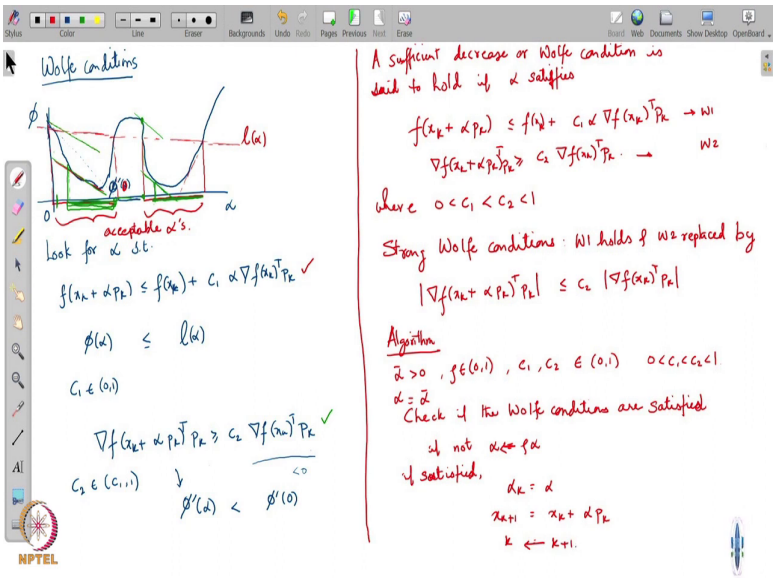
Optimization from Fundamentals
Prof. Ankur Kulkarni
Department of Systems and Control Engineering
Indian Institute of Technology, Bombay

Lecture - 19C
Line search algorithm and convergence

(Refer Slide Time: 00:16)



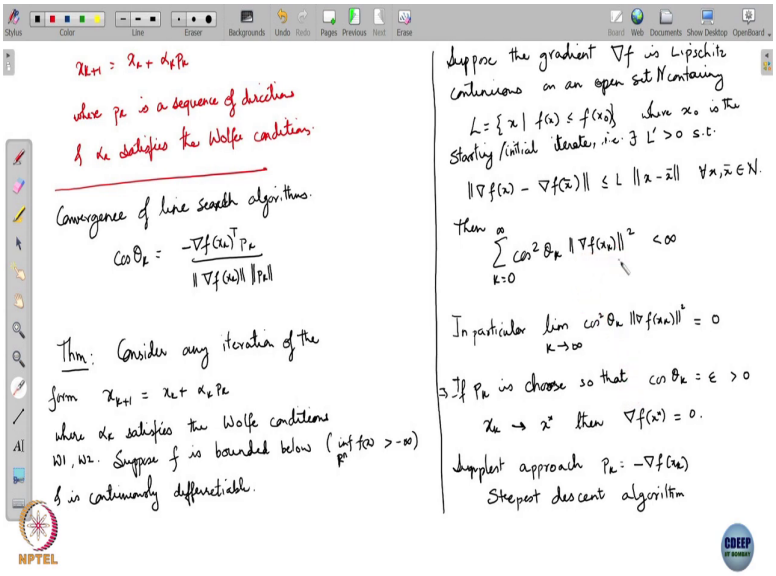
A practical line search type algorithm the algorithm stands to work like this. So, let me write this out. So, you start usually with some say an alpha bar greater than 0. You choose a rho between 0 and 1 and I will tell you why this rho is needed. You choose your C 1, you choose your, C 1 and C 2 and then what we what one does is you is that you search overall.

So, what one does is you try you we can say check if the Wolfe conditions are satisfied. So, at each iteration check if the Wolfe conditions are satisfied. Check if the Wolfe conditions are

satisfied. If not, you try an alpha. So, you in right initialize alpha equal to some alpha bar, if not, change alpha to rho times alpha.

So, what you are doing is you start with a large enough alpha and then you keep back tracking to see where you can to see what would be if your Wolfe conditions are satisfied right. You and you can keep repeating this until eventually the Wolfe conditions are satisfied. If satisfied you would define alpha K as alpha define alpha K as alpha and the next iterate x K plus one is defined as x K plus alpha P K, alright. And then and you take K as is we set K to be K plus 1.

(Refer Slide Time: 03:44)



So, what happens then is so, your algorithm then effectively then takes you through these iterations, where x K plus 1 equals x K plus alpha P K, where P K is a sequence of directions and alpha K satisfies the Wolfe conditions at each K. So, this is basically so, what we get therefore, is an is a sequence of iterate such that that are given by this recursion x K plus 1

equals x K plus alpha P K alpha K P K, where P K is a sequence of directions and alpha K is no satisfies the Wolfe conditions.

Now, what I have not told you yet is what how is this P K to be chosen. So, we have been we will been silent on this particular topic about how we how about what the choice of P K is. So, the P K is to be is can be chosen in a variety of ways and I will give you a general result here, but the main thing is that the P K should be a direction in which the function decreases.

So, the simplest thing you could do for this sort of purpose is to simply look for a d P K to be the negative of the gradient of the function. So, locally we are assured that whenever the in that if we take a step in that direction if we move infinitesimally in that particular direction the function is guaranteed to decrease.

So, P K can be taken to be the negative of the gradient of gradient of the function, but it that is not that it is that is not the only choice, one could do many other things to choose the choose the P K. In fact, variants of this give you gives rise to various different types of algorithms itself ok.

So, I will now also mention to you is a sort of a generic sweeping result which ensures which tells us what sort what sort of P K's to line search algorithms actually to line search actually a line search algorithm are actually converge ok. So, now, define. So, that brings us to this topic of convergence of line search algorithms.

So, define cos of theta K as basically the cosine between the negative of the gradient of the function and P K. So obviously, this kind of quantity is because so, its I am taking the inner product in the numerator negative of the inner product and dividing by the norms of these two vectors.

So obviously, this is well defined only when the when these two vectors are not 0. So, P K is obviously, a direction we are choosing. So, it is a non-zero direction and gradient of the

function is if it is should be non zero, then this quantity is well defined ok. So, the so, what is the; what is theta K here?

Theta K is capturing the angle between P K and the steepest possible descent direction you could pick. The direction that you could pick which is which gives you the steepest decrease steepest most decrease in the vicinity of the function, right.

So, this is the so, theta K is capturing that the angle between these two. So, the theorem which we will not prove, but I will just mention to you is this. So, consider any iteration of the form; consider any iteration of the form x K plus 1 equals x K plus alpha K P K.

Now, where alpha K satisfies the Wolfe conditions, W 1 comma W 2. If now suppose f is bounded below. Bounded below means, it has the infimum of f over R n is greater than minus infinity and is continuously differentiable. Suppose, the gradient grad f is Lipschitz continuous on an open set containing the set containing the set L.

Let us call this set. What is this set? This set is called the level set its the set of axis for which f x takes value less than equal to f of x 0, where f where x 0 is your starting iteration; starting iterate starting or initial literate. So, now what does it mean? So, the suppose the gradient f is Lipschitz continuous on an open set containing this level set, where x 0 is the starting iterate. What does this mean?

That is there exists an L L dash say greater than 0, such that gradient of f at x minus gradient of f at x bar open set let us call this open set N less than equal to L times norm of x minus x bar for all x x bar in this open set N ok. So, we have we can take we can take P takes P K to be any set of directions like and choose alpha K such that the Wolfe conditions are satisfied ok. And suppose the function is bounded and continuously differentiable and the gradient is Lipschitz in an in the set that we are considering.

So, what is this set L? L is the level set means that it is from starting from take all the x's that give you a better value than the one that you are starting with this the one you are starting with is x 0 all the x's that give you a better value a lower value than what you are starting

with that that entire region is called a level set that is your set L ok. And so, we want we are assuming that the gradient is Lipschitz on that set.

Then what does it say? Well then it says then cos square theta K times norm of gradient x K square this summation is finite. This whole sum is finite. Now, what does this mean? So, the claim is that if you can if you choose your iterations this way if your function has these properties and you choose your iteration this way in the way that is indicated and your alpha K satisfy the Wolfe condition, then this particular condition this has to be finite.

Now, this looks like a like a bizarre technical conclusion, but actually it it is says a lot in one in one sentence. So, since this summation this is an infinite sum right. So, if this infinite sum is finite what is that? If that infinite sum is finite then it means that the limiting value of limiting term here should be going to 0.

So, in particular in particular limit as K tends to infinity cos square theta K times norm square of this is equal to 0. So, it says that well your, the this limit goes to 0. Now, if you look at this limit what is this limit? It takes it has two terms here. It has the norm of the gradient of the function and it has the cos the square of the cosine of the angle between the negative gradient and the direction P K that you have chosen.

So, now this tells you something quite nice and powerful. It tells you that if I can choose my, if I choose my P K in such a way so, if P K, so, what does this mean? If P K is chosen is chosen so that say cos theta K is say always equal to some epsilon ok say minus epsilon ok, rather since you have already a minus sign there this is equal to epsilon say. So, if I choose my P K so that the cost theta K is always equal to some constant epsilon, which is positive.

Then in that case in this limit here; in this limit here this cos square theta would always be epsilon and it could it can jump out of the limit. And in that in the and then what we are left with is just is that the gradient of the limiting value of the gradient of the function should be 0, which means that x K converges to if x K converges to some x star then the limiting value of the gradient of x star should be equal to 0 right.

So, if P K is chosen in such a way that you that you are making an making a that P K makes an acute angle ok. If P K is chosen in such a way that P K makes an acute angle with the negative of the gradient right, so, it has some component. What does this means is it has some component along the negative; it has a, some non it has a non zero component along the negative gradient.

Then you are guaranteed that the sequence of iterates actually get you to a point where the gradient becomes equal to 0; that means, you are satisfying the necessary conditions of optimality. Now, this has this means that. So, this it is important here that this angle is this the cosine here is some epsilon equal which is positive and remains and remains positive and so, the way we have done this is by choosing an epsilon that is independent of K.

If epsilon also depends on K and starts decreasing to 0 then in that case the limit of this product being 0 does not let me conclude that the gradient is equal to 0. It could well be that you know this product the gradient still remains positive and yet you your, this limit is going to 0.

So, the so this condition is effectively telling you that the way to ensure that your gradient vanishes is by is by making sure that the, this cosine remains bounded away from 0. So, you; that means, you should continue to make if a non vanishing angle the that P K should continue to make a non vanishing angle with the negative gradient whatever that gradient may be ok. So, P K if it continues to make this non vanishing angle with the negative gradient then the gradient itself will go to 0, right.

So, the simplest way of ensuring that is the simplest way is to take approach to doing that is to take P K to be the negative gradient itself. So, then you are collinear with the negative gradient and in that case you would you would obviously, make your yeah. So, in that case this would actually go down to the angle would go the angle will the epsilon in that case will always will be 1 and then the gradient would be equal to 0. This kind of algorithm is what is called the steepest descent; steepest descent algorithm.

Now, steepest descent only ensures that you are taking the steepest the direction of the steepest descent at each point that may or may not be right for you in the long run. In the sense that the steepest what looks like the steepest descent at a particular point may not give you a sustainable decrease all the way down when you go further.

So, you have to take into account also how the directions of steepest descent themselves change. So, you so, the ideal way is to actually take into account also the curvature of the function and that those kind of that gives you a much richer class of algorithms that, but they are once again another class in this kind that I have mentioned.

So, they so, long as the, you when you are designing these sort of algorithms or when you are coming up with your iterate make. So, long as you make sure that your cosine of the angle is continues to be bounded away from 0 you should be fine and you your, you will be going to your iterates will take you to the grade to the to a point, where the gradient vanishes alright.

So, I will so, with that I will stop this lecture and we will continue next time.