Numerical Methods and Programming

P. B. Sunil Kumar
Department of Physics
Indian Institute of Technology, Madras

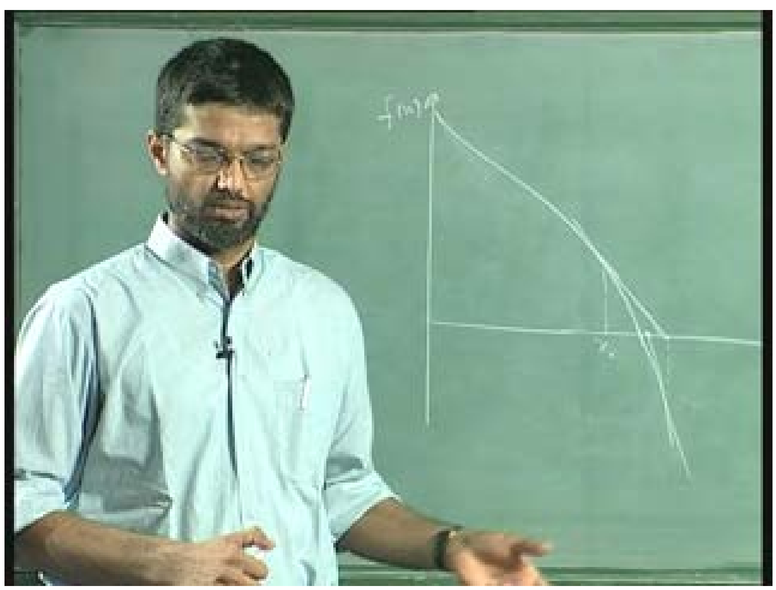
Lecture - 23

Methods for Solving Non-Linear Equations a Comparison

The last class, we were looking at finding the solutions of non linear functions. We looked at many methods, one which need two guess values either side of the 0 of the function or another method which requires only one guess value. So the first method which need two guess values on either side of the function we called the bracketing methods and the ones which need which require only one guess values where Newton Raphson fixed point iteration and also the secant method. So we will continue looked at Newton Raphson today again.

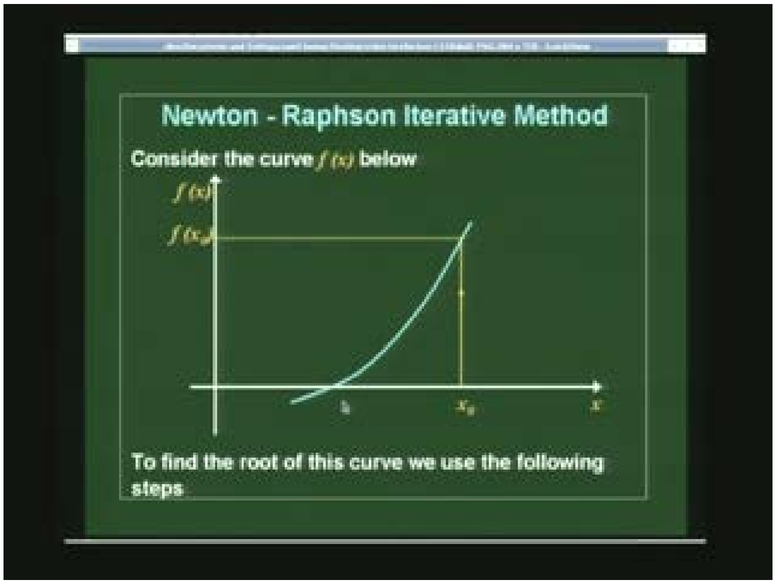
So in the Newton Raphson method, you remember what we are doing was, we had a function, if you have a function of this form that is f of x versus x, we are looking a function of one variable to start with and then we had x axis on this and then we were looking at where the function would cross 0 that is the solution point which we are interested. So one method of doing this actually plot, take any iteration function point for example if I could take redraw this little bit better.

(Refer Slide Time: 03:49)

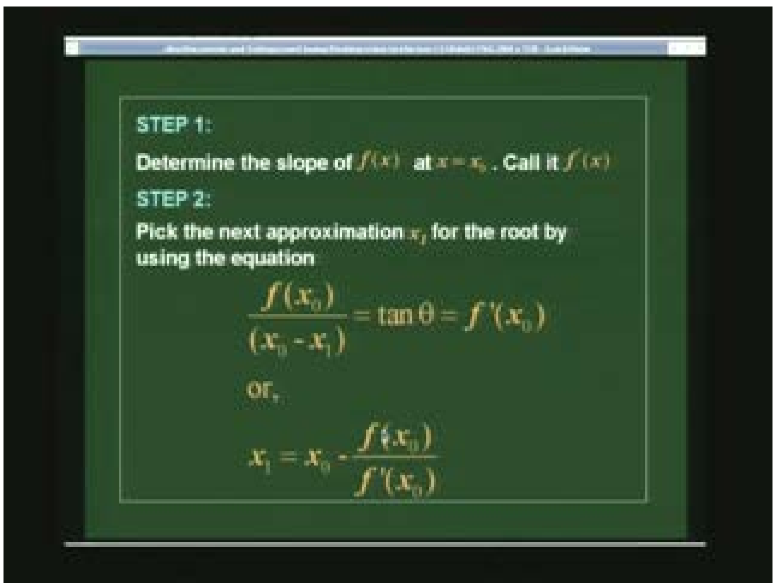


So if you had something like this and then if you had initial guess make an one initial guess in this problem, so that that we called x_0 and then what idea was draw a tangent at that point graphically this is was it means we draw a tangent at that point and look at the point where the tangent crosses the x axis. Okay take that as a next iteration point the x value corresponding to that, okay that was our next iteration point and we drew a tangent at that there okay and then we took as next iteration point etcetera. We this value taken and then we draw a tangent there to get the next iteration point and that is how we continued to get the function that was the summary here, if u have some function which is of this form okay then we make one guess value here I taken as x_0, you draw a tangent at there and take the point which crosses the x axis and then take that as a x value next iteration point, that is next guess value of the next approximation 0 and then you go there and draw the another tangent and then go to the next point x.

(Refer slide time: 03:51)

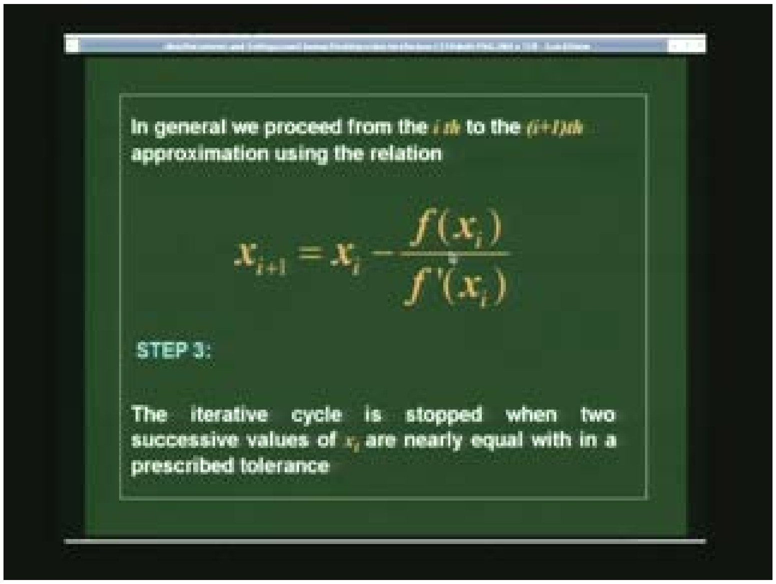


(Refer slide time: 04:26)

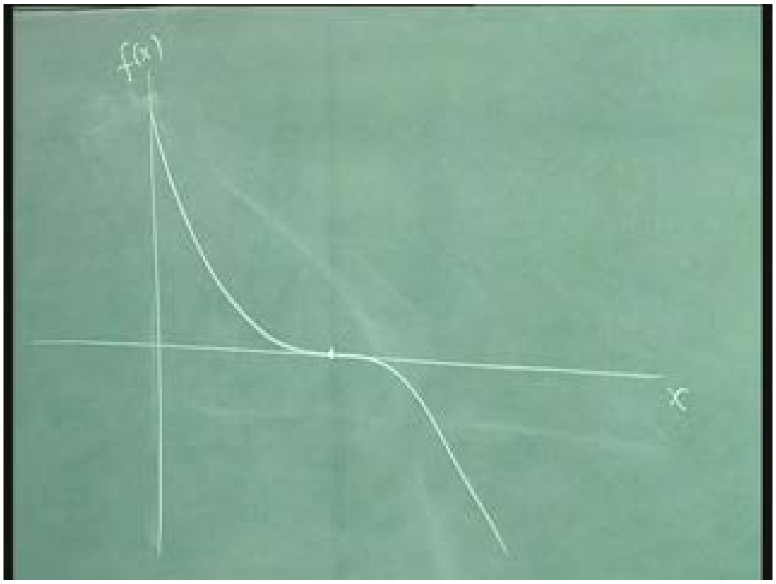


So that is what we looking at we also look at a program implement this and so here, so at every point we had we find derivative which is f prime of x and the idea was to find the point and it which crosses the 0. So 0 that, this tangent meet the x axis that gives the formula for the next iterative point. It is a very simple formula where x_1 equal to x_0 minus f by f prime and then this is easy to implement in a program and also we have to saw that converges very fast now compare to any other methods which we are looked at okay that is the advantage of this method is there one, it converges very fast and second is that it needs only one guess value to be begin with but there is also a problem, the problem being that is by f by the formula contains f by f prime of x. So in cases where the f prime of x tend to go very low this can be lead to large errors.

(Refer slide time: 05:07)

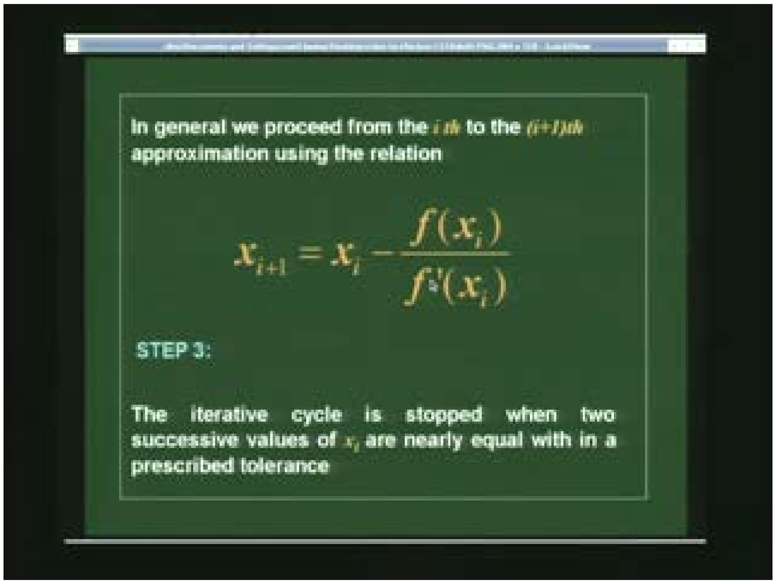


(Refer slide time: 06:05)



For example if I have a case like this, okay so you could have function f of x verses x and we could other function going to 0 something like this. Okay so now that is our 0 crossing point. So that, this in this case, the f prime if x itself it extremely small okay in that cases you could get into trouble using this formula, we will discuss and we will show we will look at particular case where thus such situations arise. So write now let us look at the advantage and disadvantage of this function.

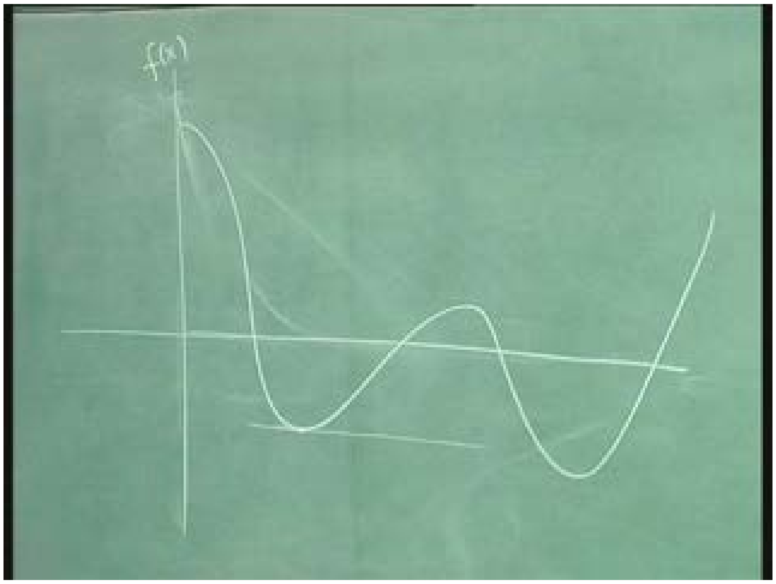
(Refer slide time: 06:07)



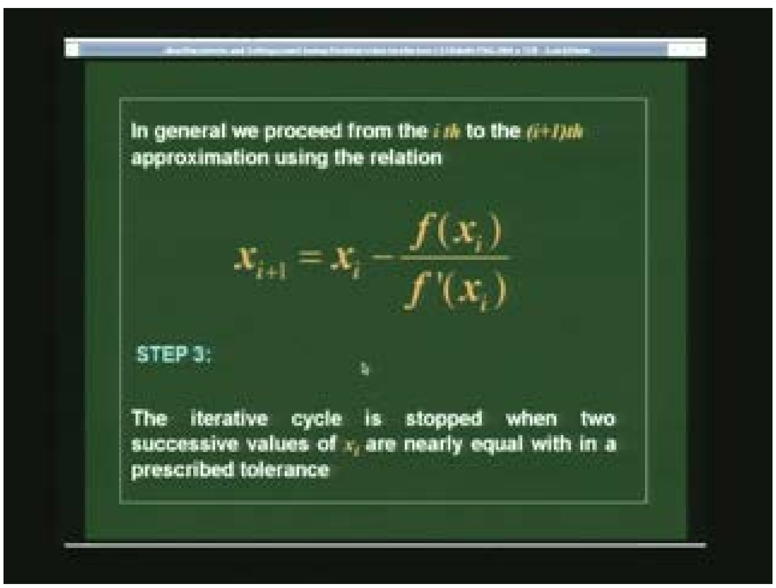
So the advantage being that need only one guess value and then iterate this converges very fast, when it converges okay the case, where it converges it converges pretty fast but that can be cases where it does not converge the reason being that the one of the reason being that the f prime of x go to very small value, an another reason could be another function reason where it need not converges is when you have, when you have functions of a when you have functions of this form and you make this as a guess value, okay this also a problem because the tangent here does not meet the x axis, that can also be a problem and then also we saw cases were they can actually get trap between two trap points.

So there are cases, pathological cases where this thing does not converge. So if it converges and then the convergence criteria being that x_i minus x_i plus 1 minus x_i should divided by x_i minus 1 should below some predetermined error tolerance which we have put. Okay that is the idea basically idea of Newton Raphson tailor expansion that is we say that the function x_i plus 1 is determined from function x_i by tailor expansion and we terminate this at first derivative and we determine that this function x_i plus 1 should be 0 and we determine h, such that x_i plus 1 that is x_i plus h the function value is 0 that is the idea what leads to an very simple formula we saw in the last case, the last lecturer.

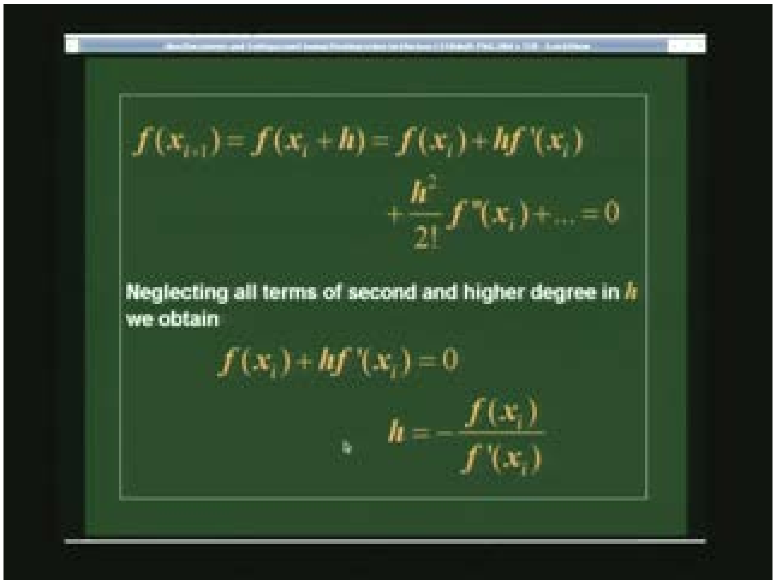
(Refer slide time: 07:09)



(Refer slide time: 07:11)

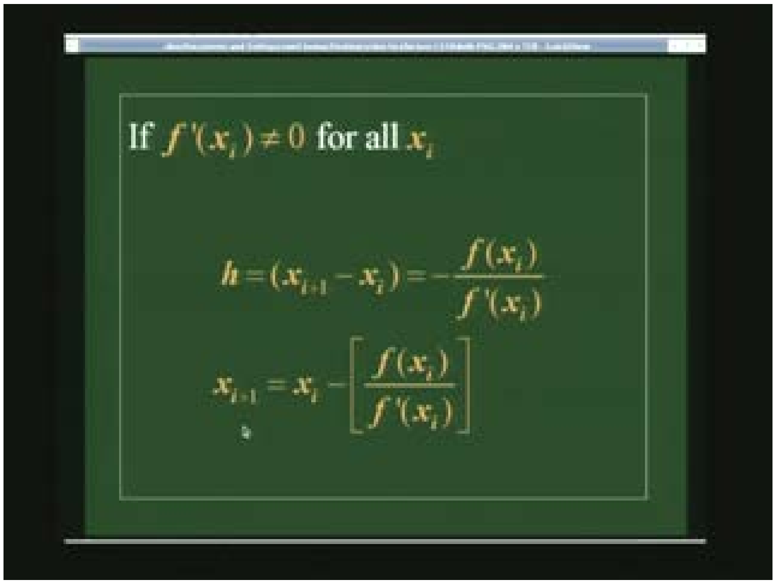


(Refer slide time: 07:36)



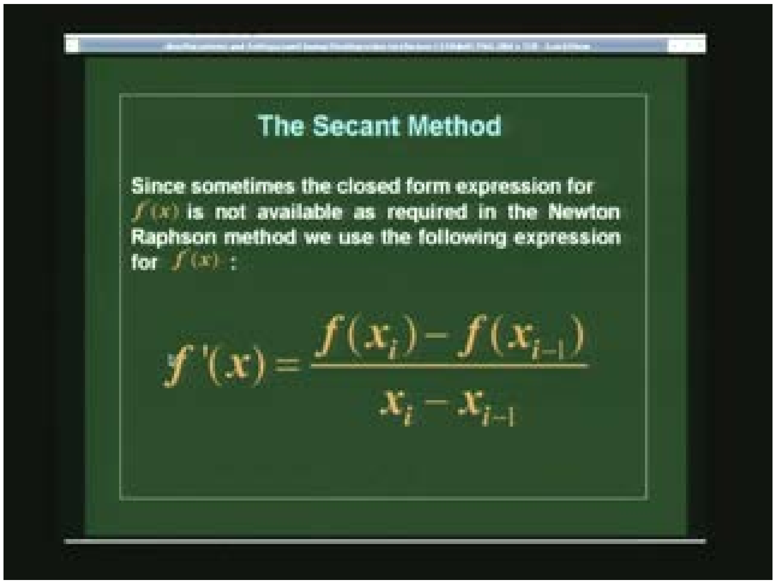
So wherever f prime of x is not equal to 0 then we can write this formula of this form as an iterative formula for the, for the variables.

(Refer slide time: 08:14)



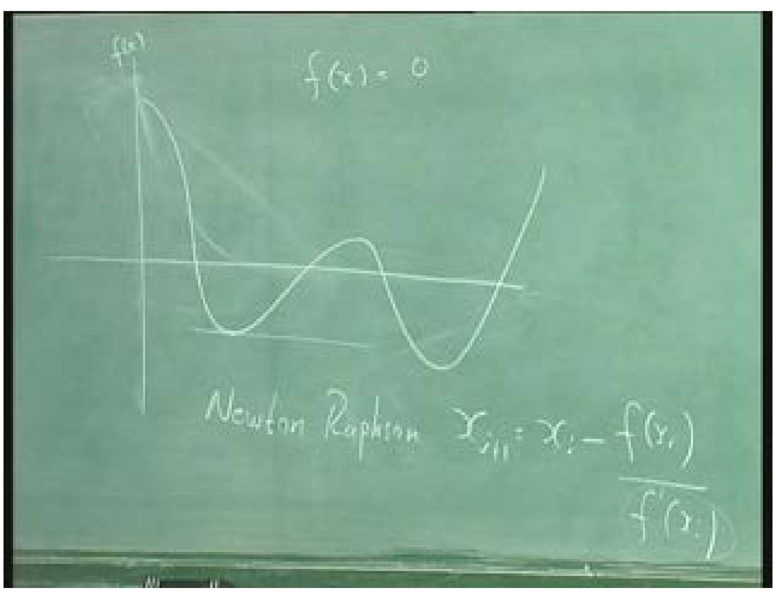
Another method which we saw was the secant method, in the secant method, again we had two guess values, okay this different from the fixed point iteration or Newton Raphson, so this was the secant method so there still we need two guess values we called x_i x_1 x_0 or x_i x_i minus 1 in general. So the next point there is to determine from this two guess values, so the idea is again that you could have difficulty in computing the f prime of x.

(Refer slide time: 08:27)



So in the Newton Raphson formula, we saw that in newton raphson in the Newton Raphson we had formula that x_i plus 1 is equal to x_i minus f of x_i divided by f prime of x_i at f prime at x_i. So now the problem is to predetermine the this f prime of x_i that is the first derivative of the function which we want to solve we are trying to solve the function of this form f of x equal to 0 that is what we are trying to do to find the minima of. So the difficulty might be may arise when we do not know what the derivative of the first function x, that is this function itself could be coming from another program.

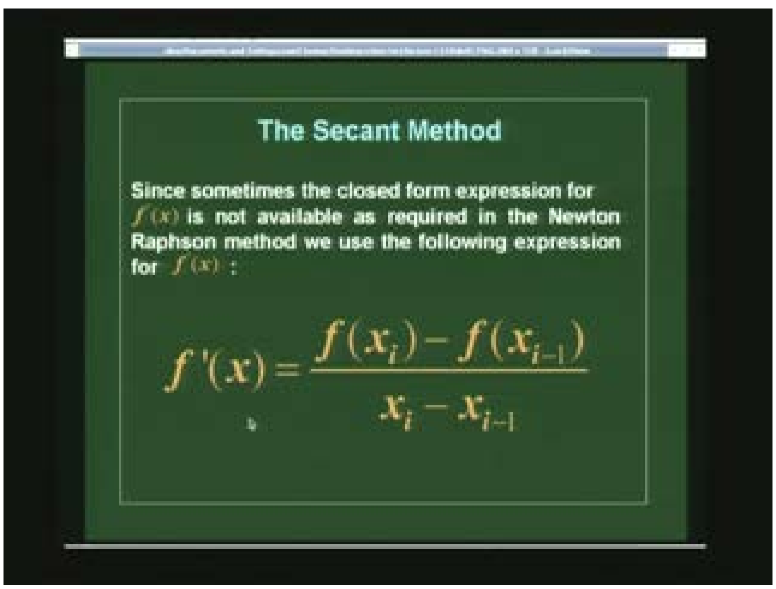
(Refer slide time: 09:56)



So we do not know actual functional formula of this, we know only the data points. We know that we give x value it gives a number some other program which gives you that and we need to find still 0s function we do not know the derivatives. So in that case, we determine the derivative from something like backward difference scheme, so that is what secant method basically uses. So we begin with two points, so we need two guess values to begin with and then we compute the derivatives from derivative, from that this 2 guess values using this formula that is f of x, f prime of x_i that is equal to f of x_i minus f of x_i minus 1 divided by x_i minus x_i minus 1.

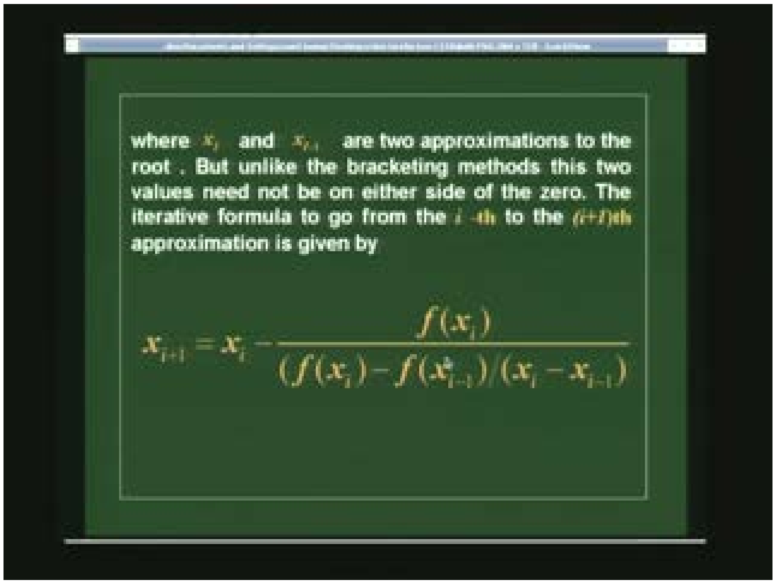
So this is the normal backward difference scheme, so we can use that so one would wonder are we are doing same two point iterations which we done at beginning that is the bracketing method. Now this, this is not because x_i x_i minus1 need not be on either side of the 0 okay that is one important point to remember here. We do not take x_i x_i minus 1 as either side of the 0 that is we do not need f of x_i minus 1 to be positive, when f of x_i minus 1 to be negative we do not need that this can be both on either side, the same side still get solution to that. So this is basically a slight modification of the Newton Raphson, where in Newton Raphson we had the knowledge of the f prime of x, we did not have in cases where we do not have knowledge we could use this formula and using two formulas in two points and then iterative.

(Refer slide time: 09:59)

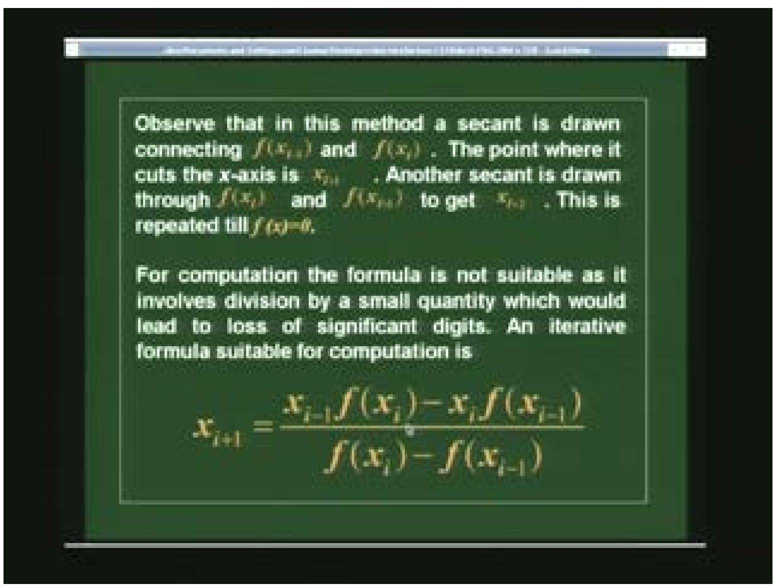


So in the basic Newton raphson we will the f prime of x by particular formula, so you remember Newton Raphson was x_i plus 1 equal to minus f of x_i divided by f of x_i minus f of x_i minus 1 x_i divided by f prime of x_i that is what Newton Raphson x_i minus 1 equal to x_i minus f of x_i divided by f prime of x. So now f prime of x that is the first derivative of f is now replaced by this backward difference scheme and we can write this in a form that varies by now given by this just rearranging the terms.

(Refer slide time: 12:00)



(Refer slide time: 12:31)

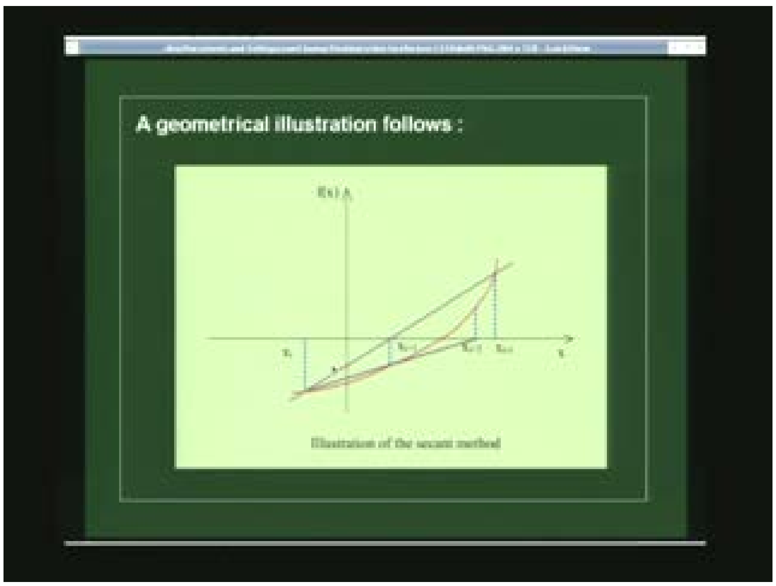


Okay graphically this could mean this that you have the function the red line, if the function of this form to be minimized whose 0 have to be find okay that is what we want to find out this is crossing point here is. So we guess with starting point with x_i minus x_i two guesses values in this particular example which was written either side of 0. We do not need to know that, we choose it from choose it on same side itself and then now instead of drawing a tangent that we can draw a secant just a line connecting two points so the backward difference for the tangent could be quiet off right, when the two points are very far apart write that is what it is. So in this two different point the two points which we guess x_i x_i minus 1 and that is very far away and then and then which connect the secants and this line crosses the x axis here, so we take as a next iteration point.

So now here comes the difference between this secant method and the bracketing method, in the bracketing method what we did was we replace x_i minus 1 always by f of x_i minus 1 is negative. We will look at whether f of x are minus 1or f of x_i which was negative and we will replace that particular point by this that is if f of x_i plus 1 here will replace x of i if x of i is negative because f of x_i plus 1 is negative otherwise you would replace this point that is the bracketing method but in this case we will always replace x_i minus 1 by x_i and x_i by x_i minus 1 that is the tangent the secant is now drawn between x_i and x_i plus 1 that is this line the next line which is below

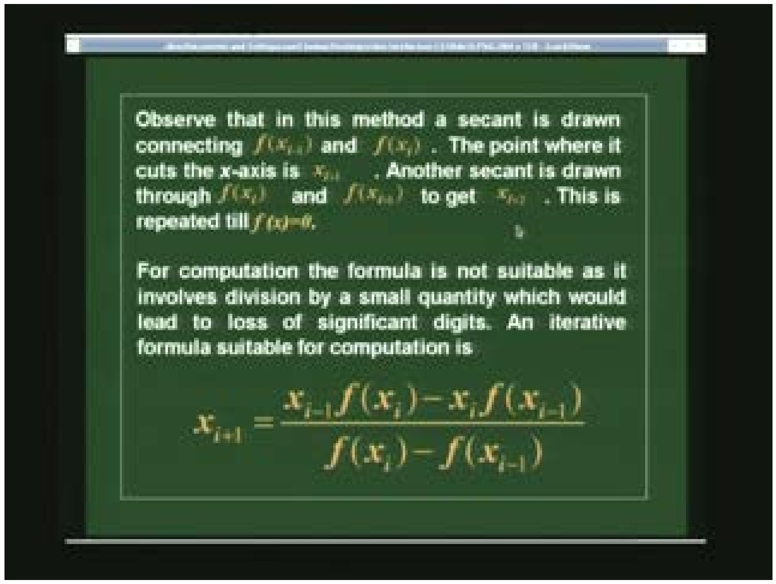
Okay that is the line which will be next draw and look at were it crossing points that is here and then we will draw between these two points now we will draw this point and we will draw this two points that is the method okay so we will continue to do notice that when we went to the second case that is started from x_i minus 1 x_i and then we went to x_i x_i plus 1 then you have the secant is the same side of the 0

(Refer slide time: 12:39)



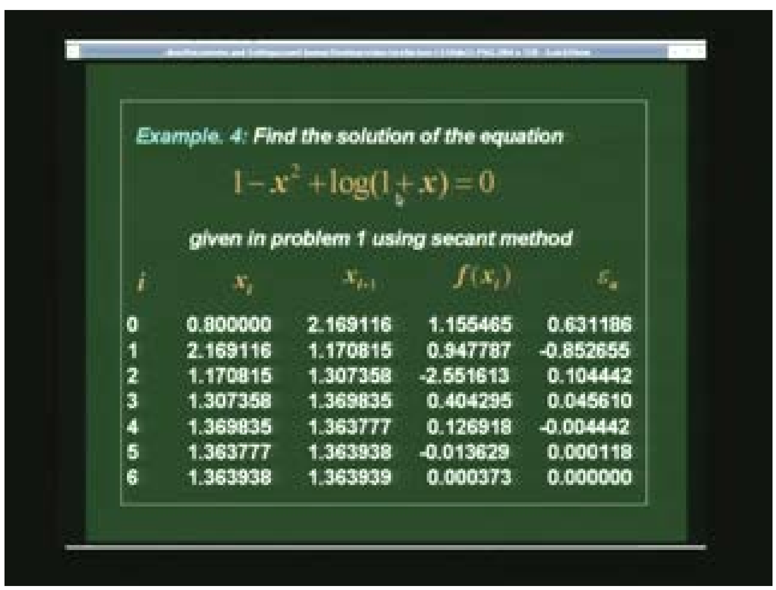
So this will not happen in the bracketing method write the bracketing method we would write say the new point x_i plus 1so now f of x_i plus 1 is negative and f of x_i is also negative I will keep the x_i minus 1, I will throw away this point now I will iterate between 3 2 point okay in the bracketing method I will draw a line connecting this point to this point x_i minus 1 but here is not done we just throw away and then we will draw a line between these two points. So I have the clear difference between the secant method and the bracketing method.

(Refer slide time: 15:48)



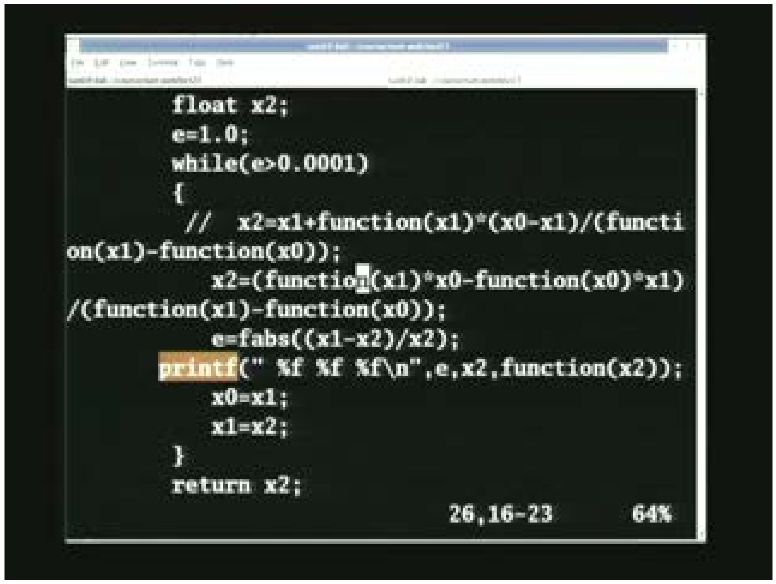
So that is the points is line is draw connecting f of x_i minus 1 and f of x_i, okay were the points crosses the x axis is taken as x_i plus 1 and then we will draw something between x_i and x_i plus 1. So that this x_i plus 1 is always dropped okay to get the next point now this is different from the bracketing method in that sense.

(Refer slide time: 16:16)

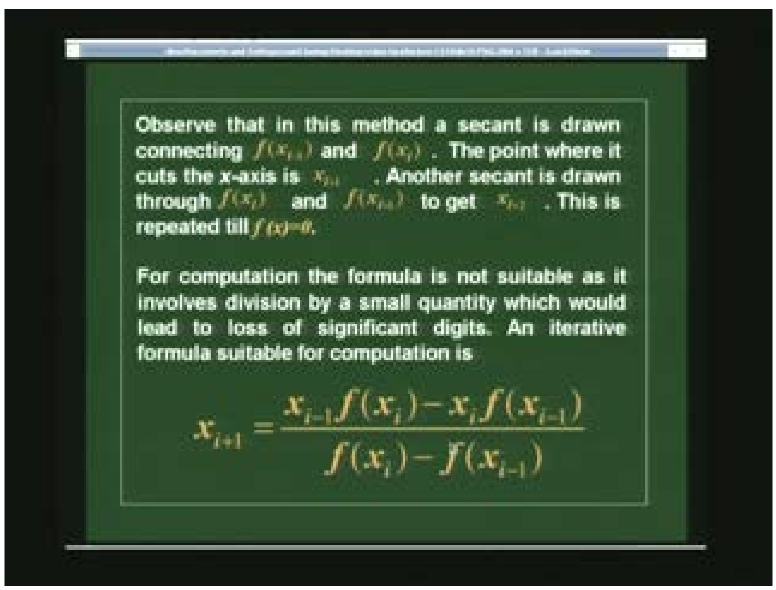


Okay, so now we will saw some example of this, now we just look at some example, we are using secant method this equation 1 minus x square plus log 1 plus x that is what we look at now. So we have a program here called that is a this program basically implements, that remember in our last class, we discuss general framework for this program that is the main function we have here okay that is start from here and ends here small main function which take two guess values.

(Refer slide time: 16:29)



(Refer slide time: 17:26)



So this program basically already does is, it defines what is the method which we are going to use here, it says okay now this is secant method that is what we are going to use this is similar to this, okay now I want to show you is this comparison with this and the false position. Okay so here, let us the bracketing method this is we are going to use here the x_2 so this is the formula which we are going to use x_2 is determined by that formula which we just saw this is a formula which we are going to use write for the from x_1 to x_2 we use this formula that is what shown here.

So that is what shown in here the function value at x_1 multiplied by x_0 minus this function value at x_0 multiplied by x_1 divided by function value x_1 minus the function value x_0 determined x_2 .So then x_0 always replace by x_0 by x_2, okay so we do not look at for what function at x_2 value to determine when x_0 should go to x_1 or x_1 or x_2 should go to x_0 should replace by x_2 or x_1 should replace by x_2 we just replace both by the next iteration point that this is x_2 is always given by this function and we always replace x_1 by x_2 and x_0 by x_1 that is the method, that way is different from the false position method and so then this secant method we just called here we will two guess values here that is the function the main function would simply read that to guess values okay and it called this method, it changes to secant method.

So from false position, we go to secant method that is what we have to use the secant method this function now we pass x_0 and x_1 whatever we read as the initial guesses into this value x_0 and x_1 to the, to the program secant method and then we also passed point to the function which we want to whose 0 we want to find out that is the general framework for the main program it is all does it is initial guesses determine which method to use the secant method.

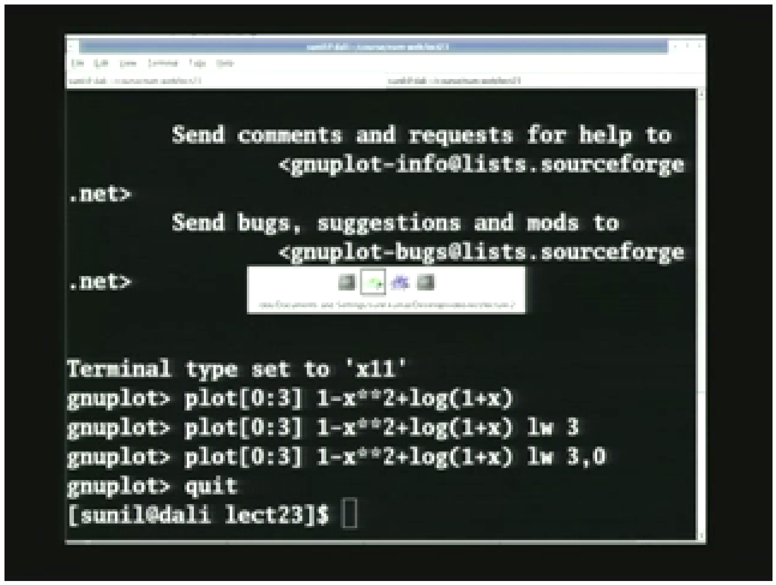
Okay and then it will passed this to guess values and the pointer to the function which we want to minimize the pointer the function called func, so we look at what is func is func is 1 minus x square plus log 1 plus x. Okay the func simply return the function value at that point that is all the program the needs, it does not need the derivative unlike in the case of Newton Raphson where we need to find the derivative.

Okay then we just, we start with the error set to be one and then it does the iteration and till the error is less than some predetermine tolerance. Okay and the error being x_1 minus x_2, the previous guess to the solution the minus the present guess divided by present guess absolute value the error, okay that is the program and then we just start with two guess values and see what we get okay.

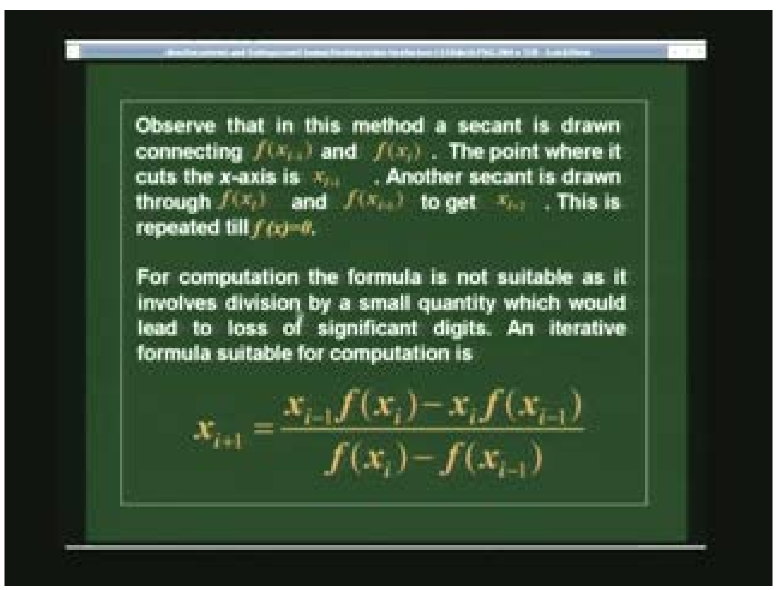
We need, we compile the program, we need two guess values just give it as 2 and 1, okay that is the 0. So okay, that it converge very fast okay compare to if you remember the bracketing method again this is as fast as the case of in the case of the Newton Raphson this is converges very fast, that is the we have printing out the error this is the error okay that is the function value and this is the x value which we got. We can see that we started with error being “.21” gone into a very small value function is converge to 0 to 0.

We could so the function itself which we looked at we looking at 1 minus x square plus log 1 plus x function, we just we looked at okay and so we plot the line also we will look at the this this the function which we saw just plotted that is the function. So we can see the error 0 is somewhere around “1.3” that is what we just did but we started with the guess which is 1and 2 that is what we guess as initial the initial guesses the 1 and 2, we could try something different we could try for example “1.5” 1.25” what we get that is that is the same side of the 0 that is makes different from the false position method. We could just do that and "1.5" and "2.5" we can see that it converges the same the rate which converges pretty fast again to "1.36".

(Refer slide time: 17:34)



(Refer slide time: 22:53)



Okay that shows that, we do not need to actually do the, we do not need to actually make the guess at the same side, the 2 different sides 0 in this particular method that is make different from the false position method and easier, to easier to implement in that sets okay so going back to our discussion here. So this is the method this is the last method which we look at, okay so we had 454 different method we are the midpoint method just begin with we had midpoint method we need it is bracketing that is we always took the next guess as x_i plus x_i plus x_i plus 1 x_i plus x_i minus 1by 2 that is our next ever x_i, x_i minus 1 we made sure on either side of 0 of the function and then we look at the false position method which again took x_i x_i minus 1 and then we drew a line between connecting this 2 points, the x_i, f of x_i and x_i minus 1 f of x_i minus 1 and look at that point cross the 0 were it crosses the x axis and we took at the next guess and we replaced one of the points by the new guesses such that we the true solutions bracketed between this 2 values then we look at the fixed point iteration method were we had f of x_i as f of x we written it as g of x minus x and then f of x equal to 0 will become g of x minus x equal to 0 or x equal to g of x.

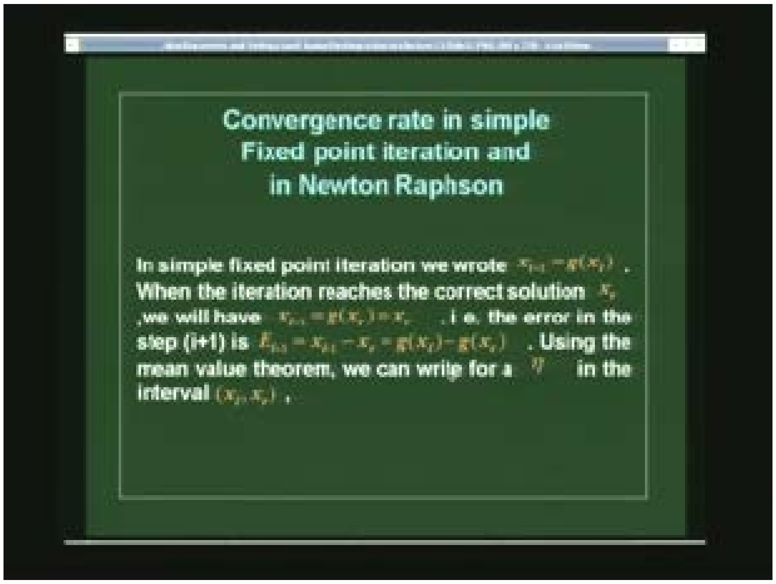
So we just at iterations that x_i plus 1g of x_1 f of x_i that needed only one guess and then we look at the Newton Raphson method and were we had the x_i plus 1 given by x_i minus f by f prime of x_i that based on tailor expansion around x_i and then we look at the secant method were the f prime of x Newton Raphson which is replaced kind of different scheme and that is saying that f prime of x_i is f of x_i minus f prime f of x_i minus f prime of x_i x_1 divided by x_i minus x_i minus 1 that is, that gives the secant method where we again we need two guesses to begin with but it does not have to be on the either side of the function or two guesses do not bracket the true solutions that is the method we looked at.

Now we will look at various convergence of this one point iteration scheme that is where we need to make only one guess what is the convergence rate of that what is the rate in fixed point iteration scheme and Newton Raphson scheme that is what we looked at the advantage of using Newton Raphson will become pretty clear if you look at this plots where newton Raphson works that is the best method that is everybody using everybody uses so whenever Newton Raphson works that is the best method so the reason for that is clear from this.

Let us look at fixed point iteration scheme first. So in that case we write x_i plus 1 as g of x_i that is what fixed point iteration which we just now we had and then we let say x of r is our true solution, okay then we would say that x_i plus 1 is g of x but if then true solution g of x_r equal to x_r if it is true solution, now if its not a true solution and then we have this error right the x_i plus 1 minus x_r that is our error okay so we call that is a error in the i plus 1th iteration. So I go this is clear, so we say that x_i plus 1 is equal to g of x_i that is our iteration scheme and then we have if this is the true solution then g of x_i this x of i x of i plus 1 should be equal to x_i we called that is x_r that is been true solution. So the difference between x_i plus 1 and true solution is our error right.

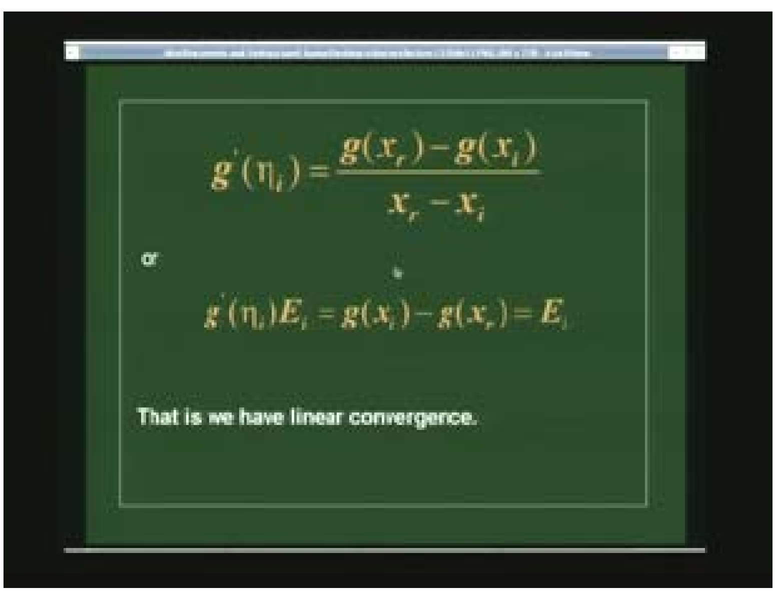
So that is our error in the i plus 1 iteration scheme okay, so now g of x_i plus 1 is g of x_i sorry x_i plus 1 is g of x_i and x_r is g of x_r because that is true solution right hand side and so we could use that formula here for the error okay. So e_i plus 1 which is x_i plus 1 minus x_r that is is the difference between the value obtained at I th I plus 1th iteration to the value to the true value is equal to the value of function g of function not the f g at x_i minus g of x_i.

(Refer Slide Time: 23:10)

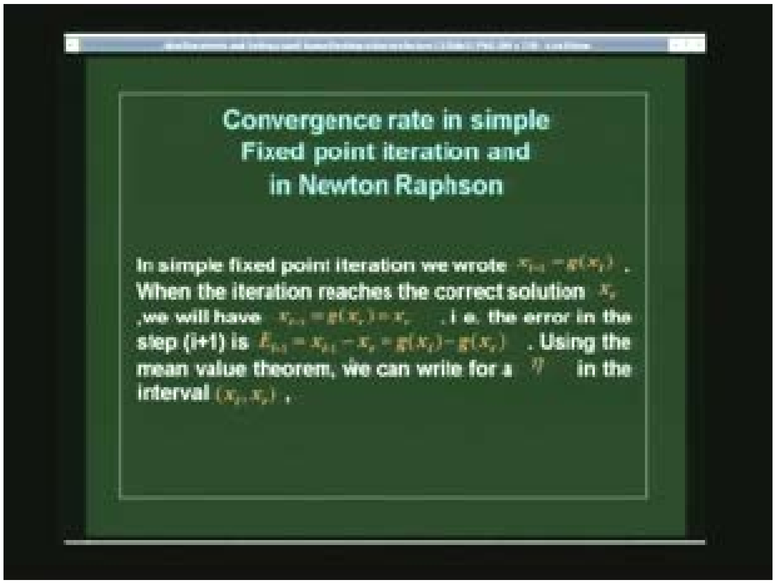


So we can write that okay then you remember the mean value theorem which we used in the case of polynomial interpolation that is to say that we can always find a point it be called at Etta I between x_i and x_r such that g of x_r minus g of x_i divide by x_r minus x_i is the derivative of g at that point. Okay we can always use that so we will use that term property that g um one point where between x_r and x_i where we have the derivative is equal to the equal to this mean value, there is x_i minus x_r g of x_i minus g of x_r divided by x_r minus x_i and then we can write the g prime of Etta I in to e_i, now remember this x_i minus x_r would be the error at the i th step right, So if x_i plus 1 minus x_r is the error I plus step x_i minus x_r is the error the i th step.

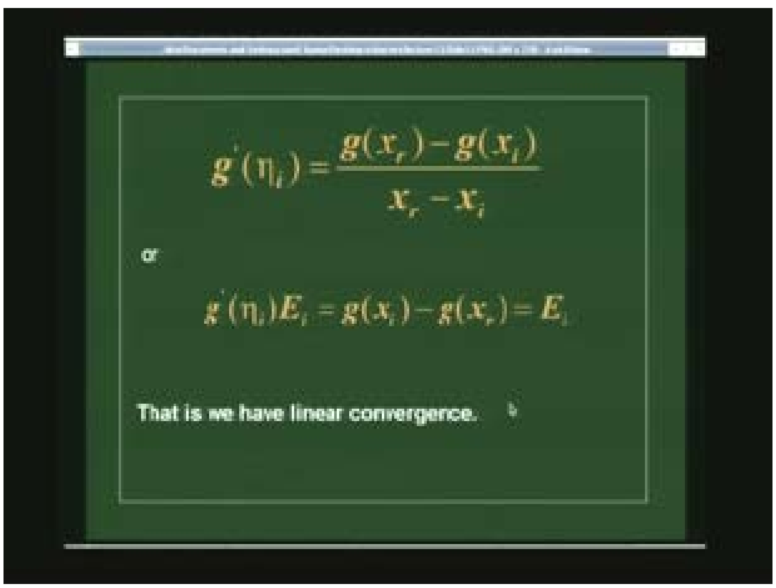
(Refer Slide Time: 27:48)



(Refer Slide Time: 278:37)

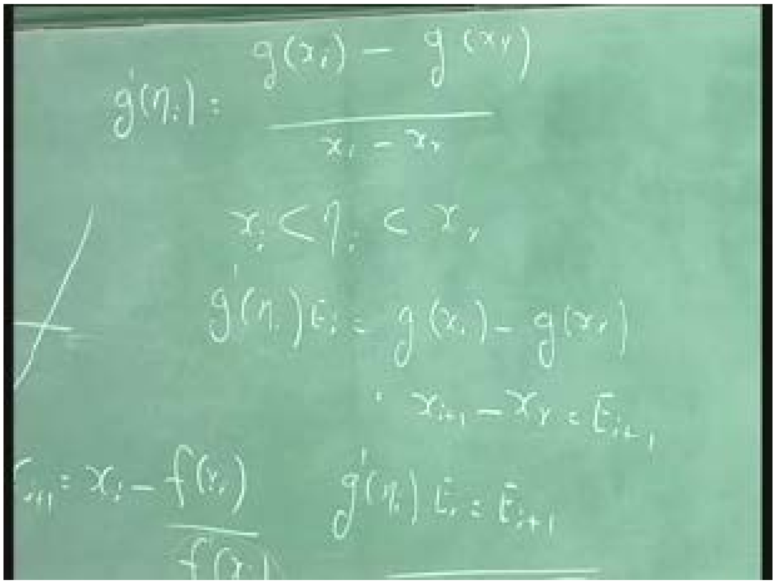


(Refer Slide Time: 28:45)

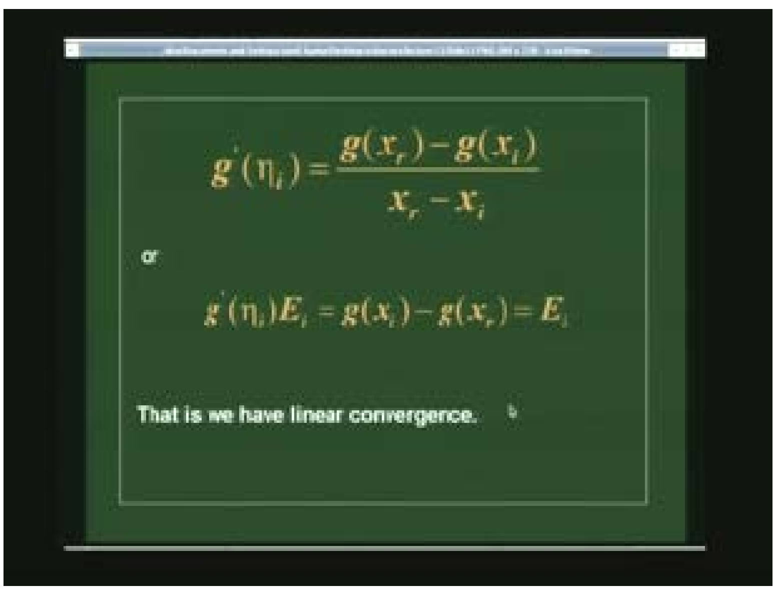


So I can multiply by this by e_i and write it as g prime of Etta I in to e_i as g of x_i minus g of x $_i$ right. So the minus sign here the minus sign here, so it is e_i there should be e_i plus 1. So the formula we want to do is following that is so what we saying is that I have g of x_i and so g of x_i minus g of x_r, so I want to write g of x_i minus g of x_r divided by x_i minus x $_r$ we want to call at g prime at Etta I using the mean value theorem. So Etta I is some point between x_i and x $_r$, so Etta I is greater than x $_i$ less than x $_i$ somewhere in between x $_i$ and x is this value Etta. So then x_i minus x_r is my e $_i$, so I can write g prime of Etta I in to e $_i$ as g of x_i minus g of this x_r x_r but g of x_r is nothing but x_i plus 1, okay this is g of x_r is x_r that is e_i plus so we have a simple formula its says g prime of Etta I g prime at Etta I in to e_i equal to e_i plus 1 that is the formula which we need to use.

(Refer Slide Time: 30:24)

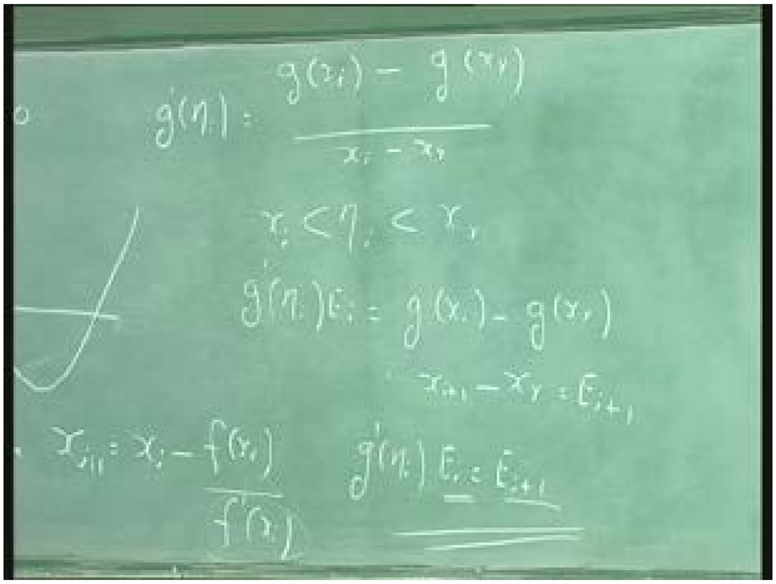


(Refer Slide Time: 30:26)

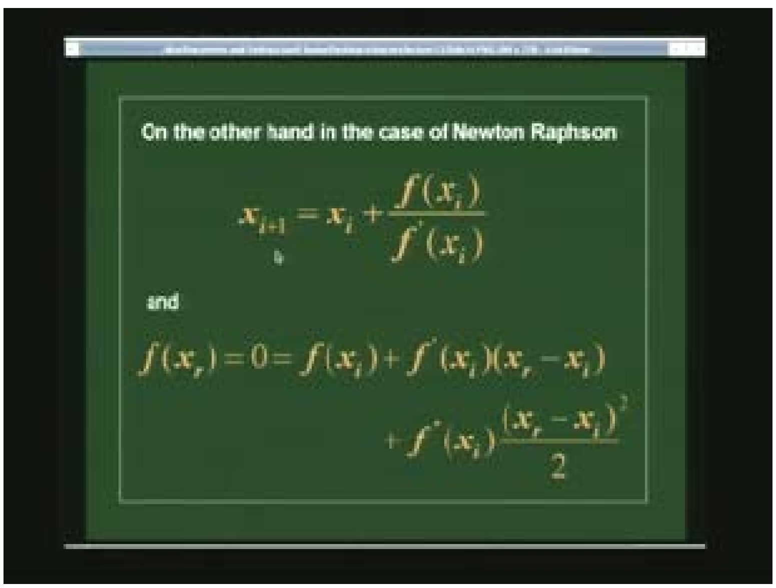


So that formula tells that we have a linear convergence okay its means the error in the i plus 1 is linearly related to the error in the i th step okay that is the point which I want to emphasize here. so the error in the i th step is related to the error in the i th step by some derivative of function some point, it is not important what this is some derivative but it is a linear derivative at some point, it is a but linearly related these two are linearly related so this 2 errors are linearly related. So we have a linear convergence that is what we would say we have a linear convergence.

(Refer Slide Time: 31:07)

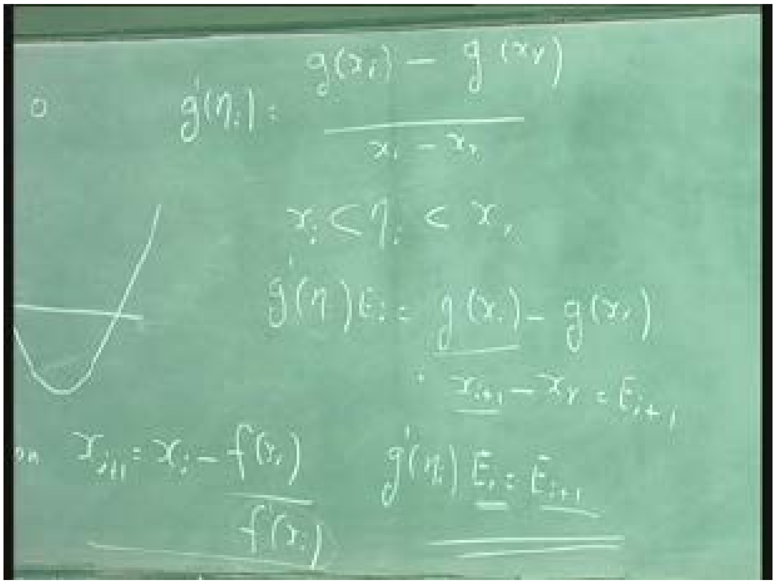


(Refer Slide Time: 31:09)

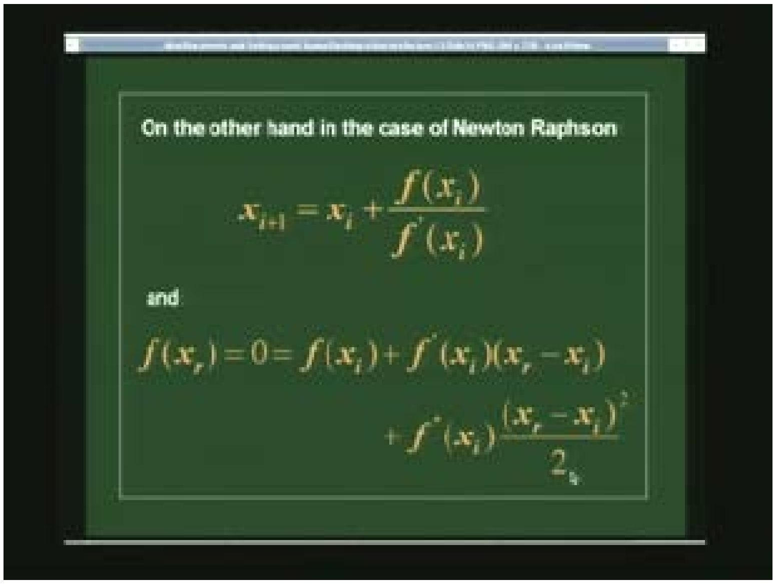


So it should be e $_i$ plus 1 and then the next scheme that is we look at the same thing for Newton Raphson so for now in this particular case we had g of x_i has x_i plus 1. So that is were scheme, we had x_i plus 1 given by g of x_i. So when we look at the Newton Raphson we have this formula x_i plus 1 is equal to x_i minus f of x_i by f prime of x_i. So let us see whether we could again using the same definition for error as x $_i$ plus 1 minus x_r as e of i plus 1 what kind of relation between e_i and e_i plus 1, we can get this in this particular

following. That is what we look it now okay that is so we have x_i plus 1 given by x_i f of x_i by f prime of x_i. Okay and then so now if f of x_r is the true solution in this case f of x_r would be 0 right so then I have f of x_i if I expand around f of x.
(Refer Slide Time: 31:48)



(Refer Slide Time: 31:51)

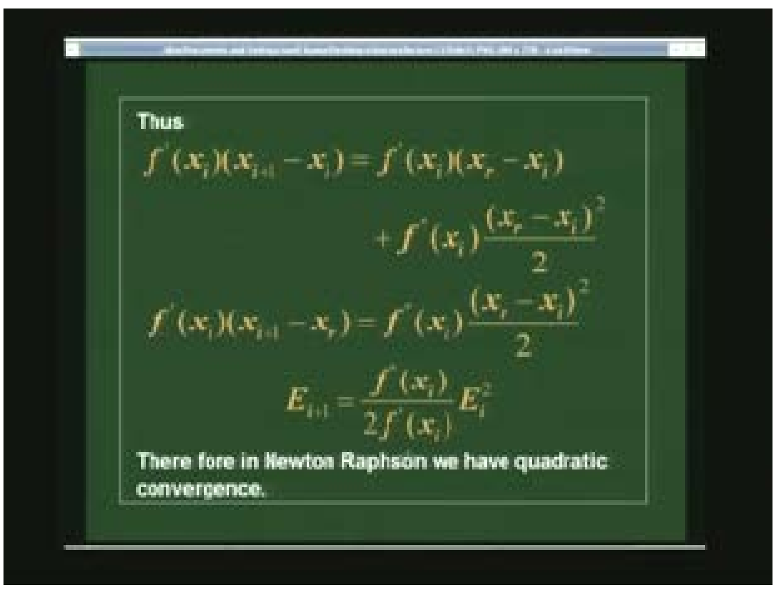


Okay so if expand around f of x_i, so I get f of x_i as f prime of x_i and x_r minus x_i plus f double prime of x_i etcetera right. So that is the tailor expansion around x_i, so I tell you to expand the expansion around guess x, x_i okay and I say go to x_r from that okay if x_r is the true solution then the function value is 0, so I have 0 equal to f of x_i plus f prime in to x_r minus x_i, f double prime the second derivative x_r minus x_i whole square by 2.

So I can rewrite that as you know f prime of x_i in to x_i minus x_i plus 1 minus x_i as this. I am just rewriting the function. Okay, so I have just rewriting this, this whole function that I have written as f prime of x_i in to x_i plus 1 minus x_i is now given by f prime of x_i x_r minus x_i and if double prime of x_i in to x_r minus x_i by 2 whole square. So then I can rearrange this again, so basically just I take function to left hand side and then write as f prime of x_i in to x_i plus 1 minus x_r equal to f double prime x_i in to x_r minus x_i whole square by 2. So that is the case, so we have then obtained action between x_i plus 1 x_r is e_i plus 1. We are doing exactly as what we did in the case of fixed point iteration and then now we have x_r minus $_i$ as e_i this x_r minus x_i whole square, so e_i square.

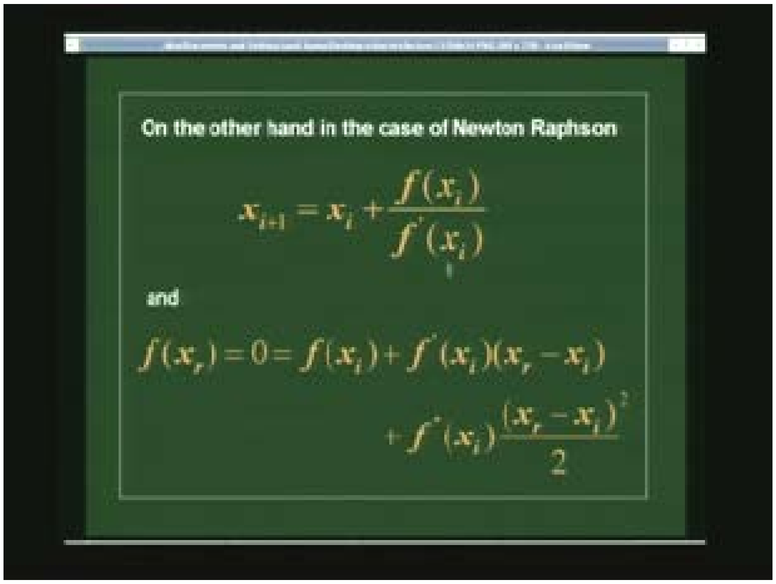
So we have now found that e_i plus 1 is equal to okay we have e_i plus 1 is given by this is equal to f double prime divided by f prime f double prime divided by f prime in to e_i square, okay so in this case the Newton Raphson case we have e_i plus 1 it is the error at I plus one step being related to the error in the I th step if this quadratic relation its e_i plus 1 is equal to e_i some quantities some derivative some quantity which involves the second derivative and the first derivative multiplied by e_i square.

(Refer Slide Time: 32:45)

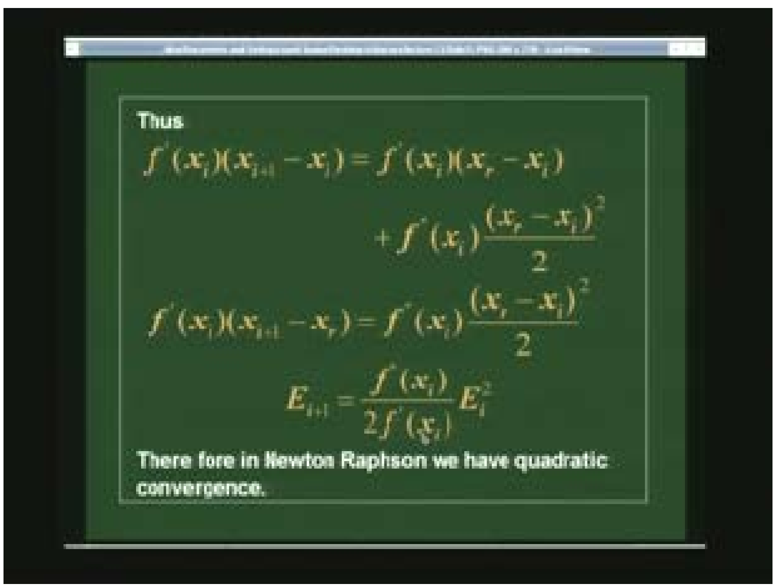


So again where if the f prime of x is not too small this is quadratic convergence we have that is the reason why Newton Raphson convergence much fast much faster than any other method provided here f prime of x_i. Okay that is the that is the summary here we just used this particular idea that is we can write you know f of x_i plus f of tailor expansion this function around x_i and we say that we go to x_r that that means we can write it as f prime of x_i in to x_r minus x_i and double prime of x_i in to x minus x_i whole squared etcetera and then I can use this f of this value as you know x_i minus 1 x_i plus 1 minus x_i in to f prime of x_i. So that also gives me f of x $_i$ right, so two ways of getting it, so this f of x_i also given by x_i plus 1 minus x_i in to f prime of x_i that is what I am comparing here.

(Refer Slide Time: 34:45)

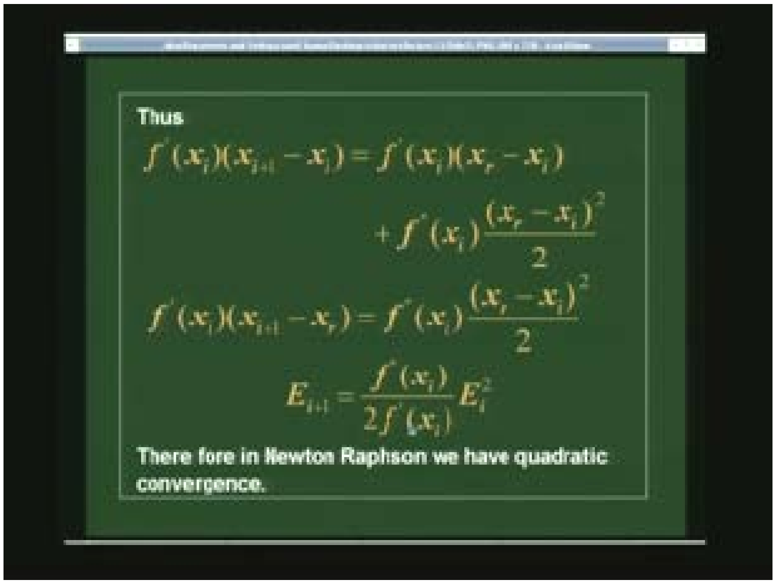


(Refer Slide Time: 35:42)



I compare these two and then from there I get the error by subtracting this term from this term when I writing this form so I get e_i plus 1 and I get e_i square on the right hand side that is the simple formula for the linear as quadratic convergence of the Newton Raphson method. So we saw, what we see here, we see that in the case of fixed point iteration, we have linear convergence, where in the case of Newton Raphson we have quadratic convergence. So that is what the point is.

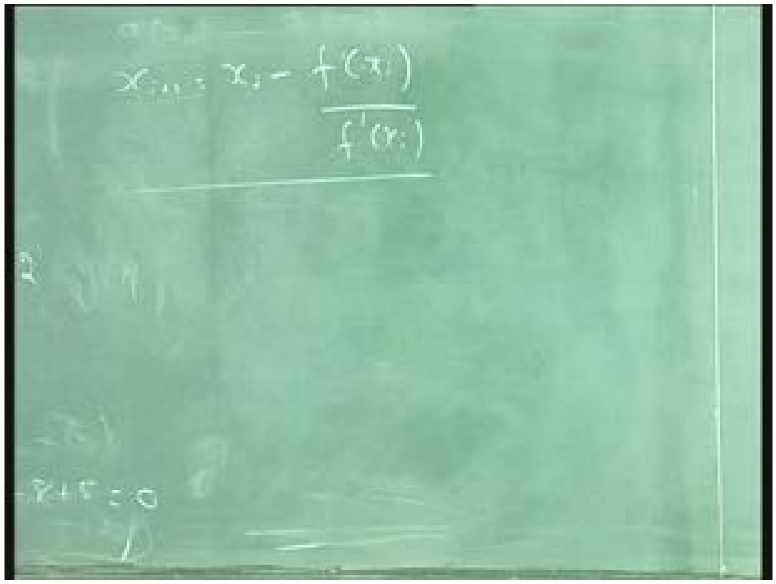
(Refer Slide Time: 36:03)



Okay so now this again points of fact that Newton Raphson is the best method provided we do not have f prime of x $_i$ going to, going to very small value. So one place where that actually comes when we have what is known as multiple root that is suppose we have a function which is of this form, that is suppose we have a function which is of this form which is x_i minus 1 whole square in to x_i plus 2 x_i minus 2. So now we know that where the root functions are, this function roots would be at x equal to 1, if definitely root of this function. Okay so now if I look at the derivative of this function, f prime of x_i okay that is lets expand this easier to see this way so we can write function f of x as x square minus 2 x plus 1 in to x minus 2. So that is x cube minus 2 x square plus x, so we have a function of this form that is the function we have and because we know form this we can easily see this okay, so x equal to 1 this is f at 1 is 1 minus 4 plus 5 minus 2 that is 0 right that is, the that is which is 0 okay. So look at f prime of x at 1, okay at x equal to 1 that is what we want to look at.

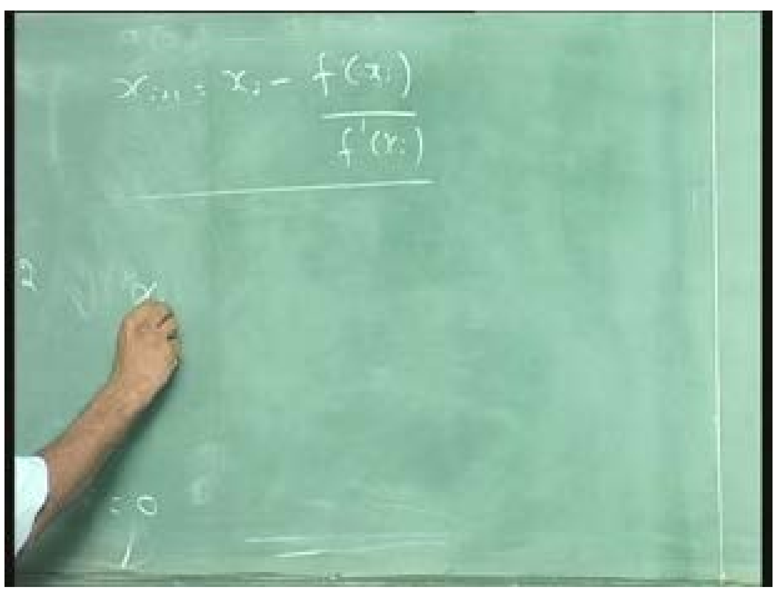
So we look at f prime of x equal to 1 f prime of x is 3 x square minus 8 x plus 5 okay so then we look at x equal to 1 that value of that is 3 minus 8 plus 5 we get that as 0. So that is this so problem point for um Newton Raphson. So we had we know that root is equal to one okay so then we saw that f of x at 1 is actually 0 that is the root but then how do we go to Newton when Newton Raphson how do we approach this point. So we approach this point by using the formula that x_i plus 1 is x_i minus f of x_i minus f of x_i divided by f prime of x_i that f prime at x_i but f prime at x_i is 0 so as you go towards this point so of course we are not interested exactly at point f prime at this, at that point x equal to 1 that is the root both f of x and f of x, f prime of x_i are 0. So the question then is what goes to 0 faster okay so that if this goes to 0 faster then we are in travel that is an example. So I am just trying to giving you reason a point where it can have travel.

(Refer Slide Time: 40:03)



So if you actually plot this function so you would see that this point the derivatives is also 0 and then what happens to either side of the function right if you look at a value which is slightly larger. So since the first derivative also go to 0 this function actually does not change sign then okay so see it touches that 0 and goes away. So this kind of points depending upon f prime of x_i goes faster f of x_i goes faster you can get difficulties.

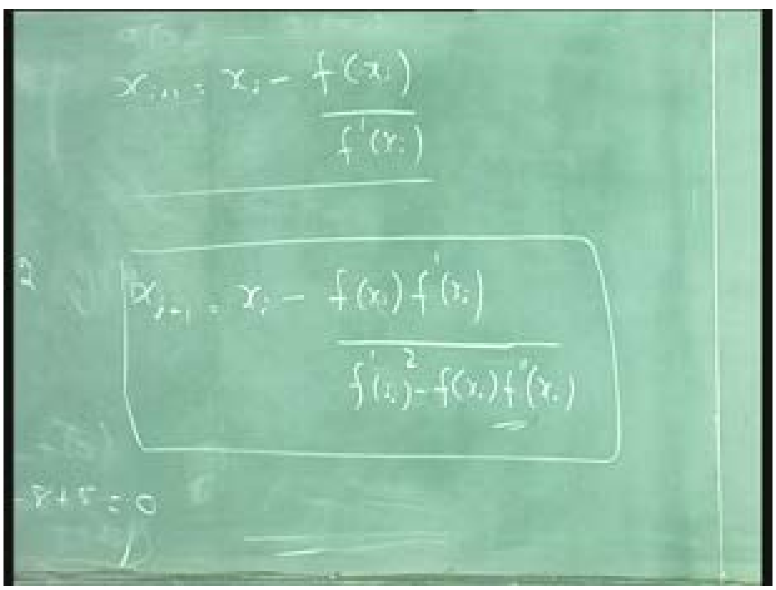
(Refer Slide Time: 40:37)



So if f prime of x goes to 0 lower than f, fx then of course we can it will converts so a better formula to use this point to use the second derivative also and that is what we would see in implementation here, I will show you that implementation this okay here is the this is the formula which your using before this x_2 is equal to x_1 minus the function value divided by the derivative of the function value function that was we are using before, this is the derivative of the function. So this this formula is one replaced by slightly by different formula which is x_2 is equal to x_1 minus the function value multiplied by the derivative of value divided by the derivative multiplied by derivative square f minus, f minus derivative 2 that is we are going to write the formula that is x_i plus 1 equal to x_i minus function now is multiplied by function at x_i is multiplied by f prime of x_i divided by then f prime of x_i square minus, okay that is basically just taking the 2 derivatives of this function itself taking the derivative here and then replace by the formula that okay that is what we are doing here.

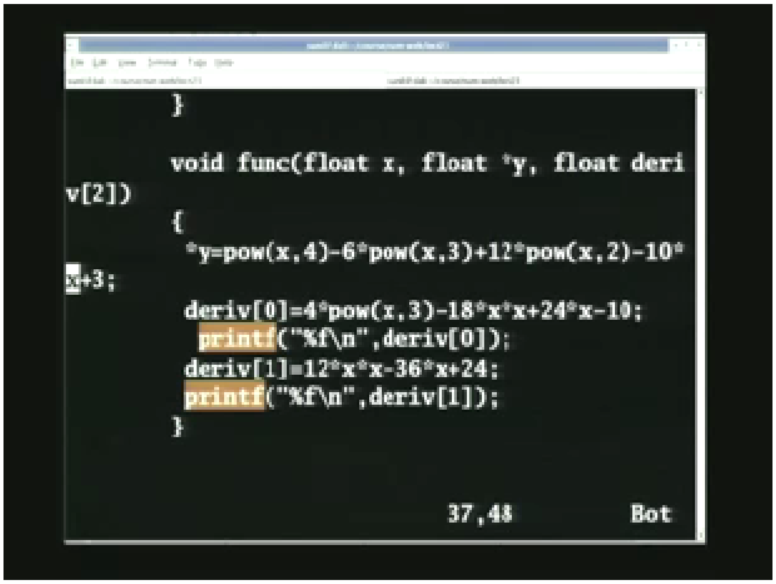
So we are saying that x_i plus 1 now x_i before we had f of x_i by f prime of x_i so in the case now what is called multiple roots that is you have x_i minus 1 whole square into x_i minus 2 or x_i minus whole cube thinks like that where many derivative it that vanishes okay that is in this case the two derivatives the first the function and its derivative vanish at the root but, if you could if the cases where you know if x_i plus 1 whole cube for example, then you would have the first derivative and second derivative would vanish at this point okay.

(Refer Slide Time: 43:07)

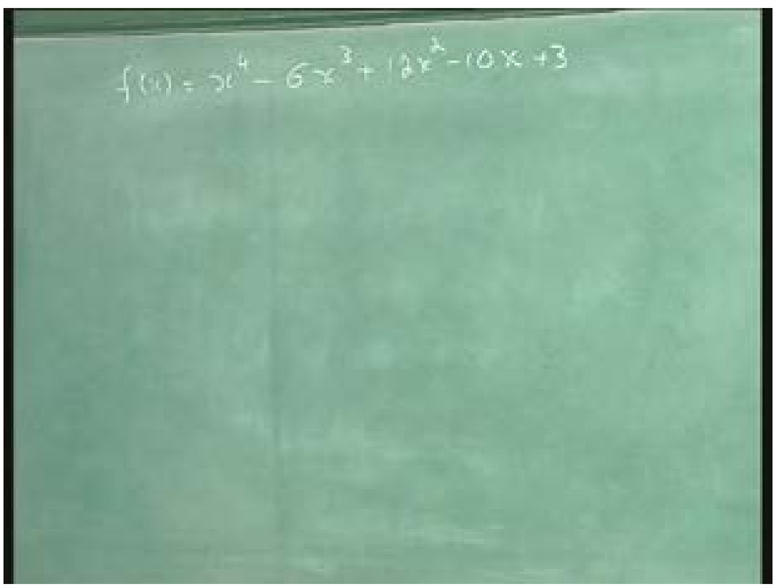


So those cases it will be advantage to use the formula of this form that is x_i plus 1minus equal to x_i plus f of x_i and f prime of x_i divided by f prime of x_i square that is first derivative squared minus the function to the point multiplied by second derivative. So the disadvantage is of course to that now we also need also to determine the second derivative, so where the second derivative is available and then this is the faster method for convergences, just sees that right now. So let us use the whole formula, so this program here actually minimize the function of this form that is function here, hope you can see the function, that is the function is x to the power four minus x_6 x cube plus 12 x square minus 10 x, I will write this function.

(Refer Slide Time: 43:09)

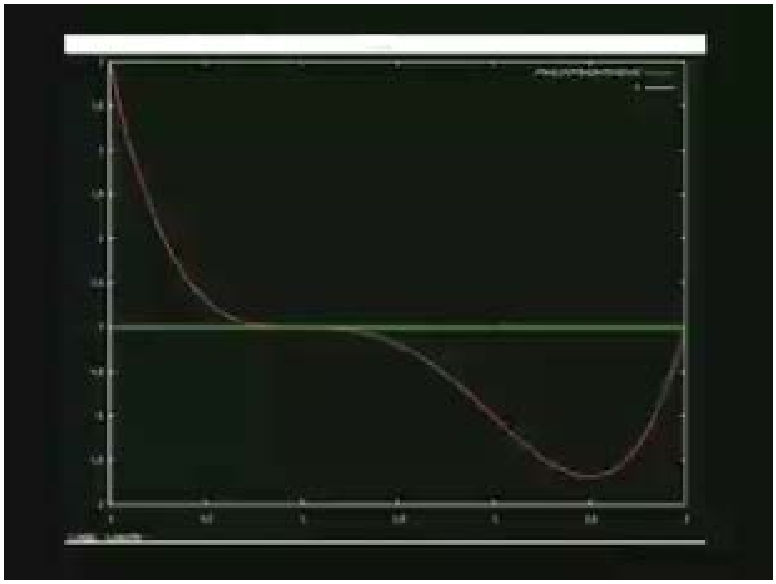


(Refer Slide Time: 43:51)



So we will try to minimize the function of this form that is the x to the power 4 minus 6 x cube plus 12 x square minus 10 x plus 3, okay so you want to find the 0 of this function so I plot the function here. So I am just plotting this function here, that is x to the power minus 6 x cube plus 12 square minus 10 x, 10 x plus 3. So I am just plotting that, so that is given here, so you notice that it goes to 0 point around 1 right, set one goes to 0 but it goes to 0 extremely slowly that is it almost flat this, the first derivative and second derivative vanishes.

(Refer Slide Time: 43:54)



So now we will use Newton Raphson to find the 0. So let us start from point 5 something and see how fast it converges to that point using our old formula for Newton Raphson method okay. So we will use that, so we will compile this program. So remember that we are using here is, this same function that is x to the power four this p or w is the power function built in function gives x to the power 4 here and x to the power 3 in to 6 x to the power 2 in to 12 and 10 x plus 3 that is what the function we have, okay and then we are using the simple scheme that is, we are going to use the normal newton Raphson that is x_2 is now x_1 minus f by derivative function derivative here.

So here I called this function okay which is pass on to method, I am using is Newton Raphson again, so initial guess value in the main program, okay I have one initial guess value which I defined, okay and it passes the program called Newton where initial guess value is passed and the pointer to this function is passed and this return the 0 and the program will just print the 0 out. So now the function Newton takes this pointer as some other value called function does not matter.

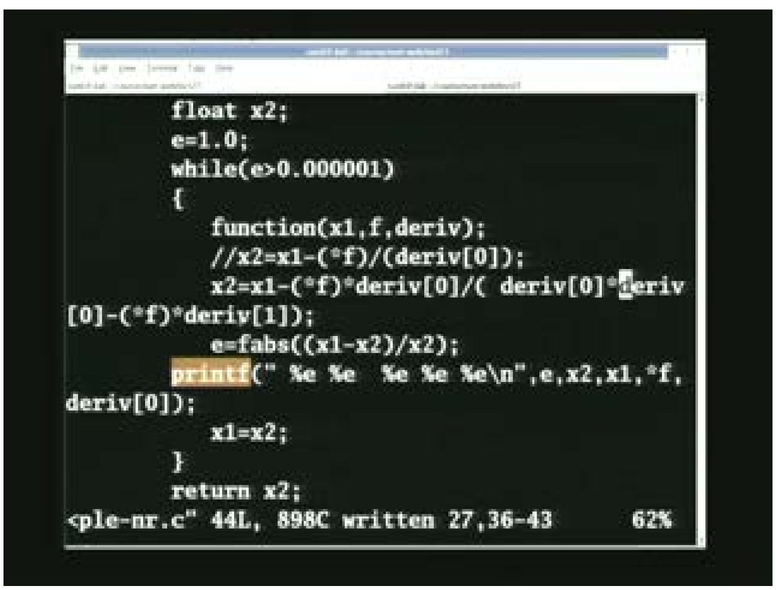
So then x_1 is the initial guess value and, the function now needs this this thing called function that is basically func right. So func now takes an argument if 1 x value okay and 1 point where which is actually the function value itself okay and array in this array now written array of derivative that array 0, the first element of array does not derive 0 take the first derivative and derive 1 takes the second derivative.

So that is 0 and one first derivative and second derivative of the function that is what it written but to begin not to use that lets just use only the first derivative use by f by divided by the first derivative formula okay start of f where f is a pointer here, as we saw this is a pointer this is an array of derivative. So the first element being the first derivative and a second element being second derivative otherwise the program is same as before. Okay so now this x_2 is x minus f divided by derivative of f, so that is what we are going to use in this program. So we will run this program okay, so we will run this program now with this particular function, so give some initial value “.8” and we see that the uses convergences. So it converges to 1 we saw that the root was around 1 in that we plot at this thing converges to 1 okay, so but this takes to many steps to converge, okay it has to go through many iteration scheme to converge, okay so but we will look at the slightly different formula.

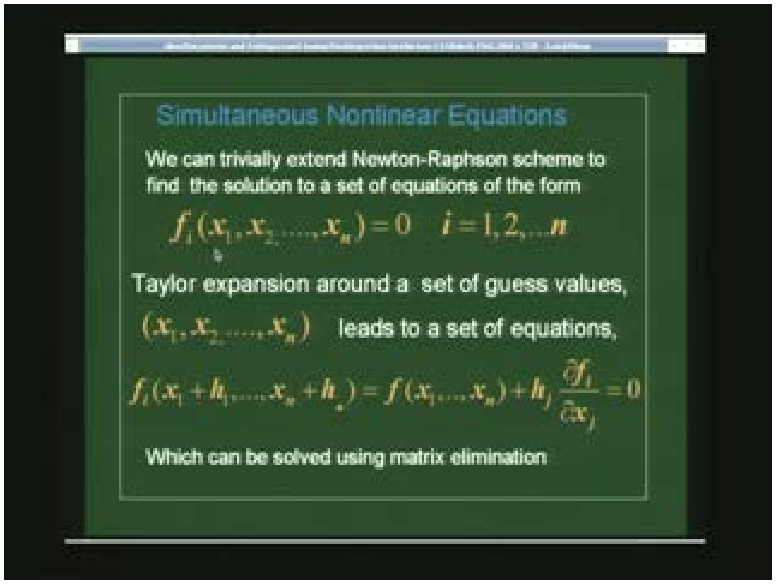
So that is we will now look at this case where we have so what we have here is now $x_1$minus f in to derivative the first derivative divided by the derivatives squared the first derivative squared minus f function in to the second derivative. So that is the formula which we are going to use now, so we will run that formula. So we will compile the program with that formula and then we will run the same iteration itself. So again this converges to 1 but we will see that converges much faster see, we started from here and it converges much faster here, then the previous iteration.

Okay that is the thing which I want to tell you that in this case where we use this modified formula that is where we use the derivative of 0 multiplying the function and also use the second derivative okay the convergences rate is much faster, okay then the formula which uses only the first derivative, okay you can see that in various functions which you can try out yourself. So the Newton Raphson with the second derivative converges much faster than Newton Raphson first derivative provided of course that which you would be find the second derivative of the function. So this works much better and the round of errors and other errors are much lower in this case where we use the second derivative.

(Refer Slide Time: 44:40)

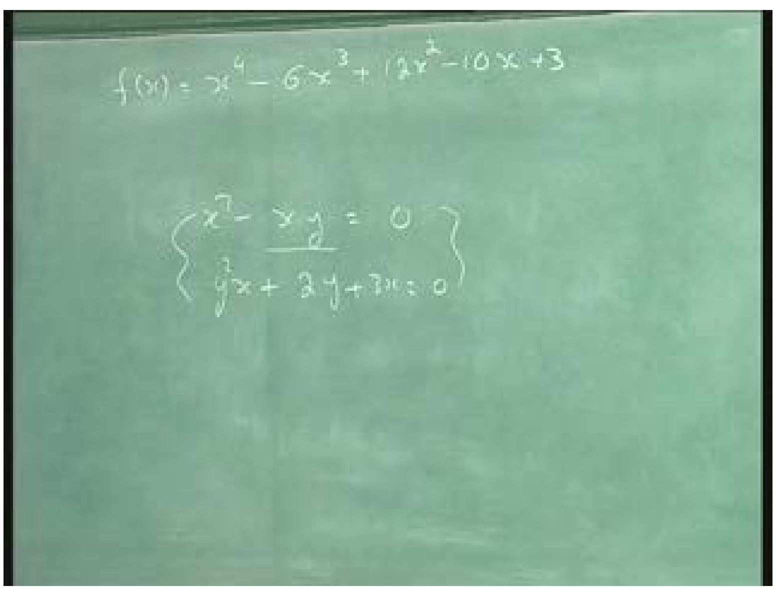


(Refer Slide Time: 49:39)



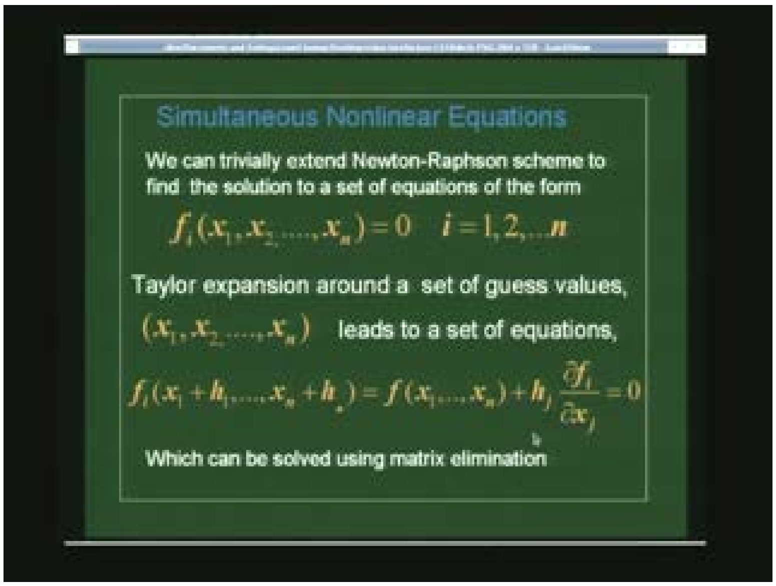
So before we end this session on the Newton Raphson method and finding the zeros of non-linear function, okay we will just look at one more problem that is what happens if you have instead of one function, we have many function and we want to find a simultaneous root of this set of non-linear functions. Okay given for symbolically here you could have a series of function n function okay which has n unknowns, so we could you know for example, we could just have a function say x square minus xy equal to 0 and then you have some other function which is y square x plus 2y plus 3 x equal to 0 some function like this, okay you have 2 unknowns x and y and then you have 2 equations.

(Refer Slide Time: 50:41)



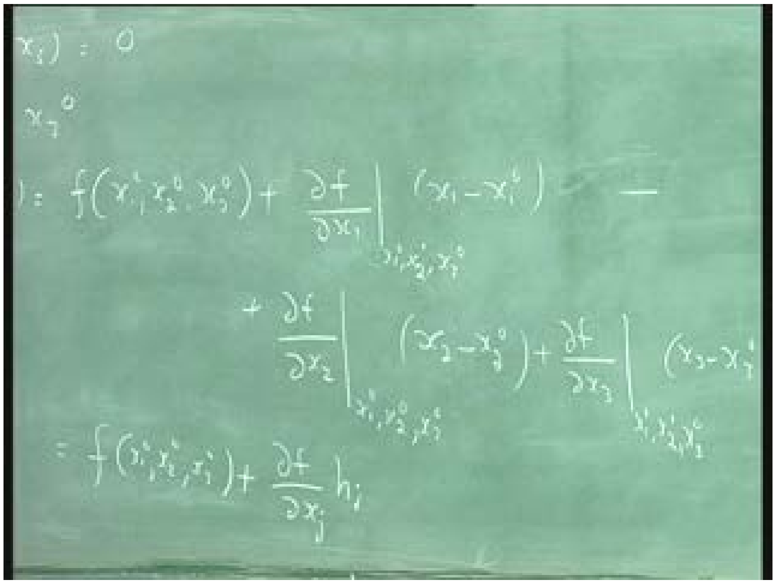
Okay so you could have series of function of this form, you could in principle have a series of function, we call them f_1, f_2, f_3, f_n, f_n are unknowns. So in that case how do we find the solution of this equations again it turns out the Newton Raphson pretty easy because we can make the guess value for all the solutions that this case n values to be found find the solution this function the set of function and then, so let us assume that the x_1, x_2, x_3 up to x_n and then we want to go and find out the values of x_1, x_2, $x_3$etcetera, such that all those functions goes to 0 right all of them goes to 0 simultaneously. So the way to do that is again expand around that guess value which we have wee do tailor expansion, so when we do a tailor expansion such form, so we would now have f of let us take an example x_1, x_2, x_3 equal to 0 that is what we want to find out solution of this. Okay we start with some x_1^0 start with some x_1^0, x_2^0 and x_3^0 as our guess solution and then we will write f of x_1, x_2, x_3 as the value at x_1^0, x_2^0, x_3^0plus derivative of function by x_1 right.

(Refer Slide Time: 50:44)



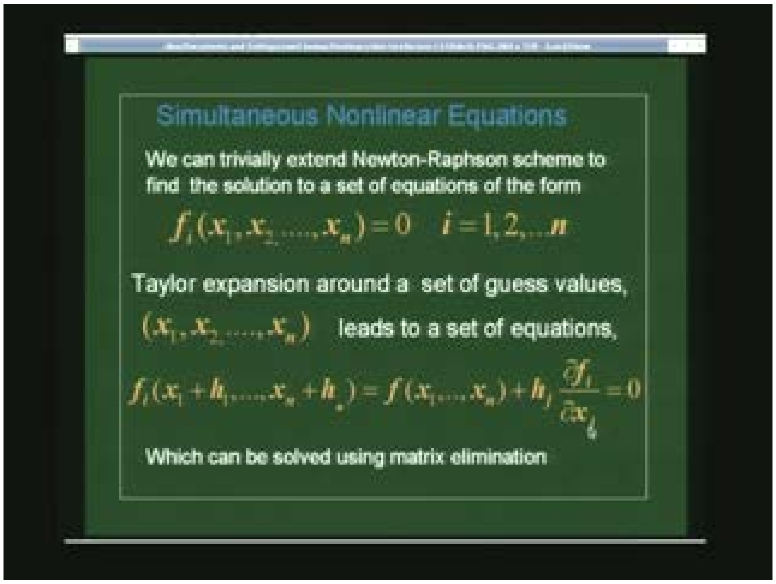
So that is the first derivative at x_1^0, x_2^0, x_3^0 in to x_1 minus x_1^0 and then we have the derivative of the function plus the derivative of the function at with respect to x_2 second variable again at x_1^0, x_2^0, x_3^0 in to x_2^0 minus x_2 minus x_2^0 right and then we have the next one that is derivative of function with respect to x_3 at the guess the initial guess value in to x_3 minus x_3^0. So that is what we call h_1, h_2, h_3 okay, so I can write the in the symbolic form as f at x_1^0 del f by del x_j in to h_j, so were h_j is x_2 x_j minus h_j 0 that is what I can write sum over all the j $_s$, so sum over all the unknowns okay that is what I have written here.

(Refer Slide Time: 53:33)

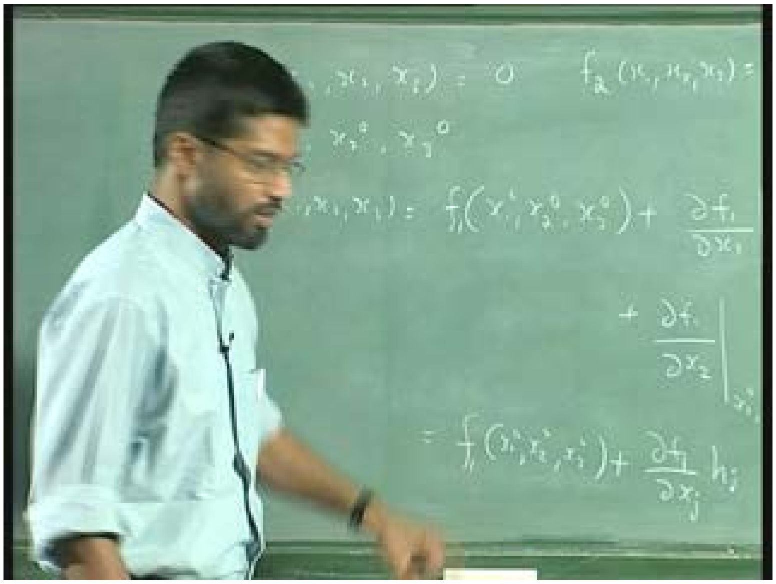


So that is this function I have written as. So then I can say that, okay as same as before same as one variable okay left hand side should be 0, so then I have a set of linear equations because I know the derivative for and then I have x_j del f by f_i by del x_j is equal to okay that is what I would have here.

(Refer Slide Time: 53:35)

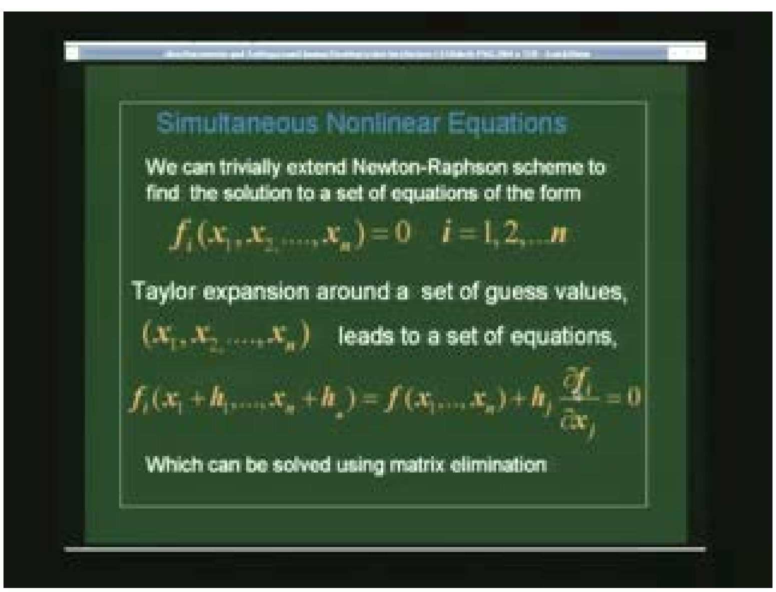


(Refer Slide Time: 54:36)



Again I have to write both the cases, so either 3 equations like this f $_1$ function f $_1$ and that is for function f $_1$. Okay I will have similar equation for f_2 and f_3 either 3 equations right. So will actually have three equations for this we will have f_1 f_2 here which is the 3 unknowns I need 3 equations and then I will have a f_3 of x_1, x_2, x_3 equal to 0, there are 3 equations so to expand all the three functions in this form okay that is the summary which you will get so you will get 3 linear equations and we know how to solve a set of linear equations, we are going to look at elimination method etcetera. So we can use those set of equations to solve particular problem I have just briefly show you that thing here.

(Refer Slide Time: 54:38)



So we have, if you have system of equations okay, so if you have system of equations to look at, okay again we will have to use some metrics method okay set have non-linear equations here and this function. So this function has before now the main program is same as before only thing is now, our function will have set of equations to be given okay, so now that function now we have to construct this metrics equation, okay that is I am going to show here.

So here, so this called its function okay we have x is now the initial guess value all same as before, so we had the function which returns the function value now the function value return in an array okay and derivatives are written in a 2 d array and the guess value itself is now an array. Okay so that is would we see the details of this program and the implementation of it in the next lecturer, but just for the time being look at this yourself. So what we have to do is now the modification from the earlier program would be just that the when you call the function value now we have to give it in array of initial guesses because there are many unknowns and then the function return as itself as an array1 d array because, one dimensional array because it has many function and the derivative now as a 2 dimensional array of derivatives we will look at this program in the in the next class.

(Refer Slide Time: 55:00)