


**Course Name: Optimization Theory and Algorithms**  
**Professor Name: Dr. Uday K. Khankhoje**  
**Department Name: Electrical Engineering**  
**Institute Name: Indian Institute of Technology Madras**  
**Week - 07**  
**Lecture - 53**

### Quasi newton methods

Okay good, let us start with the feedback chips from yesterday. I think everyone had some of the other comment on our Hessian modification scheme. Looks like it generated a lot of interest. Again one student, I do not know the name and saying that, so remember we said we could take our matrix  $A$  and add  $\tau$  times the identity matrix to it. And we said that that definitely pushes all the Gershgorin disks to the right half plane and therefore we are saved. This student says that, We had a follow up suggestion I think from you that why not modify only those discs which were going into the left half plane instead of modifying all the discs.

And the student feels that that cannot be done, it would be possible only if  $A$  was in diagonal form. But that is not the case. The Gershgorin and disc theorem said that you can find the center for any matrix, it need not be a diagonal matrix. Any dense matrix the center is given by what? The diagonal element that is the center and the radius is coming from absolute value of the sum of the other elements.



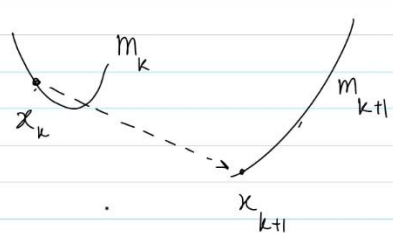
## Quasi Newton Method.


NM  $\rightarrow$  quadratic

QN  $\rightarrow$  superlinear

SD/CG  $\rightarrow$  linear

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \begin{cases} 0 & \text{S.L} \\ \mu & 0 < \mu < 1 \text{ . Lin} \end{cases}$$





So as long as I move all the disks to the right half plane, I do not have to move all of the disks, only those guys which are have some spillover on the left half plane, it should work. So whoever this is should give a little bit more of a reasoning. Is everyone convinced by that argument? That we need not move all the disks, we should just move those disks which are, have the, give the possibility of a negative eigenvalue. To save the modified Hessian from poor condition number

we make  $\delta$  large, but if we make  $\delta$  large therefore the value of  $\tau$  increases. So how can we solve both problems with the same? Obviously, we cannot, this is classic engineering, it is a trade off.

If you try to make the condition number better, you are adding more, you are modifying the matrix more. So, the deviation from the original problem is more, but you got a better condition number. So, you cannot win in both situations. I think I clarified this by the end of the class, but the question is still here, that Gershgorin simply mentions that the eigenvalue lies somewhere in the disk, but where and what exactly it is we do not know, right. So, this, supposing this was one of the, this is the origin.

So, does this automatically imply that we have a negative eigenvalue? No, right? Because the eigenvalue could be sitting here. So, that is why Gershgorin is a very coarse tool. It is not, it is giving us a range. So, an eigenvalue could be here, in which case we are in trouble. But to be on the safe side, what you could do? If you shift this over here to something like this, then there is no chance of a negative eigenvalue, right? That is what we are doing.

NPTEL

Q(N) - quadratic

Q(N)  $\rightarrow$  Superlinear

SD/CG  $\rightarrow$  linear

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \begin{cases} 0 & \text{S.L} \\ \mu & 0 < \mu < 1 \text{ Lin} \end{cases}$$

Diagram showing a curve with points  $x_k$  and  $x_{k+1}$ . A dashed line segment connects them, labeled  $\alpha_k p_k$ . The curve is labeled  $m_k$  and  $m_{k+1}$ .

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p$$

$$\nabla m_k(p) = \nabla f_k + B_k p$$

quadratic model of  $f$  around  $x_k$   
assume  $B_k$  symmetric

But as the student is observing that, we do not know exactly where this negative eigenvalue will be. So, it is a bit of an overkill and the cost is very low. So, that is the tradeoff. If the Hessian is being modified, this is a good question. Is not it similar to the quasi-Newton method where the Hessian is being approximated? In a sense the philosophy is similar, right? That we do not. It is a little bit, the Newton method is of course more expensive because first you calculate the Hessian, then you find out whether or not it is positive definite.

So you already paid the price. In quasi-Newton you do not even attempt to do that. In quasi-Newton you never compute the Hessian, you calculate an approximation of the Hessian. So in that sense they are not similar. How do we guarantee quadratic rate of convergence with the Hessian modification? And related question, I think this is Omkar.

It was mentioned that  $\delta_i = \tau \cdot i$  gives the minimum norm, but as the other students suggested only negative eigenvalues changed, will that give a smaller norm modification? Yes, it would. So, how do we guarantee quadratic rate of convergence with this Hessian modification? So, this is a chapter 3 of our textbook discusses this quite in detail. So, in fact there is, Theorem 3.8 of Nocedal Wright? It gives you a, it is a theorem, again it is a tricky theorem because he himself does not give the proof. He refers you to yet another reference, right? But what he says is that if the Hessian modification.

No. Yeah, okay. Should be back now. Is it okay? Okay. Yeah, so this theorem tells us that if the modification that I made to the Hessian each time has a bounded condition number, I still have quadratic rate of convergence. Okay, so that is what it is.

But remember all of this discussion becomes kind of unimportant for one very simple reason and this is really the bottom line that you should take away from this class. As I approach the solution. As I come close to the solution, in fact even this quadratic rate of convergence of the Newton method, it is in a region close to the solution where the Hessian becomes positive definite. So, we do not have to worry about this. It is when I start far away from the solution is where I need to do this Hessian modification. But as I get closer, it is guaranteed to be positive definite.


So, all of these tricks, they do not spoil the real convergence rate. And as I mentioned, I spoke only about Hessian modification. There are other tricks in the book next to this theorem about eigenvalue modification and so on. Lots of tricks are there which we will not discuss in detail.

So, all right. So, as promised now we will start talking about our more economical alternative to the Newton method which is the quasi-Newton method. As I mentioned the claim to fame of this Quasi-Newton method is that it sits somewhere between the Newton method which was quadratic and then I have the steepest descent or conjugate gradient method this was linear and Quasi-Newton sits in between it is super linear ok. So, if I look at  $\|x_{k+1} - x^*\|$ , this is the norm that I am looking at. So, as  $k \rightarrow \infty$ . What was the difference between super linear and linear? So, this is I can write it like this.

So, if it is 0, what do I call it? Super linear and if it is  $\mu$  where  $0 < \mu < 1$  this is linear. So, slight difference between these two methods, ok. So, again the Newton method is built on like most things that we have studied in optimization, it is built on our Taylor's theorem, ok. If I give you, let us start with a very simple high school discussion. If I give you two points, a scalar function  $f(x)$  and I give you  $x_1, x_2$ , what is the best model that you could construct? Could you construct a quadratic? No.

I can construct, best I could do is straight line. In more sophisticated language, what is a straight line called? an affine function, right? Affine functions means a constant plus a linear term, right? So, strictly when you say a function is linear, it means it has no intercept, it goes through the origin, that is what you, that is what is called strictly speaking a linear function, ok. So, with two points you could do affine, three points you could do quadratic, I mean parabola or quadratic function, right? So, in the quasi-Newton method, the idea again is that I am going to construct models of my function at each point, ok. Because there is the word Newton in quasi-Newton, that model is going to be what order? Second order, that is why the, that is why quasi-Newton, ok. So, it is a little bit like this, let us say I am here.

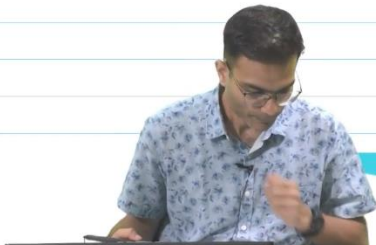
This is my  $x_k$  and let us say this is my  $x_{k+1}$  right. So, I am going to construct maybe like this I should call it ok, something like this ok or it may even be linear whatever I mean this some kind of a function being constructed over here. So this model is gonna be called  $m_k$ , okay. And the model I construct at a later point is gonna be called obviously  $m_{k+1}$ , okay. And quasi-Newton is also a line search method.



$$m_k(0) = f_k, \quad \nabla m_k(0) = \nabla f_k$$

We want  $\nabla m_k(p) = 0 \Rightarrow p_k = B_k^{-1} \nabla f_k$   
to minimize the model  $m_k$

QN motivation  $\rightarrow$  DO NOT compute  $B_{k+1}$  each time,  
Instead, update from  $B_k$ .



That means the way I go from  $x_k$  to  $x_{k+1}$  is, right. That's how I go. And what will this distance be? What is the distance between  $x_k$  and  $x_{k+1}$ ?  $\alpha_k$  times  $p_k$ , right. That is our general recipe of a line search method, all right. Now, knowing our Taylor's theorem, if I am standing at  $x_k$ , ok, I want to construct a quadratic model.

How would I construct this quadratic model? So for example, at  $p = 0$ , this model should agree with the function. So  $m_k(0)$ , what should it be? Not  $x_k$ , I am trying to model the function. So what should it be?  $f(x_k)$ ,  $f(x_k)$  has a shorthand  $f_k$ , okay. So, this is our very zeroth order term. Now, I will construct a linear term and a quadratic term, ok.

So, linear term, any point, anyone wants to guess what my linear term should be?  $\nabla f_k \cdot p$ , ok. And now comes the second term which is also straightforward and easy to guess, what would be? There has to be a half, right?  $p^T$ . Hessian, should I put Hessian? If I put Hessian, if I commit to writing Hessian there, I am in trouble because then it becomes Newton. So, what should I write instead? Some  $B_k$ , which is in this case, since I am doing quasi-Newton, it will not be Hessian and  $p$ , right? So, this is strictly speaking just a quadratic model.

The word model is very important because  $B_k$  I have not specified right now. Quadratic model of  $f$  around  $x_k$ , this is what it is. I mean I take inspiration from Taylor's theorem. Taylor's theorem only difference is  $B_k$  would be the Hessian. In Taylor's theorem, if the larger you take  $p$ , the more terms you need to keep.

So, that will always happen there. Remember our step length  $\alpha$  is there to save us in that regard. So, in that sense  $p$  is a direction. So, before we go to  $x_{k+1}$ , let us do some very simple algebra over here. What will be the gradient of  $m_k$ ? So, if I do this, So, remember in this expression I have written for  $m_k$ , what is the variable? What is changing?  $p$ . Is  $p$  a vector or a scalar? It is a vector, ok.

NPTEL

Value of  $\nabla f$  at  $x_k$ ?

$x_k \xrightarrow{\alpha_k p_k} x_{k+1}$

$m_{k+1}(p)$

$p=0 \rightarrow x_{k+1}$

$p=? \rightarrow x_k$       $p = -\alpha_k p_k$

$\nabla m_{k+1}(p) = \nabla f_{k+1} + B_{k+1} p$

$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - B_{k+1} \alpha_k p_k = \nabla f_k$      Critical design choice

from defn of  $\nabla m_{k+1}$

**OPTIMIZATION THEORY AND ALGORITHMS**

So, when I write  $\nabla m_k$ , obviously what are my  $\frac{d}{dp_1}, \frac{d}{dp_2}, \frac{d}{dp_3}$ ?  $p$  is the variable. So, what is the gradient of  $m_k$ ? Let us go term by term. First term, 0, constant. Second term,  $\nabla f_k$ , ok. And what else? That half will get cancelled when I do product rule, I will be left with a  $B_k p$ , ok.

We are, we made one small assumption here, what is that? Right, we assume that  $B_k$  is going to be symmetric, ok. So, let us put that in over here. Assume  $B_k$  is symmetric. So, let us look at, so  $m_k(0)$  is clearly  $f_k$ ,  $\nabla m_k(0)$  is what?  $\nabla f_k$ . So, you can see that  $m_k$  is agreeing with  $f_k$  or rather with  $f$  on two counts, it is agreeing with the function values, agreeing with the gradient value, ok. In fact, in addition to  $B_k$  being symmetric in this model since I am using a quadratic model, I should insert one more condition on it.

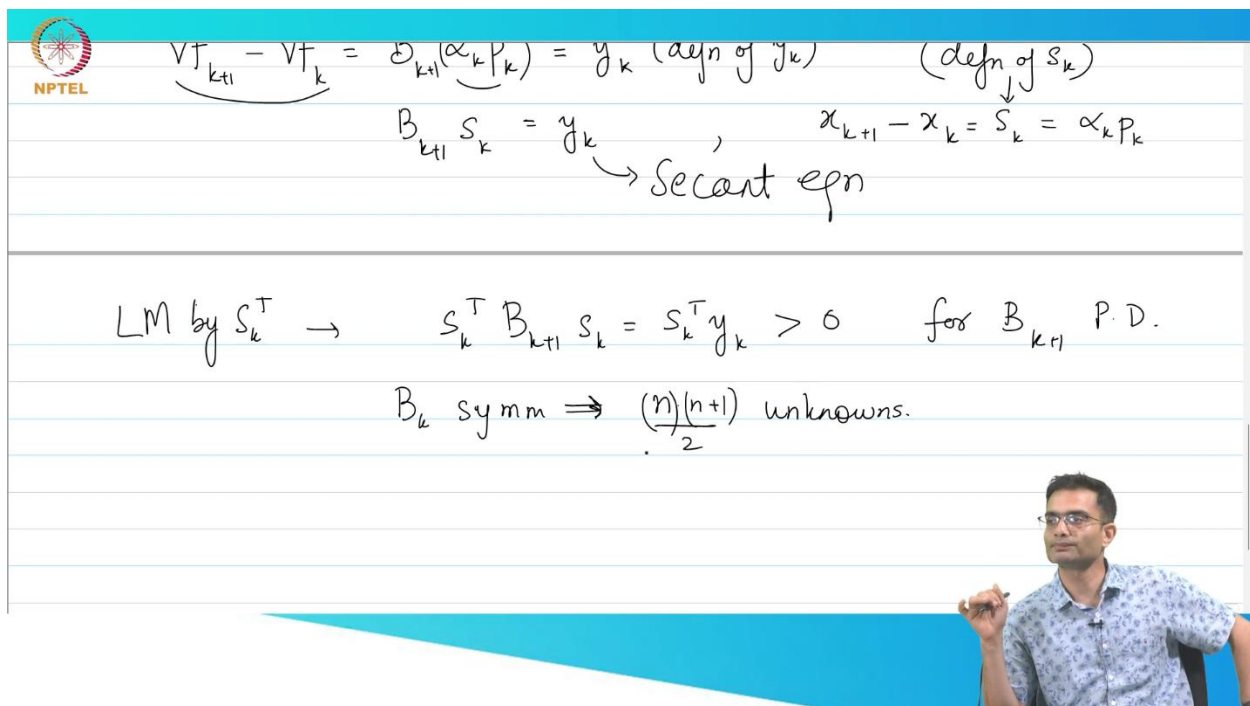
What should that be from our entire past discussion? Positive definite, right? Assume  $B_k$  is symmetric and positive definite, ok. Now what is the ideal  $p$  that I want? If I am at, so I am sitting here at  $x_k$ , I have constructed a model and I want to know now what is the step  $p$  that I should take. So again without going into math, common like if you put it into English, we should choose that value of  $p$  such that my model  $m_k$  is minimized. When is my model  $m_k$  minimized?  $\nabla m_k = 0$ , right. So, this tells us we want  $\nabla m_k(p) = 0$ .

This gives us the value for  $p_k$ , right. So, implies that  $p_k$  will therefore be equal to  $B_k^{-1}$ . I am at iteration  $k$ , I am at the point  $x_k$ . Because I am, so it is like this, I have reached a certain point in my iterations. Now I want to go further to a place where the function is minimized further, right.

So I am going to say let me restrict myself to a small neighborhood around  $x_k$  and based on what data I have about this function I am going to construct a model. Why am I constructing a quadratic model? For the precise reason what you know what I just wrote over here.

If I write a quadratic model I can tell you the minimizer analytically. Well, at a point in the sense this is the model is not at a point, the model is in the neighborhood. The model is constructed in the neighborhood of a point. So, I am minimizing the model, I am not minimizing a point, right. So, I am this let me just write that down, to minimize the model.

and the model  $m_k$  is valid or legitimate in the neighborhood of  $x_k$ . That is why the subscript  $k$  goes with  $m$ . As I move to a new point, obviously I should construct a new model, right. This is in very intuitive words, this is similar to the idea that you, if you have a very complicated function, you could keep linearizing it at each point, moving a small distance and linearizing it, right. So here we are going one step further and saying, hey why linear, let us do quadratic, ok.



NPTEL

$$\underbrace{Vf_{k+1} - Vf_k}_{\text{defn of } y_k} = \nabla_{k+1}(\alpha_k p_k) = y_k \quad (\text{defn of } s_k)$$

$$B_{k+1} s_k = y_k, \quad x_{k+1} - x_k = s_k = \alpha_k p_k$$

→ Secant eqn

---

LM by  $s_k^T \rightarrow s_k^T B_{k+1} s_k = s_k^T y_k > 0$  for  $B_{k+1}$  P.D.

$B_k$  symm  $\Rightarrow \frac{(n)(n+1)}{2}$  unknowns.

Alright, so we have got this. Now what was the whole argument of a quasi-Newton method that I do not want to compute Hessian, Hessian is expensive. So, if Hessian is expensive to calculate therefore I have replaced it by this  $B_k$ . So, what I would like is that this  $B_{k+1}$  should somehow come from  $B_k$ . So if I have something like that, an update rule which helps me, like I have an update rule for  $x_{k+1}$ . It is something I take  $x_k$  and make some modification, get  $x_{k+1}$ .

In conjugate gradient,  $p_{k+1}$  came from  $r_k$  and  $p_k$ , right? But you notice in the Newton method, every time I calculate the Hessian fresh. right. So, people had this intuition that why not apply this kind of similar logic to the Hessian also. So, this approximate Hessian I want some update rule in order that I use all the effort I have done in the previous step to get  $B_{k+1}$ . That is the sort of underlined intuition behind the quasi-Newton model, ok.



So, let us just note that the quasi-Newton motivation. So, the only thing that we have to worry about is well possibly  $B_0$  you need to spend some effort in computing and then I keep updating. So, if now let us just draw your attention to this. This is one piece of data,  $m_k(0) = f_k$ . Similarly, if I ask you what is  $m_{k+1}(0)$ , what would it be? So, I am considering  $x_{k+1}$ .

So, this is simply going to be  $f_{k+1}$ . Similarly,  $\nabla m_{k+1}(0)$  would be  $\nabla f_{k+1}$ . So, now what has happened is, I have Both of these models, they, I mean none of these models invoke my  $B$ , the guy that I am interested in, right? So I need to do some kind of a trick to connect these two red bubbles over here. This is the first bubble, right? And the second bubble. What is connecting these two guys is going to be some kind of a way to update  $B$ .

That is the missing link between these two. Right now they look like independent models, right? So, with this kind of intuition in mind, so let us note that down, this is the motivation. So, now let us consider  $m_{k+1}$ . Let us make or force this  $m_{k+1}$  model to satisfy two things.  $m_{k+1}$ , does it give me the correct gradient at  $x_{k+1}$ ? Let us ask question 1. Does it give me the correct value of  $\nabla f$  at  $x_{k+1}$ ? So, very, is that a yes or no? Yes, we saw it by construction right, we saw this over here.

It gives me the correct value of the gradient. So, let us put a tick mark over here. Now, let me ask you a second question. Does it give me the correct value of  $\nabla f$  at  $x_k$ ? Yes, no or we do not know or not necessary. It is definitely not, I mean it is not obvious. Looking at what I have written about  $m_{k+1}$ ,  $m_{k+1}$  if you look at the definition of  $m_k$ , if you look at for example over here, just replace  $k$  by  $k + 1$ .

There is no information about the previous iteration in this, right? It is all  $k, k, k$ , there is nothing about  $k - 1$  over here. So, if I just pose the question like this, does it give me the correct value of  $\nabla f$  at  $x_k$ , the real answer is I do not know. If I do not know, maybe there is a way for me to make it do it and in that way I am going to get some kind of an additional constraint, I can introduce  $B$  into the picture, ok. So, let us see that. So, in order for us to do this, I am standing here at  $x_{k+1}$ .

I came from here at  $x_k$ . If I want my model  $m_{k+1}(p)$ . So, let us pay attention here. I want this model to tell me something about  $x_k$ . If I put  $p = 0$ , it is telling me at behavior at which point?  $x_{k+1}$ , right. So,  $p = 0$  is  $x_{k+1}$ . So, what is that value of  $p$  so that it corresponds to  $x_k$ ? I have one answer over here, any other answer? What value of  $p$  should I put inside this  $m_{k+1}$  model, so that this function tells me something about  $x_{k+1}$ ,  $x_k$ ?

Minus right, because the way I arrived from here to here was by walking a distance  $\alpha_k p_k$ . That means if I am now already at  $x_{k+1}$ , I need to walk backwards right. Correct.  $\alpha_k$ , in this case do not worry about  $\alpha_k$ .

Let us assume that  $\alpha_k \times p_k$  is what we are talking about. So, this is going to be  $p = -\alpha_k p_k$ . If I substitute this inside my model  $m_{k+1}$ , it will tell me about  $x_k$ . So, let us look at that and also let us note this one thing over here. What is  $\nabla m_{k+1}(p)$ ? It is nothing but  $\nabla f_{k+1} +$  one more term was there, what was that? Plus  $B_k p$ .

So, this is going to be  $B_k + 1p$ . This was just so, in order for me to get  $\nabla f_k$  correct from  $m_{k+1}$ , I need to substitute this value over here. So,  $\nabla m_{k+1}$  evaluated at  $-\alpha_k p_k$ . This is going to give me information about the gradient at  $x_k$ , right. So, there are two ways in which I can compute this.

One is just substitute into the above formula, right? So that is going to give me  $\nabla f_k + 1 - B_k + 1\alpha_k p_k$ , right? I am just substituting.

Ideally what should this be? This should be  $\nabla f_k$ , okay? So this is where the choice of model or choice of  $B$  is coming into play. Because  $B$  was not specified so far, here is where the choice comes in over here. So this is a critical step. Critical, you can call it a design choice.


So, let us just rearrange these characters. So, you will have  $\nabla f_k + 1 - \nabla f_k$  is going to be equal to  $B_k \alpha_k$  minus looks like a rho p, looks a little not very intuitive what is going on over here. So, we are going to introduce some question,  $B_k + 1$ , sorry. So, this over here  $\nabla f_k + 1 - \nabla f_k$  there is a simple notation for it that we are going to introduce, I am going to call it  $y_k$ .

So, this is the definition of  $y_k$ . So, what is  $y_k$ ? Difference of gradients between two iterations. What is the  $\alpha_k p_k$  also known as? Can I write  $\alpha_k p_k$  in terms of the  $S$ 's, it is  $x_{k+1} - x_k$ , right? So, I am going to write this as  $B_k + 1S_k = y_k$ , right because  $x_{k+1} - x_k = S_k$  is also equal to  $\alpha_k B_k$ , okay. Now, this equation has come about by our choice. We have said I want this to be true and this equation is important enough to get a name of its own.

It is actually called the secant equation. Now, we wanted this matrix to be to have what property? Symmetric and positive definite. How do I ensure positive definite over here? if matrix  $B_k + 1$  is positive definite. The definition of positive definite is  $z^T A z$  greater than 0 for all  $z$ . So, in particular is there a nice choice of vector I can stick from the left over here,  $S_k^T$  right. So, if I left multiply by  $S_k^T$ , what do I get?  $S_k^T B_k + 1S_k = S_k^T y_k$  and this should be greater than 0 for  $B_k + 1$  positive definite, okay.

This is, if this is ensured, I get if I can somehow ensure this, my matrix  $B$  will be positive definite. So, how we can ensure this, I will come to in a, in a short while, okay. Now, notice one thing, I have said what are the two things I wanted about  $B_k$ ? It should be symmetric, it should be positive definite. Now, if I think of a  $n \times n$  matrix, how many unknowns are there in a  $n \times n$  matrix?  $n^2$ . If I tell you it should be symmetric, How many unknowns are there? The upper triangular, I mean one triangular part right.






$\text{LM by } S_k \rightarrow S_k^T B_{k+1} S_k = S_k^T y_k > 0 \text{ for } S_{k+1} \text{ r.v.}$   
 $B_k \text{ symm} \Rightarrow \frac{(n)(n+1)}{2} \text{ unknowns.}$   
 I have  $n$  constraints from secant eqn.  
 Remaining constraints up to us.  
BFGS relation  $S_k = B_{k+1}^{-1} y_k = H_{k+1} y_k.$   

$$H_{k+1} = (I - \rho_k S_k y_k^T) H_k (I - \rho_k S_k y_k^T) + \rho_k S_k S_k^T$$

$$\rho_k = (y_k^T S_k)^{-1}$$



So, that is roughly how much?  $\frac{n(n+1)}{2}$  right. So,  $B_k$  is symmetric implies  $\frac{n(n+1)}{2}$  unknowns. Now  $\frac{n(n+1)}{2}$  is greater than  $n$  or less than  $n$ ? Is greater than  $n$ , right. Now so far if you notice I have not, I have not given you enough information to determine  $B$ . Because  $B$  has how many unknowns? Roughly let us say  $\frac{n^2}{2}$ . How many conditions am I imposing on  $B$ ? The secant equation is it one constraint or  $n$  constraints? It is actually a matrix equation, right? So, it is actually  $n$  constraints.

Each row of that equation is giving me one linear equation in some of the elements of  $B$ . So, I have  $n$  constraints from the secant equation, So what does that mean? I do not have enough constraints. That is a good place to be. Why is that a good place to be? Because that means that I can do further design choices to supply the remaining missing pieces of information and which is why you do not have one quasi-Newton method, again you have a family of quasi-Newton methods. those are the additional constraints which you know people have worked out over the last maybe couple of decades which will give you different choices to nail down the remaining right.

So, remaining constraints So I will tell you, we will discuss the one very very popular constraint in the literature. I am sure many of you have probably heard of this. It is named after the inventors of this method. It is called the BFGS relation. Anyone heard of this BFGS? Okay, if you, when you are working with the MATLAB toolbox and optimization, sometimes it will ask you for choice of optimizer.

That is where you will see these different letters come up. So, this is one such choice. So, what is this choice? This, okay, so it is a long equation to write down. I will just write it down for sake of completeness, okay. Notice one thing, this secant equation over here, I wrote it like this, right? So,  $B_k + 1S_k = y_k$ .

There is another way of writing it. What is  $B_k^{-1}$ ? Hessian. Remember  $B_k$  was the proxy for Hessian inverse. So, you will find some equations they work with the inverse of the Hessian. I could have as well written the equation like this and this inverse has a special name, it is the Hessian. So, this BFGS relation what it does is, it gives us an update rule for  $H_{k+1}$  in terms of  $H_k$ .

It is helping us fix the other degrees of freedom. So,  $H_{k+1}$  is You are right. Yeah, okay. That is, when we wrote it, it was  $S$ , you are right, it is kind of backwards. I am not proving this, this has its own kind of proof over here, but this is in a sense this our quasi-Newton method is sort of working with this update rule, ok.