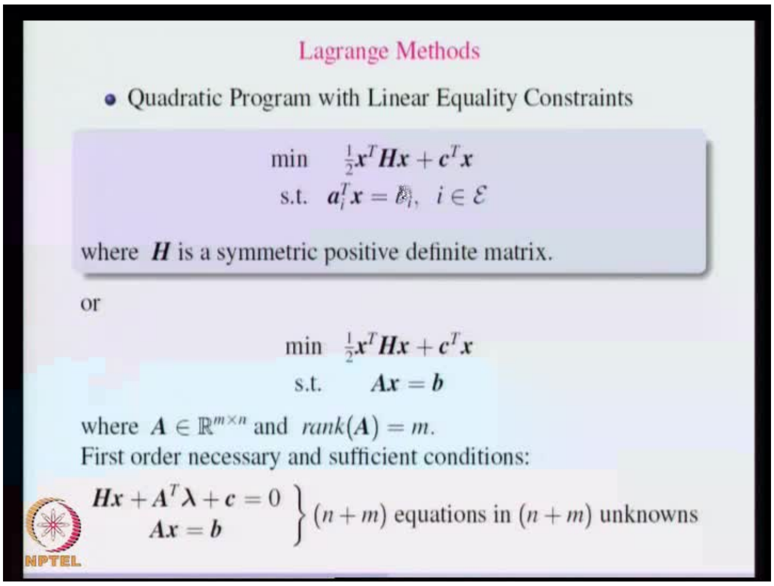
Numerical Optimization
Prof. Shirish K. Shevade
Department of Computer Science and Automation
Indian Institute of Technology, Bangalore

Lecture - 39
Active Set Method (Contd)

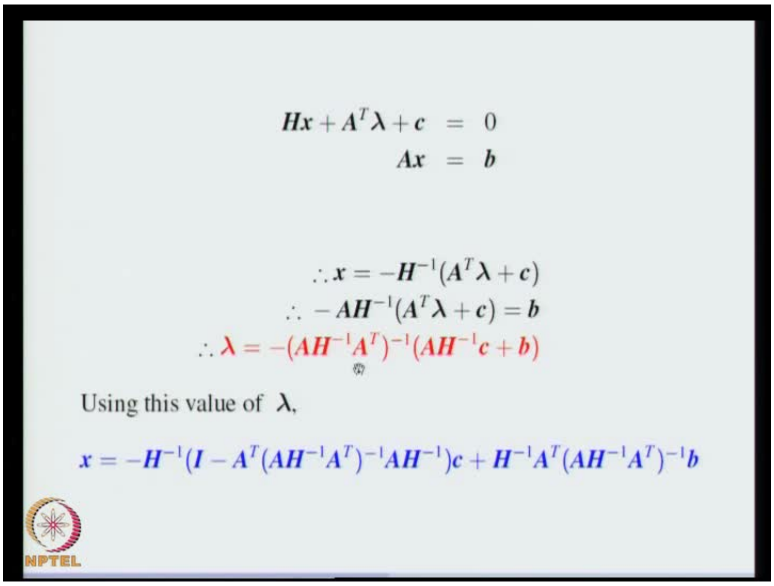
Welcome back; in the last class, we started discussing about algorithms, for constraint optimization problems. And in particular, we saw quadratic programs with linear equality constraints.

(Refer Slide Time: 00:35)



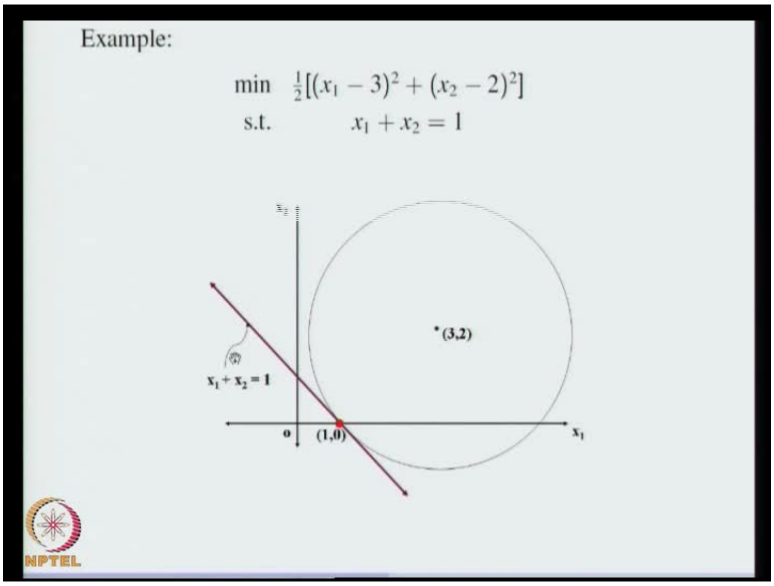
So, the for a given quadratic program, where the Hessian matrix is symmetric positive definite, at least which is the case in many of the practical problems. If we write down the Lagrangian, and set it is gradients with respect to x to 0, we get the first equation, H x plus A transpose lambda plus c equal to 0. And the feasibility of x will ensure with A x equal to b, and n and this case, we have n plus m equations, in n plus m unknowns and if in this case H is a inveritable.

(Refer Slide Time: 01:11)



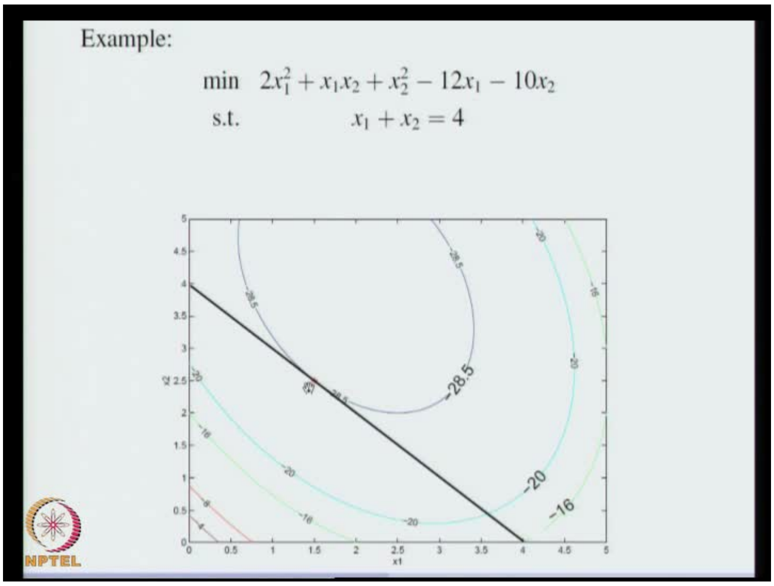
So, using that fact, we derived the solution for x to be this, with the corresponding Lagrangian multiplier lambda, obtained using this formula. Now, this being a equality constraint problem, the lambdas are unrestricted in sign.

(Refer Slide Time: 01:28)



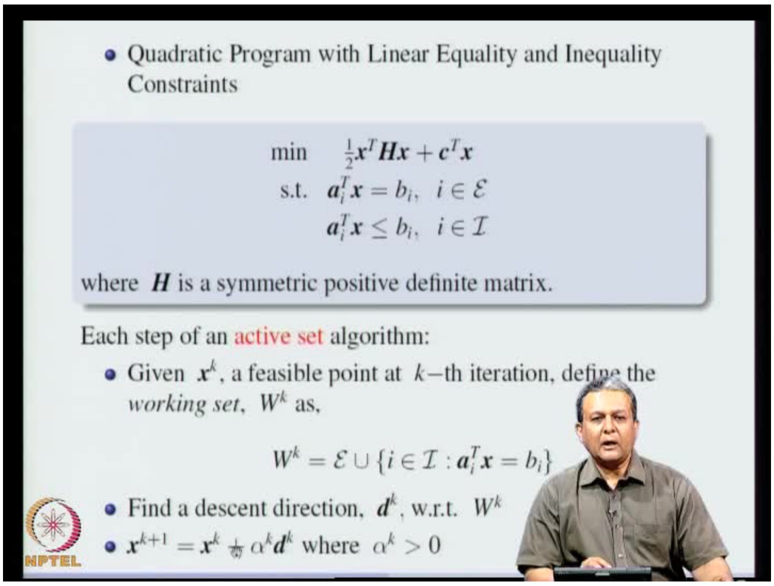
And then we saw a couple of examples, where this example, the problem is to find out a circle of minimum radius, which touches the line x 1 plus x 2 is equal to 1. And as you can see here, that the circle centered at (3, 2), will touch this line at (1, 0) and will have minimum radius, so this is the optimal x.

(Refer Slide Time: 01:59)



Now, we have similar problem, where we are still looking for the circles of minimum radius centered at (3, 2) which touch the line x 1 plus x 2 equal to 4, and as you can see from this figure that this is the solution. And similarly, we have another quadratic function, whose contours are elliptical, and we are interested in those contours which touch this line x 1 plus x 2 equal to 4, and the obviously the solution is this point.

(Refer Slide Time: 02:31)

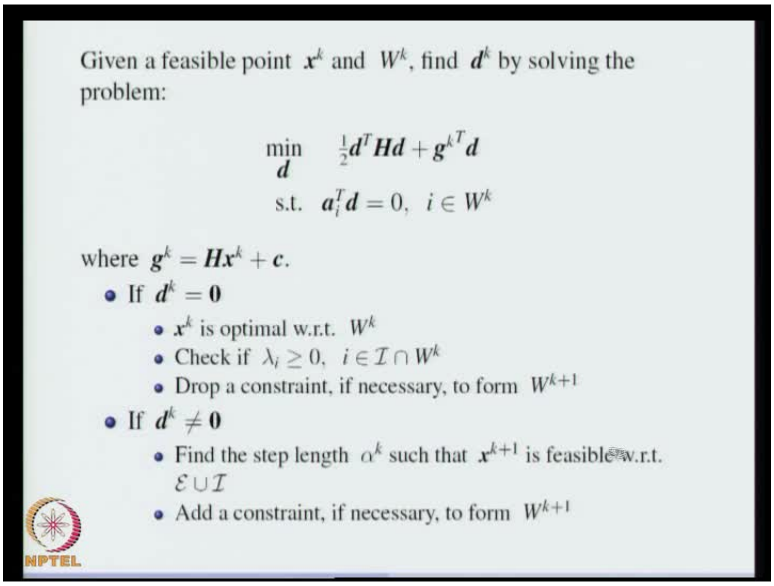


So, let us start looking at a general case, where we have a quadratic program with positive definite Hessian matrix. And there are some equality constraints, and some inequality constraints, so the equality constraints will be denoted by a i transpose x equal to b i, and a i inequality constraints will be denoted by a i transpose x less than or equal to b i. So, depending upon from which set the index i comes, they will be either equality or inequality constraints. Now, we want to device an algorithm to solve this problem, and the idea is that given a point which is feasible, find out the set of constraints which are active.

And once we have the set of constraints which are active, then solve the quadratic problem with respect to those active constraints. So, in that case, we simply have to consider the equality constraint problem; and once you find a solution of that problem we have to check whether that solution is in the feasible set or not. If it is in the feasible set, then at that point what is the new active set of constraints, and if it is not feasible then we have to bring the solution back to the feasible rear; and this process needs to be repeated, till we get the optimal solution.

So, this simple procedure is called active set method, and let us study this algorithm now, now if you given a x k which is a feasible point at the k th iteration of the algorithm. Let us define the working set W k to be the set of all equality constraints, and the set of all inequality constraints, which are satisfied with equality. So, the union of these two sets would determine the working set W k at the k th iteration, now I have determined W k the next step is to find the descent direction with respect to W k. And then we take a step along the distinct direction by determining an appropriate step length, which is alpha k which is a positive number.

(Refer Slide Time: 04:57)



Now, as we saw in the class, the associated with W k we need to solve the problem which is minimize half of d transpose H d plus g k transpose d, where g k is the Hessian I am sorry, g k is the gradient of the objective function at the given point x k; and this is with respect to the constraint that a i transpose d equal to 0. So, it is easy to derive this program using our original program, so if you substitute x by x k plus d here, and then to ensure the feasibility we need to make sure that, a i transpose d has to be 0, because a i transpose x k equal to b.

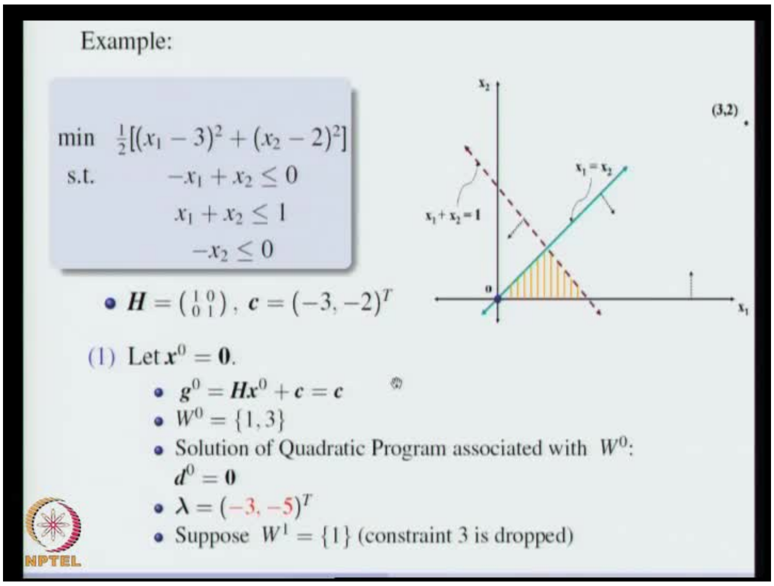
Remember that, the i's come from the set of active constraints or the working set at the iteration k, so this is very important that this set would consist of all the equality constraints; and some in equality constraints; also if it is possible to have some active inequality constraints at the given point x k. Now, if we solve this problem, and we get d k to be 0, then what it means is that, the current point x k is optimal with respect to W k that means that there is no need to move from x k as far as W k is concerned.

Remember that, this is not the optimal solution with respect to the entire original problem, but it is optimal only with respect to W k. Now, in that case we have to check whether the Lagrangian multipliers corresponding to the inequality constraints, which are part of the working, set W k, whether they are non negative, because as we seen in the set of a as we have seen when we discussed about the (( )) conditions for the problem, we saw that at optimality, the Lagrangian multipliers of the inequality constraints are non negative, at optimality.

So, we have to now check whether x k is optimal with respect to the entire original program and for that purpose we need to find out what are the inactive constraints which are part of W k, and whether the corresponding lambdas are non negative. Now, if that is the case, if lambda i is greater than or equal to 0 for all inactive constraints in the working set, then we have solve the problem completely for the original quadratic program.

Because, for the remaining in active constraints, which are not part of the working set the lambda is i's can be set to 0 and therefore, we have solve the problem. Now, if that is not the case, then what one have to do is that pick one lambda i which is non non-positive or which is negative, and drop that corresponding constraint from the set W k to form W k plus 1.

(Refer Slide Time: 09:24)



Now, if d k is not equal to 0 that means, we are able to find a direction d k along which the objective function can be improved. The next step is to determine the step length alpha k such that, x k plus 1 which is nothing but, x k plus alpha k d k is feasible with respect to all the equality constraints, and all the inequality constraints. So, remember that every point that we generate in this algorithm, has to be feasible with respect to all the constraints, and this is very important.

And by moving to x k plus 1, if we have identified a new constraint or new inequality constraint which has become active, then we need to add that constraint to the set W k plus 1, so these ideas will be more clear when we see some examples. For example, let us consider this problem which we have already seen, minimize half of x x 1 minus 3 square plus x 2 minus 2 square subject to the constraint minus x 1 plus x 2 less than or equal to 0; and x 1 plus x 2 less than or equal to 1, and minus 2 less than or equal to 0.

Now, on the right side you will see the constraint set, so this line corresponds to x 1 equal to x 2, and this line corresponds to x 1 plus x 2 equal to 1, and x 2 greater than or equal to 0, so which means the half space indicated by this arrow. So, we are looking at the intersection of half spaces indicated by this arrow, this arrow and this arrow corresponding to the three constraints. And this is the shaded region which is given here.

Now, what we want to find out is, to find out a point which is or a circle of minimum radius centered at (3, 2) which touches this feasible region. So, we are interested in finding out the circle of minimum radius which touches this feasible region, so let us see how to solve this. Now, if you look at the Hessian matrix of this objective function it is an identity matrix, and the c vector is shown here, remember that we have put this objective function in the form, half x transpose H x plus c transpose x; and in this case the hessian matrix is of symmetric positive definite matrix.

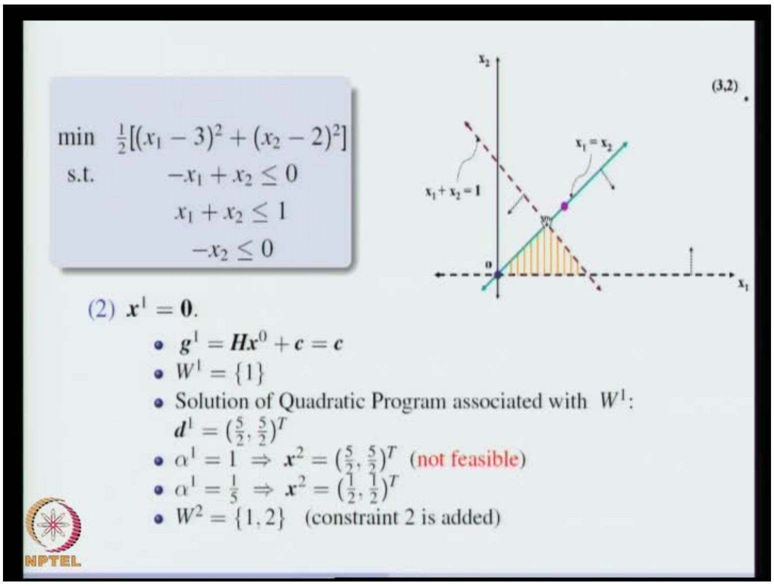
Now, let us start with an initial point which is the origin that is a feasible point, as I said that every at every iteration the point which is obtained has to be feasible with respect to all the constraints. Now, at this point we find out what is the gradient and since, x 0 is 0 the gradient is nothing but, c remember that this gradient will be needed, when we want to solve a problem to get d k. Because, in that case we have to solve a problem associated with the working set.

Now, add 0 this constraint is an active, the first constraint minus x 1 plus x 2 less than or equal to 0, and then third constraint minus x 2 or less than or equal to 0, they are active. So, the working set at 0th iteration is the first constraint and the third constraint. The second constraint which is in active is shown here by dotted line, now now if we solve the quadratic program associated with W 0, so W 0 is this working set. And if we solve that quadratic program that is a equality constrained quadratic program, and it is easy to solve, because we already found out away to solve a quadratic program with equality constraint, when the Hessian is symmetric positive definite matrix.

So, you can check that the solution to that quadratic program is d 0 equal to 0, so it is easy to see that, because if we take this constraint and this constraint, then this is the only point which is feasible as far as these two constraints are concern. And therefore, that has to be the minimum point and therefore, d 0 is equal to 0 and since, is d 0 is equal to 0; we will look for the corresponding value of lambdas. The Lagrangian multipliers of the inequality constraints, which are part of W k and, those lambdas using the close form solution that we saw earlier, those lambdas turnout to be minus 3 and minus 5 and both are negative.

And therefore, we may have to drop one of the existing active constraints to form W 1, so suppose we drop the constraint three, so constraint three is this constraint, and suppose we drop this constraint and retain only one constraint which is x 1 minus x 2 greater than or equal to 0. So, if you retain that constraint, now for the next iteration our point x 1 will be still the same, but now we will be working with W 1 to be containing only one constraint, and that is the first constraint.

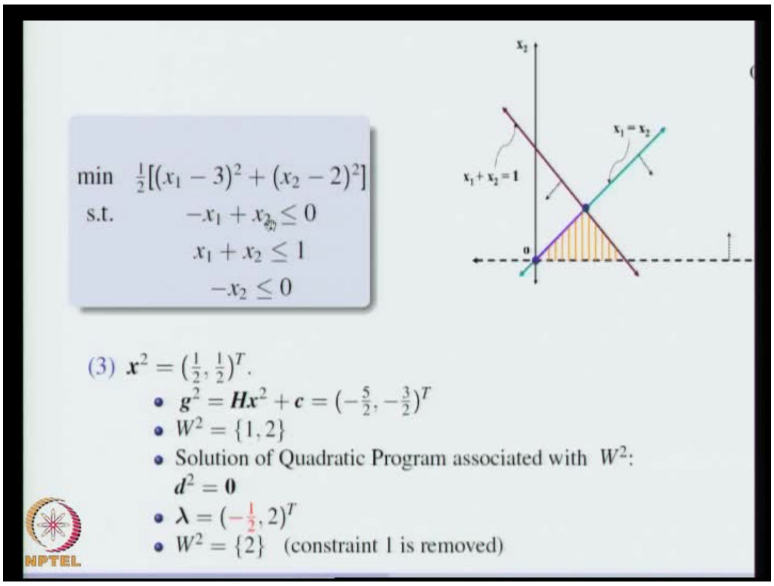
(Refer Slide Time: 14:28)



And this is shown here, as you see here that the third constraint is shown by dotted line, which means that, that is not a an active constraint. The second constraint anyway was not active constraint at this point, so the only active constraint that we are considering at this point is the constraint minus x 1 plus x 2 less than or equal to 0. So, our current point is still the same, and therefore the gradient at that point is also is still the same, but the working set is now only the first constraint. Now, with respect to that we solve the problem to get d 1, in turns that d 1 is nothing but, this direction where both the x 1 and x 2 components are 5 by 2, so it is this direction that we are considering.

Now, if you consider this direction, and if we set alpha 1 equal to 1 the step length, then what we get is x 2 to be 0 plus this quantity, which is 5 by 5, 5 by 2, so which is the point here. And clearly this point is not feasible point, because we have crossed the feasible region here, so in such a case we need to back track, so we need to move back along this the direction, till we have a point which is feasible. And that is obtained by back tracking or setting alpha 1 to be 1 by 5, in this case to get x 2 to be half, half which is the here, which are the intersection of these two constraints. Now, at this inter section which is, which has the coordinates half and half for x 1, and x 2, you will see that this constraint the second constraint as well has the first constraint are active; and therefore, the second constraint gets added to the working set.

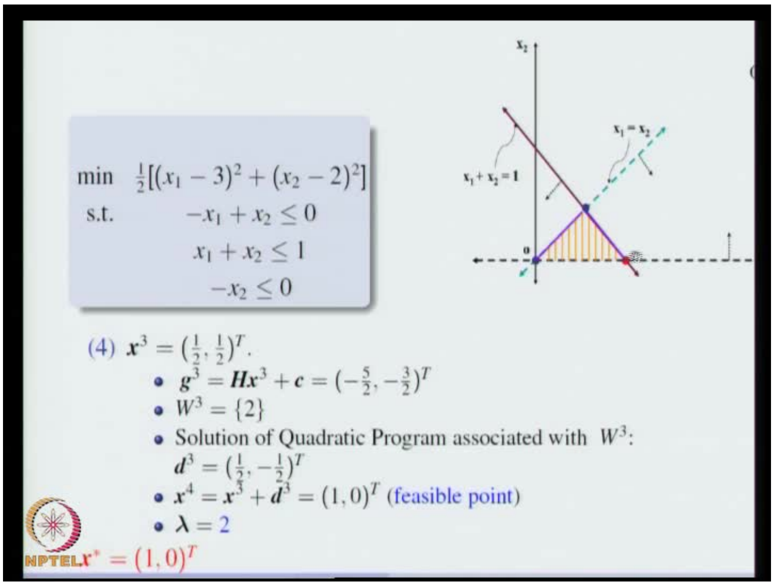
(Refer Slide Time: 16:40)



And here is the situation where we have the current point which is having x 1 and x 2 coordinates has half and half, and the two active constraints are the first two constraints, the third constraint is an inactive shown by the dotted line. Now, this is the working set, and again with respect to this working set, if we solve the problem to get d 2, we will see that d 2 is equal to 0. So, d 2 equal to 0 is obvious, because since in two dimensional space we have two non parallel lines, they intersect at a point and therefore, that is the only feasible point.

So, there is no direction that we can find to improve the objective function, so d 2 is equal to 0 and therefore, the next question to be asked is that which constraint can we drop, if this point is not optimal, if it is optimal then we will be done. So, now if you calculate lambda, we will see that lambda equal to minus half and 2, so the first constraint has negative Lagrangian multipliers. And the first constraint is a inequality constraint, so certainly this point is not an optimal point. And therefore, we drop the first constraint, and retain only the second constraint, and we are ready for the next iteration.

(Refer Slide Time: 18:13)



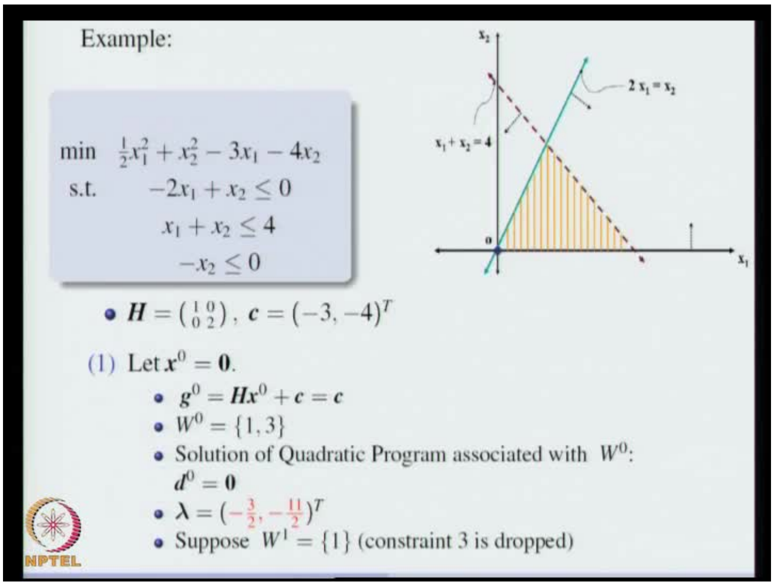
So, at the next iteration, we will see that at this point this is the only constraint which is active, this constraint is dropped, and this constraint is dropped from the working set. Now, what is the next direction d that we should take with respect to this working set, so has to improve the objective function. And it turns out that, that direction is this direction, and if we set alpha equal to 1, then x 3 plus d 3 will give us (1, 0) and that point x 4, and the point is shown here, is the point (1, 0) here, and that is the feasible point.

So, it satisfies all the three constraints, and it is a feasible point, so alpha equal to 1 is a good choice here, now we have to check whether the Lagrangian multipliers are positive.

Because we have only one constraint which is active here, we have to see whether the corresponding Lagrangian multiplier is non negative. And it is equal to 2 which is certainly, which certainly satisfies the condition that lambda should be non negative for active inequality constraints, and therefore this point is the optimal point.

So, if we trace the path that we obtained is in algorithm, you will see that we started from this point, and then moved on to this point, and then from this point we moved on to this point. And at this point, we found that the Lagrangian multipliers of the active inequality constraints are positive and therefore, we have got the minimum solution for this problem. Now, this was the case where we moved along the boundary, and got the solution, but this need not always be the case.

(Refer Slide Time: 20:31)

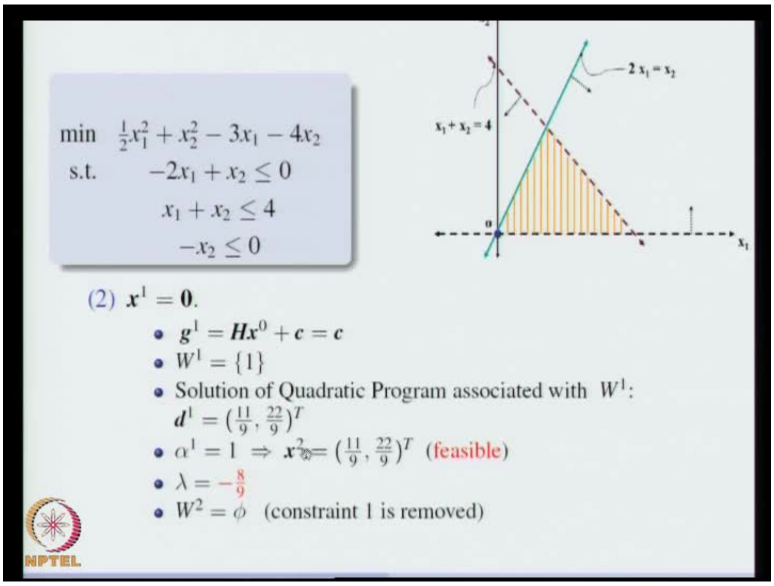


So, let us see another example, so here is another example where the quadratic, convex quadratic function is given subject to some linear inequality constraints. And if we draw the feasible region, so the constraint 2 x 1 equal to, 2 x 1 minus x 2 greater than or equal to 0 points in the half space shown by this arrow, the constraint x 1 plus x 2 less than or equal to 4 shows the half space given by this arrow. And x 2 greater than or equal to 0 gives the half space indicated by this arrow, so the intersection of this three half spaces is the shaded region which is shown here, and we want to minimize this objective function subject to this constraint.

Now, since 0 is the is one of the feasible points let us start with 0, now for this objective function the Hessian is symmetric positive definite, but not a identity matrix. Now, if we start with x 0 equal to 0 and and at this point, the two constraints which are active are the first constraint and the third constraint, the second constraint shown by the dotted line is inactive. So, as we saw earlier that if we now determine the direction to be moved, we should get t 0 to be 0, because that is the only feasible point as far has these two constraints are concerns.

And therefore, we would like to find out which constraint should be drop from this, and that is then finding out the Lagrangian multipliers. So, if we calculate the Lagrangian multipliers using the formula that we saw earlier, we will see that both the Lagrangian multipliers corresponding to this inequality constraint are negative. So, we choose the one with the least value to be removed, so the second and third constraint gets removed from the working set and therefore, now the working set has only one constraint, which is the first constraint, so the first constraint is here.

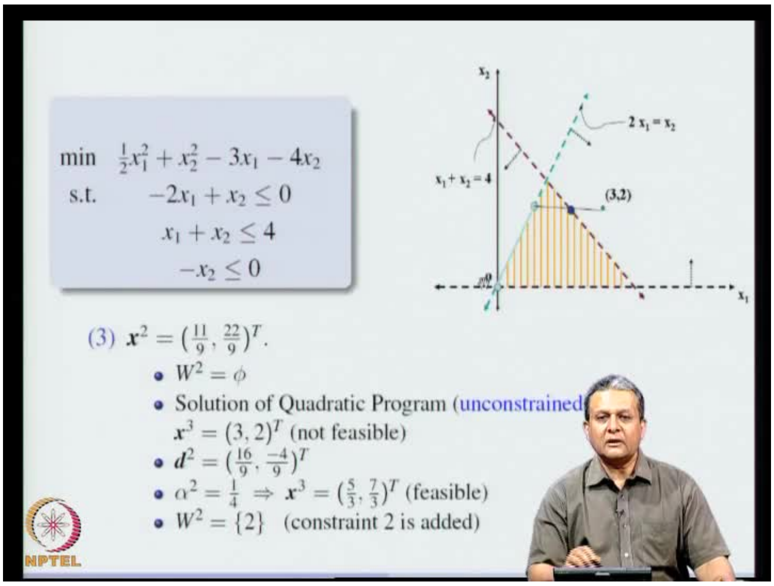
(Refer Slide Time: 22:55)



And this is shown here in this figure, where only the first constraint is active, the other two constraints are inactive as far has this current point is concerned. Now, we are still at the point 0, so x 1 is still 0, W 1 is 1 and then we solve the problem to get d 1 associated with the working set W 1. And that d 1 turns out to be this direction 11 by 9, 22 by 9, so this direction is given here.

Now, if we take alpha 1 equal to 1, we get x 2 which is this point, and that point one can check that that point is feasible; and the corresponding lambda if we calculate, the corresponding lambda is negative and therefore, we need to drop this constraint of this working set. And after removing that constraint, we get W 2 to be a null set, so now we will be at a point x 2 where the working set is a null set.

(Refer Slide Time: 24:00)



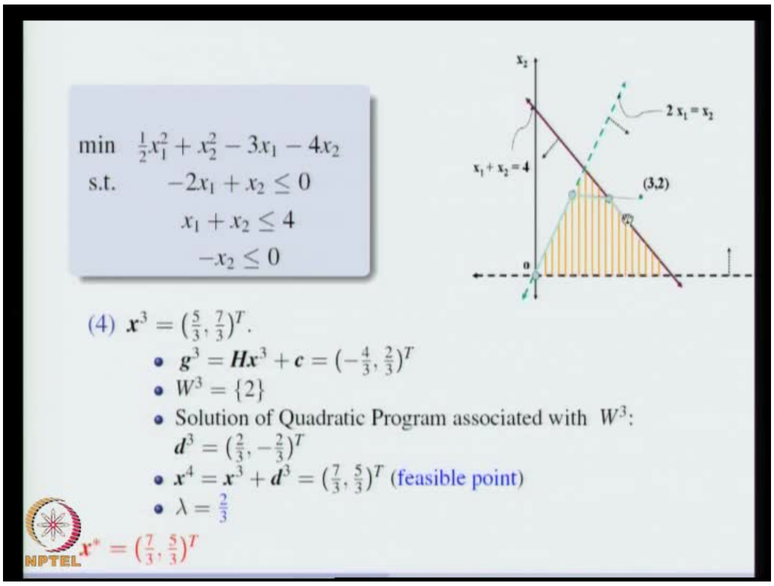
And this is shown here, where this is the current point that we have, where x 2 is a 11 by 9 and 22 by 9 which was obtained earlier, and the only although this point lies on this constraint. You can see that, because that constraint had negative Lagrangian multiplier and that was the only active constraint at that point, we removed that constraint from the working set, and therefore we are left with W 2 to be 5 the null set.

Now, now we solve an unconstraint optimization problem to get the solution of this, because there is no constraint in the working set, so solution of unconstraint quadratic program is a 0.32, whose x 1 coordinate is 3 and x 2 coordinate is 2. And that point is shown somewhere here, so which means that we have definitely crossed the feasible region, or moved away from the feasible region. Now, in moving to that point this direction was followed, and therefore we need to back track to come to a point, which is feasible as far has this constraint set is concerned.

And therefore, that back tracking gives alpha 2 to be 1 by 4, and therefore x is equal to this point which is having x 1 coordinates has 5 by 3, and x 2 coordinates has 7 by 3.

Now, at this point we will see that, this is the only active constraint which is x 1 plus x 2 equal to 4. So, therefore, that constraint is added to the working set, so now we have this point and now constraint two is added, so which means that the constraint two is now active.

(Refer Slide Time: 26:08)



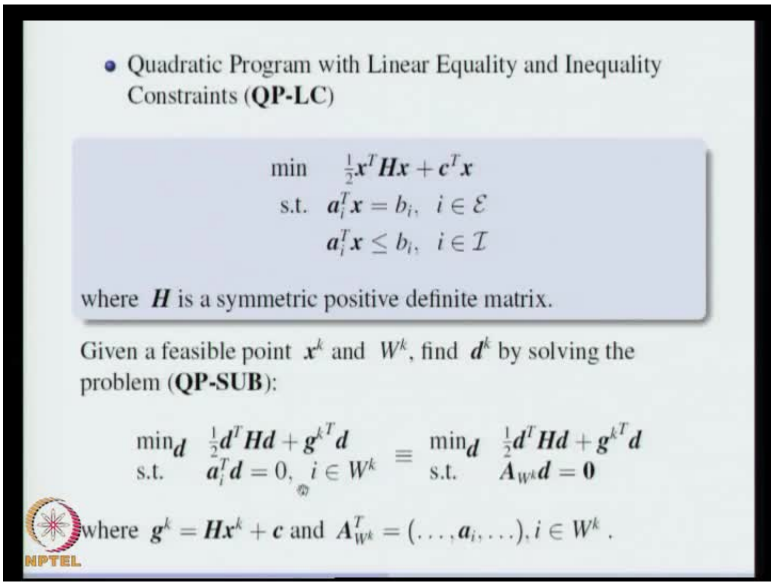
We now sitting at this point with this constraint active, and again we like to we would like to determine the direction d along which we can move, and that direction turns out to be this direction. Now, how far should we move, so if we set alpha 1 equal to 1, then x 1 is equal to x 3 plus d 3 that is equal to say 7 by 3, 5 by 3, so that is the feasible point, so that is shown here, and x star is equal to say 7 by 3, 5 by 3 in this case.

Now, if you look at the path that we followed, so we started from this point, then it came to this point, from this we reached the point (3, 2) which was outside the feasible regions, so we back tracked came back to this point which is one the feasible region. And then at this point only the second constraint was active, so we moved we found the direction d that direction turned out to be this direction, and took a step of unit length around this direction to come to this point.

And at this point, if we check the Lagrangian multiplier of this associated with this constraint, we found out that it is 2 by 3 which is positive, so certainly this point is the minimum point. So, so you will see that, the active set method at any iteration tries to find out the set of constraints which could be active, and solves the equality constraint problem with respect to those constraints. And this procedure is repeated till we get a solution, now since the number of possible active constraints is finite in number, this algorithm will definitely converge infinite number of iterations for convex functions.

Because every time we minimize the function where there is decrease in the objective function, and the number of combinations or number of possible active constraints at any iteration is finite in size. Therefore, the number of ways of selecting, the active constraints from the given set of constraints is also finite, though it is exponentially large in number, it is still a finite number. So, for quadratic functions, convex quadratic functions this algorithm can terminate in finite number of iterations.

(Refer Slide Time: 28:59)

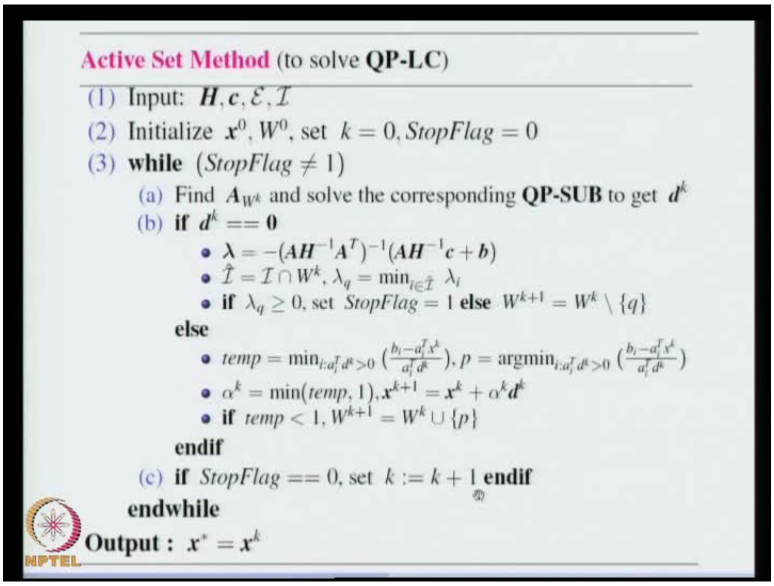


So, let us formally give the algorithm for a quadratic program with linear equality, and equal, inequality constraints, which are going to denote in short by QP-LC, quadratic program for linearly constraint problem. So, H is a symmetric positive definite matrix and E and I denote the set of equality and inequality constraints, and as I mentioned earlier that at every iteration we need to find out a working W k, and solve a problem to get the direction along which we need to move. So, given a feasible point x k, and the working set W k, the idea is to get d k by following this sub problem.

Now, as we saw earlier that, there is a term linear term in the objective function which involves the gradient of the objective function at k, and the Hessian is constant, so that is independent of k for a quadratic objective function. Now, a i transpose d is equal to 0 is a constraint where i belongs to the working set; now if we combine these a i's and put together in the form of a matrix a. So, this program will look like the objective function is still the same, but now we have a compact way of writing this active constraints, so A W k d equal to 0.

So, that the suffix W k means that, we pick a as based on those indexes in the working set, and put them in the form of the row of the a matrix or the columns of the a transpose matrix. So, this compact form of the sub problem, we are going to denote it has QP-sub, so at every iteration we need to find out what is the working set, and then solve QP-sub.

(Refer Slide Time: 31:18)



So, let us see the algorithm, this algorithm is called active set method, and the algorithm given, given here is solve of quadratic program with linear constraints. So, we are given the input matrix H and the c, and the equality and the inequality constraints, so they contain both a i's and b i's for the equality and inequality constraints. So, the first step is to initialize a point which is the feasible point, now this feasible point can be obtained by using phase one of simplex method, because we have the set of equality and inequality constraints.

Now, those can be written in the form of equality constraints by hiding some slake of surplus variables and artificial variables, an artificial linear program is solved to get an initial point x 0. The working set at that point is determined, the iteration contrive set to 0, and then the flag which denotes when to stop that is set to 0. Now, while that flag stop flag is not set to one, the first step of the algorithm is to get A W k, the a i’s corresponding to the working set, and then solve the corresponding sub problem QP-sup to get the direction d k along which we can move.

Now, if d k turns out to be 0, then we check whether the current point is optimal and for that purpose, we need to determine what are lambdas, so there is the close form expression to get this lambdas based on the a matrix. So, here A is nothing but, A W k but, to avoid notational letter we have not indicated the dependence of A on W k explicitly in this expression, but remember that this A is same has A W k.

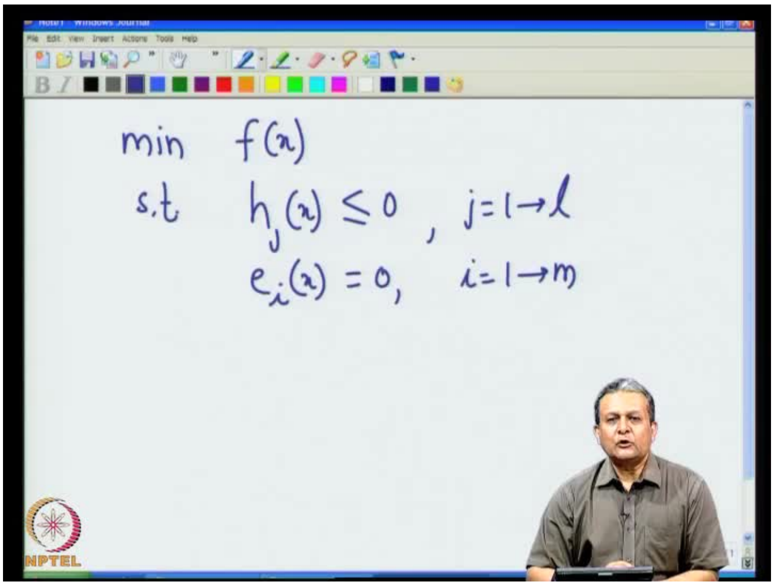
And then, we find out the minimum value of lambda among for all those inequality constraints which are active, or which are part of the working set. Now, if that minimum value is non-negative then we stop, so stop flag is said to one, otherwise the index q corresponding to which the lambda value was minimum that constraint is removed. So, the active inequality constraint for which the lambda was the least, and was negative gets removed from this working set.

Now, if d is not 0 that means we have got a directional d along which to move, now the next step is what is this to find out what is alpha k, and for that purpose we need to find out, whether the step size of 1 will cross the feasible region or not. So, if we come to this b i minus a i transpose x k by a i transpose d k or all those i’s such that, a i’s transpose d k greater than 0 that gives us a clue. Now, if that quantity is less than 1, so that means that we do not have to worry about over stepping, but if that quantity is 1 then certainly alpha k can be set to 1,

And which essentially means that we can take a full step of length 1 without worrying about the feasibility, and then x k plus 1 is set based on x k plus alpha k d k. Now, if this value was less than 1 or the alpha k value was less than 1, then that means that we are now at a point, where the new constraint needs to be added to W k, and that is then in this step.

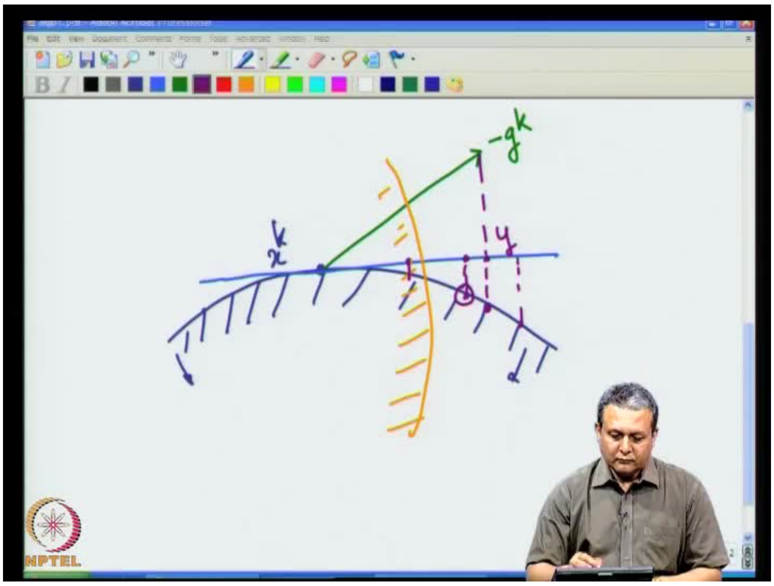
Now, having than this, if stop flag is still set to be 0, we increase the iteration counter and go back here. So, this procedure is repeated iteratively till stop flag becomes 1, and stop flag will become 1 when at a current point the Lagrangian multipliers of the active inequality constraints which are part of the working set W k are non negative. And that at that point we stop, and what we get is x star to be the value at the current iteration, so this active method and for quadratic programs, it is easy to show that this method coverts is infinite number of iterations because the number of possibilities of active constraints is finite in number, although it is exponentially large, now these ideas can be used to solve a general non-linear program.

(Refer Slide Time: 37:22)



Now, if we consider a general non-linear program minimize f x subject to h j x less than or equal to 0, such they are going from 1 to l, and e i x equal to 0, i going from 1 to m. Now, we have a general non-linear program, now the active set method can be used based on the approximation of f has a quadratic function. So, if we approximate f x quadratic function at a given point, and approximate this constraints, where affine constraints at a given point, then we have at a given point minimization of a quadratic function subject to linear equality and inequality constraints. And we have already seen the active set method to find a solution of this problem.

(Refer Slide Time: 38:46)



Now, one has to be careful about this approach, because there could be situations where one needs to take next of k r, so for example, suppose that we have an inequality constraint which is of this type. So, the shaded region denotes the, so this side is the feasible region, now suppose that this is the current point, so let us call this point has x k. Now, this is the constraint, we need to approximate this constraint by an affine constraint, so let us show the approximation of this constraint to be like this, now at x k let us assume that we have a direction which is minus g k, so negative of the (( )) direction for the function.

Now, so far we talked about the projection of this negative (( )) direction on the constraint set, but here we have to now talk about the projection of the (( )) direction on the affine approximation of the active constraints, at the current point. So, now if you project this we get a point, so let us call this point as y, now you will see that this y is not active, y is not part of the feasible set, because the constraint is violated at the point y.

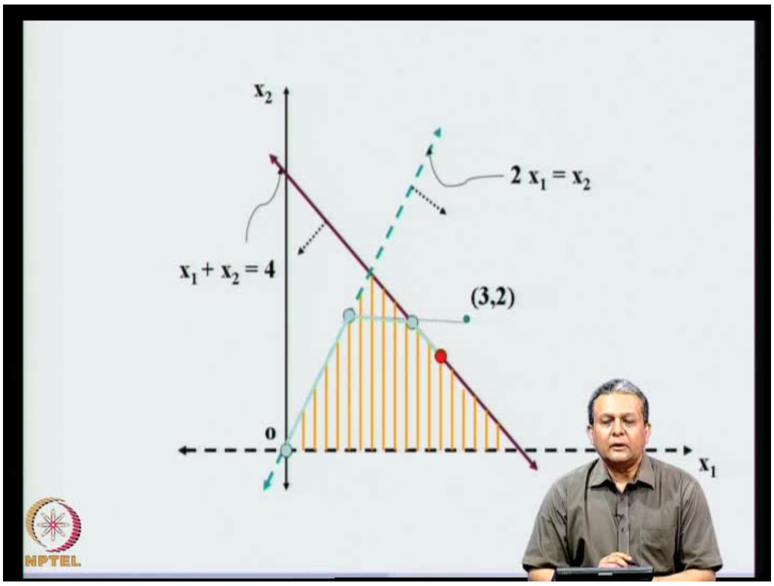
So, what we need to do is that, we need to find a point on this constraint set such that, at that point the all the decent conditions are satisfied like the objective function decreases. So, one way to do that is move along the, so if take a point x k, take a tangent plane to the constraint set at the point x k, and move along the direction which is perpendicular to that plane, so if you move further, so we may come back to a point. Now, at this point we have to check whether, the constraint is active which the case is certainly and whether the objective function has decreased sufficiently.

Now, if it has not decreased sufficiently, what we may have to do is that, we may have to move in the neighborhood of y. So, we may have to move along this direction, and then fin a point again in the direction perpendicular to a tangent plane, to see whether the objective function is decreased. Or we may have to go further, and then come to this constraint set and then, check whether the objective function has decreased. So, once you find a point, so suppose that this is a point where the objective function has decreased, then the next thing that one has to do is that, we have to find out whether the working set has changed or not at this point.

Now, it may so happen that we may have another constraint which is shown here, so we may have another constraint, now this point has satisfied the condition with respect to the original constraint. But as far has this new constraint is concerned, this point is violating that feasibility constraint, feasibility, so we again need to back track till we come to a point where the point somewhere here, where if you project it on to this, then we satisfy both the concern.

So, the method becomes more and more difficult, because the projection is on the tangent plane, and from there one has to move to the curve, and then at that point we need to check whether no new constraints are added. So, the procedure becomes more and more difficult, but there exists some other methods which could be useful for solving such problems.

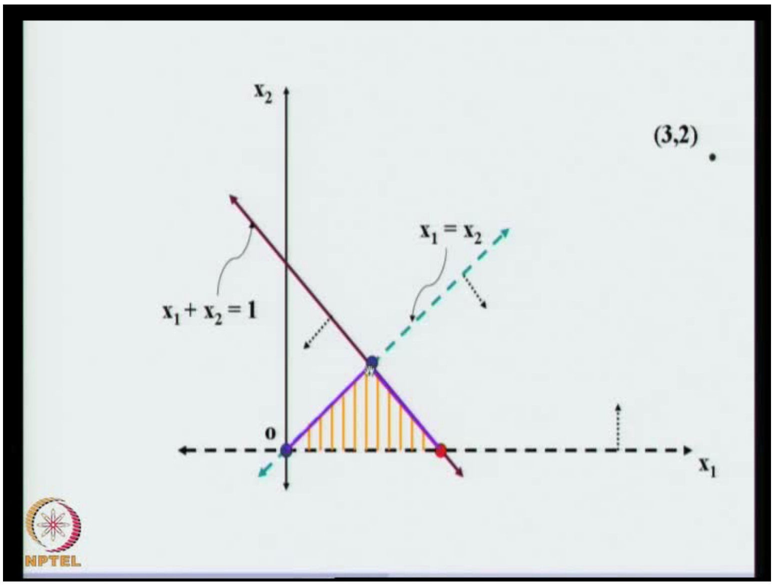
(Refer Slide Time: 43:54)



Now, another disadvantage of the active set method is that, one needs to find out a proper combination active constraints to get the solution. So, we started with this combination of active constraints, and then we have to move through another combination finally, to get a solution. So, the number of combinations of active constraints is though finite, it is exponentially large in number and therefore, finding the correct subset of active constraints from the given set of constraints is always a difficult task.

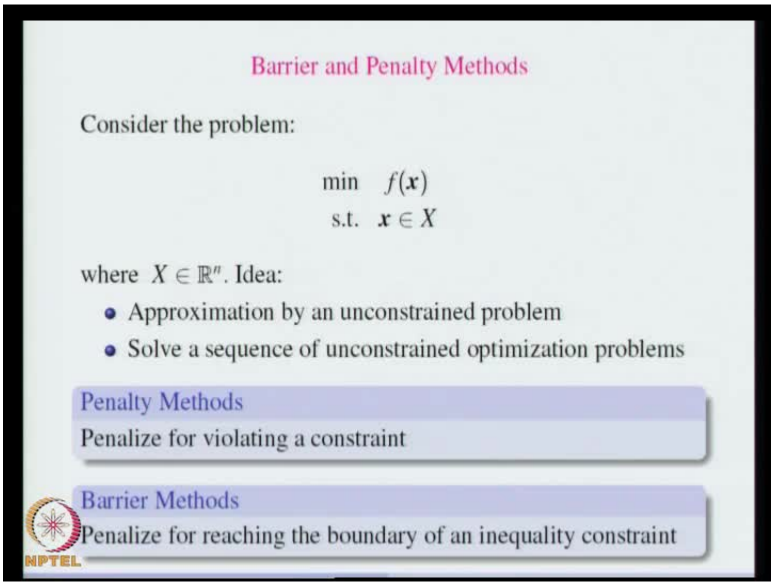
And second point that we need to note here is that, all the points which are generated always lie on or inside the feasible region that means, we are not allowed to deviate at any point of time from the feasible region. Even deviation which is very small is also not allowed, so we always have to remain in the feasible region, and that could be a disadvantage in some cases. So, it may so happen that we might have got solution by a visiting only a small number of points or in a shorter time, if we had not use d active set method.

(Refer Slide Time: 45:38)



So, there exist some methods which are not directly associated with the set of active constraints at any point of time, but they work in the entire N dimensional space and solve the original problem. Now, this was the path traced by the active set method, and we will see that one has to go to this vertex, and then to this vertex to reach this solution, but are there any better ways of solving this problem, so let us look at some of those methods.

(Refer Slide Time: 46:30)



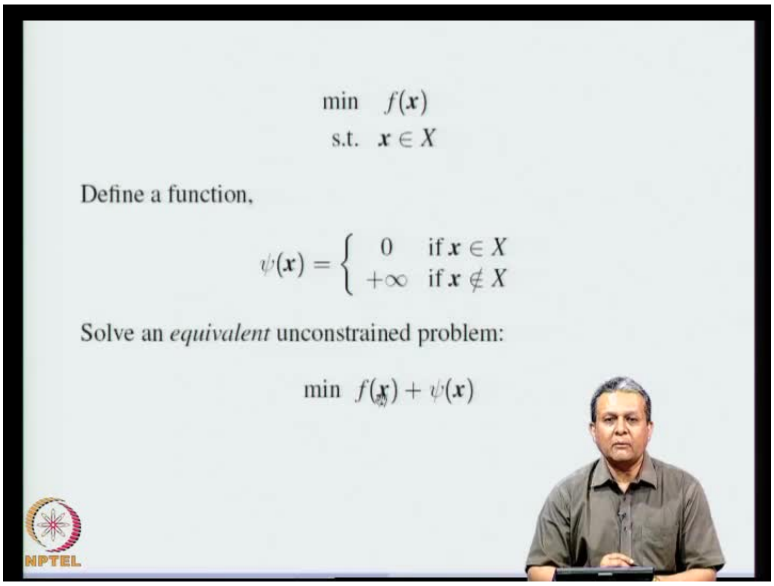
Now, those methods are called barrier and penalty methods, where we do not have to worry about finding the active set at any point of time, and these methods also find points has the iteration progress, which could be away from the feasible region. But when the algorithms converge one will form a point which is close to the optimal point, in the feasible region. So, let us consider a problem of this type to minimize f x subject to the constraint that x belongs to the set X, which is a subset of R n.

Now, the idea in this barrier or penalty methods is to approximate this constraint problem by an unconstraint problem, but then the approximation is not based on only one unconstraint problem but, they will be a series of unconstraint problems which will be solved. And the idea is that at every time, when we find a solution of an unconstraint problem, and take the sequence of such solutions which are formulated, then we will see that, that sequence of solutions will converge to the solution of this problem.

So, sequence of unconstraint problems is solved, and that will give us a sequence of solutions of this unconstraint problem, and that sequence is expected to converge to solution of this problem. So, the two methods differ in the way, they design this unconstraint problem, so the penalty methods as the name suggest they penalize for violating a constraint. So, a violation of a constraint is allowed, but there is a extra penalty or extra cost associated with this violation, and the penalty methods are based on that.

The barrier methods on the other hand, they do not let the feasible point move away from the boundary of the feasible region. So, one can think of it has putting the barrier around boundary, so that the feasible point generated are always in the interior of the feasible region. So, these methods can also be thought of has interior point methods for solving constraint optimization problems.

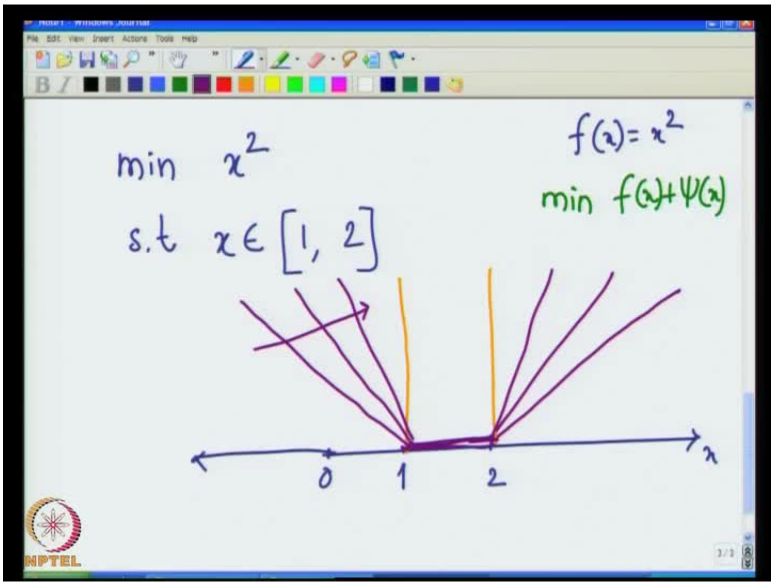
(Refer Slide Time: 49:38)



Now, if you look at this optimization problem, minimize f x subject to the constraint at x belongs to the set X, now suppose if we define a function phi x which is 0, in the space where x is belong to the set X, and it is plus infinity if x does not belong to the set X. And if we define such a function, and then define a new function to be f x plus phi x, now one can see that if we minimize f x plus phi x, this problem is equivalent to the original problem. The reason is that, if x star is a solution to this problem, then certainly x star belongs to X and since, x star belongs to X phi x star will be 0, and therefore we have got the minimum solution.

So, any solution of this problem will be same has the solution of this problem, because since we want to minimize a function the minimum of this occurs where x belongs to the set X. And once x belongs to the set X, then this term does not have any role to play in this minimization, and then it just amounts to minimization of f x, now, now if you think of this function, so let us consider a simple problem.

(Refer Slide Time: 51:27)

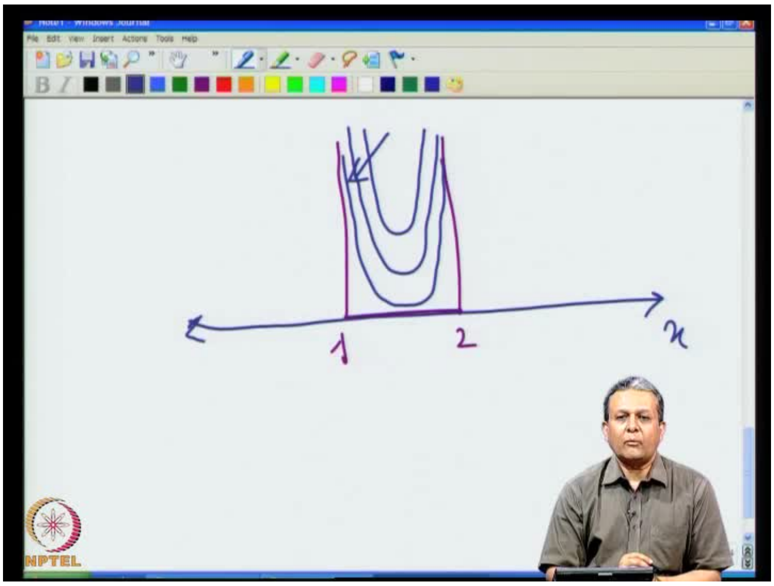


So, let us consider a problem, set to minimize x square subject to the constraint that x belongs to 1, 2. Now, on the real line, so this is the 1 and 2, now f x is equal to x square, now let us look that how the phi x would look like. So, if you consider phi x, phi x is 0 in this region, and goes to plus infinity in this region, now so if we minimize, first of all if you want to minimize this f x plus phi x, where phi x is plus infinity on, on the boundary of this feasible region. You will see that numerically it is not possible, because plus infinity cannot be indicated using a a number in a computer.

So, even if you replace plus infinity by larger, very large number still the problem is that this function is a discontinues, because it is 0 in the feasible region, but then goes to plus infinity, at the points which are outside this feasible region and therefore, this function is discontinues. So, since the function is discontinues that problem cannot be solved so easily, but suppose we design a function, so let us design a function which is, this is certainly not a good approximation of the function we are looking for.

What we are looking for is that, once we move away from the boundary the function the phi x function should go to infinity, now the original function that we saw was not discontinues was, was discontinues. So, therefore, we have designed a new function which is continuous, but which is not a good approximation, so suppose phi x phi x text the form like this, we might not be able to solve the actual original problem. So, what one can do is that generate a sequence of functions which are continuous, and which finally, can be good approximations of the function phi x. So, such functions if we start using them, so initially suppose if we use this function, then we use this function, in this direction finally, we will get a function which is very close to the function phi x that we are looking for. Now, this is one approach, this is called the penalty function approach.

(Refer Slide Time: 55:46)



Now, the other approach is the following that we have, now what we do is that, we use a function which is like this, and then keep on changing the shape of this function. So, that finally, it, it is the good approximation of the function that we are looking for. Now during this process you would see that we are finalizing heavily as we move towards the boundary. So if, if there is a way to control this functions, so that if we move along this direction, the function that we get are good approximations of the functions phi x that we are looking for.

So, these are called the barrier function methods, because they do not let the feasible point move towards the boundary. So, one starts with a point which is in the interior of the feasible region, and then one gets a solution, but the barrier functions will ensure that there is a extra or heavy penalty that one has to pay, while moving close to the boundary. So, these methods are called penalty functions or barrier function methods; and we will see more about them in the next class.

Thank you.