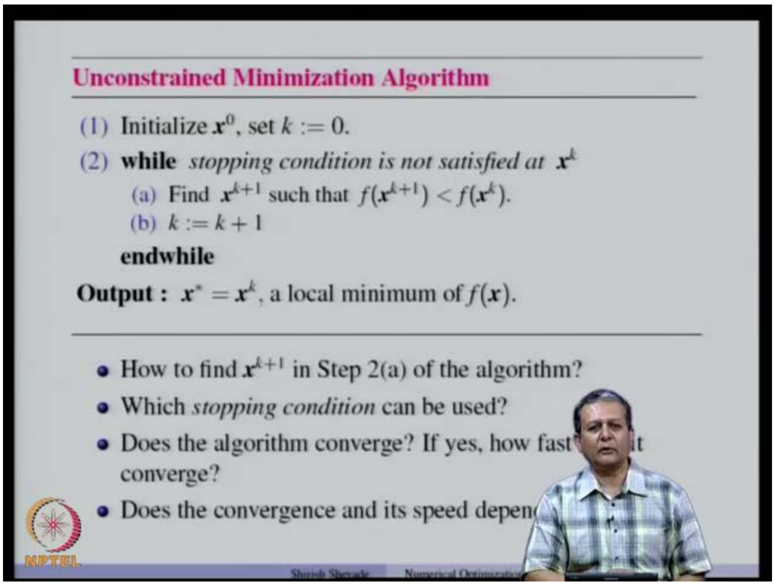
Numerical Optimization
Prof. Shirish K. Shevade
Department of Computer Science and Automation
Indian Institution of Science, Bangalore

Lecture - 11
Line Search Techniques

Series of lectures on Numerical Optimization, so in the last class we started discussing about unconstraint optimization and in particular, we looked at the necessary and sufficient conditions or the existence of a local minimum for an unconstraint optimization problem. So, we saw that the second order necessary conditions for a none constraint local minimum are that, the gradient at a particular point should vanish and the Hessian should be positive semi definite. So, these are necessary conditions and the second order sufficient conditions are that, if at a particular point x star. The gradient of the function vanishes and the Hessian at that x star is positive definite, then x star is a strict local minimum.

(Refer Slide Time: 01:16)



And then we started looking at the algorithm for a unconstrained minimization problem. So, this was a conceptual algorithm for solving a minimization problem where we are trying to minimize the function f of x. So, the first step of the algorithm is to initialize x 0, so initialize to some point in the space of real numbers and said the iteration count to 0. And then while some stopping condition is not satisfied at x k, one finds a point x k plus 1 such that the value of the function at the new point is less than the value of the function at the old point.

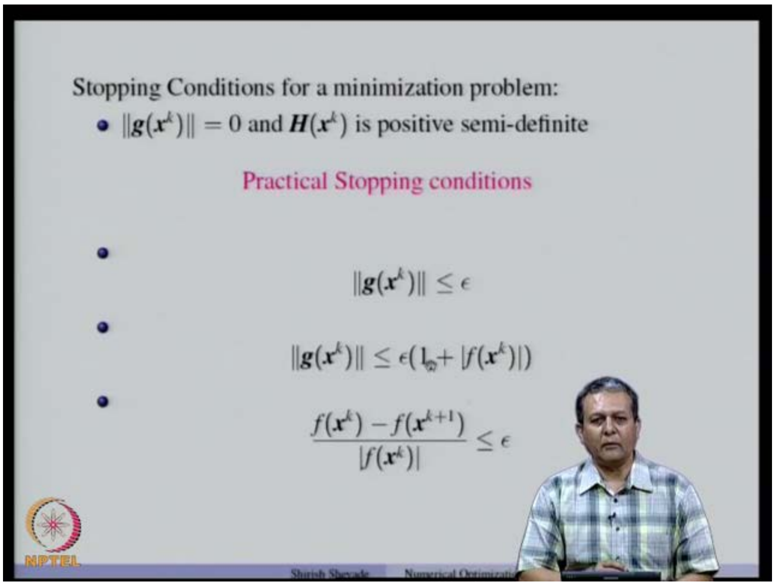
After having found the new point x k plus 1, we just increment the iteration counter and check whether the stopping condition is satisfied at that point new point, and the procedure is repeated till some stopping condition is satisfied at x k. And what we except to get is a point x star which is nothing but the x k at the end of the last iteration, and the that x star is expected to be a local minimum of f of x.

Now, there are different questions that we would like to answer related to this conceptual algorithm. Now, one of the first and the most important point is that how to find x k plus 1, so that the value of the function and that x k plus 1 is less than, the value of the function and the current point x k. So, this is a very important question and many optimization methods use different strategies to find out this x k plus 1.

Now, the next point is that what is the stopping condition for a given algorithm, again there exist different stopping conditions we will see them in today’s class. And those stopping conditions are typically derived from the necessary and sufficient conditions that we studied in the last class. Now, another important question that needs to be answered is that does the algorithm converge, under what conditions will it converge, and how fast will it converge, if at all it converges.

So, this speed of the algorithm in terms of the number of iterations or the input dimension is also sometimes play a important role in deciding the speed of the convergence. Now, one of the parameters that we used here is the initialization to some point x naught, now though does the convergence and the speed of convergence depend on x naught the initial point. So, we will answer some of these questions in today’s class and then the remaining questions in the next few classes.

(Refer Slide Time: 04:23)



Now, as we saw in the last class, second order necessary conditions for the existence of a local minimum are that the now of the gradient at x k should be 0, and the Hessian at x k should be positive semi-definite. Now, many algorithms that we are going to study do not use the second order information, so it is many times difficult or computationally expensive to compute H of x k. So, for all practical purposes we will not use this condition, but remember that if some extra information is about the Hessian is available one can use that.

So, we will mainly concentrating on this first condition, which is so the gradient vector vanishes at a local minimum, essentially means that the norm of the gradient vector should be 0. Now, when we talk about the numerical implementations of the some of the algorithms, it is very difficult to ensure that the norm of the gradient vector reaches exactly 0, because we always work with the final precision arithmetic. So, instead of checking this condition exactly, it may be a good idea to make some make use of some approximate condition.

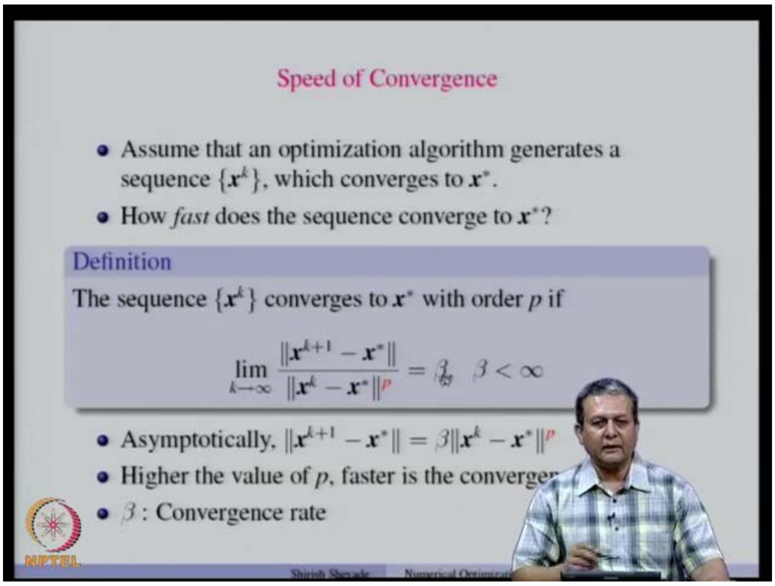
So, we will study some practical ways of determining this stopping conditions, now one of the first condition that one can think of is that norm of g of x k is less than or equal to epsilon. Where, epsilon is a small positive quantity that is defined by the user. So, in many algorithms we will see this kind of condition being used. Of course, the convergence will depend on what convergence point that one gets depend upon depends upon what epsilon 1 uses. So, it is a function of epsilon, so epsilon has to be carefully chosen, because that is a use a different parameter.

Now, many times what happens is that suppose for a optimization problem, we want to find out we want to minimize the distance between the points. Now, if the distance suppose is the distance is specified in kilometers, now later on it is found that the distances have to be converted to say millimeters. Then the value of the gradient would change considerably and therefore, this condition does not take care of the scaling of the variables.

So, many a times another condition that is used is norm of the gradient at x k should be less than or equal to epsilon into 1 plus absolute value of f of x k. So, these makes use of the function value absolute function value, so that takes care of the scaling point. Now, in many cases it is propose to use the relate decrease in the function value as the stopping criteria. So, this is the f of x k is the current function value and f of x k plus 1 is the new function value. So, f of x k minus f of x k plus 1 divided by the absolute value of f of x k, that should be less than or equal to epsilon.

So, remember that these all epsilons need not be same, they could be different on the epsilons used in these expressions, but the important point here is that this criteria is independent of the scaling of the variables. So, one could use this criteria to stop the algorithm of course, in this case this epsilon is a is a different parameter. And in this course we will typically use this criteria, but we do not have to restrict ourselves to this criteria one can use any of this criteria depending upon the requirement.

(Refer Slide Time: 08:55)



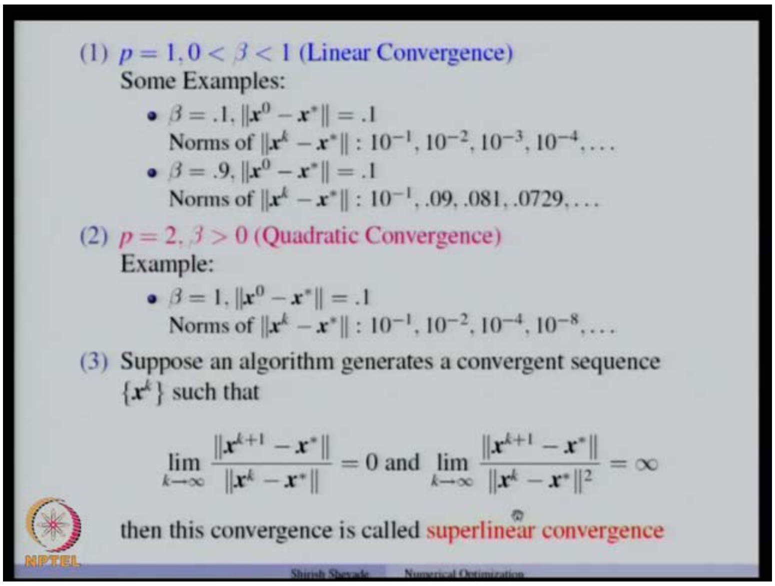
Now, another important point that we have to worry about is the speed of convergence of a optimization algorithm. Now, let us assume that a we have an iterative optimization algorithm, which generate the sequence x k, and that sequence convergence to some x star. Now, we are interested in finding out how fast does the algorithm converge to x star.

Now, we for that purpose, we need to see this definition, so the sequence x k which converges to x star is said to converge to x star with order of p. If you take the ratio of norm of x k plus 1 minus x star, and norm of x k minus x star to the power p. And if that is equal to beta as k tends to infinity where beta is a finite positive construct. So, typically this beta is finite because all this quantities on the left side, typically beta is positive all the quantities on the left side you will see that they are positive quantities and beta is finite.

So, this p is call the order of convergence and beta is call the convergence rate. So, if we have sequence x k which convergence to x star, and then this relationship holds then we can find out the order of convergence and the convergence rate. Now, so you will see that x k plus 1 minus x star norm of this quantity is the distance between x k plus 1 and x star.

So, let us assume that we use the l 2 now, and x k minus x star is the distance between x k and x star. So, what we are interested in finding out is that how close does x k plus 1 go to x star compare to x k. And the order of convergence is the factor parameter p here. Now, asymptotically you can see that the distance between x k plus 1 minus x star is beta times the distance between the x k and x star to the power p, so this happens k increases. Now, the important point to be noted is that, if the value of p is higher then the convergence is faster. So, you can see that if p is higher then the distance between x k plus 1 and x star reduces to 0 at a faster rate. So, that is why higher the value of p faster is the convergence, now beta as I said that earlier that is called to convergence rate.

(Refer Slide Time: 11:58)



Now, let us see some examples, so let us considered the case where p is equal to 1 and beta is in the open interval 0 to 1. Now, this is called linier convergence, so p has to be 1 and beta has to be the open interval 0 to 1, so let us take some example. So, let us the assume that beta is point 1 and the distance between the initial point and x star is also point 1. And let us see how the sequence is generated, so let us observe the norms of x k minus x star as k going goes from 0 to infinity.

Now, you will see that initially the x 0 minus x star, the norm of that is point 1, so we have that initial value. Then that because p is 1, so this quantity gets multiplied by beta and point 1 in to point 1 is 10 to the power minus 2. So, the norm of x 1 minus x star is 10 to the power of minus 2 and again the whole quantity is multiplied by beta. And the norm of x 2 minus x star becomes 10 to the power of minus 3 and so on.

So, you will see that in every iteration the x k minus x star, the norm of x k minus x star gets reduced by 10, so here it was one-tenth 1 by 100 1 by 100 1 by 10.000 and so on. Now, to show that this linier convergence depends a lot on this convergence rate beta, we will take a similar example, so where we will take a value of beta to be 0.9. So, which is more close to 1 other than close to 0, in the earlier case the value of beta was close to 0, now the value of beta is close to 1.

So, let us keep the same initial conditions, that norm of x 0 minus x star is 0.1, and then let us observe the norms of x k minus x star. So, as is the previous case the first x 0 minus x star norm of that quantity is 1 by 10, now as now that will be multiplied by 0.9 in for the norm of x 1 minus x star. So, what we get is 0.09, and then again this will be multiplied by 0.9, remember that we are working with the case p equal to 1.

So, what we get is point 0 at 1 and 0.0729, so now, let us compare the two norms of x k minus x star that we obtain with different values of beta. So, initially they were same, but now after one iteration it was 1 by 100, while here it was 9 by 100, after two iterations it was 1 by 1000, this was 81 by 1000 and then 1 by 10,729 by 10,000. So, you would realize that if beta is close to 0, then this sequence the x norm of x k minus x star goes to 0 at a faster rate compare to the case where beta is to close to 1.

So, typically when we talk about linier convergence we would prefer beta to be more close to 0, rather than close to 1. And if beta equal to 1 then you will see that around be a change in this particular example case the first example. Now, let us look at p equal to 2 and beta is greater than 0, now when p equal to 2 and beta is greater than 0 that is called the quadratic convergence. So, let us take again some example, so let us take beta to be 1 and norm of x 0 minus x star to be 0.1, and let us observe the sequence of norms of x k minus x star.

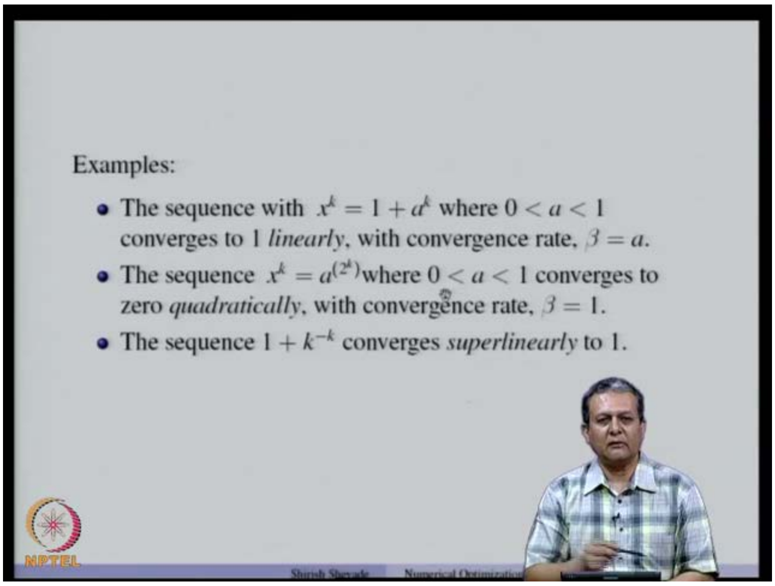
Now, if we look at this, so initially it is 0.1 into point initially it is 10 to the power minus 1 and then it moves at a rate which is shown here. So, it is now moves that 10 to the power minus, so the norm of x 1 minus x star is 10 to the power minus 2. Then remember that we have p 2, so norm of x k minus x star gets squared in determining norm x k plus 1 minus x star. Let show, so in the fourth in the 3 iterations the norm of x k minus x star has reduced to 10 to the power minus 8.

Now, we can compare the three cases which are given here, and clearly this is the faster rate of convergence compare to this and this is faster compare to this. So, clearly this method the quadratic convergence gives a faster convergence provided beta is greater than 0, now the important point that you have to note here is that this quantity because beta is any quantity greater than 0.

So, beta can be less than 1 or greater than 1, so typically this quantity should be less than 1 so; that means, that your x 0 should be reasonably close to x star, so that one can achieve faster convergence. Now, there exist some algorithms where if we take the ratio of norm x k plus 1 minus x star and x k minus x star. So, as k tends to infinity this ratio goes to 0 and norm of x k plus 1 minus x star divided by norm of x k minus x star square that ratio goes to infinity, then this convergences is called super linier convergence.

So, this convergence is in between the linier and quadratic, so many optimization algorithms that we are going to study how this super linier convergence. So, that means, they are greater than the algorithms linier convergence, but not as good as the algorithms which have quadratic convergence. So, as we will see later that the quadratic convergence the algorithms which follow quadratic convergence would requires lot of extra information, and that is difficult to maintain. So, that is all practically it will be preferable to use algorithms which are somewhat between linier and which have convergence between linier and quadratic, and those are the algorithms which haves super linier convergence.

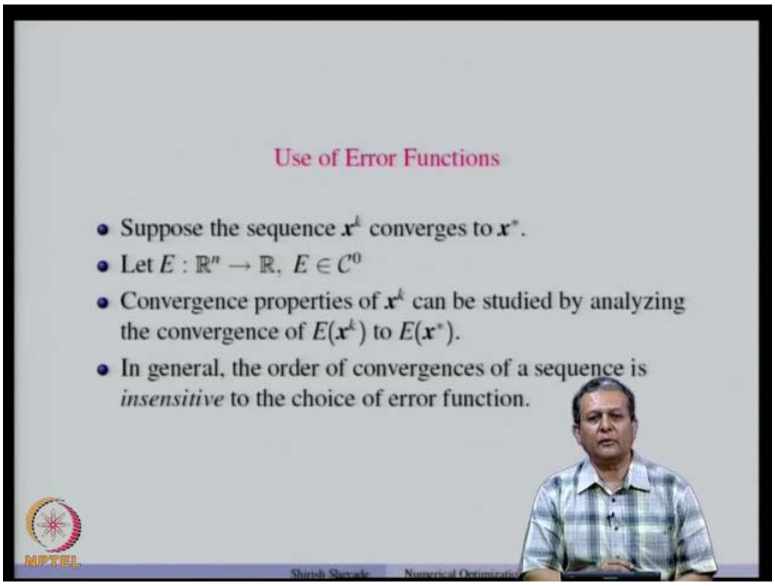
(Refer Slide Time: 19:21)



Now, let us see some examples let us consider the sequence with x k is equal to 1 plus a to the power k, where a is a positive fraction. Now, since a is a positive fraction as k tends to infinity, x k tends to 1. Now, one can verify that this sequence converges linearly with the convergence rate as beta is equal to a and this is a fraction, so this is a linier convergence with 0 less than beta less than 1.

On the other hand if you take the sequence x k equal to a to the power 2 to the power k, where a is a positive fraction. Now, this converges to 0 limit as x k tends infinity x k goes to 0, quadraticlly with convergence rate beta to be 1, and here is a example of a super linier convergence. So, one can verify that the sequence 1 plus k to the power minus k it converges super linearly to 1. So, these are some examples of sequence sequences which are generated by different algorithms, where in the first case the convergence was linier, in the second case the convergence was quadratic and in the third case the convergence of super linier.

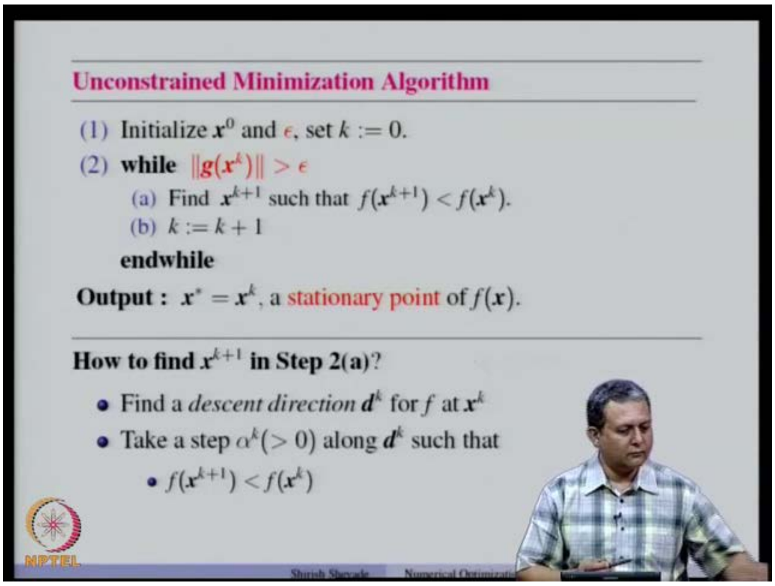
(Refer Slide Time: 20:50)



Now, when we talked about the convergence, we use the norm of x k plus 1 minus x star and norm of x k minus x star. So, many a times it is good idea to use error functions, solet us see what are those error functions. So, suppose we have convergence sequence x k which converges to x star, so let a b is some function real valued functions which is continuous. So, instead of observing the behavior of x k to x star many times it may be a good idea to observe the behavior of E x k to E x star. So, convergence properties of x k can also be studied by analyzing the convergence of E x k to E x star.

Now, one would wonder that if one uses different types of error functions will the convergence behavior be different. So, in general the order of convergence a of sequence is insensitive to the choice of error functions. So, this is not a this statement does not hold always, but in general it holds. So, let us not worry about the exceptions in this movement and take this result in a general form, where the order of convergence of sequence does not depend on the choice of the error function.

(Refer Slide Time: 22:25)



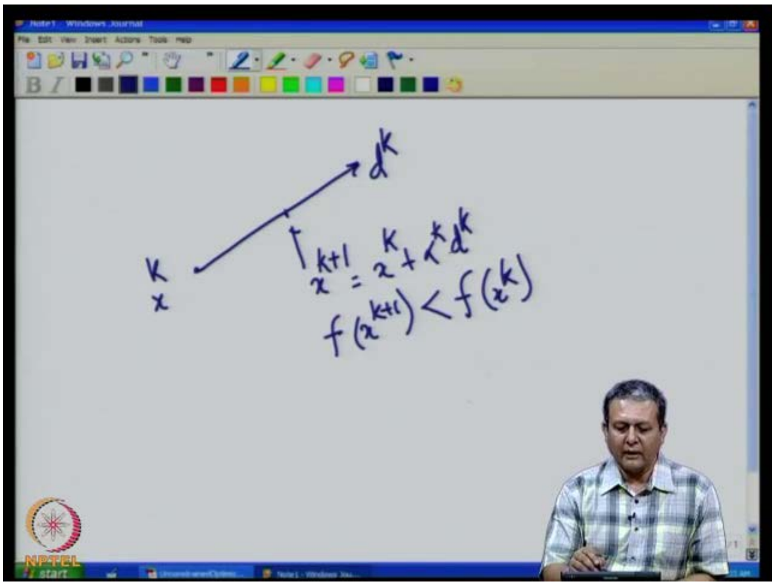
Now, with this background, now let us revisit our conceptual optimization algorithm, now remember that we are not going to use the second order conditions. As I said earlier and also we want to use some practical stopping condition, and one of the practical stopping condition is that the gradient of the function should be norm of the gradient of the function should be less than or equal to epsilon. So, till that condition is satisfied, so , so while norm of the gradient of the function at x k is greater than epsilon the algorithm gets executed.

Now, epsilon is a new parameter that we have introduced, so that is are typically user defined parameters, so we have to initialize that value of epsilon. And then finally, we will end of in a situation where norm of g of x k is less than or equal to epsilon. Now, these does not guaranty that x k is a local minimum, so output as a output of this algorithm we get x star to be a stationary point of f of x. And then one needs to check whether it is indeed a local minimum or not.

Now so we have answered questions related to the stopping condition and the rate of convergence of the algorithm, now we will worry about the step in the step 2 a of this algorithm. So, the step 2 a says that find the h k plus 1 such that f of x k plus 1 there is an f of x k; that means, that find a new point x k plus 1, where the function value at that new point is less than the function value at the current point x k.

Now, how does one find this? Now, there are it involves 2 steps, so the first thing the first step is that one finds a descent direction d k for the function f at x k, and take a positive step of length alpha k along the direction such that f of x k plus 1 is less than f of x k.

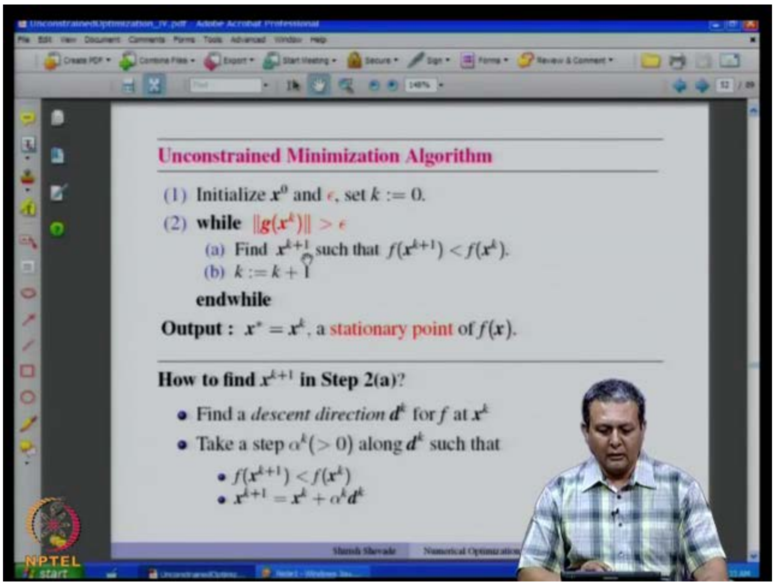
(Refer Slide Time: 25:00)



So, suppose this is a current point x k, and we suppose we find that the direction d k which is a descent direction at d k. Then we take a step of length alpha k along this direction, so this is the new point that we get and that point is x k plus 1 and that is nothing but x k plus alpha k d k. So, recall the definition of a descent direction that the direction d k is said to be descent direction for f at x k, if there exist some delta greater than 0.

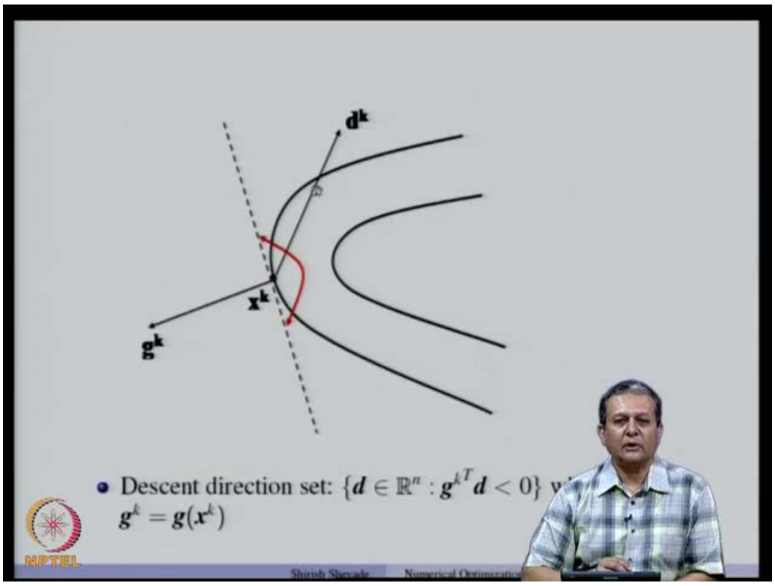
So, that f of x k plus alpha d k is less than f of x k for all alpha in the range 0 to delta. So, since d k is a descent direction we are able to find some alpha k along the direction such that f of x k plus 1 is less than f of x k and where x k plus 1 is nothing but x k plus alpha k d k.

(Refer Slide Time: 26:18)



So, these are the two steps that unnecessary for this finding out x k plus 1.

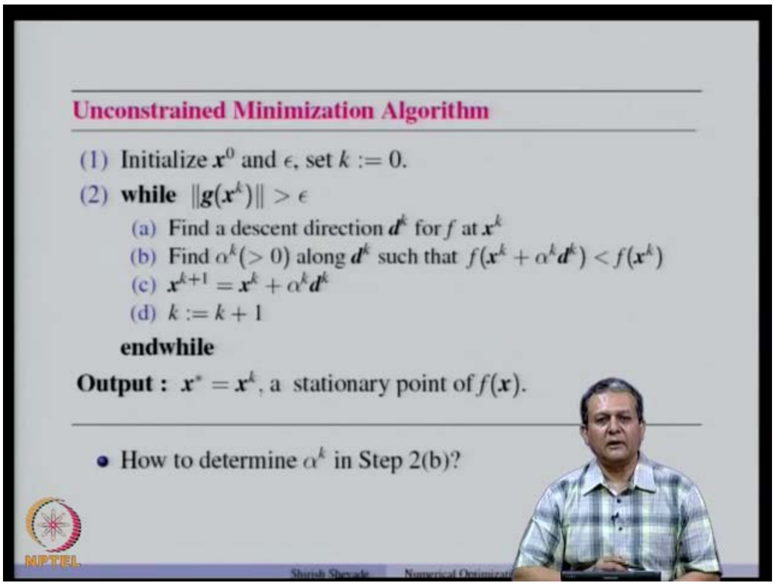
(Refer Slide Time: 26:27)



So, now let us look at the let us look at the set of descent directions, now these are the contours of the given function, this is the current point x k. Now, at x k this is the gradient vector g k, so in this direction the function value increases, now d k is a descent direction. Now, what is the characteristic of descent direction, so we saw in the last class, that if g k transpose d k is less than 0 then d k is a descent direction for f at x k. So, which means that d k should be in the open cone which is shown by a red arc here, so d k should lie in this open cone and that will make an obtuse angle with g k.

So, the descent direction said is a set of all directions in R n such that g k transpose d is less than 0, where we are using the notation g k is nothing but g k is written as g of x k. So, set of all directions d such that g k transpose d is less than 0 is shown here by a red arc, and d k can be any of this descent directions. Now, if you see this descent direction, now if you take a step along this descent direction, now if you come here at a point then the value the function would have increased. So, we are not interested in that, so we are interested in the step where the value of the function decreases, so that is a very important step.

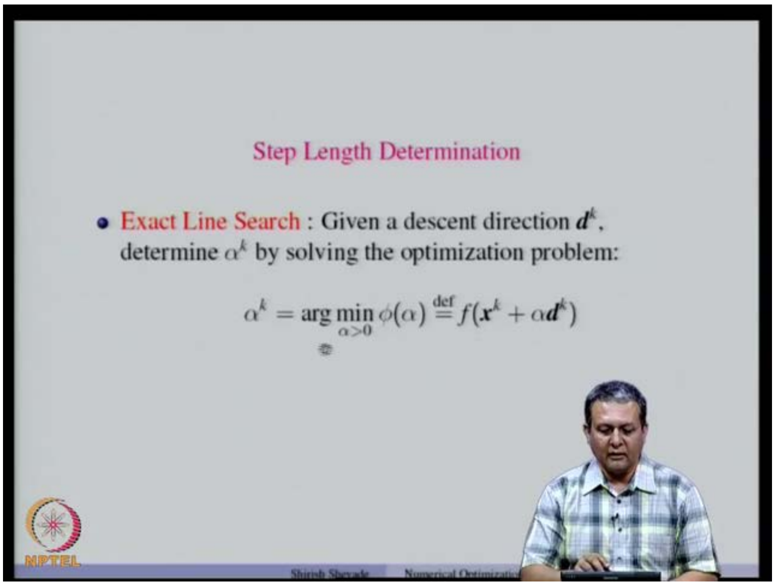
(Refer Slide Time: 28:34)



So, now we have revised our algorithm, so earlier we had given that find x k plus 1 such that f of x k plus 1 is less than f of x k. now, instead the first step will be to find the descent direction d k for f at x k, and this descent direction is such that g k transpose d k is less than 0. After, having found a descent direction he next step to get alpha k which is greater than 0 along the direction d k such that f of x k plus alpha k d k is less than f of x k.

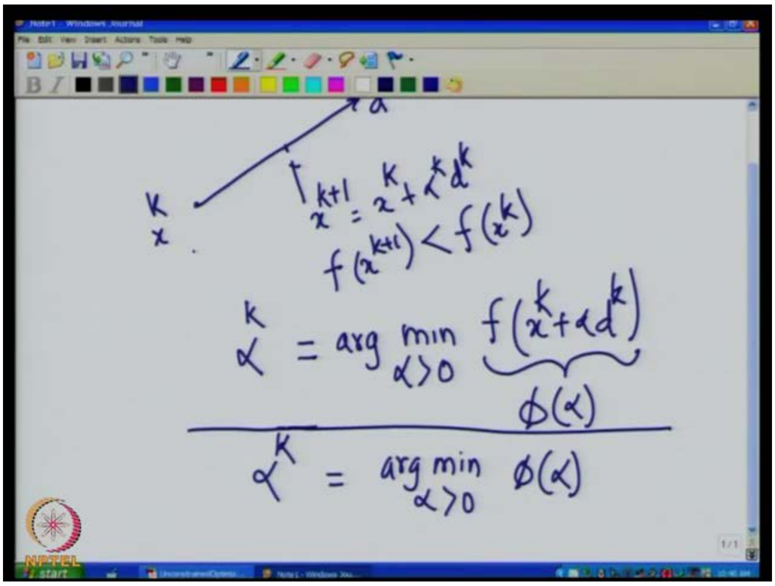
And this point is nothing but the new point x k plus 1, so the new point is set appropriately, the iteration counter is increase by 1. And the procedure is repeated till the norm of the gradient vector is less than or equal to epsilon. Now we know that the descent direction can be obtained by finding out the direction d, such that g k transpose d is less than 0. Now, how to find out alpha k in step 2 b of this algorithm and we will discuss that now.

(Refer Slide Time: 29:57)



Now, there are two ways to determine this step length one way is called exact line search and the other one is called inexact line search. So, in exact line search given a descent direction d k one determines alpha k by solving another optimization problem. So, remember that we are interested in minimizing f of x k plus alpha d k where alpha is greater than 0.

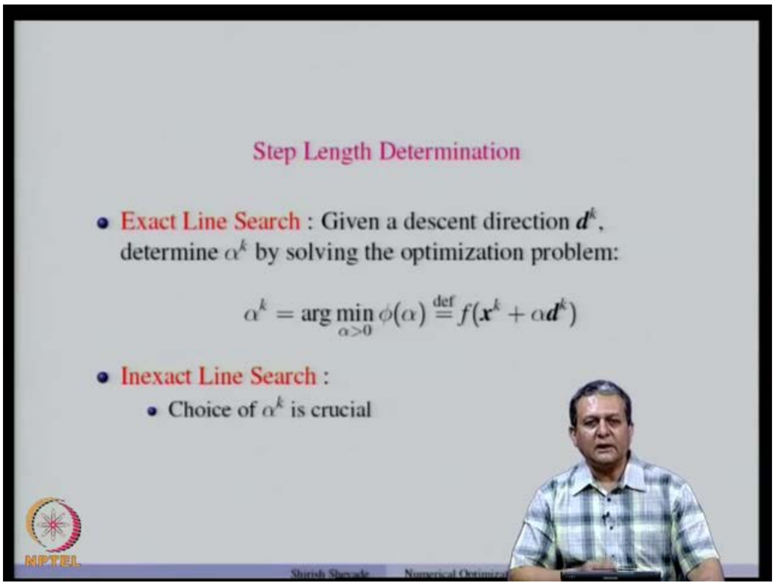
(Refer Slide Time: 30:33)



So if we are at a current x k this is the descent direction d k, we are interested in finding out the minimum of f of x k plus alpha d k where alpha is greater than 0. Now, along this direction we this is d k is a descent direction, so we can always find some alpha, such that this quantity is minimized. Now, if we take the arg min of that then what we get is alpha k plus 1. So, alpha k plus 1 is nothing but arg min alpha greater than 0 f of x k plus alpha d k. So, now, let us define this quantity as a function of alpha, so remember now that x k is fixed d k is fixed and alpha is a variable.

So, we can simply write alpha k plus 1 is equal to arg min alpha greater than 0 (()) by defining phi alpha to be f of x k plus alpha d k. So, alpha is a scalar because it is a step length, so this is a one dimensional optimization problem. So, whenever we are even x k and d k, we have to solve a one dimensional optimization problem to get alpha k, I am sorry which is alpha k. So, we need to solve a one dimensional optimization problem in alpha to get alpha k, and then once we get alpha k we plug in that value in this expression to get the new point a x k plus 1. And clearly since we are minimizing this f of x k plus 1 will be less than f of x k plus 0 alpha d k which is nothing but f of x k. So, at every iteration one needs to solve a one dimensional optimization problem in alpha to get alpha k and that is computationally expensive.

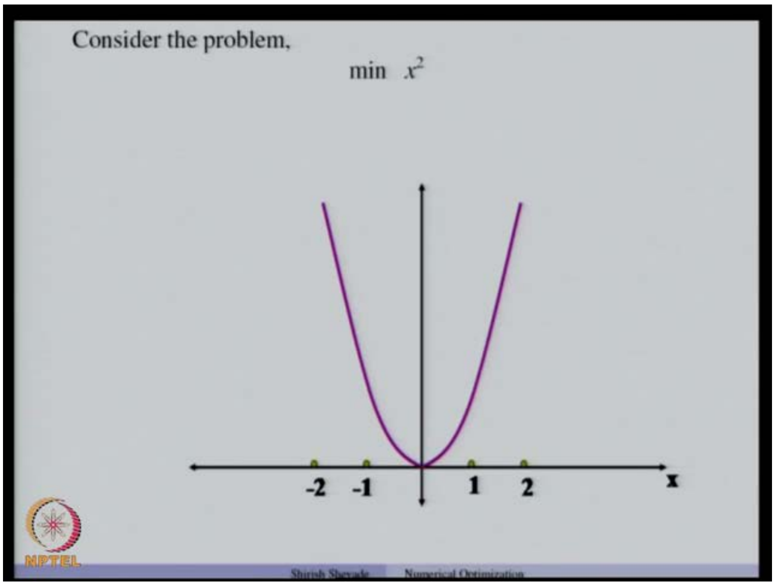
(Refer Slide Time: 33:24)



So, although this approach of doing one dimensional optimization problem would give us the exact alpha k, it is a computational expensive approach and remember that this has to be executed at every x k given d k. So, many times this procedure needs to be adopted or the optimization problem needs to be solved completely to get the solution alpha. And in most of the cases we are not really interest in solving this problem exactly, but our main problem is to solve the minimization of f of x.

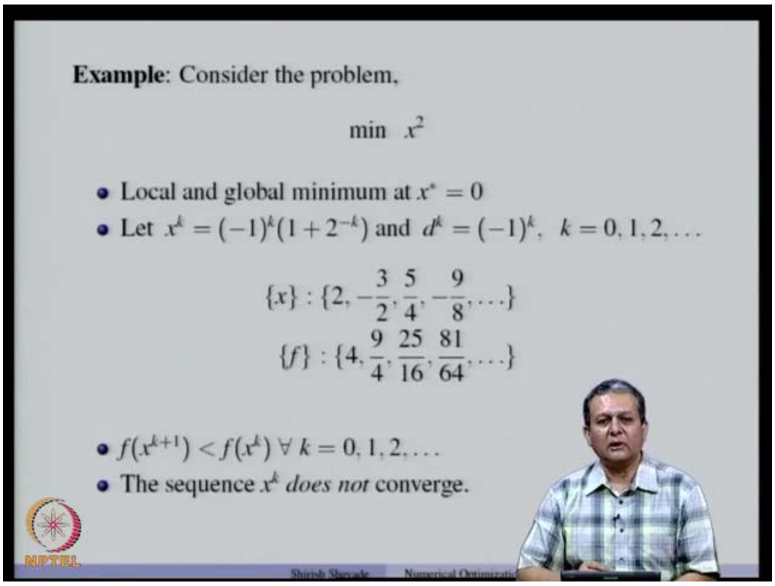
So, it may not be a good idea to do this exact line search in most of the practical algorithm. So, therefore, there are different ways to do inexact line search, now while doing inexact line search we find some alpha k, where the function value at the new point is less than the function value at the current point. But, the choice of alpha k turns out to be very crucial, so here since we are solving the direct optimization problem, we do not have to worry about the choice of alpha k, but once we move to the inexact line search. The choice of alpha k becomes very crucial, and we will see that using some examples.

(Refer Slide Time: 34:48)



So, let us consider the problem to minimize x square, this is just of one dimensional problem that we are considering, and the function would look something like this. And we know that this is going to be the minimum and this is also a convex function, so this minimum x equal to 0 is also a global minimum of this function. Now, suppose that we use some iterative optimization algorithm, which will generate the sequence x k using some mechanism.

(Refer Slide Time: 35:31)



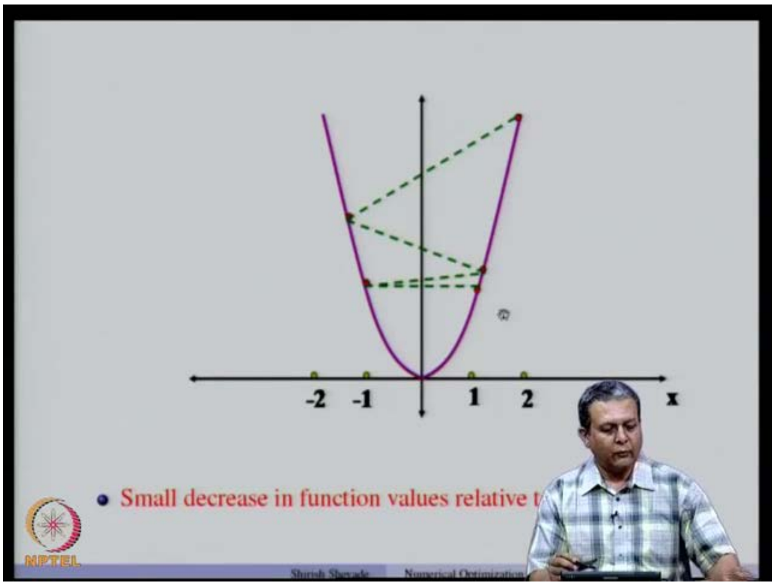
Now, for this problem we know that the local and global minima both are at the same point x star equal to 0. Let us assume that optimization algorithm generates a sequence x k to be minus 1 to the power k into 1 plus 2 to the power minus k and d k to be minus one to the power k. So, remember that this is a one dimensional optimization problem, so d k will take the value either plus 1 or minus 1, so one either increments or decrements. One either moves on the right direction or on the left direction, so suppose this is the sequence which is generated by an optimization algorithm.

Now, let us see how the sequence will look like, so if you look at the sequence x, so when k equal to 0 this quantity is 1 and this quantity is 2. So, the at k equal to 0, x 0 is 1, k equal to 1, this quantity is minus 1, and this is 3 by 2, so x equal to, so x 1 equal to minus 3 by 2, then x 2 equal to 5 by 4 and x 3 equal to minus 9 by 8 and so on. Now, let us look at the corresponding function values at those x case, so f of x 0 is 4 we use the same function x square, so f of x 1 is 9 by 4, then f of x 2 is 25 by 16 and so on.

So, now, you will see that this sequence has generated this algorithm has generated a sequence x k, such that f of x k plus 1 is less than f of x k. So, you will see that 4 is greater than 9 by 4 9 by 4 is greater than 25 by 16 and so on. So, every time when the algorithm moves to a new point in the sequence, the function value decreases. So, this is what we were looking for that we wanted an algorithm which will which will reduce the function value in every iteration.

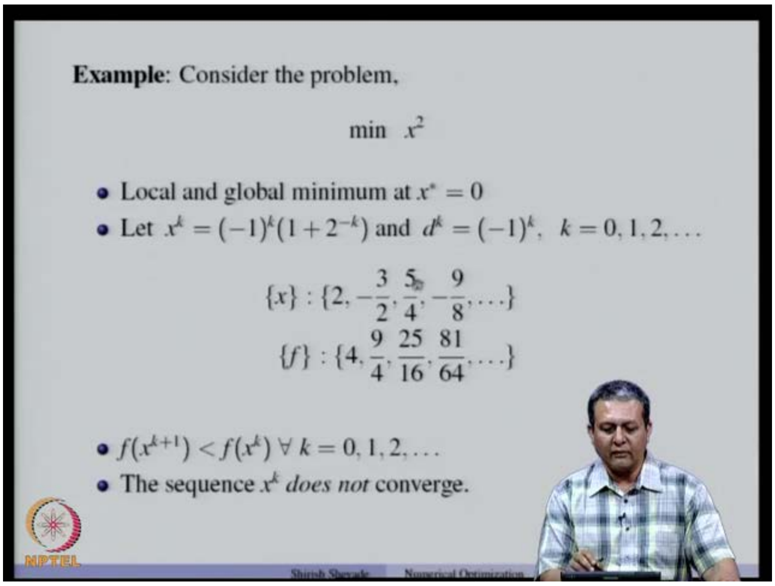
Now, you will see that the sequence, so if you just look at this sequence limit as k tends to infinity, the second term goes to 1 and the first term keeps oscillating between minus 1 and plus 1. So, one can realize that this algorithm although, it decreases the function value in every iteration that does not converge, it in fact oscillates between plus 1 and minus 1. So, we really cannot think of convergence to a local minimum, which is 0 algorithm oscillates between plus 1 and minus 1.

(Refer Slide Time: 38:37)



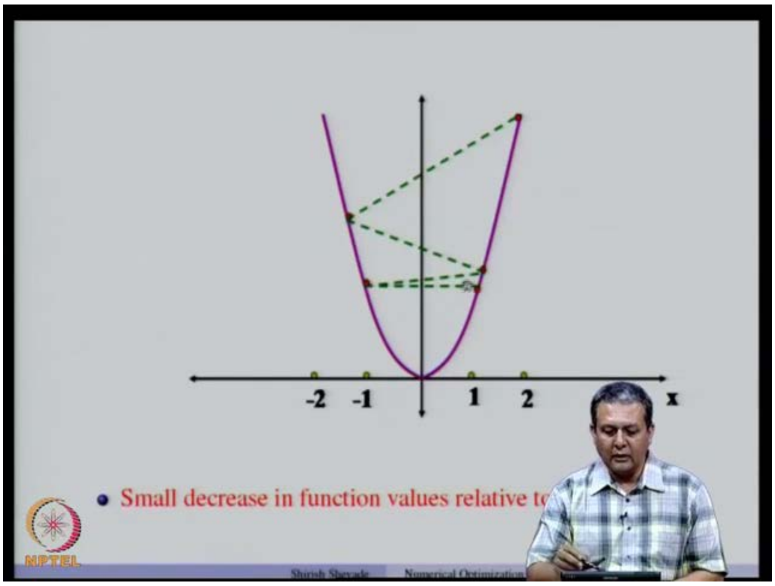
Now, let us study the behavior of that sequence, so we started with x equal to 2.

(Refer Slide Time: 38:46)



And then if you look at the started with x equal to 2, then move to minus 3 by 2 then 5 by 4 minus 9 by 8 and so on.

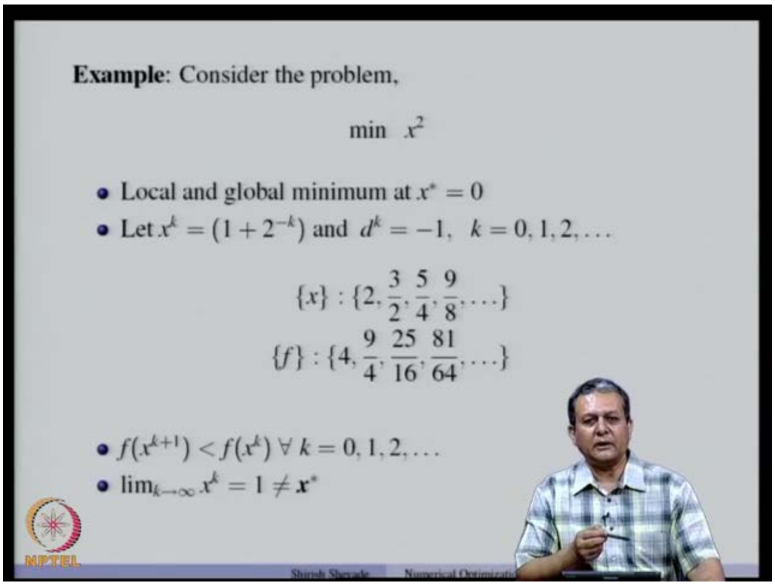
(Refer Slide Time: 38:54)



So, started with x equal to 2 move to minus 5 minus 3 by 2 and then so on and so forth. Now, what is the problem here, the problem here is that the decreasing function values is very small compare to the step length. Especially, if you see here that the step length that one takes is of the size close to 2, but the decrease in the function value is much, much smaller, and k goes to infinity the algorithm starts oscillating between plus 1 and minus 1. So, there is a very small decrease in the function value compare to the step length, and step length was quit large.

So, this is the drawback with this method, and one way to fix this is to make sure that there is a sufficient decrease in the objective function value compare to the decrease at the at the case where alpha equal to 0 compare to the initial decrease in the function value.

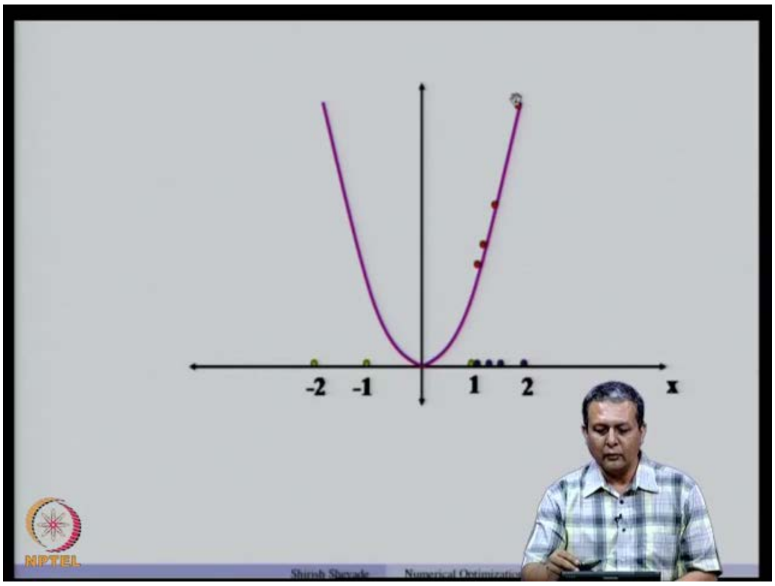
(Refer Slide Time: 40:18)



Now, let us see another example again let us consider the same problem to minimize x square, and it has local and global minimum at x star is equal to 0. So, let us assume that this stand the algorithm generates a sequence x k to be 1 plus 2 to the power 1 minus k and d k is minus 1 always and k goes from 0 to infinity. So, the sequence generated by the algorithm is 2 3 by 2 5 by 4 9 by 8 and so on, and the corresponding the function values are 4 9 by 4 25 by 16 and so on.

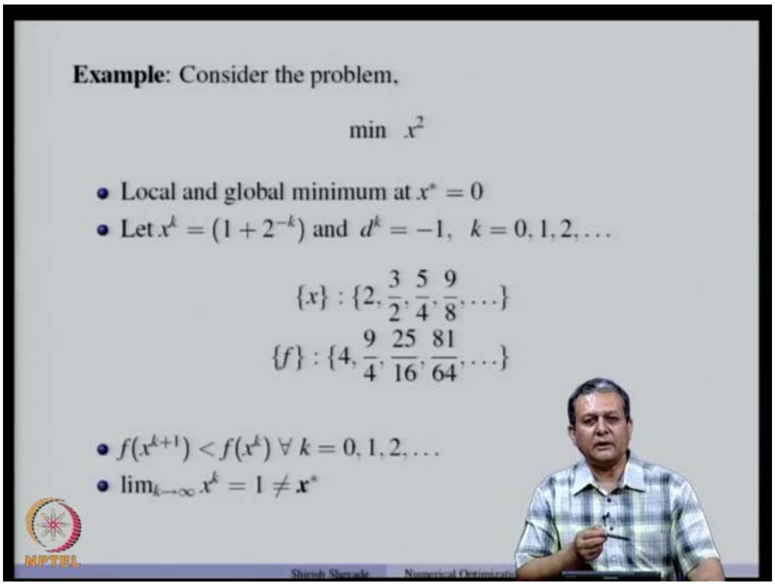
Now, you will see that again the function value decreases in every iteration. So, that condition is ensured f of x k plus 1 less than f of x k, but then what happens to the algorithm does it convert to the local minimum. Now, as k tends to infinity x k converges to 1. So, unlike the previous case we have the algorithm we gave a sequence which osculated between plus and minus 1, here we have a sequence which converges, but now if you look at the point where the sequence converges that is not same as the local minimum. So, here is the case of an algorithm which does convert, but it does not converge to a local minimum, now what is the problem in this case.

(Refer Slide Time: 41:53)



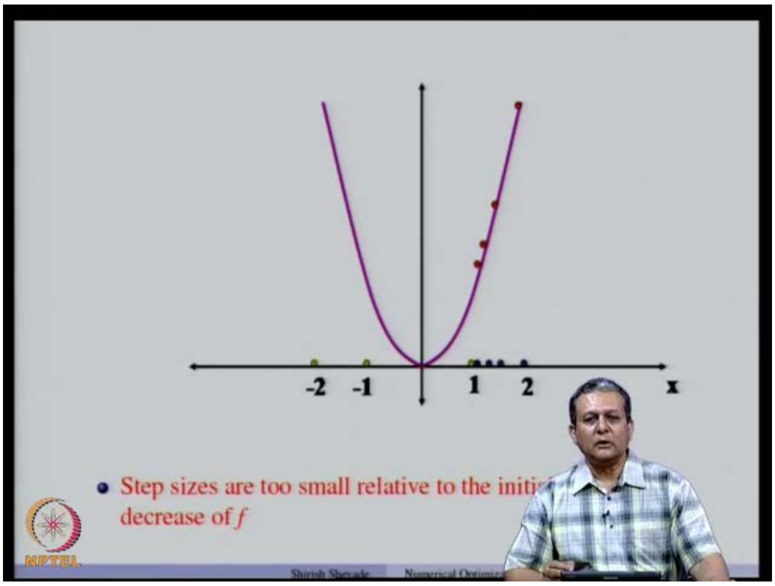
So, let us try to answer that, so initially we started with x k x 0 equal to 2 which is this point.

(Refer Slide Time: 42:09)



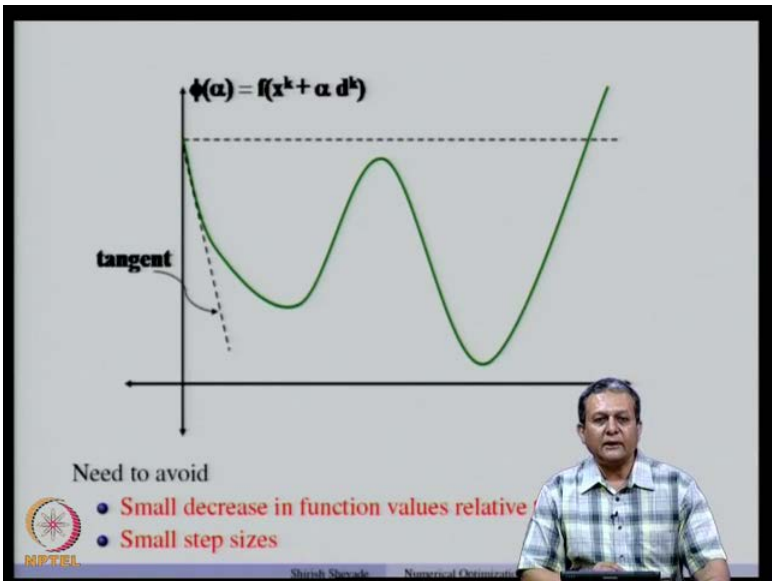
And then x 1 was 3 by 2 and then 5 by 4 and 9 by 8.

(Refer Slide Time: 42:13)



So, 3 by 4 5 by 4 9 by 8, so you will see that initially the decrease in the function value was quit large, but then the step lengths, where running out to be very small. So, step sizes were too small relative to the initial rate of decrease of the function.

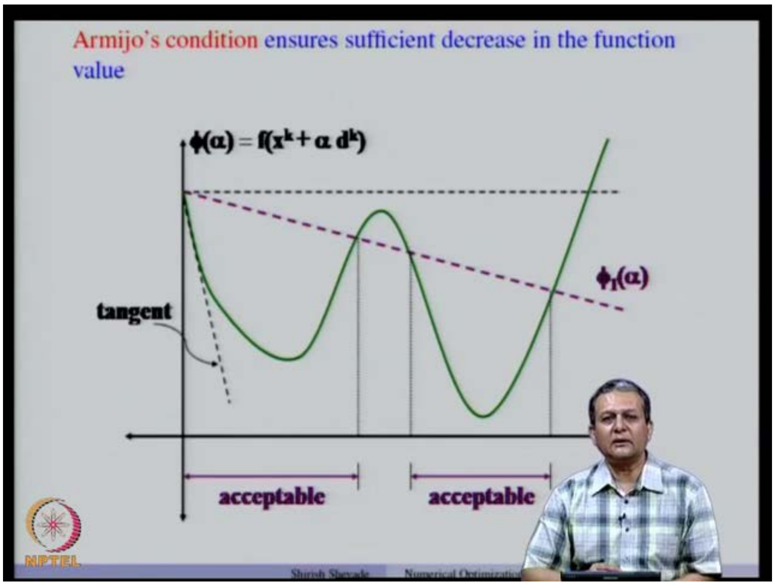
(Refer Slide Time: 42:43)



Now, to tackle these 2 problems we need to use some extra conditions and this conditions were developed some time back by Armijo and worst in Wolfe also gave some of the conditions to ensure that there is decrease in the sufficient decrease in the objective function, as well as the step sizes are not too small. So, we are in this figure you will see that on the x axis we have alpha and y axis is the function phi alpha which is nothing but f of x k plus alpha d k.

So, this is that function phi alpha which is floated here for some problem, now this is the tangent to this curve at alpha equal to 0. Now, if you do exact line search we might end up in one of this local minima, ideally we would like to find global minimum which is here, but we have seen that the exact line searches are computationally expensive. So, we are not interest in finding out the exact minimum of this function with respect to alpha, but instead we want to find out some alpha. Such that, we avoid the small decrease in the function value related to the step length and the small step sizes which were the problems in the last 2 examples that we saw.

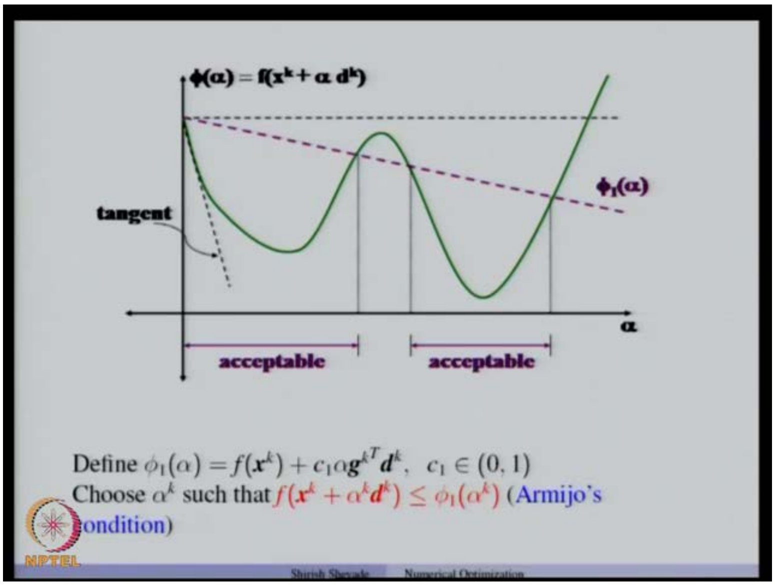
(Refer Slide Time: 44:35)



So, let us look at different ways to avoid this, now one of the conditions which was propose sometime back by Armijo, that condition ensures that there is a sufficient decrease in the function value. So, let us see with respect to the same function phi alpha what goes on, now as I mentioned earlier to ensure that there is a sufficient decrease in the function value we have to make sure that. The alpha should be chosen such that the decrease at the new alpha in the objective function value is at least some fraction times the decrease in the objective function value at alpha equal to 0.

So, suppose we consider one line which is which one here as a function of alpha that is so we call it as a phi one of alpha. So, this line is drawn here we still do not know what is the exact form of this phi one alpha, but this line is drawn here. Now, what Armijo suggested is that you chose those alphas where the function lies below this line, so in this case, so if you look at this part of the curve that lies below this line. Then this part lies above the line, this part lies below the line and then again it lies above the line. So, according to Armijo's condition the alpha can be chosen such that it lies in either in this range or in this range. So, this is call the acceptable range for alpha according to Armijo's condition.

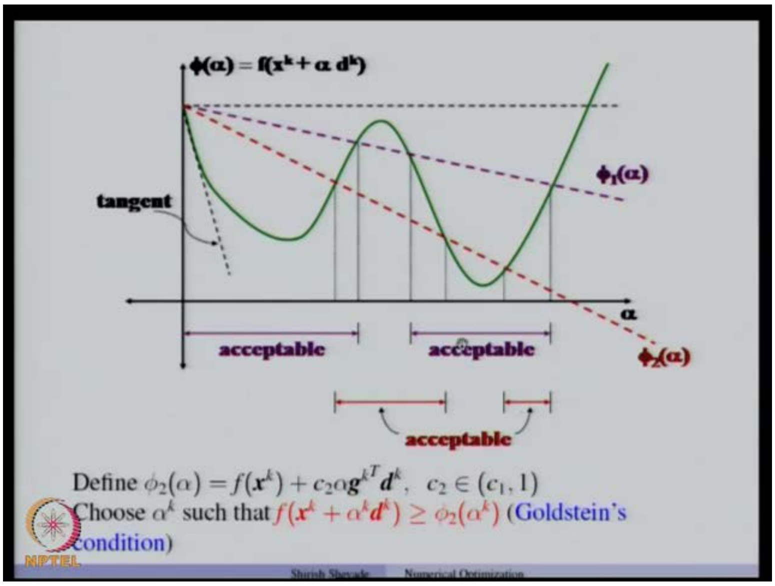
(Refer Slide Time: 46:38)



Now, you will see that this condition will ensure that there is a decrease sufficient decrease in the function value, now more formally suppose if you define phi 1 alpha to be f of x k plus c 1 alpha g k transpose d k. So, this is an this is a function of alpha, so a fine function alpha where c 1 is a constant which is a positive fraction. So, if you define phi 1 alpha given c 1 like this then according to Armijo's condition the alpha k the step length should be chosen such that the value of the function at the new point is less than or equal to phi 1 of alpha k, so this is called Armijo's condition. So, this step is very important that f of x k plus alpha k d k should be less than or equal to phi 1 alpha k and that is satisfied by all alpha k s which are either in this range or in this range. So, that is why this is denoted here by acceptable step lengths according to Armijo's conditions.

Now so this say insures that, so if you look at the right hand expression for phi 1 alpha, now g k transpose d k is a rate of decrease of f in the direction d k. So, direction g k transpose d k is a directional derivative of f at x k, now what we are interested in that that f of x k plus alpha k should be less than or equal to f of x k plus c 1 times g k transpose d k into alpha k. So, there should be a sufficient decrease in the objective function compare to the initial decrease which was g k transpose d k initial rate of decrease which was g k transpose d k.

(Refer Slide Time: 48:51)

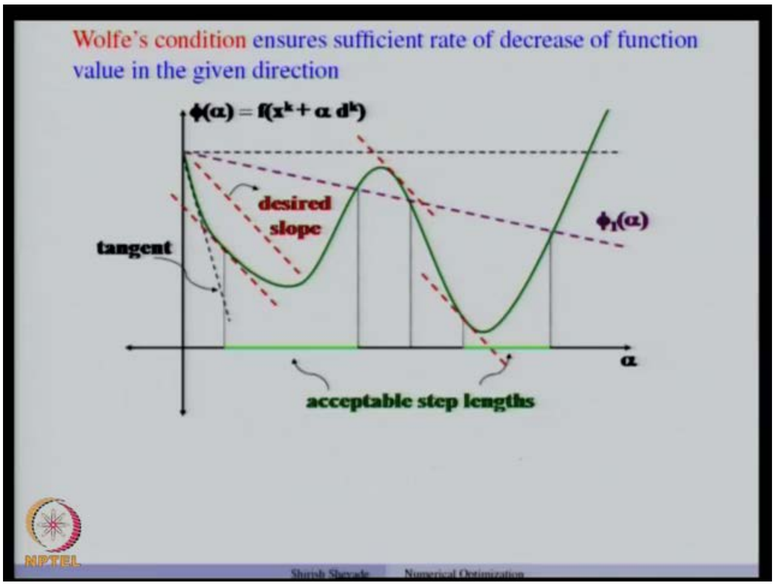


Now, this condition does not insured that the step lengths are small, so there was a condition which was proposed by Goldstein which ensures that the step lengths are not too small. Now, the phi 1 alpha was the function which was related to Armijo's condition, now here we have another function phi 2 alpha and according to Goldstein's condition. The function should the function phi alpha should lie above this line shown by the red color so; that means, that according to Goldstein's condition this part of the function and this part of the function are phi.

So, according to Goldstein's condition this part is acceptable and anything beyond this should be acceptable, but we are not interested in going beyond the case where the function is greater than phi 1 alpha. So, we clip the inter all for alpha at this point, so now if you combined these 2 conditions Armijo's and Goldstein's condition. So, then you will see that this is the acceptable interval length for Armijo's condition and this is the acceptable interval length for Goldstein's condition.

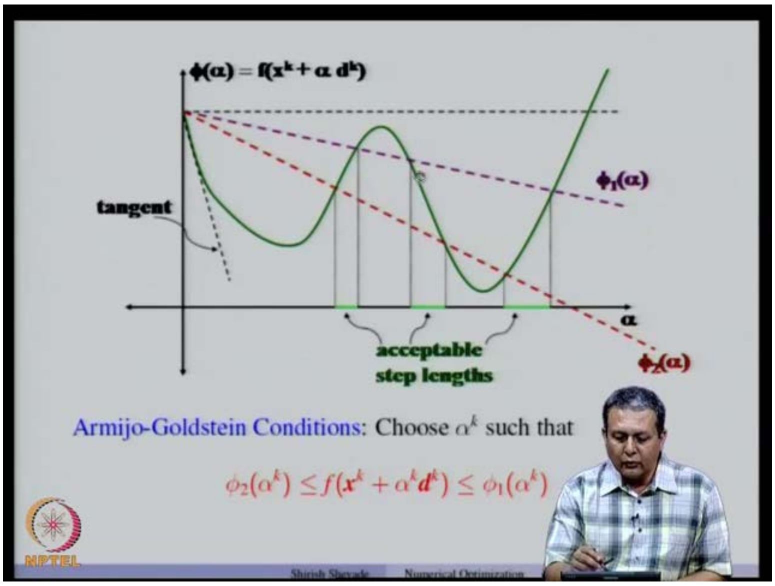
In fact, for Goldstein’s condition any alpha where the function value is greater than this phi 2 alpha should be. So, ideally we should be getting this interval to be going towards plus infinity, but we have clipped it because we do not want to evaluate the Armijo’s condition. So, if we define phi 2 alpha to be a function of to a function where phi 2 alpha is f of x k plus c 2 alpha g k transpose d k where c 2 is a constraint in is the range c 1 to 1. Then the Goldstein’s condition say that your alpha k should be chosen such that f of x k plus alpha k t k is greater than or equal to phi 2 alpha k. So, these are the acceptable intervals according to Goldstein’s condition, these are the acceptable intervals this according to Armijo’s condition.

(Refer Slide Time: 51:40)



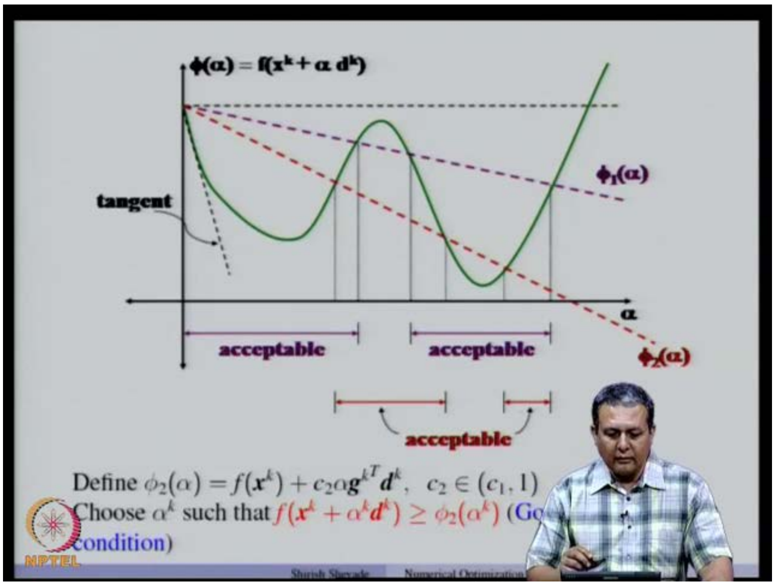
Now, if you combine them, then we get what are called Armijo’s conditions for in exact line search, so according to these conditions we chose alpha k such that the value of the function at alpha k. So, f of x k plus alpha k d k is less than or equal to phi 1 alpha k and should be greater than or equal to phi 2 alpha k, where c 1 and c 2 where phi 1 and phi 2 are define using the constant c 1 and c 2 which are fractions. So, c 1 is in the open interval 0 to 1 and c 2 is in the open interval c 1 to 1.

(Refer Slide Time: 52:22)



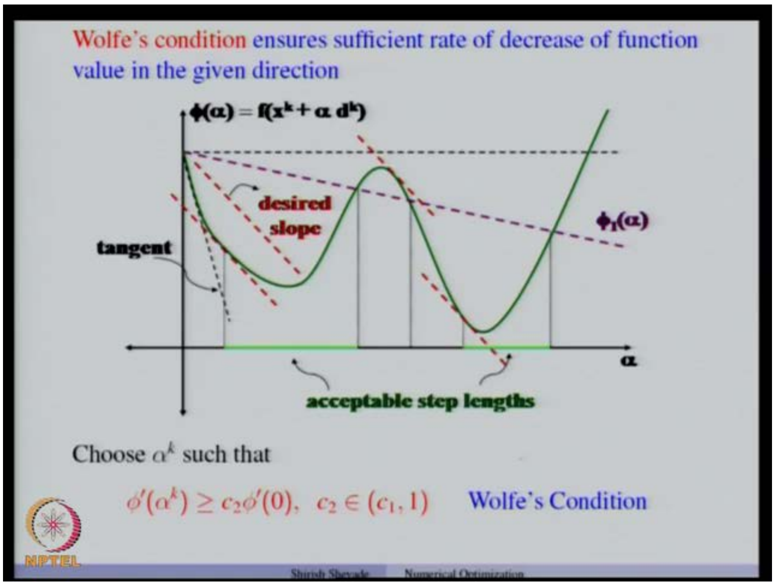
Now so if you take the intersection of the acceptable step lengths for Armijo's conditions and Goldstein's condition, so you will see that you are interested in those intervals where the function lies below the phi 1 alpha line and above the phi 2 alpha line. So, this interval this interval and this interval, these are the acceptable step lengths we satisfy Armijo Goldstein's condition.

(Refer Slide Time: 53:03)



Now, if you look at Goldstein's condition you will see that this local minimum was script by the because of the Goldstein's condition, because Goldstein's condition does not allow the function to be below this line phi 2 alpha. So, the initial local minimum of this function phi alpha was missed by the Goldstein's condition.

(Refer Slide Time: 53:27)



So, Wolfe proposed a condition which ensures that there is a sufficient rate of decrease of function value in the given direction. So, this is the initial slope of the function phi alpha, now what Wolfe suggested is that the new the desired slope of the function should be at some prescribed times the slope at the initial point. So, this is this could be the desired slop, which is some multiple of the initial slope and now we are interested in those intervals where the function value has slope in the function phi alpha has a slope in this desired range.

So, this becomes acceptable step length for Wolfe condition and similarly this becomes a acceptable step length for Wolfe condition plus Armijo's condition. So, one can use Armijo's conditions along with Wolfe condition instead of Armijo's conditions and Goldstein's conditions. So, say if we use Armijo's conditions and Wolfe conditions these are the acceptable step lengths.

And the formally the Wolfe's conditions says that chose alpha k such that the derivative of the function phi at alpha k is at least c 2 times the derivative of the function at c 0 at 0 where c 2 is again a constant in the range c 1 to 1. So, once we fix the once we have the initial slope then what we are interested in is that finding out all those alphas, where the gradient of the function phi at those alphas is at least c 2 times the gradient of the function at 0.

So, if we multiply the initial gradient by c 2 and we get the desired. So, what we are interested in that finding out all those function points where or flinging out all those alphas where the gradient of or the derivatives of phi at those alphas is at least this much the desired slope. So, once we do that then we combine that those conditions with Armijo's conditions and what we get are called Armijo Wolfe condition. So, either one can use Armijo Goldstein's conditions or Armijo Wolfe's conditions to do in exact line search, now given all this what is a guarantee that the algorithm would converges. So, that is if we ensure that the direction at every iteration is a descent direction and alpha k found is using a inexact line search, but which ensures that Armijo Goldstein or Armijo Wolfe conditions are satisfied. Then will the algorithm converge or are there any extra conditions which are needed, so we will see those things in the next class.

Thank you.