

Inverse Methods in Heat Transfer
Prof. Balaji Srinivasan
Department of Mechanical Engineering
Indian Institute of Technology, Madras

Lecture - 28
Tikhonov Regularization and Levenberg – Marquardt - Theory

(Refer Slide Time: 00:19)

Overfitting - Ill-posedness $\xrightarrow{\text{2017}}$ Regularization

Add terms to the objective fn, penalty that penalize high coeffs

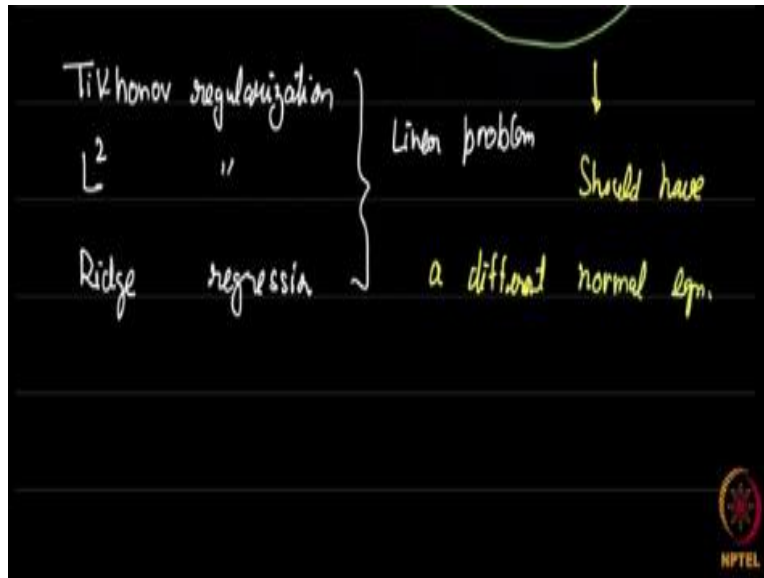
$$J = \frac{1}{2m} \sum_{i=1}^m (y_i - q_i)^2 + \frac{1}{2} \lambda \sum_{i=1}^n w_i^2 \rightarrow \text{Penalty}$$

$\|w\|_2$

In the last video we saw that linear methods suffer from overfitting; in case you use two complex a model and this overfitting actually is a sign of ill posedness of the problem. And the solution to ill posedness is actually something called regularization. And the way we added or the way we accomplished this regularization was by adding terms to the objective function. These terms penalize high coefficients or high parameters.

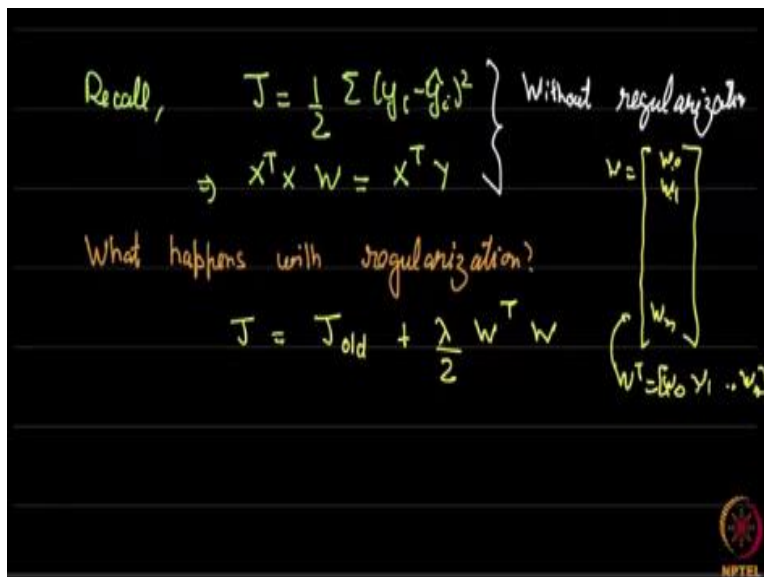
So, for example we saw that our usual loss function now becomes J is a composite loss function which is, $\frac{1}{2m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$. This is the original plus this additional penalty term plus the penalty term which is, $\frac{1}{2} \lambda \sum_{i=1}^m w_i^2$ with the summation which we also call $\|w\|$ this is the penalty term; this is also known as the regularization term.

(Refer Slide Time: 02:18)



As I showed, as I told you this term is variously called Tikhonov regularization, also called L^2 regularization also, called Ridge regression when applied to linear problems. Now Tikhonov regularization is a little bit, general than this but I am not going to get into that immediately. But let me show you so now because the loss function is different, this should have been different normal equation.

(Refer Slide Time: 03:09)



So, recall that when we started with the pure loss function J is,

$$J = \frac{1}{2} \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

So, this led to,

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

This is without regularization. Now within one of the earlier exercises I had asked you, but this question was kind of sitting there earlier what happens to the case with regularization with the extra term. So, now our new J is,

$$J = J_{old} + \frac{\lambda}{2} W^T W$$

Why? because if w is,

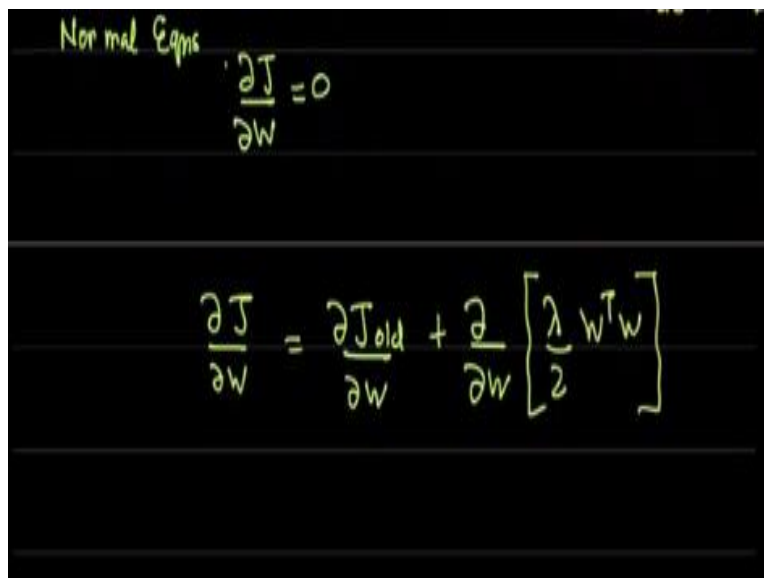
$$W = [w_0 \quad w_1 \quad \cdot \quad \cdot \quad w_n]$$

and W^T is,

$$W^T = \begin{bmatrix} w_0 \\ w_1 \\ \cdot \\ \cdot \\ w_n \end{bmatrix}$$

and as we saw in the earlier video W^T , we will simply give you sigma of or $w_0^2 + w_1^2 + \dots + w_n^2$.

(Refer Slide Time: 04:50)



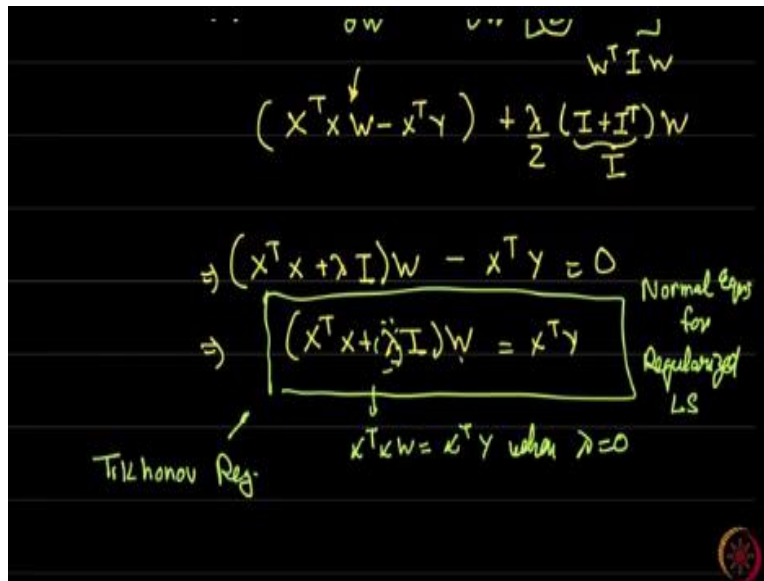
Normal Eqns

$$\frac{\partial J}{\partial W} = 0$$
$$\frac{\partial J}{\partial W} = \frac{\partial J_{old}}{\partial W} + \frac{\partial}{\partial W} \left[\frac{\lambda}{2} W^T W \right]$$

Now remember how we calculated the normal equations; the normal equations were obtained by setting $\frac{\partial J}{\partial W} = 0$. Now the new $\frac{\partial J}{\partial W}$ is going to be,

$$\frac{\partial J}{\partial W} = \frac{\partial J_{old}}{\partial W} + \frac{\partial}{\partial W} \left[\frac{\lambda}{2} W^T W \right]$$

(Refer Slide Time: 05:18)



Now this you might recollect from your old video. otherwise, I would recommend that you look back at it this gives us $X^T X W - X^T Y$, that is what this term was. This extra thing is the same as you look at this matrix $(W^T I W)$ and we had differentiated $(W^T A W)$ before. So, this is a scalar and it is a constant we can take that out $\frac{\lambda}{2}$ if you go back to our previous videos this will be $(I + I^T) W$, $(I + I^T)$ of course since it is an identity matrix it is simply I .

So, this gives us $\frac{\partial J}{\partial W}$ is,

$$(X^T X + \lambda I) W - X^T Y = 0$$

So, this tells us that,

$$(X^T X + \lambda I) W = X^T Y$$

This is the normalized normal equations for the regularized Least square equations. So, you can use this to generate the results that I showed you in the previous case all you need to do is put W going from 1 to 9 in case you have a ninth-order polynomial and simply put change lambda values.

You see this is not very different this becomes $X^T X W = X^T Y$, when $\lambda = 0$. So, the case when $\lambda = 0$ recovers the original normal equations also. So, this we can call again the same name that I gave

you I am going to call this Tikhonov regularized normal equations. But this is for a linear system what happens for a non-linear system. Let us look at that next.

(Refer Slide Time: 07:45)

$$J = \frac{1}{2} \sum_{\text{data}} (y_i - \hat{f}_i)^2 + \frac{1}{2} \lambda \sum_{\text{features}} W_i^2$$

Gauss-Newton $\rightarrow \Delta W$
 $X^T X W = X^T Y \rightarrow Z^T Z \Delta W = Z^T D$
 $D = (Y - \hat{f})$

Tikhonov
 $(X^T X + \lambda I) W = X^T Y \rightarrow (Z^T Z + \lambda I) \Delta W = Z^T D$
 λI is the Tikhonov term

So, when we want to find out how the same idea can be used to regularize, let us say non-linear equations, we apply the same trick. We say that J is once again,

$$J = \frac{1}{2} \sum (y_i - \hat{y}_i)^2 + \frac{1}{2} \lambda \sum W_i^2$$

Let us call it, because this summation is over data. And this summation is over features or the number of degrees of polynomials. Now we need the equivalent of Gauss Newton. now remember Gauss Newton worked on delta W it worked on ΔW .

And whenever we had when the linear equation was $X^T X W = X^T Y$, the corresponding Gauss Newton was $Z^T Z \Delta W = Z^T \Delta Y$, which basically was D, we can call this ΔY basically $(Y - \hat{Y})$. Now you can rederive the whole thing in this case with this additional term but I am just going to appeal to intuition just in order to save time.

So, similarly for Tikhonov when you have $(X^T X + \lambda I) W = X^T Y$. You can derive this in exactly the same way that we derived Gauss Newton I will save your time by not doing that. You can simply write some of you can see this intuitively,

$$(Z^T Z + \lambda I) \Delta W = Z^T D$$

So, the only change really speaking this is the additional Tikhonov term.

(Refer Slide Time: 10:19)

Handwritten slide content:

$(X^T X + \lambda I) w = X^T y \rightarrow (Z^T Z + \lambda I) \Delta w = Z^T D$

λI is circled and labeled "Tikhonov term".

Stabilizes Gauss-Newton - Tikhonov damps oscillations
Slower but stable, converges

Levenberg - Marquardt

NPTL logo in the bottom right corner.

And what this does again I will not be able to show it easily in practice, this is easy to see in the linear case when we ordered more and more terms. But this basically stabilizes I will show you an example. But it will not be a very clear example I will show you an example in the next video for the non-linear risk. But this stabilizes Gauss Newton. so typically, if you try Gauss Newton for some problem let us say and it diverges it is a good idea to add the Tikhonov term.

Because it will damp it as you can see as you could have seen it seen in the linear case also. So, Tikhonov will damp. so, this is sort of a damping term, damps oscillations or poor convergence. So, this has typically slower but stabler convergence. So, sometimes when Gauss Newton diverges Tikhonov will actually slowly converge it will converge it will be slower. In many cases when Gauss Newton does converge as I will show you in the next video.

In many cases when Gauss Newton converges taken off convert this a little bit slower but it is sort of the typical slow but steady. So, in case you get bad answers it is a good idea to put Tikhonov regularization there and that will give you slightly better answers.

(Refer Slide Time: 11:56)

But that is not all, there is a slightly better version of this called Levenberg Marquardt. Now the way Levenberg Marquardt was originally derived at least in at least one direction. It was not through Tikhonov but through an entirely different argument and I will show you that argument also now. But first let us see Levenberg Marquardt as if it is just a variation of Tikhonov. So, let us look at this equation, so let me take an example.

Let us take X is a data set and it is 6 cross 3, for example you could have 1, 1, 1, 1, 1, 1 and there are two features X_1, X_2 . So, for example a quadratic model would be a decent example of this W is let us say w_0, w_1, w_2 . So, this is 3 cross 1 now Y has to have the same number of data points but it is only one output, so Y is y_1, y_2, \dots, y_6 , so this is a 6 cross 1. In case this is the case, then if I look at Tikhonov regularization $X^T X$ or let us look at Gauss Newton itself $(Z^T Z + \lambda I) \Delta W = Z^T D$.

Now Z is going to have the same size as X, as you might remember $\frac{\partial y}{\partial w}$. So, Z is going to be 6 cross 3, Z transpose is going to be 3 cross 6, so this matrix is going to be 3 cross 3 matrix. Lambda I simply mean lambda, lambda, lambda so you are going to have a 3 cross 3 matrix here. And the diagonal terms alone will be modified and each of them will go to this plus lambda the same two terms same term this plus lambda then same 2 terms this plus lambda. So, that is what it goes to because of the addition of this lambda.

(Refer Slide Time: 14:17)

Levenberg-Marquardt

$$\underbrace{Z^T Z}_P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} + \lambda \begin{bmatrix} p_{11} & 0 & 0 \\ 0 & p_{22} & 0 \\ 0 & 0 & p_{33} \end{bmatrix}$$

L. M algorithm = $\begin{bmatrix} p_{11} + \lambda p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} + \lambda p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} + \lambda p_{33} \end{bmatrix}$

Now the question you can ask in Levenberg Marquardt is why not have these three lambdas as different lambda 1, lambda 2, lambda 3. So, the way Levenberg Marquardt works is I will show you a different justification shortly, but let us look at the first justifications. Levenberg Marquardt is, I do not want this modification term to just have constant terms, I am going to have slightly different terms now, instead of this will be trouble.

Because now you have to play with 3 hyper parameters, I have to decide on lambda 1, I have to decide on lambda 2, I have to decide on lambda 3 over time what they found out was. Let us say if Z transpose Z is let us call this something, I will call this P, so let us just say this is,

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

and you want to add the modification term the lambda term. So, a good idea seems to be multiplied by a lambda but multiply only on the diagonal terms that is still true.

So, these terms are still 0 you are not adding anything here. But instead of adding 1 here which was taken off you add p_{11}, p_{22}, p_{33} . So, there are numerical reasons for this so basically if it was this before it goes to,

$$\begin{bmatrix} p_{11} + \lambda p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} + \lambda p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} + \lambda p_{33} \end{bmatrix}$$

This is the Levenberg Marquardt algorithm. So, the idea is simple you see again that in effect what you have done is you have added 3 different terms earlier you were only adding lambda.

Now lambda is scaled according to what the original term was that is the only idea it can be seen basically as a modification of the Tikhonov algorithm.

(Refer Slide Time: 16:47)

Levenberg-Marquardt - Less stable than Tikhonov but faster

$$[Z^T Z + \lambda \text{diag}(Z^T Z)] \Delta W = Z^T D$$

Std algorithm
non-linear regression

Tikhonov

$$[Z^T Z + \lambda I] \Delta W = Z^T D$$

So, you can write a Levenberg Marquardt as,

$$[Z^T Z + \lambda \text{diag}(Z^T Z)] \Delta W = Z^T D$$

$Z^T Z + \lambda \text{diag}(Z^T Z)$. So, only the diagonal terms get added that is this term. This is basically diagonal of Z transpose Z multiplying delta W = Z transpose times delta Y which we called D. So, compare this with Tikhonov which is,

$$[Z^T Z + \lambda I] \Delta W = Z^T D$$

So, Levenberg Marquardt is less stable than Tikhonov but faster. so, this is the standard algorithm for non-linear inverse problems. So, for non-linear regression this is the standard algorithm and it works pretty well.

(Refer Slide Time: 18:24)

Alternate Interpretation of Tikhonov algorithm

$$(Z^T Z + \lambda I) \Delta W = Z^T (Y - \hat{Y})$$

When λ is very high.

$$\lambda I \Delta W = Z^T (Y - \hat{Y})$$

$$\Rightarrow \Delta W = \frac{1}{\lambda} Z^T (Y - \hat{Y})$$

Now I want to give you alternate interpretation as I said there are multiple interpretations of both these algorithms. So, the alternate interpretation of the LM algorithm it is like this, so suppose I again looked at this Z transpose $Z + \lambda$ times diagonal or actually let me give an alternate interpretation of the Tikhonov version itself, Tikhonov algorithm because it is a little bit easier to see there.

So,

$$[Z^T Z + \lambda I] \Delta W = Z^T (Y - \hat{Y})$$

λ times I times $\Delta W = Z^T D$ and instead of calling it $Z^T D$, I am going to call it $Z^T (Y - \hat{Y})$, where you have. Now what happens when λ is very high, so imagine λ is very high then this becomes,

$$\lambda I \Delta W = Z^T (Y - \hat{Y})$$

which means ΔW ,

$$\Delta W = \frac{1}{\lambda} Z^T (Y - \hat{Y})$$

(Refer Slide Time: 20:05)

$$\Rightarrow \Delta W = \frac{1}{\lambda} Z^T (y - \hat{y})$$

$$= -\beta Z^T \frac{\partial J}{\partial \hat{y}}$$

$$\frac{\partial \hat{y}}{\partial w}$$

$$\Rightarrow \Delta W = -\beta \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w}$$

$$\Delta W = -\beta \frac{\partial J}{\partial w}$$

$$\Delta W = -\alpha \frac{\partial J}{\partial w}$$

$$\alpha = \frac{1}{\lambda}$$

→ This is Gradient Descent!

Now let us call this something let us call $1/\lambda$ as β the Z transpose times this, we know is negative of $\frac{\partial J}{\partial \hat{y}}$, this we know is $\frac{\partial \hat{y}}{\partial w}$. So, basically you get ΔW as,

$$\Delta W = -\beta \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w}$$

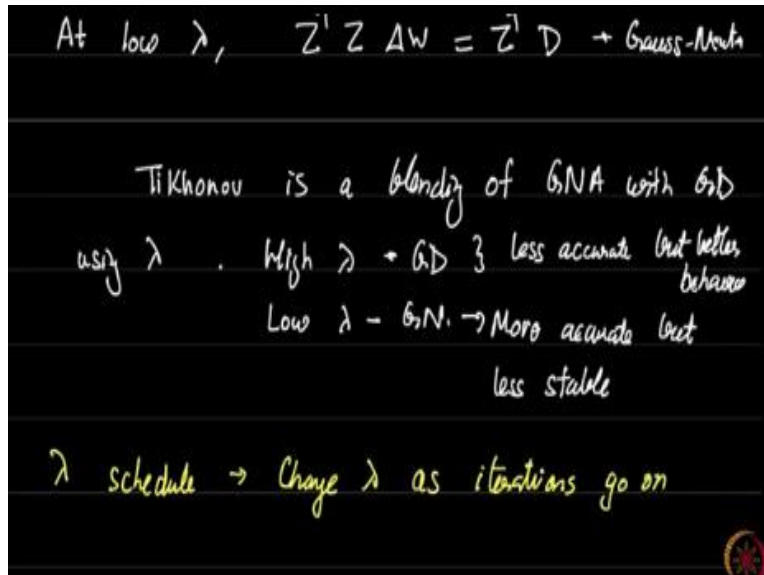
I can show it in a matrix term but I am kind of running through this a little bit faster than you might be comfortable with. Because I am just trying to show a particular point rather than do some mathematical derivation.

So, this basically says so, all these matrices you have to be a little bit careful about transposes etcetera I am not being careful this is not a formal derivation. But I want you to see this $\frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w}$ is the same as say $\frac{\partial J}{\partial w}$. So, this is gradient descent, so notice remember that delta w for gradient descent was minus alpha times $\frac{\partial J}{\partial w}$. So, all we have is gradient descent with $\alpha = 1/\lambda$.

Now what does this mean when λ is very high, α is very low. So, it is exactly corresponding to that so if I keep the regularization parameter or what looks like the regularization parameter as very high. What Tikhonov regularization does is? It does just do gradient descent many people call this version of Tikhonov also a Sullivan but not what it is just not quite accurate. But that is I mean different names are sitting in the literature.

So, this at high alpha or high lambda Levenberg Marquardt or Tikhonov is equal to gradient descent.

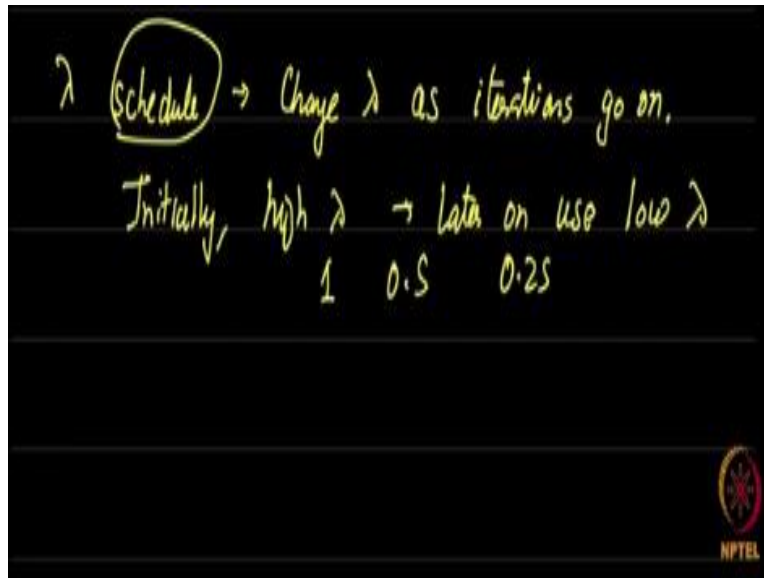
(Refer Slide Time: 22:27)



Now what happens at lower lambda? At low lambda this becomes $Z^T Z \Delta W = Z^T D$, which is Gauss Newton. So, another interpretation of Tikhonov is a blending of Gauss Newton with gradient descent using lambda. So, high lambda gives me gradient descent, low lambda gives me Gauss Newton algorithm and this is a nice way of seeing a Tikhonov. So, this puts us right in between.

Now what happens when you use gradient descent is, it is less accurate but better behaved this is more accurate, but less stable. Now what you want to do ultimately something else so we usually have something called a lambda schedule, that is change lambda as iterations go on.

(Refer Slide Time: 24:06)



So, initially we used high lambda this is called a schedule and later on use low lambda. We are not going to do any such thing typically what will happen is you will start with let us say $\lambda = 1$ then after a few iterations you make it 0.5, 0.25 so on and so forth and as it comes closer and closer to the actual answer you get closer and closer to Gauss Newton which will converge very fast.

Initially you want to explore a lot of the parameter space and as you come later on you come to better and better accuracy as you come to the parameters. So, what we saw within this video were these two variants of regularization, Tikhonov as well as Levenberg Marquardt. In the next video we will look at a simple coding example with the same case that we did for Gauss Newton unsteady convection plus heat generation case which was a non-linear case.

And we will see how you can write the program for this you will see that the variations are actually quite small and I have a few comments about performance of these algorithms also in the next video. So, see you in the next video. Thank you.