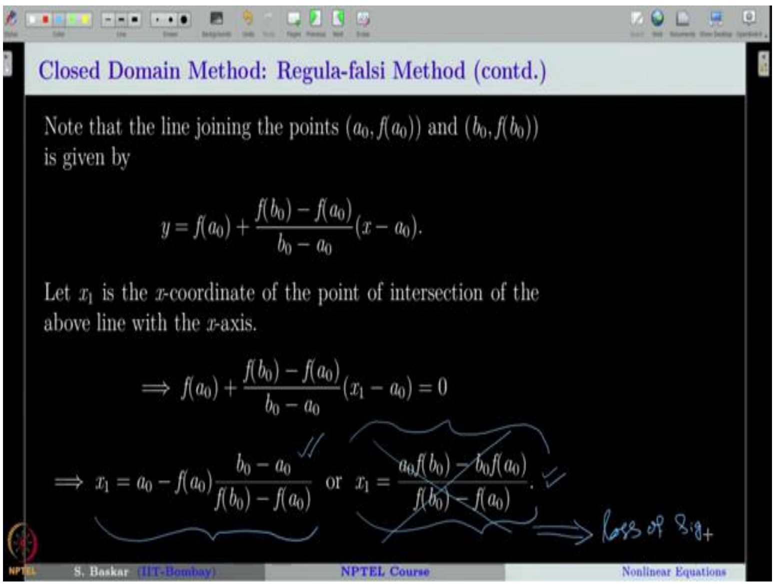
**(Refer Slide Time: 09:42)**




So, what you do? You put $y = 0$ and $x = x_1$ in this expression and that gives you $y = 0$ and $x = x_1$ here. From here you can get what is $x_1$ and $x_1$ is given like this. So, therefore, we obtained the expression for $x_1$ and now, how to get the interval $[a_1, b_1]$? That is the question well, we have already explained it, also we know it from our bisection method itself. Once you get $a_1$ and $b_1$ again, you calculate $x_2$ which is given by $a_1 - f(a_1)\frac{b_1 - a_1}{f(b_1) - f(a_1)}$.

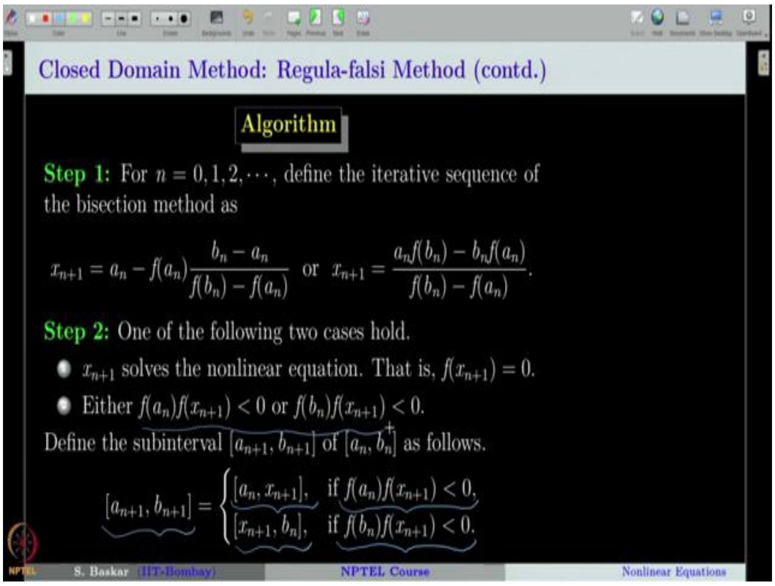
Once you get $x_2$ from there, you will get the interval $[a_2, b_2]$ and like that you will go on with your iteration. That is the idea of the Regula-Falsi method. The only new thing that comes is how you will define the point at which you have to split your interval at every iteration. That is the only new thing that you have to understand and that has a very nice geometric interpretation like this.

**(Refer Slide Time: 11:20)**



I hope you have understood the Regula-Falsi method, note that this expression can also be written like this. This is just a mathematical simplification. You can also write $x_1$ in this form but never use this form of the expression. You always use this form of the expression in your code. Why it is, I will leave it to you to see why we do not prefer this form of the expression, although both these are mathematically same. The reason is that this will may lead to loss of significance. I leave it to you to think why it is?

**(Refer Slide Time: 12: 07)**



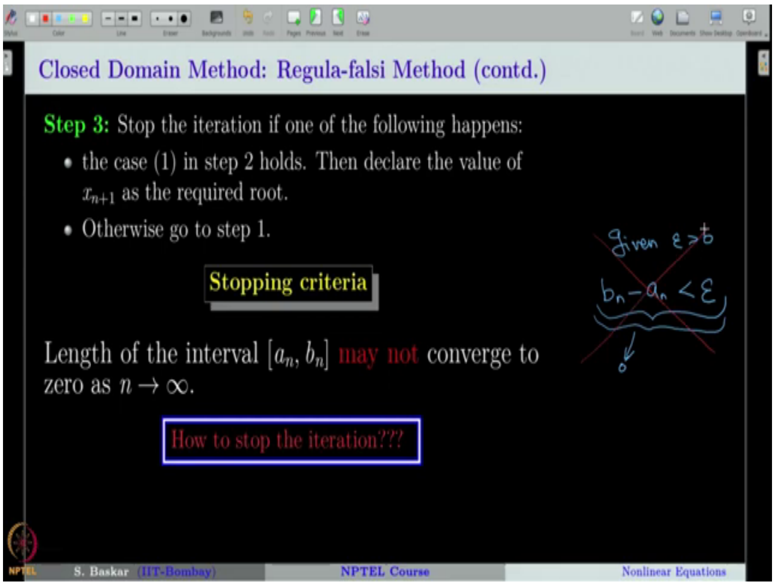

Let us formulate the algorithm now for $n = 0, 1, 2$ and so on. You should look at this expression by putting $n = 0$, first and then $n = 1$ and so on. When you put $n = 0$, you get $x_1$ and that is given in terms of $a_0$ and $b_0$ and $a_0$ and $b_0$ are given to us as inputs. Therefore, in that way you can compute $x_1$ once you get $x_1$ then you have to go ahead with the algorithm. How the algorithm goes?

From now onwards, it is exactly the same as what we did with the bisection method. First, you see whether the computed value $x_{n+1}$ coincides exactly with the root that is, check whether $f(x_{n+1}) = 0$. As I told you, this may not happen at all computationally because you always have a rounding error. Therefore, you may not always get exactly the root. But still, we may just check this condition just for formality.

And then, if this is not satisfied then you will go to check this condition. If this is also not satisfied then this will surely be satisfied. So, one of these two will surely be satisfied. If this is not satisfied and again, I will tell you this is more likely will not be satisfied. So, you will come across one of these two conditions. Only if the first condition is satisfied then you will take $(a_{n+1}, b_{n+1})$ as $(a_n, x_{n+1})$.

If this condition is satisfied. On the other hand, if this condition is satisfied then you will take our interval as $(x_{n+1}, b_n)$. And now, you are ready to go for the next iteration that is the *n* plus second iteration. Again, you will go to compute $x_n$ plus second equation. Again, you will come to check this condition and the iteration will keep on going.

**(Refer Slide Time: 14:26)**




Now, you have to worry about where to stop this iteration because on a computer we cannot go on doing this, somewhere we have to stop. What are all the situation where you can stop? Obviously, if $x_{n+1}$ coincides with the exact root then we are stopping our iteration. But this may not happen always. Therefore, one has to think about the stopping criteria. If you recall in the bisection method, we had a very natural stopping criteria.

That $|b_n - a_n| < \epsilon$ for some given $\epsilon$ which is the tolerance parameter. This is the stopping criteria we have imposed for bisection method. Why we did this? Because at every iteration you are necessarily breaking the interval into half. In that way, the length of the interval is actually shrinking to 0 as *n* tends to infinity. That was the nice part of the bisection method because of that, this condition will always hold at some *n*.

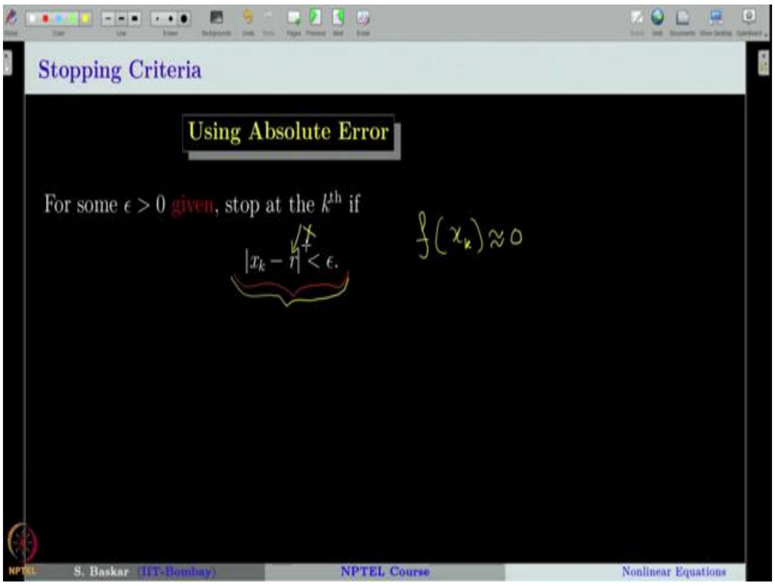
That is what you can see because it is actually this number is going to 0. Therefore, at some *n* it will surely cross whatever $\epsilon$ you give, however small it is. But in Regula-Falsi method, this may not happen. Why? Let us go back to our geometry and see, it is very clear from the way we have drawn the geometrical interpretation of the Regula-Falsi method at our first slide. You can see here that $x_2$ is given like this.

Now, $x_3$ will come the point of intersection of this secant line with the $x$-axis so, this is your $x_3$. And how $x_4$ is given? $x_4$ comes like this. Now, you can see that $x_3$ gives you a smaller interval like this, $x_4$ gives you a smaller interval like this. So, from here you can clearly see that the interval length is shrinking but, as $n$ tends to infinity, it will only tend to $r$. That is, $x_n$ will only tend to or at least geometrically you can see.

And therefore, the length of the interval will tend to the length of the interval $[r, b_0]$ in this particular example. So, this is a very good example where you can see that the interval length may not shrink to 0 in the Regula-Falsi method, unlike in the bisection method. Therefore, the natural criteria for stopping your iteration that we followed in the bisection method will not hold in the Regula-Falsi method.

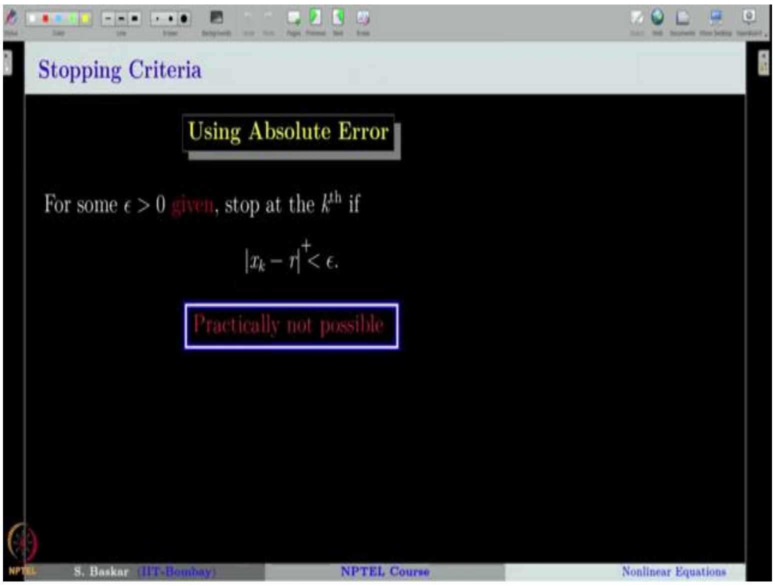
So, this will not hold in the Regula-Falsi method. Therefore, it is really a difficult question for us to now think how to stop the iteration.

**(Refer Slide Time: 17:44)**



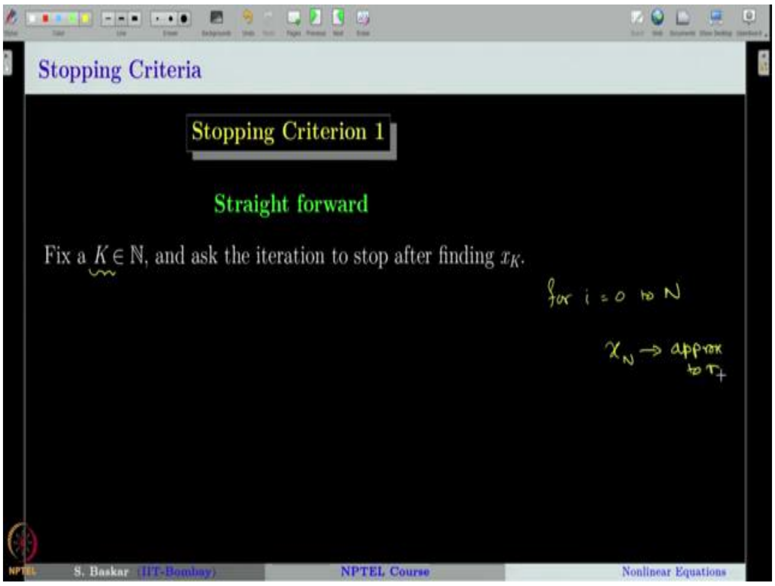

Well, one way to stop the iteration is to check this condition. This is the typical error that we would like to make it to 0. Of course, you have one more condition that $f(x_K)$ is also almost equal to 0 that we will come little later. But mostly we are interested in this but unfortunately you cannot check this condition. Why? Because we do not know $r$. The root of the equation that we are interested to capture is not known to us. In fact, that is why we are going for an approximation method.

**(Refer Slide Time: 18:34)**



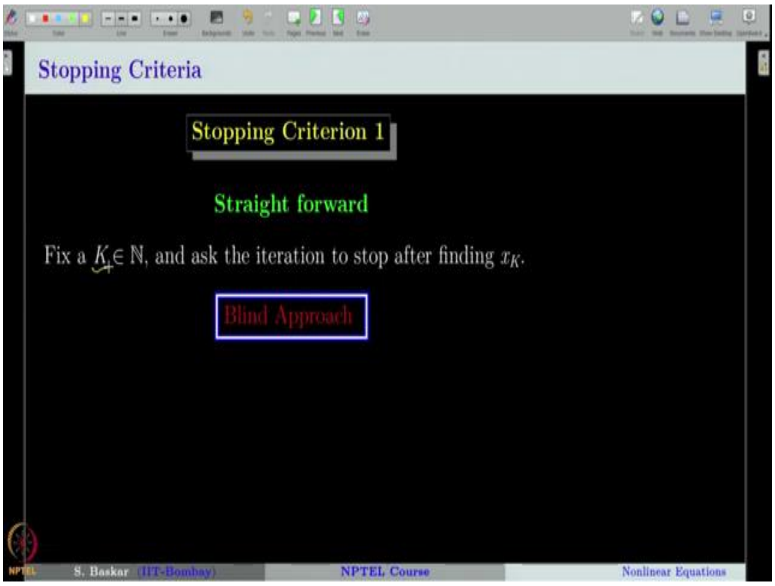

Therefore, this condition, although may be preferred but practically not possible.

**(Refer Slide Time: 18:42)**



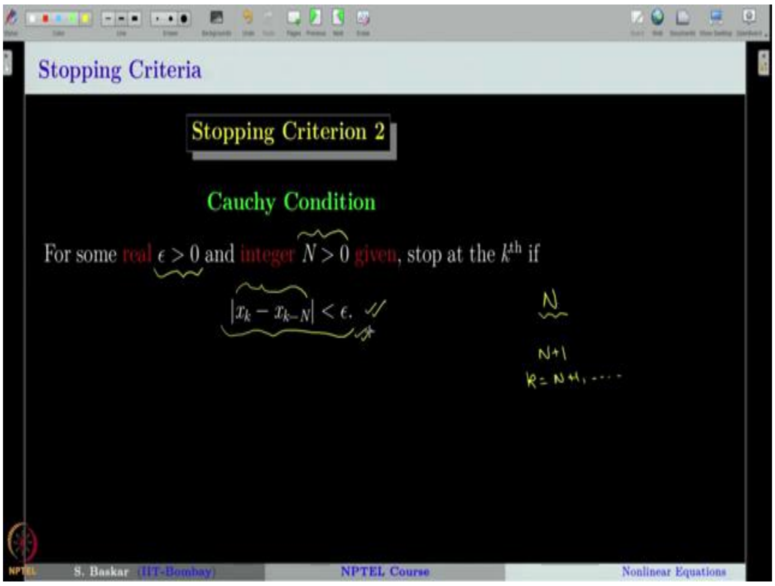

The next criteria is, to just decide some natural number K and run the loop, for *i* equal to say 0 to *N*. So, in that way you perform *N* iterations.

**(Refer Slide Time: 19:01)**




And then finally, declare whatever $x_N$ comes as the approximation to $r$ but this is rather blind. Why? Because, even if you have a large value of $K$ there is no guarantee that after performing that many iterations, you are pretty close to the root. Sometimes the convergence may happen very slowly and therefore, even after a large number of iterations, you may not be tending very close to the exact root. Therefore, this may be easy to implement but rather it is a blind approach.

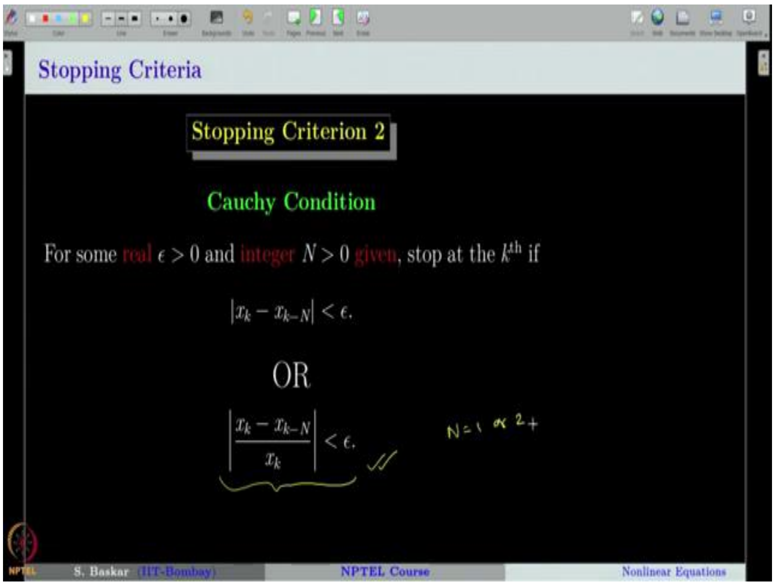
**(Refer Slide Time: 19:44)**




Another stopping criteria which we actually prefer is the Cauchy condition. So, what it says that for some real number $\epsilon$ that is your tolerance parameter. What you do is you first compute, the $N$ iterations, where $N$ is also decided by you beforehand and once you compute $N$ iterations from $(N + 1)$th iteration onwards, you check this condition that is $x_K$, where $K$ runs from $N + 1$ and so on. So, $|x_K - x_{K-N}| < \epsilon$.

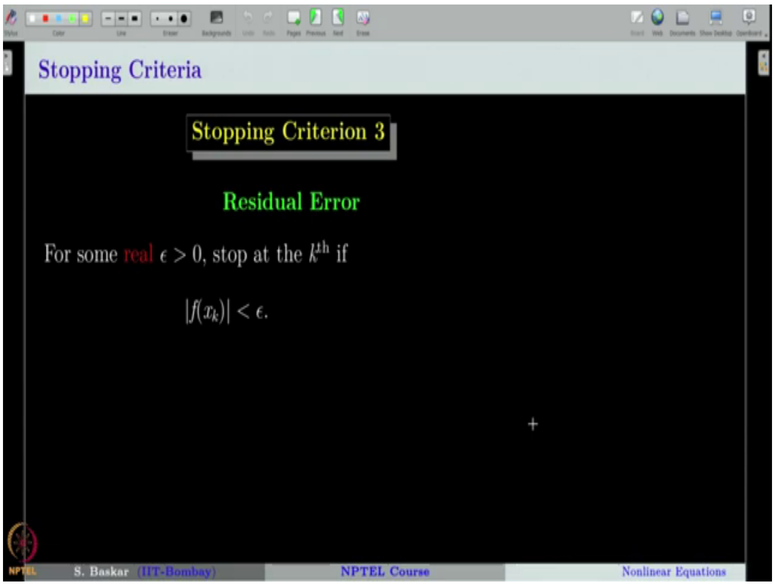
After running each iteration, you check this condition. If that condition is satisfied for some $K$ then you stop the iteration. You can see that all these quantities are known to us. Therefore, you can explicitly calculate this number and therefore, you can check this condition. So, this is a nice condition that can be implemented and we know that Cauchy sequence converges on a real line. Therefore, if you check this then you can in fact be confident that you are going very close to the root of your equation.

**(Refer Slide Time: 21:09)**



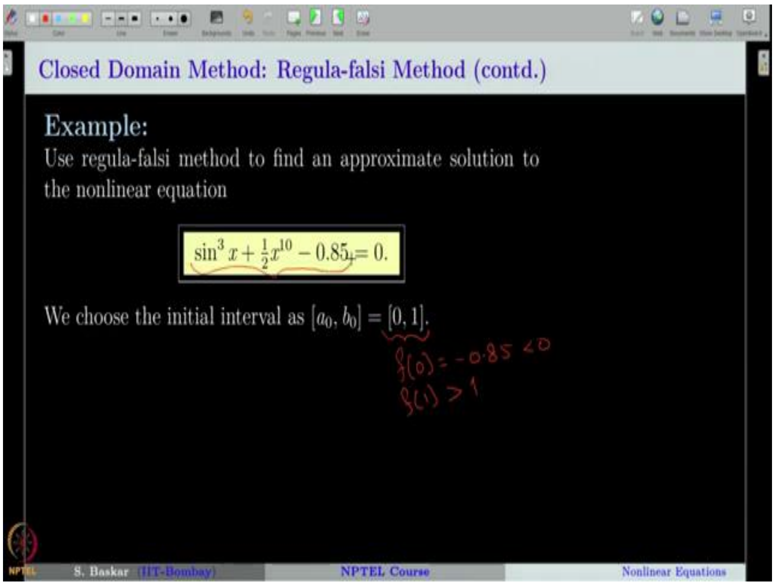

You can also impose the relative form of the Cauchy criteria. That is, you just take the relative error in $x_{K-N}$ when compared to $x_K$ and that will also give you a very reliable stopping criteria. So, we mostly prefer this criterion with possibly $N$ = 1 or 2. You do not need to go for a bigger $N$.

**(Refer Slide Time: 21:41)**




Another one is the well-known residual error we have already seen this while solving linear systems. This may also be a criterion for us to stop the iteration, as I told the residual error alone or Cauchy criteria alone, may not lead to a good idea of the approximation. In fact, you have to check both the conditions simultaneously but often we tend to use one of them. More often, we prefer to use the Cauchy criteria as this stopping criterion for our iterations.

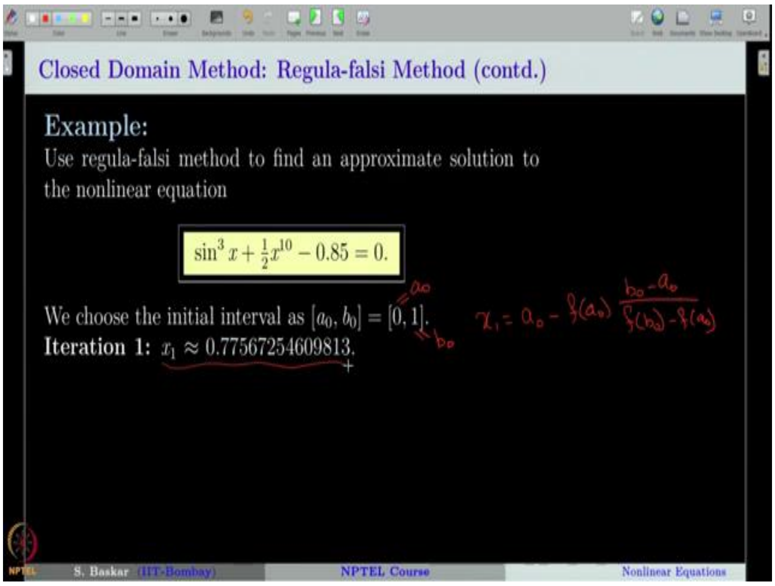
**(Refer Slide Time: 22:22)**




Let us take an example, use Regula-Falsi method to find an approximate solution to the non-linear equation, $sin^3 x + \frac{1}{2}x^{10} - 0.85 = 0$. Therefore, we are taking this as our function $f(x)$. You can see that the interval $[0,1]$ can be taken as the initial guess. You can take anything else but I am just taking here the interval $[0,1]$ as my initial guess, you can check that $f(0) = -0.85$ which is negative and $f(1)$ is also positive.

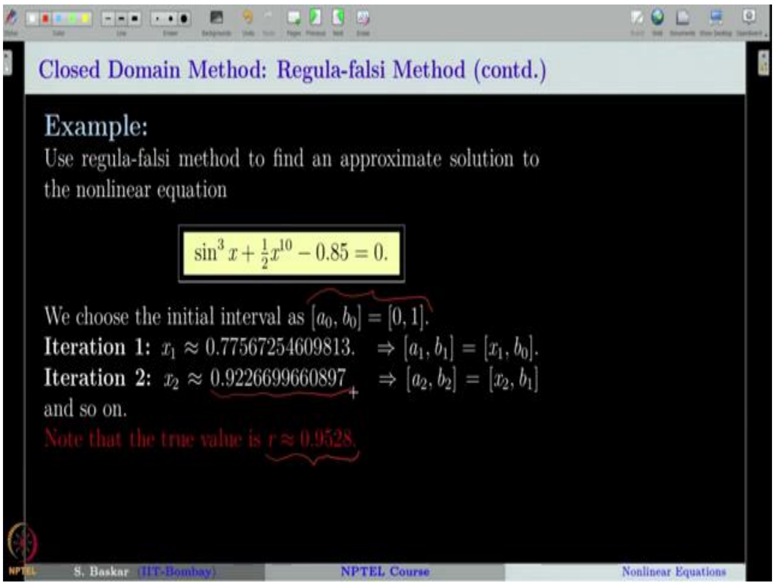
You can check that therefore [0, 1] is a nice interval that satisfies our hypothesis and also the function given here is a continuous function defined on the interval [0, 1].

**(Refer Slide Time: 23:24)**




Let us go to compute $x_1$ remember, $x_1$ is given by $a_0 - f(a_0)\frac{b_0 - a_0}{f(b_0) - f(a_0)}$ where $a_0$ is given by 0 and $b_0$ is given by 1.

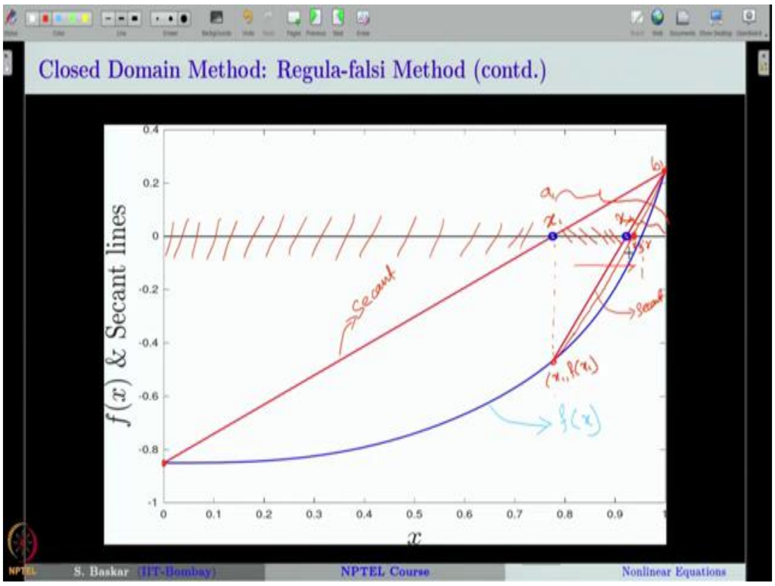
**(Refer Slide Time: 23:54)**




From here you can get $x_1$ and now, once you get $x_1$ you have to find the interval $[a_1, b_1]$ and that is given by $[x_1, b_0]$. I leave it to you to check this. It is not very difficult, in fact, we will see the geometric interpretation in the next slide and once you get $a_1, b_1$ then you plug in that into your formula for Regula-Falsi method to get $x_2$. And that gives you the interval for the next iteration that is the third iteration and that is taken to be $[x_2, b_1]$.

And you keep on going like this. Just for the information the root that we are trying to converge by taking this as the interval is given by $r$ = 0.9528 and so on. Therefore, we are trying to converge to this root which is also seen from these two iterations. In fact, if you go on doing the iteration, you will in fact see that the sequence is actually, converging to this root.

**(Refer Slide Time: 25:03)**



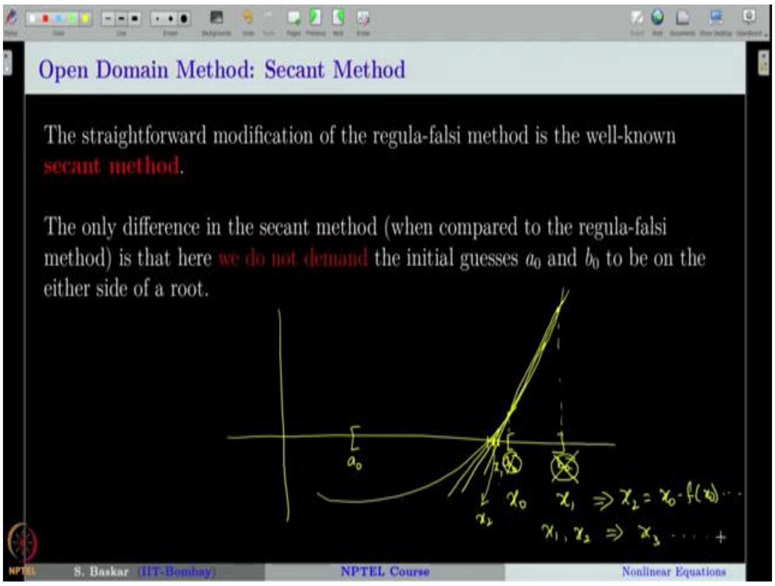

This is the geometric interpretation for the example that we have taken, the function graph is given like this. This is your $f(x)$ and the red lines are the secant lines and the first iteration is obtained by looking at the point of intersection of this line with the $x$-axis and that is given by $x_1$ here and once you have $x_1$, then you take $(x_1, f(x_1))$ as the next step you have to see in which interval they root lies in this case, the root lies in this interval.

Therefore, we are taking $a_1, b_1$ as this and then you go to draw the next secant line and that gives you $x_2$ and it goes on like this. Then the third iteration is done by joining this point and this point just assume that this is a straight line. Then you have $x_3$ somewhere here and every iteration. You have to see where the root lies. That is the idea of Regula-Falsi method. It is a bracketing method.

So, in the second iteration you will discard this interval and take this interval and it goes on like this. You can also see that in this example that the interval length is not shrinking to 0. Because you will always go from this side and at most you will go to the point $r$. Your iterations $x_N$ will never jump on the right-hand side of $r$ that is the problem. So, therefore, this is also an example where Regula-Falsi method iteration intervals may not shrink to 0 in their length.

With this, our discussion on Regula-Falsi method is over. Let us continue our discussion to the non-bracketing methods that is open domain methods. In open domain methods we will discuss the secant method as our first method. Secant method is almost the same as Regula-Falsi method. There is no new idea involved in secant method.

**(Refer Slide Time: 27:41)**




The only new idea is the way we implement the method that is all. It is otherwise a straightforward modification of the Regula-Falsi method. What is the idea in secant method? If you recall in Regula-Falsi method what we did is we have to locate the interval $[a_0, b_0]$ in such a way that there is at least one root lying in the interval $[a_0, b_0]$.

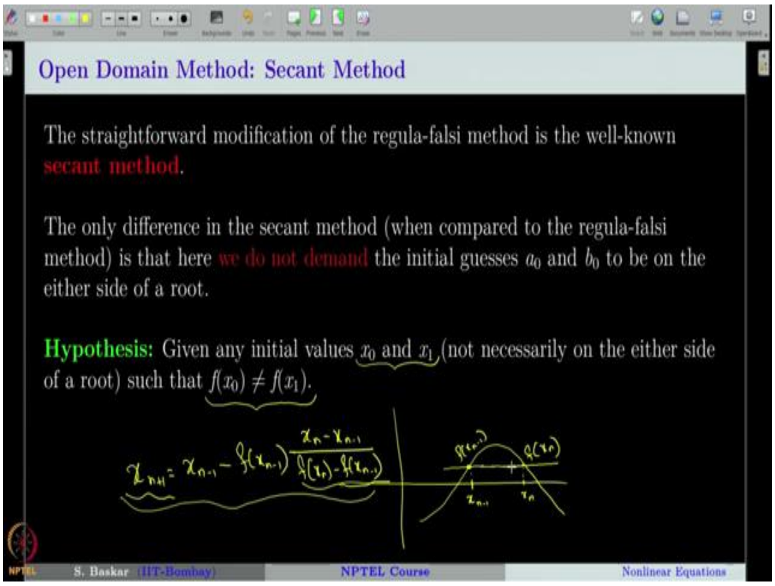
Now, this may not be always required because just take instead of $a_0, b_0$ on the either side of the root, let us take $[a_0, b_0]$ on only one side of the root and see what is happening. Then you can see that you have $a_0$ here and $b_0$ here and therefore, the point $x_1$ is obtained by joining these two points. And see where that secant line cuts the $x$-axis that is your $x_1$. Now, what you do? You, remove $a_0$ and take $x_1$ and $b_0$ and try to draw the secant line again.

You see the same convergence in fact happening in a very nice way here. So, this is your $x_2$ and now, what you do? You remove $b_0$, take $x_1$ and $x_2$. And again, draw the secant line between them and that will go like this. And in that way also we are forming a sequence that is converging to the root at least in this example. Therefore, the fact is that locating the root in an interval as far as the Regula-Falsi method is concerned is something which is not that important.

So, you can relax that because that is the main burden for us because we do not have a automated algorithm to locate a root. Either you have to do it by trial and error on a computer which may take long lot of time or you better do it manually. Even manually doing is some kind of trial and error only or you just have to draw the graph of the function. And see where it cuts the x axis and in that way only you can find an initial interval for your bracketing methods.

But that is not required as far as the Regula-Falsi idea is concerned. You just throw any orbital two points, say you throw $x_0$ and $x_1$ instead of $a_0$ and $b_0$. Let me use the notation $x_0$, $x_1$ and then you compute from here you can compute $x_2$ because it is given by $x_0 - f(x_0)$ into that formula you can write. And get $x_2$ once you get $x_2$, you keep $x_1$ and $x_2$ to get $x_3$. Using the same formula as we give in the Regula-Falsi method and you can keep on going.

**(Refer Slide Time: 31:01)**



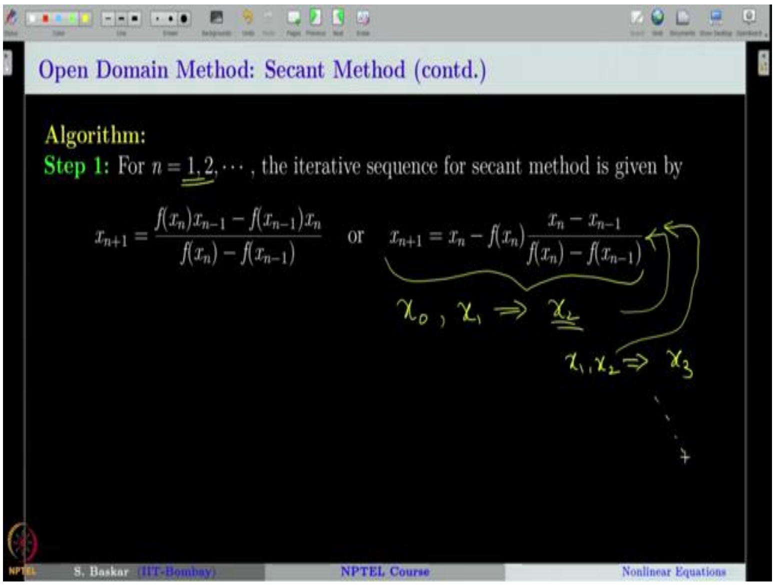

So that is the idea of secant method but one problem is suppose your function is something like this. Now, what happens if at some iteration $x_{n-1}$ and $x_n$ are such that $f(x_{n-1})$ and $f(x_n)$ are equal then what happens? If you recall $x_{n+1}$ is given by $x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$. If these two are equal then $x_{n+1}$ is not defined. That is what you can see from the expression.

Geometrically, what happens? If these two points are equal then the secant line will become parallel to the $x$-axis. Therefore, it will never intersect the $x$-axis. Therefore, the point of intersection will never exist. Therefore, at least to start with, you have to be careful in choosing your $x_0$ and $x_1$. You should choose these two points such that $f(x_0) \neq f(x_1)$.

However, such a situation may happen at the further iterations, in which case the secant method will fail. You can also check that in the Regula-Falsi method, such a situation will never occur. That is why, once you locate the root in an interval Regula-Falsi method, surely will take you to a root. That is, it will surely converge to a root, whereas in secant method that is not guaranteed to you.

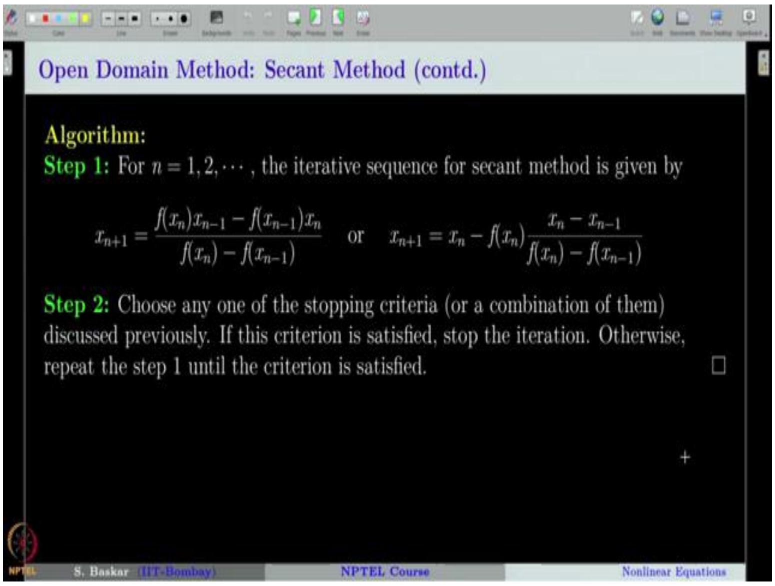
Because there is a danger that at some iteration that is at some $n$ you may face a problem where $f(x_n) = f(x_{n-1})$ in which case the secant method fails.

**(Refer Slide Time: 33:09)**



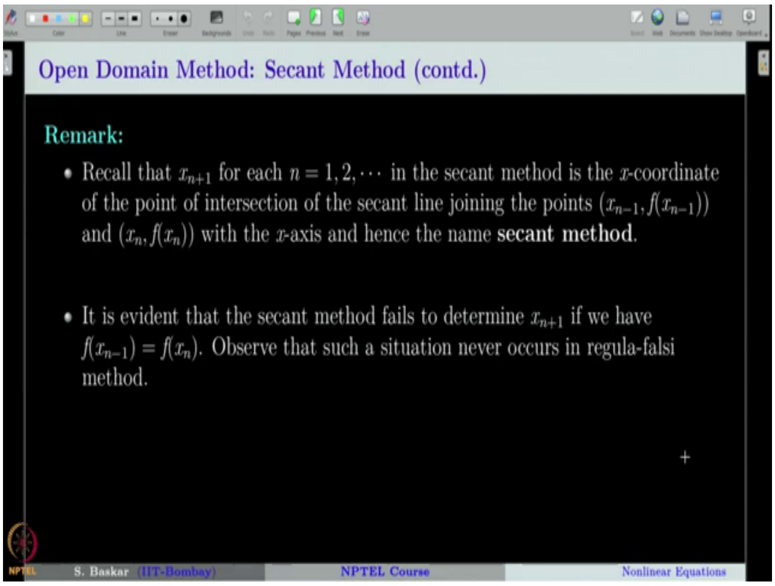

So, let us give the algorithm quickly. The algorithm is given like this. What you have to do? You have to give $x_0$ and $x_1$ and then compute $x_2$ that is take $n$ = 1 that gives you $x_2$. You have to plug in $x_0$ and $x_1$ in this expression get $x_2$. Once you get $x_2$ then give $x_1$ and $x_2$ to get $x_3$ again you have to plug in this into this formula and get $x_3$ and so on.

**(Refer Slide Time: 33:47)**



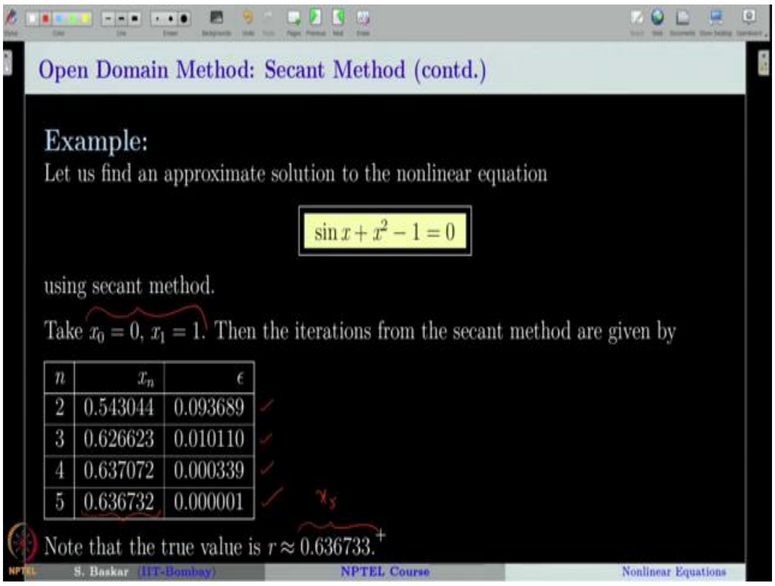

So that is the algorithm again stopping criteria one has to carefully choose, as I told you, Cauchy criteria is something which we can prefer.

**(Refer Slide Time: 33:56)**




As I told you, the secant method is obtained by looking at the point of intersection of the secant line with the $x$-coordinate. That is why the name of the method is secant method and one situation where the secant method fails is that at some $n$ if $f(x_n) = f(x_{n-1})$ then the secant method fails.

**(Refer Slide Time: 33:22)**




Let us take an example. Let us take $f(x) = sin\, x + x^2 - 1$. Then we can use the secant method for that. You have to give now two initial conditions. Remember you do not need to take $x_0$ and $x_1$ in such a way that there is a root in between them. You can just throw these numbers arbitrarily just to have a comparison. I have taken the same numbers 0 and 1 as we did with the Regula-Falsi method but remember this is not required.

You can take any numbers as the initial guess then I have just computed up to five iterations. And you can see that you obtained $x_5$ is pretty close to the exact solution which is approximately given like this. So, in this example, secant method is nicely converging to the exact root 0.6367 and so on. With this, we will finish our discussion on secant methods. Thank you for your attention.