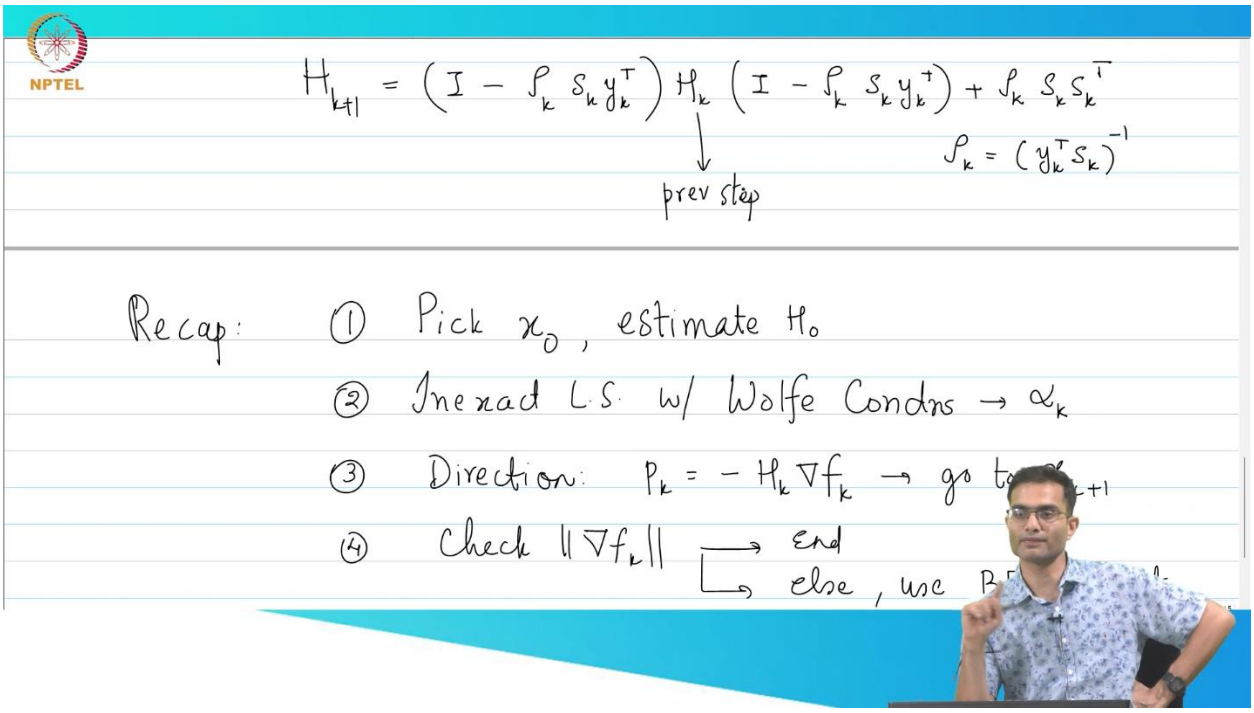
Course Name: Optimization Theory and Algorithms
Professor Name: Dr. Uday K. Khankhoje
Department Name: Electrical Engineering
Institute Name: Indian Institute of Technology Madras
Week - 08
Lecture - 54

BFGS method

So now let me, it seems like leaving things incomplete, but actually the description of the quasi-Newton method is complete. Why? Let us try to just quickly summarize how we would start. So at step 0, what do I like with any algorithm? What is step 1? Pick a starting point. Pick some x_0. Now notice this BFGS relation. Can I compute it from scratch? For example, at $k = 0$, what do I need in order to kick start this expression? I need H_0.



Somehow you need to tell me H_0. So, I am going to say pick x_0 and estimate H_0, ok. So, maybe you need to pay a one-time price to get H_0, ok. Next, what would I do? Correct, that is one choice.

If it were very expensive, then maybe I may need to do some approximation there also, but otherwise H_0 could come from the Hessian, ok. Then what do I need to do? I need to do To go from x_0 to the next place what do I need? I need α, I need p, right. So, for α what is the standard procedure? In exact line search with Wolfe conditions. Ok, I got it. Do I know which direction to go to? I do.

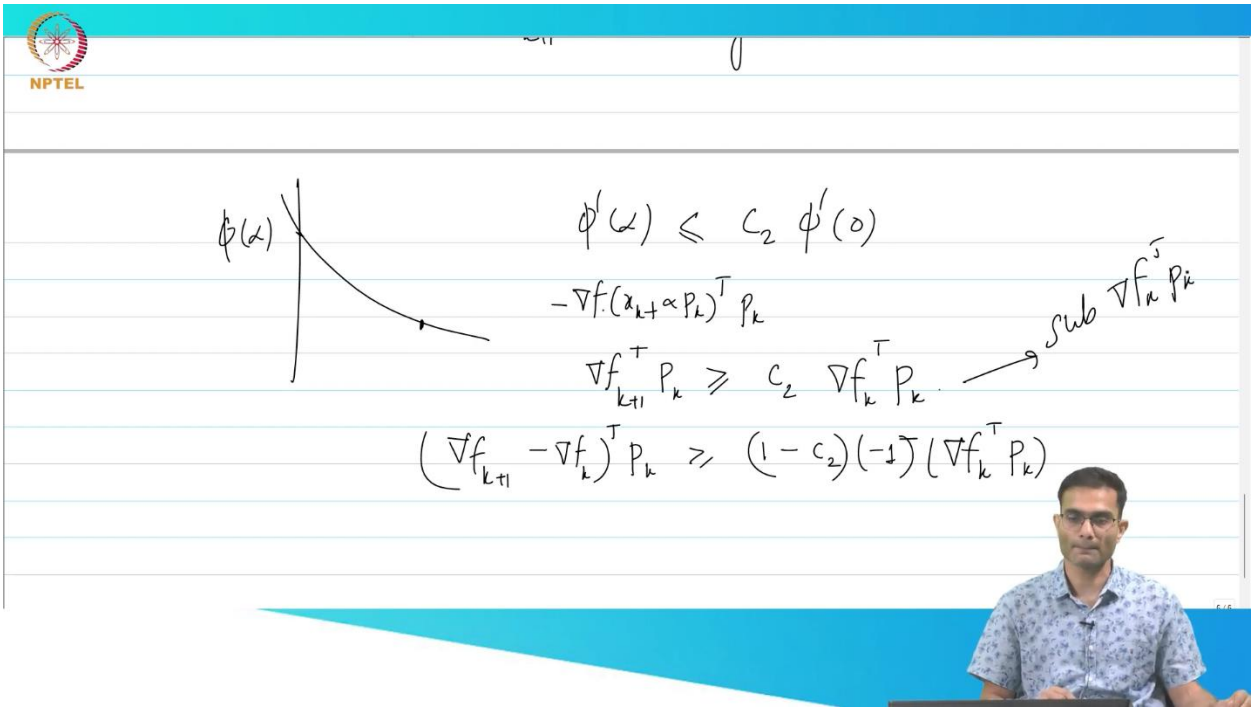
How do I know it? I calculated it right at the beginning, right. So, direction is simply $p_k = -H_k \nabla f_k$, right. So, if you notice right at the beginning, where did it go? Yeah, here, $B_k^{-1} \nabla f_k$. Now, B_k^{-1} the notation I have got for it is is this, ok. So, this will allow me to go to the, right.

So, that means go to x_{k+1}, right. What should I do next? Compute, ok. Before I do that, should I do something else? I have moved to the next point. Correct, we should always have a check for convergence, right. So, I would say for example, check $\| \nabla f_k \|$, ok.

If it is low enough, end. Else what should I do? Should I go to step 2, step 3, what should I do? I need to update my H because unless I update my H I cannot compute the direction which is sitting in step 3 right. So, here is else use BFGS to get H_{k+1} then 2 and 3 right. So that is our quasi-Newton method. Did I ever need to compute a Hessian? I never needed to compute a Hessian, right? Because my BFGS relation for example, what does it need? This guy is coming from the previous step, right? S_k is coming from what? The difference of the previous two, right? S_k was defined as? we had it over here right, difference of iterates $x_{k+1} - x_k$.

So, if I know where I am and if I know my previous step I can calculate S. Similarly, y I can calculate right and so I have my S_k, I have my y_k and what else, ρ is given over here. So, basically this whole expression can be computed. $S_k S_k^T$, quick check, what is $S_k S_k^T$ into scalar vector matrix? rank one matrix and so this entire expression is a matrix. So, given the previous H_k I can update this calculate the new H and move on, ok.

You will find actually there is something even more popular than BFGS in the relation and particularly in MATLAB you will find there is something called LBFGS. L stands for a low memory version of the BFGS which is extensively used in optimization. where you do not want to do, where you want to do Quasi-Newton method this is one. So, these letters BFGS they stand for the founders of this method, ok. So, the one thing that we left kind of hanging is this over here.

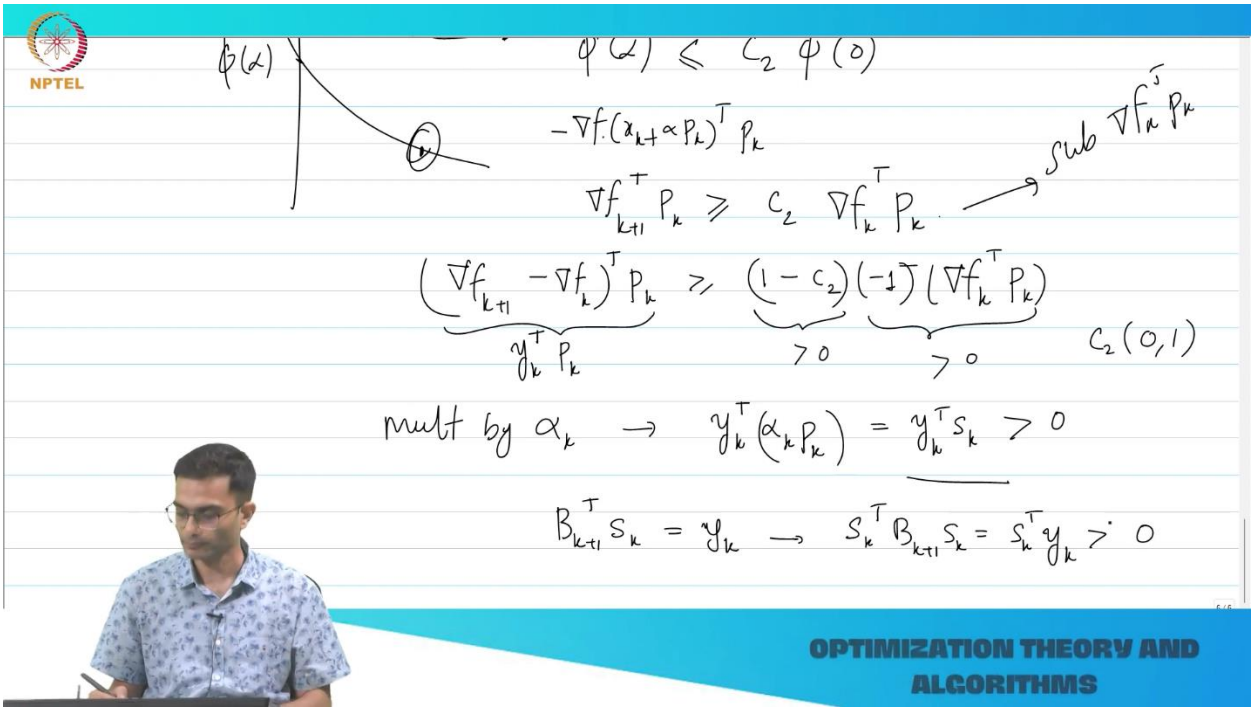


We never, so far I have not shown you or given you a guarantee that this B that I get is going to be positive definite. BFGS is fixing the remaining sort of degrees of freedom, but how do I ensure that it is positive definite that is still left to me. Now, it turns out, I will just, we will work

it out right now. Our Wolfe conditions, they save the story in innumerable examples and this is one more of them. So, it is a very cute little result.

Let us show it to you over here. It says that if α_k satisfies the Wolfe condition in particular the curvature condition, then B_{k+1} is always surprising result, right. And it is not only surprising, the proof is also very simple. So, what was the curvature condition? If you remember the English you will be able to tell. What was it? Remember we had something like this.

So, what did it say? This was my $\phi(\alpha)$. I said that the slope at $\phi'(\alpha)$ should be less than what? C_2 times $\phi'(0)$. If you remember it in words, you will always remember what is going on over here. Now $\phi'(\alpha)$, remember the way I have drawn this graph over here, $\phi'(\alpha)$ is positive or negative? Negative, right? So and this is the curvature condition, not the strong curvature condition. So there is no absolute sign over here.



So in order for me to write down this over here, I had to put a minus sign. So $\phi'(\alpha)$ was what? $\nabla f(x_k) + \alpha p_k^T p_k$, right? And similarly for this we can write it over here. But in order to get the slope over here with a positive sign I had to put a negative, ok. That is how we had worked it out. So, basically what this would translate to is $\nabla f_{k+1}^T p_k$ is greater than or equal to $C_2 \nabla f_k^T p_k$.

If you are little rusty on your curvature condition, this was a quick recap, ok. Now, supposing I subtract $\nabla f_k^T p_k$ from both sides, what will I get? Left hand side is going to give me $\nabla f_{k+1} - \nabla f^T p_k$ greater than or equal to, I am going to write this as $(1 - C_2) \cdot (-1) \cdot \nabla f_k^T p_k$, ok. It should be $C_2 - 1$, I am writing it as $(1 - C_2) \cdot (-1)$, ok. What is ∇, is this ok with everyone? We just subtracted over here, subtract $\nabla f_k^T p_k$ from both sides. What is this left hand side? $\nabla f_{k+1} - \nabla f$ is what? y_k.

So, this is $y_k^T p_k$. So, when I wrote this expression, the first expression, it was intuitively that this slope should be less than, I mean the slope at this point α should have reduced to less than C_2

times the original slope. Magnitude of this slope. Magnitude, right. And to get the magnitude I multiplied by minus sign to get it as a positive number.

These are all valid descent directions. Correct. These are all valid descent, this is a valid descent direction. That is the intuitive way to remember. So, I get $y_k + y_k^T p_k$ on the left-hand side.

What is $1 - C_2$? Is it greater than 0 or less than 0? It is greater than 0, right? Because C_2 was between 0 and 1. So, this is greater than or equal to 0. What about this guy? p_k is a legit descent direction. So, $\nabla f_k^T p_k$ is less than 0 multiplied by minus 1, this is greater than or equal to 0, right. So, if I multiply this by a positive number α_k, what will I get, right? If I multiply by α_k, I am going to get $y_k^T \alpha_k p_k$, but $\alpha_k p_k$ is nothing but $x_{k+1} - x_k$ which is s_k.

So, this is $y_k^T s_k$ and we have seen that this is greater than or equal to 0, right. Now, if you look back at what we had said over $s_k^T y_k$ should be greater than 0, then this is going to be positive definite. $s_k^T y_k$, do I have that? $y_k^T s_k$, same thing, right. So, the secant equation was $B_{k+1}^T s_k = y_k$, right. And then when I left-multiplied by s_k^T, I got $s_k^T B_{k+1} s_k = s_k^T y_k$ and we just showed that this is greater than 0.

Actually there is one, it should not be, yeah this should, this is greater than 0 and it is a legitimate descent direction therefore this angle cannot be, they are all strict, strictly greater than because C_2 was, sorry C_2 was in the open interval $(0,1)$. So, this term cannot be 0. So, it is not greater than or equal to, it is greater than. So, implies that d_{k+1} is.

Correct. So, right. So, let us see. Well, it is at least true for And do I need it actually here we need to look at the proof, do I need it for other vectors that is something to think about. I will think about that. It is not clear to me looking at this BFGS relation whether or not that would be the case.

Maybe it will be. Yeah, we need to look at this a little bit more. So, if I start with a positive definite H_k, will H_{k+1} also be positive definite? Yeah, I do not want to make a casual comment. Let us look at this a little bit more carefully. The same matrix is the way. So, Wolfe conditions have helped us earlier also, it is helping us now also.

So, whenever you do a backtracking line search, it is always a good idea for the termination condition of the backtracking to be one of the Wolfe conditions. So, this essentially completes our, you know, whatever description that we needed to learn about the quasi-Newton way, quasi-Newton method, right? The recap is over here. We know, we have all the pieces in place. What is the design choice or you can say the place where you have room for innovation could replace BFGS by something else. The secant equation is going to be there that does not change.

The remaining $n^2/2 - n$ relations are going to come from something else. BFGS is one example of it. So, when you, this is another something for you to look at in your course project, choices, different ways of implementing quasi-Newton, this is one example, ok. and as we said this is super linear in convergence, ok. And as far as coding goes they are all relatively similar complexity.

This is in fact simpler to compute than Newton because there is no Hessian computation involved, right, ok. So, what we have done, I mean if we just take a quick recap of the different line search methods that we have done. We have started with steepest descent, we looked at it in

great detail, we move to conjugate and we move to the non-linear version which is non-linear conjugate gradient and then we looked at the Newton method and the quasi-Newton method. So, in terms of tools of optimization you have got a nice range of tools available with you depending on how difficult or computationally challenging the problem is you can choose one or the other method, ok.