Numerical Methods and Computation

Prof. S.R.K. Iyengar

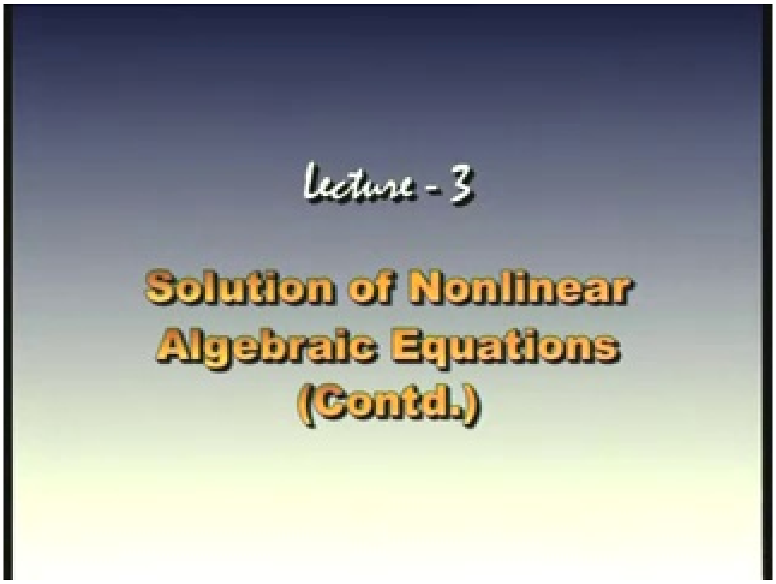
Department of Mathematics

Indian Institute of Technology, Delhi

Lecture No # 3

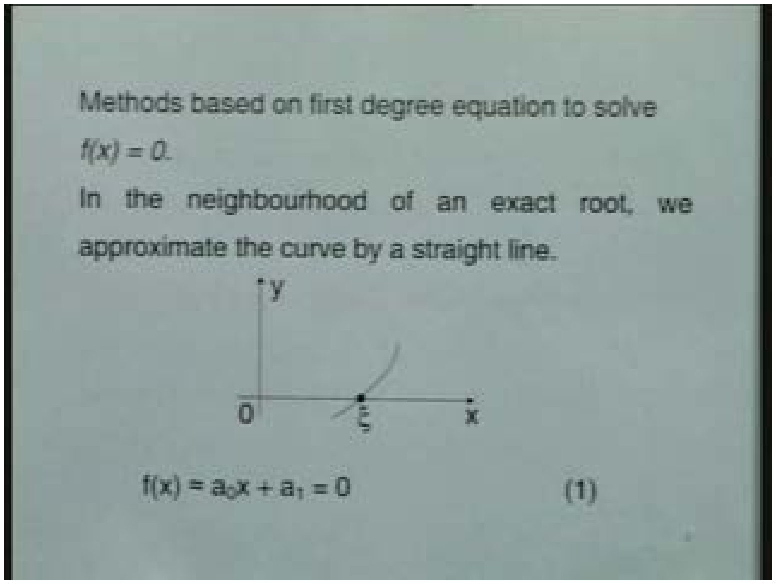
Solution of Nonlinear Algebraic Equations (Continued)

(Refer Slide Time: 00:00:56 min)



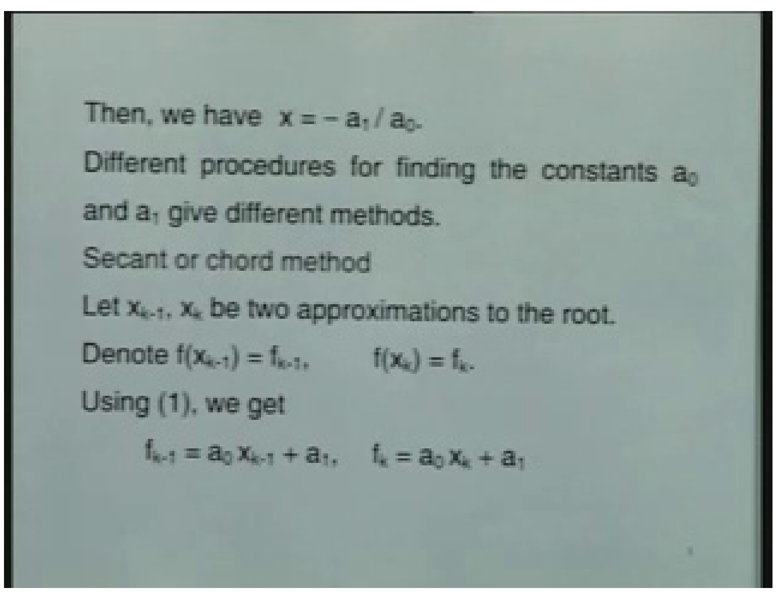
Let us briefly review what we have done last time in the last lecture. We were trying to derive a method for finding the root of an equation fx is equal to zero. fx is an algebraic equation.

(Refer Slide Time: 00:01:18 min)



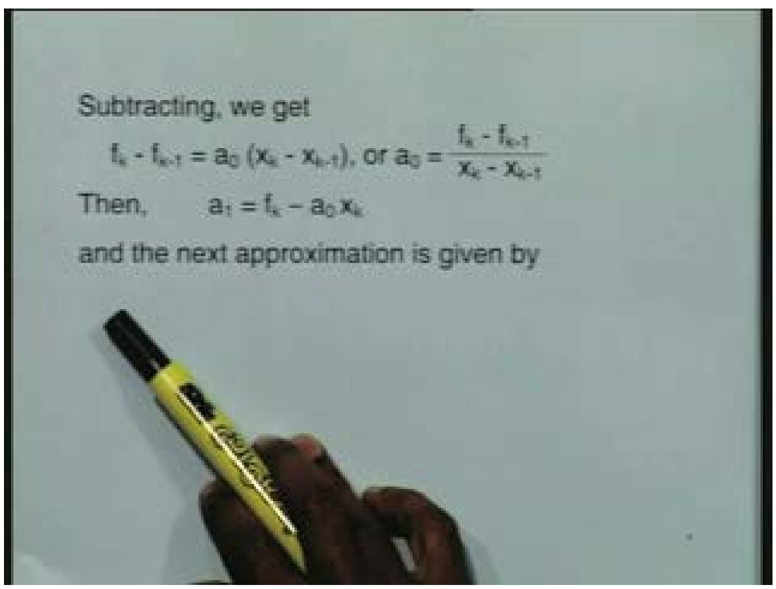
So it is a transcendental or a polynomial equation. What we discussed last time was that, in the neighborhood of the exact root we shall approximate the curve by straight line. So it is a valid justification because in the neighborhood of this exact root the curve can be approximated by straight line. This straight line is fx is equal to a_0x plus a_1 is equal to zero and we said that the different methods can be derived by trying to derive the values of a_0 and a_0 in a different way.

(Refer Slide Time: 00:01:52 min)



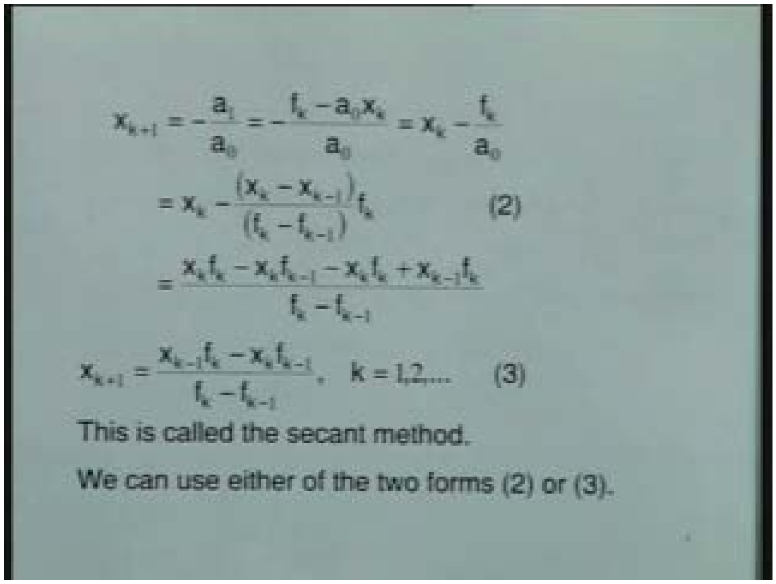
Now if I solve this particular equation what I would get is, x is simply equal to minus a one upon a_0. Now the first method that we were derived last time was the secant or the chord method. We have taken two approximations, xk minus one xk to the root. There are any two approximations, as we said it is not necessary that the root should lie between these two approximations. So these are two arbitrary approximations near the root. Then we take the point on the curve, so we have f of x minus k minus one is fk minus one; f of xk is fk. Now this point should lie on the given straight line therefore it should satisfy these two equations that we have derived. Now there are two equations for two unknowns' a_0 and a_1, so I solve these two equations for a_0 and a_1 one and we have then produced the values of a_0 and a_1 one as follows.

(Refer Slide Time: 00:02:58 min)



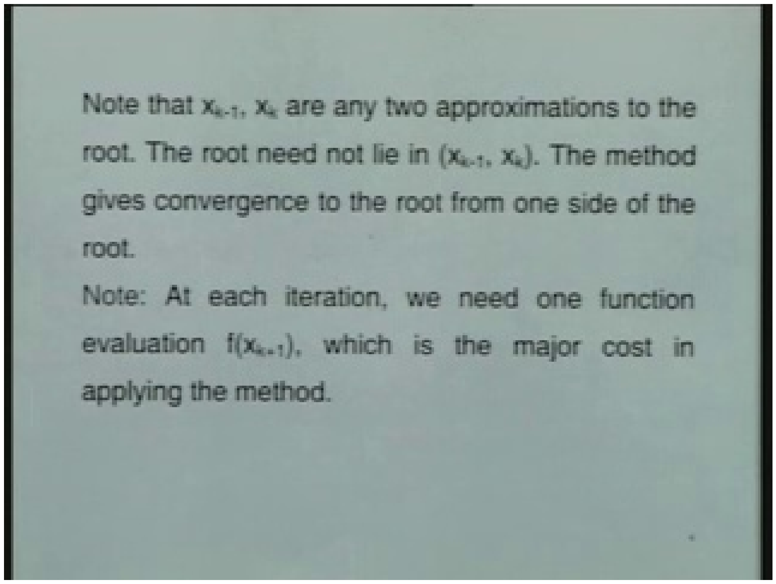
We just subtracted the two equations and produced the value of a_0, and then from the first equation we have produce a_1 is equal to a_0. Therefore since the solution of this equation is simply x is equal to minus a_1 upon a_0.

(Refer Slide Time: 00:03:08 min)



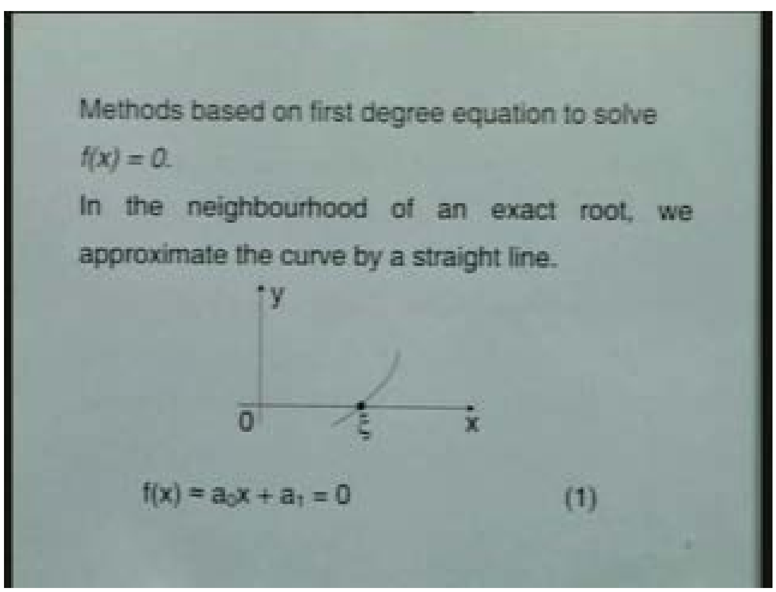
We have then written the x type approximation as xk plus one is minus a_1 upon a_0. We substitute the values of a_1 and a_0 that we have got over here and I have retained a_1 one in terms of a_0 so that division by a_0 is easier, cancellation is easy. So I write this as xk minus fk upon a_0, I substitute the value of a_0 that we have got in the previous slide. So if I substitute it and simplify, this is what we have done last time. Now here this cancels and I can write this particular form. We also said that in one of these two forms, either of these two can be used, for computation purposes three is useful for us because it is simple; whereas for error analyses we would like to use the form of two which we shall see why it is convenient for error analyses. We call this is a secant method. Now I would briefly tell you why we call it as a secant method, what is the geometrical representation before we make some comments.

(Refer Slide Time: 00:04:11 min)



Now we mentioned earlier that xk minus one and xk are any two are approximations to the root. The root need not lie particular these two values (xk minus one, xk).The reason is the method gives convergence to the root from one side of the root that means we have seen the graph.

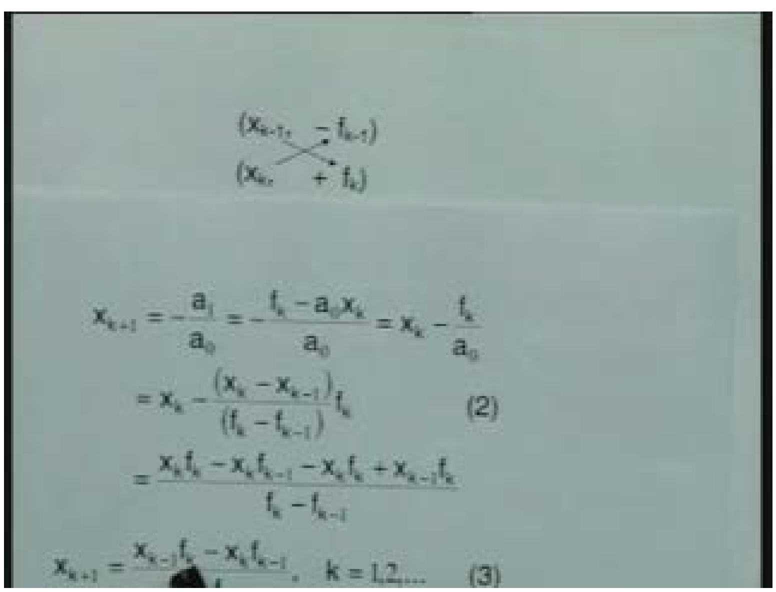
(Refer Slide Time: 00:04:32 min)



We have seen here in the graph that this is the root. Now I take any two approximations on one side and these two approximations will converge to the root from this side or from this side. Now what happens is if by chance you take the root lying between xk minus one and xk, it would take one or two iterations to go to one side of the root. So it will go either to the right or the left of the root and then start converging. So we may lose one or two iterations before this actual procedure starts.

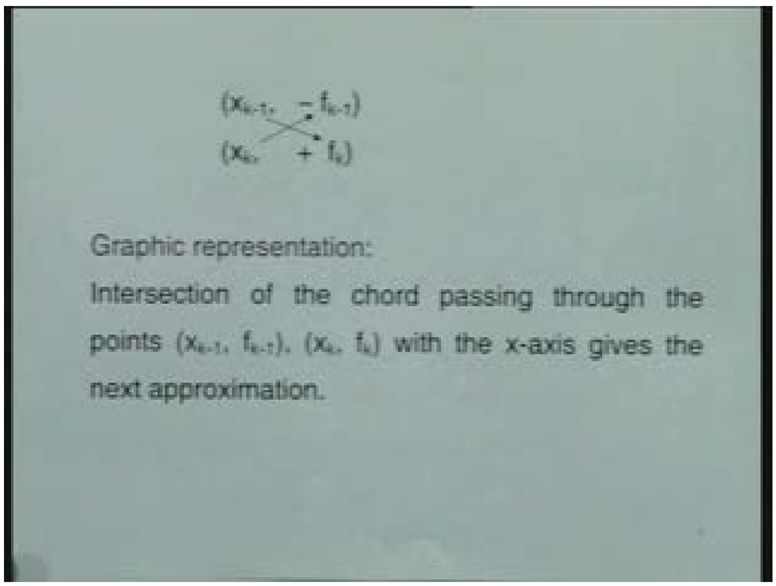
Now it is necessary to compare all the methods. The correct way of comparing these two methods are the cost of evaluation of the function. The other costs are very trivial because if you look at this method you can see that there are two multiplications, one division here and two subtractions here. So it is very trivial because the numbers of total operations are simply five operations which would not take even few micro seconds. So the cost of the computation of this is trivial therefore the only cost that will be expensive would be the evaluation of fx which we expected to be a very complicated function. Therefore we shall say that the cost of this is evaluating one function i.e. f of xk plus one at each iteration.

(Refer Slide Time: 00:06:11 min)



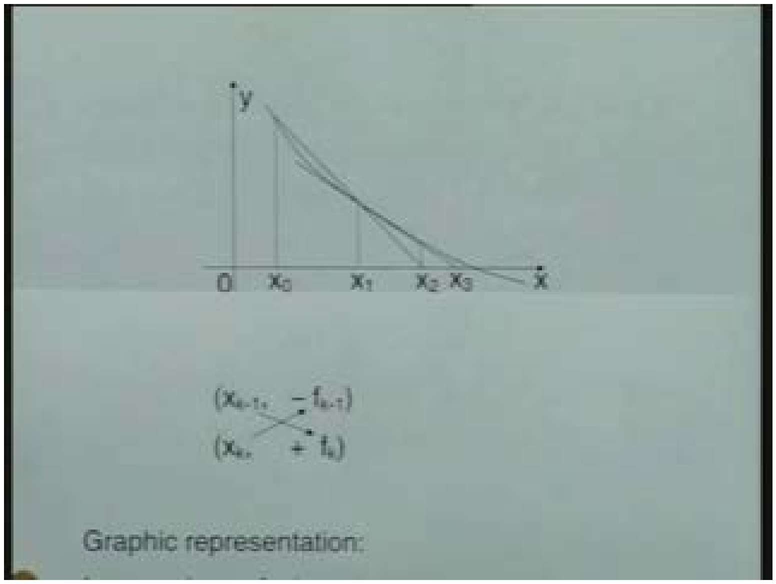
Now it is very easy to remember how this right hand side could be written to next approximation. Now if you look at the two points that we have here xk minus one fk minus one, the point on the curve is xk minus one fk minus one xk and fk. Now if you look at the numerator of this, this is nothing but the product of these two, this abscissa this ordinate minus this; so this sign represents that we are taking the plus sign for this product and this sign represent that we are taking the minus sign for this product and the denominator is just the difference between the ordinates.

(Refer Slide Time: 00:06:45 min)



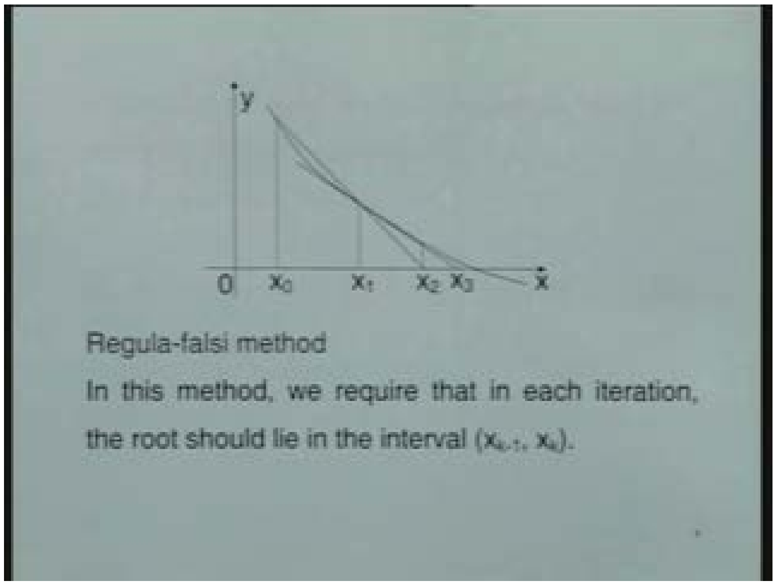
So it is very easy to write down the secant method as the product of these two, minus product of these two and subtract these two ordinates. Two approximations are available, two values of k minus one fk is available. Now this computation will give me fxk plus one. Now when I go to the next iteration what I would then {nee} need, i would need this next point xk plus one fk plus one. Now that means I have to compute fk plus one to go to the next iterations. Now when I go to the next iteration these two are available to me. Now I go to the next iteration so at each iteration we will be evaluating one extra function.

(Refer Slide Time: 00:08:27 min)



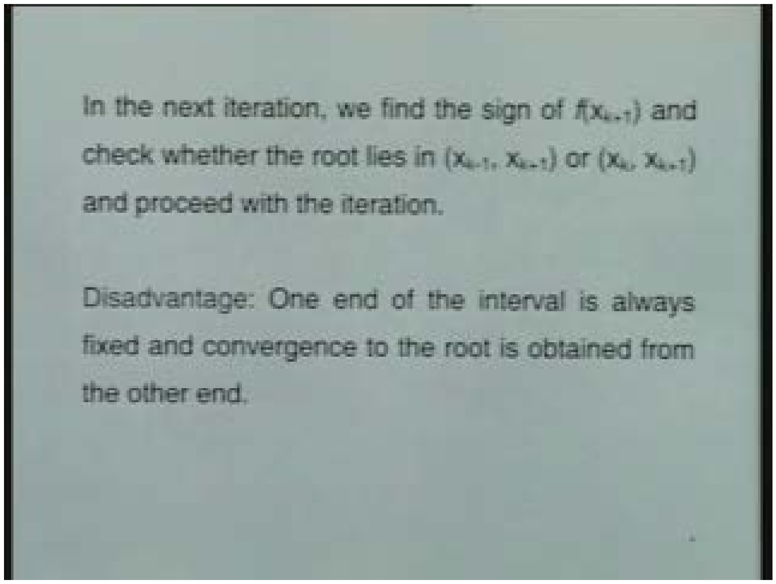
Now there is a very interesting graphical representation for this. It depicts the intersection of the chord passing through the points these xk minus one fk minus one xk fk with the x axis is the next approximation. The reason is when we approximated by a straight line, in the neighborhood of the root we are taking the next approximation at the point of intersection of that line with the x axis. So what we have really done here is we have taken this point xk minus one fk minus one on the curve xk and fk on the curve, join these two points by a straight line. Let that straight line intersect the x axis at a point and that point will be our next approximation. Now I draw this particular thing again here. I have taken a point x_0, so $x_{0,}$ f_0 is another point on the curve; this is another approximation x_1; x_1 fx_1 is another point. Now according to this I might join these two points on the graph i.e. x_0f_0 and x_1f_1 join these and this straight line intersects the x axis at x_2. Now I find the point x_2 which is one evaluation $fx_{2,}$ then I join the point x_1fx_1, x_2fx_2. Now it goes in a sequence. We are not testing whether the root is lying or nothing. We are just going in a sequence. Then f_1fx_1, x_2fx_2, we join these two and draw a line and we will cut the x_3. Now I take fx_3, I join that point x_2 fx_2, x_3fx_3 and so on. So the convergence will be going from this side to this particular root. So that is known as the secant method or simply the chord method. And as you can see here the convergence to the root is from one side. If the graph was different the convergence would be on the other side. So we do not exactly know how the graph is for a particular function and we do not know from which side it is going to converge.

(Refer Slide Time: 00:09:47 min)



Now we can modify this secant method and we are going to make very minor modifications and then we shall call this as a Regula falsi method also called as regular false position method. In this method what we do is, in the secant method we said that the root need not lie between any xk minus one xk; but in Regula falsi we will force that the root should always be within the particular interval. So if I start with an xk minus one xk, xi must lie between xk minus one xk. Then what I will have to do, I will compute my f of xk plus one. So I compute my fxk plus one and depending on this sign on f of xk minus one, I will now decide whether the root lies between xk minus one xk plus one or xk, xk plus one.

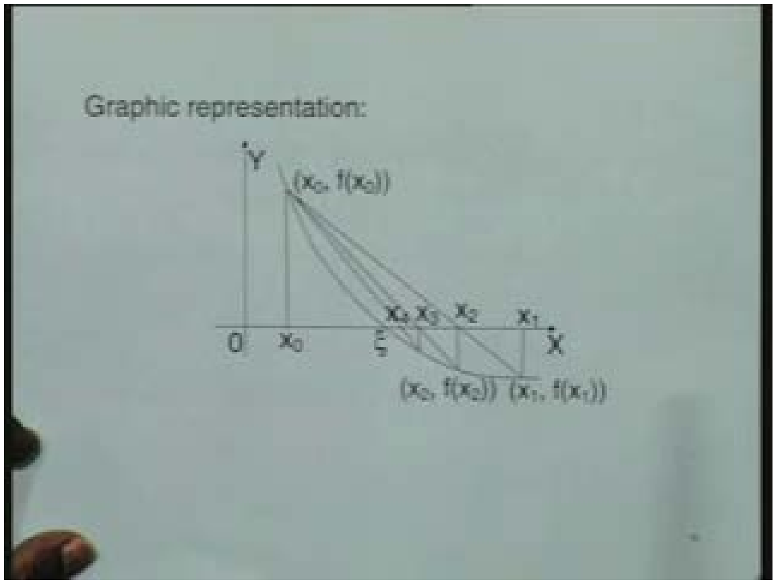
(Refer Slide Time: 00:10:27 min)



Suppose the root is lying between one and two. Now I produce the new value as 1.8. Now I would decide whether root lies between 1 and 1.8 or 1.8 and 2. Then I would proceed with a secant method further and then every time before I go to the next iteration I will test whether the root is in the given interval or not.

So here we are making sure that will always converge but the disadvantage is one end of the interval that means the starting interval or the initial approximation i.e. x_0 x_1; one of the end points of this interval is always fixed and convergence to the root is obtained from other end. So what we are stating is, if we start with x_0 and x_1 root is between this, either x_0 will be fixed and $x_1$will move towards the root or x_1 will be fixed and this will move towards the root. This is a disadvantage because suppose you start with a large length of the interval. Let us say the interval is lying between one and three, suppose the root is 2.2, so one is fixed in the root. This is the approximation or the number of iterations it will take will be much more. And the accuracy, the order of the method will also drop down because one end point is always fixed.

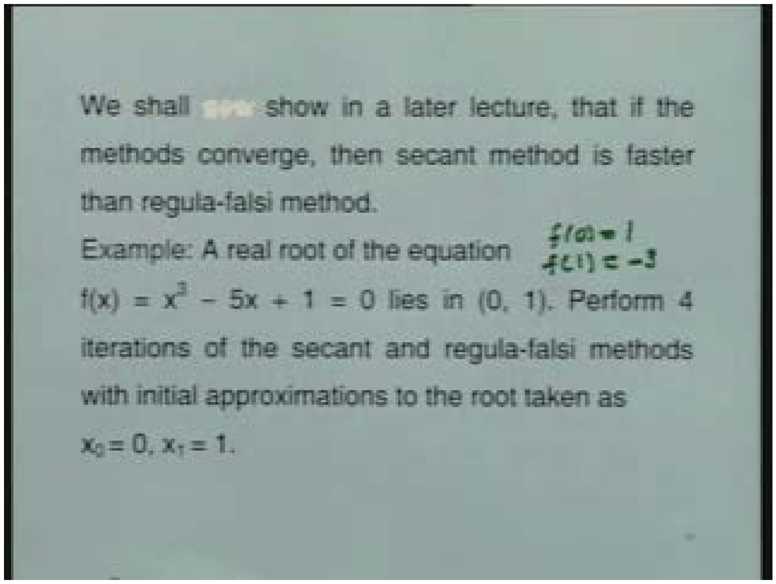
(Refer Slide Time: 00:12:22 min)



Now let me illustrate it graphically, what this really means and how it is happening? Let us take the point x_0 and x_1 in which the exact root lies. I take the points, x_0, fx_0, x1, fx_1 and I join this straight line. I get the point of intersection x_2 as my new approximation. I determine fx_2 and take the point x_2 fx_2, then the root lies between x_0 and x_2 two. Since the root lies between x_0 and x_2, I now throw away the interval x_2 to x_1. Now I consider x_0 fx_0, x_2 fx_2 and I join these two points again. It will now intersect at x_3. Again I test whether the root lies between x_0 x_3 or x_3 x_2. Root lies in x_0 x_3. Therefore I now consider x_3 fx_3 and then join the line x_0 $f_{0,}$ x_3 fx_3.

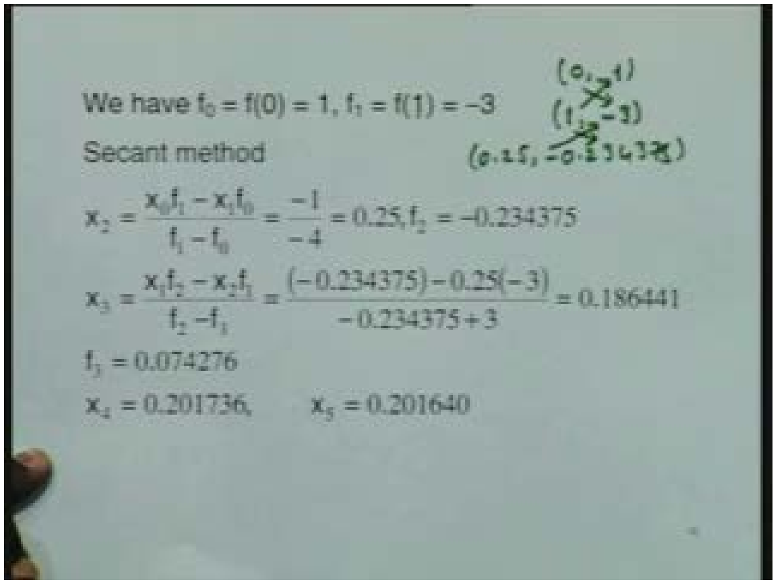
Next will be iteration. Now you can see in this iteration process, this end point x_0 f_0 is always fixed. It is only the other approximation which is moving towards this. This is what we meant by saying the disadvantage of the Regula falsi method; that the one end of the approximation is always fixed. Therefore if you want a good accuracy in Regula falsi, we must have a sufficiently small interval so that the accuracy achieved is high; otherwise it is going to take lots of iterations in order to have the convergence because the convergence rate of Regula falsi method is lower than the secant method, about which we will be seeing in a later lecture.

(Refer Slide Time: 00:14:31 min)



So if the both the methods converge, then secant method is going to be faster than Regula falsi method. We know the reason why it is happening but we will be able to show mathematically that the secant method is going to be faster than the Regula falsi method. Now let us take a simple example. Let us take the equation fx is equal to x cubed minus five x plus one. Now this example we have tested it earlier. We can show that f of zero. I can just write down what is the value of f of zero here. f of zero is equal to one here and on substituting one in f of one gives you minus three. So I know the f_0 f_1 is of opposite sign, therefore the root lies between zero and one. Now the example is, perform four iterations of the secant and Regula falsi methods with initial approximations to the root taken as zero and one. So that is the interval in which the root lies.

(Refer Slide Time: 00:16:05 min)

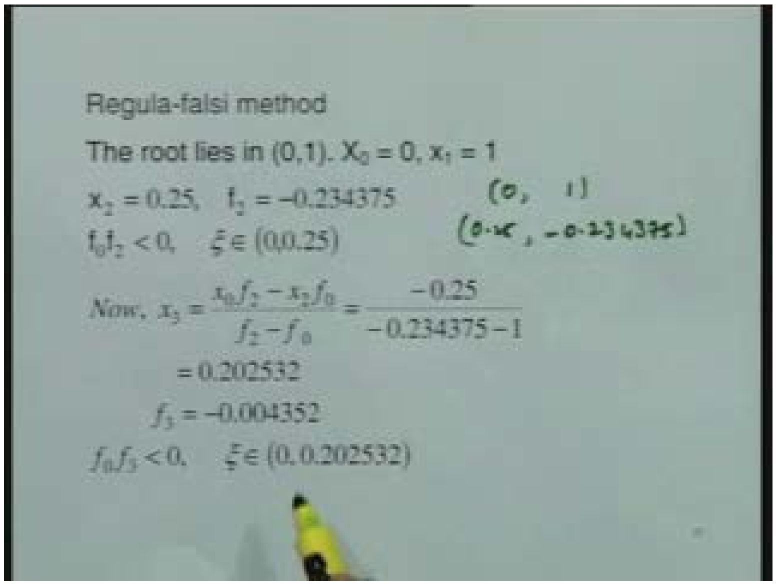


Now we compute f of zero i.e. one; compute f of one i.e. minus three. Now we have the values – x_0 is the abscissa, the value of the ordinate is one, abscissa is one and the ordinate is minus three. Therefore the next iteration would be the product of these two, zero into minus three minus one i.e. the numerator and the difference between the ordinates, minus three minus one. So the denominator is minus four. So we take this product with a positive sign, this product with a negative sign and then I find the ratio of this which is 0.25 as my next approximation. Now I determine f of x_2 that is f of 0.25. I evaluated the substitute in the given polynomial and I get this value. So the next value is 0.25 as the abscissa and I have the ordinate as minus 2.234375. Now I am proceeding forward. I multiply this two with a positive sign, multiply these two and multiply by negative sign and then divide by these two ordinates. So I can now produce the next value x_3 as this values minus this this is equal to this. Now I compute f of x_3. We have given the polynomial as x cubed minus five x plus one. Then I compute the next iterations. X_4 comes out to be 0.201736; x_5 is 0.201640. Indeed this is accurate to all the decimal places.

The application of the secant method is very simple. The amount of computation or the computed time that would take will be hardly few depending on the problem. Because as we have seen, the number of operations that are involved that is the major operations are only three;

two multiplication one division and two minor operations of addition and only one function evaluation. Therefore even if you do hundred iterations you will not even cross a second, even for computing a particular root.

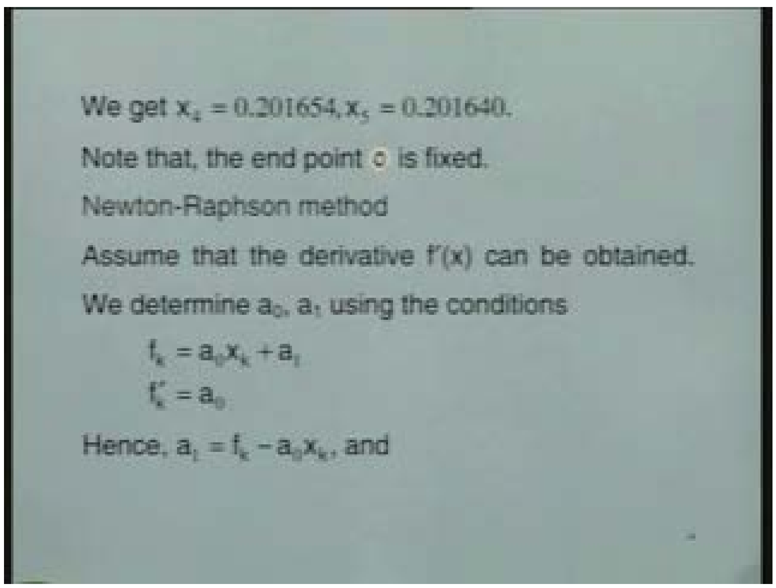
(Refer Slide Time: 00:18:50 min)



Now let us do the same thing by Regula falsi method. So we started with x_0 is zero, x_1 is one. So the first step is the same i.e. your evaluation of the first one. The first step will be the identically same because this will be the same. So therefore x_2 is 0.25, f_2 is minus 0.234375. Now I will determine whether the root lies between 0 and 0.25 or it lies between 0.25 and 1. I find f_0 f_2 is negative; therefore the root should lie between 0 and 0.25. Once I decide that the root lies between these two, in actual computation if you are writing a program for this one you will initialize the right hand side as 0.25.

If you started with x_{ab} or x_0 x_1 and if you take it, then right hand side is initialized as the new point; so that the same formula can be repeatedly used. You can put it in loop and then come back and then do it. So you can initially reinitialize this as the new value. If the left hand is changing you will change accordingly the left one, so that same formula is written in the program. So now I compute x_3 with these two values, that means now what i am taking here is 0, 1 and I am taking 0.25 and minus 0.234375. So now you can see the secant method the difference. In secant method we are taking one after the other in sequence x_0 x_1 x_2 x_3. There is no testing. Here everything will change depending on the where the root lies. Therefore the root lies between 0 and 0.25. Now you can guess one thing here, that this right hand side has moved, left hand side is fixed. So in all our future iteration zero will always be retained. Now you can see x_3 is 0.202532; f_3 is minus 0.04352. Now again f_0 f_3 is negative, therefore the root lies between these two values.

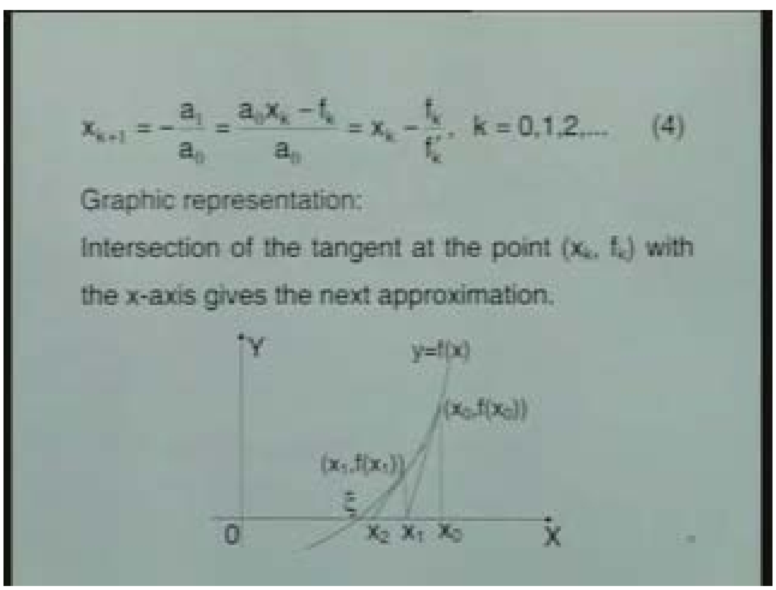
(Refer Slide Time: 00:19:37 min)



Now I have given the next values as x_4 as this and x_5 as this and we have seen that the left end point zero is always fixed. We have now derived the secant method and a variation to it, so that the secant method converges faster than the Regula falsi. Now in many problems this convergence may not be sufficient for us which means it is not converging sufficiently faster for us in a particular problem; so you would like to have some methods which converge faster than the secant method. One such method is called Newton Raphson method. In deriving the Newton Raphson method, in literature you will be surprised that Newton Raphson method is today also one of the most powerful methods for finding a root of the equation. We will see later on a multiple root of the equation or a system of nonlinear algebraic equations in which this method can be adopted. So they will see why it is so, even though higher order methods are available we still prefer this because of some reason.

We assume that the derivative of dash x can be obtained. So that means, given fx, differentiate it and get f dash x. Then I will use fx and f dash x to find the straight line. We are still doing the approximation by straight line. So I will try to evaluate a_0 and a_1 one which is there in the equation in the straight line by using the data of fx and f dash x. Therefore I need one approximation only for using the Newton Raphson method. Let x_0 or xk be any first approximation. If xk is approximation, I substitute it in the given polynomial approximation that is your linear polynomial. fk is $a_0$xk plus a_1. Now I differentiate a_0x plus a_1. Derivative is simply a_0. Therefore f dash at xk is going to be constant that is a_0. Therefore a_0 is found; a_0 is f prime of k. Then I can find a_1 from the first equation as fk minus $a_0$xk. Since we have approximated equation by a straight line, its derivative is only a constant. Therefore derivative at that point, wherever the approximation is, it is simply a_0.

(Refer Slide Time: 00:22:05 min)

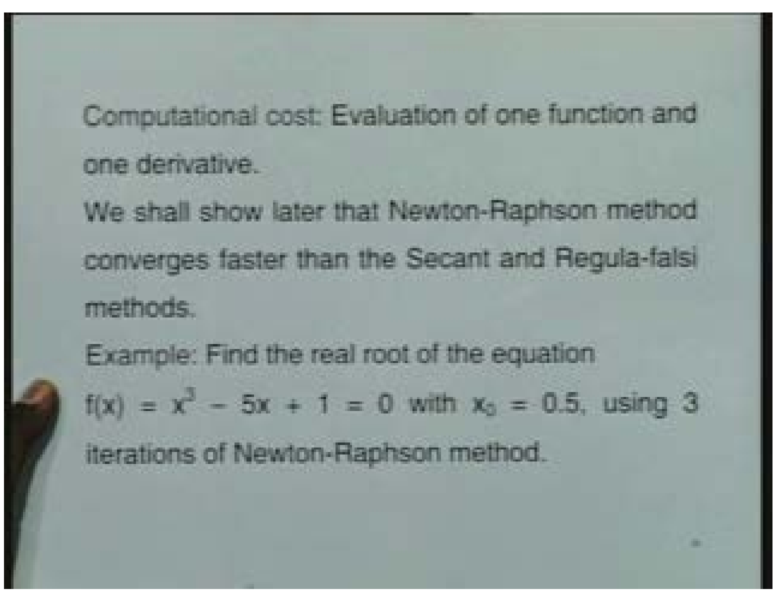


Now the value of x i.e. next approximation is minus a_1 upon a_0. So I would therefore write xk plus one is minus a_1 upon a_0. We have just now derived a_1. a_1a_0 xk minus fk by a_0. I take this minus sign inside, divide it out and I will have xk minus fk upon f dash k i.e. k is equal to 0, 1, 2…

Now before we make a comment on the application of this let us see how this is same as what we are now approximating by straight line in the neighborhood of the exact root. Now what we are doing in the Newton Raphson method is, we are taking the point xk fk on the curve. We write down the equation of the tangent at this point to the curve and take the point of intersection of the tangent to the curve with x axis as our next approximation. So I take a point x_0 as starting approximation, x_0f_0 as my point on the curve, then I draw the tangent to the curve at this point. Now this line meets the x axis at x_1 and this is my new approximation.
Now I take the point on the curve x_1 fx_1, I draw again the tangent at this particular point. This tangent cuts the axis at x_2.

Now this is my next approximation. So I continue on and then approach the root xi here. This is the graphical representation of the Newton Raphson method.

(Refer Slide Time: 00:24:08 min)

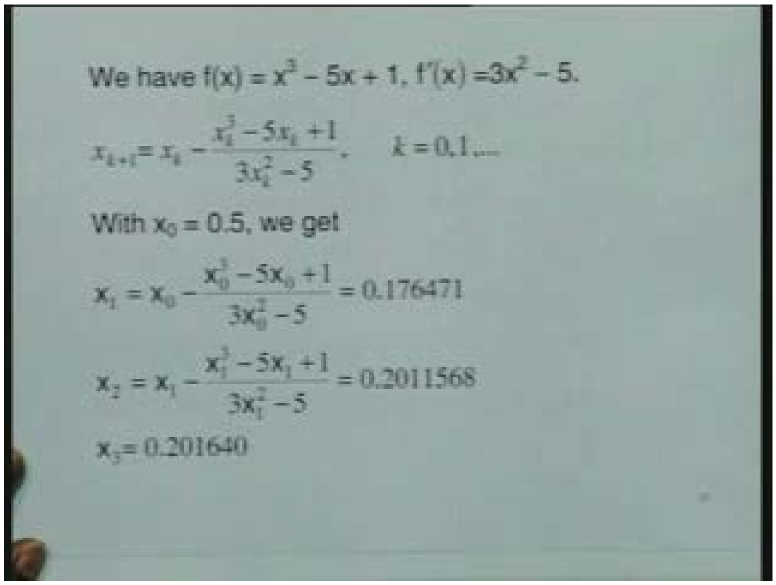


Now you can see that the root is converging from one side but it has got better properties. Let us see what these better properties are. Now first of all the computational cost has increased compared to secant method. We have to evaluate one function and one derivative; given a xk we need f of xk and f primate at xk. So the cost of evaluation has gone up by one function. But we shall show later that the Newton Raphson method converges faster than the Secant and Regula falsi methods. Since we get faster convergence for this we do not mind if Newton Raphson method takes ten iterations and whether the secant method has taken fifteen iterations. So if you are talking of the cost, secant method will be taking fifteen function evaluations but this has taken ten iterations; but the total evaluations are twenty; but the number of iterations that will be taken will be much less. So that the total cost that we are computing when we get the final answer would be much less for the Newton Raphson method than for the secant or the Ragula falsi methods.

Now let us repeat the example that we have just now done. So find the real root of the equation x cubed minus five x plus one is zero, it is the same example. But I am starting by giving an initial approximation as 0.5 using three iterations of the Newton Raphson method. Now here the idea is once you have found out the interval in which the root lies, we arbitrarily take any point in this interval as our starting approximation. For example f of zero is lying between one and four. f of one and f of four is known, whichever one is closer to zero will be that particular point. Suppose the value of the f_{81} is say let us say minus 0.5, f_{84} is plus 1.5. The root lies between this but one is closer to zero. Therefore I will take a value closer to one as my initial approximation. So that will reduce the number of iterations and second most important thing that happens here is, as I just showed in the slide, the convergence of Newton Raphson method is also from one side of the root.

You may ask what would happen if instead of taking 1.2 as your starting approximation, you take 3.8 as the approximation; then since it is to converge from this sometimes Newton Raphson method goes around that root once or twice then starts converging from here. It depends therefore on how properly you have chosen initial approximation and the convergence will be obtained. But however if the number of iterations being taking as fifty, you may not mind to lose three four iterations in the beginning so that it will adjust value to come to the one side and then start converging.

(Refer Slide Time: 00:27:06 min)

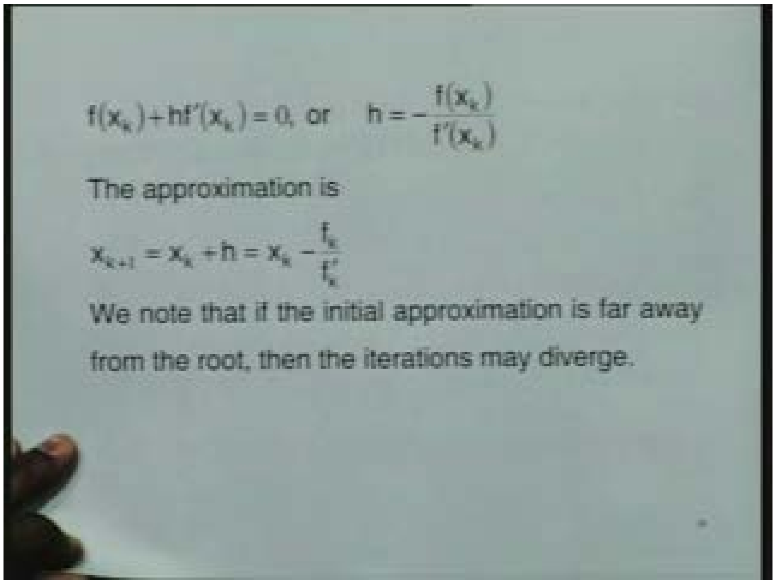


So here I need to evaluate two quantities that is our fx and f dash x fx is x cubed minus five x plus one. I differentiate it and I get three x square minus five. Then I will first of all construct my Newton Raphson method. The method will be xk plus one is xk minus f of xk i.e. evaluation of fx of xk derivative in the denominator, three times xk square minus five. So this is my basic iteration method - Newton Raphson method, from which I will substitute the successive values and then get the next iteration onwards.

In the problem we are given you start with approximation x_0 is 0.5. So we will substitute x is equal to 0.5 in this and write x_1 is equal to x_0 minus x_0 whole cube minus five times x_0 plus one by three times x_0 square minus five and we simplify this and get 0.176471. Now I use this value of x_1 to compute the next approximation from here for k is equal to one. When k is one, have x_2 as equal to x one minus x one cubed minus five x one plus one by three x one square minus five. Now I compute this by using this x_1 and I produce this value; and I have not written the third iteration steps but the third iteration value is 0.201640 which is the same value which we have obtained in the two other cases but with four iterations, five iterations which we have obtained. We can derive it in an alternative way which helps us also to construct still higher order method than the Newton Raphson method. Now at the moment I may state we will prove it this lecture or the next lecture that Newton Raphson method has got what is known as second order convergence whereas the Regula falsi method has got order of convergence as one; secant method has got it not an integer it but as a fraction, it is 1.618 as the order of convergence. We will see later what does this order of convergence really mean but let me just say that it connects the errors at the present step to the previous iteration. So that means the error at xk is connected to error at xk plus one; it shows how the error is changing. If the order of convergence is quadratic, the error in the present step will be the square of the error at the previous step.

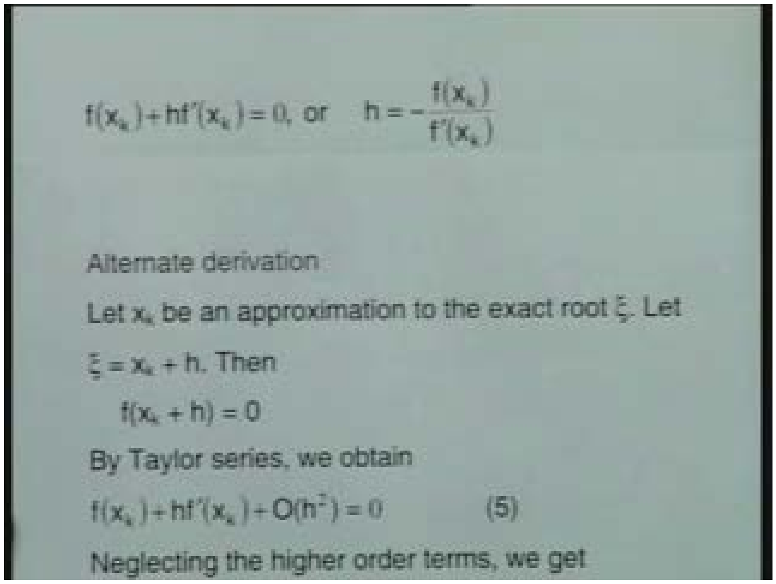
Suppose in the previous step you were at an error of 0.1, you can surely say that my present result is point one whole square accurate which means 0.01 to ten to power minus two accurate. Next iteration is ten to power of minus two whole square, so ten to power of minus four accurate. So the convergence will be very fast and that is quadratic convergence, whereas if I take the Regula falsi the order is one. If the error is 0.1, I would expect only one decimal place. So in the next case it is again order one, so I will get only one decimal place, whereas Newton Raphson method is jumping by square of the error in the previous case; and we would expect the error to be less than one in any case. So if the error is greater than one you may say it is meaningless but yes it is meaningless because that error means we are closer to the root, therefore it is less than magnitude one. Therefore the error would always be smaller than one, so we are talking about square or cube or so.

(Refer Slide Time: 00:30:50 min)



Now I would like to have an alternate derivation for Newton Raphson method which would help us to construct a still higher order method than Newton Raphson method. Let us take xk as approximation to the root xi. Now if I add the quantity h - the correct quantity, I can get the exact root that means xk plus h is equal to xi which is my exact value. Since xi is exact, f of xi must be zero. By definition f of x is zero, x is equal to exact solution, xi is the exact solution, therefore f at xi should be zero i.e. fx k plus h. Now just open it up by Taylor series. Write down the Taylor series and retain only first term. So f of eight xk, h times f prime at xk, plus order of h square terms (the higher order terms of h square terms).Then to derive Newton Raphson method we simply drop the second or higher order terms. We drop the order of h square terms, which means I set all of them as zero. I neglect this higher order terms. If I neglect the higher order terms, what is left out for you is f of xk plus h times f prime of xk only.

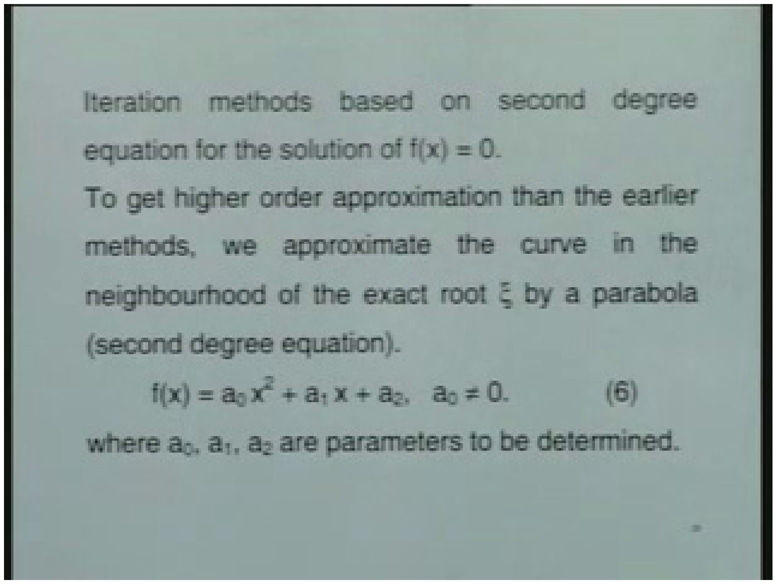
(Refer Slide Time: 00:32:05 min)



Now if I neglect this higher order terms what I would get is f of xk plus h times f xk is equal to zero. I solve for h. Now this is an approximation; h is equal to minus f of xk by f dash of xk. But mind you h is here an exact value, but here we have now approximated in this. Therefore whatever I get here is not exact h but an approximate h. Now the new approximation therefore will be xk plus h, because what I have got here is not the exact value of h. So what I am getting here is not the exact xi but an approximate xi. So the next approximation will be, I will put it here xk plus one is equal to xk plus h, therefore xk minus f of k minus f dash k. Therefore simple Taylor series would give me the alternative method of deriving the Newton Raphson method.

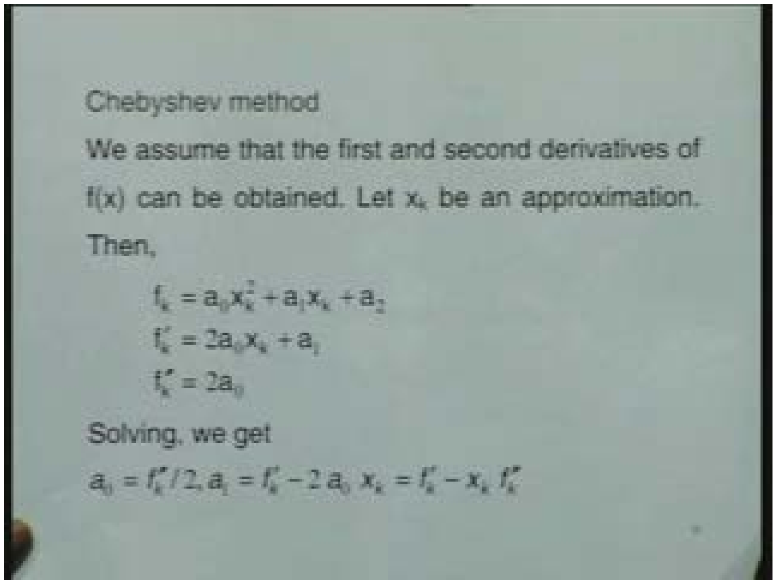
We are just making a comment here that if initial approximation is far away from the root then the iterations may diverge. For example the root lies at minus ten, if I start the approximation on the positive side with plus five the root is fifteen units away. Therefore we may not expect the Newton Raphson method or the secant method to converge at all. Therefore the initial approximation should be closer to the root and that will be obtained by using the initial value over the theorem that we have given. We can use the intermediate value theorem to get the root.

(Refer Slide Time: 00:33:46 min)



Now let us go to the methods which I can construct superior to Newton Raphson or the secant method which I would call as iteration methods based on second degree equation for the solution of fx is equal to zero. Now in the earlier discussion we said that in the neighborhood of the root I can approximate the curve by a straight line and then obtain the methods. But now I would say, I would like to take a better approximation than a straight line. The better approximation than a straight line is a parabola passing through three points which is nothing but a second degree equation. So instead of approximating by a linear polynomial, I will approximate x by a quadratic polynomial which will be a parabola that is a second degree equation. Now to get higher order approximation than this we approximate the curve in the neighborhood of the exact root by parabola which is a second degree equation. So I will take the second degree equation as a_0 x square plus a_1 x plus a_2, where of course a_0 is not equal to zero; otherwise it is going to be a linear polynomial and these three parameters are to be determined. Now again the way in which you determine these three parameters a_0, a_1, a_2 would design a new method and which we can call by different names.

(Refer Slide Time: 00:35:09 min)

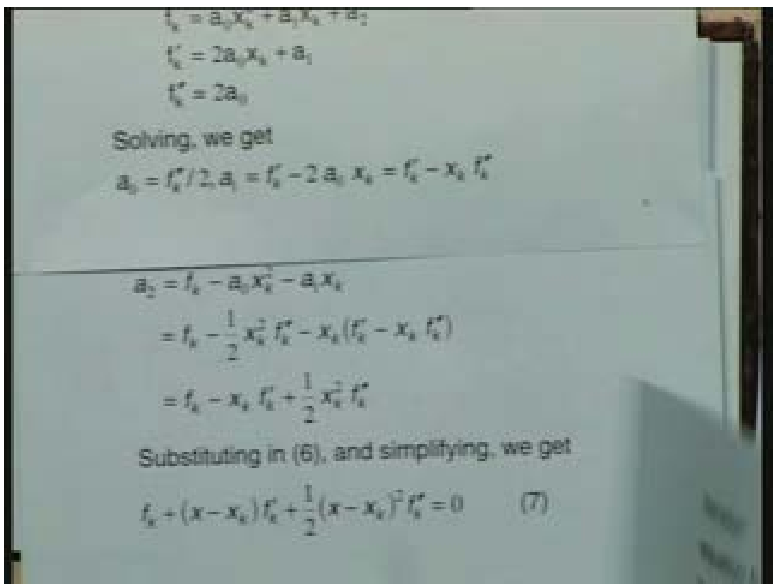


Now the first method is called the Chebyshev method. Now in logic the Chebyshev method is the alternative derivation of Newton Raphson method we are given using the Taylor series. The Chebyshev method states that instead of taking order of h square by neglecting, use order of h square terms also but neglect order of hk. Keeping the second order terms would imply that we are approximating the curve in the neighborhood of the root by a quadratic curve that means a second order polynomial approximation.

I am deriving the values of a_0 a_1 a_2 here in a different way. We start with only approximation xk. Now when we derived the Newton Raphson method we said, let us assume that f derivative exists and evaluating f dash xk. Now in Chebyshev method we say, let one more derivative exist which means second derivative also exist i.e. f, f prime and f double prime. So we will use the three values f, f prime and f double prime. So fk is $a_0$xk square plus $a_1$xk plus a_2. If I differentiate the given approximation, I will get here two times a_0x plus a_1. So I take a to the point xk, I will get f prime at xk, two times $a_0$xk plus a_1. If I differentiate it once more I simply get two times a_0. So the second derivative at xk is two times a_0. Therefore this gives us three equations in which we can find the solution of a_0, a_1, a_2 in the backward direction. The last equation gives us a_0 is half of f double prime at k.

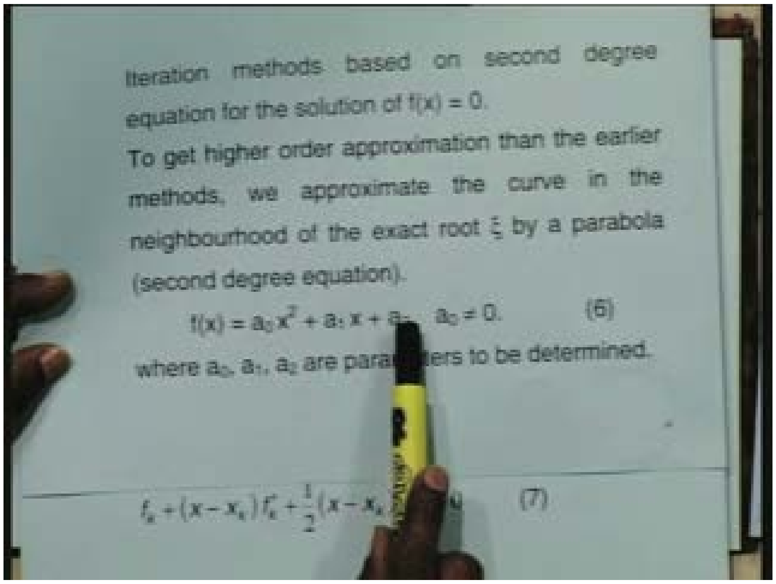
So a_0 is determined. Substitute it in the previous equation and you will get a_1 is equal to f prime of k minus twice $a_0$xk, so I bring it to this side. Then the a_0 is f prime k, f double prime k by two. I substitute it here so, a_1 one is simply f prime of k minus xk f double prime of k. Now I will use these two values of a_0 and a_1 over here and find my a_0 from here.

(Refer Slide Time: 00:37:45 min)



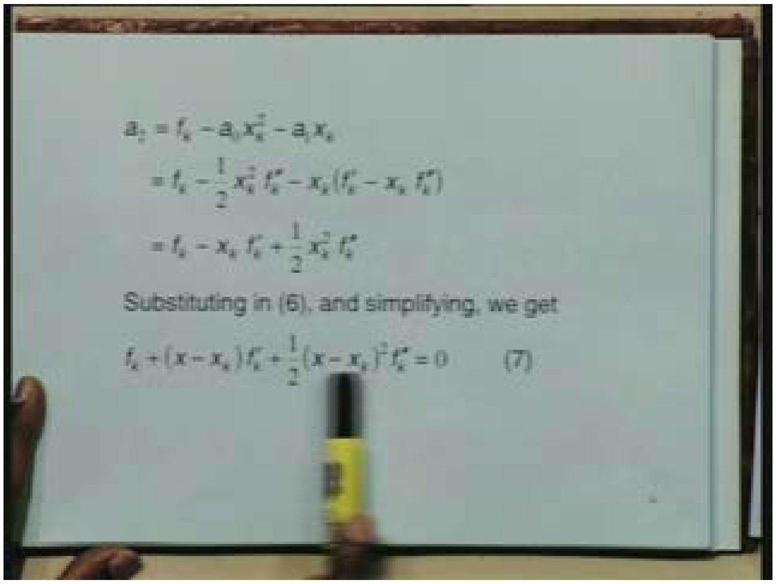
So what we have done here is I have taken a_0 from here. This is fk minus $a_0$xk square and this is minus a_1 of xk. I substitute the value of a_0 which is f double prime k by two and I substitute the value of $a_1$that is f prime k minus xk f double prime k and this minus xk is multiplying it. Now here we can see that these both are second derivative, so they simplify and we have fk here and this a minus xk f prime k and this gives me plus half xk square f double k.

(Refer Slide Time: 00:38:25 min)



Now I have determined the values of a_0, a_1, a_2 and what I will have to do here is, I have to go back and substitute.

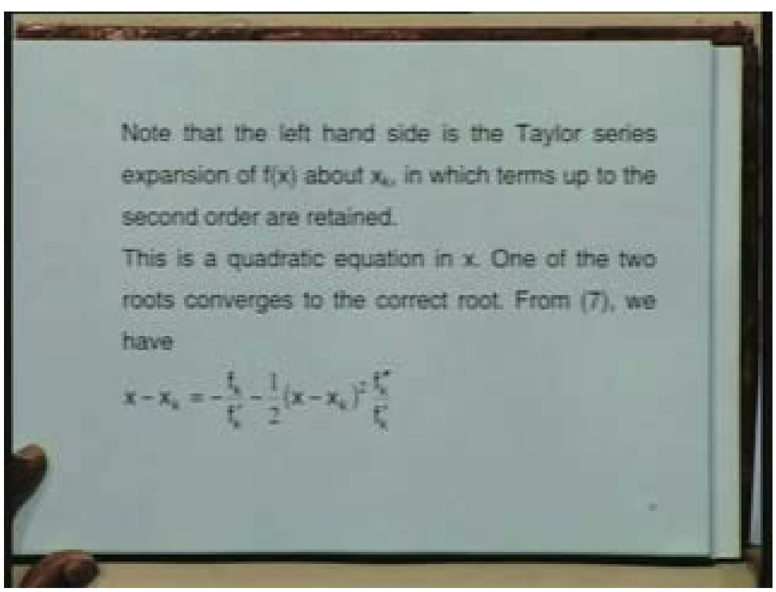
(Refer Slide Time: 00:38:45 min)



I go back to the previous slide and substitute a_0, a_1, a_2 over here. So if I substitute it back, I will leave little bit of algebra to simplify; By substituting I simply collect the terms f prime k and f double prime k and when I collect it just comes out like this. All these terms combine to form fk plus x minus x xk of prime k plus half x minus xk whole square f double dash k is equal to zero. Now here we have to make a decision. This is a quadratic equation; therefore it will cut the x axis at two points. Only one of the points is our root, the other point is not a root. Therefore we have to throw away one of the roots and then take one of the root as our approximation. Which one of these will be taken as a root will be determined from here. Therefore out of these two roots, to find the next approximation we will use a further approximation inside this so that we are definitely towards the root.

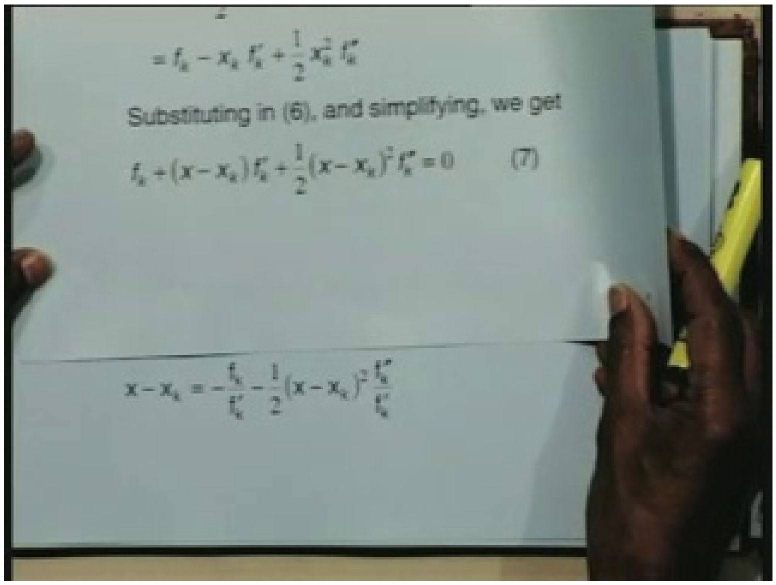
If I had written the Taylor series alternative derivation of the Newton Raphson method, then I have neglected order of h square terms and got my Newton Raphson method. Now if I retain the order of h square terms then I would get the Chebyshev method. You can see this is nothing but the Taylor expansion of fxk plus h and h is xk x minus xk. So this is fk plus hf one k plus h square by two f double prime k is equal to zero. So we have put the value. Therefore in the Taylor expansion if I use one more term order h square I get Chebyshev method.

(Refer Slide Time: 00:40:31 min)



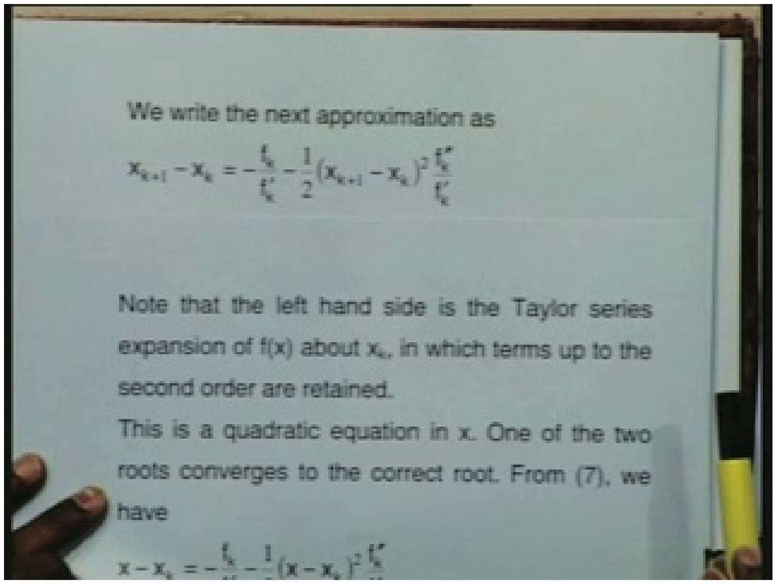
Now as I said one of the roots only converges to the root. We have to decide which one would be that root.

(Refer Slide Time: 00:40:39 min)



To decide which one would be that root, I will retain this on the left hand side, this middle term on the left hand side, take these two terms to the right hand side. I have taken fk to the right hand side, minus half x minus xk square f double dash to the right hand side, and then divided by f prime k. So I used this equation - the middle term on the left hand side the other two on the right hand side and written this particular thing which is still exact.

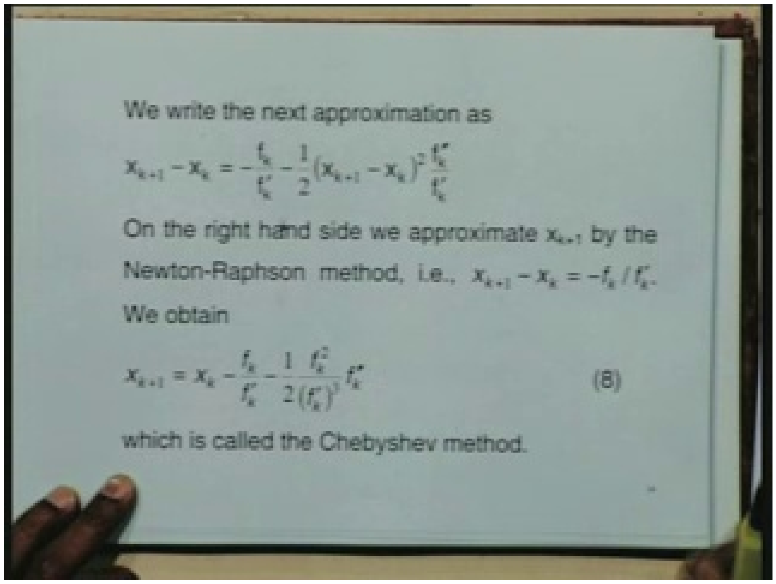
(Refer Slide Time: 00:41:17 min)



Now I will define my next approximation. As I replace this x by xk plus one, I write the next approximation as xk plus one here, xk plus one here which will give me the the next approximation. Now it is still quadratic in xk plus one. It is here I will now make an approximation wherein I will use one of the approximations which is surely converging to the root. That means I will insert either a Newton Raphson method or a secant method over here so that it will now become explicit. Left hand side is only xk plus x right hand side will be in terms of xk only. On the right hand side we approximate xk plus one by the Newton Raphson method i.e. xk plus one minus xk is minus fk by f prime k. That is our Newton Raphson method.

I have brought xk from the right hand side to left hand side, so xk plus one minus xk is minus fk divided f prime k. So I would like to replace this xk plus one minus xk by the Newton Raphson method, which says that this can be approximated by minus fk by f prime k.

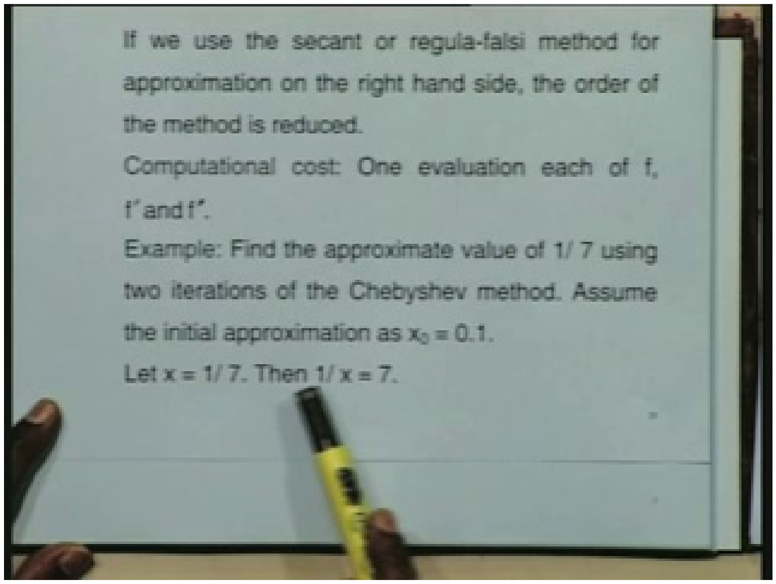
(Refer slide Time: 00:42:28 min)



If I replace this xk plus one minus xk by the Newton Raphson method, which says that this can be approximated by minus fk by f prime k, I will get xk plus one here. Take this to the right hand side as xk, this term retained as it is, fk upon f dash k minus half here, this is square of this; therefore I will have fk square, denominator is f prime k square; but there is also f prime k here, so f prime k whole cube into f double dash k. So this is how the computation of the Chebyshev method looks like. Obviously we are evaluating fk f prime k f double prime k.

Now we would like to make a comment here that we are replaced by the Newton Raphson method. I made a mark earlier that Newton Raphson method is of second order accuracy. So I used a second order accurate method here and then got this one. I will show later on that this method now becomes a third order method which means the error at each step is now reduced by factor of whole cubed. So this is of a next higher order than the Newton Raphson method but if I use it in place of Newton Raphson method or a secant method or a Regula falsi method, the order of the method falls down from three to two. Therefore the method is of no use because if it is only of order two it is going to more expensive than Newton Raphson method. Newton Raphson method is of order two. But here it is more expensive, therefore if it gives me next higher order method I would be willing to use one more function evaluation and then get the next higher order method; because the convergence number of iterations will be much less.

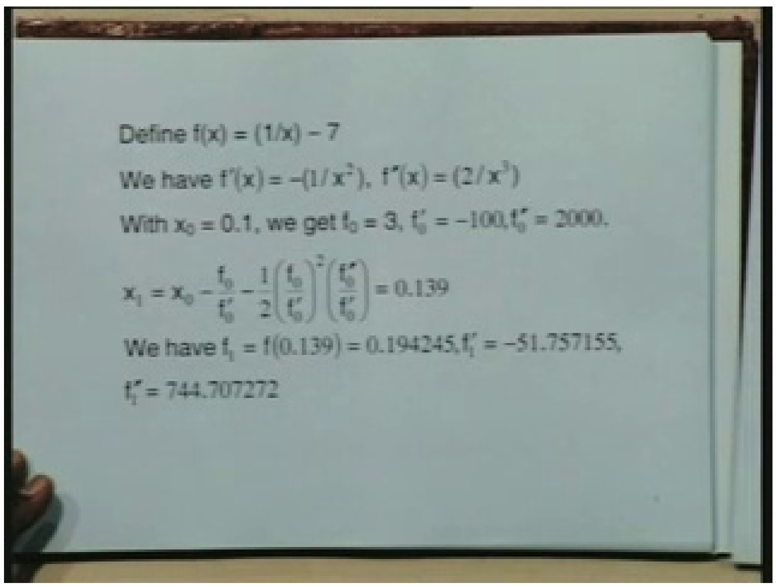
(Refer Slide Time: 00:44:14 min)



Now if you use the secant or Regula falsi method for approximation on the right hand side the order of the method is reduced. Now to again count the computational cost, one evaluation each of f, f prime and f double prime, we have three evaluations compared to Newton Raphson method which has got one f and f prime i.e. two evaluations. But since the order is three, the total computation time that is taken for a problem would be less than even the Newton Raphson method. The reason being, here accuracy is third order so the error is reduced by a power of three. Therefore the accuracy that you would be getting will be much faster than what it would be for a secant method. Therefore even though at each iteration we are having three evaluations, the total computational cost will be much less than what it would be for a secant method and most of the times less than the Newton Raphson method. Only thing is if you say that I cannot evaluate f double dash in my problem, then we will not be able to use Chebyshev method. I will have to take Newton Raphson method or I must find an alternative method to get the root.

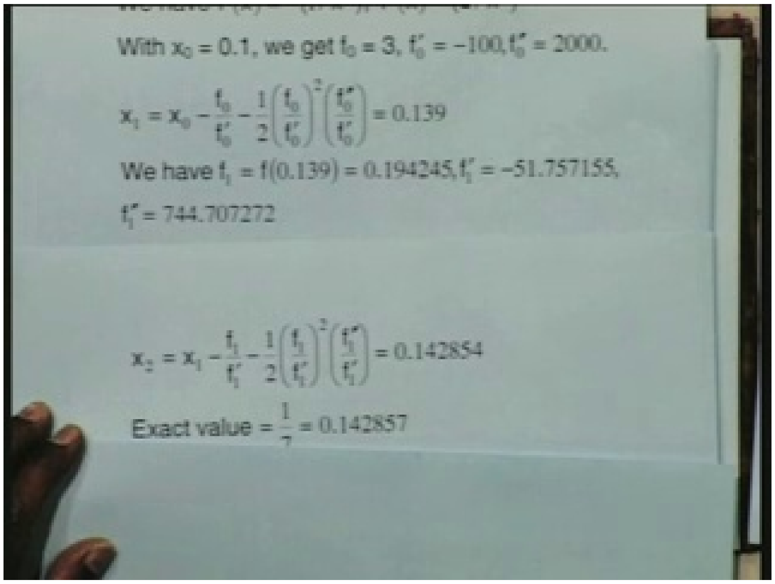
Now let us take an example. I want to find the approximate value of one by seven using two iterations of the Chebyshev method. Assume the initial approximation as x_0 is equal to 0.1.Now to do a problem like this I will first of all need to construct my fx. Only when I construct fx is equal to zero I would be able to construct the value. Let x be equal to one by seven, and then I will invert it and write it as one by x is equal to seven. Then I will take my fx as one by x minus seven. So I would therefore define my fx as one by x minus seven. So given a problem typical problem like this it is necessary for us to correctly write down what is my function x so that anyone of these methods can be used.

(Refer Slide Time: 00:46:22 min)



Therefore I would define fx as one upon x minus seven and applying the Chebyshev method I reach two derivatives. I differentiate this - f prime x is minus one upon x square; second derivative is two upon x cubed. Now the next remaining thing is simple computation .So with x_0 is equal 0.1 I find out what is f_0 i.e. one by point one, then ten minus seven i.e. three f_0 prime which is one upon ten to the power of minus two which is minus hundred; this is two upon ten to the power of minus three and that is two thousand. So I just substitute this in the formula that we have written here, x_0 minus x_0 f prime zero minus half f_0 minus f_0 prime whole square into second derivative minus. I compute this and get the value. Now I have to use this value and then compute my f, f prime and f double prime. So at each iteration I need to compute these three values - this f at this value, f prime at this value and f double prime at this value which I call as f_1, f_1 prime and f_1 double prime. Now these values shall be used in this same formula x_2 is equal to x_1 minus f_1 by f_1 prime minus half, f_1 by f_1 prime square, f_1 prime by f_1 prime.

(Refer Slide Time: 00:48:12 min)



Now you can see that I am writing x_2 is x_1 minus f_1 by $f_1$prime, half of f_1 by f_1 prime whole square, f_1 double prime by f_1 prime and by using these values I get 0.142854. Now I can stop after two iterations. The exact value turns out to be 0.142857. We have got five plus accuracy in just two iterations. Of course we have started with approximation as 0.1. If we had taken a different approximation maybe it would have taken one more iteration but in two three iterations it has given the exact value.