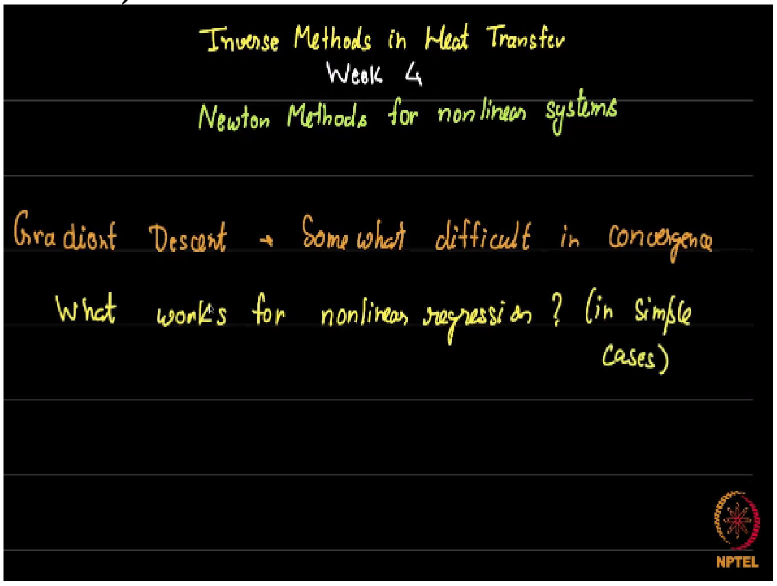
Inverse Method in Heat Transfer

Prof. Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology – Madras

Lecture - 25
Newton Algorithm for a System of Equations

(Refer Slide Time: 00:18)



Inverse Methods in Heat Transfer
Week 4
Newton Methods for nonlinear systems

Gradient Descent → Somewhat difficult in convergence

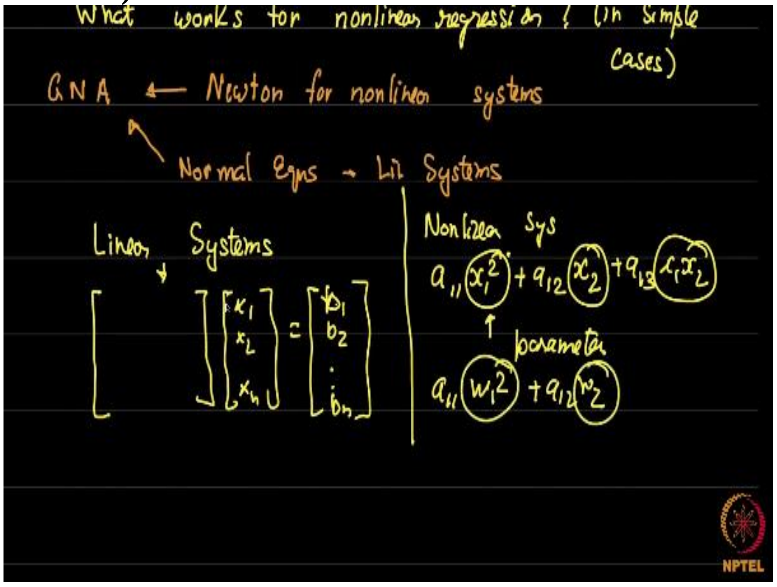
What works for nonlinear regression? (in simple cases)

Welcome back, we are in week 4 for inverse methods in heat transfer. In the previous video, we saw that when you try to apply gradient descent to nonlinear regression at least the problem that we looked at. So, this is somewhat difficult to make it converge. So, we got some answers far away from physical answers etcetera. There are other problems that tend to happen in gradient descent, which I will discuss more when we come to week 9 or so within this course.

But gradient descent is not ideal for the simple inverse problems that we tend to do in heat transfer, it is more, it is actually better provided you can play a little bit with it for actually ironically did more difficult nonlinear problems, because it is harder to do some of the techniques that we are going to use in such problems they gradient descent is the only one that scales in that case. So, we are going to define something that works, what works for nonlinear regression.

So, remember, we are trying to simply minimize a function a square function of the loss. So, we are trying to find out what works for nonlinear regression, especially in somewhat simple cases. So, there are a couple of algorithms that work here and we are going to look at 1. one is called the conjugate gradient method to promote probably we would have the time to cover that with an output. So that is a popular way to but we are going to look at what is known as the gauss newton algorithm.

(Refer Slide Time: 02:13)



What works for nonlinear regression? (in simple cases)

GNA ← Newton for nonlinear systems

Normal Eqns → Lin Systems

Linear Systems

$$\begin{bmatrix} & & \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Nonlinear Sys

$$a_{11}x_1^2 + a_{12}x_2 + a_{13}x_1x_2$$

parameter

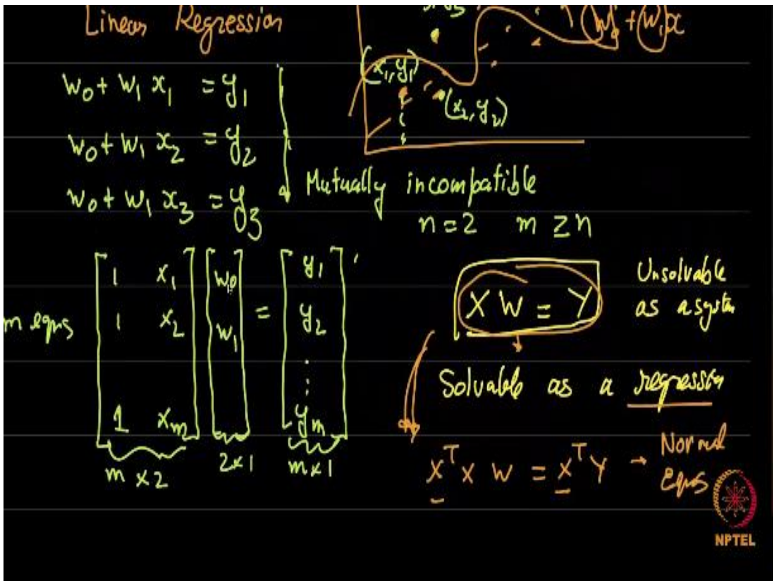
$$a_{11}w_1^2 + a_{12}w_2$$

But before going to gauss newton algorithm, so before going to GNA, as we call it, we are going to have a few steps really speaking, the origin of GNA the way I am going to derive it is from 2 places, one is the normal equations for linear systems, this was powerful as you remember. And the other technique is what is known as the Newton Raphson or the Newton set of equations for nonlinear systems.

Now, nonlinear systems are just an extension of linear systems of equations. So, please remember the difference between linear systems versus nonlinear systems. So, linear systems are like this or also remember this difference between linear systems versus linear regression. So, linear systems are something like you have a matrix and you have x_1, x_2, up to x_n. And you have some right-hand side let us call it b_1, b_2, up to b_n so this is a linear system.

A nonlinear system is something like you could have $a_{11}{x_1}^2 + a_{12}x_2 + a_{13}x_1x_2$, so these are the variables we are solving please remember. And these are the parameters that we have solving form. In such case, in such a case this is of course, non-non-linear in x_1 x_2, some of you might be more comfortable now, if I call this have $a_{11}{w_1}^2 + a_{12}w_2$ etcetera. So, in such a case it is nonlinear in w so this is a nonlinear system.

(Refer Slide Time: 04:10)



But look at a third case, which we have already solved, which is linear regression. Linear regression was the problem that we had these data points and none of them actually were on a line, but we fit a best fit line. But there is another way to see this linear regression system, it is like this. So, suppose the line we are trying to fit is $w_0 + w_1 x$. And again, right now, I am using w_0 and w_1 as given knows.

So, 1 way to say is if this point is the point $x_1 y_1$ and this is the point $x_2 y_2$ we are trying to solve 2 incompatible equations. So, for example, I am trying to say that $w_0 + w_1 x_1$ should be y_1 at the same time $w_0 + w_1 x_2$ should be y_2 you could solve these 2, but I have a third point, which is $x_3 y_3$ and I am trying to solve $w_0 + w_1 x_3 = y_3$, these seem like mutually incompatible conditions. Now, that is a beautiful argument through linear algebra, where you can show that you can think of this as solving an overdetermined system

So, here you have only 2 unknowns, w_0 and w_1 and then you have,

$$\begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ . & . \\ 1 & X_m \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ y_m \end{bmatrix}$$

you have only 2 variables. So, n = 2 but you have m equations, and m is much greater than n. Now all these we are trying to match y1, y2 ... y m, note that this multiplication is a valid multiplication because this is m cross 2 this is 2 cross 1 and this is m cross 1. So, this multiplication is well. Now, if you look at it this will start looking like a set of equations,

$$XW = Y$$

This is what we want to solve but obviously this is unsolvable as a system. But this is solvable as a regression problem. As you do not solve any of these equations exactly but you solve them in an approximate sense. Now, some of you might notice that, strangely enough, it looks like our normal equations have something very close to this. If you look at the normal equations, the normal equations for this are,

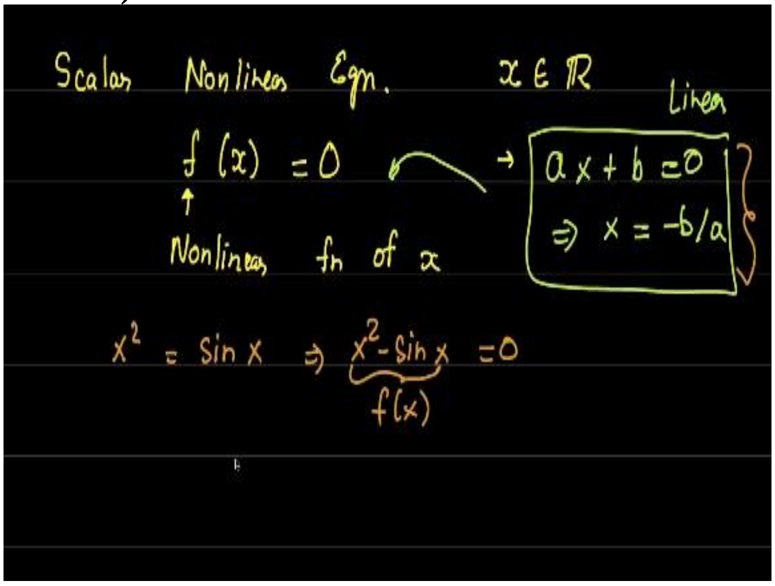
$$X^T X W = X^T Y$$

This is actually this solution to this regression problem it is not a solution to the original problem.

Because that problem is unsolvable, there is no line, which will go through all these points, but the best fit line looks remarkably like this equation, except with an X transpose multiplied upfront okay. So, something like this will happen when we come to nonlinear regression also. So, first what we will do is, we will gain intuition for nonlinear regression by solving the nonlinear system problem, which is a well-defined problem that is the number of equations equals the number of unknowns and then we come to the nonlinear regression problem.

What would the nonlinear regression problem look like? Physically it will look like this, we are trying to fit a best fit line, which is very curvy and it is not just a polynomial, it is actually nonlinear within the parameters itself. So, we will find out that algorithm is called the gauss newton algorithm within this video, we will simply look at a nonlinear system of equations where the number of unknowns is equal to the number of variables solve that and then we will extend that to a nonlinear regression problem in the next video.

(Refer Slide Time: 08:20)



Scalar Nonlinear Eqn. $x \in \mathbb{R}$

Linear

$f(x) = 0 \rightarrow$ $\boxed{\begin{array}{l} ax + b = 0 \\ \Rightarrow x = -b/a \end{array}}$

$\uparrow$

Nonlinear fn of x

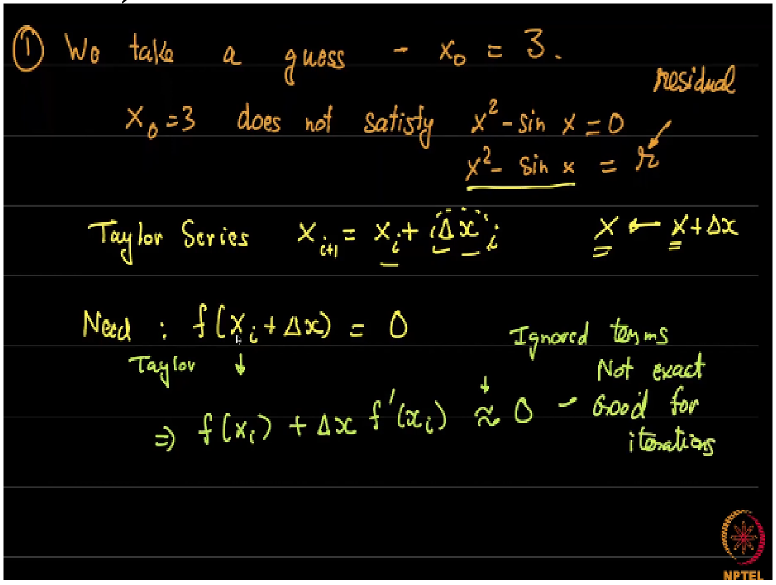
$$x^2 = \sin x \Rightarrow \underbrace{x^2 - \sin x}_{f(x)} = 0$$

So, let us look at the first simplest case, which is the scalar nonlinear equation. What is a scalar nonlinear equation? It means x the unknown is just 1 single variable; it is a real number. And we have an equation like $f(x) = 0$, where f is a nonlinear function of x if it was linear, it is straightforward. So, for example, if $f(x)$ was linear, it would look like $ax + b = 0$, then of course, we can directly solve this as $x = -\frac{b}{a}$.

Now, this trivial thing I am pointing out for a specific reason, notice how our knowledge of this linear solution will now be used to solve this nonlinear equation using the Newton Raphson method. Now, all of you would be familiar with the Newton Raphson method. Nonetheless, I hope I am going to give you at least a slightly new perspective on this, look at it as a way to solve a nonlinear equation using our knowledge of how do we solve linear equations? Okay, So, we want to solve so let us take an example.

Let us say we have an equation something like $x^2 = \sin x$, then I will write it on one side, I will write this as $x^2 - \sin x = 0$ and this then becomes $f(x)$. Okay? It is a clearly a nonlinear function of x. Okay, how do we solve?

(Refer Slide Time: 09:55)



① We take a guess - $x_0 = 3$.

$x_0 = 3$ does not satisfy $x^2 - \sin x = 0$

residual

$\underline{x^2 - \sin x} = r$

Taylor Series $x_{i+1} = \underline{x_i} + (\Delta x)_i$ $\quad \underline{\underline{x}} \leftarrow \underline{\underline{x}} + \Delta x$

Need : $f(x_i + \Delta x) = 0$

Taylor ↓

$\Rightarrow f(x_i) + \Delta x\, f'(x_i) \approx 0$ - Good for iterations

Ignored terms

Not exact

We know that we take a guess Okay, so please notice the within the next 3 algorithms, we point out this step will always be the case we take a guess let us say X_0 equal to some value 3 okay. Now, obviously, at X = 3 does not satisfy they should remind you of how we were searching for our parameters of gradient descent. Also, it does not satisfy $x^2 - \sin x = 0$. In fact, you will get $x^2 - \sin x = r$, where r is something, we will call a residual, that is, this remains for our guess.

Now, how do we proceed? This is where we use Taylor series, you can interpret what we are doing geometrically all of you would have seen this, I am going to purely use Taylor series, because that is what lets you extend this to nonlinear systems as well as to nonlinear regression. So, Taylor series, Taylor series says, Okay, let the,

$$X = X + \Delta x$$

Okay, so notice this, we have a nonlinear equation in x, notice what will happen to what the equation becomes in Δx.

So let this new X be, if you wish, you can call this,

$$X_{i+1} = X_i + \Delta x_i$$

But I am generally going to use the computer science notation or the programming notation that $X + \Delta x$ goes to X should be understood that this is the next iteration. This is the previous iteration. Okay? Now, what I want is that even though at X_i is not 0, i would like that,

$$f(X_i + \Delta x) = 0$$

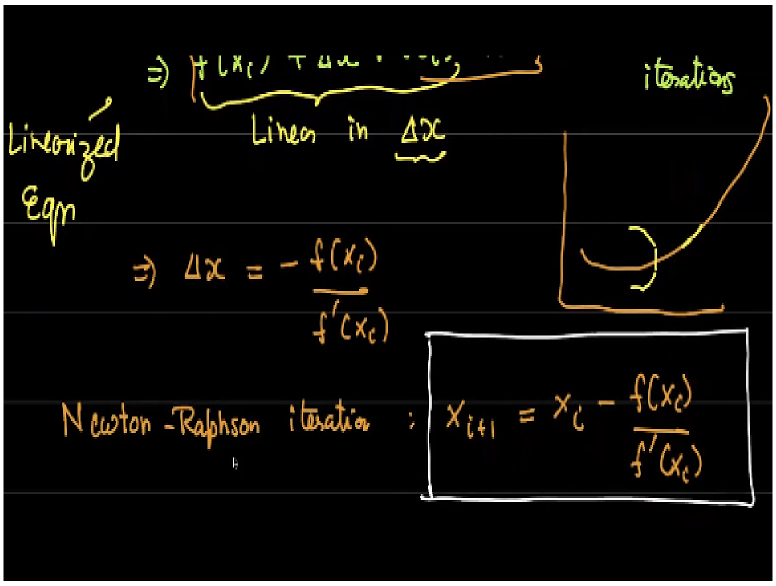
Okay, now we expand this equation on Taylor series.

And say, this means that,

$$f(X_i) + \Delta x f'(X_i) \approx 0$$

Now, it would not be exactly 0 because we have ignored terms. So obviously, this equation is not exact. It is approximate, but it is good for iterations. Okay, So, many people confuse this idea saying that then the next step, I should get an answer. Obviously, I would not. all it is good for is for iterations. Now, notice, what has happened magically is this original equation, which was nonlinear in X is linear in Δx.

(Refer Slide Time: 13:06)


Linearized Eqn
Linear in Δx
iterations
⇒ Δx = − f(x_i) / f′(x_i)
Newton-Raphson iteration : x_{i+1} = x_i − f(x_i) / f′(x_i)

So, the original equation was nonlinear in the parameters we were solving for, the new modified equation, which I am going to call the linearized equation is linear in Δx, this is actually a fundamental insight in a lot of science and engineering, if you have a nonlinear curve, a nonlinear curve is sort of like repeated linear curves. So, if I really zoom in, it starts looking linear.

So, you can put together a bunch of linear segments and it will start looking nonlinear when you zoom out, okay as you should be able to see okay. So, similarly, what was nonlinear in X becomes linear, when you put together many small Δx. So, you take small steps and the overall result becomes nonlinear. so, this is simple we already know how to solve this equation and like I said this is trivial.

So, you can now say Δx is,

$$\Delta x = \frac{-f(X_i)}{f'(X_i)}$$

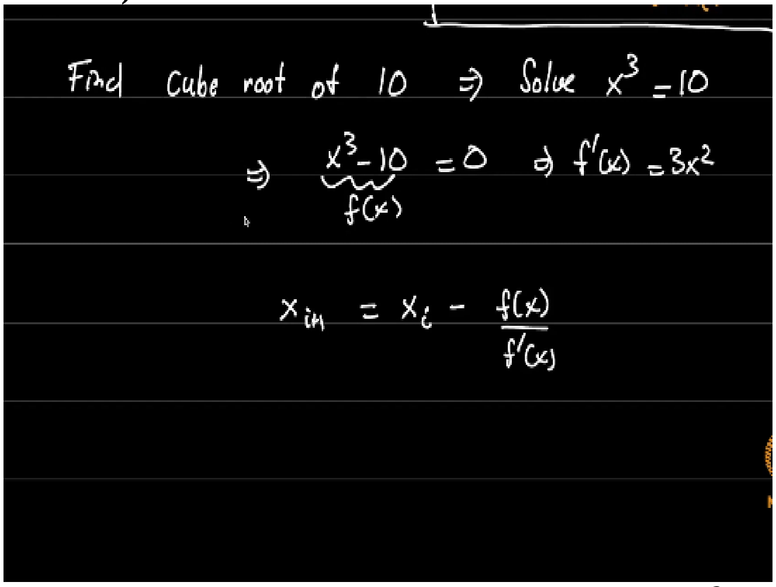
So, overall you can write the algorithm the Newton Raphson iteration is,

$$X_{i+1} = X_i - \frac{f(X_i)}{f'(X_i)}$$

It, as an easy way for you to remember which comes about which comes below is dimensional analysis, f has the dimensions of f, f prime has the dimensions of f divided by Δx. So, you need X, Δx and X new.

So, this is the Newton Raphson step, this is previous all of you would have seen it we even have a question of 2 within the assignment just to refresh your memory. but you will see that slowly, we will add complexity to it. So as long as you understand whatever I talked about so far, it should be easy for you to go further and actually go to the nonlinear regression case. So let us quickly see an example of this.

(Refer Slide Time: 15:16)



Find cube root of 10 $\Rightarrow$ Solve $x^3 = 10$

$$\Rightarrow \underbrace{x^3 - 10}_{f(x)} = 0 \Rightarrow f'(x) = 3x^2$$

$$x_{i+1} = x_i - \frac{f(x)}{f'(x)}$$

So let us say I want to find the cube root of 10, so this means solve $x^3 = 10$, which means solve $x^3 - 10 = 0$. So, this is our $f(x)$ so our iteration is going to be,

$$X_{i+1} = X_i - \frac{f(X_i)}{f'(X_i)}$$

and

$$f'(X) = 3x^2$$

So, we can repeat it from some initial guess again, like I said in the previous videos, in case you are doing this for credit, please do get used to doing these things by hand. Let us look at a quick code to calculate this from a few initial guesses just to see how Newton Raphson and moves, and then we will actually move to the nonlinear systems case.

(Video Starts: 16:18)

So let us go to the code now so here is a very simple code for Newton Raphson, you can see that I have defined the function $x^3 - 10$. I have also defined the gradient of the function or the derivative of the function, which is $3x^2$, and I have taken some random initial guess I am running it for 10 iterations, this is not the best way to do it. But we could just look at what happens with let us say 5 iterations.

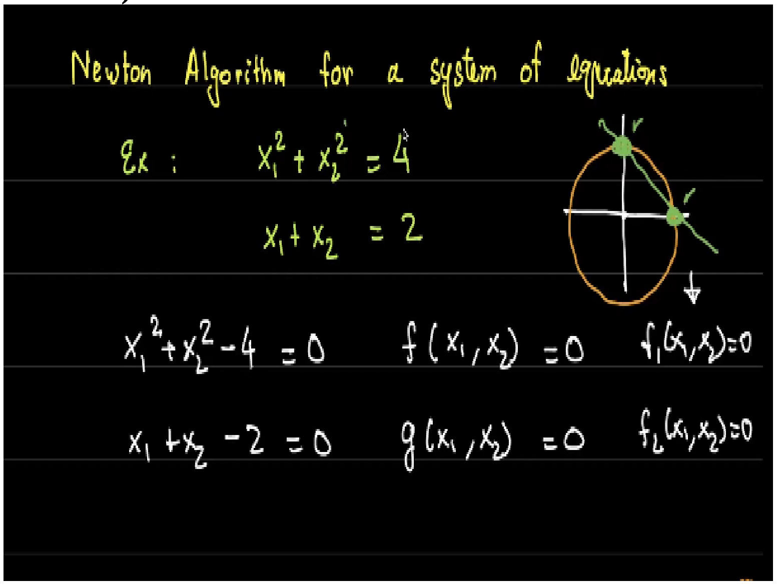
Quickly, we can also check what the actual cube root of so I think I made a mistake. So, what is the actual cube root of 10. So, you can see this number here, 2.1544 etcetera, etc, ending in 884 now let us actually try the Newton Raphson method. Okay, So I will put a breakpoint here and run it. So, the current value of x is 3. Now, you can now check x very rapidly has become 2.37 from 3, we could also start with a power of x it would still converge in this case. Okay. So now you see it has now updated it to 2.17.

Now, if I continue, and you can see it has actually converged to the final value in about 5 steps. Once again, you can actually check the cube root of 10. Non different, so it has actually converged to 16 degrees, 16 digits of accuracy, just within 5 steps, this is one of the magic of Newton Raphson, when it converges, especially when it is moving the root it is it has what is known as quadratic convergence, and it can converge really fast. Okay, So, for all practical purposes, it will converge to the right answer really rapidly.

So, there is some difference between here and here so we can actually run for a few more iterations and see how well Newton Raphson does. So now you can see from here, there is some improvement and I can also check the actual cube root. so now this is no different from before. So, this is the advantage of Newton Raphson. As we proceed, we will now see how to extend the same idea to a system of equations. So let us now do that.

(Video Ends: 18:56)

(Refer Slide Time: 19:01)



Newton Algorithm for a system of equations

Ex : $x_1^2 + x_2^2 = 4$

$x_1 + x_2 = 2$

$x_1^2 + x_2^2 - 4 = 0 \qquad f(x_1, x_2) = 0 \qquad f_1(x_1, x_2) = 0$

$x_1 + x_2 - 2 = 0 \qquad g(x_1, x_2) = 0 \qquad f_2(x_1, x_2) = 0$

So, we can now look at Newton Raphson for our newton algorithm for a system of equations. So, let us say we have 2 equations and 2 unknowns so let us take an example, an example makes things much easier. So, example, let us say we have this equation

$$x_1{}^2 + x_2{}^2 = 4$$

And we also want this as 2 variables so we also want let us say

$$x_1 + x_2 = 2$$

So all of you can quickly imagine what this would look like on the plane.

So, something like this, the first 1 is corresponds to a circle of radius 2, the second equation is a line. So, it would be something like this and you can see that used to intersect at 2 points so this has 2 solutions. So, 2 equations, 2 variables because of the quadratic, and because it is nonlinear, it actually has 2 solutions. So, Newton's method, as I will show you depends on where it converges depends on where you start.

So, this could have been also a problem with gradient descent, and might be one of the reasons why it converges to an unphysical answer, but more importantly, for our case, gradient descent has too many iterations. So, what we are trying to get rid of within the nonlinear regression case is to get rid of the too many iterations problem. So let us say we now reformulate this, again as 2 equations of the form,

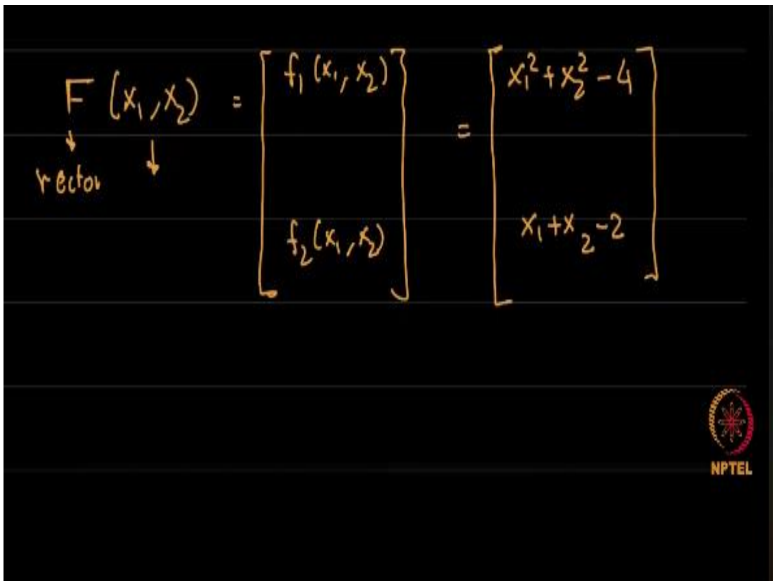
$$x_1{}^2 + x_2{}^2 - 4 = 0$$

and similarly,

$$x_1 + x_2 - 2 = 0$$

So now we have 2 equations so I will call this $f(x_1, x_2) = 0$, we can call this $g(x_1, x_2) = 0$. Another option is to call this $f_1(x_1, x_2) = 0$, and call this $f_2(x_1, x_2) = 0$. So let us make this choice Okay, f and g if required, but I will typically use f_1 and f_2.

(Refer Slide Time: 21:27)


$F(x_1, x_2) = \begin{bmatrix} f_1(x_1,x_2) \\ f_2(x_1,x_2) \end{bmatrix} = \begin{bmatrix} x_1^2+x_2^2-4 \\ x_1+x_2-2 \end{bmatrix}$
vector
NPTEL

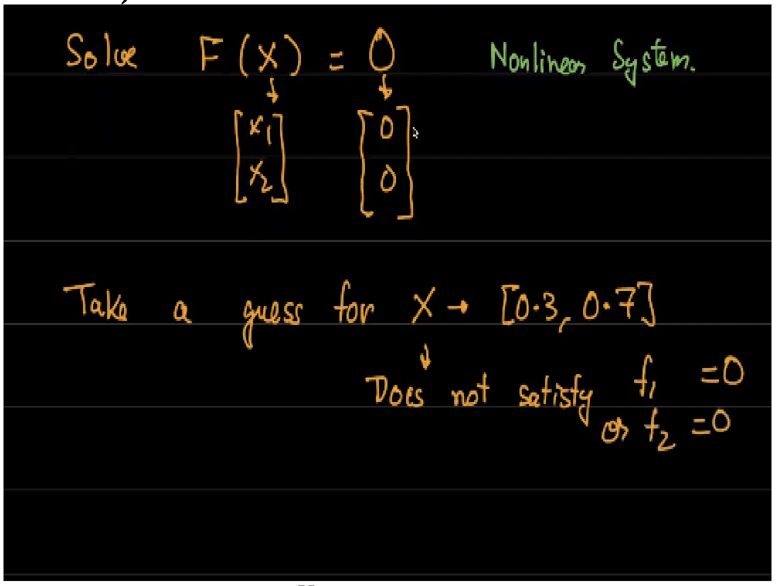
Now put together we can now see this as a case where I am saying F, capital $F(x_1, x_2) = 0$, where F is now a vector, which is,

$$F(x_1, x_2) = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix}$$

So, which is of course, in this case,

$$= \begin{bmatrix} {x_1}^2 + {x_2}^2 - 4 \\ x_1 + x_2 - 2 \end{bmatrix}$$

(Refer Slide Time: 22:01)

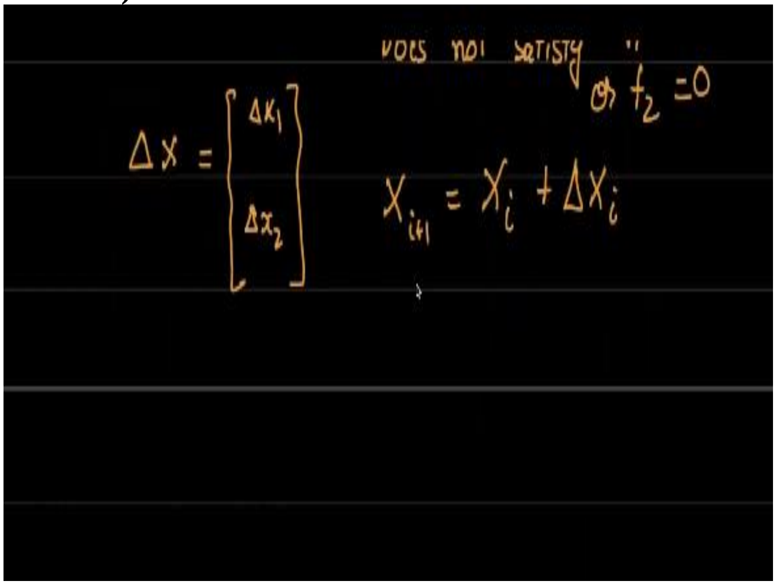

Solve $F(X) = 0$
Nonlinear System.
$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
Take a guess for $X \rightarrow [0.3, 0.7]$
Does not satisfy $f_1 = 0$
or $f_2 = 0$

We want to solve $F(X)$ is again, $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ = 0 and this 0 on in term is actually $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$. So this is now the nonlinear system. Now once again, we use the same idea as before, see, this is nonlinear in X. But maybe we can linearize so that it is somewhat linear in delta x or it is linearizable and delta x and we ignore the higher order terms. And we know how to solve a linear system of equations. So, we do the same thing that we did for 1 equation, we know how to solve a linear equation in 1 variable, We use that to solve a nonlinear equation in 1 variable.

Similarly, we know how to solve a linear equation in many variables, we use that to solve nonlinear equations and many variables is the same idea again, and again, we will do the same thing with regression, we know how to solve a regression problem, or linear regression problems. So, we will use that to solve a nonlinear regression problem, the same trick, we apply multiple times. Okay, so let us come to this. So let us say we take a guess for X.

So let us choose some guess something like 0.3, 0.7, we will know immediately that does not satisfy either f_1 = 0, or f_2 = 0, you can plug this in and check very quickly that this does not satisfy either of these cases. So now, what do we have to do?

(Refer Slide Time: 23:58)


Does not satisfy "
or $f_2 = 0$
$\Delta X = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}$
$X_{i+1} = X_i + \Delta X_i$

Once again, we will postulate that there is a ΔX, which corresponds to some,

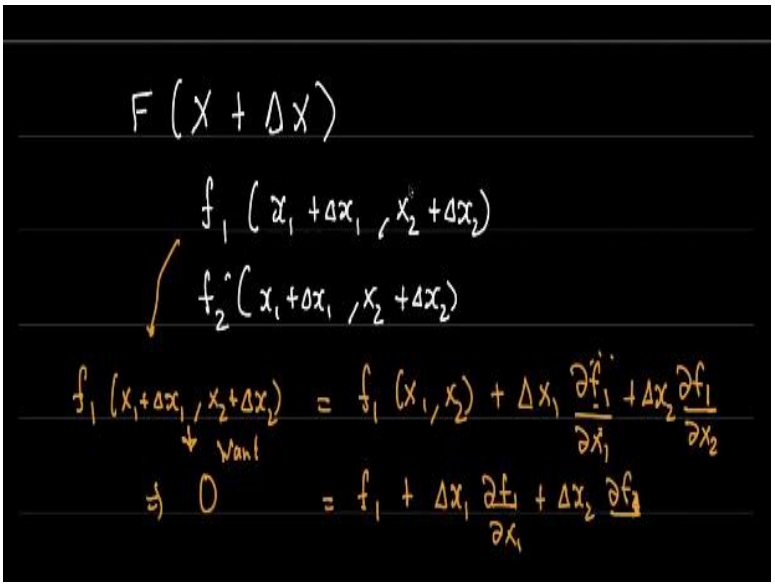
$$\Delta X = \begin{bmatrix} \Delta X_1 \\ \Delta X_2 \end{bmatrix}$$

and

$$X_{i+1} = X_i + \Delta X_i$$

Okay. All right. We have the same problem as before.

(Refer Slide Time: 24:23)



$$F(X+\Delta X)$$

$$f_1(x_1+\Delta x_1, x_2+\Delta x_2)$$

$$f_2(x_1+\Delta x_1, x_2+\Delta x_2)$$

$$\underset{\downarrow \text{ want}}{f_1(x_1+\Delta x_1, x_2+\Delta x_2)} = f_1(x_1, x_2) + \Delta x_1 \frac{\partial f_1}{\partial x_1} + \Delta x_2 \frac{\partial f_1}{\partial x_2}$$

$$\Rightarrow 0 = f_1 + \Delta x_1 \frac{\partial f_1}{\partial x_1} + \Delta x_2\, \partial f_1$$

Now we want $F(X+\Delta X)$ another way of saying it is, what is $f_1(x_1+\Delta x_1, x_2+\Delta x_2)$. And what is $f_2(x_1+\Delta x_1, x_2+\Delta x_2)$. it is for this precise reason that we had worked out the Taylor series for these 2. So, let us take this so f_1 at this new position,

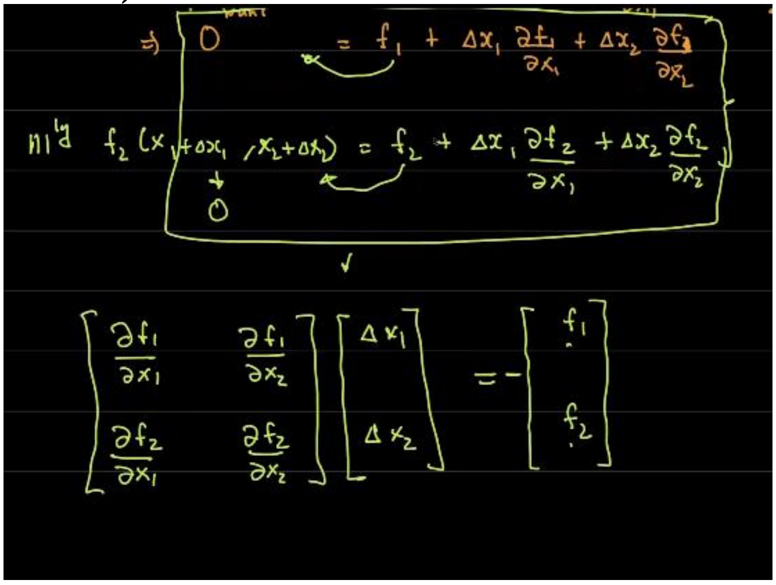
$$f_1(x_1+\Delta x_1, x_2+\Delta x_2) = f_1(x_1, x_2) + \Delta x_1 \frac{\partial f_1}{\partial x_1} + \Delta x_2 \frac{\partial f_1}{\partial x_2}$$

So, this tells us that this is going to be, we want this to be driven to 0, this is what we want. Okay. So this tells us f_1, I am going to leave it as f_1,

$$f_1(x_1+\Delta x_1, x_2+\Delta x_2) = f_1 + \Delta x_1 \frac{\partial f_1}{\partial x_1} + \Delta x_2 \frac{\partial f_1}{\partial x_2}$$

It is understood that is being calculated at (x_1, x_2).

(Refer Slide Time: 26:00)



$$\Rightarrow 0 = f_1 + \Delta x_1 \frac{\partial f_1}{\partial x_1} + \Delta x_2 \frac{\partial f_1}{\partial x_2}$$

lly $$\underset{\downarrow\ 0}{f_2(x_1+\Delta x_1, x_2+\Delta x_2)} = f_2 + \Delta x_1 \frac{\partial f_2}{\partial x_1} + \Delta x_2 \frac{\partial f_2}{\partial x_2}$$

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = -\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

Similarly, you can see quite easily the same trick applies for f_2,

$$f_2(x_1 + \Delta x_1, x_2 + \Delta x_2) = \; f_2 + \Delta x_1 \frac{\partial f_2}{\partial x_1} + \Delta x_2 \frac{\partial f_2}{\partial x_2}$$

Now notice this, what are the unknowns in these 2 equations? The unknowns in these 2 equations are these terms, this is what we were trying to find out just like delta x was the unknown in the 1 variable case, Δx_1 and Δx_2 are the unknowns in the 2-variable case, what is known?

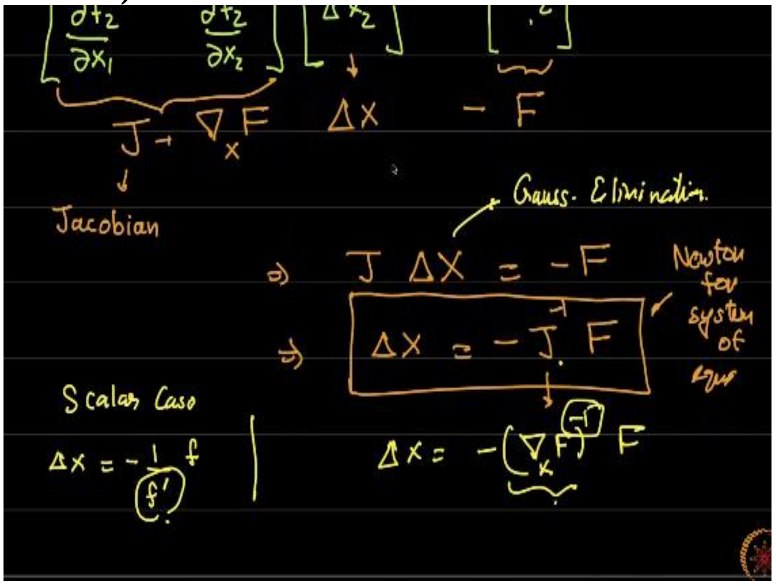
We already know what the current value of f_1 and f_2 of, we also can calculate the $\frac{\partial f_1}{\partial x_1}$, $\frac{\partial f_1}{\partial x_2}$, etcetera etc. So let us rewrite this so we can rewrite this as the,

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \end{bmatrix}$$

So let us see what this corresponds to I will write the equations and hopefully this will make things clearer. This is equal to $\begin{bmatrix} -f_1 \\ -f_2 \end{bmatrix}$. How did this come about?

I moved this to the left-hand side or flip this side of the equation, so you get a $\begin{bmatrix} -f_1 \\ -f_2 \end{bmatrix}$ here. In fact, we can now adjust this and call this minus of $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$. Now notice this multiplication matrix multiplication, here, it comes to the $\frac{\partial f_1}{\partial x_1}$ times Δx_1, $\frac{\partial f_1}{\partial x_2}$ times Δx_2, first equation, second equation, Δx_1 multiplied by $\frac{\partial f_2}{\partial x_1}$, Δx_2 multiplied by $\frac{\partial f_2}{\partial x_2}$.

(Refer Slide Time: 28:30)



$\frac{\partial f_2}{\partial x_1}$ $\frac{\partial f_2}{\partial x_2}$
$J \rightarrow \nabla_x F$ ΔX $-F$
Jacobian
Gauss. Elimination
$\Rightarrow J\,\Delta X = -F$
$\Rightarrow \Delta X = -J^{-1} F$
Newton for system of eqns
Scalar Case
$\Delta x = -\frac{1}{f'} f$
$\Delta x = -(\nabla_x F)^{-1} F$

Now notice the structure of this equation this basically is our Jacobian again; this is not the lost function. This is Jacobian this is Jacobian in another way of saying it is this is the gradient of capital F with respect to X. I am going to call this the Jacobian. And try to avoid confusion between the J of Jacobian J of the loss function. This, of course, is our delta X vector, and this is simply negative of our F vector.

So, let us see what the equation becomes this becomes,

$$J\Delta X = -F$$

which means,

$$\Delta X = -J^{-1}F$$

So, this is the newton for a system of equations formula. How so? Very simple once you have guessed for X, now you write a linear equation for ΔX, how much should I change X and that equation becomes this. Now, notice J actually is a gradient it is very much like gradient of F with X.

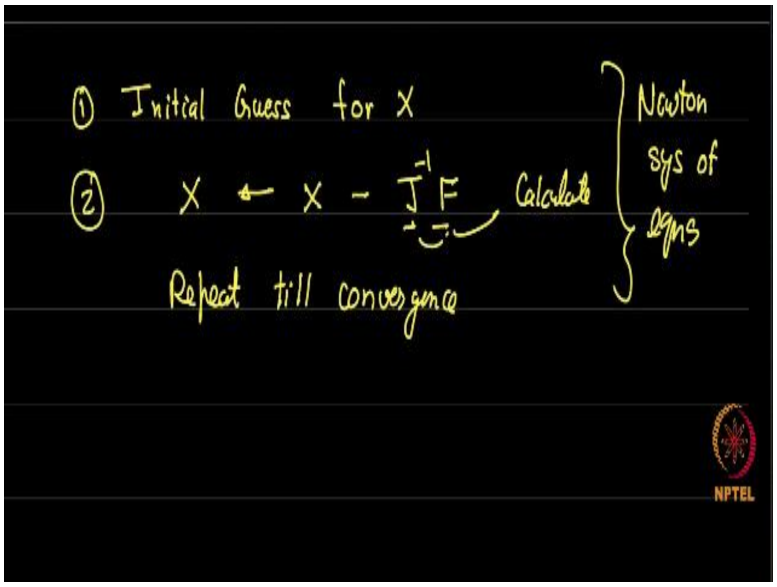
We never really invert. we use as I have said before even the normal equations typically you will use go gauss elimination, or some other method to solve this most exam problems in this course will be something like 2 by 2 systems or something of that sort. So those you can solve by hand, it is usually making the Jacobian which is a little bit harder or else for the assignments you can use programs if you wish to.

Now notice how symmetric this is to the scalar case, the scalar case was,

$$\Delta X = \frac{-1}{f'}f$$

Another way of saying it is $f'^{-1}f$ is the same thing, okay, it is exactly the same relationship as before, except now J is a matrix here it was a f prime was a scalar Of course, there is a matrix because it is now a system of equations, it is 2F's and 2X's.

(Refer Slide Time: 31:04)


① Initial Guess for X
② $X \leftarrow X - J^{-1}F$ Calculate
Repeat till convergence
Newton Sys of eqns

So, once again, to summarize, the idea is initializing the guess for X then update X as X - Jacobian inverse of F of course, these 2 have to be calculated. And you repeat till converge that is it, this is newton for a system of equations. And this is used in many practical places. Of course, It has trouble in case J is not invertible, you will have trouble just like in case F prime goes to 0, you will have trouble there are all sorts of convergence problems that occur for newton also.

Nonetheless, in practice, we will see next week, not this week that you can combine this newton approach with other approaches and make it better behave. So, let me just show you a quick code for solving the system that I just popped up, just so that you have completion. And then in the next video, we will move on to nonlinear regression. So let me show you the code right now.

(Video Starts: 32:20)

So, this is newton for system of equations, hopefully, you are able to see it within your screens here. Maybe I can make it bigger, just so that it is clearer. I have the same system of equations that I used in the theory, just before. So, F is made up of 2 parts, ${x_1}^2 + {x_2}^2 - 4$ and $x_1 + x_2 - 2$. And then we have the Jacobian which we can actually calculate del f1 del x1 is 2x1, del f1 del x2 is actually we should change this to 2 X2, I have written 2 X2, so del f1 del x2 is 2 X2 so that is correct.

Similarly, del f2 del x1 is 1 and del f2 del x2 is also 1. So, you have this Jacobian and you have the actual original function. So, I give a random initial guess, let us say I gave a guess that X = 0.1, 0.7. So let us go here and run. Now you can see that and take a quick step you can see an updated X. So but just to check what was the case, you can check what J of X is going to be in the next step F of X is going to be in the next step.

And you can also do something like inverse of J of X, multiplied by F of X and solve this, but that is not very appropriate, because for large systems, this is kind of inefficient as far as MATLAB is concerned. Okay, So, but you can now see that X has been updated, the new X now is from where it was, it has now become -1.41, and to 3.41. This might or might not be a good thing, let us see whether this initial guess works for what we are trying. So, I have given 10 iterations.

Now you can see that after a few iterations, or maybe after 6 or 7 iterations, it has now converged, it was already going to 0 and 2 very rapidly, if you remember the graph that I had shown, one of the points where we could get to was 0, 2 that was on the y axis. Now, it is probably possible that if I change the order of the guesses and make this 0.7 and 0.1, it will actually converge to the other value. So let us check that that is the case Yeah, that is indeed the case.

So, the other solution now attracts it, because this point is closer to is origin the x axis and this point is closer to the origin and the y axis, you actually converge to the other solutions. So, this is the power of a newton iteration for systems. So, just got a 2 by 2 system we actually rapidly converge to the answers. And now our question is going to be can we do the same thing for a system of equations especially when in inverse problems of the sort that we have been considering So far.

we have very, very few parameters in the nonlinear equations, when the parameters are large gradient descent is better, but for a low number of parameters, it turns out that something like what we did for a system of equations here, if we could do it for regression, that would work well for a low number of parameters also. So, we will both derive as well as see a solution with what is known as the gauss newton algorithm in the next video, so I will see you in the next video. Thank you.

(Video Ends: 36:08)