Distributed Optimization and Machine Learning

Prof. Mayank Baranwal

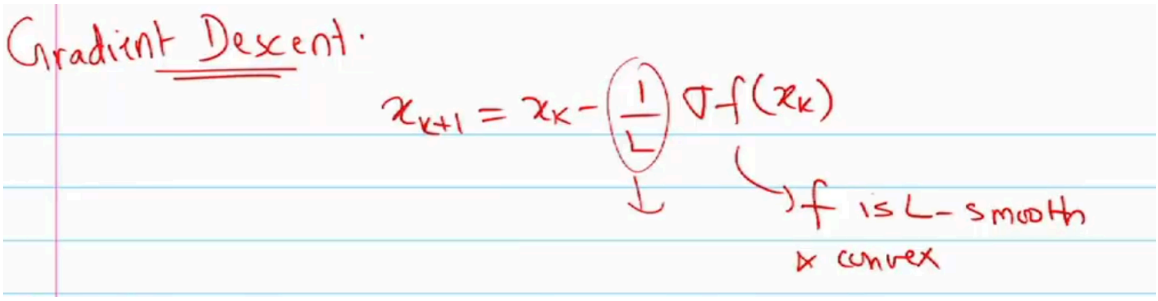
Computer Science & Engineering, Electrical Engineering, Mathematics

Indian Institute of Technology Bombay

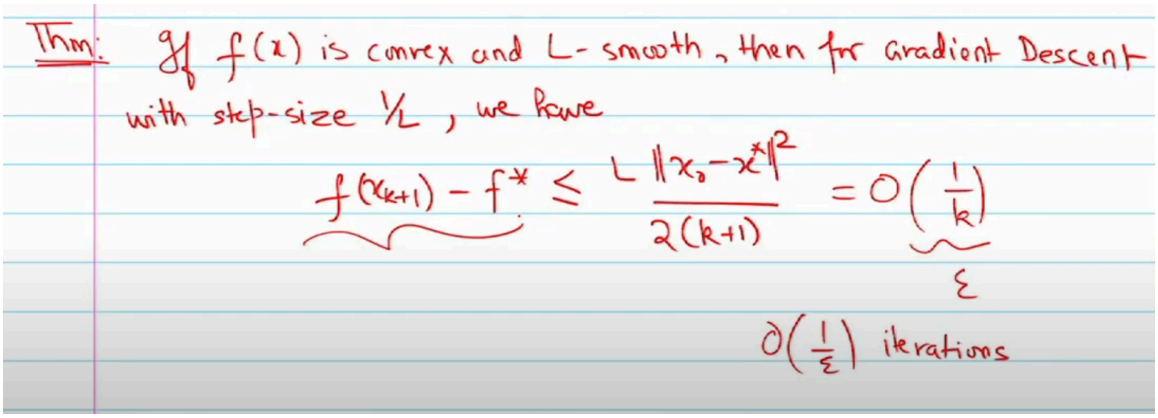
Week-4

Lecture - 12: Acceleration under strong convexity

Alright, so in the last lecture we kind of looked at gradient descent algorithm and we found that gradient descent with a step size of 1 over L. So, we assume that f is L smooth and convex. So, this has I mean in some sense we saw that 1 over L was a suitable choice for the learning rate right and we looked at this particular theorem. So, let me rewrite it. It is convex and L smooth. Then for gradient descent with step size 1 over L, we have So, this is like order 1 over k kind of convergence and as I said as we discussed in the last class this 1 over k you can equate this to epsilon and that means if you want to be epsilon close to the optimal solution you have to have order 1 over epsilon number of iterations.



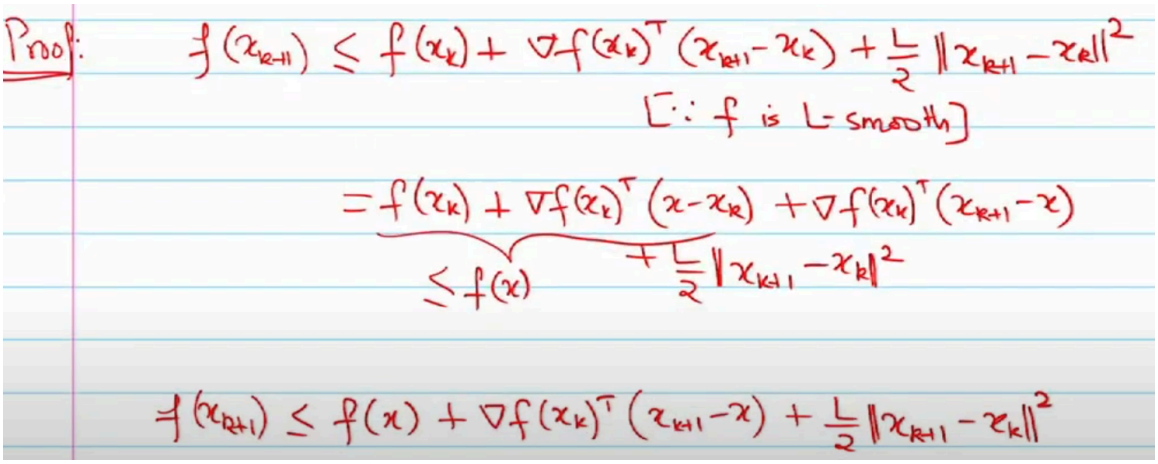
that means in order to be epsilon in order to make this difference epsilon close to each other. So, you have to have these many iterations. So, let us look at a way to basically let us derive this particular result. And the question the natural question that we are going to be answering is and that is through the subsequent in this lecture and maybe the next lecture.



So, right now we know that if you want to be epsilon close we would need to have order 1 over epsilon number of iterations, but can we accelerate this even further. So, there are two ways to accelerate it either you assume more structure on your function. So, let us say if I also make f to be strongly convex. then that means we are assuming more structure to the function. So, for this restricted class we can possibly accelerate even the gradient descent algorithm.

The other way to accelerate optimization is you do not assume further structure into your into the function, but maybe you assume more structure or maybe you would come up with a different type of algorithm and that is another way to accelerate optimization of particular function. So, that is that is what we are going to do in today's lecture. ok, but this is the first result that we are going to be deriving all right. So, what do we know f of x is L smooth right and from last lecture we know that using Taylor's expansion this is going to be true right. this is since f is L smooth.

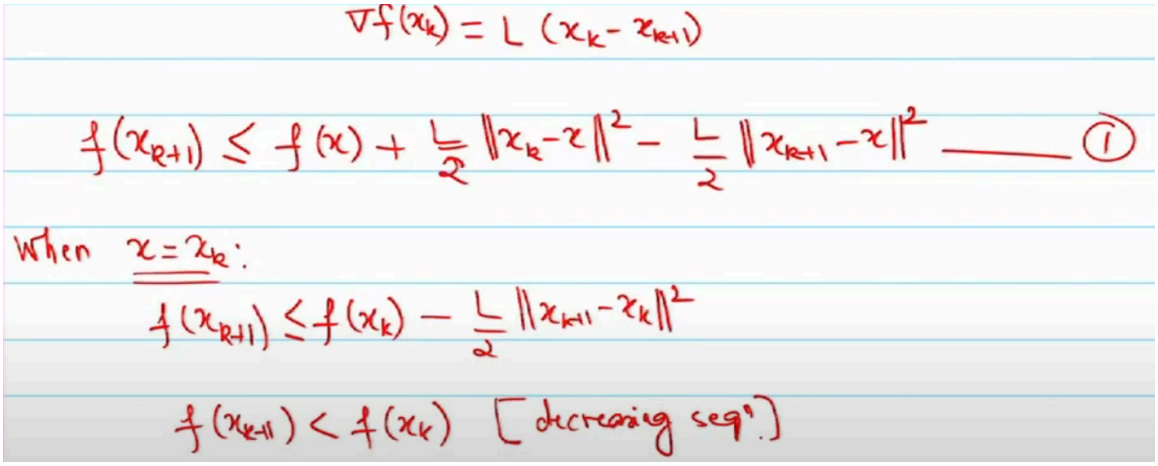
Is everyone with me on this? So, what do we know about this function f? It is L smooth and convex. So, somewhere we would have to use the convexity argument. and the only way or at least by looking at what kind of grade like terms we have on this right hand side that gives us idea as to how to suitably use this argument and for this what we are going to do is we are going to add and subtract terms. So, I am include instead of x k plus 1 I am writing x here and this So, just adding and subtracting x to it and the last term is there as it is. Now, is there a way for me to use convexity? So, what does this term because f is convex, what is the first order condition for convexity? So, this should be less than this will be less than equal to f of x right.



So, basically this allows us to rewrite everything as group x k plus 1 plus Alright, so how do we proceed further? So, what else do we know? So, we know that f is convex, we have used this. We know that f is l smooth, we have used this. What else we can use from this statement? That it is a gradient descent algorithm right with a step size 1 over L. So, obviously the final thing that we can use is x k plus 1 is x k minus 1 over L gradient of f of x k. So, this is because we run gradient descent with step size 1 over L right and this basically allows us to write gradient of f of x k in terms of x k and x k minus 1 x k plus 1 ok.

So, you can do a bit of algebra and you can show that this basically comes down to f of x k is less than f of x plus I think L by 2 x k minus x 1 square minus L over 2 plus 1 minus x. So, this is I mean you can just substitute this value back and then you can with little bit of algebra you can show that this is what it comes down to. So, let us call this equation 1 and in this equation if I choose let us say if I choose x to be x k or let us say let us say I choose x to be x k. So, what do we get? f of x k plus 1 is less than or equal to f of x k. the second term is 0 and the third term is L by 2.

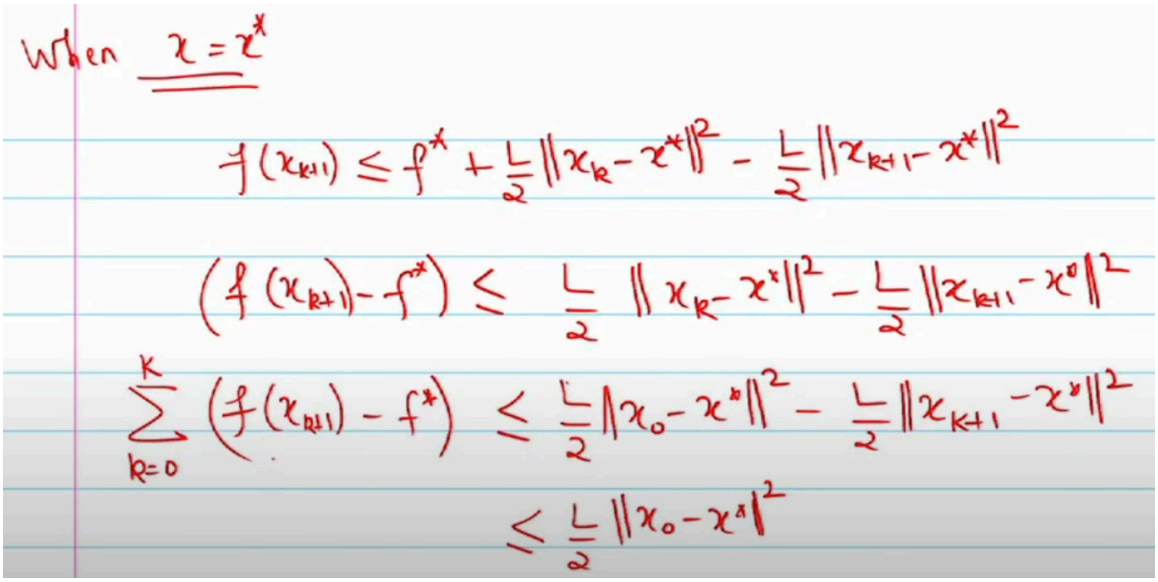
So, this implies that if this difference is non-zero, then f of x k plus 1 is strictly less than f of x k right, if this difference is non-zero and therefore, it is a decreasing sequence. decreasing sequence the optimal value is going to be f star bounded from below. So, it is going to be it we know that bounded like if the sequence is monotonic and bounded it is going to converge right. So, this is decreasing sequence and we know it converges ok. So, that is one thing everyone with me on this.



So, in the same equation 1 So, if I choose x to be x star. So, let me revisit equation 1. So, I get f of x k plus 1 less than equal to x star, f of x star or which is f star, then you get L by 2 x k minus x star and L by 2 x k plus 1 minus x star. So, let us do that. So, do you now see the result coming up from here? How can we do that? So, we just take this to the left hand side and then we just do a telescopic sum.

or just sum it over from. So, if I just sum it over both left and the right hand side from k equal to 0 to let us say k some let us say capital K or ok. So, that gives me. So, what is the right hand side evaluate to? So, you get L by 2 x naught minus x star square minus L by 2 ok. So, this is less than equal to L by 2 ok.

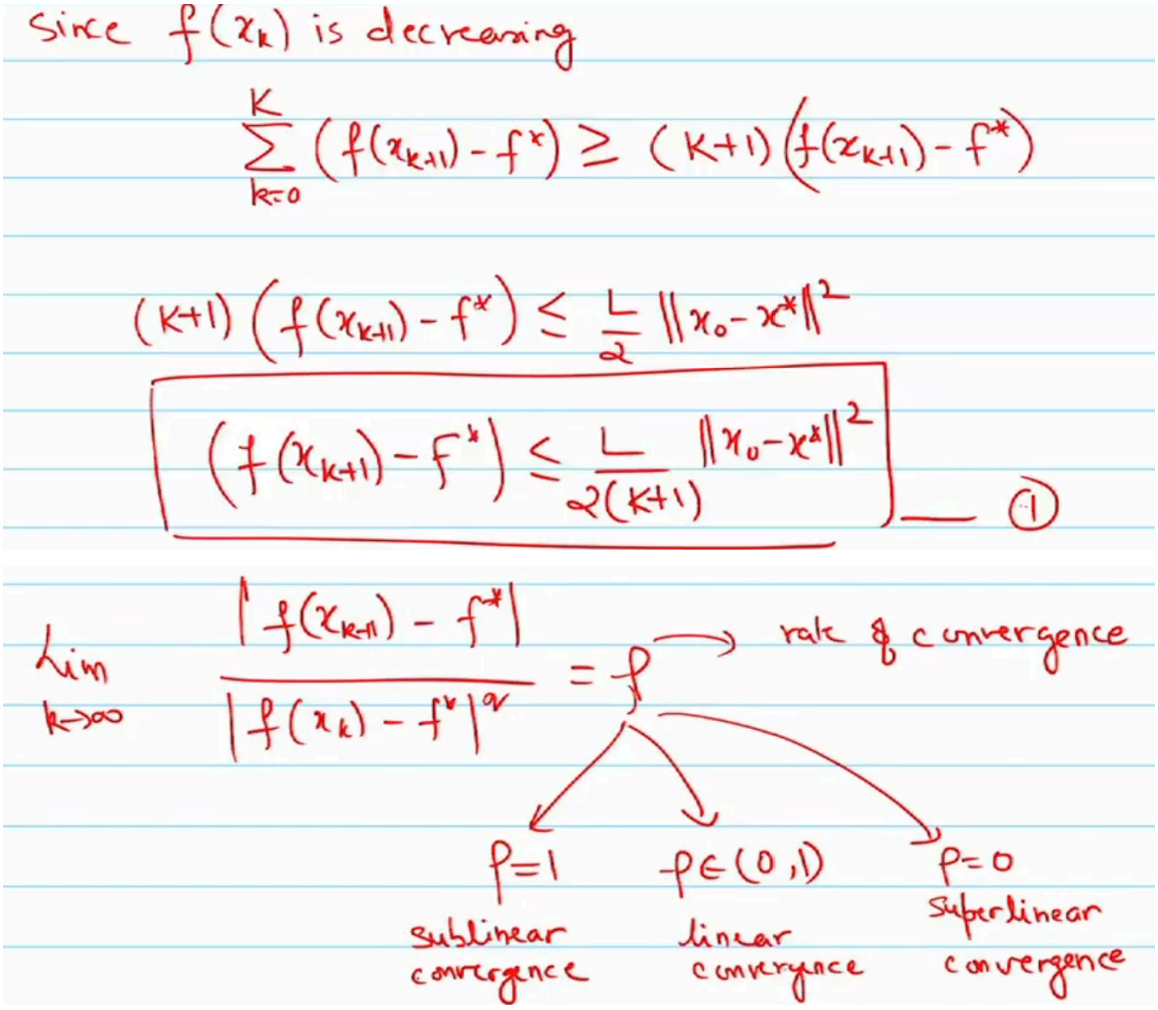
Is this clear? right and what do we know about f of x k? It is a decreasing sequence. So, this is bounded from below. So, this term essentially is greater than or equal to or let me rewrite it. We know that this summation ok right because it is a decreasing sequence right. So, now from here what do we get? k plus 1 f of x k plus 1 minus f star that is less than equal to L time L by 2 and this basically gives us a result which is Is this clear to everyone? So, this is gradient descent when function is L smooth and convex.



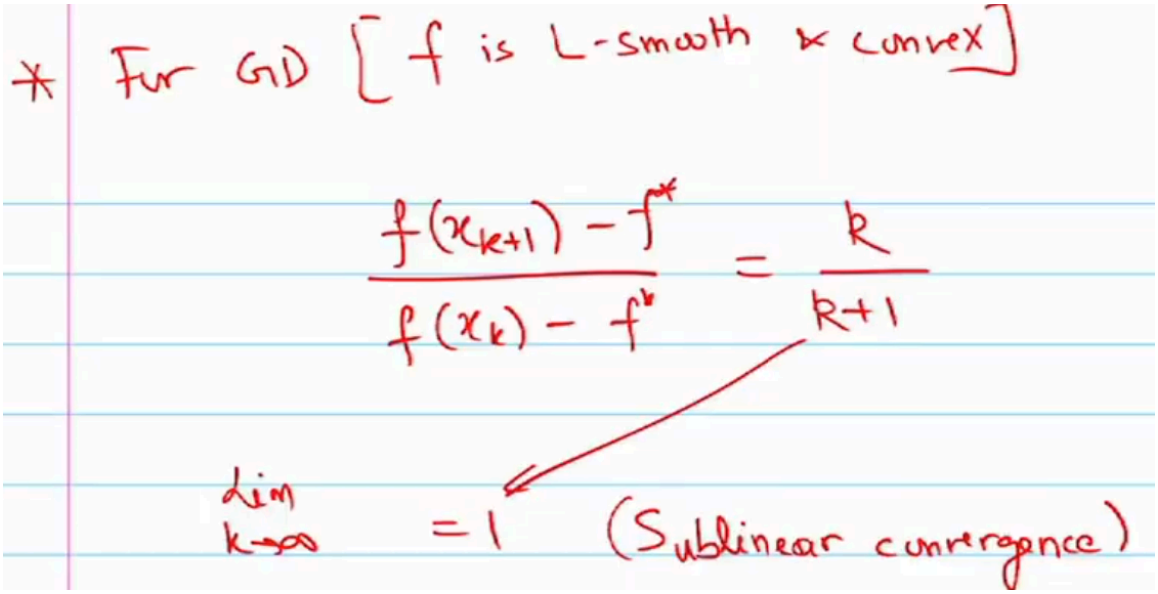
So, in the last class we also looked at something called rates of convergence and the idea was, so if we have a sequence f of x k plus 1 minus f star limit k goes to infinity, this is equal to rho and rho, let us say also consider q here. So, this was the rate of convergence. Now, this value rho, so let me, so if rho is equal to 1, what do we call? So, we call it sub-linear convergence. If rho is a number between 0 and 1, we call it linear convergence and if rho is equal to 0, we call it super linear convergence. What happens if rho is greater than 1? is not even convergent right, it is a diverging sequence.

Because this ratio let us say for q equal to 1, this ratio would start diverging if rho is greater than 1 right. So, we only consider the ranges rho is equal to 0, rho between 0 and 1 and rho equal to 1. And based on which based on the value of rho, we define the rate of convergence whether it is sub-linear, linear or super-linear. Is this clear? So, what is the rate of convergence in this gradient descent algorithm here? So, let me so for gradient descent we assume f is L smooth and convex. So, f of x k plus 1 minus f star divided by f of x k minus f star.

So, this is equal to k over k plus 1 right and limit k goes to infinity, what does this term, what is the limit for this? 1 right. So, this limit is 1 which means sub-linear convergence. So, gradient descent for l smooth function, it is sub-linear convergence. is this thing clear? So, k goes to infinity this is this one k plus 1. So, yeah I mean in most cases you I mean yeah it is less than equal to what I mean you usually write order 1 over k kind of convergence right.

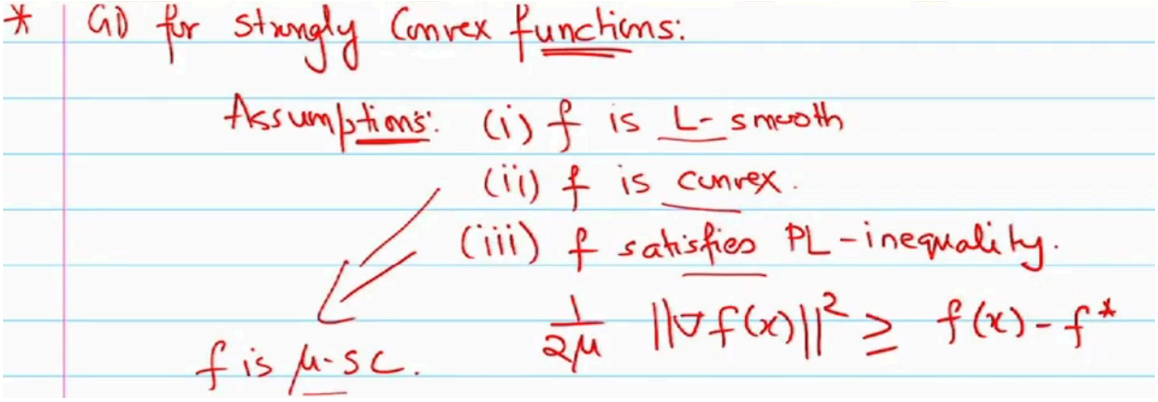


So, that is what yeah ok. So, we know that The other thing that we can notice from here that we have something that we have already mentioned you get order 1 over. So, number of iterations required is order 1 over epsilon right. So, we have an algorithm which is gradient descent and the case that we assume is L smoothness or l smoothness plus convexity and for this number of iterations required is order 1 over epsilon. Now, as I said you can in basically provide faster convergence guarantees in two ways either you assume more on your function.



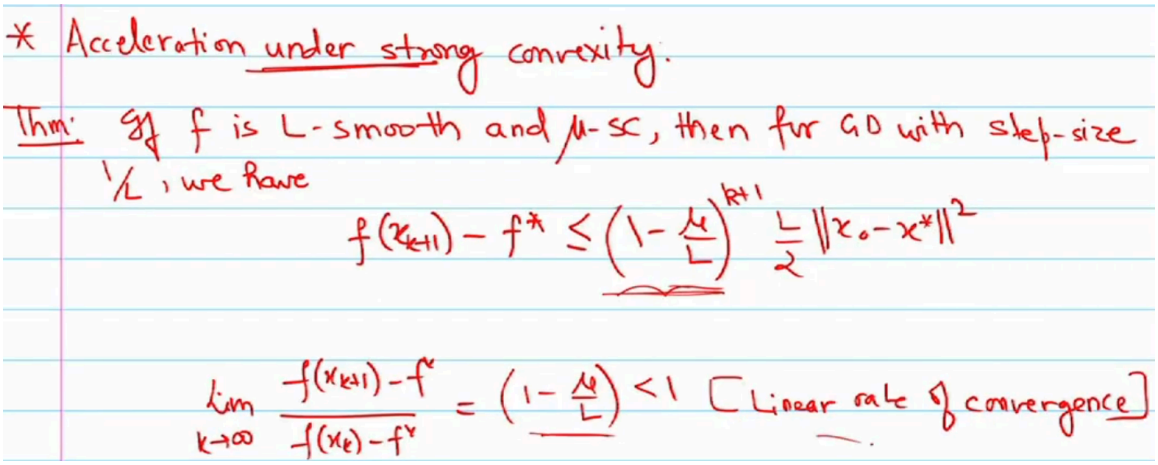
So, you assume that function let us say it is strongly convex and because you are now dealing with the restricted class of function you can provide better convergence guarantees or you change the algorithm right. So, let us first look at the gradient descent itself, but we assume more. So, we assume that the function is strongly convex. So, gradient descent So, what are the assumptions? So, again in discrete time in most cases we always assume that like when we talk about convergence rates implicitly it is assumed that the f is always L smooth. So, l smooth the second assumption is that f is convex and the additional assumption that we make is f satisfies PL inequality.

So, what was PL inequality? So, 1 over 2 mu ok. So, this was the PL inequality and we showed that every strongly convex function satisfies PL inequality the converse may not be true right. But then if we know that f is convex and f satisfies P L inequality unless f is a constant function these two assumptions together implies f is strongly convex or f is mu strongly convex ok. So, we assume that f is L smooth and f is mu strongly convex. So, under these two assumptions let us see if we can derive a better rate of convergence So, let me write the theorem first, if f is L smooth and mu is strongly convex.



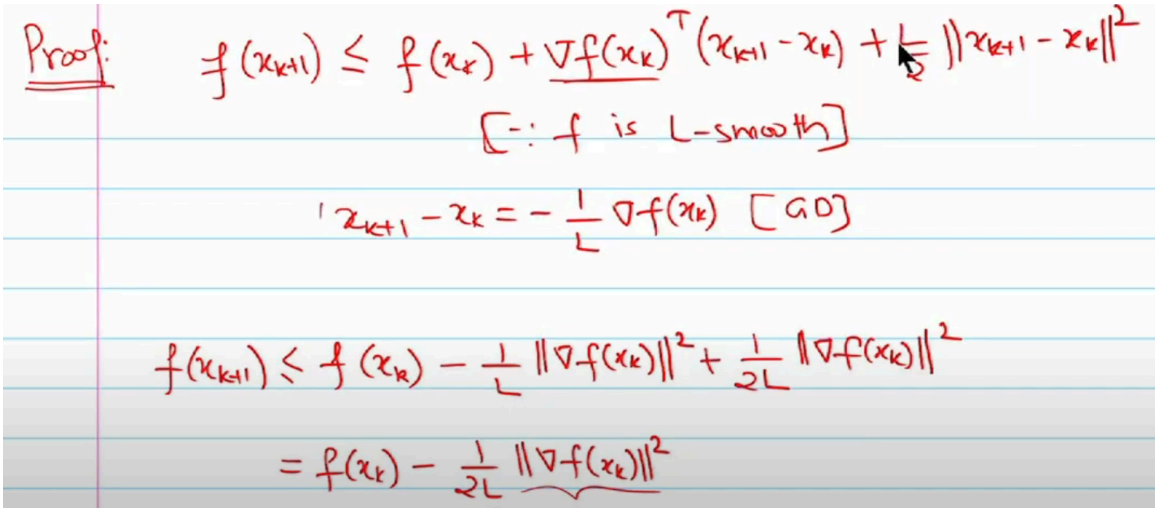
then for gradient descent with well with step size 1 over L we have. So, earlier this term was 1 over 2 k plus 1 right ok. So, when we looked at the non strongly convex case what was this term equal to? This was like 1 over k plus 1, but now the same term is this particular term over. So, 1 over k plus instead of having 1 over k plus 1 you get this kind of coefficient sitting in front. So, now if I try to talk about the rate of convergence.

So, this would be what? So, limit k goes to infinity f of x k plus 1 minus f star. So, this is of the form 1 minus mu over l right which is the number strictly less than 1 right. So, that means linear rate of convergence. So, we have already established in previous lectures that mu is less than or equal to L right. So, this number is a number between 0 and it may include 0 depending on whether mu is equal to L, but otherwise mu is let us say less than L.



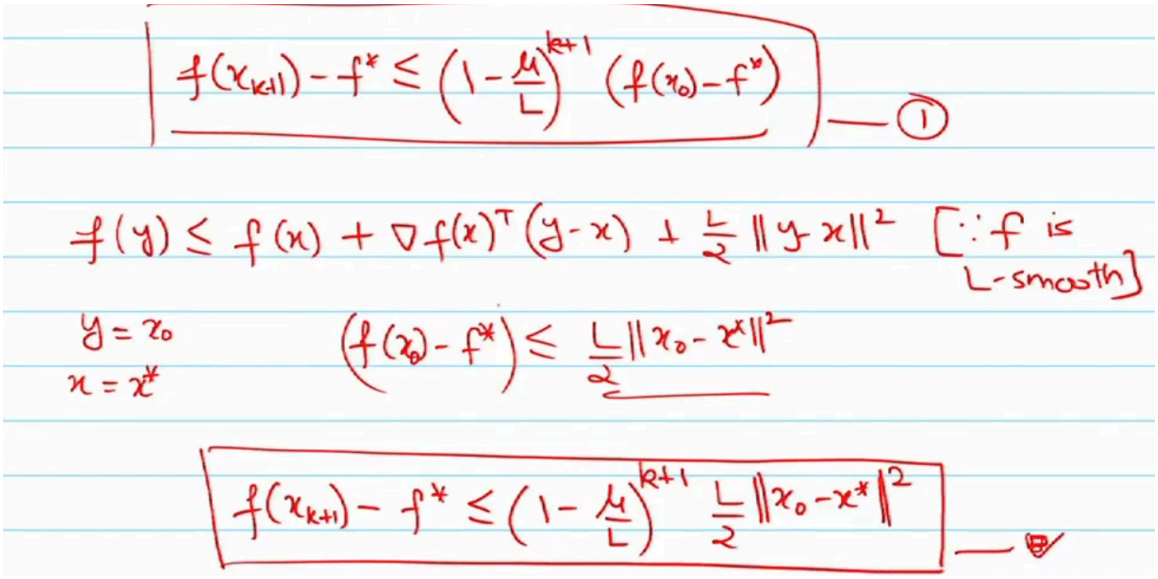
So, in that case it is this particular term is less than 1 and you get linear rate of convergence ok. Let us see how we can obtain this particular result. So this is basically again using the Taylor's expansion and the fact that f is l smooth. So since f is L smooth, so this particular statement is true. So now we want to use the fact that we somehow want to first of all write everything in terms of gradient ff of x k, why? Because we want to use PL inequality right.

Since it is a gradient descent x k plus 1 minus x k is what? minus 1 over L gradient of f of x k, since this is gradient descent, so gradient descent. So, therefore, if I substitute for x k plus 1 minus x k, if I substitute this for minus 1 over L gradient of f of x k, so what do I get, f of x k plus 1. So, which basically is equal to f of x k minus 1 over 2 L right. So, we now got a term which is norm gradient square and we know since f is strongly convex or satisfies sphere inequality, we can actually write this in terms of f of x and f star So, that was the reason why we wrote it the way we wrote it, because we somehow want to use this particular inequality. And in order to use that peer inequality, I can rewrite this as minus mu over L times 1 over 2 mu and this is less than equal to f of x k minus mu over L f of x k minus f star and this follows from peer inequality.



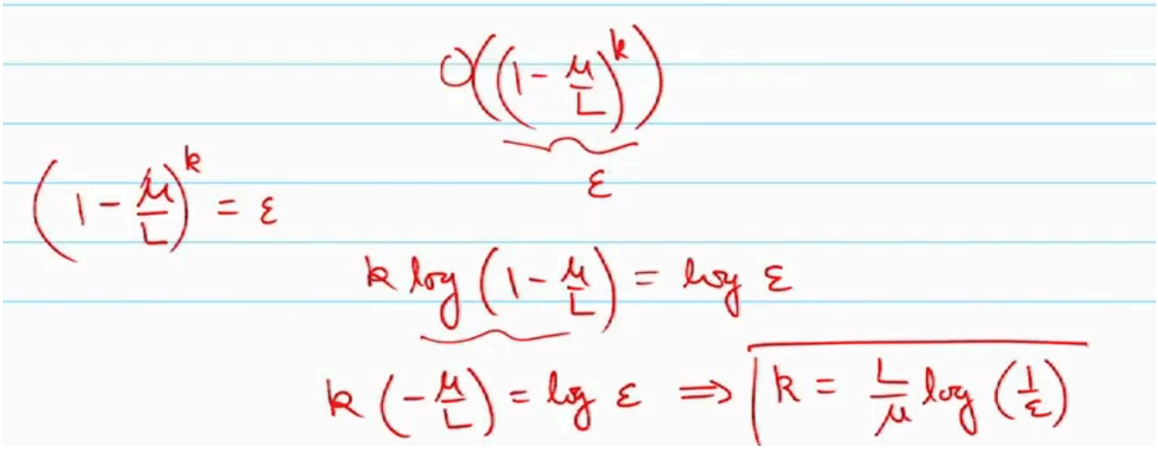
Is this clear? which basically tells you that f of x k plus 1 minus f star is less than equal to 1 minus mu over L f of x k minus f star ok. And if I apply this from k equal to 0 to k plus 1, what do I get? f of x k plus 1 minus f star is less than equal to 1 minus mu over L k plus 1 f of x naught minus x star. Now, we want the result in the form of x naught and x star, but what do we have here? f naught and f star right or f of x naught and f star. So, how can we get that? So, if I want the result in terms of something as l over 2 x naught minus x star, so somehow we have to use the l smoothness of the function and for that if I look at this particular or this particular equation or inequality over here and I choose.

So, let me rewrite this. So, is this clear to everyone? So, let us call it equation 1 right and since function is l smooth, we know that f of y So, this comes from L smoothness and if I choose y, I set y to be x naught and x to be x star, what do I get? f of x minus x naught minus f star, what is the gradient of the function at x star? it is an unconstrained minimization. So, the gradient is 0. So, this term becomes 0 and we are left with L by 2 of right and that is how we can replace this term with this particular. So, this gives us So, this completes the proof ok. So, what is the rate order of convergence here? So, earlier we had order 1 over k kind of convergence right.



When we looked at the simple gradient descent, it had order 1 over k kind of convergence and the number of iterations was order 1 over epsilon iterations right. So, what is the order of convergence here? If I look at the complexity, so this is order this kind of complexity right. So, that is the complexity of this algorithm and as I said if I want if you want to derive the number of iterations what do you need to do? set this to epsilon right. So, let us say I equate this to be to be equal to epsilon. So, we basically get 1 minus mu over L k is equal to epsilon.

In order to derive the number of iterations basically I want to get a value of k in terms of epsilon right. So, in order to do that you just take the log on both sides. So, you get k log 1 minus mu over L is log epsilon log 1 minus x you can approximate this using minus x. So, this would be k times negative mu over L is log epsilon or implies that k equal to ok. So, that means if you want to be epsilon close to your solution the number of iterations.



number of iterations okay. So, L over mu log 1 over epsilon iterations okay. So, if you

want to be epsilon close, this is the number of iterations that you have to spend. So, in the in the previous case, we had one order 1 over epsilon, but now we have order log 1 over epsilon right. And that means, we need fewer iterations to be epsilon close, because log of x it I mean compared to x it grows slowly right. So, you require fewer iterations to be epsilon close to your solution.

So, now if I look at the algorithm and I have gradient descent here, then for convex f we had order 1 over epsilon for strongly convex setting. And by the way this term L over mu or mu over L that is something that you would see often in deriving rates of convergence even in problems you would see that this term plays a significant role right. So, this is for the gradient descent algorithm when we looked at the convex setting and the strongly convex setting and in both cases. So, in this case the rate of convergence was sub-linear, in this case the rate of convergence is linear and these are the number of iterations right. number of iterations needed.