Optimization Algorithms: Theory and Software Implementation

Prof. Thirumulanathan D

Department of Mathematics

Institute of IIT Kanpur

Lecture: 33

So, recall that in the last two lectures we were looking at algorithms for solving a constrained optimization problem. We started with the quadratic penalty method and we looked at the algorithm when you have a single equality constraint and no inequality constraint. Looking at the algorithm here, you can see that the method actually has both an inner loop and an outer loop. Whatever I have written here is just the outer loop. The inner loop is in finding the minimizer of $Q(\mathbf{x}, \gamma_k)$ in each of the step. This is the outer loop and this part is the inner loop. Here we have actually used Newton's method. You could use any other method as well like gradient descent or conjugate gradient method or quasi-Newton methods. I am not going to do that here as we will look at more interesting questions.

This is what we have written for when there is only one equality constraint. In case of one equality constraint we have

$Q(\mathbf{x}, \gamma) = f(\mathbf{x}) + \gamma\, (h(\mathbf{x}))^2.$

We calculated it by hand in the last lecture.

You can figure out that the gradient is

$\nabla f(\mathbf{x}) + 2\gamma\, h(\mathbf{x})\, \nabla h(\mathbf{x}).$

If you want to find the Hessian that will be

$\nabla^2 f(\mathbf{x}) + 2\gamma\, h(\mathbf{x})\, \nabla^2 h(\mathbf{x}) + 2\gamma\, \nabla h(\mathbf{x})\, \nabla h(\mathbf{x})^T.$

We implemented all of that in whatever we had written.

We computed $\nabla h(\mathbf{x})$ and $\nabla^2 h(\mathbf{x})$ by hand and we had all the expressions that we wanted to have.

Now the question is what happens when you have multiple equality constraints. Let us consider a system where you have no inequality constraint, but you have multiple equality constraints. In other words, you have $p = 0$, but $m > 1$, m could be any general positive integer. In that case how should you write your $Q(\mathbf{x}, \gamma)$? When you had one equality constraint you wrote it as $f(\mathbf{x}) + \gamma\, (h(\mathbf{x}))^2$.

When you have multiple equality constraints we write it as

$f(\mathbf{x}) + \gamma \sum_{j=1}^{m} (h_j(\mathbf{x}))^2.$

Suppose you have m equality constraints, you levy a penalty of γ for each of the constraints that are getting violated. If only one of the constraint is getting violated, only for that constraint a penalty of γ is levied and for all the others since $h_j(\mathbf{x})$ will be 0, so no penalty is levied. If there are multiple constraints for each of those constraints the same penalty of γ is levied. Writing the gradient as well as the Hessian is simple.

The gradient is just

$\nabla f(\mathbf{x}) + 2\gamma \sum_{j=1}^{m} h_j(\mathbf{x}) \nabla h_j(\mathbf{x})$

and

Hessian is again the Hessian of $f(\mathbf{x}) + 2\gamma \sum_{j=1}^{m} [h_j(\mathbf{x}) \nabla^2 h_j(\mathbf{x}) + \nabla h_j(\mathbf{x}) \nabla h_j(\mathbf{x})^T]$ with the summation running from j equal to 1 to m. This is just plain differentiation, so nothing great.

You can recall from the last class that we mentioned that Q needs to be differentiable. We have achieved them here. Whether it is one equality constraint or multiple equality constraint, the Q is actually twice differentiable. When γ is very large we are actually attaining the answer. We saw an example where when γ was 2^{18} as large as that we actually arrived at the answer. We expect that to happen even when you have multiple equality constraints.

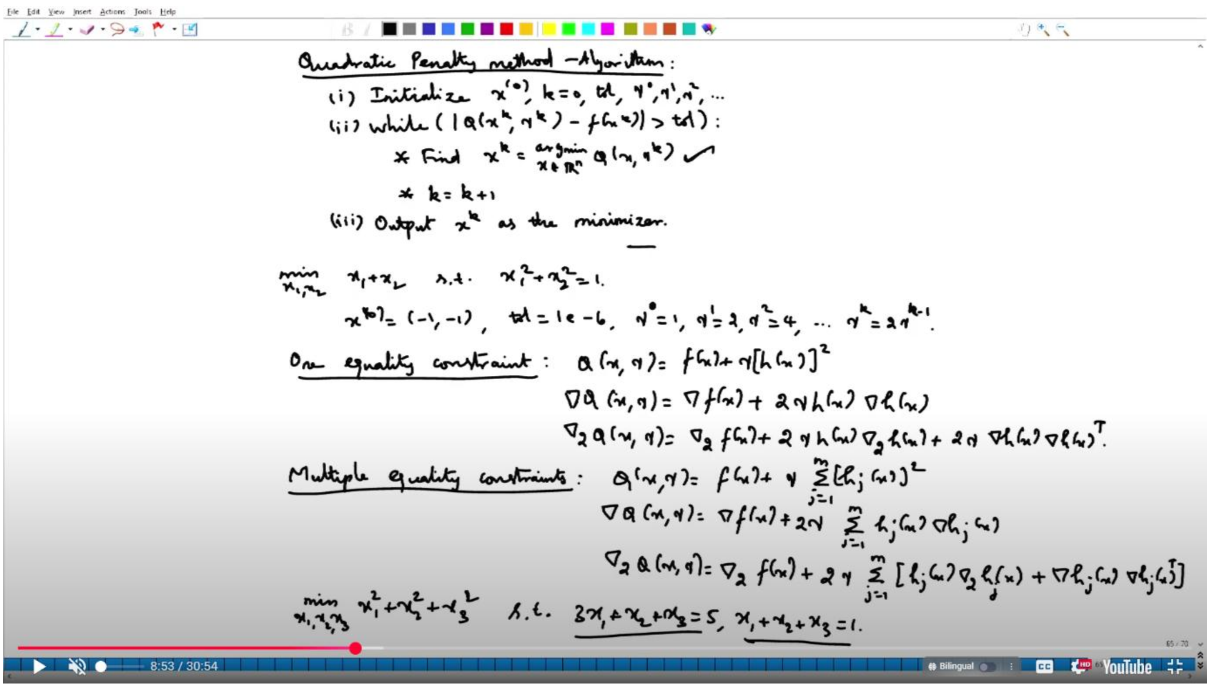
Let us consider an example. Let us say we have this problem.

We are minimizing $x_1^2 + x_2^2 + x_3^2$ with respect to x_1, x_2, x_3,

subject to $3x_1 + x_2 + x_3 = 5$ and $x_1 + x_2 + x_3 = 1$.

If you are trying to understand geometrically, you are trying to find the minimum distance from the origin. $x_1^2 + x_2^2 + x_3^2$ represents the distance from the origin to the intersection of these two planes. $3x_1 + x_2 + x_3 = 5$ is one plane and $x_1 + x_2 + x_3 = 1$ is another plane.

(Refer Slide Time 8:40)



Analytically finding an answer to this problem is quite easy because the answer must satisfy these two conditions. You can also include the KKT conditions.

The Lagrangian $L = x_1^2 + x_2^2 + x_3^2 + \lambda_1(3x_1 + x_2 + x_3 - 5) + \lambda_2(x_1 + x_2 + x_3 - 1)$.

When you differentiate with respect to x_1, you have $2x_1 + 3\lambda_1 + \lambda_2 = 0$. $2x_2 + \lambda_1 + \lambda_2 = 0$ and $2x_3 + \lambda_1 + \lambda_2 = 0$.

And of course, these two constraints you have $3x_1 + x_2 + x_3 = 5$ and $x_1 + x_2 + x_3 = 1$.

You have 5 equations and 5 unknowns. It is much simpler than you think because you can easily find that by subtracting these two, you have $2x_1 = 4$. So $x_1 = 2$. From these two you can see that $x_2 = x_3$.

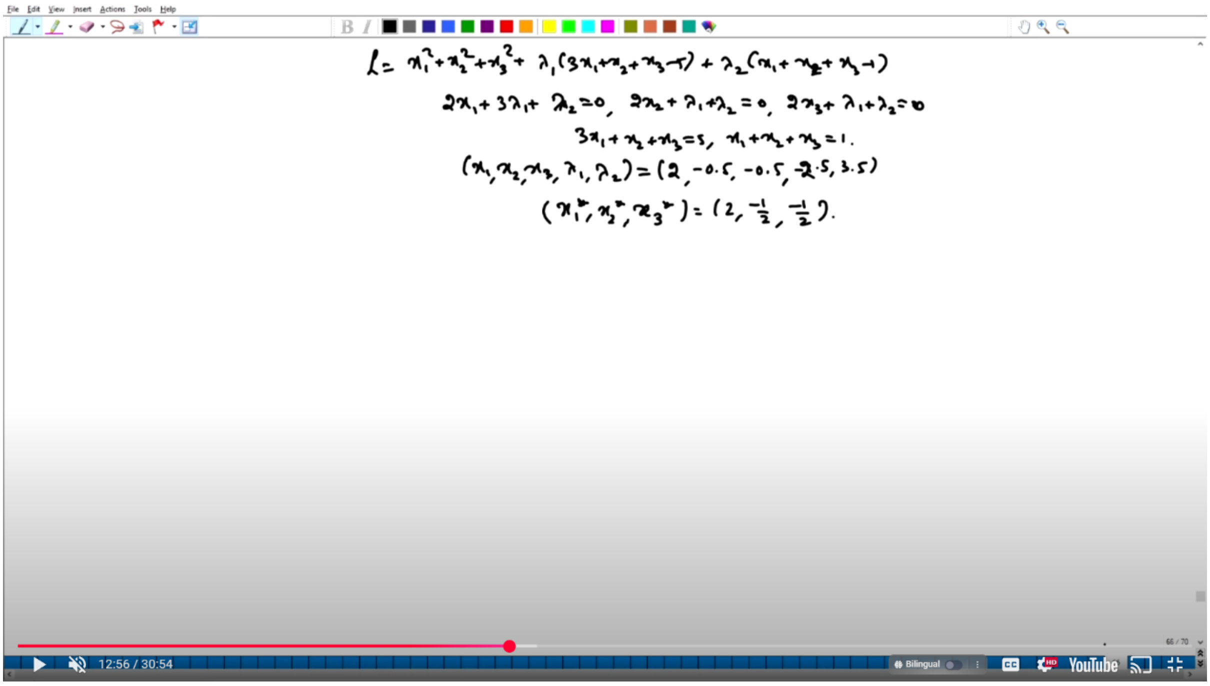
If $3x_1$ is 2, you have $x_2 + x_3 = -1$. So, this will be -0.5 and -0.5. For λ_1 and λ_2, you have to solve an equation. You have $2x_1$ is 4 so you have $3\lambda_1 + \lambda_2 = -4$.

In all the equations you have λ_1 and λ_2. It is good to check if at least one pair of λ_1 and λ_2 exists. Here you have $\lambda_1 + \lambda_2 = -2x_2$ and here you have $3\lambda_1 + \lambda_2 = -4$.

On solving you will have -2.5 and 3.5. So you have -2.5 + 3.5 = 1.

You have -7.5 + 3.5 = -4. So all right. This is the solution. The final answer that we are looking for is x_1^* x_2^* x_3^* is 2, -½, -½.

(Refer Slide Time 12:00)



We will use this particular algorithm, the quadratic penalty method algorithm. We can initialize $\mathbf{x}_0$ as [0, 0, 0]. We can use the same tolerance 1e-6 and the same set of γs as well. We will start with $\gamma_0 = 1$ and we will successively multiply the γ values by a factor of 2. We will check if we are getting the answer 2, -½, -½.

We will copy this algorithm and name it quadratic penalty method for multiple equality constraints.

$f(\mathbf{x}) = x[0]^2 + x[1]^2 + x[2]^2$.

We will write the equality constraints that is

$h(\mathbf{x})$ as the constraints which is $3x_1 + x_2 + x_3 - 5 = 0$ and $x[0] + x[1] + x[2] - 1 = 0$.

We write the $\nabla h(\mathbf{x})$ as follows. When you take the gradient of the first equality you get [3, 1, 1] and for the second you get [1, 1, 1].

Now that we have defined h and ∇h, we can write

Q as $f(\mathbf{x}) + \gamma * (h_1(\mathbf{x}))^2 + (h_2(\mathbf{x}))^2$.

Instead of writing them one by one given that we have defined it here, we will write it as $h(\mathbf{x}) \cdot h(\mathbf{x})$. We are just taking the dot product of this vector into the same vector, so that is just sum of $(h_j(\mathbf{x}))^2$. This is a simpler way of doing this. Similarly, here we can write $\nabla f(\mathbf{x})$. I could have straight away used $\nabla f(\mathbf{x})$, but since it is just $2\mathbf{x}$, I will just write it as $2\mathbf{x}$, which is much easier.

Here I am going to write this equation $2\gamma \sum_j h_j \nabla h_j$ so this is $2\gamma * h(\mathbf{x}) * \nabla h(\mathbf{x})$. Since I have defined h and ∇h, so I need not mull over writing h value, ∇h, multiplying them and so on. This is a simpler way of doing it.

If you recall what is the Hessian, it is the Hessian of $f + 2\gamma \sum_j [h_j \nabla^2 h_j + \nabla h_j \nabla h_j^T]$.

Note that $\nabla^2 h_j$ is actually 0 because ∇h is a constant.

So, $\nabla^2 h_j$ is actually 0 and $\nabla^2 f$ is actually 2I. So, I can write it as 2 * np.eye(3) and here it is going to be $2\gamma * \nabla h_j$ so it is this part.

Since ∇^2 is 0, you just need $\nabla h_j \nabla h_j^T$ so just write that $2\gamma * \nabla h(\mathbf{x})^T$ @ $\nabla h(\mathbf{x})$.

This is because ∇h usually is 2 x 3 in this case. ∇h is a column vector. Since I have written ∇h as a row vector, so we get ∇h^T @ ∇h.

Everything is the same so again I have to modify $\mathbf{x}$.

This condition is all same and here I need one more since this is a three-dimensional problem. This is basically the code and let us check if you get the answer 2, -½, -½.

I see an error so this is 2γ * so there has to be a sum here so this is h so $h(\mathbf{x})$ I have to write the dot product. I am sorry, not the dot product. $h(\mathbf{x})$ is two dimensional.

There is an error that I had to fix, so this is, so $h(\mathbf{x})$ is 1 x 2 matrix and this is 2 x 3 matrix, so that you get a 1 x 3 matrix here, so this has to be a matrix multiplication and not an element wise multiplication as I had written before. Let us apply that change and check if we get the right answer.

We started with $\gamma = 1$.

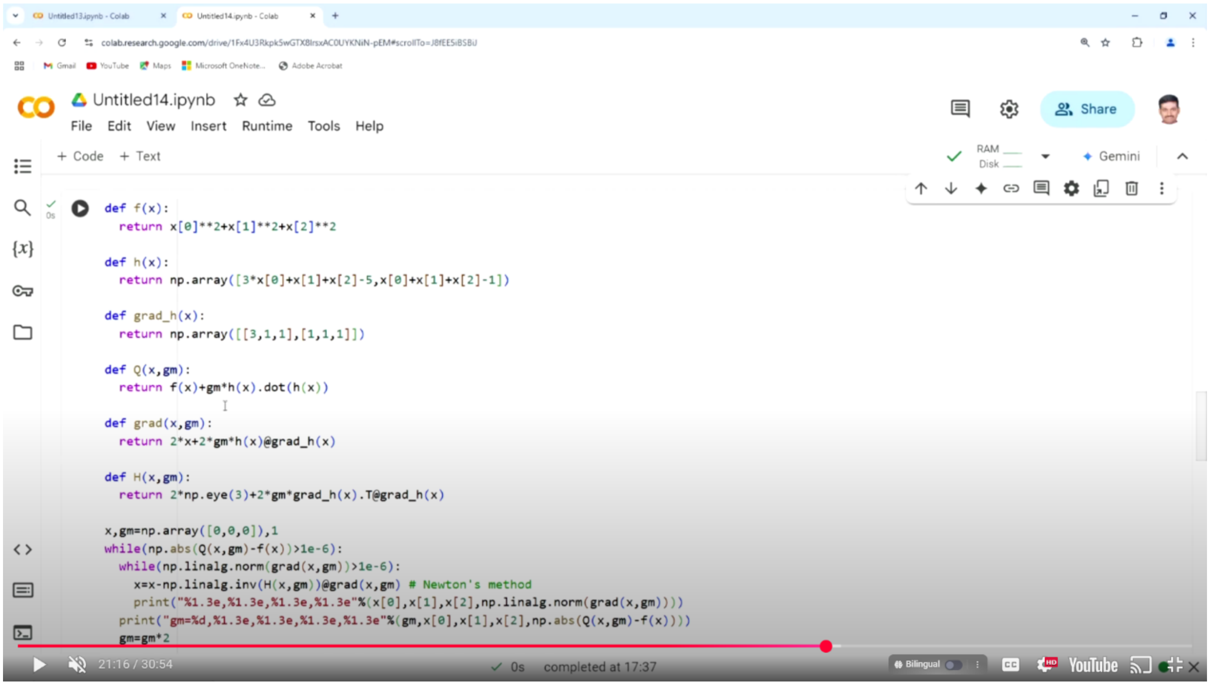
The answer for $\gamma = 1$, the minimizer is 1.391, 0.08696, 0.08696 and when $\gamma = 2$, the minimizer is 1.574, -0.06557, -0.06557 and so on.

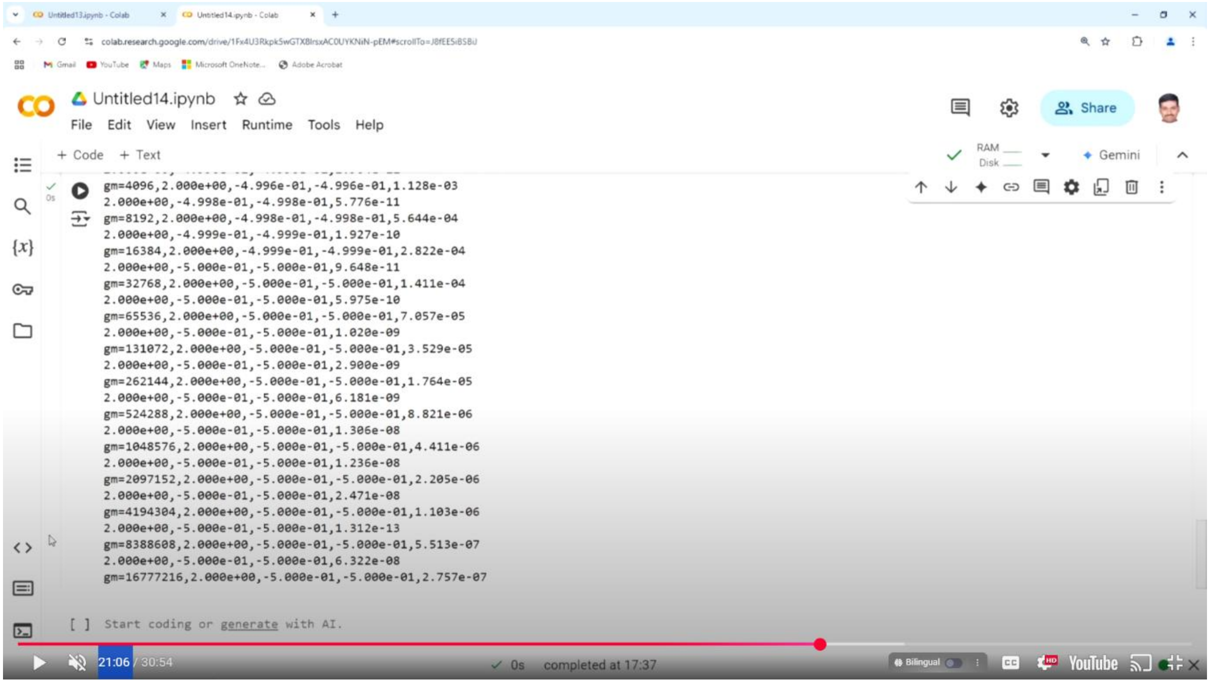
You can see that each of the inner loop has completed in just one step because that is how Newton's method is quite fast. But you get the final answer 2, -½, -½ when the γ value is actually 2^{24}.

So that is in $\gamma = \gamma^{25}$. The number needs to be that large.

It is $1.6 * 10^7$ only for that high a value of γ we get the right answer. Nevertheless, you can see that the two things to observe: one is that we are able to solve a problem that has multiple equality constraints and the other one is that by seeing this code you might have thought that writing the expression for gradient the Hessian and so on might become more cumbersome when you have more constraints, but if you represent all of them as separate constraints and write down the gradient, you can use the same formula whatever you have here when you have multiple equality constraints.

(Refer Slide Time 20:55)





Now, we understand that we can use the quadratic penalty method for when you have multiple equality constraints as well. Now, we will move to the case when you have inequality constraints. There is something that we need to understand when we are using inequality constraints. Consider the case of **inequality constraints**.

In this case $Q(\mathbf{x}, \gamma) = f(\mathbf{x}) + \gamma \sum_{i=1}^{p} (\max(0, g_i(\mathbf{x})))^2$.

It is not just like what we had $(g_i(\mathbf{x}))^2$.

You have to take a maximum of 0 and the value of the constraint and then take a square. Why do we need this?

Please recall that the penalty is levied only when the constraint is violated. $h_j(\mathbf{x})$ is violated if $h_j(\mathbf{x})$ is not equal to 0.

So for $h_j(\mathbf{x}) = 0$, you have a 0 penalty, $h_j(\mathbf{x})$ not equal to 0, you have a positive penalty. But for inequality constraint, the constraint is satisfied when $g_i(\mathbf{x}) \leq 0$.

So, even if it is -5, -10, -100, $-\infty$, in all those cases $g_i(\mathbf{x}) \leq 0$ is satisfied. Only when $g_i(\mathbf{x}) > 0$, the constraint is violated.

So, that is why we have $\max(0, g_i(\mathbf{x}))$. If $g_i(\mathbf{x})$ is negative, then 0 penalty is levied and if $g_i(\mathbf{x})$ is positive, only then a positive penalty is levied.

This is in some sense a straightforward extension of the penalty function that you have for inequality constraints, but there is a problem which I will tell you shortly.

Let us find the gradient and

Hessian of this Q.

$\nabla Q(\mathbf{x}, \gamma) = \nabla f(\mathbf{x}) + 2\gamma \sum_{i=1}^{p} \max(0, g_i(\mathbf{x}))\, \nabla g_i(\mathbf{x})$.

You might be wondering if what I have written is correct, because when you have a max operator, the differentiability of Q comes into question. That is correct. At least in the boundary when $g_i(\mathbf{x}) = 0$, it is not clear if $Q(\mathbf{x}, \gamma)$ is differentiable.

But I claim that it is actually differentiable because you can see that when $g_i(\mathbf{x}) < 0$, you only have $Q(\mathbf{x}, \gamma) = f(\mathbf{x})$ and $\nabla Q(\mathbf{x}, \gamma) = \nabla f(\mathbf{x})$.

And if $g_i(\mathbf{x}) = 0$, then again $Q(\mathbf{x}, \gamma)$ is $f(\mathbf{x}) + \gamma\ (g_i(\mathbf{x}))^2$, but $g_i(\mathbf{x})$ is 0 so it is just $f(\mathbf{x})$ and $\nabla Q(\mathbf{x}, \gamma) = \nabla f(\mathbf{x}) + 2\gamma\ g_i(\mathbf{x})\ \nabla g_i(\mathbf{x})$. $g_i(\mathbf{x}) = 0$, so this is actually just $\nabla f(\mathbf{x})$.

The reason why ∇Q is actually differentiable is because the issue will occur only when $g_i(\mathbf{x}) = 0$ and in that case this term actually turns out to be 0.

So, that is the reason you have the expression for $Q(\mathbf{x}, \gamma)$ is actually differentiable. There is no issue with the differentiability of Q, but unfortunately it is not twice differentiable.

Why is that the case? If we have, if $g_i(\mathbf{x}) < 0$, then Q is f, ∇Q is ∇f, and the Hessian of Q is just the Hessian of f.

No problem until here. But when $g_i(\mathbf{x}) = 0$, Q is $f(\mathbf{x}) + \gamma\ (g_i(\mathbf{x}))^2$, ∇Q is $\nabla f(\mathbf{x}) + 2\gamma\ g_i\ \nabla g_i$.

Now what is the Hessian of $Q(\mathbf{x}, \gamma)$?

That will be $\nabla^2 f(\mathbf{x}) + 2\gamma\ g_i(\mathbf{x})\ \nabla^2 g_i(\mathbf{x}) + 2\gamma\ \nabla g_i(\mathbf{x})\ \nabla g_i(\mathbf{x})^T$.

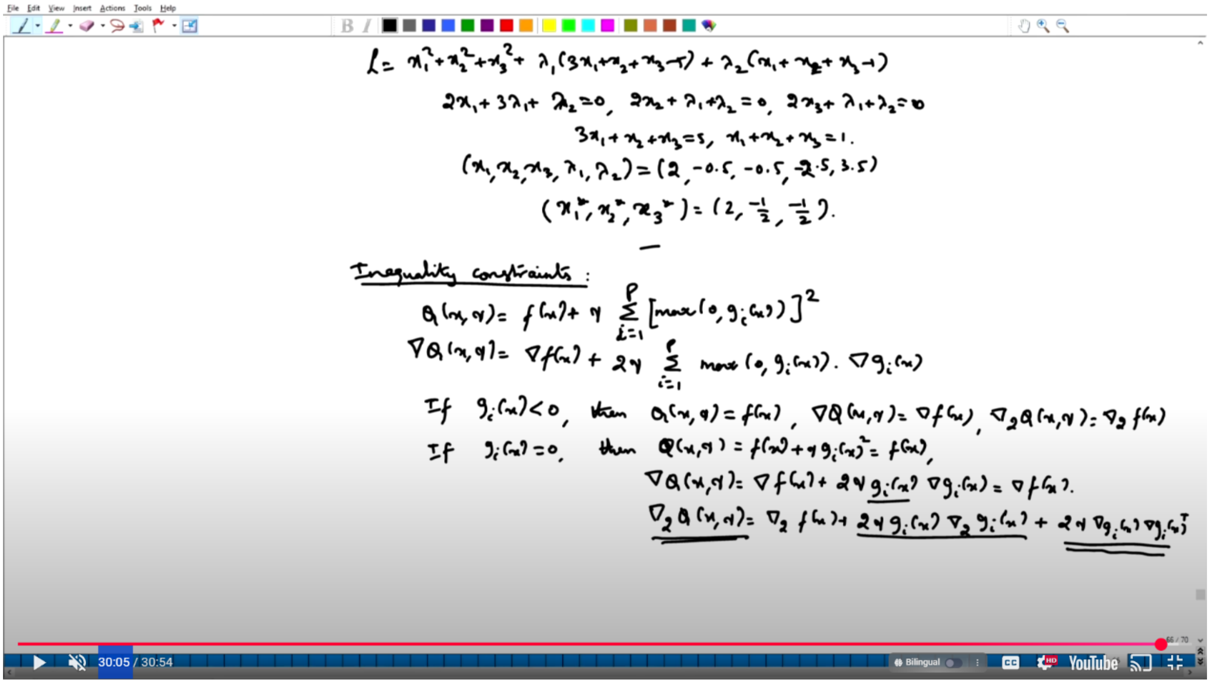
This term becomes 0, agree, because $g_i(\mathbf{x}) = 0$, but this term does not become 0.

When you come from $g_i(\mathbf{x})$ negative to $g_i(\mathbf{x}) = 0$, you will not have this final term $2\gamma\ \nabla g_i(\mathbf{x})\ \nabla g_i(\mathbf{x})^T$ will be 0.

But if you come from the positive direction, you will have this particular term where $2\gamma\ \nabla g_i(\mathbf{x})\ \nabla g_i(\mathbf{x})^T$.

So, if $\nabla g_i(\mathbf{x})\ \nabla g_i(\mathbf{x})^T$ is not 0, if it is non-zero, then the Hessian of this particular expression $Q(\mathbf{x}, \gamma)$ does not exist.

(Refer Slide Time 30:00)



This is something that you need to remember. Actually using Hessian method to solve a problem with inequality constraints sometimes turn out to be a problem. In many of the practical cases if you do give this particular Hessian it usually works, but you can say that it is actually slightly different from Newton's method because the Hessian that we have is not actually the Hessian. That is good to remember. In the next lecture, we will actually look at an example where we solve a problem that has inequality constraints. Thank you.