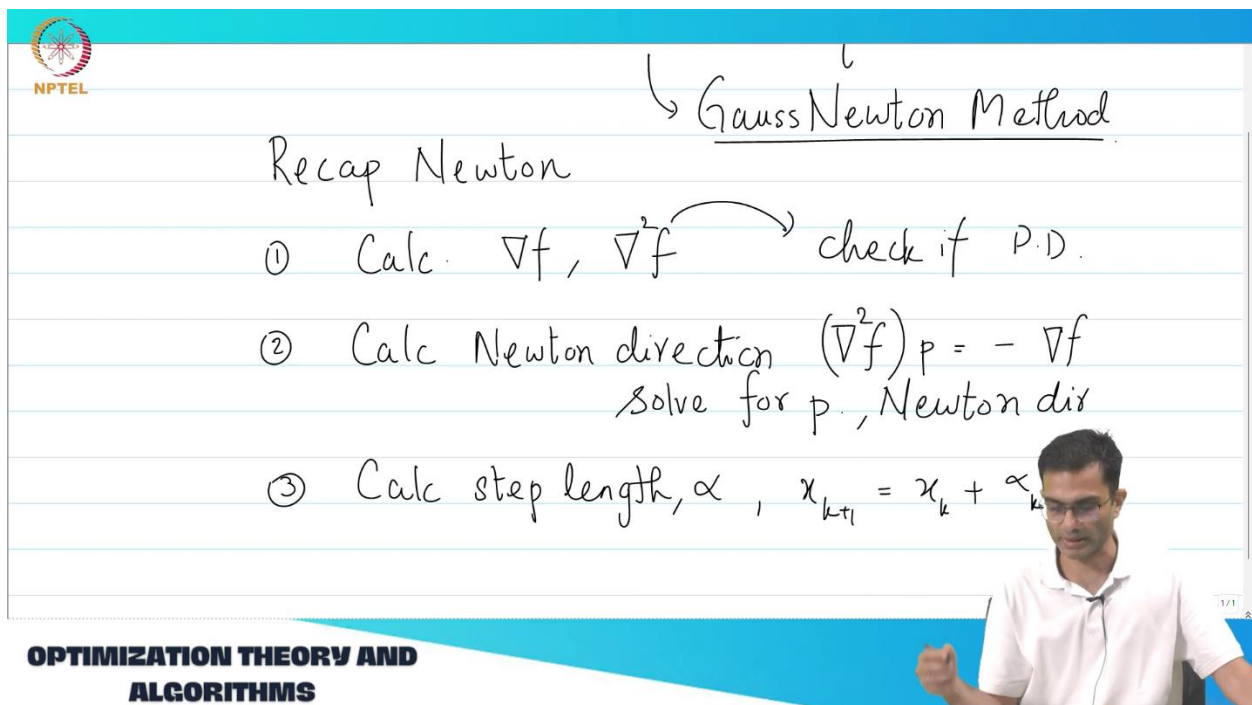


Non linear least squares

Okay, so let us get started. I would have liked to have covered this before the quiz so that we started on a new note today, but there is one small topic that is still left over. We were talking about the least squares problem, and as I mentioned, least squares problems are among the most commonly encountered problems in engineering. We looked at the linear least squares problem. Today, we will complete the loop and look at non-linear least squares problems. That is the first half of today's class. So, in particular, the method that we are going to talk about today is a very traditionally well-known, well-used, and very popular method. It is called the Gauss-Newton method.

Many of you may have heard of this, and we are already familiar with the Newton method, right? The Newton method we will summarize now, plus some kind of small set of modifications to deal with the fact that the objective function is written in least squares form. That is what the Gauss-Newton method is. So, let us quickly recap. In fact, this was a question you did yesterday, right? You want to recap the Newton method.



NPTEL

Recap Newton

① Calc. ∇f , $\nabla^2 f$ → check if P.D.

② Calc Newton direction $(\nabla^2 f)p = -\nabla f$
Solve for p, Newton dir

③ Calc step length, α , $x_{k+1} = x_k + \alpha_k$

Gauss Newton Method

OPTIMIZATION THEORY AND ALGORITHMS

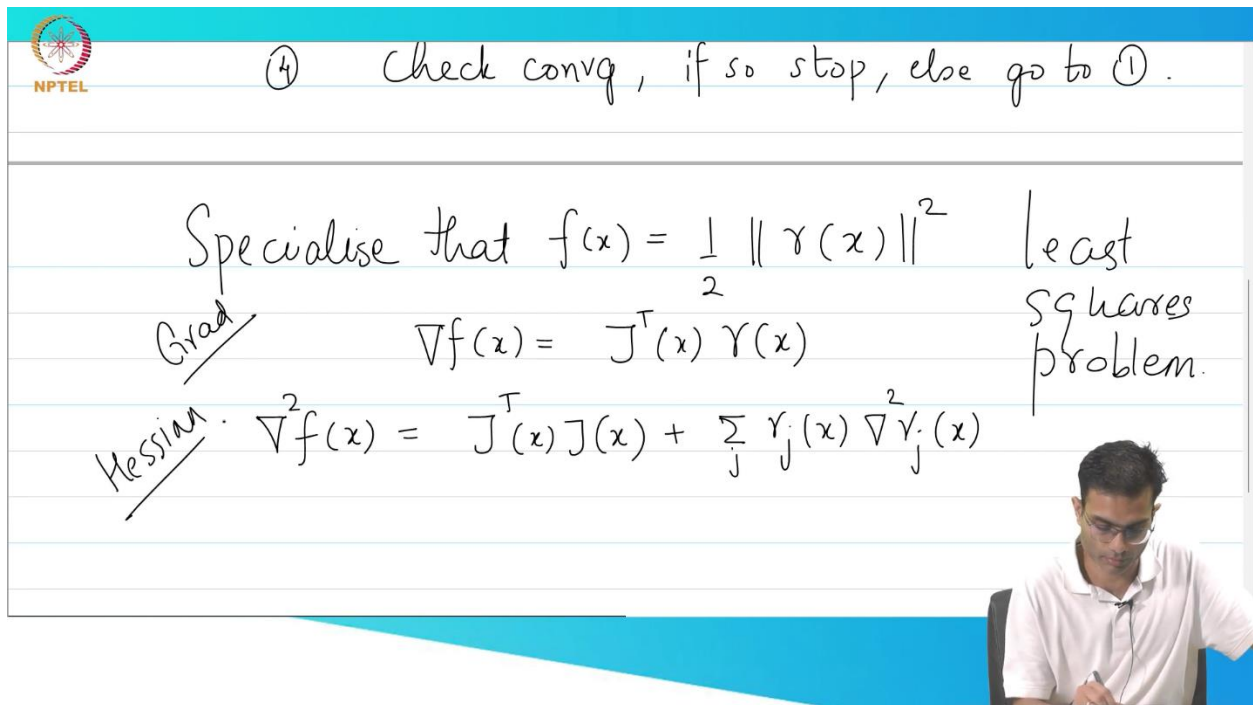
What is the first thing that you want to do after picking the initial point? Even before that, I mean you need to first calculate the gradient and Hessian, right? So, let us just say that you need to first calculate, because it is not going to be given to you, right? So, calculate ∇f and the Hessian, okay? And after having calculated, you need to check if this is positive definite. Let us assume

that for simplicity, it is positive definite. Okay. Next, after having gotten these two quantities, what do I compute next? The Newton direction, right?

So, calculate the Newton direction, which is simply the solution to this:

$$\mathbf{H}\mathbf{p} = -\nabla f$$

I need to solve for \mathbf{p} in order to get the Newton direction, right? So, solve for \mathbf{p} , okay, what would be next? Calculate the step length. There are many strategies for this.



④ Check convg, if so stop, else go to ①.

Specialise that $f(x) = \frac{1}{2} \|\mathbf{r}(x)\|^2$ least squares problem.

Grad $\nabla f(x) = \mathbf{J}^T(x) \mathbf{r}(x)$

Hessian $\nabla^2 f(x) = \mathbf{J}^T(x) \mathbf{J}(x) + \sum_j r_j(x) \nabla^2 \mathbf{r}_j(x)$

Wolfe conditions will help us with an exact line search. That is the most practical thing because exact line search is almost always impractical. So, let us say we did it. After we got our step length, what do we do? We do the update, right? So, we will do

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

I move to the next step, and I am almost done. What do I need to check now? Check for convergence, right? So, check for convergence. I am leaving it general; this convergence could be that you have exhausted your maximum number of iterations or the norm of ∇f is very low, whatever, right? And so, if so, stop, right? Else, I go back to which step? Step 1, right? This was the general recap of the Newton method.

Here, we did not make any assumptions on the form of f , f can be any function, this is how we have studied the Newton method. Now, let us specialize it to the extra information that is being given to you, that f is composed of a sum of squares, right? That is what the least squares problem is. So now, let us make the specialization. So specialize that f is... We had written it in this form, right? A sum of residual squares. So, this is in the least squares problem, and the advantage of writing it specially in this way is that you get some special form of the gradient Hessian, etc.

Everyone remembers this sum of squares. Each of those components of $r(\mathbf{x})$ is, for example, what the difference between a measurement and a model is, the prediction of the model and the measurement. That is what I have in $r(\mathbf{x})$. So, what was ∇f ? We had a nice form for ∇f , remember, in terms of the Jacobian of r . Right, so I had:

When can I ignore:

- ① If $r_j(\mathbf{x})$ is affine
- ② When we are close to the soln.

$$\nabla^2 f(\mathbf{x}) \approx \mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x})$$

$$\mathbf{z}^T \nabla^2 f \mathbf{z} = \|\mathbf{J}(\mathbf{x}) \mathbf{z}\|^2 > 0$$

$$\nabla f = \mathbf{J}^T \mathbf{x} \cdot r(\mathbf{x})$$

This was my gradient. Similarly, I had a nice expression for the Hessian, right? So, I had:

$$\mathbf{J}^T \mathbf{x} \mathbf{J} \mathbf{x}$$

That was the first term, and the second term was a summation; there was no nice closed form expression. So, I just wrote it like this, right? So, this is my gradient, this is my Hessian, right?

And the place where the Gauss-Newton method actually begins, this is just a recap of least squares problems, was an observation we had made previously. This term over here, we had said that there are some conditions under which this term can be low in magnitude, right? So, does anyone remember what those conditions were? Okay. One is if r_j is affine; affine is not the exactly correct word, but affine. If r_j is affine, then by the time I take the Hessian of r , I am going to get 0, right? So, when can I ignore the red term? Right. And the second was when I am actually close to the solution, when I reach close to the solution, the residual is going to be small, right? So, close to the solution, okay. And then when in either of these two situations, it is reasonable to approximate the Hessian like this, okay? Just by looking at this, what is the advantage of having the Hessian in this form as opposed to the general form? Is there something that pops out? It looks to be... It seems that very easily it could be positive definite. Why? Because if I do:

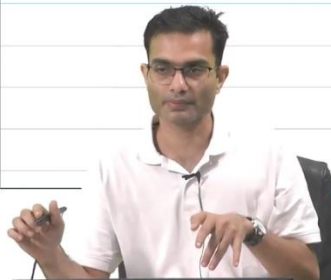
$$\mathbf{z}^T \mathbf{H} \mathbf{z}$$

$\Rightarrow J$ must be full col. rank.

$$\hookrightarrow \text{Calc } p_N \rightarrow (\nabla^2 f)_p = -\nabla f$$

$$J_k^T J_k p_k = -J_k^T r_k \rightarrow \text{all are fns of } x.$$

Solve for p_k , calc α_k .



OPTIMIZATION THEORY AND ALGORITHMS

Okay. So, we said that possibly one of the advantages of this is that this is easy for me to check whether it is positive definite or not, and the easy way to look at it is this expression. What is this become? $\|Jx \cdot z\|^2$. So, this is going to be... If I want it to be positive definite, that means I want this to be greater than 0, and this will happen under what condition? J has to be full column rank, J cannot be any matrix, J has to be full column rank, then this expression will always be greater than 0 for non-zero z . Remember, for positive definite, I have to consider all possible z , but they have to be non-zero, right? So, it implies that J must be full column rank, and this is an easy enough condition to check, okay? So, alright, so that is one of the advantages of this approximation.

So remember, before I go further, in our original Newton method, I had to check at each step if my Hessian was positive definite, and that can be, in general, if I give you some function $\nabla^2 f$, it is not an easy thing to check because if you did not know anything, how would you check whether a matrix is positive definite? You would have to, for example, calculate the eigenvalues. That is an expensive operation. On the other hand, because I am specializing my problem as a least squares problem with A, b , what else? If I am in a position to ignore the second term of the Hessian, then all I have to do is check full column rank, which is not as expensive an operation as calculating eigenvalues, right? So, this is a substantial advantage that we get. Okay. Now, what we will do is we will just fit our work out. What is this Gauss-Newton method now?

So, in the Newton method, step one we have done: calculate ∇f , $\nabla^2 f$, we have done. Next is calculate the Newton direction. So, let us see. Does this calculation become anything easier? Right. So, calculating the p_{Newton} . Right, so what was that?

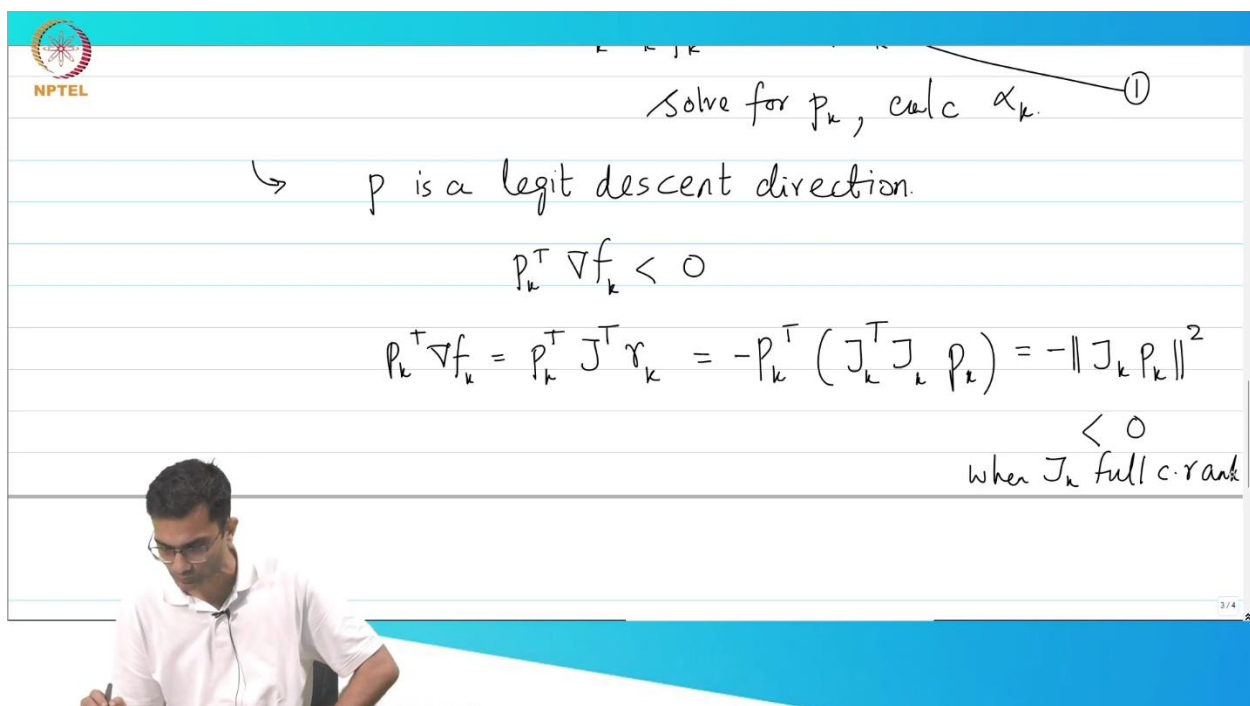
$$\nabla^2 f \cdot p = -\nabla f$$

So, what is $\nabla^2 f$ under the approximation? $J^T J$, p remains here, and ∇f was $J^T r$. Okay. I mean these are not all constants. So, I mean these are all functions of x , okay? They look like constants

over here, okay. Actually, we should be putting the subscript for the iteration on each of these, okay?

So, we need to solve for p_k . Okay. I get my, once I get this, I calculate the step length, calculating the step length moves me to the next point. And convergence is, of course, $\|\nabla f\| \rightarrow 0$, that is how the usual Newton method works, okay? So, in a nutshell, this is what the Gauss-Newton method does. It ignores the second part of the Hessian, and you are calculating the search direction each time by solving this system of equations. Okay, and you are doing an inexact line search to get α . So, it is very similar to the Newton method except a few simplifications are coming because I just called it the least squares problem.

So, one thing, what was the, if you can recall, what was the reason that I wanted to check for positive definiteness of the Hessian? No, no. Full column rank tells me that it is positive definite. Why do I want to check whether it is positive definite? So that the descent direction, so that the direction p_k is actually a descent direction, right? So, if you, let us just look at this once again, right? So, p is legit, and this means that $p_k^T \nabla f_k$, this expression should be what? Positive, negative? Be less than 0. Then it is a descent direction because I am going in the opposite side as ∇f .



The whiteboard contains the following handwritten text:

- NPTEL logo
- Solve for p_k , calc α_k . ①
- $\rightarrow p$ is a legit descent direction.
- $p_k^T \nabla f_k < 0$
- $$p_k^T \nabla f_k = p_k^T J^T r_k = -p_k^T (J_k^T J_k p_k) = -\|J_k p_k\|^2$$
- $$< 0$$

when J_k full c-rank

The man in the bottom left is wearing a white shirt and glasses, looking down at a book or paper on his desk.

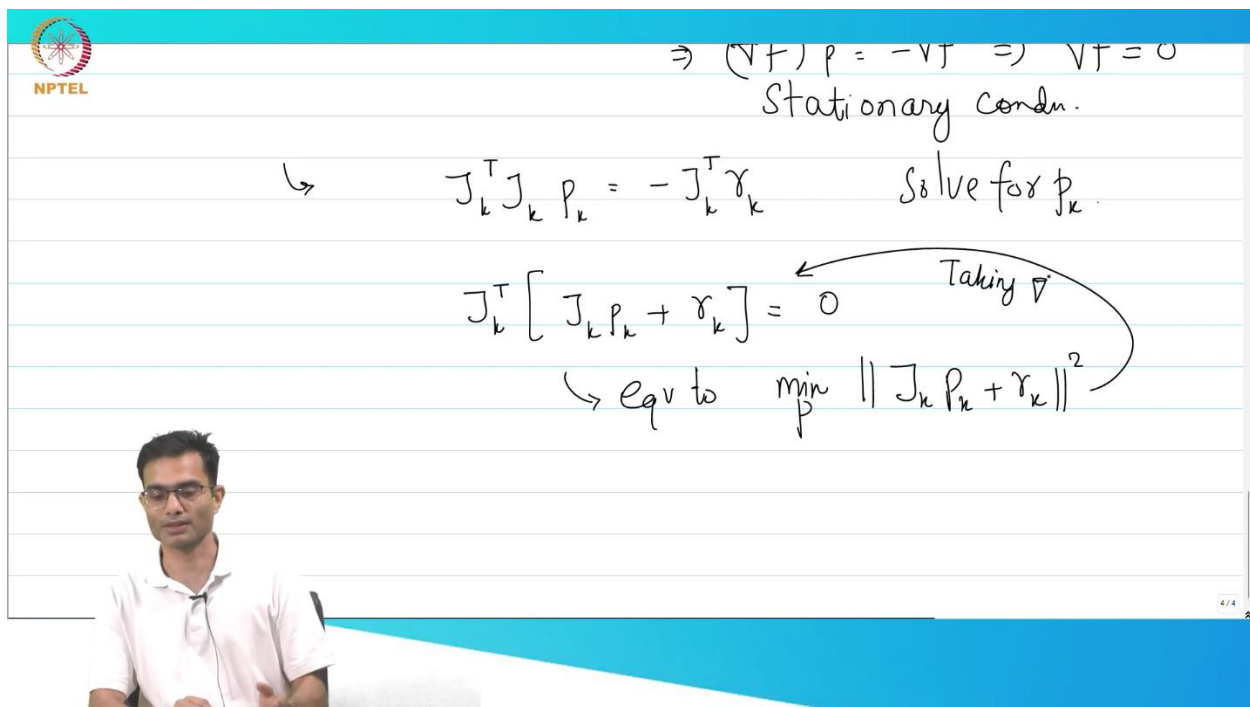
So, now, if I just substitute what I know of ∇f . So, this is $p_k^T \nabla f$ was what? $J^T r$. So, this is $J^T r_k$. Okay. I have this equation over here which has an expression for $J^T r_k$ exactly inside it. So, I can substitute this in here.

Is everyone with me on this? So, this would become $p_k^T J^T p_k$. And there is a minus sign. The minus sign should have come... No, there is no minus sign here. Where is the minus sign? Where does the minus sign come from? This can be very conveniently written as the norm squared of something.

So, this is $\|J_k \cdot p_k\|^2$, right? And you can see this is definitely something that is less than 0, okay? Now, when is it less than 0? When J_k is full column rank, okay? When can this, when will this go to 0? When can this expression actually go to 0? One step before that: J_k is full column rank, so obviously, it cannot be 0. So, what is the option? p_k has to be 0. So, the only condition is p_k going to 0 tells me what? You have reached a stationary point, but how do I relate p_k to... Can I relate p_k to anything else, ∇f ? Right. So, what is ∇f ? How do I relate p_k to ∇f ? I can use this expression, right? Right. So, if I substitute $p = 0$ into this top expression over here...

So, this is going to get me. So, the Newton equation, the Newton condition, right? This is equal to $-\nabla f$. So, this will imply that $\nabla f = 0$, and $\nabla f = 0$ is what? The condition for stationary points, right? So, this is the stationary point. So, this is... I mean, we knew that it is going to be a legitimate descent direction, we see that it is a legitimate descent direction because J is full column rank. So, when it actually goes to 0 is when my iterations have come to an end because it can only go to 0 when my $\nabla f = 0$.

The final note that I want to make is now, let me do one final connection between this non-linear least squares problem and the least squares linear least squares problem.



$\Rightarrow (\nabla f)p = -\nabla T \Rightarrow \nabla T = 0$
 Stationary condn.
 $\hookrightarrow J_k^T J_k p_k = -J_k^T r_k$ Solve for p_k .
 $J_k^T [J_k p_k + r_k] = 0$ Taking ∇
 $\hookrightarrow \text{eqv to } \min_p \|J_k p_k + r_k\|^2$

So, what is the equation that we are solving at each step? This is the equation that I have to solve at each iteration in order to get p_k . Right. So, supposing you wrote a piece of MATLAB code that solves a linear least squares problem and you want to reuse that piece of code for the non-linear least squares problem, this is an example of how you would do it. So, let us just look at this equation. So,

$$J_k^T J_k p_k = -J_k^T r_k.$$

This is what we have to do: solving for p_k . We can do a little bit of algebra over here. Can I take some term common from both? J_k^T can be pulled out, right? So, this is the same as saying that

$$\mathbf{J}_k^T \mathbf{J}_k p_k + r_k = 0.$$

Now, you all are by now experts at taking gradients of various expressions.

Does this expression look like the gradient of some well-known expression? This looks like $(A^T A x + b)$, and that is the gradient of what expression? $\|Ax + b\|^2$, right? So, this is the same as, let us say, this is equivalent to:

$$\|Ax + b\|^2.$$

We arrive at this by taking the gradient, and I get this expression. This expression over here doesn't it look like your, if you just scroll back in your notes, this looks exactly like your linear least squares problem, right? In the linear least squares problem, your A was \mathbf{J}_k , and all of these terms are exactly like that, right? So, I mean this is, I don't even have to look at that. This is essentially something like $(Ax + b)^2$, which was what my linear least squares problem was anyway, right? So, what is the conclusion over here? In your non-linear least squares problem, at each iteration, you are solving one step of a linear least squares problem.

So, if you had written that piece of code as a function, you could just call that at each step. What is the output of solving the linear least squares problem? What do you get out of it? You get the direction p_k . It gives you p_k , right? The solution to this is p , you get your p . Once you get your p , if you have by now, you would have already written some pseudocode for inexact line search that will give you your α , and you are in business because now you can go to x_{k+1} .

Okay, so if you have, you know, your research problem, which is non-linear least squares, and if the Hessian approximation is looking like, you know, you can make it if it is a reasonable approximation to make, you can very, very easily get started with this. Okay. Any questions on this? Quite straightforward, right? It is almost, there is almost nothing new over here, it is just based on a slight modification of Newton's method, or rather taking Newton's method and pouring it into the least squares formulation.

Overall, okay, so he did not get how this is non-linear least squares. The point is, okay, let us look at here. So, let us look at the Hessian. If this were a linear least squares problem, your Hessian here would be constant; it would not be \mathbf{J} as a function of \mathbf{x} , it would be constant. Here the trouble is that at each step, your Jacobian is going to, in the linear least squares problem, \mathbf{J} became A , a fixed matrix, but now it is \mathbf{J} as a function of \mathbf{x} —that is the difference. That is why you have to do it iteratively.

So, all your tricks of, what should I say, truncated SVD, this, that, and the other can be applied now to each of the iterations. Yeah, well, whether you replace b with $-b$ does not matter, the form is the same: it is $Ax + b$ or $Ax - Y$.