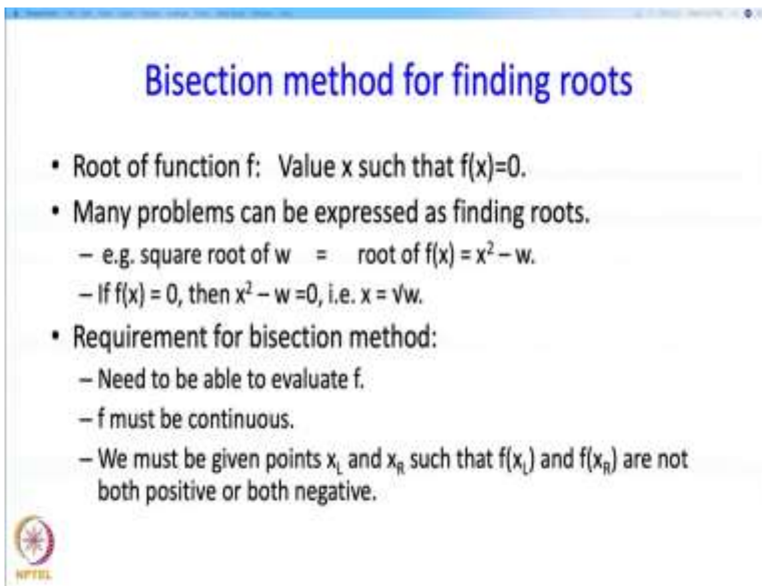
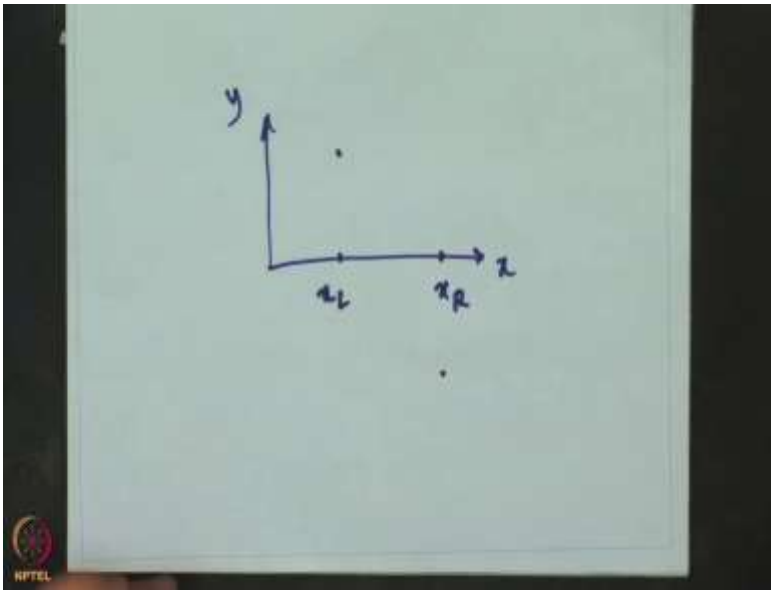
An Introduction to Programming through C++
Professor Abhiram G. Ranade
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Lecture No. 8 Part – 3
Computing Mathematical Functions
Bisection Method

Welcome back! In the previous segment, we discussed numerical integration. In this segment, we are going to discuss the bisection method for finding roots.

(Refer Slide Time: 0:31)





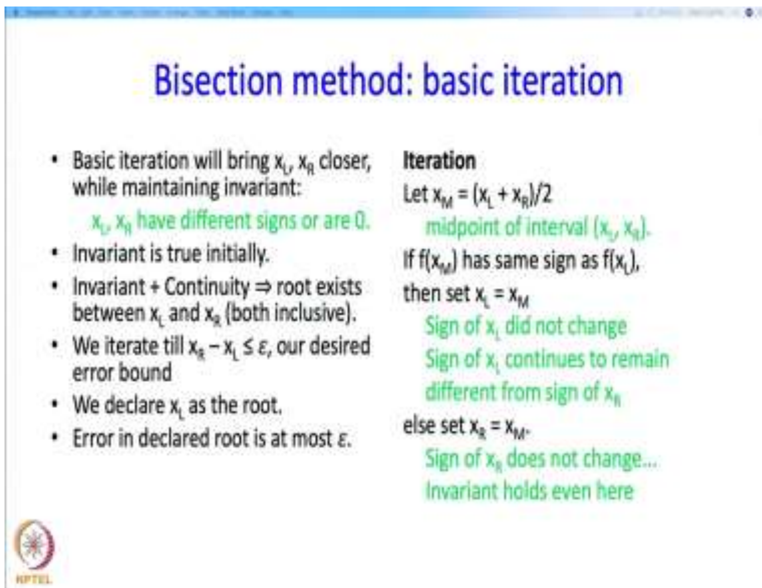
So, the root a function F is a value x such that F(x) equal to 0. Or in other words, it is the point, where F(x), the graph of F(x) crosses the x-axis. Now, many problems can be expressed as finding roots. For example, if I want the square root of some number W, then I claim that is the same thing as the root of the function F(x) equal to x^2-W. Why is that? Well, if we do find root, that is if we find F(x) such that, if we find x, such that F(x) equal to 0, then we know that x^2-W is equal to 0. Or, x must be square root of W.

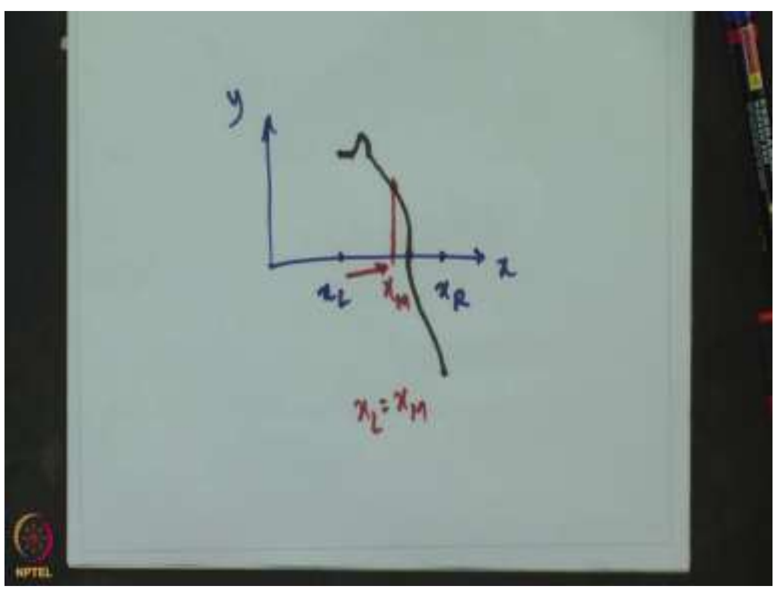
Now, the bisection method has relatively few requirements as to when it can be applicable. So for example, we require the ability to evaluate F, given an x. So, we should be able to calculate F(x) given an x. Then, we want F to be continuous. And then, we must be given points xl and xr, such that F(xl) and F(xr) are not both positive or not both negative. So these are three relatively simple requirements, but once we have these then we can use the bisection method. And the bisection method can be used for finding roots. But, as we just remarked, finding roots is really itself useful for finding, for calculating functions, such as say the square root in this case, okay. So, again to reiterate, we need to be evaluate F. F must be continuous. And, we must be given points xl and xr, such that F(xl) and F(xr) are not both positive or both negative.

Or in other words, what must be the case? So, if we plot the graph of X, graph of F. So, this is the X-axis. This is the Y-axis. And, let us say, this is the point, this is the point with X co-ordinate xl. This with xr. Then, the points F(x), the value of F at xl, say it is here, then xr should be on the other side. Or, it could be at 0. But, both of them cannot be on the same side, okay. So that is what the point is. That is what the requirement is.

So, I am going to describe the bisection method now. But, we will assume these three properties. So, in other words, we will assume that we are given xl and xr satisfying this property.

(Refer Slide Time: 3:36)





So the basic iteration of the method is as follows. So we will start with the user given xl and xr. And then, we will bring them closer. As we bring them closer, we will maintain the invariant, then, that xl and xr must have different signs or they are 0. If they are 0, it does not really matter. Now, I want to observe that at the beginning this invariant is true. That is because the user gave us two such, two such numbers. Now, before proceeding, let me discuss the implication of this. So, suppose F(xl) is indeed positive and F(xr) is negative. Now, because this function is continuous, what does it mean? The graph of the function must start somewhere over here. Maybe it will increase. Maybe it will do whatever it needs to, but it is continuous. So this graph, this line must be continuous. And being and while it is continuous, it must still reach this point. So, what does it mean? Then that means it must cross this X axis at some point. Otherwise it cannot go to the other side. But the point at which it crosses is exactly the root. So, what we know is that the root must be contained in this interval. So effectively when we said to the user please give us xl and xr. The user is effectively giving us an interval, which contains the root. So, invariant plus continuity implies that the root exists between xl and xr. And the root might be exactly at xl or exactly at xr. So, F(xl) or F(xr) are allowed to be 0. So, we are going to execute this basic iteration until the distance between xr and xl, becomes smaller than some error bound. Let us call it epsilon. If, epsilon is really small, then these two numbers are really close, and the root is somewhere in between them. Which really means that we can declare one of those numbers, say xl as the root. So, we will have error in this. But, the error is at most epsilon.

So, that is the whole, that is the whole; that is the gist of the argument, okay. So we are going to get; we are going to get an approximate answer. But you will see that the approximation can be as good as you want. No matter what epsilon you give us, we will be able to iterate until that point.

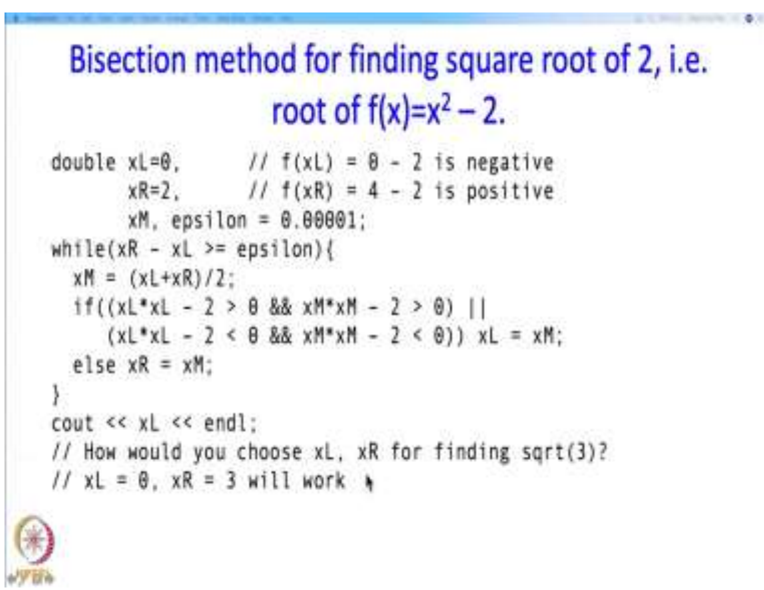
So, let me, let me tell you now, how this iteration is going to happen. How we are going to shrink this interval, okay. So, first we calculate xm, which is (xl+xr)/2. So, what is that? Well, that is the midpoint of this interval xl to xr. Next, we check the signs of xm and xl. If, these two signs are the same, okay, then we are going to set xl equal to xm. So again, let us come back to this picture. So, this is xl, xr. We find the mid point, which is over here. Then, we look at F(xl), okay. So the sign of F(xl) and the sign, the sign of F(xm). So this is xm. So the sign of F(xm) is positive. And the sign of F(xl) is positive.

So that means, F(xl) and F(xm) have the same sign. So in which case, we will execute xl equal to xm. Or in other words, we will move this point over to this point.

So, I claim now, that this is going to keep our invariant without violation. So, why is that? Well, xl and xm had the same signs. So the signs did not change as we moved xl to xm. As we assigned xl equal to xm, the signs did not change, okay. So that means the sign of xl continues to remain different from the sign of xr. So, if it was different before, then it will remain to continue to remain different. But, we are assuming that in, that this invariant was true in the beginning of the iteration. And therefore, what we have proved that in this case the invariant will hold even after the iteration. Or in other words, the root will now be contained in this smaller interval, which is true in this case, in our picture.

If, this condition is not holding that is if xm and xl do not have the same sign, then we are going to set xr equal to xm. So, why is that? Well, in that case xr must have the same sign as xm. And so again, we can conclude that the sign of xr has not changed. And so even in this case, we can conclude that the invariant is going to hold. So basically what we have is a simple method by which we can shrink this interval.

(Refer Slide Time: 9:37)

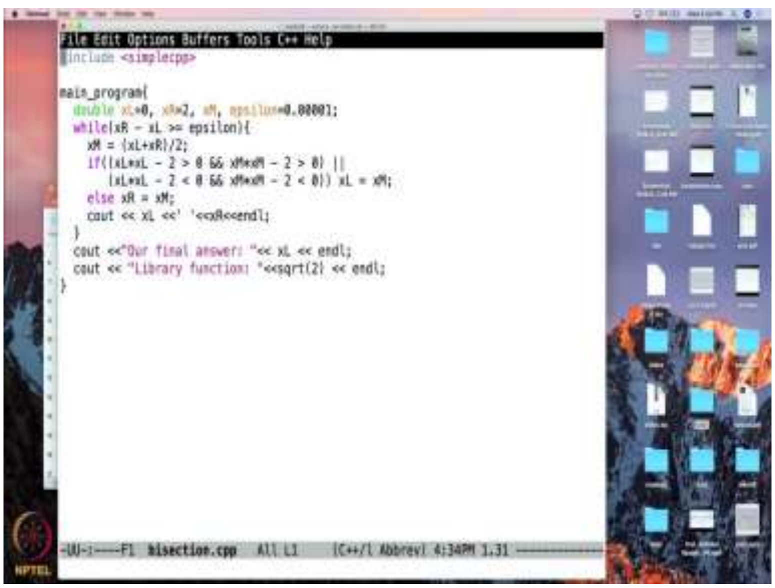


So, let us now see the code file. So, we are going to write this code for the problem that we mentioned earlier, which is finding the square root. And we will make the problem very concrete by saying that we want to find the square root of 2. So in that case, as we discussed earlier, we should be finding the root of F(x) equal to x^2-2. Because if x is root of this, then x^2-2 must be equal to 0 or x^2 must be equal to 2 or x must be square root of 2. So the root of this equation will indeed give us this square root of 2.

So first we need to supply xl and xr. So I claim that xl equal to 0 will work. So let us see what that means. So xl equal to 0 means, F(x) equal to 0 minus 2. So F(xl) is negative. So, so long as we can supply an xr, such that F(xr) is positive, we will have satisfied of requirement .So we are going to set xr equal to 2. So F(xr) is 2 square or 4 minus 2 and therefore, it is positive. So our xl and xr are really satisfying, what we set out that their signs, the signs of F(xl) and F(xr) are indeed different. So then we need to have a variable to store this xm value. And we need an epsilon as well. And let us say, just for, just as an example that we pick epsilon to be 10 to the power minus 5, 10 to the power minus 4. I am sorry, 10 to the power minus 5, okay. Anyway it does not matter. Now, we are going to just have to write our basic iteration. And, what is our basic iteration? Well, we check whether the interval xr to xl or the length of this interval which is xr-xl is bigger than epsilon. If, it is bigger than epsilon, then that means that our interval is still larger than what we want. So in that case, we should execute our basic step. So, what is our basic step? So, we are going to find xm, which is the midpoint of xl and xr. And, so it is xl+xr/2. Then, we are going to check whether xm and xl have the same signs, okay.

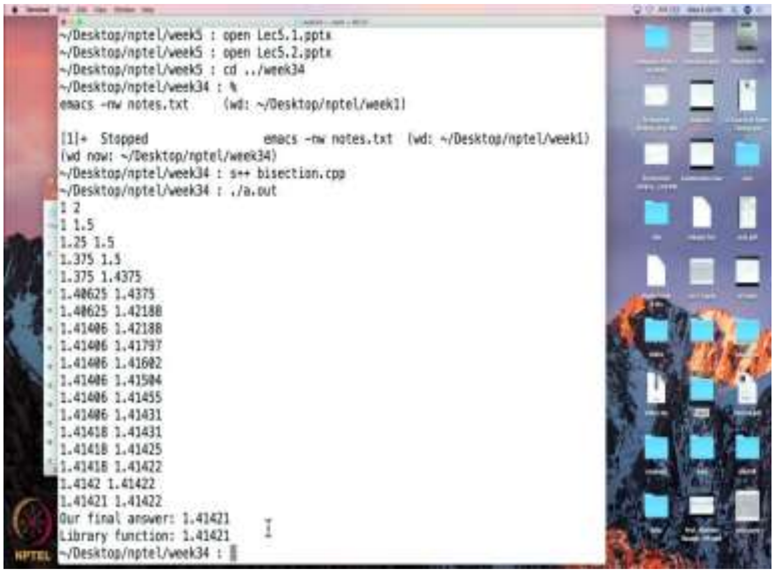
So, what is this checking? This is checking whether F(xl) is bigger than 0. So, does xl have the positive sign? And, this checks whether xl has, xm, F(xm) also have the positive sign. So, this is checking one side of the condition that we want, that F(xl) and F(xm), both have positive side, positive signs. So, either they both; so if they both have positive signs or if they both have negative signs, which is the second condition over here, then, that means they have the same signs. xl and F(xl) and F(xm) have the same sign. In which case, we are going to set xl equal to xm. Otherwise, we are going to set, xr equal to xm. This is exactly what we said on the previous slide. That is it. So we are going to repeat this. We are going to repeat while this condition holds. We are going to stop as soon as this condition stops holding or in other words, when xr-xl becomes smaller than epsilon. Or in other words, our interval have become smaller than epsilon. So at that point we are going to print out xl as the root, alright. So, this is, this is what we said on the last slide and this is what we have here. So, this should work, and we will yeah. So, before moving on, I just want to ask you a simple question. So suppose instead of finding the square of 2, square root of 2, I want the square root of some other number. So, say square root of 3. How, could you choose xl and xr? So, we can follow; we can see what is happening over here and maybe we will try to do something that is over here. So here we have 2. So let us try if we put, what happens if we put 3 over here. So indeed we will see that it works. Because, again this will be negative and this will be 9-3, so this will be positive, okay. So, you can use the same idea to solve this for any number larger than 1, for any number yeah. And, we are, we are, yeah. So, let me leave it at that.

(Refer Slide Time: 14:49)



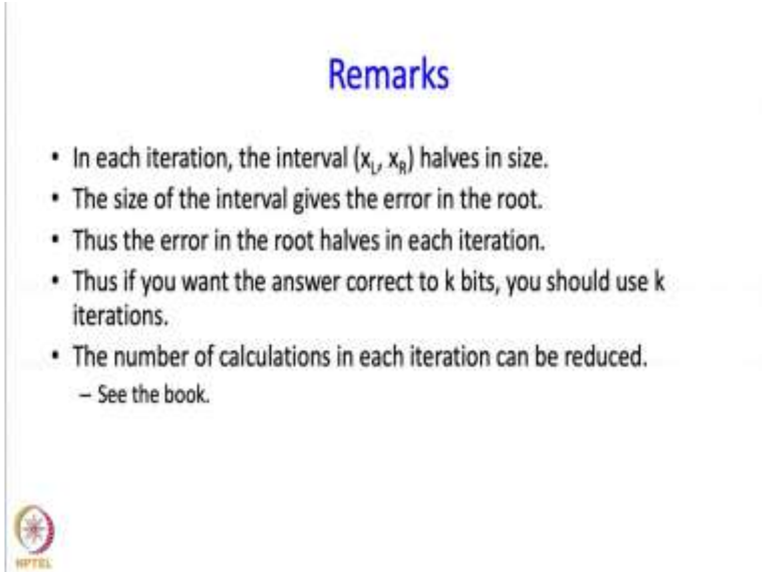
So, let us do a quick demo of this. So, this is the program that we had on the slides. There is a slight difference. So we are printing our final answer that we calculated over here. But, in addition, in each iteration, we are also printing out the current interval. So the current values of xl and xr. And we are doing this at the end of iteration. So at the end of the first iteration, we will calculate what xl and xr are and so on for each iteration. So we are calculating the intervals. We are calculating the final answer. And then we know that there already exists a library function sqrt which gives us the square root. So we will print that also. So that will tell us, how good our answer is.

(Refer Slide Time: 15:40)



So, let me compile this. Okay, so let us see, what it has printed. So at the end of the first iteration the value, the interval was 1 to 2, okay. Does not makes sense? Well, we stared off with 0 to 2, okay. xl was 0 and xr was 2. But then we moved xl to the midpoint, so it was 1 to 2. And so the interval keeps on shrinking. And eventually it has got down to this. So it take some amount of work, number of iterations. But it eventually did get to this. And as you can see this is indeed identical to what the library function tells us. Of course, there may be additional digits after this, which are not being printed over here. And maybe the final answer is different. So, if you want to check that, you can print the additional digits also. But anyway to 5 digits, this is certainly the same.

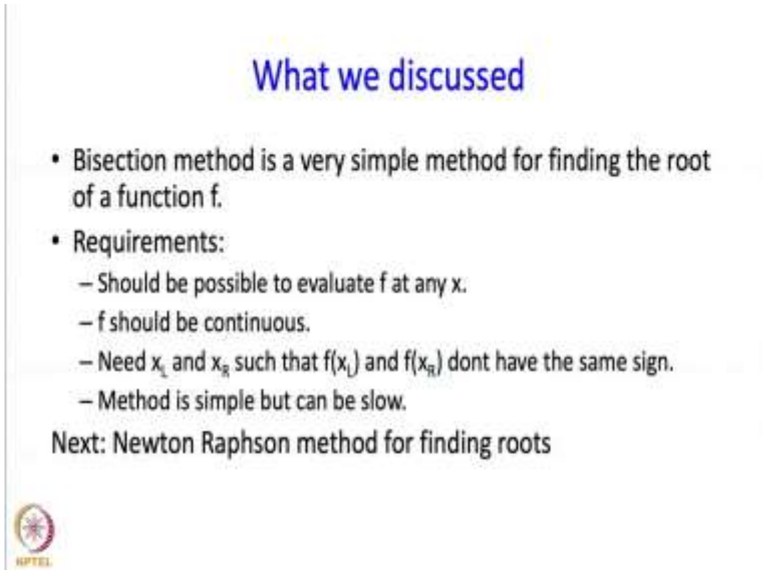
(Refer Slide Time: 16:55)



Okay, so let us get back to the slides. So a few remarks. So because we are picking the midpoint in each iteration, this interval xl to xr is halving. So in each iterations the interval halves. So the uncertainty in our, in our estimate goes down by the factor of 2. Or in other words, we get one more bit of our final answer. And the size of interval gives the error in the root, we said. And so the error halves. And so if you want to answer correct to K bits you should use K iterations.

Now, the code that I have given over here is doing a few additional calculations over and above what is strictly needed. So I do not want to get into that detail right now in this lecture. But, you can take a look at the book and see how some of the work that you are doing could be reduced. And let me leave you with an exercise. I would like you to modify the program, so that it calculates the cube root of any number W. So the number W should be read in from the keyboard. And your program should correctly initialize xl and xr now. So think about how you would do, how you would have to do that. And then we should print out the cube root this time, not the square root.

(Refer Slide Time: 18:27)



Okay, so what did we discuss? So we discussed the bisection method, which is a very simple method for finding the roots of a function. So it requires that we should be able to evaluate F for any x and that F should be continuous. And we need to have xl and xr, such that F(xl) and F(xr) do not have the same sign. So if these three requirements are met, then we can use the bisection method. And I should point out that the method is simple but it is slow, in the sense that there are faster methods available. And next we will see the Newton-Raphson method, which is one such method. But before that we will take a break.