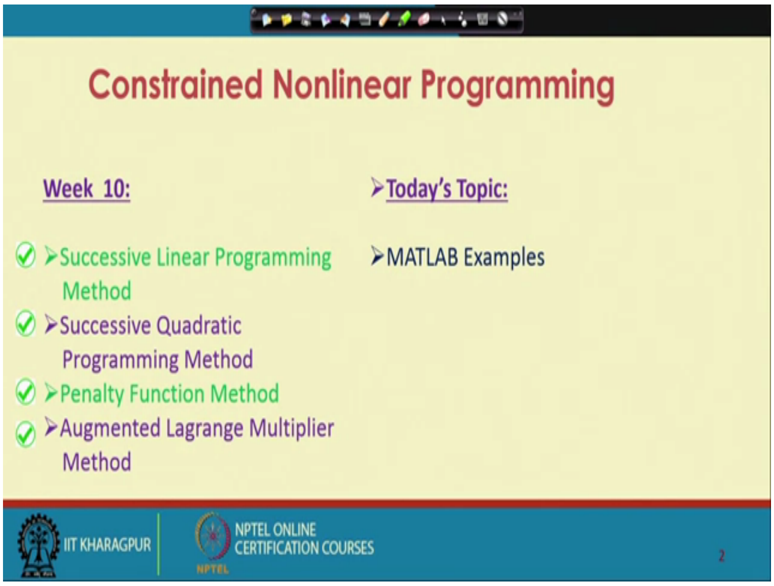
Optimization in Chemical Engineering
Prof. Debasis Sarkar
Department of Chemical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 50
Constrained Nonlinear Programming (Contd.)

Welcome to lecture 50. This is last lecture of week 10, in this week 10 we were talking about Constrained Nonlinear Programming. So, we have talked about 4 different methods: successive linearization, successive quadratic programming approximation, then penalty function method, and then augmented Lagrangian multiplier method. We have seen some of the examples to understand the working of these methods.

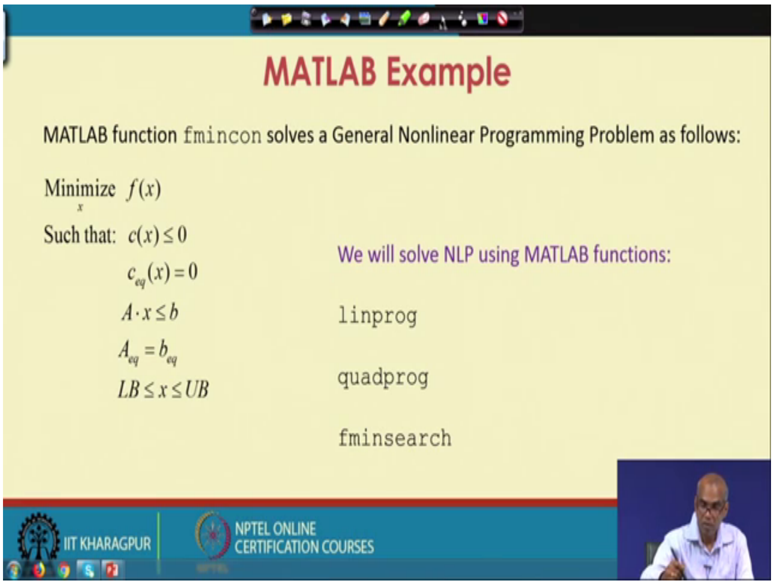
Now, in today's lecture we will go through some of those examples again and this time we will do at least 2 - 3 iterations using MATLAB inbuilt functions. So, this will help us to understand the working of these methods in a better way.

(Refer Slide Time: 01:32)



So, we will basically now talk about some MATLAB examples and we will see the working of Successive Linear Programming method, Successive Quadratic Programming Method and Penalty Function Method.

(Refer Slide Time: 01:53)



MATLAB has a solver known as fmincon. That can solve a General Nonlinear Programming Problem with constraints. And it solves a problem as shown here, minimize f x, where x may be an n vector such that c x less or equal to 0, c equality of x equal to 0, A x less or equal to b, A equality equal to b equality and x is bounded between lower bound and upper bound. Here, the objective function f x maybe nonlinear the c less or equal to 0 c of x less or equal to 0 maybe a nonlinear in equality constraint, c equality of x equal to 0 this may be a nonlinear equality constraint.

And then I have linear inequality constraint as A x less or equal to b linear equality constraint as A equality equal to b equality, A equality x into b equality. There should be x here, and x is bounded between lower bound and upper bound. Now MATLAB can very efficiently solve a General Nonlinear Programming Problem with constraints using fmincon function. So, you have to (Refer Time: 04:06) your problem in this form so, that you can solve the problem using the built in function fmincon.

Now we will see the working of fmincon in the next week. In this lecture we will see how MATLAB functions such as linprog, quadprog and fminsearch can be used to solve nonlinear programming problem using successive linearization method, successive quadratic programming approximation method and penalty function method, you are already familiar with linprog and fminsearch functions. So, today we learn one more new MATLAB function quadprog, which solve a quadratic programming problem.

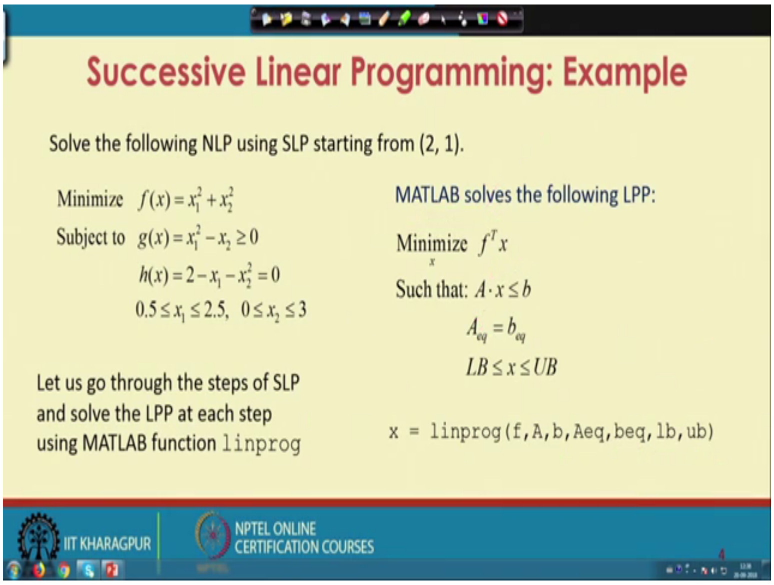
So, linprog you know solves a linear programming problem, now in case of successive linearization method we have seen that we convert a nonlinear programming problem into a sequence of linear programming problem. So, those linear programming problem I can solve using linprog and can solve the nonlinear programming problem using successive or sequential linear programming method.

Similarly quadprog solves a Quadratic Programming Problem, where the objective function is quadratic function and the constraints are linear constraints. That means, linear functions of decision variable. So, in case of quadratic or sequence successive quadratic approximation method we have seen that a general nonlinear programming problem will be converted to a sequence of quadratic programming problem. So, each quadratic programming problem can be solved by these MATLAB built in function quadprog. So, that way you will be able to understand the working of successive quadratic approximation method in a better way.

Similarly, we have seen that fminsearch can be used to solve and unconstrained optimization problem in fact, fminsearch implements simplex algorithm. Now the penalty function method transforms the nonlinear programming problem to an unconstrained optimization problem. So, that unconstraint optimization problem can be solved using fminsearch, the unconstraint optimization problem can also be solved using another built in MATLAB function known as fminunc.

So, you can either use fminsearch or you can also use fminunc. In the next week we will see that you can directly solve all these nonlinear programming problem very efficiently by use of the fmincon solver, but to understand the working of sequential linear programming problem; the working of sequential linear programming method, successive quadratic approximation method and a penalty function method. In today's lecture we will make use of these built in functions for solution of nonlinear programming problems.

(Refer Slide Time: 08:38)



So, let us start with Successive Linear Programming.

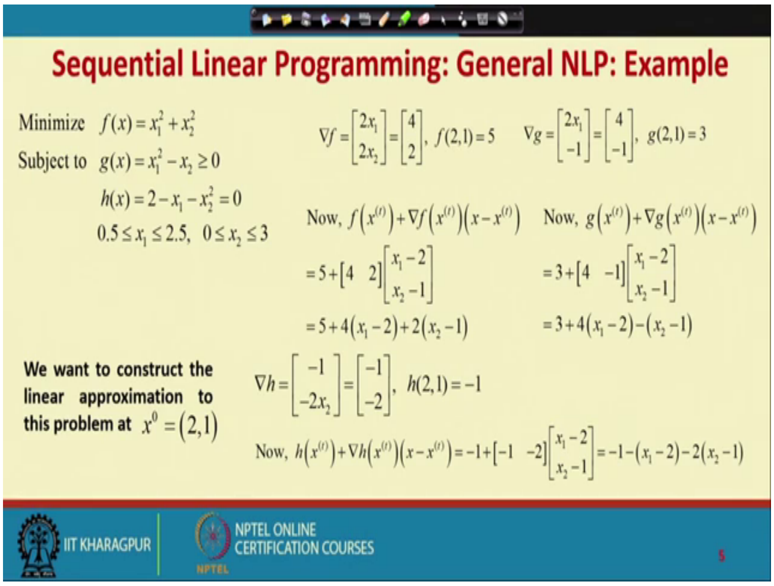
So, again you take the same example that we discussed before. So, I want to solve the given nonlinear programming problem using successive linear programming starting from the vector 2, 1; that means x 1 equal to 2 and x 2 equal to 1. So, the minimization of f x equal to x 1 square plus x 2 square subject to 2 constraints, one inequality constraints which is x 1 square minus x 2 square greater equal to 0. Another equality constraint which is 2 minus x 1 minus x 2 square equal to 0, x 1 is bounded between 0.5 and 2.5 and x 2 is bounded between 0 and 3.

Now, the true solution is basically for this problem is x 1 equal to 1 x 2 equal to 1. So, f x will be 2 does this satisfy constraint. Let us check this will be 1 minus 1 which is 0. So, this is satisfied this will be 2 minus 1 minus 1 again that is equal to 0, this is the solution. So, let us see whether you get this solution using successive linear programming method.

So, for successive linear programming method we have to solve a linear programming problem. We have already seen before that MATLAB solves a linear programming problem, such as Minimize f transpose x subject to x less or equal to b these are set of linear inequality constraint, A equality x should be x equal to b equality. So, A equality x is b equality. So, this is the set of equality constraints and you can have bounce on x, x is bounded between lower bound and upper bound.

So, the syntax to use the linprog function is this where f is this. That means, the constraint coefficients of the objective function, A is a matrix that will obtain from the inequality constraint, b is the right hand side vector for the in equality constraint. A equality is the matrix that will get for equality constraints, b quality is the right hand side vector for the equality constraints and the lower bound upper bound for the decision variables.

(Refer Slide Time: 12:16)

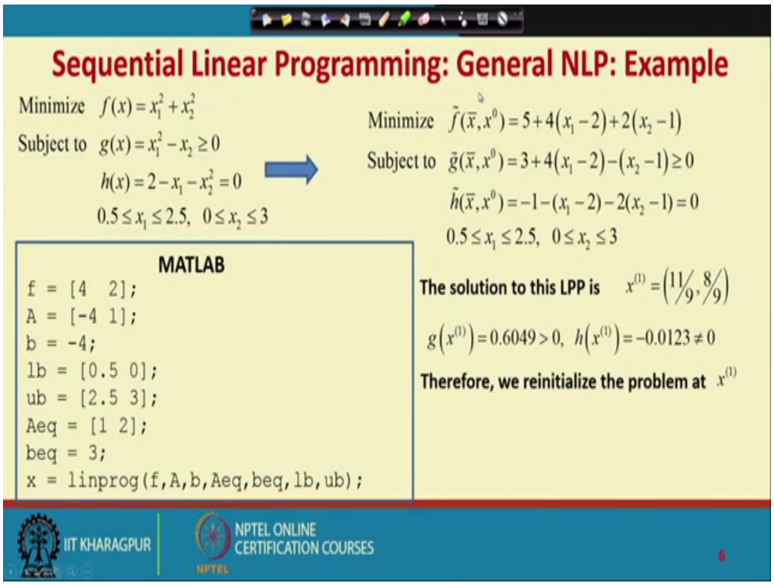


So, now we consider the given nonlinear programming problem. So, my starting solution is 2, 1. So, first thing you have to do is you want to construct the linear approximation of this nonlinear programming problem at x 0 equal to 2, 1. So, compute the gradient of the objective function evaluate it at point 2, 1 to get this, find out the gradient of the inequality constraint g and evaluate it at point 2, 1 to get this.

You can now linearize the objective function by Taylor series approximation around point 2 1 so, you get this expression. Similarly you linearize the in equality constraint around point 2, 1 by Taylor series expansion and you get this expression. Finally, you do the Taylor series expansion of the equality constraint around the given point 2, 1 and you obtain this expression as linearized equality constraint.

So, you find the gradients of the objective function, gradients of the constraints, evaluate those around the given point. So, using Taylor series expansion you will be able to obtain the linearized version of the nonlinear objective function and the constraints. So, these will help us to constitute the linearized problem for the given nonlinear programming problem.

(Refer Slide Time: 15:05)



So, that is what I now do here. So, I have this given nonlinear programming problem using Taylor series expansion as shown in the previous slide I obtain this linearized problem, now you see this problem is a linear programming problem. So, I can now solve using Simplex method and computation will be lengthy. So, I can make use of the MATLAB built in function linprog.

So, you have to then identify f, A, b etcetera, Aeq, beq etcetera from the given problem. Now let us look at the objective function is basically 5 plus 4 x 1 minus 8 plus 2 x 2 minus 2 so, some constraint plus 4 x 1 plus 2 x 2. So, minimization of some constraint plus 4 x 1 plus 2 x 2 and minimization of 4 x 1 plus 2 x 2 we will get the same solution optimal solution for x 1 and x 2.

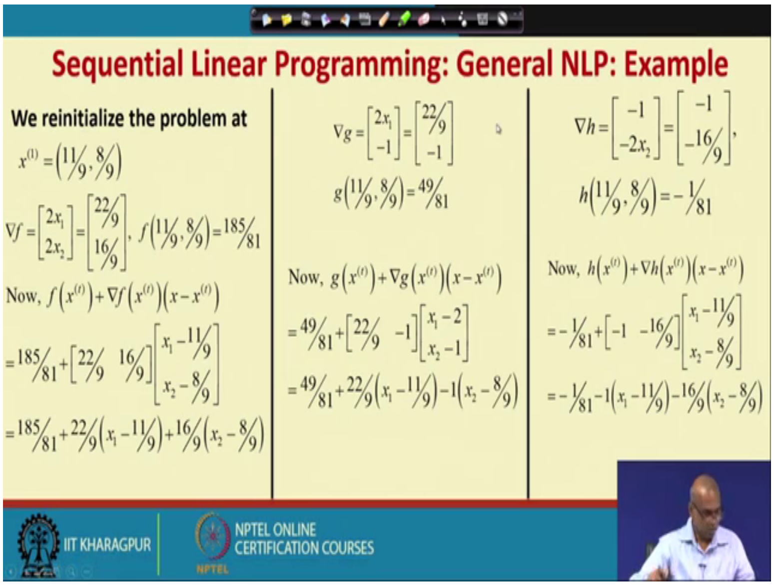
So, the f which involves the constraint cost coefficients in the objective functions are identified as 4 and 2. Note here you basically have 4 x 1 plus 2 x 2 and then 5 minus 8 is minus 3 and then minus 2 so, minus 3 minus 2 minus 5. So, this can be taken care of while evaluating the objective function, but the optimal solution of x 1 and x 2 for this and 4 x 1 plus 2 x 2 are same ok. So now, we have been able to identify f, what about A, A is for inequality constraint A x less or equal to b.

Now, you see this is greater equal to type. So, you have to convert it less or equal to type MATLAB considers the inequality linear inequality constraint as less or equal to type. So, multiply this equation throughout by minus 1. So, note minus 4 and this will have 1 as coefficients for x 1 and x 2 so, you obtain minus 4 and 1. So, b you will obtain after multiplication as minus 4, lower bound and upper bound are given 0.5, 2.5 and 0, 3. Note that, this is the lower bound for x 1, this is the lower bound for x 2, this is the upper bound for x 1 and this is the upper bound for x 2.

Now, equality constraint comes from here. So, there we have identified the coefficients and the b equality as 3. So, you solve this problem now linprog function and you get the solution as x 1 equal to 11 by 9, x 2 is equal to 8 by 9, evaluate the constraint g which is 0.6049 is greater than 0. So, that is satisfied, but h is minus 0.0123 it is not h 0.

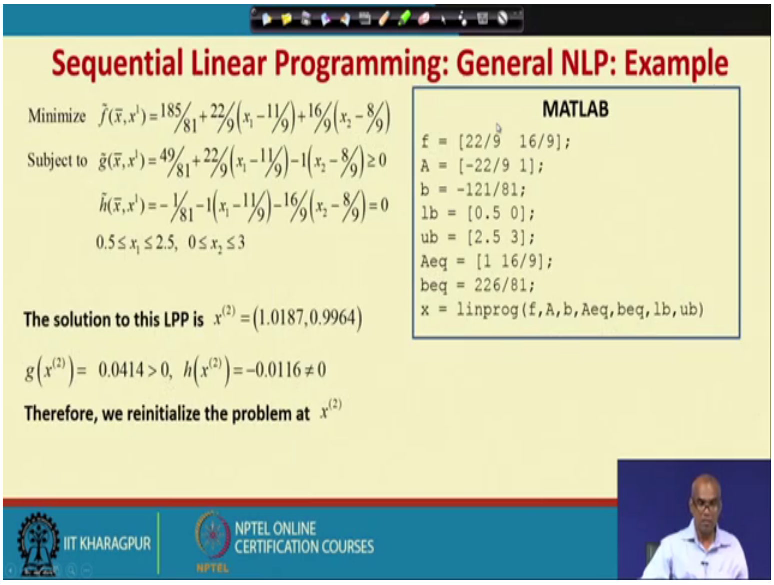
So, we want to reinitialize again the problem around these point; that means, you now get the linearized problem for the nonlinear problem around point x 1 equal to 11 by 9, x 2 equal to 8 by 9. So, do the same thing Taylor series expansion written the first order terms evaluate around 11 by 9, 8 by 9. So, the same step can be now repeated.

(Refer Slide Time: 20:20)



So, same steps can be repeated to obtain the linearized objective function, the linearized inequality constraint, the linearized equality constraint.

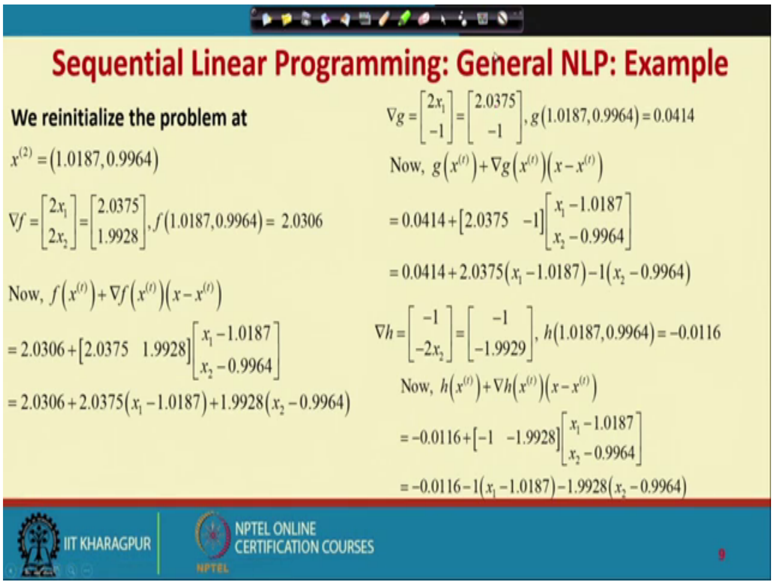
(Refer Slide Time: 20:45)



So, once you have done that we will be able to construct the linearized problem; that means, linear programming problem corresponding to the original nonlinear programming problem at point 11 by 9, 8 by 9. So, again you solve this problem using the built in function linprog identify f, A, b, A equality, b equality, etcetera and you obtain the solution as x 1 equal to 1.0187, x 2 equal to 0.9964 look at this we are very close to solution true solution 11.

Does the solution satisfy the constraints, inequality constraint satisfied, because the values greater than 0 equality constraint is still not satisfied 0.0116, but have you improved previous constraint violation was 0.0123; now it is 0.0116. So, slightly improve on that as well. So, now again we reinitialize around this given point so, again repeat the same.

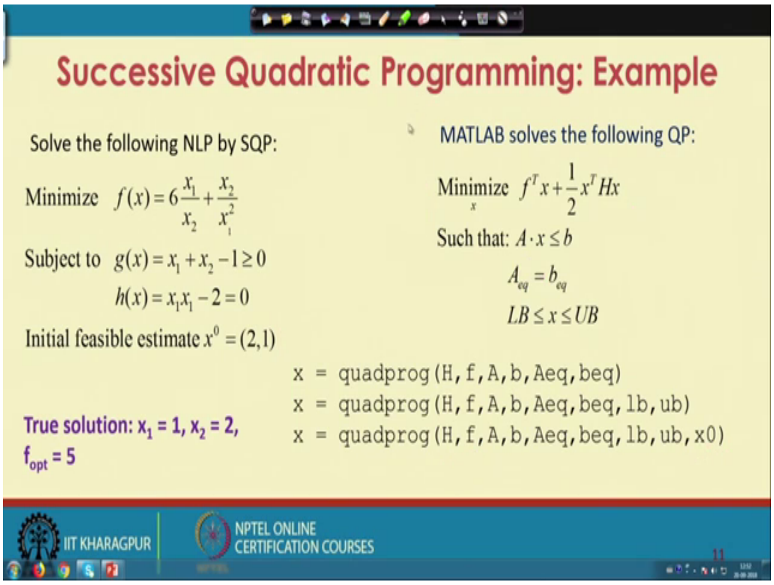
(Refer Slide Time: 22:18)



So, identify the linearized objective function, identify the linearized inequality constraint, identify the linearized equality constraint. So, obtain the linear programming problem for this stage again solve using linear programming built in function linprog identify f, A, b, etcetera and this time you obtain the solution as 1.0001, 0.9999. The inequality constraint is satisfied, the equality constraint I get as 1 into 10 to the power minus 4. So, extremely small value we consider it to be very close to 0.

So, you can stop here and you see that we have almost got that true optimal solution which is 1 and 1. So, this is how we can make use of linprog to understand the working of the successive or sequential linear programming method.

(Refer Slide Time: 23:42)

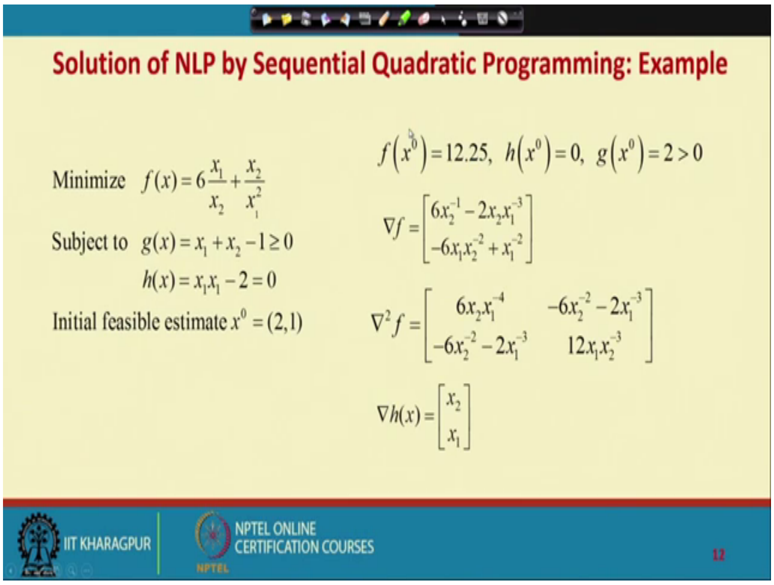


So, next let us talk about Successive Quadratic Programming method. So, you are wish to solve the again the same problem that you have solved before, you have the objective function nonlinear, you have one inequality constraint which is basically linear and one equality constraint which is nonlinear. So, what we will use is quadprog for this solution?

So, the MATLAB solves a quadratic programming problem of the form as shown minimize f transpose x plus half x transpose Hx. Note that a quadratic function can be expressed as f transpose x plus half x transpose Hx. So, subject to x less or equal to b they are linear inequality constraint and A equality x, A equality x equal to b equality as linear equality constraint. So, these are the syntax we need to identify H, f, A, b, A equality, b equality, lower bound, upper bound you can also give a starting initial vector as a guess solution.

So, during successive quadratic programming problem during successive quadratic programming method, we have seen that the given nonlinear programming problem will be successively approximated as a quadratic programming problem. So, that quadratic programming problem can be solved using quadprog built in function and this is what we do in this example.

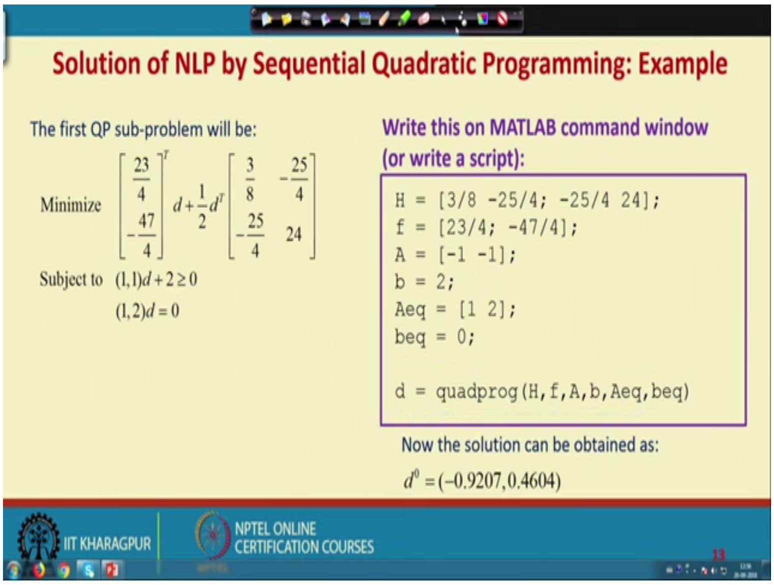
(Refer Slide Time: 25:59)



So, first identify the gradient hessian gradient of the objective function sorry gradient of the objective function gradient of the constraints, gradient of the inequality constraint actually do not have to linearize it, because the inequality constraint given as x 1 plus x 2 minus 1 greater equal to 0 is already linear. So, we obtain the objective function gradient, the hessian of the objective function and the gradient of the equality constraint the equality constraint a h is x 1 x 2.

So, equality constraint is x 1 x 2 minus 2 equal to 0. So, you get del h del x 1 as x 2 and del h del x 2 as x 1. So, this may be corrected. So, instead of x 1 x 1 this is basically x 1 x 2 minus 2 equal to 0, this is Hx. So, once you have identified the gradient and the hessian will be able to now cause the problem in the format of the MATLAB.

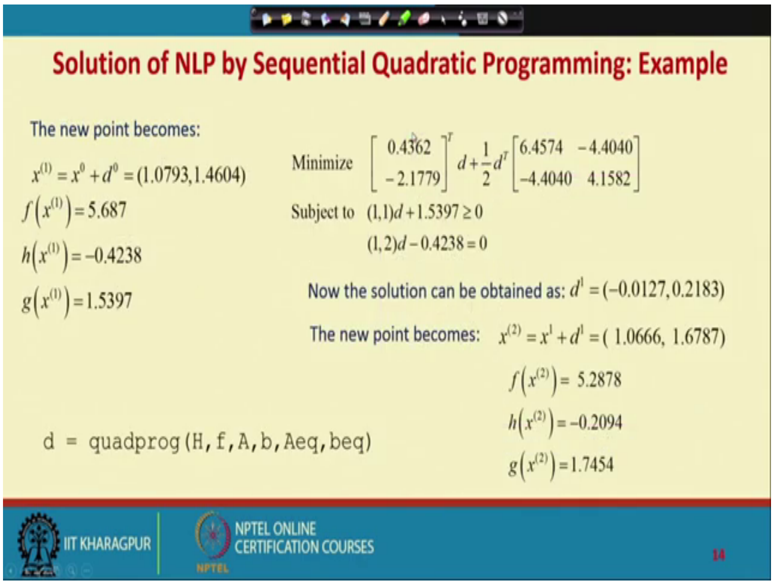
(Refer Slide Time: 28:00)



So, this is the first quadratic programming sub- problem note that this is that f transpose x is d and this is H. So, we have d here and these are 2 constraints, x less or equal to b and A equality x equal to b equality. So, the objective function is f transpose d plus half d transpose d, this is A x less or equal to b and this A equality x equal to b equality. So now, you have to identify the f, A, b, A equality, b equality and H is nothing but this matrix see 3 by 8 minus 25 by 4, minus 25 by 4 and 24.

So, this is the hessian. So, now, you use the MATLAB's built in function quadprog to solve these quadratic programming problem, and you get this solution for d which is my d 0.

(Refer Slide Time: 30:09)

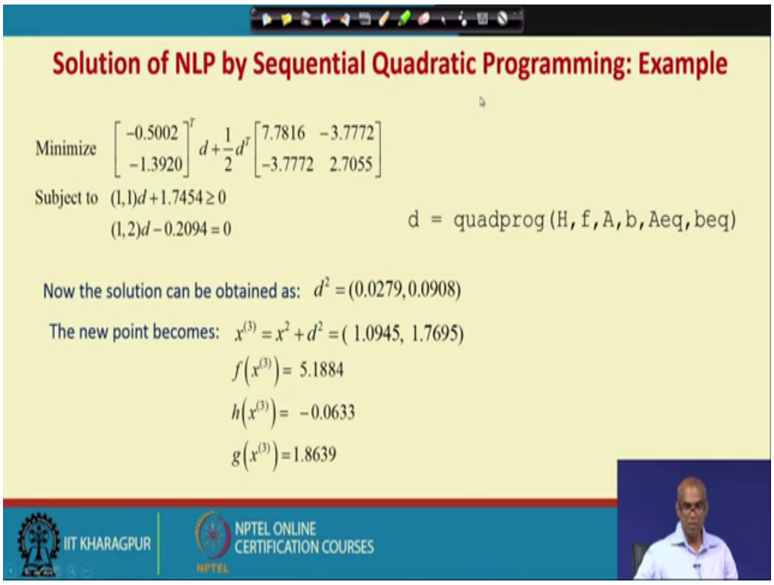


So, then the new point will become x 0 plus d 0 which is obtained as this. So, f of x f of x 1 is 5.687 and h of x 1 is minus 0.4238, g of x 1 is 1.5397 equality constraint is not satisfied will reinitialized again.

So, around this point we again reinitialize and obtain the second quadratic programming sub problem. Again we solved using quadprog and you obtain these as solution for d 1. So, the next solution is obtained as x 1 plus d 1 and you obtain as 1.0666, 1.6787 for x 1 and x 2 respectively. So, the function value is 5.2878 equality constraint minus 0.2094 and g is 1.7454.

So, equality constraint is still not satisfied we have to go on repeating this, but you can note that that true optimal solution for this problem is x 1 equal to 1, x 2 equal to 2; and the objective function optimal value of the objective function is 5. So, we have already got something close to 5, but x 1 is something closer to 1, x 2 is still not. So, further improvement is required and you have to repeat the step.

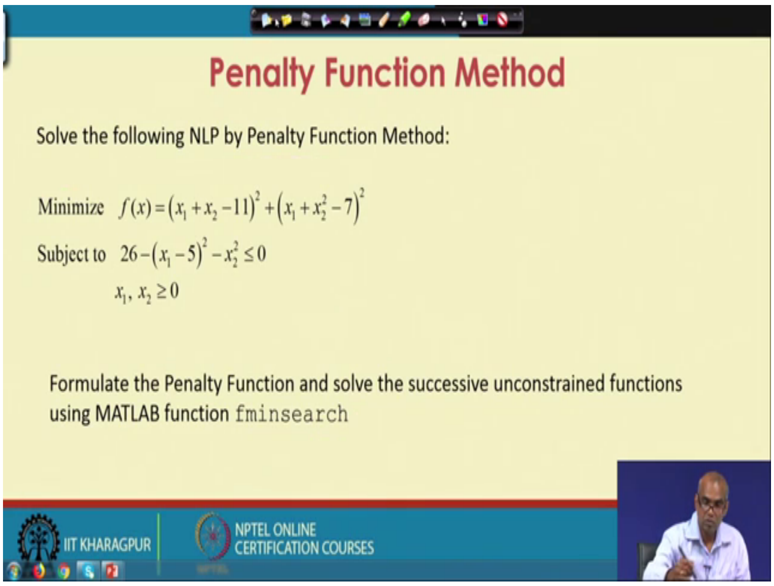
(Refer Slide Time: 32:22)



So, the third quadratic programming sub problem again is obtained by the same method and again you use quadprog to solve it. So, this time you obtain the solution as 1.0945, 1.7695. So, the function value now is 5.1884 it has become closer to 5 now. So, we are going close to the optimal solution 1 1, but it will take few more iterations. So, this is how we can solve nonlinear programming problem using sequential or successive quadratic programming method.

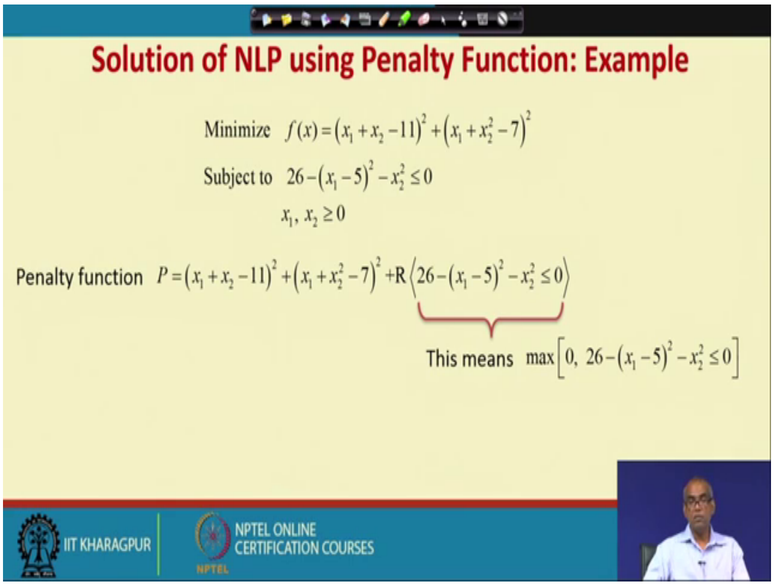
So, finally, let us talk about the Penalty Function Method.

(Refer Slide Time: 33:14)



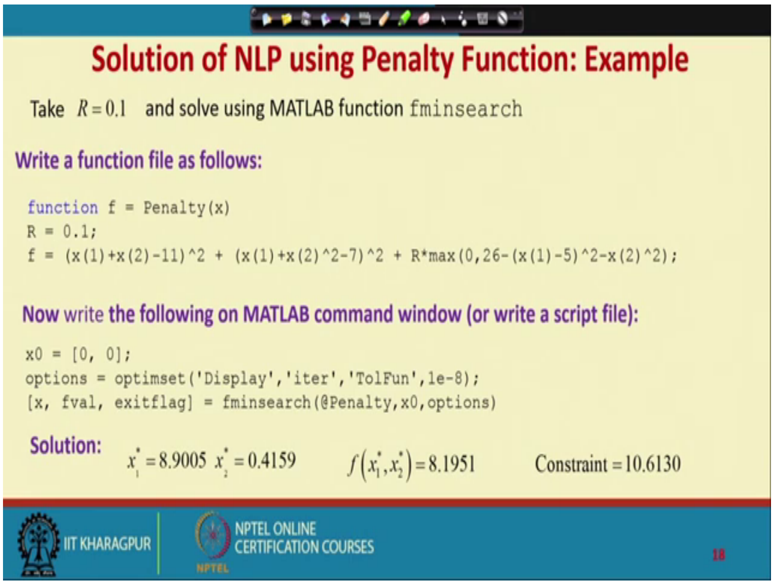
So, let us solve this (Refer Time: 33:23) Law function subject to a single inequality constraint by Penalty Function Method. So, you formulate the penalty function method, we formulate the penalty function and solve the penalty function or the unconstrained function by successively using MATLAB built in function fminsearch. So, given the nonlinear programming problem which is here (Refer Time: 33:58)Law function and one inequality constraint. We formulate the penalty function first and then successively solve using fminsearch solver of the MATLAB which solves unconstraint functions using Simplex method.

(Refer Slide Time: 34:23)



So, penalty function can be easily constructed by adding the penalty term to the objective function. So, this is the objective function, and this is the penalty term, we have seen that this is the bracket operator, and this means that you have to take maximum of 0 or this. So, this you can ignore this part. So now, let us see how you solve it.

(Refer Slide Time: 35:14)

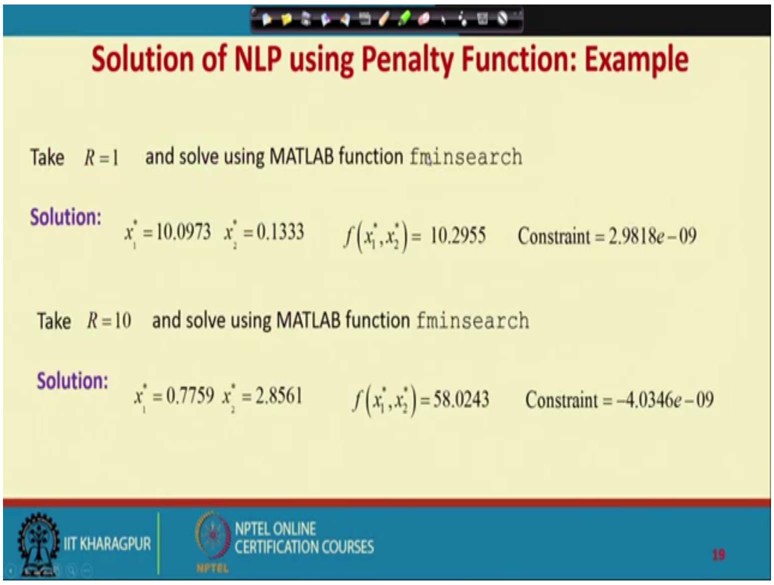


So, let us first take R equal to 0.1 and you write the penalty function. So, you just have to solve this penalty function using fminsearch of MATLAB.

So, write a function file where you define this penalty function. And then you have to call fminsearch either from MATLAB command window or you can write another file where you will write these commands and then run that file. So, write start with R equal to 0.1 then define the penalty function and then either I write these statements on command window or create another file and then run that file.

So, this is the initial case you are giving you can give these options to tell the MATLAB that you want to display each iterations and the tolerance you can specify. So, if you use the fminsearch you get the solution as x 1 equal to 8.9005, x 2 equal to 0.4159 and the function value is 8.1951 the constraint if you evaluate is 10.6130.

(Refer Slide Time: 37:07)



Next we take R equal to 1 we increase the value of the penalty parameter again repeat the same step. So, in the program that you have written all you have to do is you just have to change these line it was R equal to 0.1 just you have to right now R equal to 1 and then run the program to solve the problem or solve the unconstraint function for penalty parameter value R equal to 1.

So, this time you obtain the solution as x 1 equal to 10.0973, x 2 equal to 0.1333, the function value is 10.2955 now you see the constraint has reduced 2.9818 into 10 to the power minus 9. If you take R equal to 10 you are closer to the true optimal solution. So, this way if you proceed we will go closer and closer to the true optimal solution.

So, with this we will stop the last lecture of the week 10 here. So, what we learnt today is; how to make use of the built in functions for the solution of Linear Programming Problem, Quadratic Programming Problem, and unconstrained optimization problem for solution of Nonlinear Programming Problem.