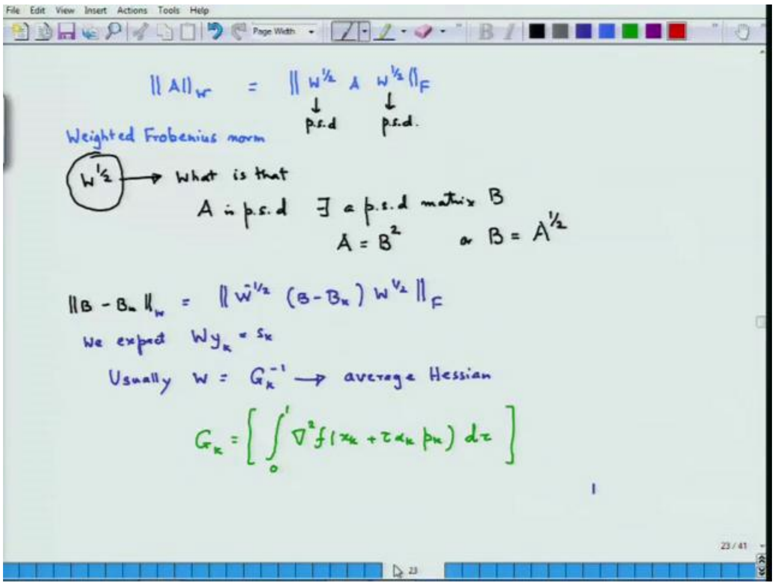
Foundation of Optimization

Prof. Dr. Joydeep Dutta
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture - 25

Let us start from where we had left. You see that we had computed how to actually compute that particular problem which was a essentially this one.

(Refer Slide Time: 00:33)



$$\|A\|_W = \|\underset{\text{p.s.d}}{\underset{\downarrow}{W^{1/2}}} A \underset{\text{p.s.d.}}{\underset{\downarrow}{W^{1/2}}}\|_F$$

Weighted Frobenius norm

$\boxed{W^{1/2}} \longrightarrow$ What is that

A is p.s.d $\exists$ a p.s.d matrix B

$A = B^2$ or $B = A^{1/2}$

$$\|B - B_k\|_W = \|W^{-1/2} (B - B_k) W^{1/2}\|_F$$

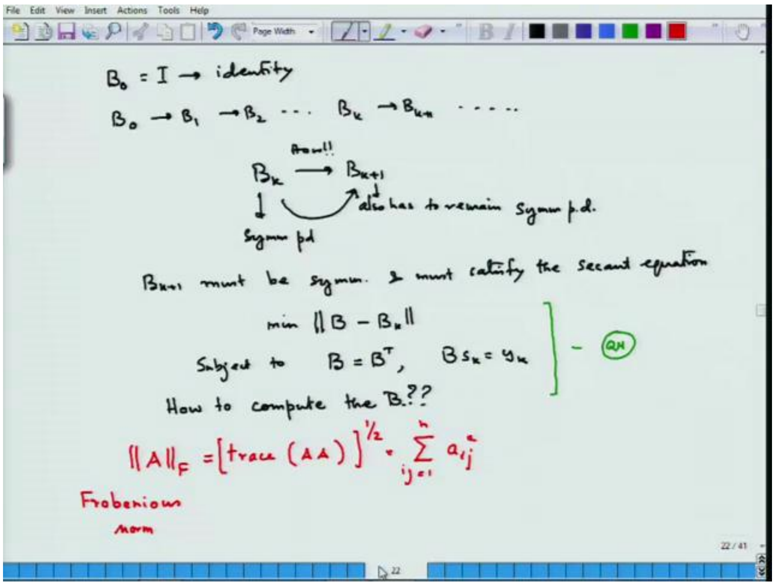
We expect $W y_k = s_k$

Usually $W = G_k^{-1} \longrightarrow$ average Hessian

$$G_k = \left[\int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k)\, d\tau\right]$$

Rather this one.

(Refer Slide Time: 00:34)



$B_0 = I \rightarrow$ identity

$B_0 \rightarrow B_1 \rightarrow B_2 \cdots B_k \rightarrow B_{k+1} \cdots\cdots$

$$\begin{tikzcd} B_k \arrow[r, "\text{How!!}"] \arrow[d] \arrow[r, bend right] & B_{k+1} \arrow[d] \\ \text{Symm pd} & \text{also has to remain Symm p.d.} \end{tikzcd}$$

B_{k+1} must be symm. & must satisfy the secant equation

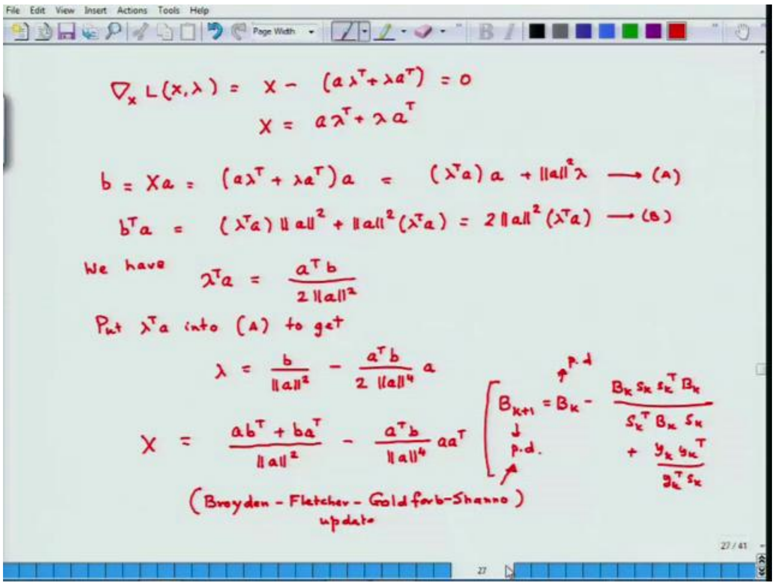
$$\left.\begin{aligned} &\min \|B - B_k\| \\ &\text{Subject to} \quad B = B^T, \quad B s_k = y_k \end{aligned}\right] - \boxed{\text{QN}}$$

How to compute the B.??

$$\|A\|_F = [\text{trace}(AA)]^{1/2} \cdot \sum_{i,j=1}^{n} a_{ij}^2$$

Frobenious norm

(Refer Slide Time: 00:36)



$$\nabla_x L(x, \lambda) = X - (a\lambda^T + \lambda a^T) = 0$$

$$X = a\lambda^T + \lambda a^T$$

$$b = Xa = (a\lambda^T + \lambda a^T)a = (\lambda^T a)a + \|a\|^2 \lambda \longrightarrow (A)$$

$$b^T a = (\lambda^T a)\|a\|^2 + \|a\|^2(\lambda^T a) = 2\|a\|^2(\lambda^T a) \longrightarrow (B)$$

We have $\lambda^T a = \dfrac{a^T b}{2\|a\|^2}$

Put $\lambda^T a$ into (A) to get

$$\lambda = \frac{b}{\|a\|^2} - \frac{a^T b}{2\|a\|^4} a$$

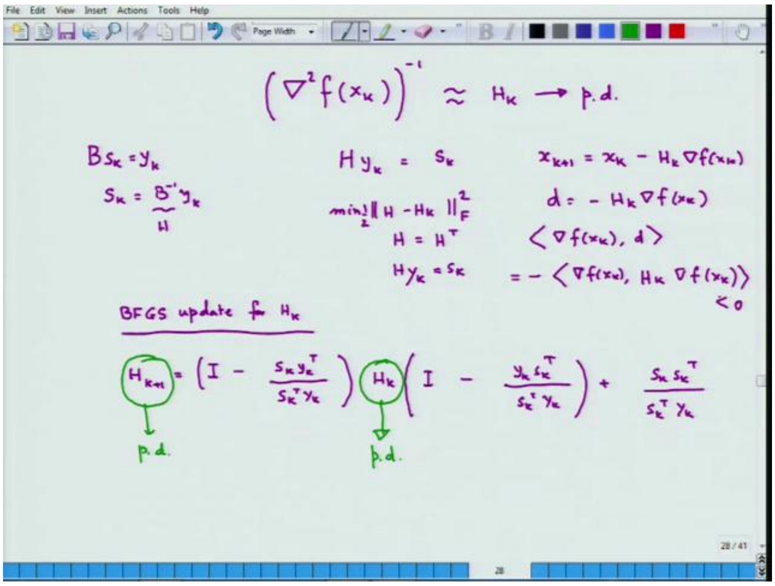
$$X = \frac{ab^T + ba^T}{\|a\|^2} - \frac{a^T b}{\|a\|^4} aa^T$$

$$\left[\begin{aligned} &\overset{\text{p.d}}{\uparrow} \\ &B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \\ &\downarrow \\ &\text{p.d.} \end{aligned}\right.$$

(Broyden - Fletcher - Goldfarb-Shanno) update

And we did not do it in detail, but we showed you the most important or rather the most useful possibly from a computational point of view, the updating formula due to Broyden Fletcher Goldfarb and Shanno. The importance of this is that this is positive definite, when b k is positive definite and B k plus 1 is positive definite. The problem however is that if you have B k plus 1 positive definite or b k positive definite, it it is fine some sense, but the problem is that you again have to compute the inverse when you are actually computing the iteration. So, how do I stop myself from doing this expensive computation of the inverse, where the matrix is large then computing the inverse could lead to troubles.

(Refer Slide Time: 01:35)



$$\left(\nabla^2 f(x_k)\right)^{-1} \approx H_k \rightarrow \text{p.d.}$$

$$B s_k = y_k$$

$$s_k = \underset{H}{\underset{\sim}{B^{-1}}} y_k$$

$$H y_k = s_k$$

$$\min \tfrac{1}{2} \| H - H_k \|_F^2$$

$$H = H^T$$

$$H y_k = s_k$$

$$x_{k+1} = x_k - H_k \nabla f(x_k)$$

$$d = - H_k \nabla f(x_k)$$

$$\langle \nabla f(x_k), d \rangle$$

$$= - \langle \nabla f(x_k), H_k \nabla f(x_k) \rangle < 0$$

BFGS update for H_k

$$\underset{\downarrow \atop \text{p.d.}}{\boxed{H_{k+1}}} = \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) \underset{\downarrow \atop \text{p.d.}}{\boxed{H_k}} \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}$$

So, we can now do something else, we can choose instead of trying to approximate the Hessian, I try to now approximate the inverse of the Hessian matrix in a Newton scheme with some matrix H k which is p d so usually this is naturally because I want this to be p d. So, basically H k in some sense is acting as the inverse of b k. So, in that sense the equation that H k should satisfy is not H k or B k y k equal to, like B k y k equal to S k or rather b k s k equal to y k. This would satisfy the reverse things. So, we have B k S k is equal to y k. That is exactly what we had done, right?

Now, if I… B is invertible, again I have S k sorry B k plus, B is some matrix like this then B inverse where B is p d then I have it like this. So, this is my H and basically H would be an equation that, H would be an (( )) and symmetric p d matrix which will be satisfying H or y k is equal to S k. So, in that sense the problem that I really want to solve in order to find H now. If I find H I do not have to invert, is this problem. Of course, it is symmetric the same story. This is what I need to actually. Now, of course, you can put this as half of this square. What is important at this stage is to understand that here if I take H then my Newton iteration simply means.

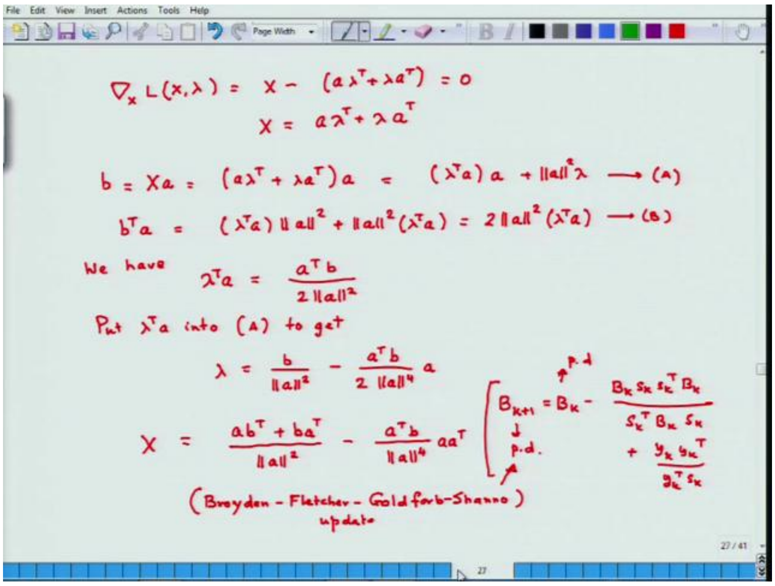
So, even if I do not have much information about twice differentiability of the function, I can assume twice differentiability. It is always good to assume twice differentiability. I can still do certain operations like this with with a positive definite matrix because again anyway if I put d is equal to minus, it is quite simple to note that this is a decent direction because this is nothing but minus of, is strictly less than 0. Now, if I solve this system, if I solve this system what will I get? I will get the following like the one I got for the Broyden case earlier Davidion Fletcher Powell case.

So, the in just like the Davidion Fletcher Powell update, I will get a similar sort of B F G S update, but now with in terms of H. So, B F G S update. See, if I just do a p d matrix, take a p d matrix basically when I start with x 0 x 0 is I and then use this updating formula and use this without even bothering whether the function is so as continuously differentiable, only knowing that it is just continuously differentiable I can actually get a, get something, some sort of solution by applying this approach.

Now, let me look into this carefully this solution of this, under though in approach that we had earlier stated would turn out to be… So, these all consists of a rank to update because here we have rank one matrix plus another rank one matrix, this one rank one matrix into another rank one matrix plus a rank one matrix. So, please remember these are the matrix. So, the interesting feature that we again get is the following.

If this H is p d that is positive definite, symmetric for p d. It is symmetric of course, because we have assumed this, that this has to satisfy this then this is also p d. I would not ask you to prove this because you can if you wish to prove it. Those who are more mathematically inclined, but it is enough for the audience here to really know that this is an update which I can use.

(Refer Slide Time: 07:46)



$$\nabla_x L(X, \lambda) = X - (a\lambda^T + \lambda a^T) = 0$$

$$X = a\lambda^T + \lambda a^T$$

$$b = Xa = (a\lambda^T + \lambda a^T)a = (\lambda^T a)a + \|a\|^2 \lambda \longrightarrow (A)$$

$$b^T a = (\lambda^T a)\|a\|^2 + \|a\|^2(\lambda^T a) = 2\|a\|^2(\lambda^T a) \longrightarrow (B)$$

We have $\lambda^T a = \dfrac{a^T b}{2\|a\|^2}$

Put $\lambda^T a$ into (A) to get

$$\lambda = \frac{b}{\|a\|^2} - \frac{a^T b}{2\|a\|^4} a$$

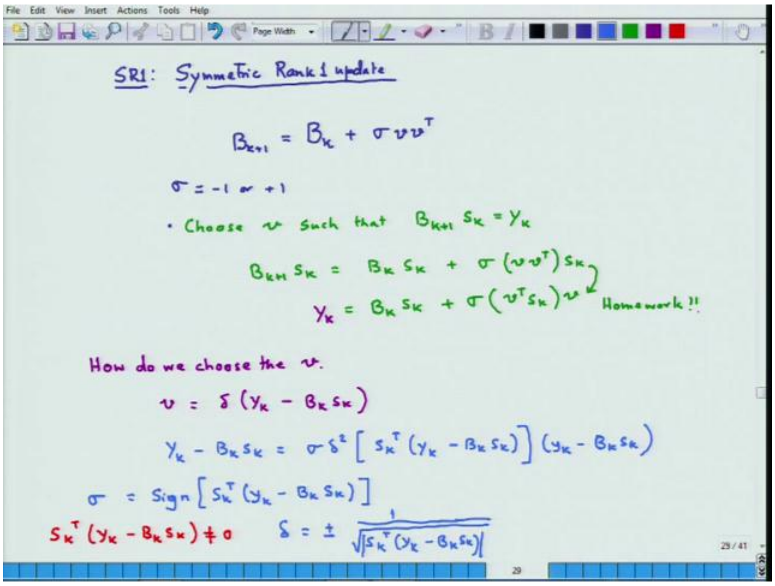
$$X = \frac{ab^T + ba^T}{\|a\|^2} - \frac{a^T b}{\|a\|^4} aa^T$$

$$\left[\underset{\substack{\downarrow \\ \text{p.d.}}}{B_{k+1}} = \overset{\substack{\text{p.d} \\ \uparrow}}{B_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \right.$$

(Broyden - Fletcher - Goldfarb-Shanno)
update

And you can compare this with the formula that I have for B where you see basically here y k has been replaced by S k. So, but observe there isn’t a symmetry in this formula. Symmetry in this formula means what the formula for H and formula for B looks quite different, the updating formula. So, people sometime use a formula called the S R 1 method or the symmetric rank one method.

(Refer Slide Time: 08:26)



SR1: Symmetric Rank 1 update

$$B_{k+1} = B_k + \sigma v v^T$$

$\sigma = -1$ or $+1$

- Choose v such that $B_{k+1} s_k = y_k$

$$B_{k+1} s_k = B_k s_k + \sigma (v v^T) s_k$$
$$y_k = B_k s_k + \sigma (v^T s_k) v$$ Homework!!

How do we choose the v.

$$v = \delta (y_k - B_k s_k)$$
$$y_k - B_k s_k = \sigma \delta^2 [s_k^T (y_k - B_k s_k)] (y_k - B_k s_k)$$
$$\sigma = \text{Sign} [s_k^T (y_k - B_k s_k)]$$

$s_k^T (y_k - B_k s_k) \neq 0$ $\quad \delta = \pm \dfrac{1}{\sqrt{|s_k^T (y_k - B_k s_k)|}}$

Where the formula for B and the formula for H has a symmetry, they look similar though y is replaced by, y k replaces S k and vice versa. So, the symmetric rank one method. So, here we have rank two matrices. So, if we come here. So, this is a rank one matrix because s k into s k transposes rank one matrix. So, this finally, this will be a, all the whole thing would be a rank one matrix. So, this is a rank one matrix. This is a rank two matrix at the maximum, so but here also it is rank two, but here we will just update it with rank one matrix.

Now, a symmetric rank one matrix is a, rank one update is adding to B k say, a rank one matrix v v transpose and that is what I call as B k 1 plus 1. Now, the problem with symmetric rank one matrix is that if B k is p d there is no way I can tell that b k plus 1 is p d, but that drawback is actually covered up by the simplicity of the whole thing and simplicity in implementation of this thing. And numerical experience has shown as Nocedal and Wright the book which I am using for this lecture partly is for example this book, Nocedal and Wright numerical optimization which published by Springer and now available in Indian edition.

So, it is a must for those who study optimization. So, you would observe that what he says that numerical experience has told us that it is not harm, not a harm to use this S R 1 update and in many cases it is not been p d sometimes becomes helpful. Now, the problem is that you can ask that if B k is p d and B k plus 1 is not p d possibly then why the method is useful for, how are you going to write down a algorithm with this method. The issue here is the following.

The issue is that you write down the algorithm with this method because of the following facts that nowadays there is a powerful group of techniques called the trust region method which I believe is not really within the scope of this syllabus, but I will talk about the trust region method as a, almost at the last some sort of special lecture you can say. So, the important part that I want to discuss here is that once we take the trust region approach we will show you the trust region approach and we will show you the use of symmetric rank one update in that lecture.

So, once we take that the trust region approach then in fact the indefiniteness of B k 1, B k plus 1 might turn out, in fact turns out to be in its advantage and that is a very, very important thing to realize that even without this I can actually do something if I use this more advanced trust region techniques. So, how do I do the update? That is very, very important. Here sigma is either minus 1 or plus 1, and then then you have to chose this v in such a way that the second equation has to be satisfied.

Choose v such that because that is the major requirement of the trust, of the quasi Newton method. So, you have to choose v in such a way. So, how do I choose that v? So, let me now operate this. So, this can be written as when you take this vector and multiply with the matrix basically you have. So, with this as a vector that we are multiplying so you can write this as because I can take v transpose S k as some sort of associative stuff. In fact if this is a matrix into S k you can show that this is same as v transpose S k into v, when you have this rank one matrix this is exactly what will happen.

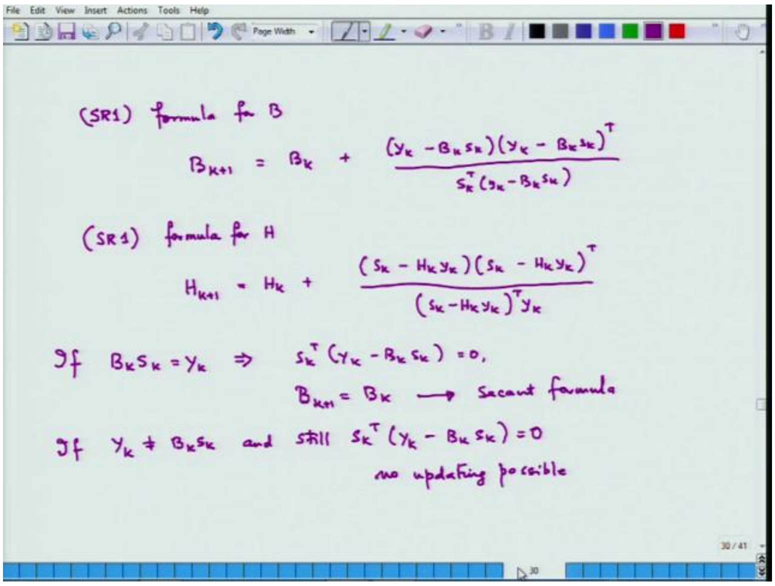
So, this is something you figure out from here to here figure out how it comes by homework. Just take a two by two matrix and just check it that it is correct and then use induction. Now, once you have that the question would be how do you choose the v. So, my question is… Let me choose v as delta times. So, basically you would have so B k plus 1 S k. So, B k plus 1 S k here is your y k. So, y k minus B k S k is sigma times this. So, how would you choose v to get something, alright?

So, let us choose it like this and if I put it back into this equation and then what I will get is so I will leave this to you to figure out. So, I am plugging this thing in place of v here. Obviously, being the B k minus S k to this side. So, once I do that then what I will have is the following. Let me now choose sigma to be the sign of. Now, if I want sigma to be plus 1 or minus 1 basically sign functions are that if it is 0 this is 0, if it is positive it is plus 1, if it is negative it is minus 1, but here we will not we should not take 0 because a thing like this would soon come in the denominator.

So, here we will take up this condition that for this time being this is not equal to 0. So, depending on the sign of this in a product, sigma would be either plus 1 or would be minus 1. Then that would allow us to choose delta as follows, plus or minus 1 by square root of S k transpose. Now, once I have done that then what I have is the following. Again, I put back all those things here, v sigma is plus or minus 1, it depends on because v, with v you will have the delta so delta would have a plus or minus 1.

So, if this is positive sorry a so if this sign is positive delta is plus 1 or it is minus 1. So, if it is negative this is minus minus plus so ultimately here you will always have the plus. Remember, you cannot have this could be negative so you cannot have a square root of negative number in optimization because we are dealing with real quantities. Now, if plug it in I will get the following result following update formula for B.

(Refer Slide Time: 19:57)



(SR1) formula for B

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T (y_k - B_k s_k)}$$

(SR1) formula for H

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}$$

If $B_k s_k = y_k \Rightarrow s_k^T (y_k - B_k s_k) = 0,$

$B_{k+1} = B_k \longrightarrow$ Secant formula

If $y_k \neq B_k s_k$ and still $s_k^T (y_k - B_k s_k) = 0$

no updating possible

S R 1 formula for B, so that would turn out to be more simpler than the others, but keeping its drawbacks. So, this forms a rank one matrix and then I divide by whose of course sigma has to be in parallel to the sign of this minus if minus plus is plus and so ultimately this will be plus and because you have taken square it will come once because you have delta. Delta is that to the power half because in, it will come from both the v’s. So, it will become delta square and it will have formula like this. Now, what I want to say is that here if I look at the formula for H it would be like this.

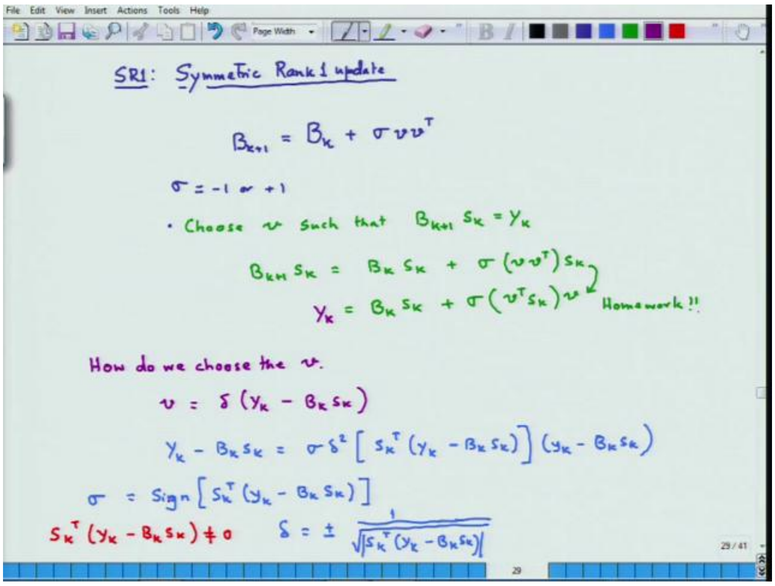
So, you see look at the symmetry. Only, B k is replaced by H k, y k is changing. y k is replaced by S k and B k is replaced by H k and everything else remains same. So, that is why it is called the symmetric rank one updating formula. Actually, if say B k S k is equal to y k that implies that S k transpose y k minus B k S k is equal to 0 then sigma would actually be 0 and the only possible updating would be B k plus 1 is same as B k.

Now, the the problem is if y k is not equal to and then actually you do not have to bother about the sigma’s being 0 here. I would say rather you can actually come to this from a very simple approach because otherwise I can always say that I would like to look at it through sigma equal to 0, but that is not the way people look into that in the literature because when y k is not equal to this and still, then usually there is people do not write down any symmetric update formula.

See what happens here is the following. You can, you actually have B k S k is equal to y k and the way Nocedal and Wright approaches, it is that they I I would actually put the sigma equal to 0 and try to get something but it does not really matter for me this equal to 0 means the same thing you you do not really update, but they are telling that if this is there then from the second formula I can say B k plus 1 equal to B k and get B k plus 1 which is same satisfying the second formula.

So, there they are telling that this updating is possible from the second formula and here no updating is possible, no updating is possible. So, it is said that this case actually marks the this simple approach we have taken here compared to the other more sophisticated approaches of B F G S and D F P. So, there are obvious reasons that why we go to, why we go to talk about this, why we go to talk about the, this S R 1 formula is that we can provide a simple thing to stop the breakdown of this.

(Refer Slide Time: 26:31)



SR1: Symmetric Rank 1 update

$$B_{k+1} = B_k + \sigma v v^T$$

$\sigma = -1$ or $+1$

- Choose v such that $B_{k+1} s_k = y_k$

$$B_{k+1} s_k = B_k s_k + \sigma (v v^T) s_k$$
$$y_k = B_k s_k + \sigma (v^T s_k) v$$

Homework !!

How do we choose the v.

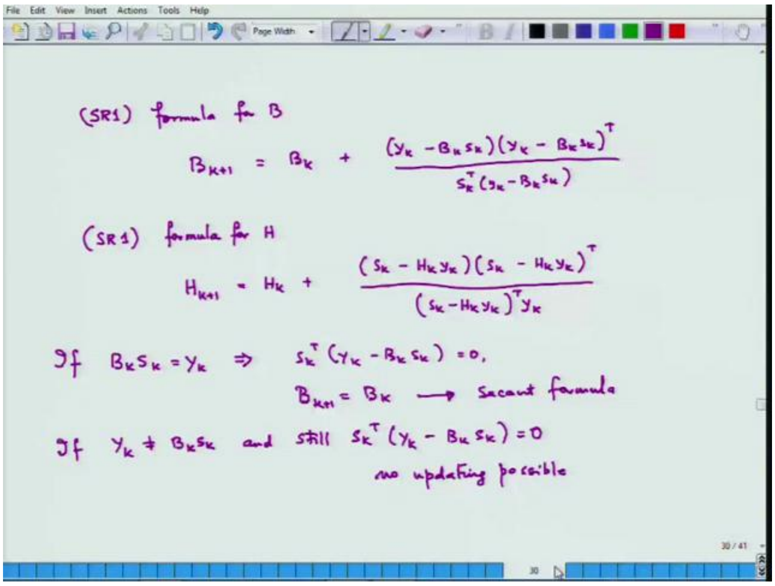
$$v = \delta (y_k - B_k s_k)$$
$$y_k - B_k s_k = \sigma \delta^2 [s_k^T (y_k - B_k s_k)] (y_k - B_k s_k)$$
$$\sigma = \text{Sign} [s_k^T (y_k - B_k s_k)]$$

$s_k^T (y_k - B_k s_k) \neq 0$ $\quad \delta = \pm \frac{1}{\sqrt{|s_k^T (y_k - B_k s_k)|}}$

You see sigma function here, there are lot of ways one can define it. Either you define it this is positive this is 1, this is negative this is minus 1, what about it is 0 what sign you will take, usually it is given the value 0, but here possibly they want to maintain just two cases and in Nocedal and Wright. So, we will just follow them and say that okay. You must be asking why this teacher follows a book while he is talking about algorithms and why does not he follow the book completely when he talks about theory because the speaker is an optimization theorist and not really an expert in algorithms though every optimizers knows knows some algorithms.

Of course, he has to because this is one of the most important part and parcel of the subject. So, hence in order to give the audience a better view point of the algorithms it is always better to go to the experts in optimization algorithms who might not likewise be very comfortable when doing advanced optimization theory.

(Refer Slide Time: 27:43)



(SR1) formula for B

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T (y_k - B_k s_k)}$$

(SR1) formula for H

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}$$

If $B_k s_k = y_k \Rightarrow s_k^T (y_k - B_k s_k) = 0,$

$B_{k+1} = B_k \longrightarrow$ Secant formula

If $y_k \neq B_k s_k$ and still $s_k^T (y_k - B_k s_k) = 0$

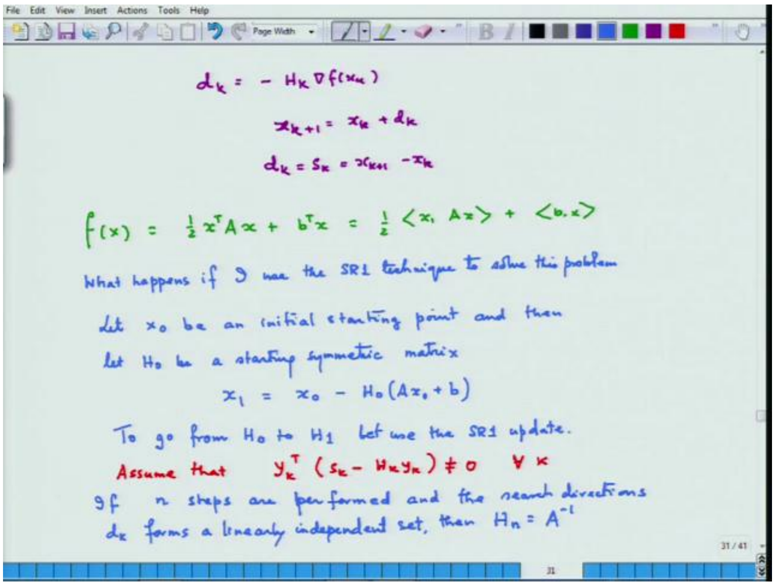
no updating possible

Now, I would say for me in both cases the only possible updating would be either B k plus 1 equal to B k or do not update at all. It, in numerical practice the updates generated by this S R 1 type thing is much better update of the Hessian than B F G S of any other approximation and the problem is that in quasi Newton method if you want to apply them for the un constraint case then it becomes very difficult to maintain the curvature formula. You you you must have seen the curvature formula at the very beginning that is fine. If you want the second condition to be true then necessarily curvature condition has to be true.

So, if curvature condition is not true then the second condition is not true so which means the curvature condition really has to be checked at every stage, that okay the curvature condition is holding. If the second condition holds then curvature condition will naturally hold. So, because if B k plus 1 is a symmetric positive definite. So, the question is a, what is this, that in this case in in constraint cases it is very difficult for curvature conditions to hold.

So, if curvature conditions are not holding in the constant case then there is no other choice if you want to apply quasi Newton to, you use to take this symmetric rank one approach. Now, symmetric rank one has a very, very important property, it is almost like a conjugate gradient property. So, there is a link between the two anyway.

(Refer Slide Time: 29:38)



$$d_k = -H_k \nabla f(x_k)$$

$$x_{k+1} = x_k + d_k$$

$$d_k = s_k = x_{k+1} - x_k$$

$$f(x) = \frac{1}{2}x^TAx + b^Tx = \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle$$

What happens if I use the SR1 technique to solve this problem

Let x_0 be an initial starting point and then let H_0 be a starting symmetric matrix

$$x_1 = x_0 - H_0(Ax_0 + b)$$

To go from H_0 to H_1 let use the SR1 update.

Assume that $y_k^T(s_k - H_k y_k) \neq 0 \quad \forall k$

If n steps are performed and the search directions d_k forms a linearly independent set, then $H_n = A^{-1}$

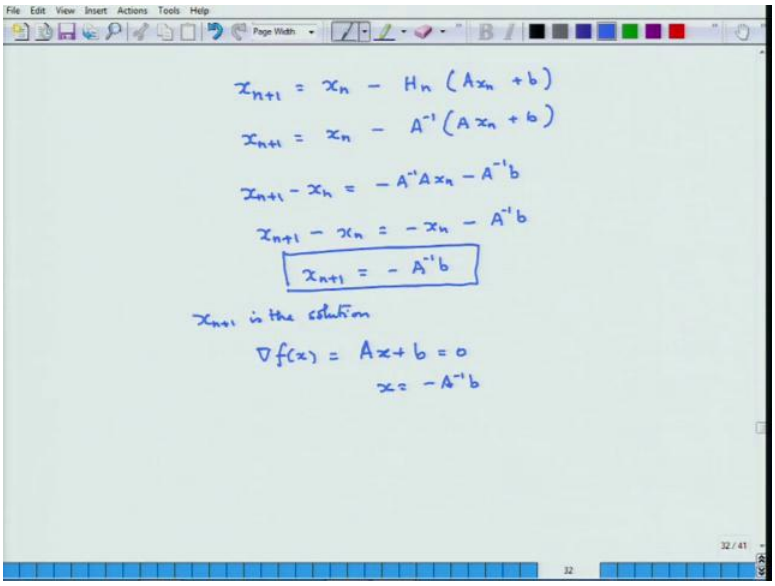
Now, if I put the direction of descent as minus H k grad f k, grad f x k then you can write x x k plus 1 is equal to x k plus d k and this d k is now your S k because this is same as… Now, we consider the use of the quasi Newton method for the case when my function f x is a quadratic function. We had studied them while we were studying conjugate gradient methods. It is same as writing like this, you know it is just and what would happen if I apply S R 1 there where my Hessian is always positive definite.

So, let me take a starting point x naught. So, take… Now, my question is what happens if I use the S R 1 technique to solve this problem? Now, let x naught be a, be an initial starting point and then H naught be a starting symmetric matrix. So, you have H 0 is given to you. So, you will go from x 0 to x 1 by writing x 0. It was a gradient of that. So, it is A of, now once you come to H 0 to go, once you go to x 1 now to get the H 1 from where you can again make the thing you can have to use S R 1 update, to go from H 0 to H 1 let us use the S R 1 update. Let us use the S R, let us use the S R 1 update.

Now, if that is the case then now, assume so this is a very important thing. Assume that y k transpose S k minus H k y k. So, it is just the thing you need for the particular H k type updating. This is not equal to 0. We will assume this fact. So, this is a very very thing true for all k, suppose this happens. So, this is a very important step if n steps are performed then performed and each search direction, search direction d k it is S k plus 1 minus S k, forms a linearly independent set or I can write and the search directions my

English has to be correct also and the search directions d k forms a linearly independent set then of course, then interesting result that we have is H of n is nothing but a inverse. So, if H of n is a inverse then what are we are going to get?

(Refer Slide Time: 36:09)



$$x_{n+1} = x_n - H_n (A x_n + b)$$

$$x_{n+1} = x_n - A^{-1} (A x_n + b)$$

$$x_{n+1} - x_n = -A^{-1} A x_n - A^{-1} b$$

$$x_{n+1} - x_n = -x_n - A^{-1} b$$

$$\boxed{x_{n+1} = -A^{-1} b}$$

x_{n+1} is the solution

$$\nabla f(x) = Ax + b = 0$$

$$x = -A^{-1} b$$

Then we are going to get x n plus 1 is equal to x n minus H n A x n plus b. So, x n plus 1 is equal to x n, H n A inverse A x n plus b. So, x n plus 1 minus x n is equal to minus A inverse A x n minus A inverse b. So, that would mean that x n plus 1 minus x n equal to minus x n minus A inverse b which means that x n plus 1 is equal to minus A inverse b and thus x n plus 1 is the solution. So, n, after n iterations you know what is the solution. So, this is just the solution because H n becomes A inverse and you know in general when you have a quadratic function then you are minimizing over r n and you are Hessian is positive definite, then you have grad of f of x is A x plus b.

So, if you want to equate that to 0 then x is nothing but minus A inverse b. So, any if x equal to minus A inverse b it is definitely the solution because it satisfies grad A x equal to this thing equal to 0, this will, this will become 0 and of course, because this is convex function every critical point is a global minima, so this is a global minima and hence x n plus 1 here you see is a solution. So, the question is how do you prove this fact? So, we will start in our next class by proving this fact, tell you slightly more facts about the rather some sort of convergence of the quasi Newton method and then go over to the special lecture on linear programming that I promised.

Thank you very much.