Computational Chemistry & Classical Molecular Dynamics

Prof. B. L. Tembe
Department of Chemistry
Indian Institute of Technology – Bombay

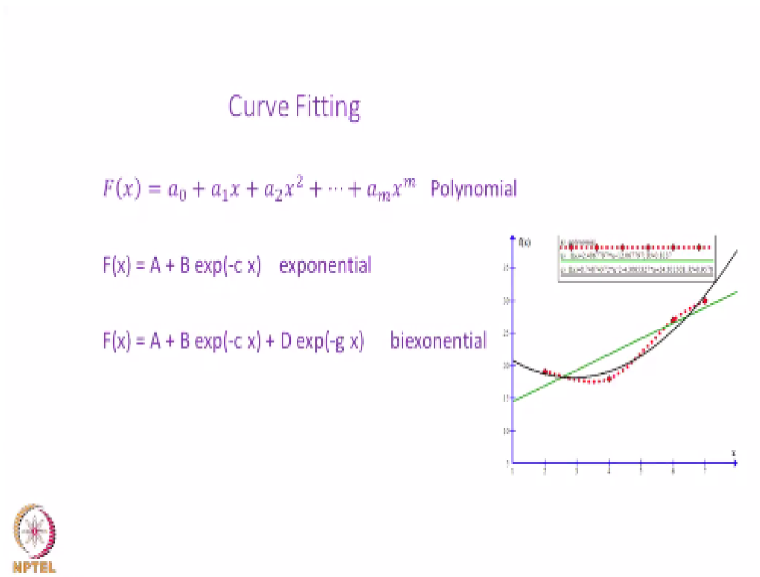
Lecture - 20
Curve Fitting, Newton Raphson Method

Hello and welcome to this session. We have almost completed 2/3 of our course and in the remaining course we will be doing curve fitting, differential equations, integration, random numbers and of course Scilab as well as molecular dynamics using our own programs. In the last class, we discussed matrices in particular the problem of inverting a matrix as well as the problem of finding the largest eigenvalue.

That was a little bit rushed because I wanted to finish it in a very short time. We will give you all the programs in the text file so that you will be able to look at it line by line compared with the algorithm and also execute. If necessary we will show in our demonstration session after one or two more lectures, details about matrix inversion as well as the eigenvalue problem. Now the matrix inversion problem becomes important because it is useful in many, many other situations.

For example, what we will show today, today our lecture is on curve fitting. How we can use the matrix inversion program to find the best fit in a linear fitting program? So now we discuss curve fitting today, so we have discussed interpolation at length.

(Refer Slide Time: 01:38)



Look at this curve; look at this graph whenever you have these data points 1, 2, 3, 4 and the red dotted curve passes through those points that is the interpolating function. But often we are not interested in a curve which goes to all the points but we are interested in the best fit a function that fits very well, so we want some kind of an analytic function because the function is far better than a large number of data points.

What are the kind; so this particular black curve this is a quadratic fit that is it is a function which is a function of x square, x and a constant, so it is a quadratic fit. So what we want to; so what is the difference between a fit and a function? The fit did not pass through all the points, but the fit is such that the error between the black curve and the red dot should be minimum we want to minimize the error that is what a fitting function is.

So let us see some examples of fitting function. The first function that we have given here Fx = a0+a1x+a2x square + am x to the power m. It is a Polynomial fit. It is a mth order polynomial, because the highest power of x that occurs here is to the power m. This is one kind of a fit. Another kind of a fit would be your function maybe a+b* exponential of –c x. This is an exponential fit. And another thing would be a bi-exponential fit. What is a bi-exponential fit?

You have two exponential functions. And what are the fitting parameters? In the first case it was a0+a1+a2 and an these are all the parameters a1 to am. These are linear parameters because they occur to the first power of those parameters whereas in this case A is a linear parameter but this c is in the exponent, okay.

So this is not a linear parameter, c in this case is A part of the exponent because when you expand an exponential you remember that it is suppose e to the x it is 1+x+x square/2 factorial; x cube/3 factorial so c occurs in all kinds of power so c is not a linear parameter. In the same way, in the last function which is a bi-exponential your A is linear parameter; B also is a linear parameter because it occurs to the first power; D is a linear parameter, but you are c and g are nonlinear parameters.

Nonlinear fitting is very complicated because there are no unique solutions. But for linear parameters there is a unique solution. And what we want to do today is to find such a unique solution for a polynomial fit. Let me take this example, okay.

(Refer Slide Time: 04:31)

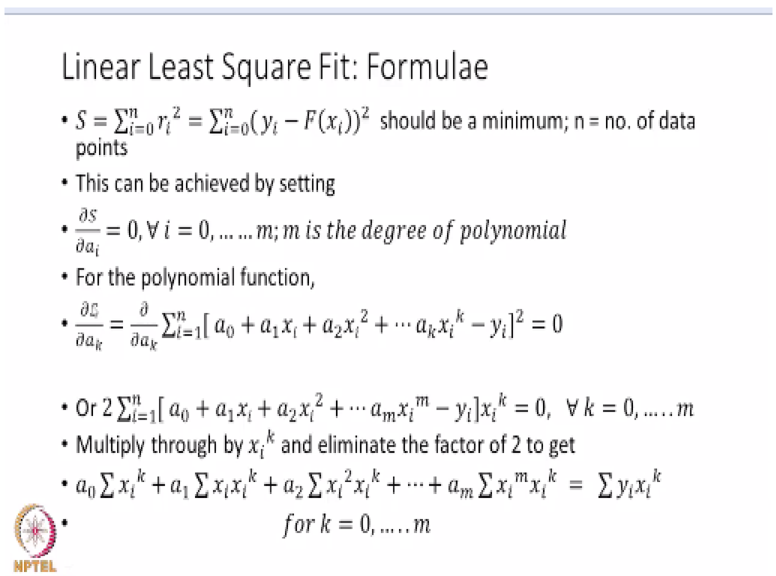
x	y	$F(x)$	$r_i = F(x_i) - y_i$
Data		Fitting Function	error/residual
x_0	y_0	$F(x_0)$	$r_0 = F(x_0) - y_0$
x_1	y_1	$F(x_1)$	$r_1 = F(x_1) - y_1$
x_2	y_2	$F(x_2)$	$r_2 = F(x_2) - y_2$
x_3	y_3	$F(x_3)$	$r_3 = F(x_3) - y_3$
x_n	y_n	$F(x_n)$	$r_n = F(x_n) - y_n$

So how are the data represented? Normally when we have a data suppose x is my independent variable and y is a dependent variable so the data will appear as x0, x1, x2 up to xn so this is the set of data. And for each of these data my independent variable y will be y0, y1, y2, y3 up to yn. So this is my given data; so through this data I want to fit a function Fx. So at the point x0 my fitted function will be Fx0 F at x1 it will be f of x1.

Similarly, for the last point it will be F of xn. So this is my fitted function. This is my original data. So to find out how good this fitted function is I will calculate the error between the fitted function and my y points. What is my error? At the first point the error would be Fx0 - y0 because Fx0 is my function and y0 is the data. At the next point it is Fx 1 - y1; at the last point it is f xn - yn so these are all called residuals or errors.

These are the errors between the fitted function and your data that is given. So the ith error or residual is f of xi – yi. So what is our strategy now? The strategy should be; how is it? I can reduce these ri's I want to minimize the ri values, okay. So if I just; so let us see how I minimize it.

(Refer Slide Time: 06:12)



So one way to minimize which is called a Least Square Fit. What is called a Least Square Fit? I sum all the square of the errors remember I had ri goes from r0 r1 r2 up to rn; I square all the ri's and that is my total error; total error is the sum of all the errors, so what is ri square? yi – Fx I square I have sum all the errors. Now we want to ensure that this is a minimum, okay. So n is a number of data points. Why am I taking ri square?

Suppose instead of taking ri square I just take ri so one of the ri can be positive and another ri can be negative, so I get a 0 error because the positive ri is cancel with negative ri if it is not a square, so therefore it is always necessary to square it because the square is always non-zero okay. Some of the squares is non-zero so I want to minimize this error sum of these errors. So that can be done by setting all the derivatives to 0. So what are my parameters?

a0 a1 a2 up to an. I had n parameters in my polynomial fit, okay. So in fact m0 n because we will have a lot of difference between n and m as we go along, so m is the degree of the polynomial. So when I fit this polynomial I will have this coefficient a0 a1 a2 up to am. So this particular sum the derivative with respect to each of those parameter should be 0 then I can expect that s to be a minimum error, so I want to set all the derivatives to be 0.

Now how do I calculate the derivatives, okay? Let us say I want to calculate the derivative of the kth parameter, so ds/d ak will be d/ d ak * S on the right side. What is my S? S is this top line here, sum of ith going from 1 to n a0+a1x+a2x square all the way up to a kx ai xk it takes all the values for all the values in which my function is defined. The function is defined at n points. So I am summing over n and taking the derivative so this should be 0.

Again I repeat m is the degree of the polynomial and n is the number of data points. So when I take this derivative; when I take this derivative this is a square function; when I take a derivative I will get two times that sum multiplied by the derivative of the kth term. So what is the derivative of the kth term? Derivative of the kth term will be suppose this; suppose this is the derivative of the second term d/d a2 will be xi square.

So d/d ak will be xi to the kth power so that comes out. So I want the derivative of this with respect to the kth; the derivative is two times the sum of all the terms which is inside the bracket * xk, so this is coming at the derivative of a xi to the k. So this derivative should be 0 for all k. So what are the values of k? 0 to m because there are m coefficients a0 to am. So there will be m+1 equation because there are 0 to m so there will be we m+1 equation.

So what are those m+1 equation? So before I go to see all the m+1 equation let us just rewrite this. How do I rewrite it, okay? So the factor of 2 is not important because 2 times 0 is also 0 so I remove the factor of 2. I take all terms involving x on the left side and terms involving y on the right side. So what are my terms now? The first term will be a0 * xi k, because see this xi to the k when I multiply, first one will be a0 * xi to the k.

So the next term will be a1 to the a1 multiplied by xi and xi to the k so that is a1 * xi * xi to the k. So actually it is xi to the k+1. Second term will be a2 * xi square * xi to the k all the way up to am last term am * xi to the m * into xi to the k; xi to the m comes from the one in the bracket and xi to that k comes from outside the bracket. So on the right side I am taking yi and xi to the k. So on the right side I am summing yi to the xi to the k.

So how many equations are there? There will be m+1 equation for all the values of the coefficients. So those equations now are for these coefficients a1 a2 up to am. Why are the equations for the coefficient? Because all the xi's are known and all the yi's are known. So knowing xi and yi I want to solve for all the coefficients a1 a2 up to am. So I can write it as a matrix equation; so that is what my next slide shows.

(Refer Slide Time: 11:27)

$$\begin{bmatrix} \sum (x_i^0) & \sum x_i^0 x_i & \sum x_i^0 x_i^2 & \cdots & \cdots & \sum x_i^0 x_i^m \\ \sum (x_i) & \sum x_i x_i & \sum x_i x_i^2 & \cdots & \cdots & \sum x_i x_i^m \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum (x_i^m) & \sum x_i^m x_i & \sum x_i^m x_i^2 & \cdots & \cdots & \sum x_i^m x_i^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^m y_i \end{bmatrix}$$

So I will write it as a matrix equation so this matrix equation so I want to solve for a0 a1 up to am; this I write as a column vector and all the coefficients I write it as a matrix. So the size of the matrix is m+1/m+1. And in the right hand side I had sum of yi; sum of xi yi and the last one is xi to the m yi. So this matrix equation I got by demanding that the square air is minimum. So I am demanding that the square of the error is minimum, so I have to solve this matrix equation.

How will I solve? I want to solve for a0 a1 up to am. So I want to invert this matrix. Once I invert this matrix I multiply this equation by this matrix inverse so matrix inverse of this square bracket into this matrix will give me identity. So what remains of the left side? Left hand side is a0 a1 up to am. On the right hand side, it will be inverse of this matrix here inverse of this matrix will come on the right hand side, so that is my solution, okay.

(Refer Slide Time: 12:39)

$$\begin{bmatrix} \sum (x_i^0) & \sum x_i^0 x_i \\ \sum (x_i) & \sum x_i x_i \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}$$

So let us take for example a simple example of a 2/2 case. What is a 2/2 case? It is a linear least square fit. What is the meaning of linear least square fit? I want to fit through by a function a0 + a1 x, so that is my straight line a0 + a1 x will be a straight line so these are all my matrix coming from the data points. This is from my dependent variable xi; dependent variable yi so if I invert this matrix so left side I multiplied by the inverse on the left side.

So this term will give me identity; so on the right hand side will be inverse of this matrix; so in other words a0 a1 this column = inverse of this matrix which multiplies this column vector which has terms yi and xi yi. So this is how I will have to do my calculations to get a0 and a1, okay. So what I want to do now I want to take an example of a cubic polynomial. So I want to fit my data with this particular cubic polynomial.

(Refer Slide Time: 13:50)

Third degree polynomial fit: Example

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
y	0.0	0.1002	0.2013	0.3045	0.4108	0.5211	0.6367	0.7586	0.8881	1.0265	1.1752

$$P_3(x) = a_1 + a_2x + a_3x^2 + a_4x^3$$

So I have taken data so this data set consists of 11 points starting from 0 0.1 0.2 0.3 up to 1, so these are my independent variables and this y is my dependent variable; I have 11 data points for 0.0 it is 0; for 0.1 it is 0.1002 and for 1.0 which is x my dependent variable is 1.1752. So all this xi and yi I want to fit with this cubic polynomial. So cubic polynomial is P3 x that is a1+a2 x + it should be a2 x square okay + a3 x cubed, okay.

So just check that it is a third order polynomial I have four coefficients a1 a2 this sorry this no; it is quite right a1+ a2 x + a3 x square + a4 x cubed. Now what we have done, in my earlier thing it was a0 a1 a2 a3 but when you actually do the program it is always good to start with integers from 1 to n, okay 1 to n. So that is why that they shift instead of a0 a1 a2 a3 it is a 1 + a2 x + a3 x square + a4 x cubed. So this is my polynomial.

(Refer Slide Time: 15:10)

$$n + 1 = 11, \qquad \sum yi = 6.023$$

$$\sum (x_i) = 5.5, \quad \sum x_i y_i = 4.28907$$

$$\sum (x_i^2) = 3.85, \quad \sum x_i^2 y_i = 3.408407$$

$$\sum (x_i^3) = 3.025, \quad \sum x_i^3 y_i = 2.8773135$$

$$\sum (x_i^4) = 2.5333$$

$$\sum (x_i^5) = 2.20825$$

$$\sum (x_i^6) = 1.987405$$

So my; to calculate the coefficient I need to calculate this entire matrix. What is this entire matrix? The first term is sum over xi, xi to the power 0 means 1 so I am going to sum from i going from 0 to i=1; so since it is power 0 each term will contribute 1 so it will be 1+1+1 up to m+1. So the second term will be xi to the 0* xi; xi to the 0 is 1 so it is just the sum of xi. So the third term will be sum of xi square.

Similarly, this is my row vector and this is my column; you will see that this matrix is symmetric. What is the meaning of symmetric? Whatever terms I have on the right side here same term will appear on the diagonal opposite. See here the last term is xi to the 0; xi to the m; xi to the 0 is nothing but 1 so it was sum of xi to the m diagonally opposite is again sum xi to the m, so this is a symmetric matrix. So I want to evaluate all these summations now.

The summations will be sum over xi to the power not sum of sum over xi sum over xi square and finally sum over why I sum over xi yi so I need to sum all these things. So all those sums are indicated on my slide here, okay; since I have 11 data points my n+1 will be 11 so that is my first term, sum of xi 0th power. Then I have sum over xi; sum over xi will be 5.5; sum over yi will be 6.023; sum of xi yi will be 4.28907; sum of xi square sum of xi cube; sum of xi square yi.

Sum of xi cube; sum over xi cubed xi then xi to the 4th power; xi to the 5th power; xi to the 6th power. So since my m was 3 the last term will be; so the last term will be xi m * into xi m. So since m was three it will be sum of xi to the 6th power. If m was 9 the last term will be sum of xi to the 18th power, so that is my last term. On the right hand side, it will be sum of xi to the highest power * yi. So that is my term, the last term will be xi cube yi.

Now I will represent this in terms of a matrix, okay.

(Refer Slide Time: 17:44)

$$\begin{bmatrix} 11.0 & 5.5 & 3.85 & 3.025 \\ 5.5 & 3.85 & 3.025 & 2.5333 \\ 3.85 & 3.025 & 2.5333 & 2.20825 \\ 3.025 & 2.5333 & 2.20825 & 1.987405 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 6.023 \\ 4.28907 \\ 3.408407 \\ 2.8773135 \end{bmatrix}$$

$$a_1 = -0.000129, a_2 = 1.004383, a_3 = -0.019651, a_4 = 0.190405$$

$$P_3(x) = -0.000129 + 1.004383x - 0.019651x^2 + 0.190405x^3$$

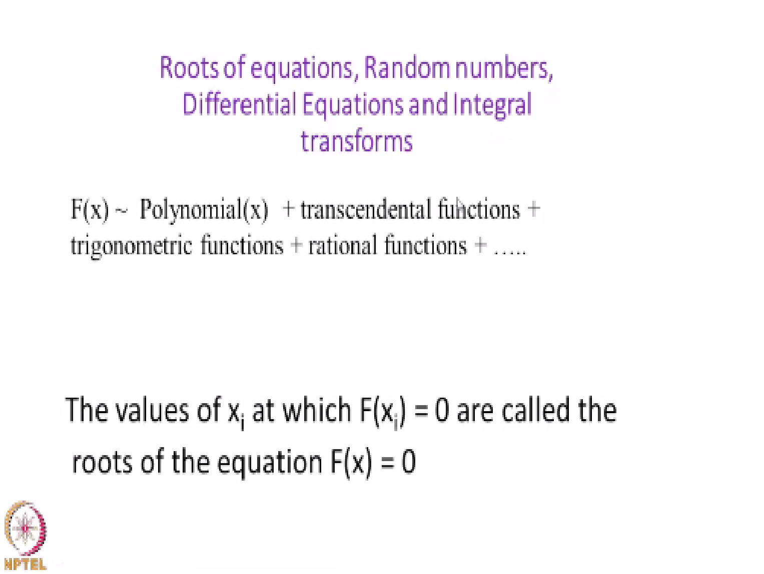
$$f(x) = \sinh(x) = x + \frac{x^3}{3!} + \cdots$$

So whatever data I showed on the last side this is my matrix; this is my vector a; this is my vector on the right hand side which has yi and yi xi summed over. So this is my matrix equation. So if I invert this matrix then that inversed matrix I operate on the right side vector so I will get a column vector a0 a1 a2 a3, so my solution would be this set of coefficient, solution will be set of coefficients a1 a2 a3 and a4.

So what we will do we will actually use the program to do this in a later class, okay. So this particular problem was chosen with Fx as a the actual function that we chose was a hyperbolic function this sine hyperbolic function is x+x cube/3 factorial + x pi/5 factorial we had taken a sine hyperbolic function and fitted that function to a cubic polynomial and these are the coefficients. We will verify this in our demonstration session.

So this is about polynomial fitting. We will not consider exponential and other fitting because the nonlinear fitting there is no unique solution. For a linear least square fitting there is a unique solution because whenever I have a matrix equation like x * a giving y. So a = x inverse y that is a solution. It is a unique solution. Whereas if it is our nonlinear parameters there is no unique solution so we will not consider it in our course so far.

(Refer Slide Time: 19:22)



So now next what I want to do I want to do the remaining topics I will introduce you today to the remaining topics. What are the remaining topics we will do in the numerical methods one would be the roots of equation? Whenever you have an equation we want to find a root that is a root of the equation. Then we also want to discuss random numbers because random numbers are extremely important when you do simulations.

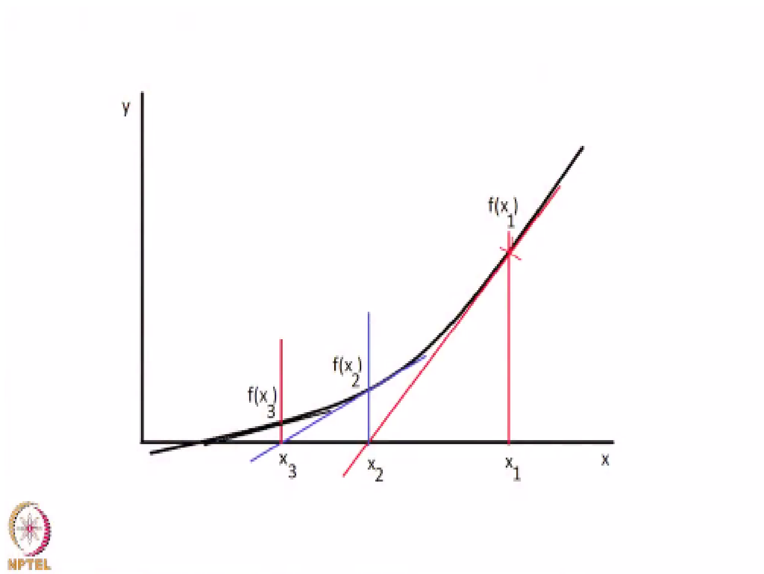
For computer simulations you need random numbers very, very frequently. So we will discuss algorithms for random numbers. Then we will also give brief introduction to differential equation and also of course when you do differential equations you have to do integrals also, so differential equations integrals and integral transforms, okay. So now the present discussion will be on roots of equations. So what are equations now?

My function Fx it can consists of a polynomial; it can consist of a transcendental function, transcendental functions are your sine hyperbolic function is an example of transcendental function, okay. You can have trigonometric functions cos, sine and so on. You can have rational functions. What are rational functions? One polynomial divided by another polynomial. So your actual function can be a very complicated function.

So what we want to find out is that at what values of x the function goes to 0. So we want to find the 0s of this function that is the whole problem of roots of equation. What are the 0s of equation? The 0s are those values of xi at which the function goes to 0. These are called the roots of the equation Fx=0, okay. So for polynomial equation suppose it is an nth order polynomial we know there are n roots. So for many situations so we know exactly how many roots are there.

In some other situations you may not know how many roots are there but we want to find out as many roots that we can find, so we want to write a program for this but before we write a program we want to find out what is the strategy for finding the roots of this equation, that strategy I will show here, okay.

(Refer Slide Time: 21:29)

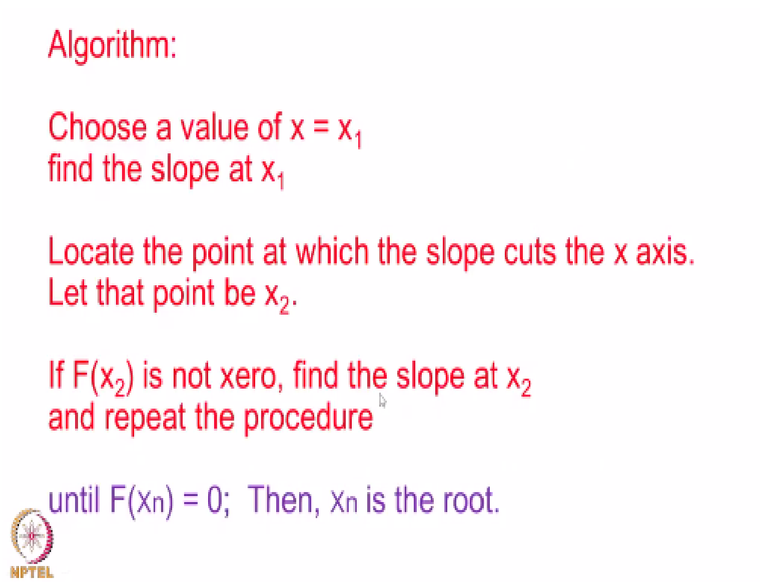


This strategy; this is called a Newton+-Raphson method. The strategies suppose I have a function Fx so I want to find the root of this function. So I start at x1, x1 can be any value so at x1 the value of the function is f of x1, so at this point I draw a slope to that curve; when I draw a slope to that curve that slope will touch the x-axis at some point x2. So if it cuts the x-axis at x2 at x2 I will find my function again this is Fx2.

So since Fx2 is at a different point the slope at Fx2 is different from the slope Fx1 unless it is a constant function where the slope is the same the slope will vary from point to point. So at Fx2 I will draw another slope which is a tangent to this function at x2. I draw another tangent it will cut the x-axis at x3 now at x3 I find Fx3 I draw another tangent, now if the tangent actually cuts the x-axis and the function also is 0 so that means x3 is your solution to Fx=0. So x3 is a root.

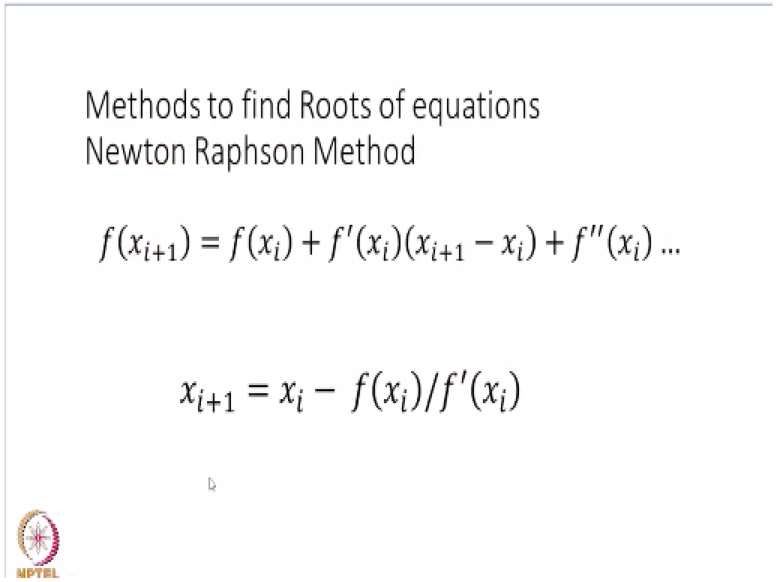
So this is the strategy. At each point I find a tangent; go to the value of x2; obtain the function at x2; find a slope tangent at that and keep continuing until it converges this is the method, okay.

(Refer Slide Time: 23:01)



So that method is outlined here, okay. Choose a value of x let us say x1. At x1 find the slope, okay. So then locate the point at which the slope cuts the x-axis, so we call that point x2. And at x2 find Fx2, if Fx2 is 0 the problem is already solved then x2 is a root; if Fx2 is not 0 then find the slope at x2 then again find function at x2 repeat the procedure until Fxn=0. So that is the root Fxn=0 is a root. So let us see the algorithm for this.

(Refer Slide Time: 23:42)



The algorithm is given in this form it is called the Newton-Raphson algorithm. So it is called an algorithm to find the root. So what we want to do we want to expand the function at xi+1 in a Taylor series about Fxi. We want to expand the function at a new point in terms of the value of the function at a neighboring point. So when I expand what do I get? The first term would be Fxi, the next term will be derivative of the function at xi this is the first derivative at xi * xi+1-xi.

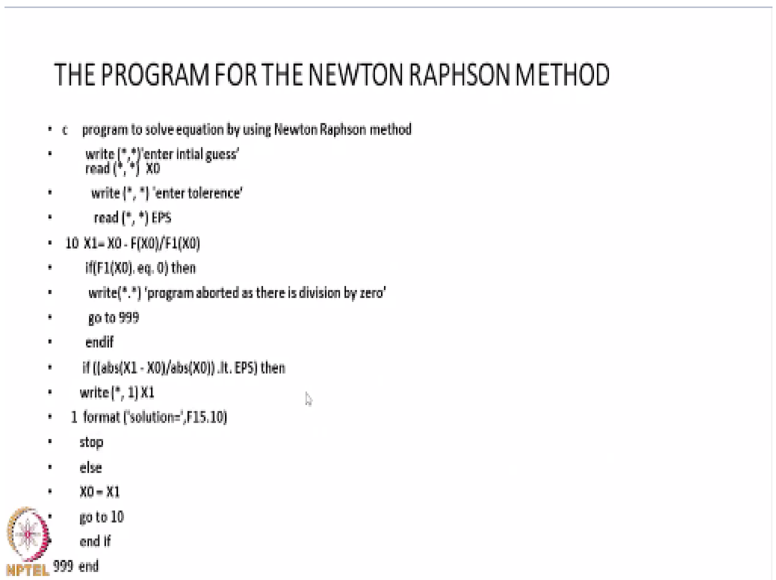
Then the second derivative of the function at xi * xi+1 – xi square/2 factorial then the third term will be the third derivative divided by 3 factorial * xi + 1 – xi cube and so on. So there will be a large number of terms. But we want to truncate this term only up to the second term. So we want to assume that all the higher terms are 0. So when you assume that, so what remains is? Only these terms, and if Fxi+1 is 0 suppose Fxi+1 is a root suppose this left term was 0; if the left term was 0 then this is 0.

I have excluded all the terms of the higher-order derivatives, so what is left is Fxi+ F prime xi * xi+1-xi=0. So since this is 0 now I will take F xi to the left side, okay. So then what will I have F prime xi * x-xi+1-xi=Fxi then divide by F prime xi okay. So since I have taken Fxi on the left side it will be minus, so when I re-express these two terms the sum of these two terms = 0 when I re-express then I have xi+1 will be xi-F xi/F prime xi.

So this is now an algorithm. Why is this an algorithm? New values of Fx; Fxi+1 is expressed in terms of the old value minus the function divided by its derivative. So this is my algorithm. So what is the strategy? Strategy start with a value of xi, determine the new value of xi+1, repeat go on repeating until xi+1 is no different from Fxi. If there is no difference that means I have reached the solution. So that is my algorithm.

So let us briefly describe the algorithm for this particular program now. I will use this algorithm to derive the program for this function okay.

(Refer Slide Time: 26:33)



Again it is an extremely simple program. So what we need to do for this program is calculate the function its derivative, okay. So this is the first line program to solve the equation by using Newton-Raphson method. Enter the initial guess, so I want to guess initial guess X0 I enter okay. I enter initial guess okay. Then, next line will be enter the tolerance, what is this tolerance. Remember at each iteration, I will be getting new values of X.

So if the new value of x is very close to the old value that is xi+1-xi is very small, I want to stop the program. So this epsilon is just the tolerance in my calculation, tolerance for finding my function. So then the next line would be I have already read x0 then I calculate X1, what is X1, X0 – FX0 /F1 X0, F1 is the derivative, F is the original function. So there is one problem here, okay.

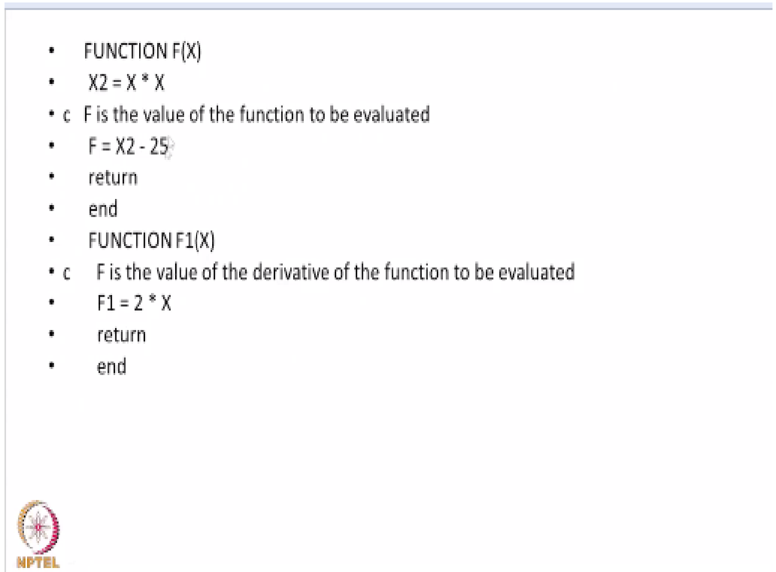
So whenever you want to divide by this F1 X0 if F1 x0 is 0 you will be dividing by 0 and there will be a problem, okay. So you need to avoid dividing by 0 so therefore you need to give this particular line if F1 X0 = 0 then write program aborted as there is a division by 0 go to 999, 999 is usually the last line, okay and then end the program. If you are dividing by 0 you do not want to do that so you want to end this program.

So therefore, this particular set of lines if write, go and end if this should come actually before this 10 because if I divide by 0 before checking I will get an error so these 4 lines 1 2 3 4 should be above this line 10 = X1= X0 10; 10 line number 10 X1 = X0 - F X0 by FX 1. So put these four lines above this line 10. Then what it does? If absolute value of X1-X0/absolute X0 < epsilon. Suppose the difference between the new value and the old value is less than that epsilon then that is my solution FX 1, okay so right X1, sum format, stop.

If the difference is not small then my X0 is set to X1 okay so the new value of X0 is X1; go back to line 10, correct? So I have replaced X0/X1 so new X1 will be the old X1- F at that value divided by F prime. So it will keep on repeating for new values each time I iterate my X1 my X0 is set to the new value; keep on doing until my X1 and X0 do not differ any more. So then that will end the program.

So the only thing that is remaining here is how I calculate FX and FX0. F1 F and F 1 I want to calculate. These are functions that is given in the next slide.

(Refer Slide Time: 29:48)



So function F(X), I am defining a function. I have taken a very simple function X square function I have taken as X square, okay. So F is the value of the function to be evaluated, okay. So what is my function? Function would be X square – 25, this is my function. So I calculate X square, evaluate the value of the function, return, so what is the derivative of this X square- 25? Derivative is nothing but 2X. So my function F1 X is its derivative, correct?

So F1 is 2 X, return. So I calculate the function; I calculate the derivative and this is used in the main program to calculate the root of that equation. So I would urge you to try to this try to run this program instead of X square - 25 try some other function like a cubic function say, X cubed - 4 X4 + X to the power 1 + 0, so try a cubic function and see whether you get the roots. So once your F is a cubic function so the derivative will be derivative of the cubic function.

So I shall conclude my lecture here. So what we have done today, we did curve fitting using a polynomial function and we also did a procedure to find the roots of equation. So we have used two different numerical methods in this lecture. So practice this then next time we will take up integration and derivatives. And after one or two more lectures we will do all these problems on the compute. So I will conclude here, thank you.