Numerical Methods and Computation

Prof. S.R.K. Iyengar

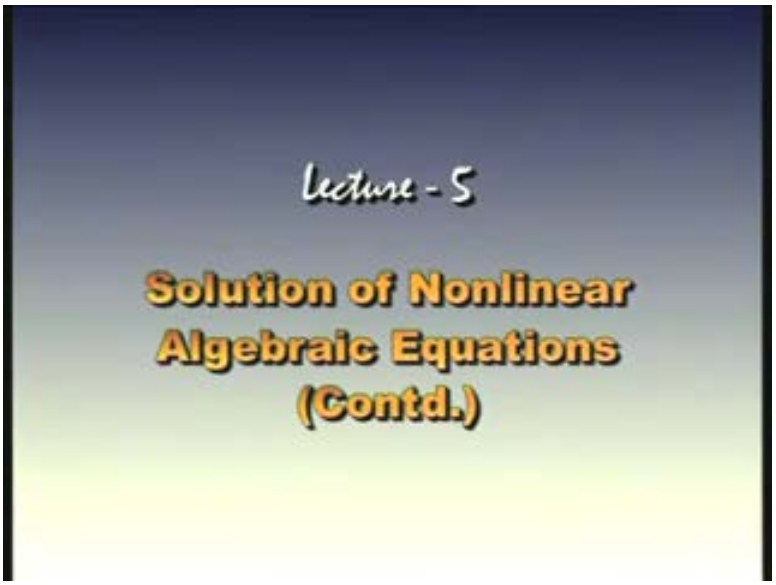
Department of Mathematics

Indian Institute of Technology Delhi

Lecture No # 5

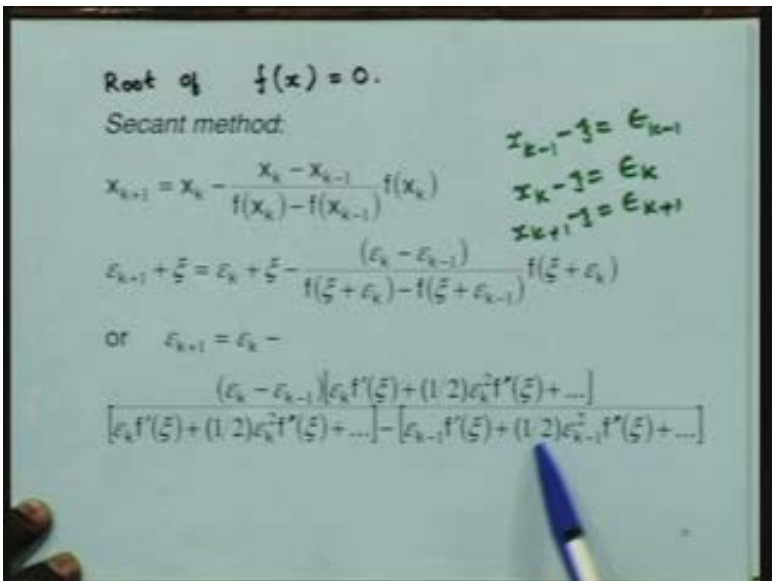
Solutions of Nonlinear Algebraic Equations (Continued)

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



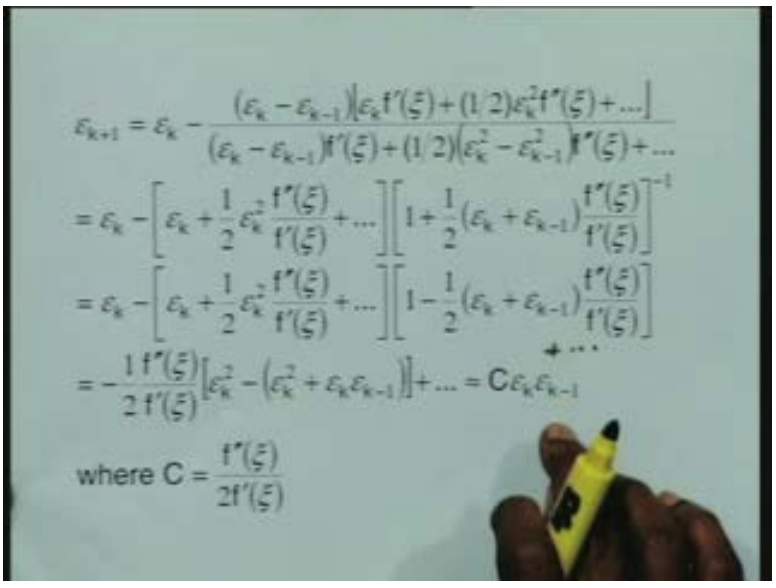
In our previous lecture we were discussing about the convergence of the secant method. Let us briefly look at what we have done last time so that we can derive the rate of convergence of the secant method.

(Refer Slide Time: 00:01:19 min)



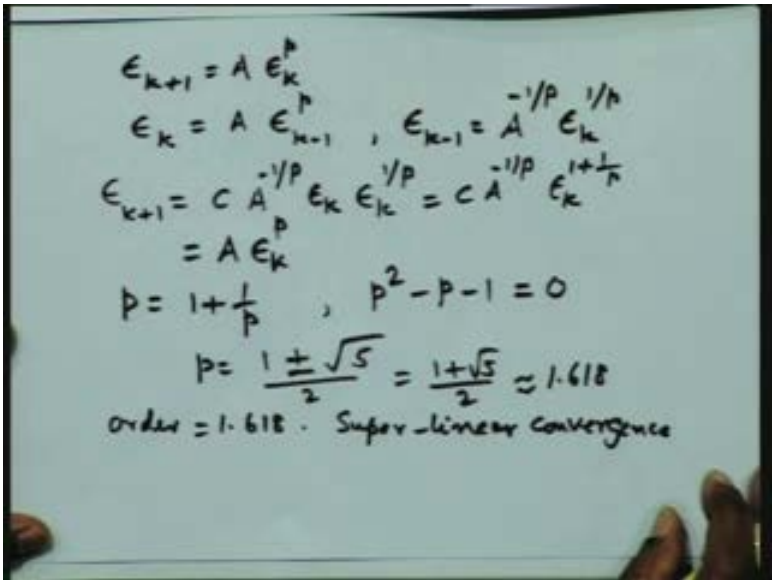
Now what we were trying to find was we have to find the simple root of the equation fx is equal to zero. We have constructed the secant method as follows: we have two initial approximations xk minus one and xk. Based on that we are using the formula xk plus one is xk minus xk minus xk minus one. In the denominator we have f of xk minus f of xk minus one. This factor is multiplied by f of xk. Then in order to obtain the convergence rate we are substituting for the errors in the solution as epsilon k minus one is the difference between the exact solution and the estimate xk minus one. Similarly epsilon k and epsilon k were defined. Then what we did was we substituted the expressions for xk minus one from here as epsilon k minus one plus xi; xk is equal to epsilon k plus xi; xk plus one is epsilon k plus one plus xi. Then the denominator was simplified. We take this up to the numerator, put it up to the power of minus one and use the binomial theorem for expanding it.

(Refer Slide Time: 00:02:28 min)



We have expanded the denominator in this particular form and multiplied these two factors and finally cancelled out. Then we have simplified and we have seen that this epsilon k square cancels here and if I denote this by constant C then I can write this as simply C times epsilon k epsilon k minus one. This was the expression that we have derived and now we would like to find the expression for the error or the order of convergence that we have defined relates the error at the step xk plus one and xk only. For example secant, Muller, all of them would contain epsilon k minus one epsilon k minus two. We must use the definition of the rate of convergence to reduce this into the form of epsilon k plus one is some factor of epsilon k to the power of something. That power would be the rate of convergence. So what we would do here is in order to get this particular result we will do the following. I would just briefly write what we will do.

(Refer Slide Time: 00:07:01 min)

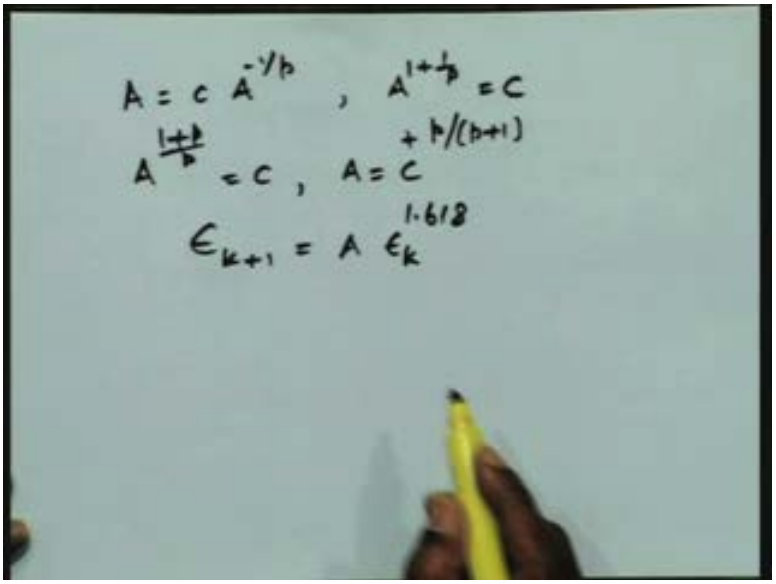


I will have to get a formula of the type epsilon k plus one is equal to A into epsilon k to the power of p. This is the definition of the rate of convergence or order of the method. In order to eliminate epsilon k minus one that I have here over here, I would substitute k is equal to k minus one, so I would get here epsilon k is equal to A into epsilon k minus one to the power of p. Now I will express epsilon k minus one back in terms of epsilon k. Therefore what I would write here is epsilon k minus one would be this to the power of one by p and A to the power of minus one by p. So I will have A to the power of minus one by p into epsilon k to the power of one by p.

Now I am able to express epsilon k minus one in terms of epsilon k and therefore I will substitute this in the expression that we have written over here, so that I will now have here epsilon of k plus one is equal to C. Let us substitute first epsilon k minus one here so I would have here A to the power of minus one by p. This epsilon k is there and now I have got epsilon k to the power of one by p. Now let us simplify this. This is C A to the power of minus one by p epsilon k to the power of one plus one by p but by definition epsilon k plus one should be equal to A into epsilon k to the power of p. So this should be equal to A into epsilon k to the power p. Therefore the constant must be equal; the powers should be equal in order that this is an identity. Therefore I will have p is equal to one plus one by p. We will equate the coefficients in a moment so the

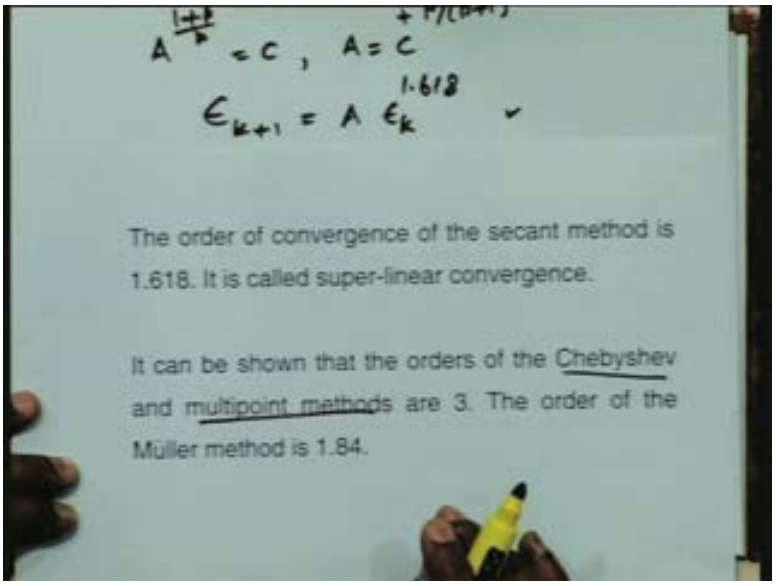
power should equate; therefore p is equal to one plus one by p. Now let us try to solve this. This gives you p square minus p minus one is equal to zero. So I am taking everything to the left hand side and solve for p. The value of p will be equal to one plus minus root of five by two. Now the power that we are using here in the definition is a positive quantity. Therefore out of the two roots I will chose the positive quantity and that is your one plus root five upon two. If I evaluate this, it is approximately equal to 1.618. Therefore the order of the secant method that we have p is 1.618 therefore we will call this order as 1.618 and we shall call this as super linear convergence between linear and quadratic. So I will call this as super linear convergence.

(Refer Slide Time: 00:07:19 min)



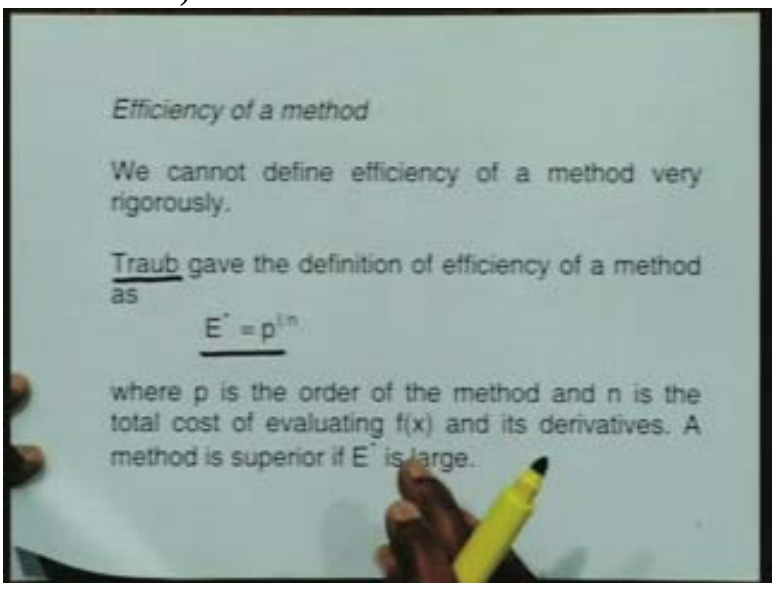
Now we would like to look at what happens to the constant. The constant should be equal so that means I should have A is equal to C into the power of minus one upon p. So I should have here A is equal to C A to the power of minus one upon p. Now bring everything to the left hand side i.e. A to the power of one plus one upon p is equal to C and I want A in terms of C. So we would write this as A into one plus p by p is equal to C or A is equal to C minus p upon p plus one. Therefore we have determined both the error constant as well as the order of the method. So the error constant in the secant method was C and this is the error constant for the method that we are defining. So we have defined the method as epsilon k plus one is A into epsilon k to the power of 1.618. So in this A is given by C, where C was defined as half of f double dash by xi by f dash xi. So this is the relationship that I can get here. Now I can similarly obtain the order of convergence of all the remaining methods. I will be leaving it as an exercise for you.

(Refer Slide Time: 00:09:04 min)



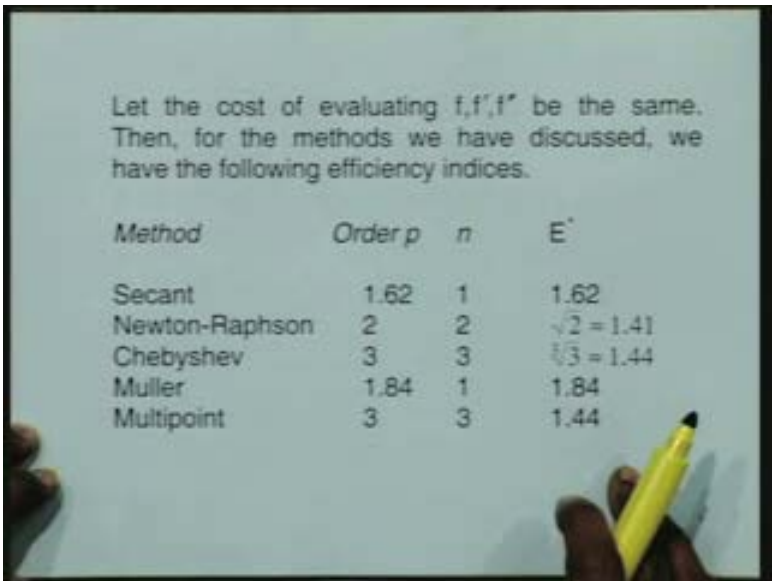
Now the order of the Chebyshev method and the multipoint methods are three. If you remember we have derived two multipoint methods. The one multipoint method was using one function evaluation and two derivative evaluations. The other multipoint method was using two function evaluations and one derivative evaluation. Depending on the cost of the evaluation of the function or the derivative, we can choose the proper multipoint method to find the solution in the problem. Now the order in the Muller method is 1.84. Now so far when we talked of the cost of these methods we were counting the major cost of the function evaluation. The other parts like multiplication, divisions are too trivial or too less. Therefore the major cost is only evaluation. In that sense Chebyshev and multipoint methods require three evaluations. Newton Raphson method requires two evaluations. Muller and secant method require one function evaluation.

(Refer Slide Time: 00:10:15 min)



Now attempts were made to compare these methods to look at it in different perspectives and one particular perspective was what they called as efficiency of a method. It is a very simple definition which is not rigorous. Based strictly on this we cannot say whether a particular method is good or bad but there is a way of looking at defining the efficiency of the method. It simply defines that you take p, order the method, and count the total number of evaluations n, which means function as well derivative evaluations, and then take the root of that which is p to the power of one by n and that shall be called as efficiency of the method. If this quantity is big, of course it is, instead of taking this you can take logarithm. It is one and the same. So if this number is large then we say that the method is better but I should say that this is not a rigorous way of approaching. It is only way of looking at the looking at these methods. And this was given by another person called Traub.

(Refer Slide Time: 00:11:20 min)



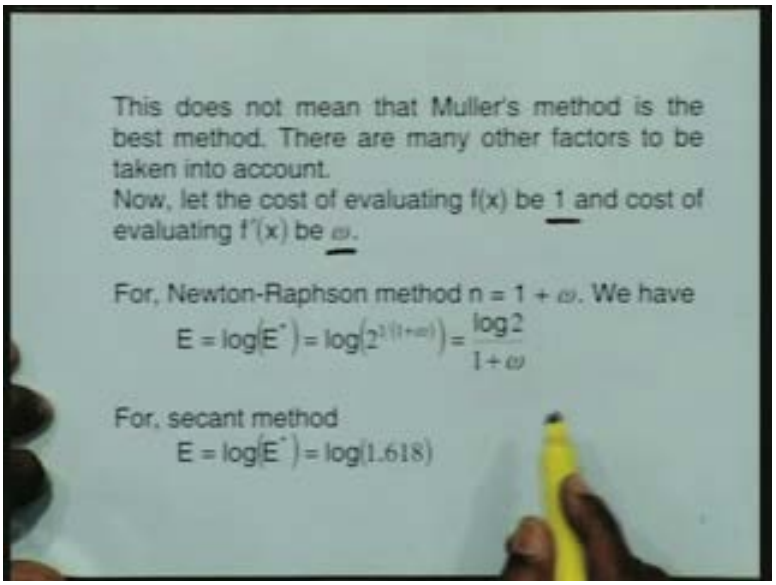
Now let us just put in a tabular form what we have done so far in the methods. So we shall assume that the cost of evaluating f, f dash, f double dash is the same. So that means the computation cost of derivative evaluation, second derivative evaluation and function is the same. Then we have this particular table of values. If you look at the secant method it has only one function evaluation and you have shown that the order is 1.618 i.e. approximately 1.62. So I have taken the nth root of 1.62. The Newton Raphson method has two evaluations; one function and one derivative, so we will take the total as one plus one. The order is two. Therefore efficiency is the second root or the square root of two which will be simply 1.41. In the Chebyshev method we have one function, one first derivative and one second derivative evaluation. So the total is three evaluations. The order is p, so the efficiency index is cube root of three. This is n, this is cube root of three i.e. 1.44.

In the Muller method there is only one function evaluation. The order of the method is 1.84. Therefore the efficiency index is1.84.The multipoint also has got three evaluations either one function and two derivatives or two derivatives and one function evaluation. So again we have cube root of three which is equal to 1.44. Now from the view point of Traub it looks like Muller

method is the best because it has the best efficiency index but however if you look at the total computation the other methods would also score, because there are other factors besides just looking at this particular way. Therefore as I said this is one way of looking at it but these do not say that the Muller is the best method for computation.

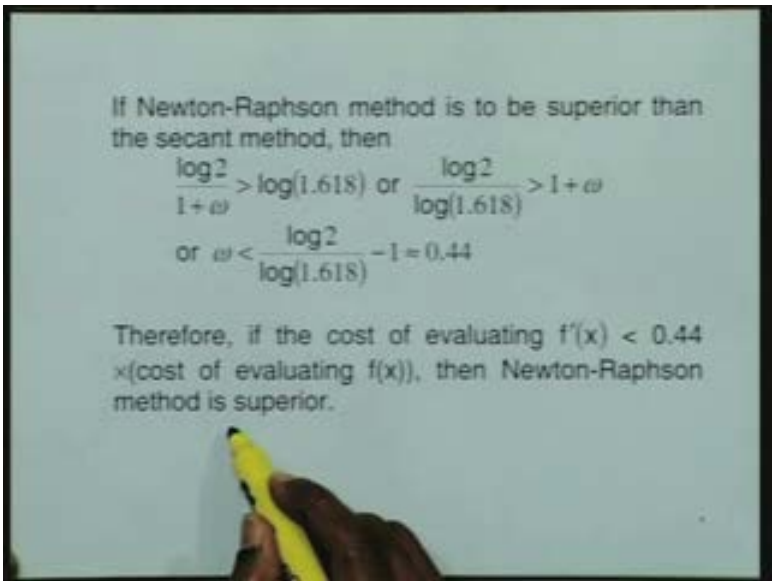
Now for example whatever we have discussed here can be looked at a slightly different way. We have said the cost of evaluating f, f dash and f double dash are the same. Suppose the cost of the evaluating derivative is less or the cost of derivative is more, then what we are talking about will become different.

(Refer Slide Time: 00:13:40 min)



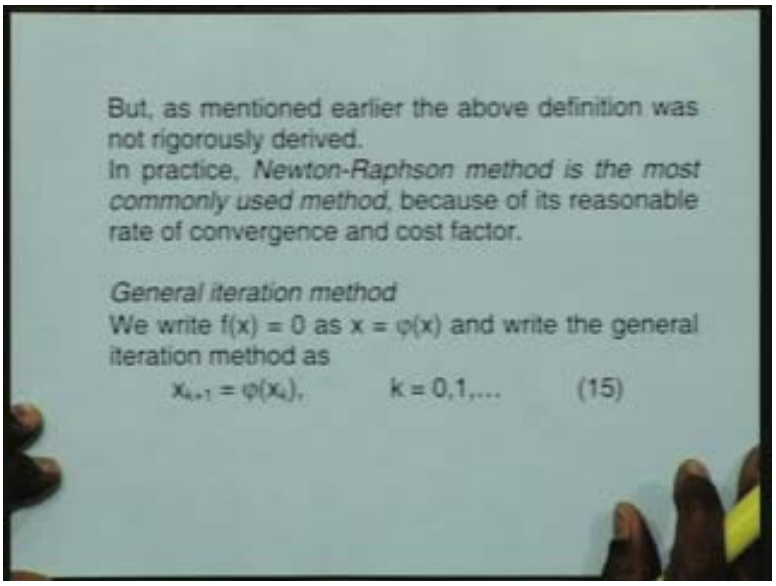
Now let us suppose for f and f dash the evaluation of the cost is different. So let us take the cost of evaluating as 1 and the cost of evaluating f dash x as omega. So we can always find the ratio and then normalize the evaluation of fx to 1 and then evaluation of f dash x will be some omega. For Newton Raphson method n will be equal to one plus omega, total cost i.e. evaluation of fx and f prime one plus omega. Now let us take the logarithm on both sides because it is easier to look at that because the power now is a fraction therefore we need to take a logarithm. So if I take the logarithm of this we will have logarithm of E^* i.e. logarithm of nth root of two i.e. n is one plus omega two to the power of one upon one plus omega. I can write this as logarithm of two divided by one plus omega. If I look at the same quantity for secant method, secant method has no evaluation of f prime, therefore whatever we have got earlier would remain as it is; logarithm of E^* is logarithm of 1.618.

(Refer Slide Time: 00:15:16 min)



Now what I am trying to do here is to see when the Newton Raphson method will be better than secant method. If I can say the cost of fx is something, cost of f dash x is something, which one do you think would be superior, Newton Raphson or secant method or under what circumstances the Newton Raphson method would be superior. Now if the Newton Raphson method is to be superior then this quantity should be greater than this quantity because the efficiency index would be bigger. So we should have logarithm of two divided by one plus omega should be greater than logarithm of 1.618. Let us simplify this. I bring logarithm of two this side; this is greater than one plus omega or omega is less than this quantity minus one. So if I just evaluate this I get it as 0.44. Therefore from this we can conclude that if the cost of evaluating f dash x is less than 0.44 into cost of evaluating fx (earlier we normalized fx cost as one and cost of evaluating f dash x as omega), then Newton Raphson method is superior. As I said this is only a way of looking at it. If you find that the first derivative and second derivative can evaluate easily, Chebyshev method would definitely score because it is a third order method and we have seen what we mean by the order of a method. The error is in the order of three i.e. epsilon k cubed and if at any time you are within the bound i.e. once the interval in which the root lies (your error epsilon k) is less than one then the convergence is going to be very fast.

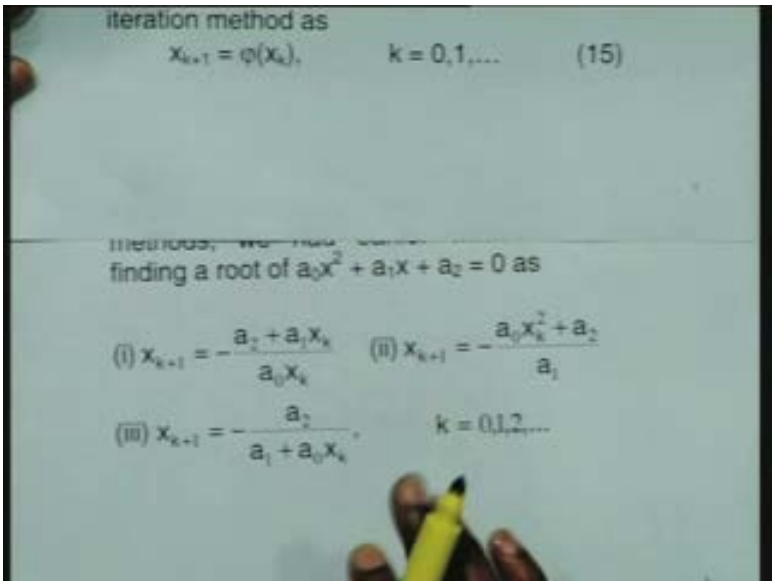
(Refer Slide Time: 00:16:54 min)



Now however in many practical applications because the second derivative evaluation may be very difficult, the Newton Raphson method was found out to be the most popular method because of the reasonable value for p which is the second order quadratic convergence and the cost factor. The cost factor is one function evaluation and one derivative evaluation. You can see that even today in many of the research problems wherever you need the roots they would go to Newton Raphson method than opting for other methods. Of course when the derivative is just not possible to be found then they go to the secant method or they opt for further modifications of Newton Raphson method in which we can only use fx and not derivative at all.

Now there is yet another concept of defining a numerical method for obtaining the simple root of an equation and that is called the general iteration method. Here we take the equation fx is equal to zero and solve it as x is equal to fx. There can be infinite number of possibilities. Given a method fx is equal to zero you can write it as x is equal to fx. So any way you can write down and write the general methods straight away as the xk plus one is phi of xk.

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

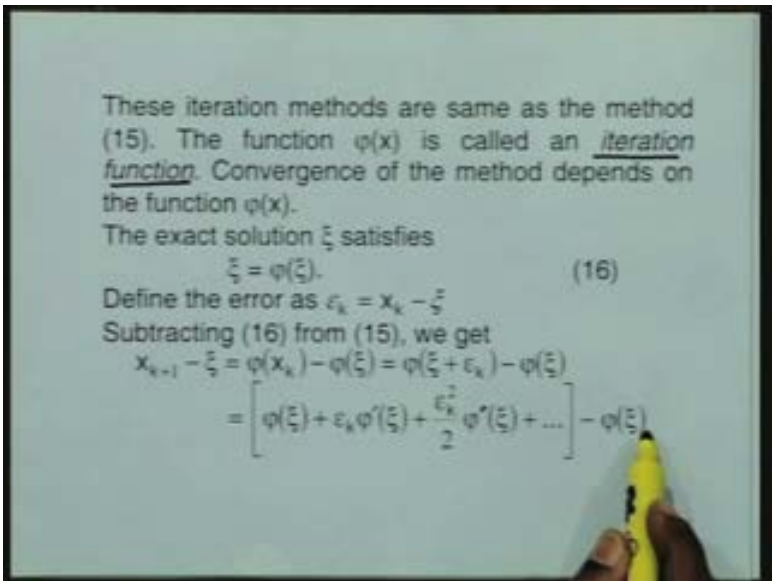


Now if you remember in the second lecture when we started defining an iterative method we took a simple example of: a_0x square a_1x plus a_2 is equal to zero and we then said that we can write an iterative method for solving this in any of these formats. We may just look at what we have really done here. We have taken a_2 plus a_1x to the right hand side, divided by a_0 xk and written a method like this. Alternatively we said that I can retain the middle term here and then take these two terms to the right hand side. A division would give me xk plus one and similarly in the third case we have taken a_2 to the right hand side. We have taken common factor x between these two and then divided it out and brought a method. So these are all forms of xk plus one is equal to phi xk which we can call it as a general iteration method.

However it will be, as I said there are almost infinite number of ways that one can drag this, because one can add x, subtract x if x is not available, so it is a way of writing it. Therefore it is necessary that we start the iterative procedure to look at whether our choice of phi xk is correct or not. If the choice of phi xk is correct, divergence will be there and if the choice is correct the convergence will be there. So let us first of all look at what is the condition on phi x such that the iteration converges.

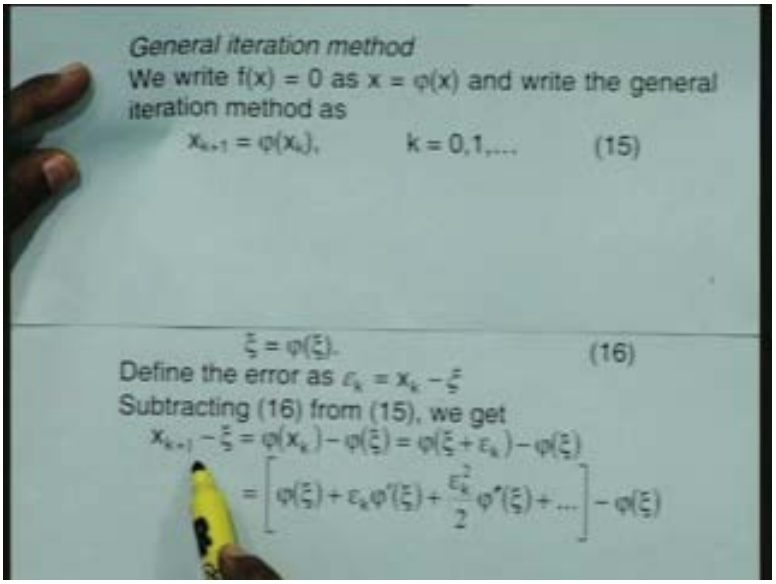
We shall call this phi x which we are using as an iteration function. So everything depends on this iteration function. The convergence of the method depends on the function fx. Whatever we are discussing about the convergence is identically same as what we have done for finding the convergence of other numerical methods.

(Refer Slide Time: 00:19:39 min)



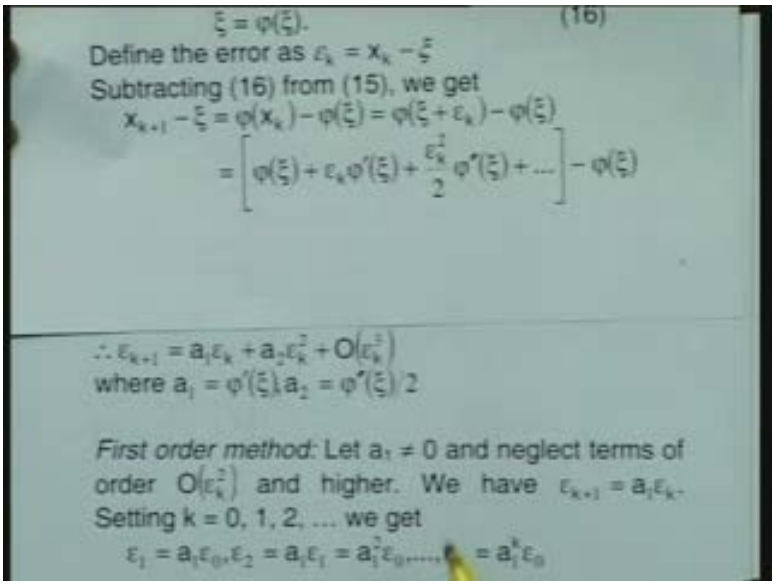
So we define the error as epsilon k is xk minus xi. The given equation is x is equal to phi x. So exact solution satisfies xi is equal to phi of xi. So from this I will substitute this definition here so that I will have here xk plus one minus xi.

(Refer Slide Time: 00:20:48 min)



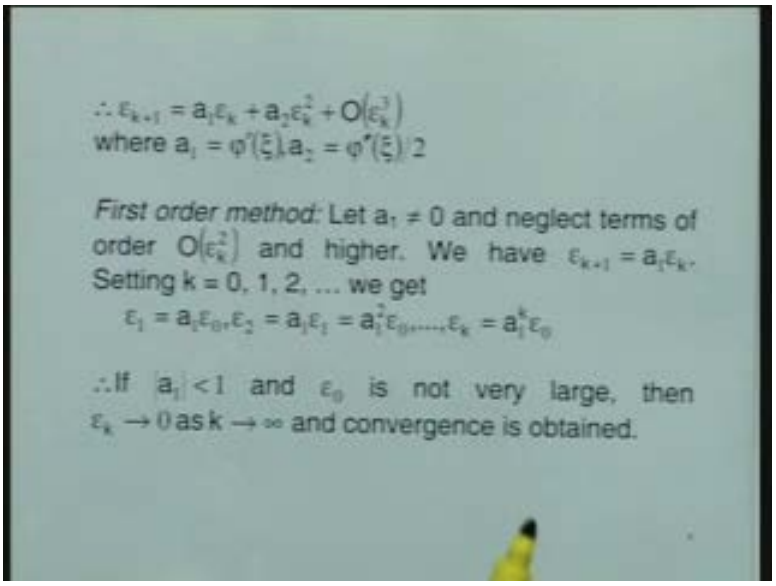
I am substituting this definition into the method that we have just defined, which is - xk plus one is equal to phi of xk. We take this and take this. Subtract these two; xk plus one minus xi on the left hand side and on the right hand side phi of xk minus phi of xi. I have subtracted these two to get xk one minus xi is equal to phi of xk minus phi of xi.

(Refer Slide Time: 00:21:29 min)



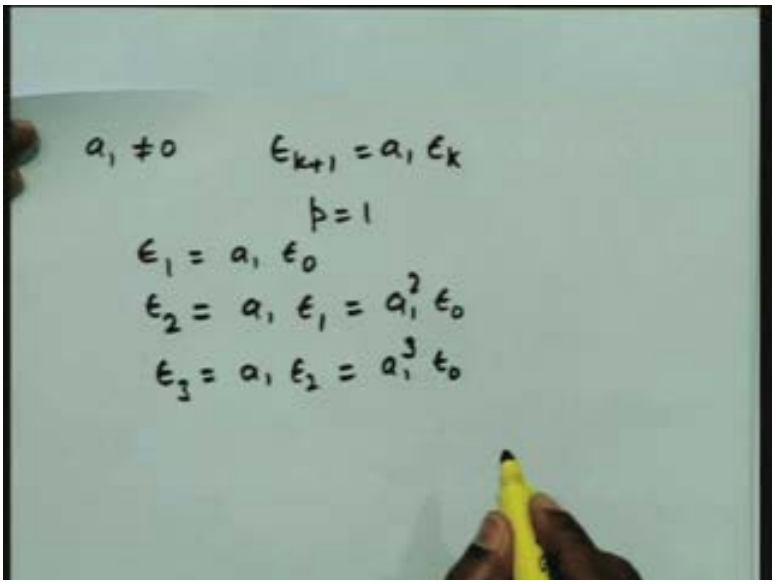
Now I will use the Taylor series expansions to find out what is the condition on phi x, such that this method converges. So let us open it up; phi of xi plus epsilon k by Taylor series; phi of xi epsilon k phi dash of xi; half epsilon k square phi double dash of xi plus or minus xi. Now we do not have to write down more terms than this. Now phi of xi and phi of xi cancels here and therefore what is left out is simply this term and this term. Furthermore xi is equal to phi of xi and the exact solution satisfies phi of xi. Now phi of xi has already cancelled. Now this definition of xk plus one minus xi is epsilon k plus one. The right hand side is given as some a_1 into epsilon k, a_2 into epsilon k square. So I have rewritten this particular step as epsilon k plus one on the left hand side; a_1 into epsilon k, a_2 into epsilon k square plus order of epsilon k cubed, where a_1 is this first term phi dash of xi, a_2 is the second term phi double dash by two.

(Refer Slide Time: 00:22:36 min)



Therefore in the case of this general iteration method that we have described in this particular way, I can write down the error on the current step as equal to a_1 epsilon k plus a_2 epsilon k square plus order of epsilon k cubed. Now if a_1 is not equal to zero in a method then the error can be written as simply as follows.

(Refer Slide Time: 00:24:05 min)

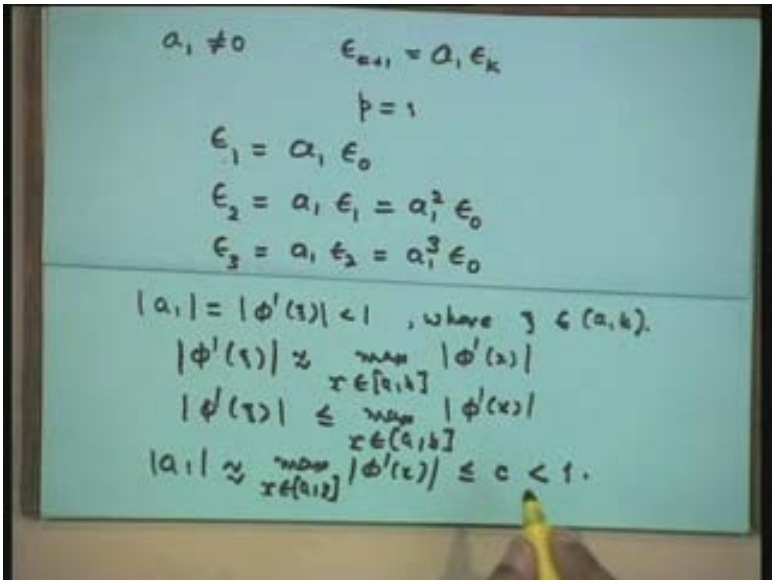


In this case for example, if you have a_1 is not equal to zero then I can drop all these terms and write this as epsilon k plus one is equal to a_1 into epsilon k. In this case p is equal to one and this is epsilon p to the power of one. Therefore the order of the method is one. Therefore when a_1 is not equal to zero I would get a first order method because in that case epsilon k plus one is equal

to a_1 epsilon k i.e. epsilon k plus one is a_1 epsilon k. Therefore the order of the method will be one.

Now here let us put k is equal to zero. So I will have epsilon one is equal to a_1 epsilon zero. I put k is equal to zero. Then let us put k is equal to one, therefore I will have epsilon one, a_1 one epsilon one but epsilon one is a_1 square $epsilon_0$. Now I proceed, epsilon three is a_1 epsilon two. Therefore this is a_1 cubed epsilon zero. Therefore if I proceed on like this what I would get is epsilon k is equal to a_1 to the power of k $epsilon_0$. So we repeatedly do this and then I produce this epsilon k is a_1 one k $epsilon_0$. Now we can see that if this method is to converge $epsilon_0$ in the initial error, the error in the initial approximation is to decay i.e. goes to zero then epsilon k should tend to zero. The only possibility is, as k tends to infinity, this has to go to zero means; magnitude of a_1 should be less than one. Therefore if magnitude of a_1 one is less than one and $epsilon_0$ is not large, (of course it is always finite quantity, even if it is two or five or ten it does not make a difference), then epsilon k will tend to zero as k tends to infinity and convergence is obtained.

(Refer Slide Time: 00:26:46 min)

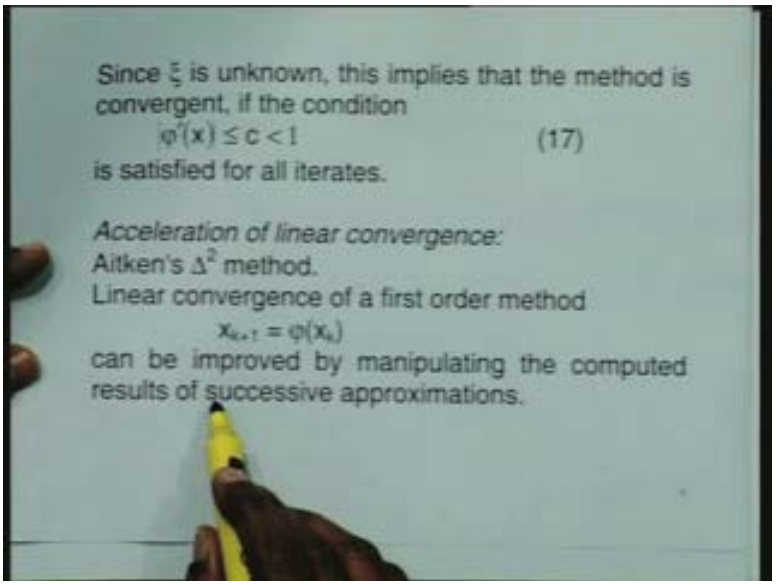


Therefore the condition for the method to converge is magnitude of a_1 should be less than one. Therefore we require magnitude of a_1 is equal to magnitude of phi dash xi is less than one, where xi is any value lying in the open interval ab. However xi is an unknown quantity. Therefore we shall approximate the magnitude of phi dash of xi by the maximum magnitude of phi dash x in the closed interval ab. This means we will approximate phi dash of xi by the maximum magnitude of phi dash of x in the closed interval ab. This approximation is valid because xi is any value in ab and hence phi dash of xi in magnitude is always less than or equal to maximum magnitude of phi dash of x. i.e. phi dash of xi is always less than or equal to maximum of phi dash of x; x lying in the interval ab. Therefore we shall require that magnitude of a_1 is approximately equal to maximum of phi dash of x. x lying in interval ab is strictly less than or equal to a finite value c less than one. Hence the iteration function should satisfy this condition for all iterates, so that the iteration sequence converges otherwise it will diverge.

In all the numerical method that we have, we have to do many experiments to see whether our solution is correct or not. If you look at the solution of an ordinary or partial differential equation or a system of ordinary partial differential equation, whose solution is not known then we are solving it with the particular step length. We have to produce some solutions. We have to be sure that what we have produced is not junk. We have to say that this is the approximate solution. So what we normally do is, we repeat the same computation with another step length and produce new solutions.

Now if these two solutions are reasonably close to each other or if they are not reasonably close to each other what can we do. Suppose you want to solve this by even smaller step length. Now we have already made some expenses or some computations, so does this computation that we have already done go waste; is it possible for us to manipulate or use this result to get a higher order result or a better result? The answer is yes. None of these values that we have obtained in this numerical method needs to be put in a waste basket. We can use them to produce much better results.

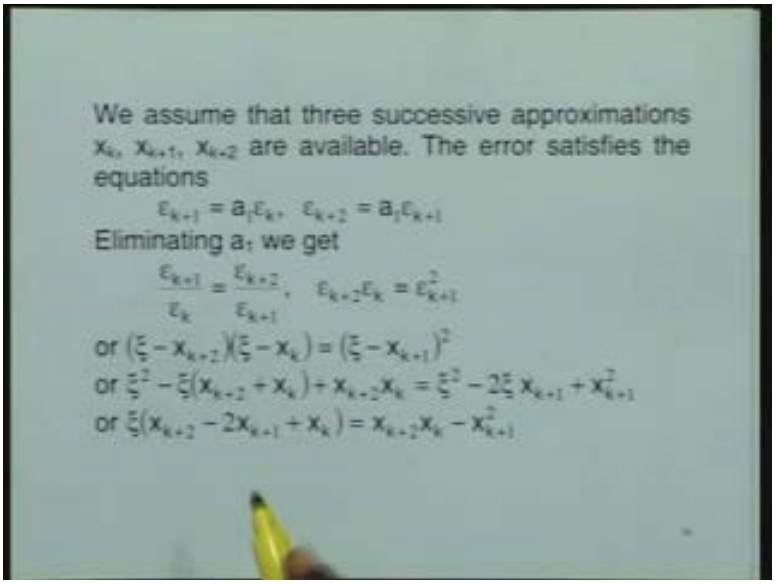
(Refer Slide Time: 00:28:23 min)



The concept lies in what we call as acceleration of linear convergence. Such a concept is called extrapolation which we shall do later on. Both in numerical differentiation and numerical integration, everywhere we use the concept of extrapolation. So the idea of extrapolation comes from what we are talking about.We shall call this as acceleration of linear convergence.

Now we start our method xk plus one is equal to phi xk. Now what we are saying is that we want to manipulate the computed result of successive approximation to produce a better result. That means I would like to produce a new value by manipulating, such that the new value is obtained as if it has been obtained from a higher order method. In this case for example it is obtained from a first order method but it is as if it is obtained from a second order method. We do it by a very simple manipulation.

(Refer Slide Time: 00:30:58 min)



Let us assume that three approximations are available for us which are xk, xk plus one and xk plus two. Starting with this we have x_0, x_1 and x_2. So we have computed with a first order method x_0, x_1 and x_2. What will be the error? By definition error looks like epsilon k plus one is a_1 epsilon k and epsilon k plus two is equal to a_1 one epsilon k plus one. This is how the error looks like.

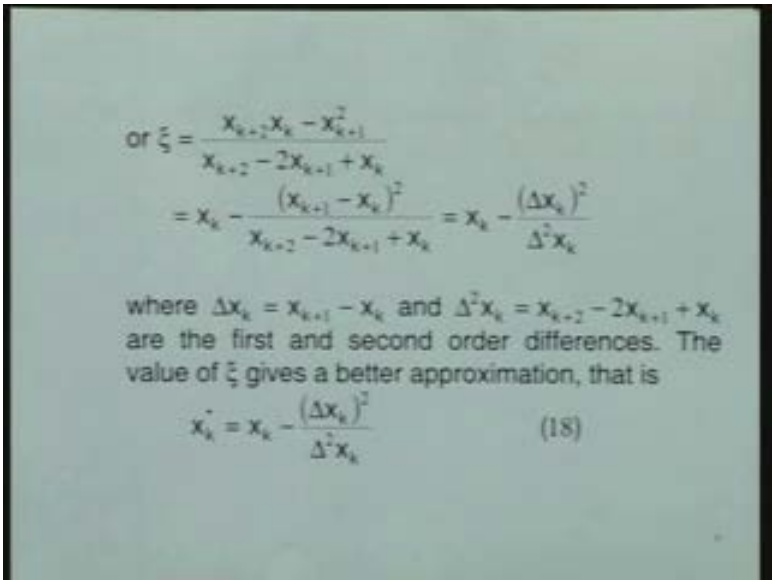
Now if the method is to behave like a second order method the error term should be epsilon k plus two is equal to some constant epsilon k square, that means a first order term should not appear in this. So what we do now is we just divide these two expressions. So I divide it out, say epsilon k plus one divided by epsilon k. I am equating a_1; epsilon k plus one divide by epsilon k is equal to epsilon k plus two divided by epsilon k plus one. That means we have equated a_1 on both sides.

Now cross multiply this; epsilon k plus two into epsilon k is epsilon k plus one square. From here I would like to retrieve a numerical method. Work backwards by using the definition of epsilon k, epsilon k plus one, epsilon k plus two. This numerical method whatever is retrieved from here is obtained by eliminating the coefficient of epsilon k i.e. the first order term epsilon k. Therefore the numerical method that we have here, when expanded in Taylor series will be starting with epsilon k square because a_1 has been eliminated. So whatever method I retrieve from here would be as if it is a second order method.

Now let us simplify this. I am substituting epsilon k plus two as xi minus epsilon k plus two and epsilon k is xi minus epsilon k. This is equal to epsilon k plus one whole square i.e. xi epsilon minus xk plus one whole square. So we are just substituting the definitions of this. Let us open it up. This is xi square; xi xk plus two plus xk, plus xk plus two into xk; on the right hand side let us take the square of this - xi square minus two xi xk plus one plus xk plus one square. Now we can see xi square cancels both sides and therefore I will bring this xi to this side and this xk plus

one to the left hand side. So I will have here xi and xk plus two plus xk, these are the two terms minus two times xk plus one. On the right hand side we will have xk plus one into xk minus xk plus whole square. So I am able to get xi alone, because xi square has been cancelled.

(Refer Slide Time: 00:32:38 min)

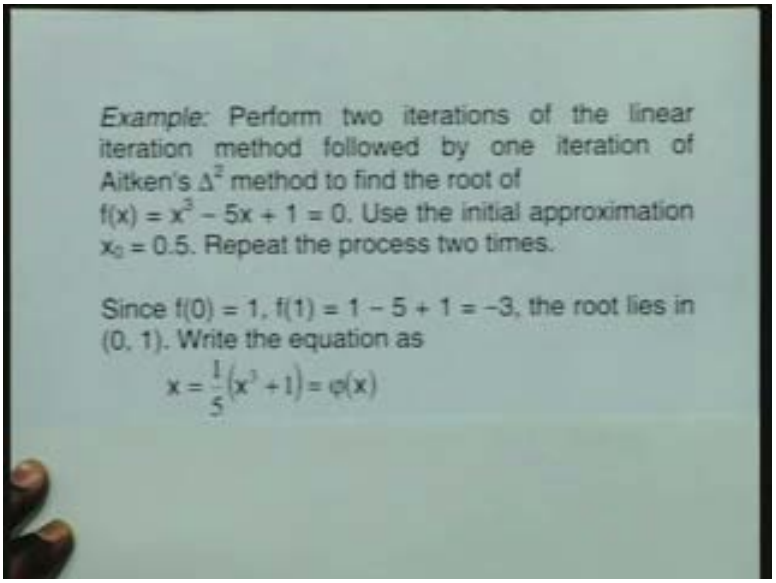


From this I will now write down my next approximation. Now from here I will take this to this side and write this as xk plus two minus xk minus xk plus one whole square divide by xk plus two minus twice xk plus one plus xk. In all the methods we are writing xk plus one is equal to xk plus something. So I would like to rewrite this in the format as xk minus some quantity (gx or some quantity). So I can simplify this and write it as xk minus xk plus one minus xk whole square divided by the same denominator xk plus two minus twice xk plus one plus xk. Now we are going to define in our later lectures the meaning of the quantities that we have written here, delta of xk, delta square of xk. I denote this delta xk as xk plus one minus xk and I denote the denominator by delta squared xk i.e. definition of delta squared xk, xk plus two minus twice xk plus one plus xk and these are called the differences. They are called the first and second order differences which we are going to define later on. For the moment let us take it as first and second order difference. Therefore I will take the new approximation to xi as x^*k i.e. xk minus delta xk whole square by delta squared xk.

Now if I want to use this computation, I need three values first to start it; which are xk, xk plus one and xk plus two. Once I have three successive approximations I can now build up a new value called x^*k which will behave as if the result has been obtained from a second order method because as I explained earlier we have eliminated a_1. In the Taylor expansion it was containing epsilon k plus x one a_1 epsilon k plus a_2 epsilon k square and so on. If a_1 is eliminated the resultant is epsilon k plus one is equal to some c into epsilon k square some instead of a_2. Therefore this is a second order method. It will behave like a result as if it has been obtained by this one.

How do you implement computationally? We start with x_0, x_1 and x_2 and then we compute this. Now this is a new better estimate. Now I will use now this x_3 as my starting point; x_4 and x_5 I will compute again for the first order method. Now I have got x_3, x_4 and x_5. Now I go back to this method and compute it. So after every three values i.e. two new values and the value that we have here, with three of these we will now go on with this one. So each of these results will be as if it is from a second order method which means somewhere in between, we are doing the midcourse correction to make it a better estimate or better value.

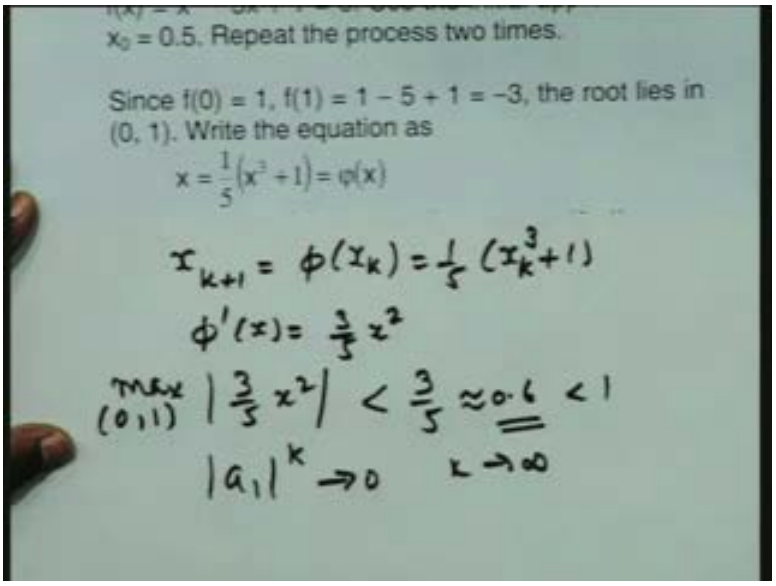
(Refer Slide Time: 00:35:47 min)



Now let us take an example on this. Now we will perform two iterations of the linear iteration method followed by one iteration of the Aitken's delta square method to find the root of x cubed minus phi x plus one is zero. And we shall use the initial approximation x_0 as 0.5 and we shall repeat this process two times which means we start with x_0, x_1, x_2 and then I use these three values to get the next estimate and then go on repeating the same thing two times. Now for this I need the values of f_0 which is one, f (1) is minus three, so that I am first estimating where the root lies. The root lies in zero and one.

Now the first thing that we will have to do is, we have to construct our method from here. What we have done is, I have taken here x cubed plus one to the right hand side, taken x as the middle term here and written it as x is one upon five, x cube plus one. From here if I write the method would be xk plus one is equal to phi of xk i.e. one upon five xk cubed plus one. Now before we proceed I would like to test whether our phi x iteration function is satisfying the convergence criteria or not. Therefore I need to check whether phi dash x in the interval in which the root lie is less than one or not.

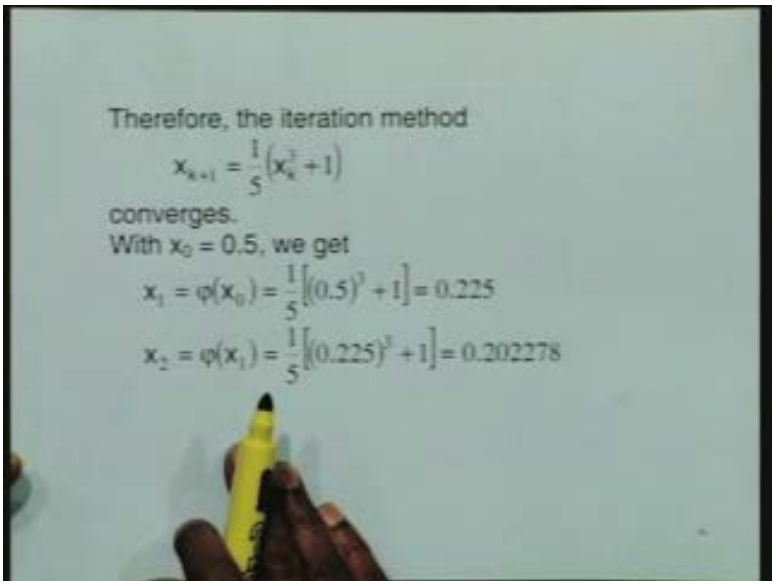
(Refer Slide Time: 00:38:24 min)



Now I can differentiate phi and get phi dash x; phi dash x will be equal to three by five x square. Now I need phi dash of x, three by five x squared in magnitude maximum; the interval in which the root lies is zero, one. So I want the maximum of phi dash of x as this and this obviously is less than three by five which is approximately 0.6. x is lying between zero and one therefore it is strictly less than one. Therefore this iteration is going to converge and is going to converge quite fast. The reason being if the magnitude of a particular number that we have taken, in this case magnitude of a_1 to the power of k tends to zero and k tends to infinity, it will go to zero very fast provided this a_1 is small. Now if this quantity was 0.1 then we can achieve convergence very fast. Now this quantity 0.6 is not bad, it is quite good because the square of this is going to be 0.36 and it is going to be reduced quite fast. Therefore this is an indicator as to how fast our method is going to converge.

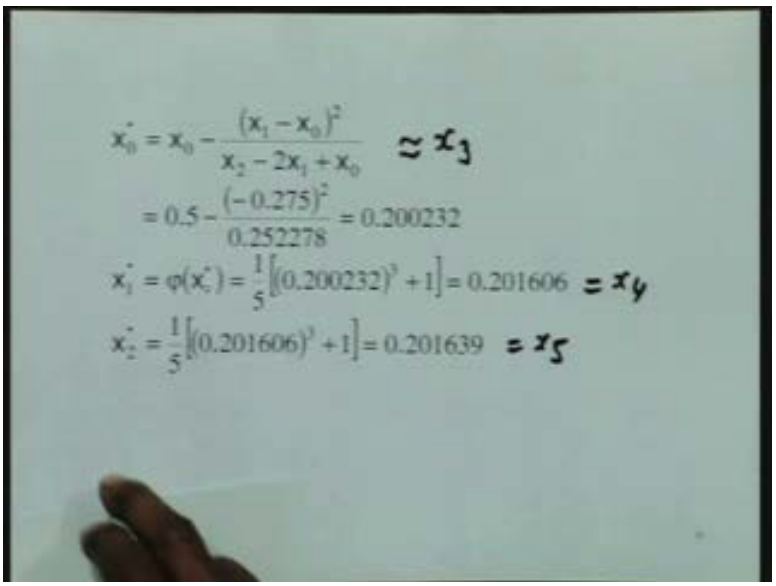
Now let us just compute these values. Now I would just get the values. Starting with x_0 given to us is 0.5. Now I would get x_1. Just substitute phi of x_0 is equal to 0; phi of x_0 i.e. one by five, point five whole cube plus one. Just simplify and you get 0.225. I then find phi of x_1 and call it as x_2 and that is one upon five. This is 0.225 whole cube plus one i.e. equal to 0.202278. Now at this stage the value of x_0, x_1 and x_2 are available for me. Now I can use Aitken's method to get a better estimate of the computation and I will be using what we called as x_0^*. You see, you can indeed call it as x_3, you need not call it x_0^*. You could as well call it as x_3. In continuation we can call it as x_3 also.

(Refer Slide Time: 00:39:02 min)



Go backwards and substitute the values of k. Now in this I am substituting k is zero. So this is x_0 x_1 minus x_0 x_2 minus two x_1 plus x_0. So that is what I have written here. So this is the approximation. Now x_0, x_1 and x_2 values are available to us. So let us substitute these values here and simplify to get 0.200232.

(Refer Slide Time: 00:40:57 min)

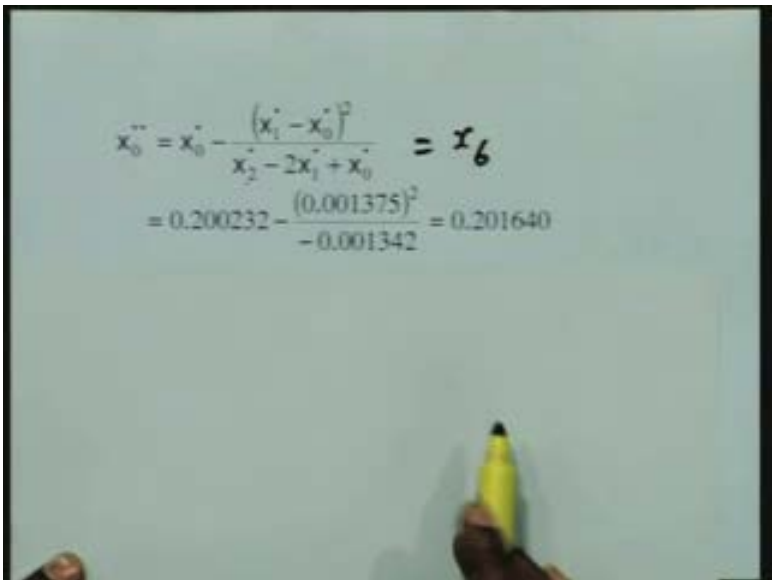


Now what we are claiming is that this value is a value as if it has been obtained from a second order method and not from a first order method. Now we use this and then proceed on to find this. We repeat this; x_1^* is equal to phi of x_0^*. You could as well call this as x_4.You could as well call this as x_5. There is no hard and fast rule that you should define them as such. So we can go on a sequence x_3, x_4 and x_5. We remember that x_3 was the update of the value to the next higher order. These two are obtained from the first order. x_1^* is phi of x_0^*. Now I substitute this, one upon five x cubed plus one. So I have the value of x_4 four as this. Then I proceed further; x_2^* is one by five, 0.201606 whole cubed plus one. So I compute this as x_5.

Now I have the updated value. Now I have computed two values, x_4 and x_5. Now we are ready for next Aitken's acceleration convergence application. So I can now use this x_0^*, x_1^* and x_2^* to use in the method. Now what is the current estimate? The current estimate is x_3. Current estimate is this value. Now I should proceed from there, because our iteration method connects only xk plus one and xk. Since x_3 is available for me, I will now use x_3 to get my x_4.

Now whenever three values are available for me I will go for the Aitken's method, because it needs three values to have an update. So once I have x_3, I will compute two values of x_4 and x_5. Now i got x_3, x_4 and x_5. Now what you are saying is that why not we take x_2, x_3 and x_4. No we cannot. The reason is the previous value is a first order, this is a second order and this is a first order and that will be a miss match. Therefore we will start from this value only and then go ahead for two steps; x_4 and x_5 is a second order result. These two are first order result. Now I manipulate these three to get a second order result.

(Refer Slide Time: 00:43:06 min)



Therefore you could as well call it as x_6. So the next value is x_5 and x_6. So this will be $x_0^* x_1^*$ minus x_0^* square x_2^* minus twice x_0^* plus x_0^* and now we just substitute these three values to make the computation. So I can compute this and get details; 0.200232 minus 0.201375 whole square by minus 0.001342 is equal to this value.

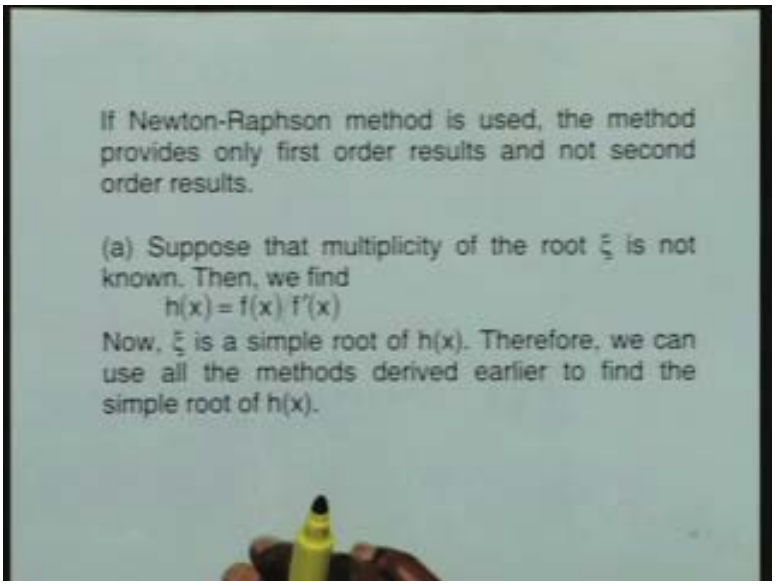
Therefore whenever we have a method and we can analyze the method and get the error expression for it, it is possible for us to manipulate the computed results or to get the results computed which is of higher order or superior to what we have.

We will see later why such a technique is to be used. If you try to find the derivate of a function which defines a table of values; now that means we have got a table which represents a function fx. But I want slope of the curve at any particular point. We do not know if the slope exists or not. We do not know how well the curve behaves for the function. Therefore finding the derivative could be an unstable process. Therefore in such cases what we do is we will try to find a value for the derivative using a very lower order method which will always work. Then we would like to manipulate or extrapolate as we have done here to show as if you have getting from a higher order method.

Now why manipulation should be done is because as you have discussed earlier, suppose you have a first order method and error is of the order of c into epsilon to the power of one. In that case each iteration will give you at the most one decimal place; that means if the error is 0.6, we can expect 0.5 or 0.56 and the next one we could expect at the most to be 0.47. Therefore you need hundreds of iterations or may be thousands of iteration to arrive at an accuracy of ten to the power of minus four or ten to power of minus x, which is too expensive. Therefore if you are able to jump from a first order to higher order method at some stage or still a higher order method, then you are avoiding few hundreds or few thousands of iterations to get the required result. That is why we need this kind of manipulation. The second comment is that most of the results that we are getting here, they are not useless even though the results which are not at all accurate, they can still be manipulated to arrive at much better results.

Now the methods that we have discussed so far can be used only for a simple root. In most of the cases you do not know in a particular problem whether the root is a simple root or a multiple root. And if you are using Newton Raphson method; the Newton Raphson method is supposed to give us the quadratic convergence. So after few iterations you will be able to see how it is going to converge. If you find that these results are not converging quadratically as it should be, that means there is something wrong in the procedure and not with the method. There is something wrong in the assumption that it is a simple root; may be it is a multiple root. If it is a multiple root, the accuracy of all the methods that we have used falls down. The Newton Raphson method would no longer behave as a quadratic method, it will behave like a first order method. In the secant method also the same thing happens; it is super linear, but it drops still further. So it will go below one or below that and you may not even get one place of accuracy in the secant method. Therefore this thing worsens if the multiplicity the root is still higher. Say it is a degree of three or four or five, then this situation worsens further. Then what is a solution for such case when you have a multiple root and not a simple root? As I said it is visible for you in the computations whether the solution is coming or not or a difficulty is clearly visible for us in the solution that we have obtained here. Therefore we will have to modify the methods that we have done. We are not going to do new methods but we modify the method that we have done, in order to take care of obtaining the multiple roots.

(Refer Slide Time: 00:47:55 min)



Now this particular part I will take it in the next lecture but let me tell the simple application of this. This we have discussed earlier. We defined a simple root and a multiple root multiple root of multiplicity m. If I differentiate that function, that f dash x is equal to zero will have multiplicity m minus one of the same root. Therefore we have shown that if I construct a new function hx is equal to fx upon f dash x, then hx will have a simple root at x is equal to xi. If fx has got a root of multiplicity four, its derivative will have a root of multiplicity three. Therefore its ratio will have a root of multiplicity one; that means it will have a simple root. Since this hx has a simple root, then I can use Newton Raphson or secant or Chebyshev method or any method on hx which will produce me the required results of the same order as you are claiming there. Therefore I can use the secant method, use the Newton Raphson method and all the other methods.

We will stop at this today but we will see why do we need new methods or modify the methods, if I can use this method. We shall show that yes, we can use it but we will have to spend little bit more computer cost; that means we will show that we have to evaluate one more quantity. Newton Raphson method was using only f and f dash but now we will show that we need second derivative also, if you want Newton Raphson method but with this approach. This means the cost of computation will go slightly higher. But then we can definitely use the Newton Raphson method using this particular method.