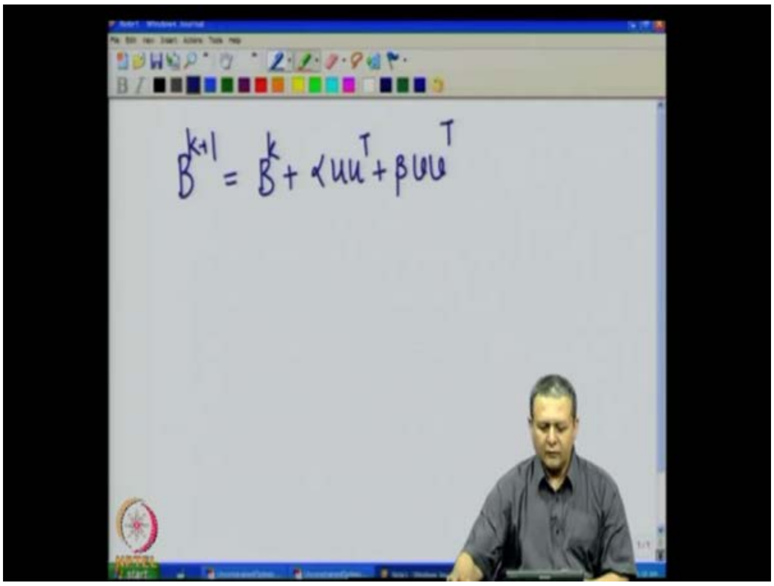
Numerical Optimization
Prof. Shirish K. Shivade
Department of Computer Science and Automation
Indian Institute of Science, Bangalore

Lecture - 17
Quasi-Newton Methods - Broyden Family
Coordinate Descent Method

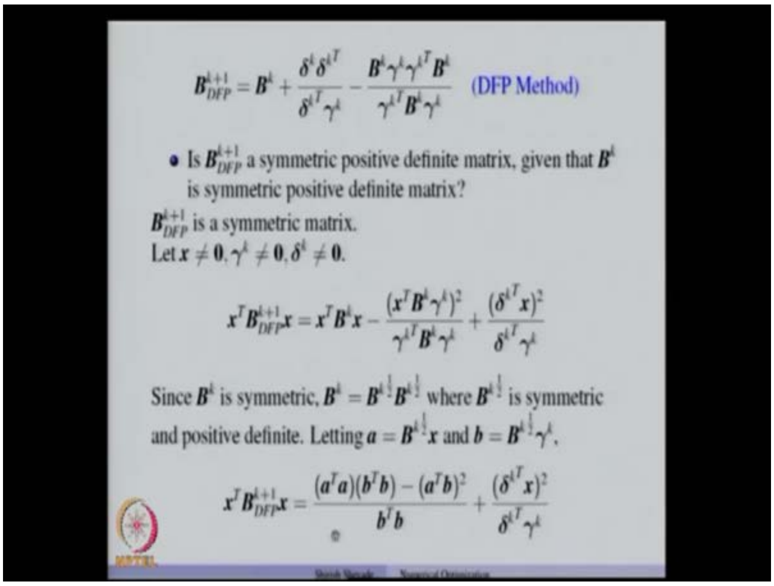
Welcome back to this series of lectures on numerical optimization. In the last class we discussed about symmetric rank one update, and rank two update. So, in particular we saw that there is no guaranty in symmetric rank one update, that the new matrix B k plus 1 is positive definite even though B k is positive definite. So, we decided to use the symmetric rank two update.

(Refer Slide Time: 00:52)



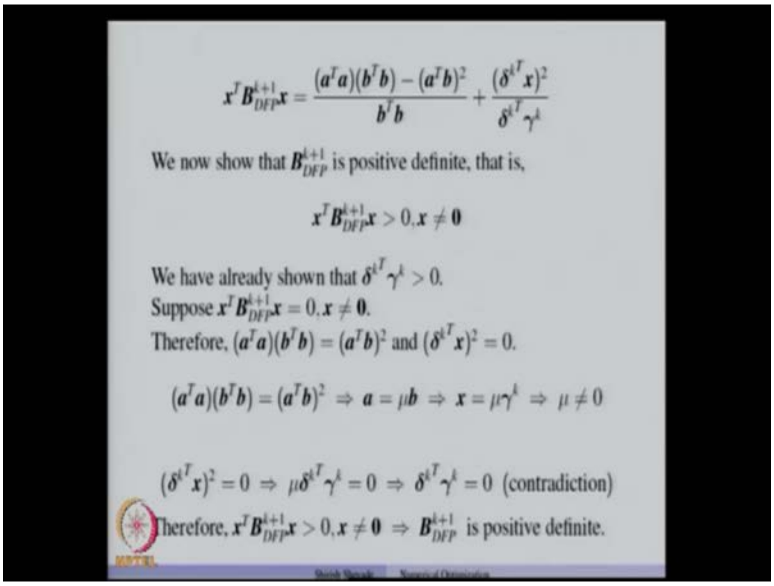
So, the matrix B k plus 1 was found using B k plus alpha u u Transpose plus beta B v v, transpose. Now, the matrix B k plus 1 should also satisfy Quasi-Newton condition, that is B k plus 1 gamma k is delta k. So, if you use that condition then we saw that we got the davidon Fletcher Powell method.

(Refer Slide Time: 01:28)



And we got this formula for B k plus 1 which is nothing but B k plus delta k, delta k transpose by delta k transpose gamma k minus B k gamma k gamma k transpose B k by gamma k transpose b k gamma, and then we started looking at the positive definiteness of this matrix. So, if us showed that this matrix is positive semi definite. So that is clear from this, along with the fact that the numerator here is non-negative using in because of the quasi shots in equality, this quantity is also non-negative because of the quasi shots in equality. So, we saw that x transpose B k plus 1 into x is greater than or equal to 0, and then we moved on to show that it is indeed the positive definite matrix.

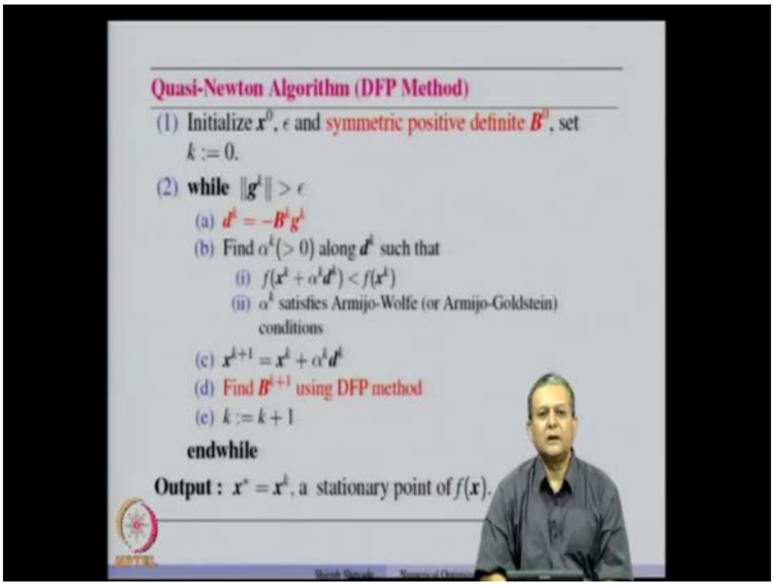
(Refer Slide Time: 02:25)



And for that purpose what we did was, we took 1 x which is non-zero and I will if you assume that x transpose B k plus 1 x is 0, so which means that both this quantities are non-negative. So, if there sum is 0 that means each of the quantities has to be 0. So, which means a transpose in to b transpose b is a transpose b square and delta k transpose x square is equal to 0 which. So, this essentially means that a is a scalar multiple of b and which means that if you recall the definition of x and gamma, we can write this x is a multiple of gamma k and since x is not 0. We can say that mu is not 0. We have assumed that x is not 0 and similarly, if you look at this now this quantity delta k transpose x equal to 0 implies that let us substitute the value of x here. So, mu into delta k transposes gamma k equal to 0.

And since mu is not 0, this will be true only when delta k transpose gamma k equal to 0 but we have already shown that delta k transpose gamma k is greater than 0 and that contradicts our earlier assumption that delta k transpose gamma k greater than 0 and therefore, we have x transpose B k plus 1 x is greater than 0 for all x not equal to 0, which implies that B k plus 1 is positive definite. So, if B k is symmetric and positive definite, we say that B k plus 1 is also symmetric and positive definite if we use the symmetric rank to correction or if we use the D F B method. So, let us see how the algorithm looks like if we use D F B method in our Quasi-Newton algorithm.

(Refer Slide Time: 04:21)

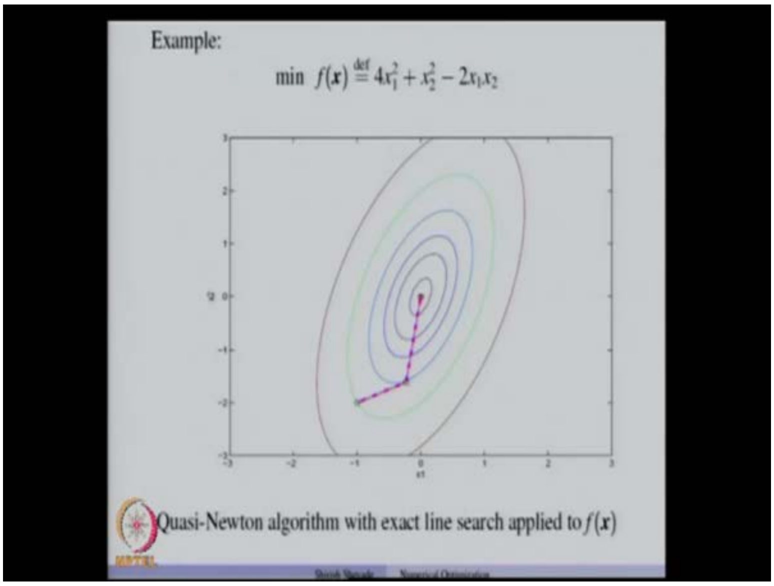


So, compare to the other approaches that we have seen. So, for what we need here is a symmetric positive definite matrix b 0 to starts with of course, we need the initial point the stopping tolerance and then we said the iteration counter to 0. Now while the norm of the gradient is greater than epsilon, we said the direction d k to be minus B k, g k. Now initially we have some matrix B 0. So, suppose if we have no knowledge about the problem then we could use the B 0 matrix to be an identity matrix then we do the line search. So, that Armijo-Goldstein or Armijo-Wolf conditions are satisfied. Then we get x k plus 1 using this formula, that the formula which we have used in our earlier algorithms also and now is the time to find out B k plus 1 using b k, x k, x k plus 1 and g k and g k plus 1. So, if you have all these information we can use D F B method to update B k two, B k plus 1.

So, we get B k plus 1 using suppose D F B method, and then increment the iteration counter and at the new point if the gradient norm of the gradient is less than or equal to epsilon we quit or stop the algorithm or otherwise continue this algorithm till this stopping condition is satisfied and at the end what we get is x star to be x k a stationary point of f x.

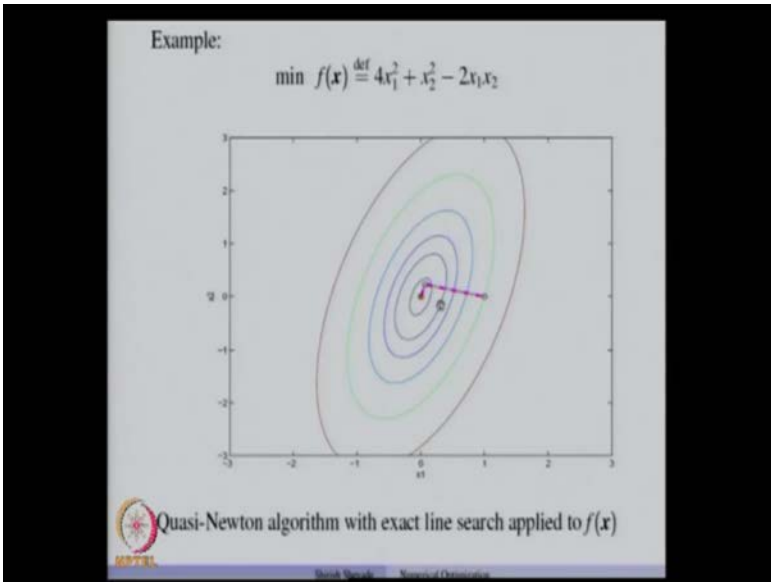
So, you will see that the main difference of this algorithm with newton algorithm or modified Newton algorithm is that, there is no need to find out inverse of any matrix in fact that direction d k is found by matrix where vector multiplication as compare to the direction that that we use in the newton method the direction that was used in the newton method was d k to be minus h in h k inverse g k. So, that required inversion of a matrix and plus the multiplication of matrix and a vector while here there is only multiplication of a matrix by a vector. So, that sales lot of computational time per iteration in D F B method and this update is symmetric rank two update and it is easy to do this update. So, in all what we did need is that matrix vector multiplications or vector multiplications to in one iteration to get the update B k plus 1.

(Refer Slide Time: 07:52)



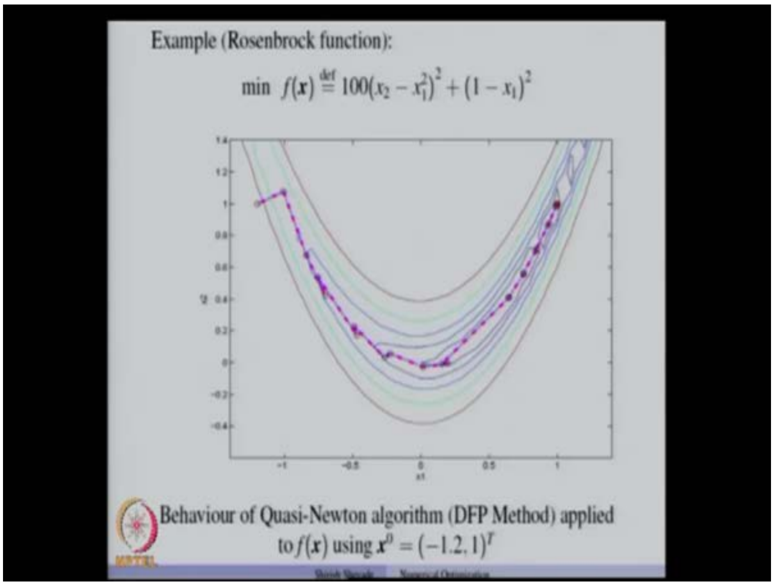
Now, let us take example consider quadratic function which is given here f x equal to 4 x 1 square plus x 2 square minus 2 x 1, x 2 contours of that function. We have already seen when we discussed about other methods. So, these are the contours this is going to be the minimum of this f and when we apply Quasi-Newton algorithm with exact line search remember that, this is the quadratic function. So, is easy to work out these formulas for exact line search. So, if we use Quasi-Newton method the D F B method in the Quasi-Newton algorithm and exact line search. So, if we start from this point. So, this is our x 0 the x 1 is here and x 2 is here and x 2 is nothing but x star. So, we require two steps to reach the solution of the given problem.

(Refer Slide Time: 09:01)



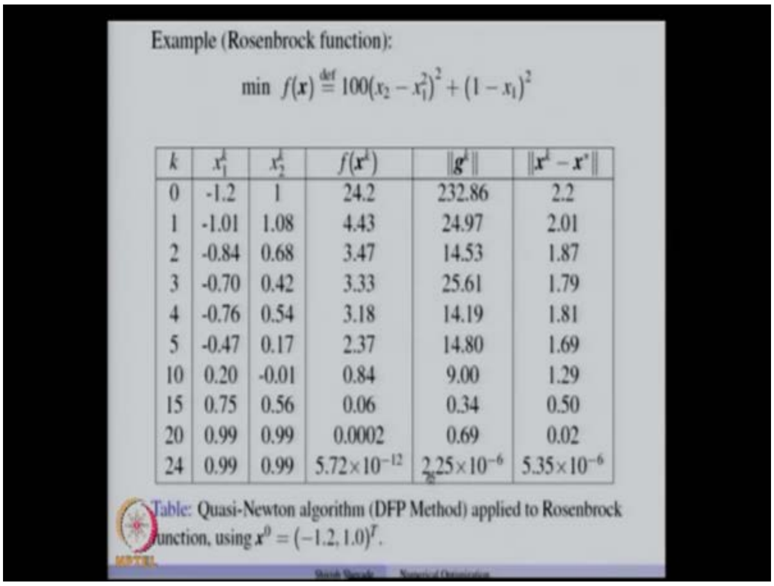
Now we start from another point. So, if you start from here. So, here we will see that will require again two steps this is x 0, x 1 and x 2 is nothing but x star. So, we reach the solution in both the cases in exactly two steps.

(Refer Slide Time: 09:22)



Now, let us look at the performance of Quasi-Newton algorithm with D F B method on the Rosenbrock function. So, if you start from this point you will see that initially it make some movement to this and then there are small steps along this and then finally it goes and converges to the local minimum.

(Refer Slide Time: 09:52)



Example (Rosenbrock function):

$$\min\ f(\boldsymbol{x}) \stackrel{\text{def}}{=} 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

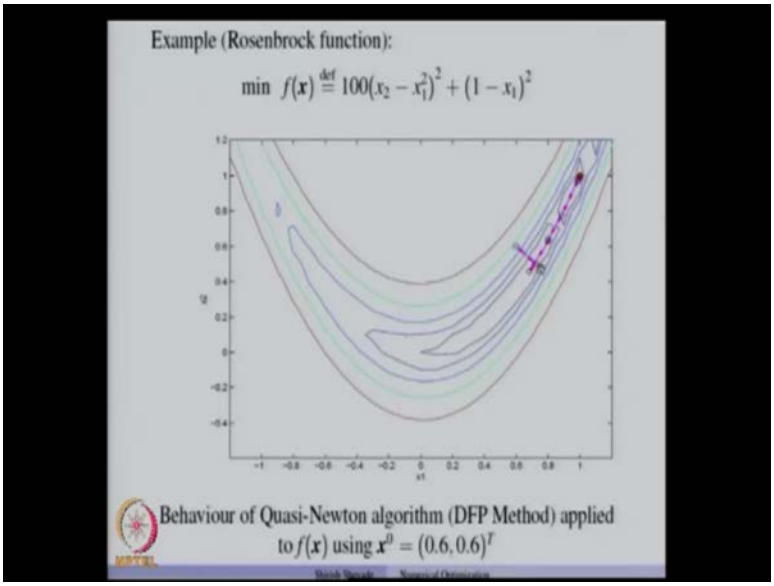
k	x_1^k	x_2^k	$f(\boldsymbol{x}^k)$	$\|\boldsymbol{g}^k\|$	$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|$
0	-1.2	1	24.2	232.86	2.2
1	-1.01	1.08	4.43	24.97	2.01
2	-0.84	0.68	3.47	14.53	1.87
3	-0.70	0.42	3.33	25.61	1.79
4	-0.76	0.54	3.18	14.19	1.81
5	-0.47	0.17	2.37	14.80	1.69
10	0.20	-0.01	0.84	9.00	1.29
15	0.75	0.56	0.06	0.34	0.50
20	0.99	0.99	0.0002	0.69	0.02
24	0.99	0.99	5.72×10^{-12}	2.25×10^{-6}	5.35×10^{-6}

Table: Quasi-Newton algorithm (DFP Method) applied to Rosenbrock function, using $\boldsymbol{x}^0 = (-1.2, 1.0)^T$.

Let us look at the iterations with that initial point. So, initial point was a minus 1.2 and 1. The value of the function at that point was 24.2 and the norm of the gradient was 232.86 and x star is 1 1. So, this turns of x k from x star is 2.2. Note that, this distance is a Euclidean distance now from iteration. So, this was a initial point. So, when we go to 0.x1 which is minus 1.01, 1.02. You would see that there is a significant decrease in the objective function value from 24.22. It has come down to 4.43 and again there is a significant decrease in the norm of g k allow the distance between the x k and x x star has not decrease that much but then as the iteration progress you will see that and the end of 10th iteration the function value is 0.84.

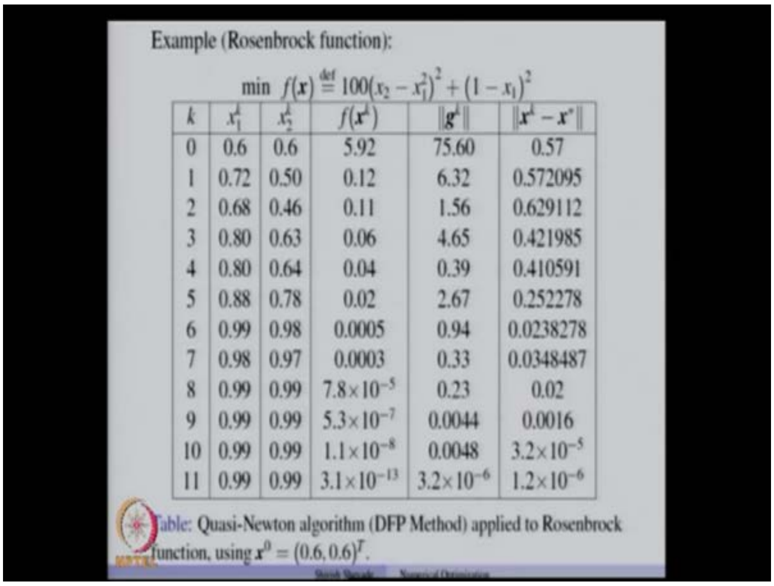
The norm of the gradient is 9 and the distance has come down to 1.29 and at the end of 20 iterations you would see that the value of the function is very small. The norm is also very small and the distance of x k from x star is also small and if you use the stopping criteria of norm g k less than or equal to epsilon where epsilon is 10 to the power minus 3, then we will see that after 24 iterations that norm g k is indeed less than epsilon and the algorithm terminates.

(Refer Slide Time: 11:42)



Now, if we start from some other point which is say 0.6, 0.6 and 0.6. So, this is how the algorithm behaves.

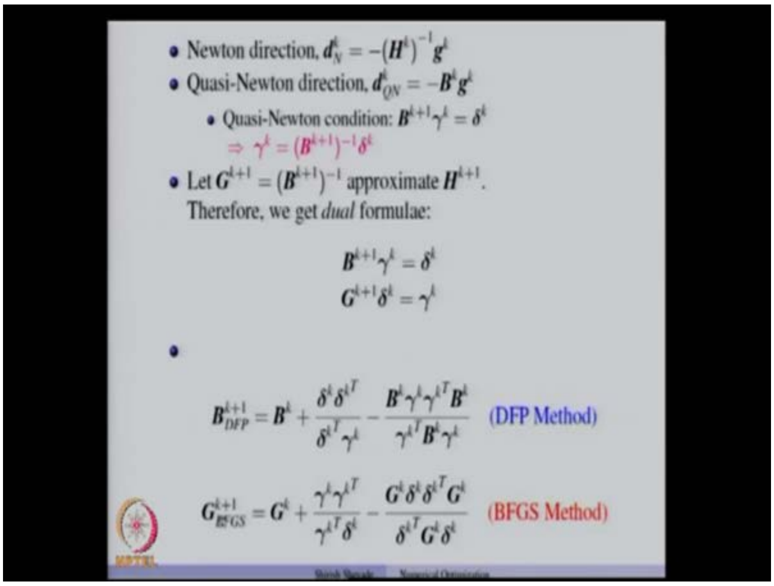
(Refer Slide Time: 11:53)



k	x_1^k	x_2^k	$f(\boldsymbol{x}^k)$	$\lVert \boldsymbol{g}^k \rVert$	$\lVert \boldsymbol{x}^k - \boldsymbol{x}^* \rVert$
0	0.6	0.6	5.92	75.60	0.57
1	0.72	0.50	0.12	6.32	0.572095
2	0.68	0.46	0.11	1.56	0.629112
3	0.80	0.63	0.06	4.65	0.421985
4	0.80	0.64	0.04	0.39	0.410591
5	0.88	0.78	0.02	2.67	0.252278
6	0.99	0.98	0.0005	0.94	0.0238278
7	0.98	0.97	0.0003	0.33	0.0348487
8	0.99	0.99	7.8×10^{-5}	0.23	0.02
9	0.99	0.99	5.3×10^{-7}	0.0044	0.0016
10	0.99	0.99	1.1×10^{-8}	0.0048	3.2×10^{-5}
11	0.99	0.99	3.1×10^{-13}	3.2×10^{-6}	1.2×10^{-6}

And if you look at the table so you will see that we started from 0.6, 0.6 the value of function of 5.92 norm of G k is 75.6 and we are reasonably close to the solution compare to the previous initial point that we saw. Now, in the first iteration itself the value of the function came down to 0.12 there was a slight increase in the distance of x k from x star but norm of G k has also come down significantly and as a iterations progress you will see that in the 7th iteration itself the value of the function is 0.0003 and norm of the gradient is 0.33. So, value of the function is quite small the distance of x k from x star is quite small. So, you will see that we are almost close to the solution here at the end of the 7th iteration and although the x 1 and x 2 co-ordinates for 8, 9^{th}, and 10th iterations would look like they are different but because of the lack of space they could not be shown exactly. So, the approximate numbers are given here but you would see that the values of the functions are different. So, although they look similar here actually the points are different and that is clear from the value of the function that you get here. So, you will see that at the end of 10th iteration norm of g k is 0.0048 and the distance of x k from x star is 3.2 into 10 to the power minus 5 and the next iteration norm g k becomes bill less than 0.001 and the algorithm terminates. So, you would see that the value of the function is close to 0 in fact the value of the function is 0. When at the local minimum and that local minimum is x 1 equal to 1 and x 2 equal to 1 and the algorithm has found that particular point with certain accuracy.

(Refer Slide Time: 14:16)



Now, recall that the Newton direction that we used was minus H k inverse g k where h k is the Hessian matrix at the iteration at the current iteration k and g k is the gradient at the current iteration k. Now, when we use Quasi-Newton method, we use Quasi-Newton direction and the idea the reset approximate the H k inverse matrix by a positive definite matrix B k and use the direction minus B k g k. So, this helps us avoid the inverse calculations as compared to the Newton direction. Now, the Quasi-Newton direction d k, which uses the matrix B k we know that for the Quasi-Newton direction to work the matrix B k should satisfy Quasi-Newton condition and that is B k plus 1, gamma k is equal to delta k.

Now, if B k plus 1 is positive definite. So, it is clearly invertible and therefore, we can red gamma k to be B k plus 1 inverse delta k. Now let us look at some matrix G k plus 1, which is obtained using B k plus 1 inverse and let us assume that, this G k plus 1 approximates H k plus 1. So, if we take the inverse of the matrix B k, we get g k to be B k inverse and that g k will be a good is expected to be a good approximation of H k.

So, if you combine this formula B k plus 1 gamma k equal to delta k and write a formula in terms of gamma k equal to B k plus 1 inverse delta k and B k plus 1 inverse is replace by g k plus 1 then what we get is g k plus 1 delta k equal to gamma k. So, if you look at this two formulae, there is a dual relationship that exist between the variables for example, there is a relationship between delta and gamma here gamma and delta here and B k plus 1, g k plus 1 here. So, that essentially tells us that if we have some update formula for the matrix B using gamma k and delta k.

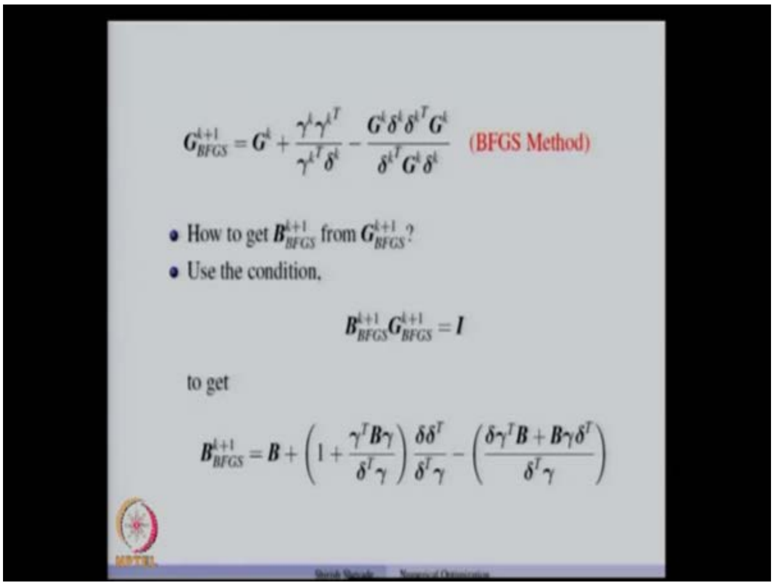
We can as well get an update formula for G by replacing gamma in the B update formula by delta and delta in the B update formula by gamma and the relationship is dual. So, one the other hand if we have a update formula for G, we can replace G by B in that formula, delta by gamma in that formula and gamma by delta and we get a update formula for B. So, this dual relationship can be exploited to get different formulas and note that here we in this formula we work with the matrix G which is an approximation of the matrix H at given iteration. So, if you look at D F B method.

Where we had this update formula B k plus 1 to be B k plus this quantity minus the other quantity which is B k, gamma k, gamma k transpose b k by gamma k transpose, b k gamma k now, if you use this dual relationship we can write a formula in terms of G delta in gamma by replacing B by G and gamma by delta and delta by gamma in this formula and what we get is called the B F Gs update for G.

So, the BFGs stands for Broyden Fletcher Goldfarb and Shannon the inventors of this met update method. So, you will see that B is replace by G delta is replace by gamma and gamma is replace by delta and we get update formula for G and we expect that.

This G is a good approximation of the actual hessian. Now remember that if we use the matrix G in this formula, then what we have to write is that d k QN is equal to minus g k inverse g k, where the first g k inverse corresponds to on the matrix g k which approximates H k. So, which again involves inversion of a matrix and that is going to be computational expensive. So, instead we would like to get a formula for G and then find a corresponding update formula for B and that formula can be use here instead of G k inverse and with let us see how to do that.

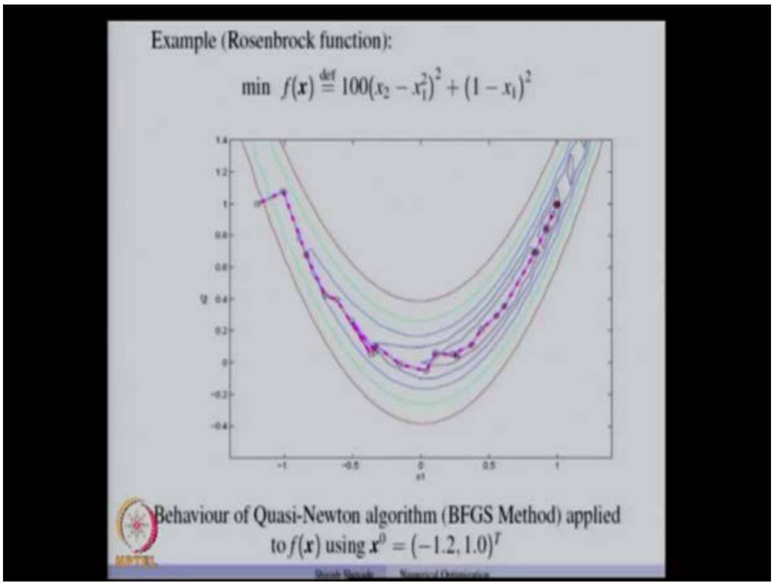
(Refer Slide Time: 19:59)



So, the natural question that one would like to ask is that how to get B k plus 1 from G k plus 1? And that is obvious from this relationship that G k plus 1 is equal to B k plus 1 inverse. So, which means that B k plus 1 into G k plus 1 should be equal to an identity matrix. So, if we use that relationship that B k plus 1 B F Gs in to G k plus 1, B F Gs equal to identity then using some matrix identities. What we get is the B F Gs update formula for the matrix B and that would be something like this.

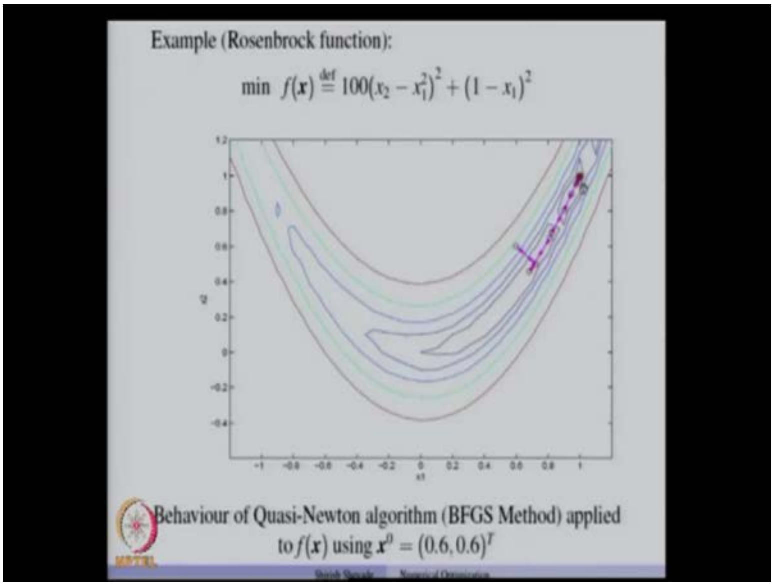
So, the right side to avoid notational clutter all the superscripts k has been suppressed but they do exist. So, all the right side terms involving B gamma and delta are the terms evaluated at the current iteration k. So, B is B k gamma is gamma k and delta is delta k in the right hand side of this expression. So, this is called B F Gs update formula for the matrix B. So, one can be derive similar formulas given any update formula for the matrix B.

(Refer Slide Time: 21:27)



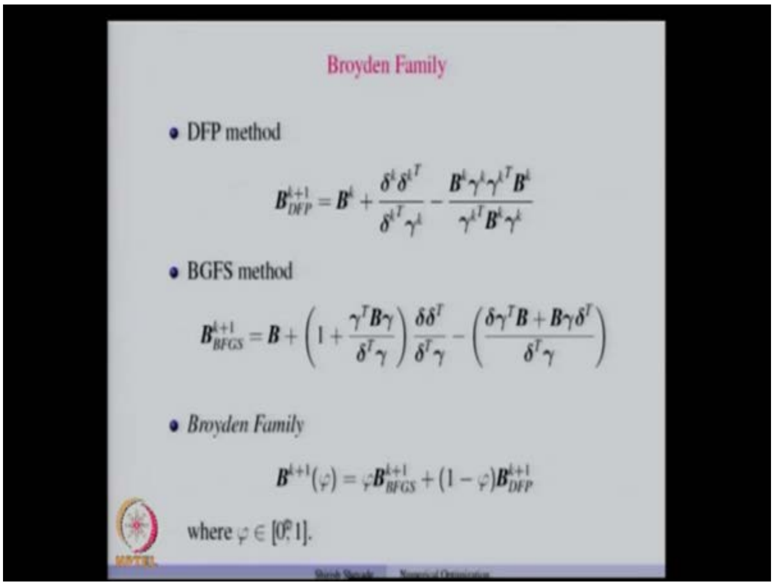
Now, let us apply this B F G S method to Rosenbrock function and this is the behavior that one gets of the Quasi-Newton algorithm where B F G S method is used.

(Refer Slide Time: 21:48)



And so if you start from a different initial point we see a similar behavior as we saw in the DFB method.

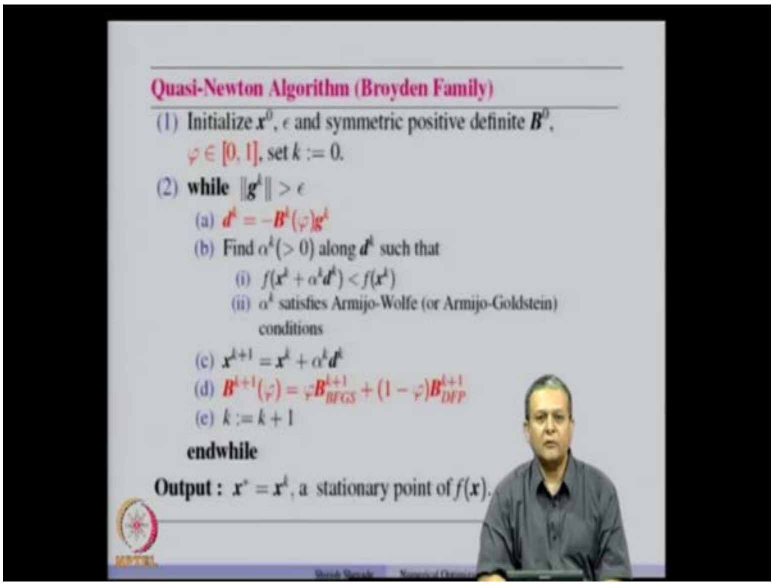
(Refer Slide Time: 21:58)



So, so far we have studied two rank one update formulas one is DFB, two rank two update formulas one is the DFB formula and other one is the B F Gs formula. Now these two formulae can be combined to get an update formula which belongs to what is called Broyden family and that would look something like this that... So, if we chose phi to be a positive fraction and B k plus 1 in the Broyden family is obtained using phi in to B k plus 1 B F Gs plus 1 minus phi into B k plus 1 DFB. So, you will see that when phi equal to 0.

What we have is the DFB update formula and when phi equal to 1 we have the B F G S update formula. So, Broyden family consists of BFP formula as well as B F Gs formula at the two extremes plus convex combination of the B F G S and DFP updates for the matrix B. So, this is also called pure Broyden family, because phi is kept constant here sometimes this phi can be also made a function of the iteration k. Now it is clear that if we chose phi in this range this B k plus 1 will be symmetric and positive definite, because we know that both B F G. And DFB update formulas for B are symmetric and positive definite. So, this gives a family of update rules for the matrix B and this is called Broyden family.

(Refer Slide Time: 24:18)

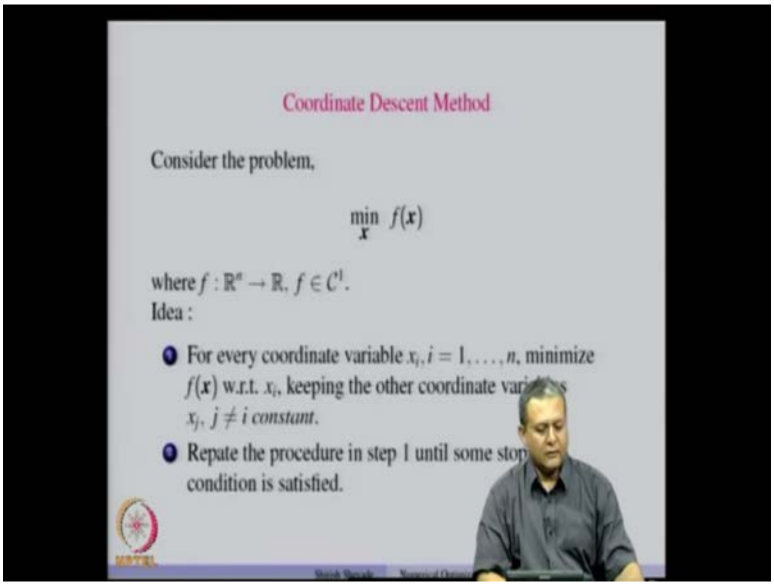


So, the algorithm when we use Broyden family is given here. So, what we need is that apart from the symmetric positive definite matrix B naught we need the variable phi which is in the closing trouble 0 to 1. So, the first step is to get direction d k which is minus B k of phi into G k and best on the value of phi will get B minus B k phi that multiply by G k would give us B k the line search is then done.

So, has to satisfy Armijo Wolf or Armijo Goldstein Conditions, then we get x k plus 1 to be x k plus alpha k d k. The new point is obtained and then we have to update B k and that is obtain using Broyden family update formula and that is B k plus 1 is a convex combination of B k plus 1, D F Gs and B k plus 1. DFB and the iteration contours are increased as in the previous cases and finally, we get stationary point when the algorithm terminates. So, these are some of the update formulas in the Quasi-Newton method and will come back to this Quasi-Newton directions and Quasi-Newton methods some time later.

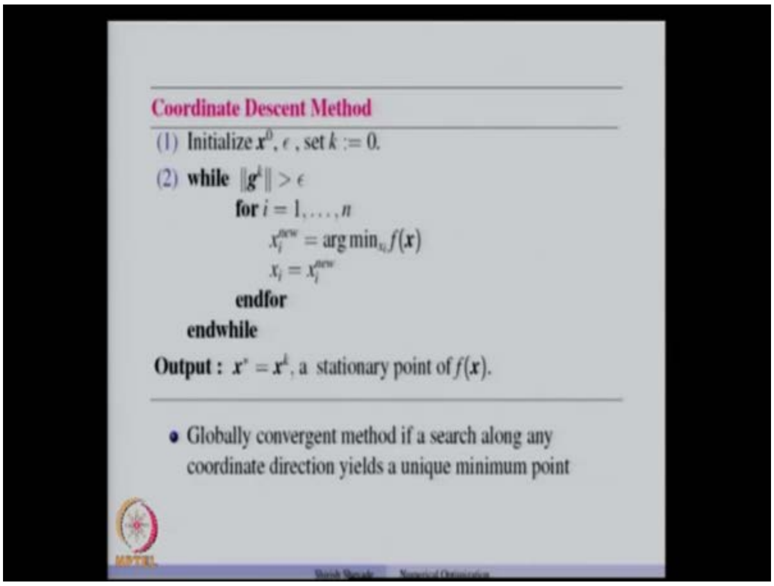
So, let us now look at some of the simplest techniques for minimizing a function and one of the simplest techniques is called Coordinate Descent Method.

(Refer Slide Time: 26:30)



So, as a name suggest the directions that are chosen are the coordinate directions and. So, we choose one coordinate direction at a time and optimize a function with respect to that coordinate alone keeping the other coordinates fixed and the procedure is repeated till we get an optimum point. So, let us see how to do that. So, consider this problem where we want to minimize f x, f is a real valued function whose domain is R n and f is differentiable. Now the idea of coordinate descent method is that, we consider every coordinate x i there are n such coordinates and minimize the function f x with respect to x i alone keeping the other coordinate variables x j where j is not equal to i constant. So, this becomes one-dimensional optimization problem which is many times easy to solve compare to n-dimensional optimization method. So, this procedure is repeated for every coordinate variable. So, after scanning all the n coordinates, once again goes back to the first coordinate and starts optimizing it and the whole procedure is repeated till some stopping criteria is satisfied.

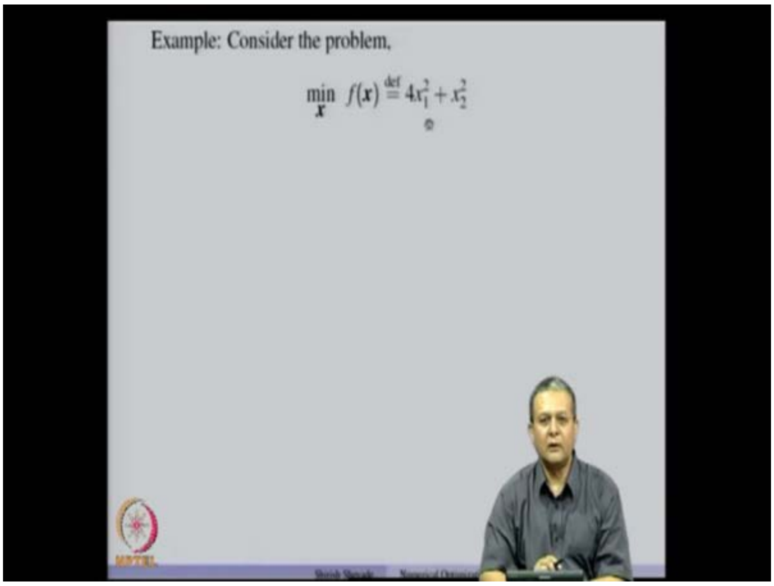
(Refer Slide Time: 28:23)



So, this procedure in step one is repeated till the stopping condition is satisfied. So, let us look at algorithm is very simple, what we need is a initial point some stopping tolerance which said the iteration contour to 0. So, while the norm of the gradient is greater than epsilon, we take each coordinate and minimize the function f x with respect to that coordinate x i and when we find the minimum we get the new value of x i and that will be x i new then the variable x is set to x i new and we are ready to go to the next coordinate so this procedure.

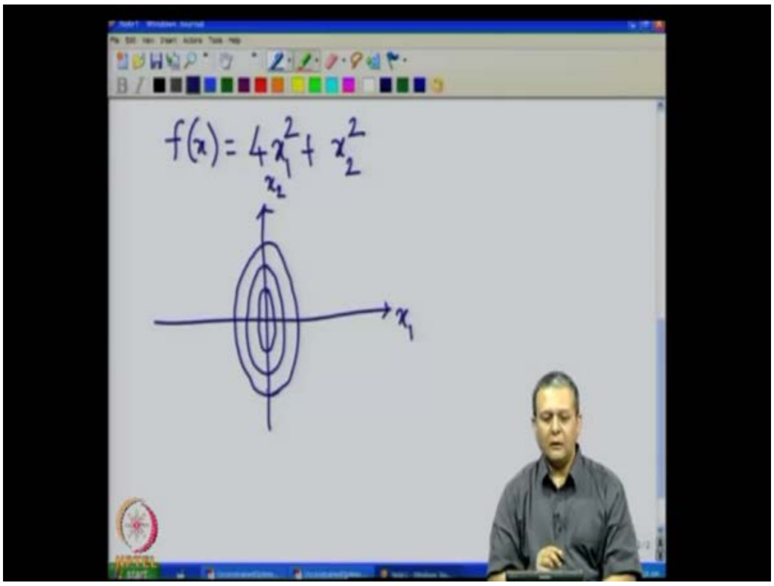
So, once scan of all the coordinates is complete then we go back and check whether the norm of the gradient is less than or equal to epsilon, if it is not we continue again scanning all the n coordinates and bring the updates till norm of g k becomes less than or equal to epsilon and that point we terminate the algorithm. So, a very simple way of optimizing a function and under certain assumptions one can show that, this method is globally convergent. That is if we assume that a search along any coordinate direction you will say unique minimum point then this method is globally convergent. So, a very simple method and under some mild assumptions of existence of unique minimum along coordinate direction right makes this method globally convergent. Now, let us take some example.

(Refer Slide Time: 30:29)



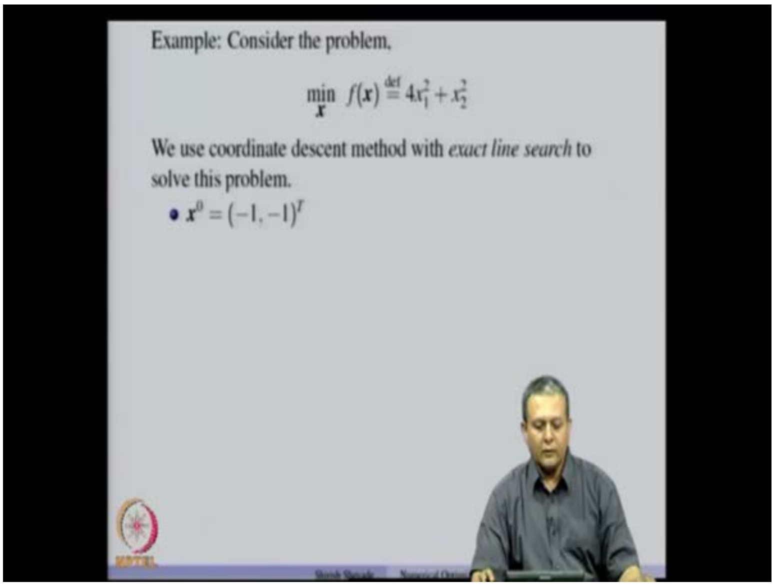
So, let us consider the problem there we want to minimize f x is the quadratic function in this case and f x is the defined as 4 x 1 square plus x 2 square. So, let us look at this function first.

(Refer Slide Time: 30:51)



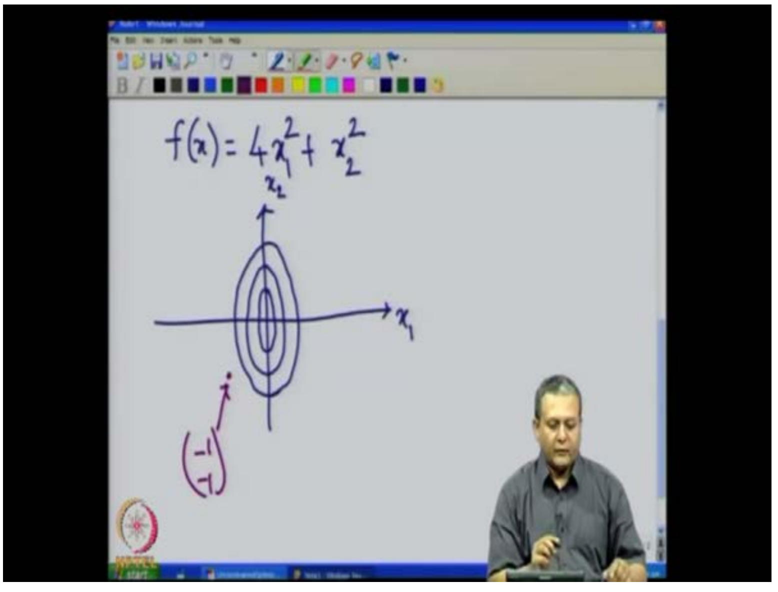
So, we have f x equal to 4 x 1 square plus x 2 square now let us draw the contours of this function. So, this is x 1 and this is x 2 and a contours would contours are electrical contours and they would look something like this. So, this is the function that we want to optimize.

(Refer Slide Time: 31:42)



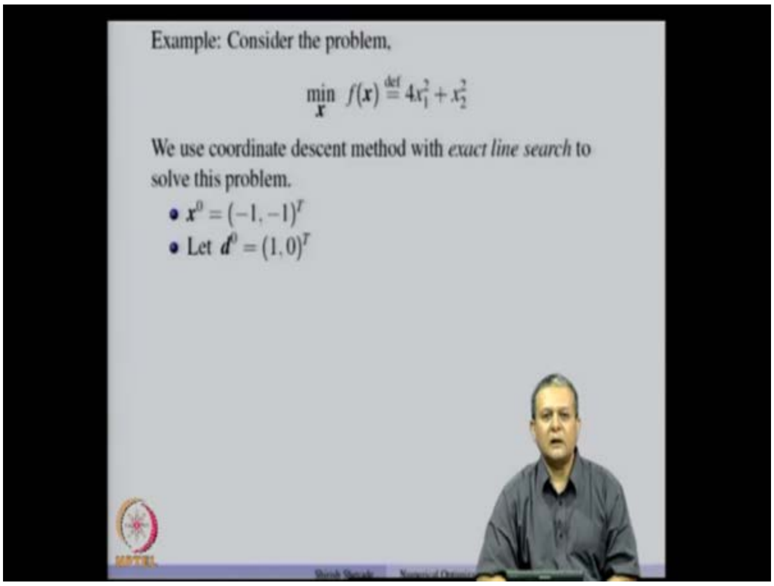
And suppose we decide to use coordinate descent method and seen the function is quadratic it is easy to use the exact line search to solve this problem so let the initial point be minus 1 minus 1. So, let us consider some initial point.

(Refer Slide Time: 32:03)



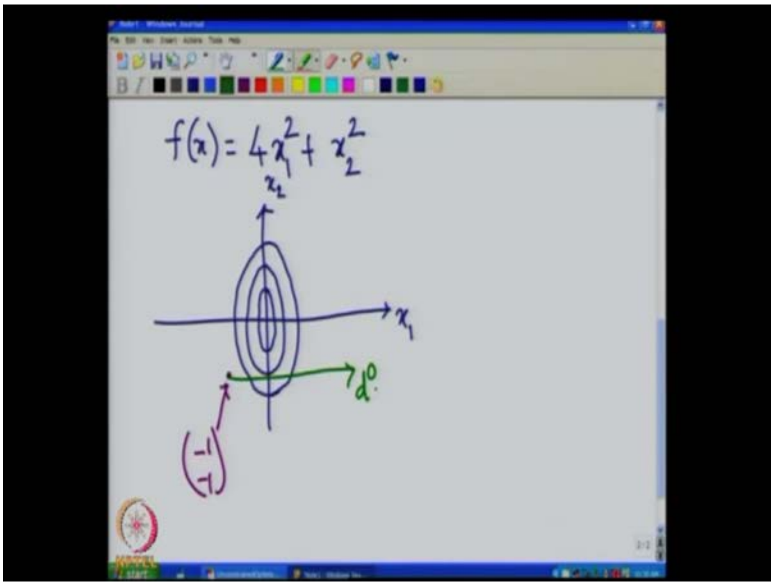
So, let me denote it suppose this point is minus 1 minus 1. So, this is the point from which we start and let us see how the algorithm works.

(Refer Slide Time: 32:30)



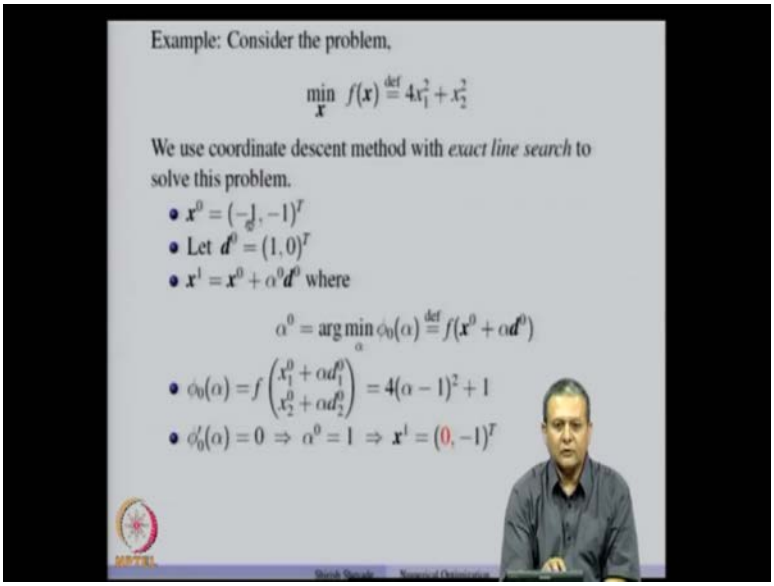
Now, let us choose the direction first direction to be 1 comma 0. So, that is the x 1 direction. So, initially we make a progress along the direction d 0 and then we use the next coordinate direction which is 0 1. So, initially we choose the direction.

(Refer Slide Time: 32:54)



So, this is d 0 which is the x 1 coordinate.

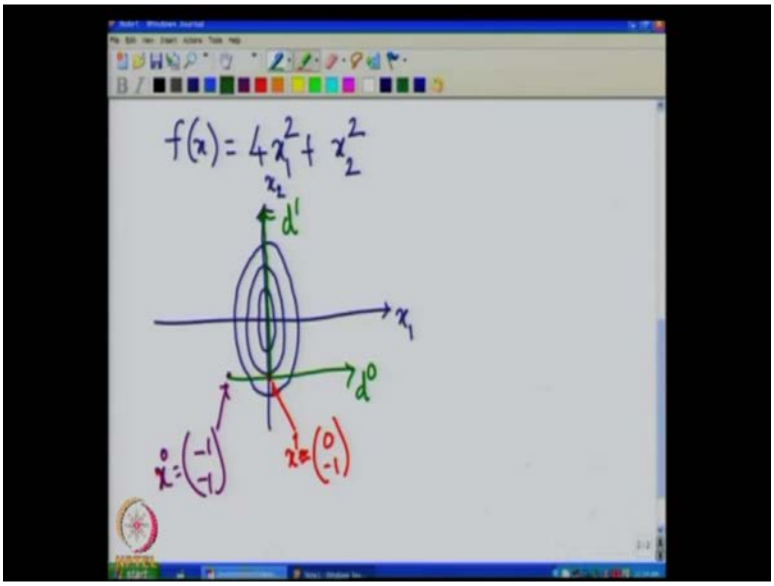
(Refer Slide Time: 33:16)



Then, we find x 1 to be x 0 plus alpha 0 d 0 using our usual update for x 1 and alpha 0 is obtained by solving another optimization problem which is minimizing a function f of x 0 plus alpha d 0. So, we know d 0 remember that, this alpha is should be greater than 0 greater or equal to 0, when we do this minimization. So, by solving this problem we get alpha 0. Now if we explicitly write the phi 0 alpha to be f of x 1 0 plus alpha d 0 and x 2 0 plus alpha d 2 0. So, these are the two coordinates for f and if we use this x 0 and d 0 substitute them here and plug them in this formula. So, what we get is an expression in terms of alphas.

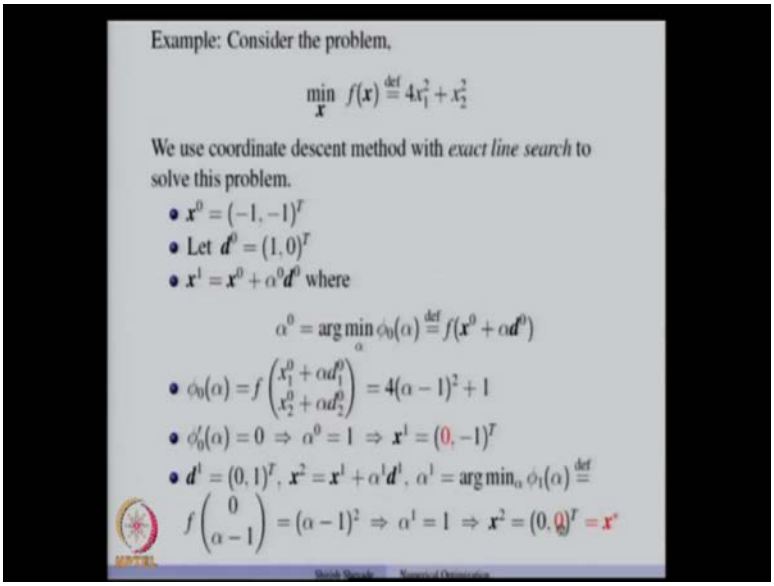
So, remember that alpha is the only variable for this function phi and this phi is define using f like this where x 0 and d 0 are known. So, finally, we get an expression in terms of only alpha now it is clear that the minimum of phi 0 alphas with respect to alpha is obtained when alpha when the derivative of phi with respect to alpha vanishes at a point alpha 0 equal to 1 and therefore, what we get is x 1 equal to 0 minus 1 on this is the point on the y axis. So, you will see that initially we started with minus 1, minus 1 and is which was this direction d 0 to be our initial search direction and when we did the exact line search we move to the point x 1 where this first coordinate become 0 second coordinate remained as it is.

(Refer Slide Time: 35:40)



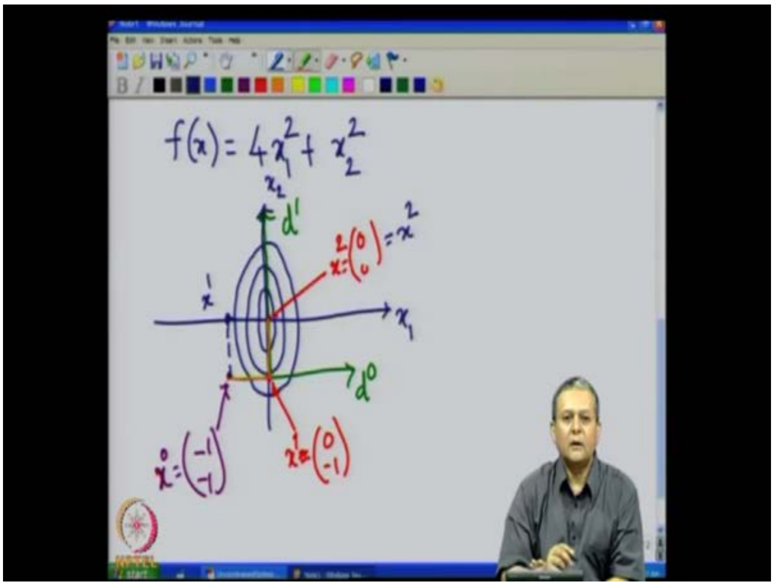
So, you will see that we reach the point. So, this was x 0 and this was x 1 to be 0 minus 1. So, this is our current point and then we have to choose the direction d 1. Now the second coordinate direction that we have to choose and that direction is d 1. Now let us see how to do that optimization.

(Refer Slide Time: 36:38)



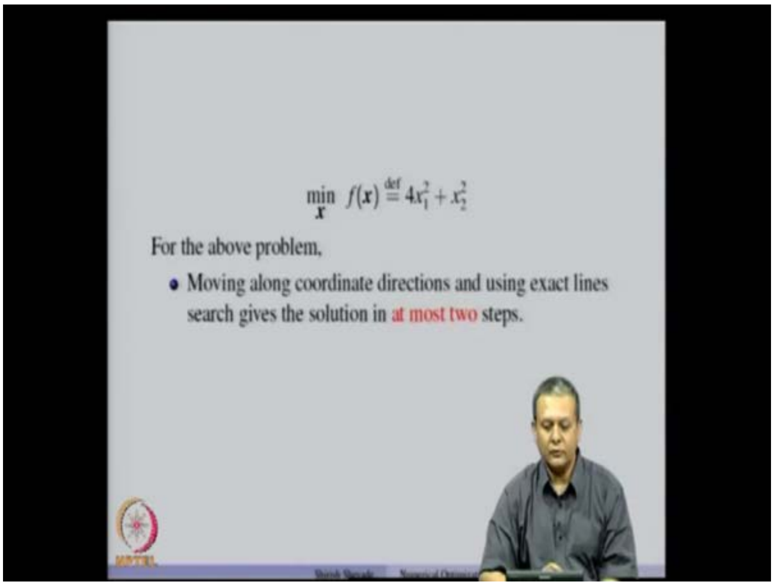
Now, we use the next coordinate direction which is 0 1 and we know that x 2 is obtained by this formula x 1 plus alpha 1 d 1 where alpha 1 is obtained by minimizing from function. So, let us see what that function is, so phi 1 of alpha is nothing but minimum of. So, if we write x 2 to be x 1 plus alpha d 1 and we know x 1 and we know d 1 plug in those values of x 1 and d 1 in x 1 plus alpha 1 d 1 we get the 2 coordinates as 0 and alpha minus 1. So, f of this if you plug in this coordinates herein the original expression what we get is alpha minus 1 square and if we minimize this with respect to alpha that is take the derivative of with respect to alpha set it to 0. We get alpha 1 equal to 1 and once we get alpha 1 equal to 1 then substitute it here we have x 1 which 0 minus 1 and d 1 which is 0 1. So, if you combine them what we get is x 2 equal to 0 comma 0 are the origin and we know that origin is the solution of this problem.

(Refer Slide Time: 38:06)



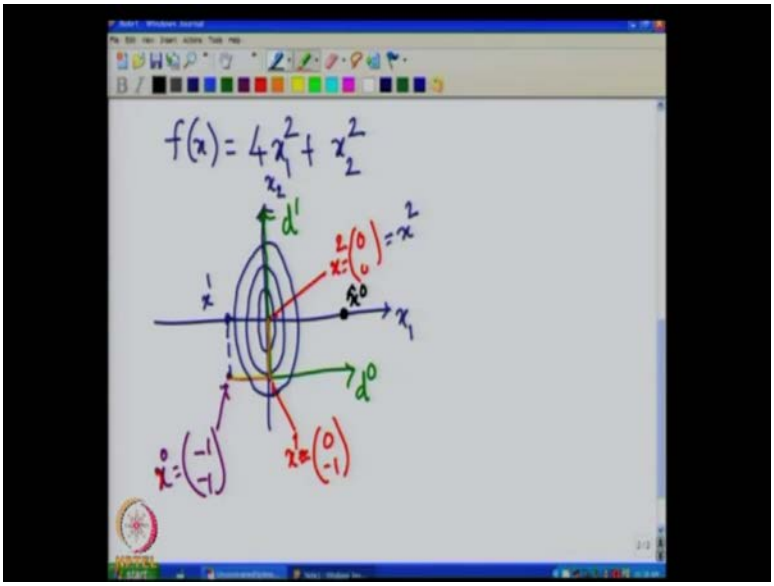
So, we get a new point which is our x 2 and that is nothing but 0 0. So, the path truest by this method is. So, initially we move up to this point to x 1 and then we move to the point x 2 0 now suppose we had use the alternate directions like initially suppose we have we had moved along d 1 and then along d 0 then you would notice that we would we would have got the similar behavior. So, in that case we would have got this point as our x 1 and this point as our x 2. So, you will see that even if you alternate our directions of search in this case d would still get the same point.

(Refer Slide Time: 39:25)



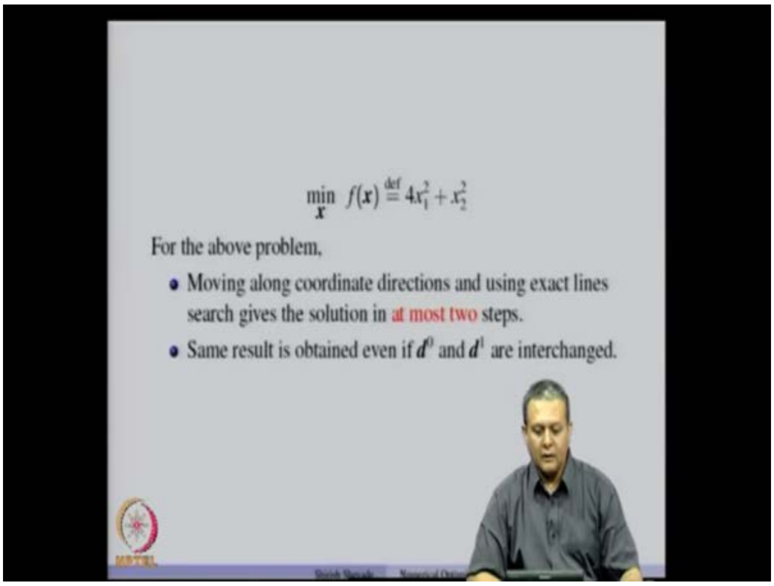
Important point to remember is that if you move along coordinate directions and used exact line search in this case we got the solution in at most two steps.

(Refer Slide Time: 39:44)



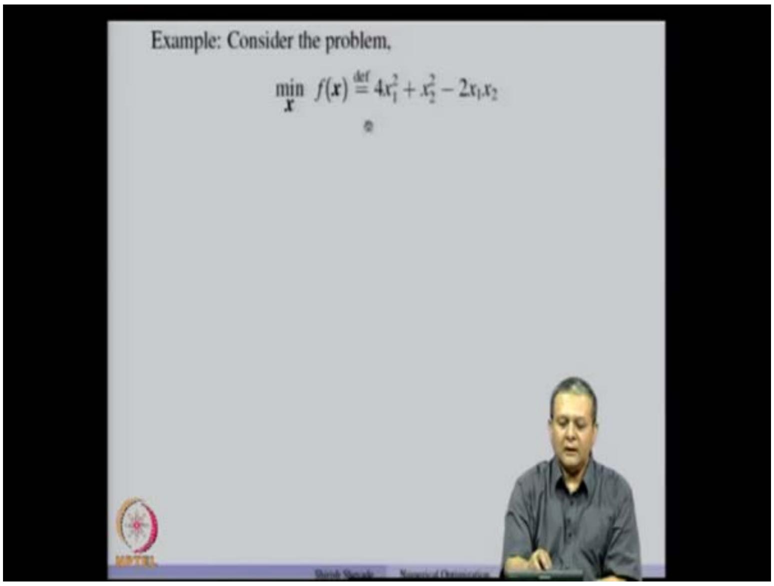
Suppose, we are lucky and we start with initial point suppose this is our x 0 hat now you will see that in one iteration we could if you move along x 1 in one iteration we could reach the solution sometimes we could get the solution in one iteration if we are along the principle axes of this elliptical contours but otherwise we would require at the most two steps to reach the solution.

(Refer Slide Time: 40:26)



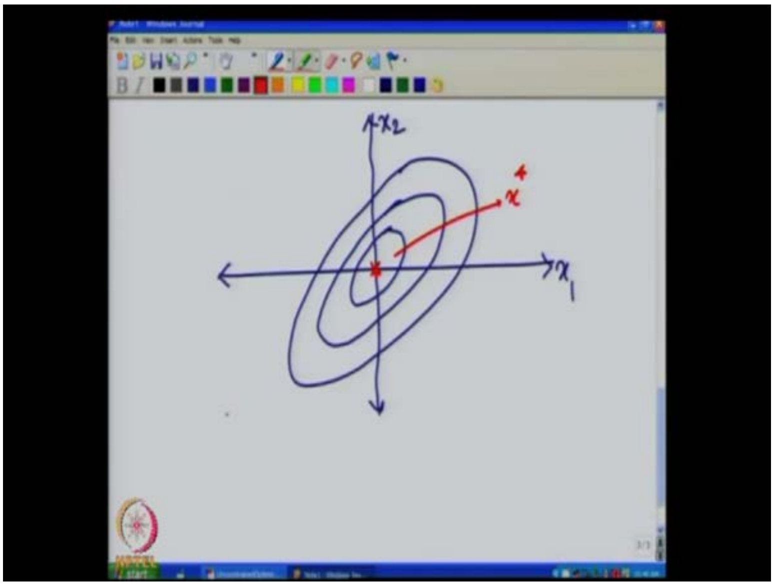
Now, we will get the same solution x star even if we alternate our search directions that is if we initially search along the direction d 1 and then along the direction d 0 we would still get the same final solution x star so that order of the descent directions is not going to matter in this case. So, these are the some important points that one can remember.

(Refer Slide Time: 40:52)



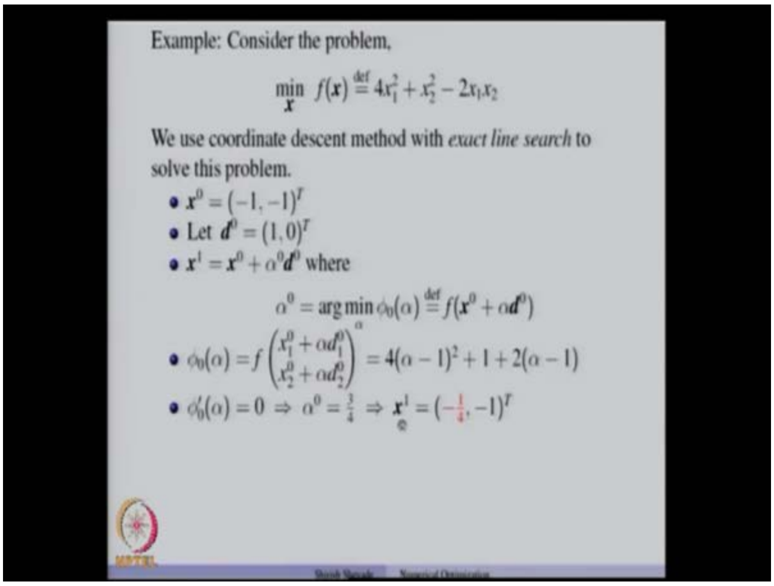
Now, let us consider a case where we have a function f x defined as 4 x 1 square plus x 2 square minus 2 x 1 x 2. So, let us draw the contours of this function.

(Refer Slide Time: 41:01)



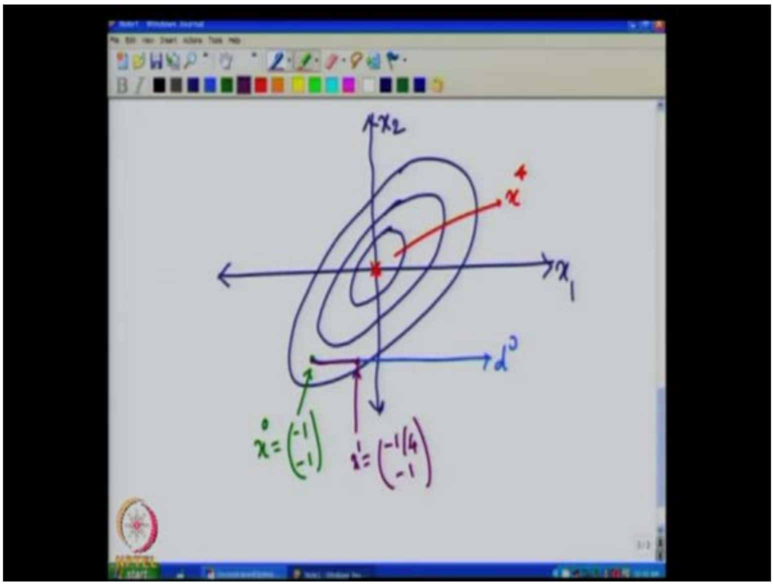
So, we have seen that the contours of this function they look like this x 1 and x 2 the contours look the like this and this is going to be the minimum of our problem and this is going to be our x star. So, let us use the coordinate descent method to solve this problem.

(Refer Slide Time: 42:04)



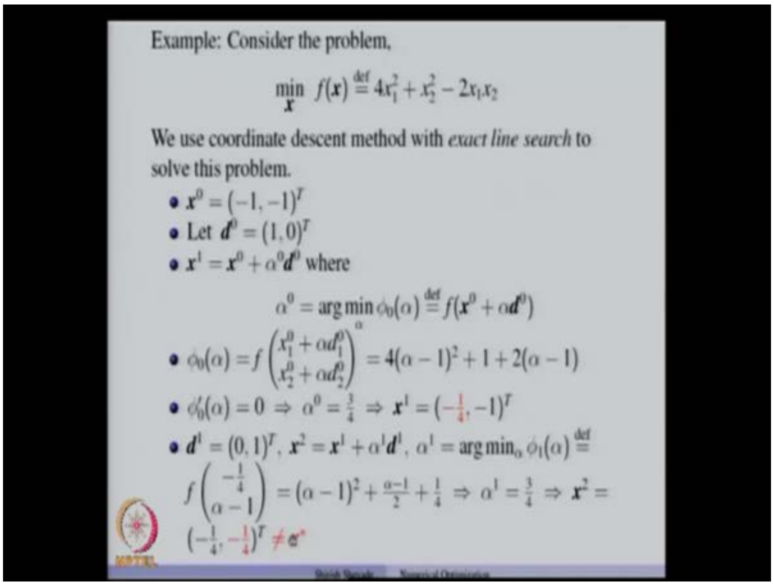
Again the function is quadratic we use the exact line search and suppose we start with the initial point minus 1 minus 1 and use the x 1 coordinate as the first coordinate along which we do the optimization first. So, we find x 1 to be using this formula x 0 plus alpha 0 d 0 where alpha 0 is obtained by minimizing phi 0 alphas which is nothing but f of x 0 plus alpha d 0. So, if you plug in these values of x 0 and d 0. So, what we get is phi 0 alpha is the function of alpha and we will see that the minimum of this function occurs at alpha 0 equal to three-fourth by taking the derivative of this function derivative of this function with respect to alpha 2 0. And we get the point which is minus 1 by 4 minus 1.

(Refer Slide Time: 43:06)



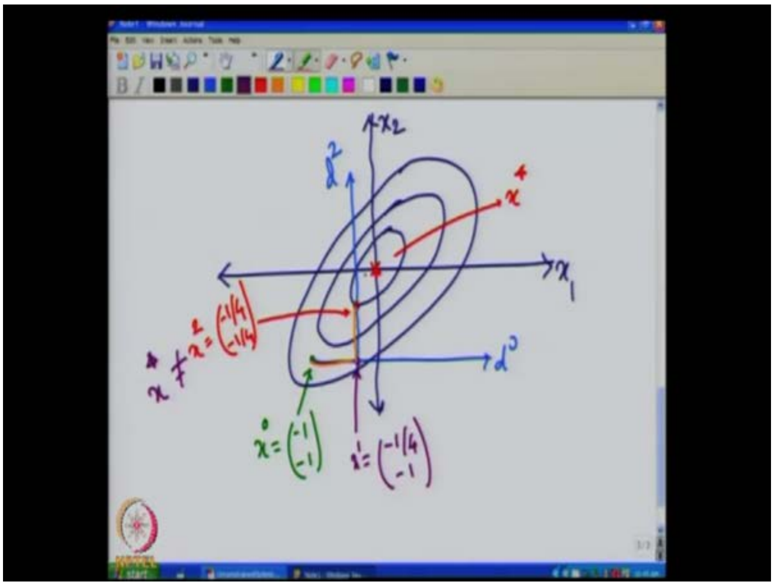
So, initial point suppose was somewhere here x 0 to be minus 1, minus 1 and when we moved along the direction. So, this was our d 0. So, when we moved here and do the exact line search we got a point which is minus one-fourth minus 1. So, the point is some were here, so let us denoted by x 1 to be minus one-fourth minus 1. So, this is the initial path that was followed by this. Now, let us go to the next coordinate direction which is 0 1.

(Refer Slide Time: 44:17)



And we know that x 2 is obtain using this formula where alpha 1 is obtained by solving another minimization problem where phi 1 alpha is define to be f of minus one-fourth alpha minus 1 and if you plug in this values in the original formula. So, we get this expression for phi 1 alpha. So, we take the derivative of this phi 1 alpha with respect to alpha set it to 0 and we get alpha 1 to be three-fourth and if you plug in this value of alpha 1 here then along with this value of x 1 and this value of d 1 what we get is x 2 and x 2 what we get is minus one- fourth minus, one-fourth which is not equal to x star.

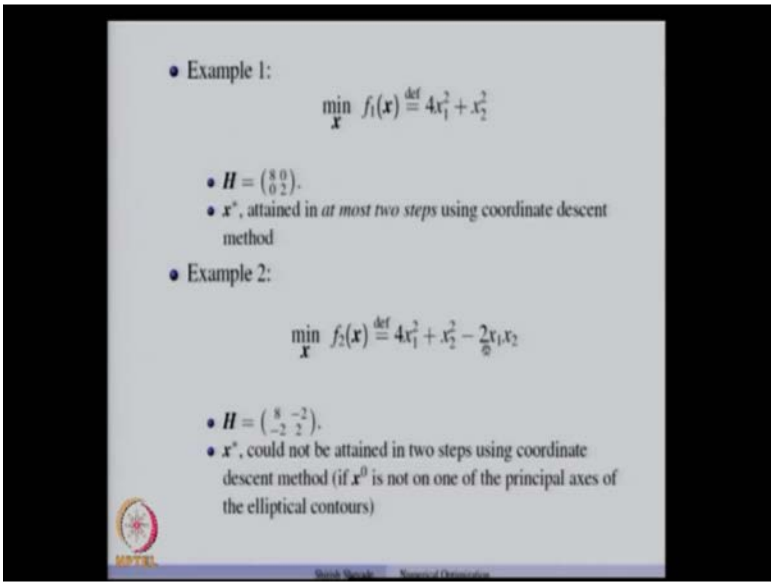
(Refer Slide Time: 45:18)



So, if you look at our. So, we use the direction d 2; d 2 is the direction here and we will see that we reach a point which is somewhere here and this point is x 2 that is minus one-fourth minus, one-fourth and we will see that we haven't reach the solution in two steps. So, the path test by this algorithm in the first two iteration is this and this and then you will see that we need some more iterations to reach the solution.

So, this x 2 is not equal to x star in this case. So, obviously will need more iterations to reach the solution x star in this case now we compare this with our previous function. So, you will see that in the previous function starting from 1 particular point we reach the solution in exactly two steps. Now, in the other case two steps were not enough to reach the solution if we use coordinate descent method. So, what is the problem? So, let us try to answer that.

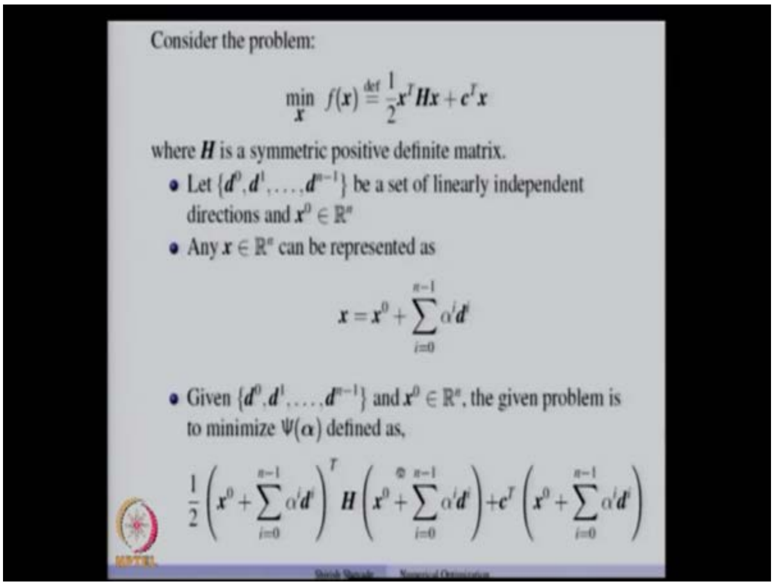
(Refer Slide Time: 47:05)



Take the function f x the first function. So, we call it f 1 x 4 x 1 square plus x 2 square and if you look at the hessian matrix of this function the hessian matrix is 8 and 2 along the diagonals and non-diagonal elements 0 and in this case the solution x star the attained at most two steps using coordinate descent method now if you look at the other function that we saw which is 4 x 1 square plus x 2 square minus 2 x 1 x 2 we will see that the hessian matrix here there are some off diagonal elements as well and these off diagonal elements are there because of this coupling between x 1 and x 2. So, such a term did not exist in the earlier case and therefore, the hessian matrix was nice diagonal matrix while here it is not a diagonal matrix and we saw that unless we choose x not to be 1 to be 1 one of the principle axes of the elliptical contours we cannot raise the solution in two steps if we use coordinate decent method. So, this term results in the non-diagonal hessian matrix and if we look at this two functions f 1 x and f 2 x and compare them you would see that it is easy to minimize x 1 keeping x 2 fixed here and similarly, minimize x 2 keeping the x 1 fixed because the terms involving x 1 and x 2 are separable.

So, here is a term which contains only x 1 here is a term which contains only x 2. So, we have separable objective functions separable in terms of x 1 and x 2 while here you will see that this term does not make the objective function separable in terms of x 1 and x 2 and it is this term which makes the hessian matrix also non-diagonal. So, the reparability of the objective function in terms of its variables is lost when we use when we have such a term which couples the different coordinates and that because of that we are not able to achieve the minimum in exactly two steps in this two-dimensional case if you use coordinate method.

(Refer Slide Time: 50:02)



So, let us consider a general function again a quadratic function which is half of x transposes h x plus c transpose x where h is a symmetric positive definite matrix now. So, remember that the hessian matrix h need not be diagonal. So, let us choose some n directions which are linearly independent and let us choose some initial point x 0 now any x in the inputs space which is n-dimensional space can be written using x 0 and linear combinations of this n independent directions because this form basis for this n-dimensional space as these vectors are linearly independent. So, they form a basis. So, any x in the n-dimensional space can be written as x 0 plus some linear combination of this stucks.

So, suppose x is written as x 0 plus sigma alpha i d i where i is varies from 0 to n minus 1 now let us rewrite this problem in terms of alpha. So, let us assume that d 0 to d n minus 1 are known quantities x 0 is also known quantity. So, d 0 to d n minus are known x 0 is known and what is unknown is alpha. So, let us rewrite this original problem by substituting x in this objective function by x 0 plus sigma alpha i d i where, i varies from 0 to n minus 1 and let us that problem will be problem where would the variables are alphas and not x. So, we are given d 0 to d n minus 1 and x 0 in r n we rewrite the original problem as minimize psi alpha where psi alpha is defined as this. So, you will see that every x in this original equation is replace by x 0 plus sigma alpha i d i now this is the quadratic function. So, h is constant. So, h does not depend on x. So, h remains as it is then c is the scalar that also does not depend on x. So, that remains as it is and we have a function psi alpha in terms of the variable alpha.

So, the original problem becomes the problem of minimizing psi alpha with respect to alpha note that there are n alphas. So, the number of variables here was n the number of variables in this new problem is also n and suppose we solve this problem with respect to alpha will get a solution alpha star and if we use that alpha star in this equation. So, if is find out x 0 plus sigma alpha star d i alpha i star d i then we get x star which is going to be the solution of this problem note also that we are working with the symmetric positive definite matrix. So, this function is strictly convex and therefore, they exist a strict local minimum and that local minimum x star is in this case obtained by solving another problem psi alpha with respect to alpha and getting alpha star now the reason for doing this will become clear when we expand this quantity take out the constant terms and make some assumptions on d's. So, we will do that in the next class.

Thank you.