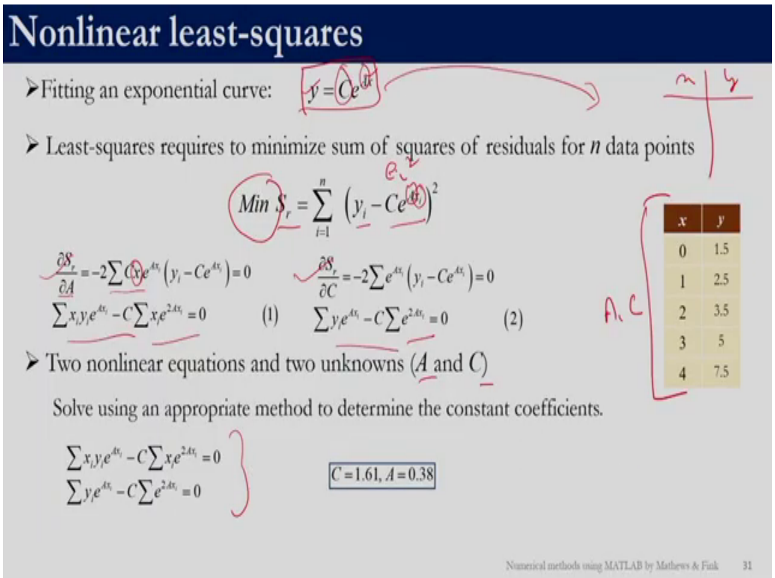
**Computer Aided Applied Single Objective Optimization**
**Dr. Prakash Kotecha**
**Department of Chemical Engineering**
**Indian Institute of Technology, Guwahati**

**Lecture - 04**
**Nonlinear Regression**

In this session we will be looking into Non-linear Regression. So, non-linear regression can be done by two methods; one is using non-linear least squares optimization right, the other one is for some models, we can employ data linearization technique to convert the non-linear model into a linear model we will be looking into both of them. So, cases wherein it is possible to convert a non-linear model into a linear regression problem using data linearization we will do that right. For cases wherein it is not possible to convert into linear regression we will employ a non-linear least square regression.

(Refer Slide Time: 01:07)



So, now let us look into the non-linear regression right. So, let us say we are given this x and y data. Let us say we have given we are we have been given this x and y data and we are required to fit this model y is equal to C e power A x.

So, y is given x is given right. So, now, we need to fit this model. Basically we need to find out the what is the C value and A value such that this model would best represent the data that we have. So, we apply the same concept again right. So, sum of square of residual. So, your observed data point is y i the model is C e power A x i where x i is the value of the ith value of the independent variable. So, this is the error associated with the ith point, square it and then sum it up. So, that is our objective function right. So, now, our job is to find out the value of C and A such that if you do dou S r by dou A and equated to 0 similarly determined dou S r dou C and equated to 0 again over here you need to carefully differentiated.

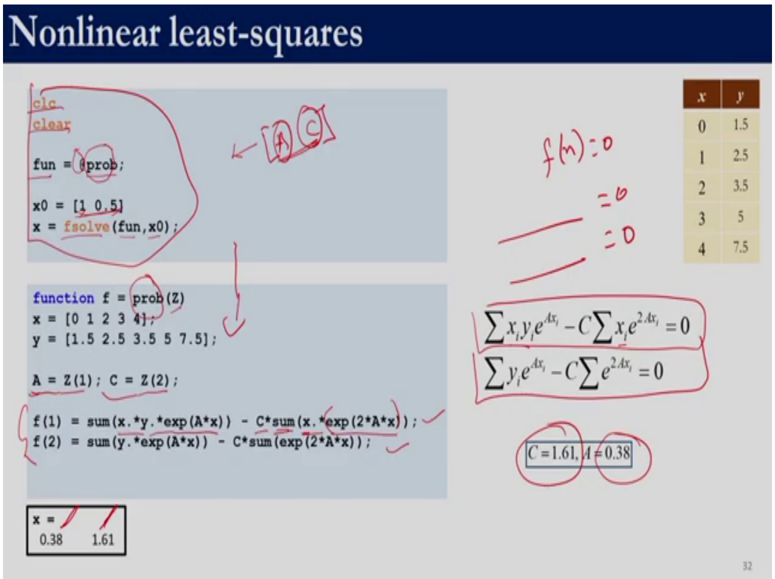
So, two times this expression and then you are differentiating with respect to A. So, x i would be a constant. So, x i e power A x i. So, remember the differentiation of e power m x is m e power m x right again here remember ms the constant. So, in this case x i is known right. So, what is unknown? Here was x we were differentiating with respect to x, here we are differentiating with respect to A. So, that is why you have this C into xi not A right.

So, this if you calculate dou S r by dou A and dou S r by dou C you will end up with two equations equated to 0, we will end up with two equations and in this case these two equations are not linear and those are non-linear equations right in terms of A and C. So, you can apply any technique that you know for example, Newton Raphson method to solve those equations right.

Two equations in (Refer Time: 03:20), if you solve them you will get the value of A and C that is how we can fit the model. So, for example, if this is our data set right and if you solve these two equations right we will end up with C is equal to 1.61 and A is equal to 0.38, right. We will not be showing you how to exactly solve this non-linear equation because we assumed that you would have done that in your first year of engineering right.

So, we will quickly show you how to do that in matlab right.

(Refer Slide Time: 04:00)




So, for those of you who are for those of you who are new to matlab we will have a separate video where you will be able to understand the same thing how to solve non-linear equations over here we will quickly rush through. So, the clc as you know will help you to clear the command window, clear will remove all the variables, clear the variables from the matlab box space right.

So, here we are defining a function handle. So, the name of the function handle is fun f u n right it is a function handle right. So, prob is the function. So, we have the problem is going to be stated in a separate function file known as prob right and that name of the name of that function is being assigned to this variable fun. So, that is why we have the symbol at the rate symbol over here. So, fun is now a function handle. So, whenever we access fun, we will be actually accessing this function prob right. So, for solving a non-linear equations in matlab, you can use this fsolve command right. For fsolve we need to give the equations that we want to solve and the initial starting point. So, here we have two variables.

So, here we are going to arrange in this format that the first variable is going to be A and the second variable is going to be C, you could choose the other way, but we have chosen this way and we will be consistent it throughout, will be consistent it with throughout right.

So, we are giving a guess value of a starting value of A to be 1 and the starting value of 0.5 to C. So, these are the two in the function file, we are going to return the value of the two equations right. So, for solving for using fsolve, the equation should be in this form f of x is equal to 0. So, if you have two equations both the equations right and side has to be 0. So, whatever we have on the left hand side is to be given to the equation right. So, remember this is our equations these are the two equations that we have right.

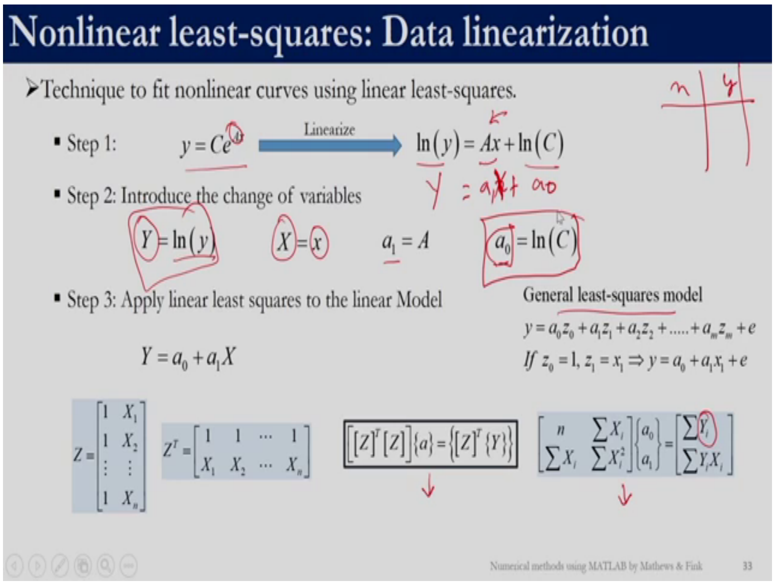
So, z is what we will get from fsolve, we need to solving right. So, the first the first value of z would be A because that is the notation which we have taken and that is the order in which we have given the starting point right. So, the first value of z is going to be A, the second value of z is going to be C right. So, once inside the function we have decoded it and then these two expressions are nothing but the left hand side of these two equations right. So, here if you see x is a vector, y is a vector.

So, we use this dot operator x dot star y and this e power A x i is going to be n times for n values of x. So, again a dot operator, dot star exponential of a A star x minus C is a constant right and the because of the summation we have the sum. So, sum function will help us to sum all the values right. So, this x i is this x vector x right this vector x and then we have this e power 2 A x i.

So, exponential of 2 into A into x and since this is going to be a vector again, we use this dot operator similarly we have written the second function right. So, this is the first function, this is the second function now if we run this code right it will repeatedly it will access this function file multiple times right and at the end of it, it will give us these two values x is equal to 0.38 and 161 right. So, since we had we had used this convention that the first value is A, the second value C A is equal to 0.38 and the C is equal to 1 1.61 right. So, those of you who do not know matlab maybe this part you may not be comfortable with this part, but that is fine.

The rest of the discussion is not going to be dependent on this one. So, it is totally fine if you have not got this matlab section.

(Refer Slide Time: 08:12)

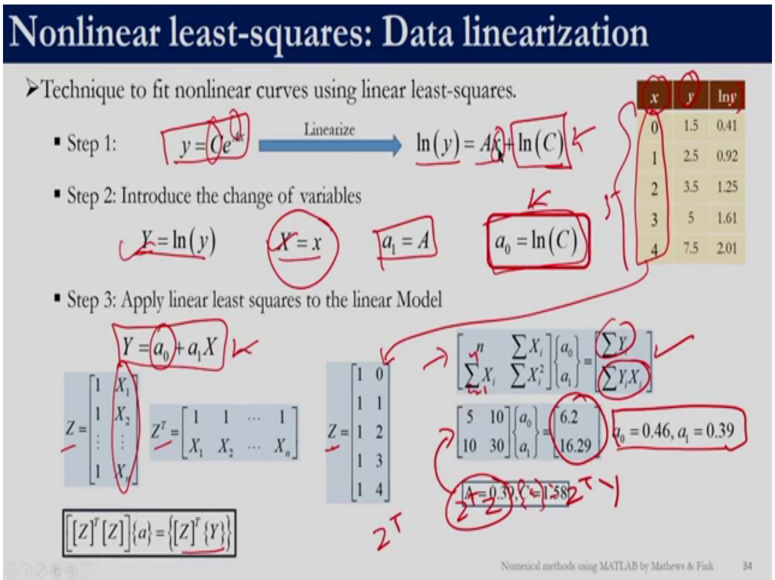



Another way to solve this non-linear regression problem. So, sometimes this is possible to do right. So, our model was y is equal to C into a e power A x right. So, what we can do is we can take lawn on both sides. So, lawn y equal to lawn C plus A x lawn e right. So, lawn A is 1. So, we will have only A x. So, now, if we see this model right. So, y is known right. So, in this case remember x and y were given. So, y is known. So, you can. So, we can calculate ln of y right. So, let us call that as capital let us denote it with the uppercase y. So, if this lower case y is given, we can calculate the upper case y using this transformation like upper case x be the lower case x right.

So, what we are doing is we are equating it to y is equal to this is a naught plus a 1 x right. So, we are basically plotting we are converting into a linear regression problem right. So, the upper case x is nothing, but the values which we get directly only the y has to be transformed and when we solve what will get us a naught and a 1. The a 1 would directly correspond to this a in our model whereas, the a naught which we get is not is not does not represent c, but we need to take the exponential of a naught to actually get the value of c right. So, here what we are doing is, the non-linear model which we have we are doing data linearization to convert into a linear form and then we can apply the same general linear least squares which we discuss right.

So, in this case four y is equal to a 1 X plus a naught the z value is nothing, but 111 right because we have a constant coefficient and the second vector is nothing, but the x values right and z transposes the transpose of this right. So, we can either use this to solve or we can use this to solve right. So, in this case remember this is capital Y, summation of summation of uppercase y and upper case y is to obtain from this transformation similarly after solving we will get a naught and a 1 to get C we need to again use this expression.

(Refer Slide Time: 10:38)




So, let us consider that we have been given data let us consider we have been given this data right. So, there are five data points where x is the independent variable, y is the dependent variable as of now ln of y is not given right.

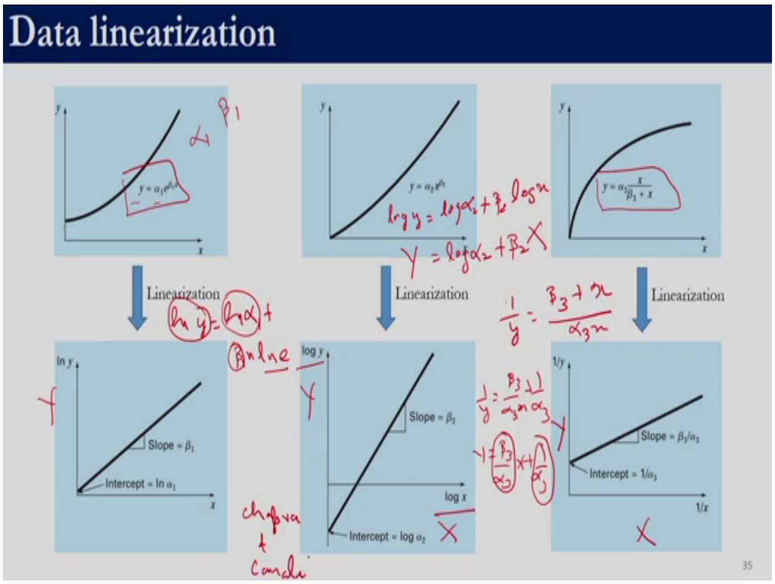
So, ln y is not given, we will come to that why we are finding ln y right. So, the model that has given is y is equal to C exponential A power x right. So, either we can solve it using non-linear regression or what we can do is we can take lon on both sides right. So, ln y is equal to Ax plus ln C. So, ln y can be represented by capital Y right. So, we will use upper case Y has l n y, X let us say is the same variable x. So, we are basically comparing it to Y is equal to a naught plus a 1 X right. So, here if we see this is a constant term right this is the independent variable, it has a constant right this is a dependent variable term right.

So we can calculate Y we can calculate X. So, a 1 is nothing, but capital A and a naught is ln C. So, we are comparing this model with this model. So, if we use this variable transformation, this model will boil down to this one right. So, using general linear square regression, we know that the solution for this problem is z transpose z into a is equal to z transpose y where z is column of 1s right because we have a constant coefficient term and the X 1 variable stacked all the end points stacked like this as second column right. So, and then once we know z, we can calculate the z transpose similarly we can calculate z transpose y right. So, if we apply this solution to this problem right. So, our z matrix will be a column of 1s there are five data points. So, five rows.

And then this independent variable is to be directly stacked over here because we do not need any transformation of that right. Upper case X is nothing, but the lower case x once we have z we can calculate what is z transpose and we can calculate what is z transpose z right. So, z transpose z in this case would be this and z transpose y would be 6.2 and 16.29 right or we could have employed this one n sigma X, sigma Xi, sigma Xi square i varying from 1 to n right sigma Y sigma Y i Xi. So, either we can use z transpose z into z transpose into a is equal to z transpose y or we could have used this in both cases we will get the same coefficient matrix and right hand side vector right. So, if we solve that we will get a naught is equal to 0.46 and a 1 is equal to 0.39 remember what we are interested is in the C and this upper case A right.

So, now that we know a naught right. So, we can calculate what is C using this relation that we know that a naught is equal to lnC.

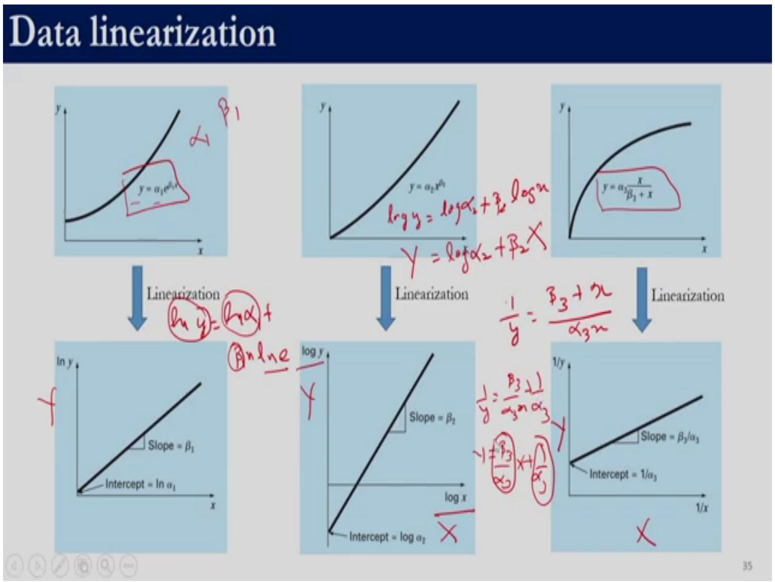
(Refer Slide Time: 13:50)




So, C can be calculated from that expression a naught is equal to l n C right. So, C we can calculate and a one is nothing, but the value of the constant coefficient a in this original model right. So, a is directly equal to a 1 because of this transformation. So, a is equal to 0.39 C is equal to 1.58. So, our model is y is equal to c 1.58 e power a x 0.39 x. So, this is our model right. So, what was a non-linear model? We use data linearization to convert into a linear model right in this case it happened to be simple linear least square right.

So, we can either use this formula or we can use that analytical solution z transpose z into a is equal to z transpose y and solve for the coefficients of a right and then if there is a data transformation, we need to appropriately do the data transformation to get the original model coefficients. So, this is an example for non-linear least squares using data linearization.

(Refer Slide Time: 15:02)



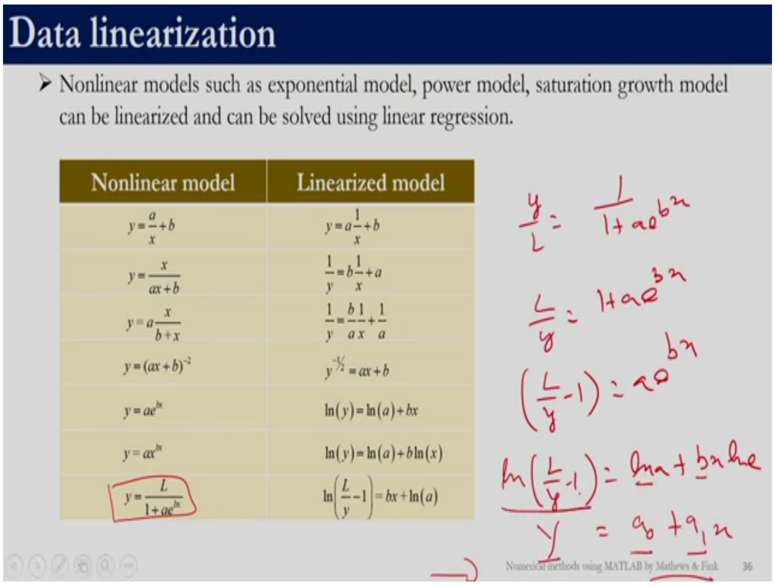

So, some of the other examples for linearization these are three other examples wherein we can linearize. So, the actual model is y is equal to alpha 1 e power beta 1 x right. So, x and y is given the our task is to find out alpha 1 and beta 1.

So, if you transform this. So, if you take lan on both sides right in this expression if we take it will be l n y is equal to l n alpha 1 plus beta 1 x ln e right. So, this is going to be 1, the regression that we are doing is between ln and y right. So, let us say that is this upper case y; l n alpha l n alpha 1 will be the intercept right and the slope will be beta 1 right and there is no transformation in x. So, if we are to solve this model this non-linear regression problem, we can actually do a data linearization and actually work with l n alpha and x right. Over here in the second model if you see it is y is equal to alpha 2 x power beta 2. So, here we can take log y is equal to log alpha 2 plus beta 2 log x right.

So, log Y let us say it is uppercase Y, log alpha 2 will be our intercept plus beta 2 would be our slope log X let us say it is upper case Y. So, we perform a regression between log between log X and log Y or upper case X and this upper case Y. So, the regression that we are doing is in the in this in this space over here. So, over here what we can do is, we can take a reciprocal. So, if the model is like this y is equal to alpha 3 x by beta 3 plus x, we can take a reciprocal of this right. So, if you take a reciprocal of that to what we will have is 1 by y is equal to beta 3 plus x divided by alpha 3 x right. So, this will be 1 by y is equal to beta 3 by alpha 3 x plus 1 by alpha 3 right.

So this will be one by y let us say it is a uppercase Y, 1 by x let us say beta 3 by alpha 3 is let us say 1 by x is upper case x right plus 1 by alpha 3. So, this will be our intercept, beta 3 by alpha 3 will be our slope right. So, here we are working with upper case x and upper case y where upper case x is 1 by x and upper case y is 1 by y. So, this is one way wherein we can convert a non-linear regression problem into linear regression problem right. So, some of the other examples are given over here right.

(Refer Slide Time: 18:04)




So, these are taken from Matthews and finks. So, all these figures have been taken from Chapra and Canale numerical methods by a Chapra and Canale. So, these are some of the other examples for non-linear models which can be converted into linear.

I would leave it to you for you to figure out how to do it right you can look into this book numerical methods using matlab by matthews and finks, it has a few more examples right. So, for example, this particular model if you see a what you can do is y by L is equal to 1 by 1 plus a e power bx right. So, we can take a reciprocal of this L by y is equal to 1 plus a e power b x L by 1 minus 1 is equal to ae power b x. So, ln of L by y minus 1 is equal to ln a plus b x l n e right. So, in this case the unknown coefficients are a and b right l is known l is given right. So, this can be calculated right this is let us say a naught plus a 1 x.

So, a 1 is nothing, but b a naught is nothing, but l n and y is nothing, but this expression right. So, we know how to solve this both linear and non-linear regression right. So, for our linear regression once we applied the stationary conditions, the non-linear optimization problem got converted into solving a set of simultaneous linear equation. For the non-linear models that we discussed as part of the session, it turned out it got converted into a set of simultaneous non-linear equations right. So, and we also have seen that some of the non-linear models can also be converted into a linear model using data linearization.