Optimization Algorithms: Theory and Software Implementation

Prof. Thirumulanathan D

Department of Mathematics

Institute of IIT Kanpur

Lecture: 24

Hello everyone, this is the fourth lecture of week 5. Recall that we were learning about Newton's method. In the previous three lectures, we learned the general form of Newton's algorithm and argued that this method has faster convergence than the methods discussed before. The other methods have linear convergence, while Newton's method has quadratic convergence. We also showed this at the end of the previous lecture

Other than this, we also mentioned three issues. One is no global convergence; there is only local convergence. We also proved local convergence at the end of the previous lecture. The direction we consider may not be a descent direction if the Hessian is not positive definite. The other issue was regarding the inversion of the Hessian matrix. We saw a lot of examples citing where those issues come from.

In this lecture, we will start looking at ways of getting around these issues. We do not have global convergence, but can we twist the algorithm a little bit to get global convergence? Similarly, for the other issues, we may not have H_k to be positive definite, and the inversion issues will be there. Is there a small tweak we can do to the algorithm so that we actually nullify these issues and get on track? That is what we will be discussing in this lecture.

Let me recall the issues: no global convergence, $-H_k^{-1}g_k$ need not be a descent direction, and matrix inversion is a computationally expensive process. We will first consider the global convergence issue.

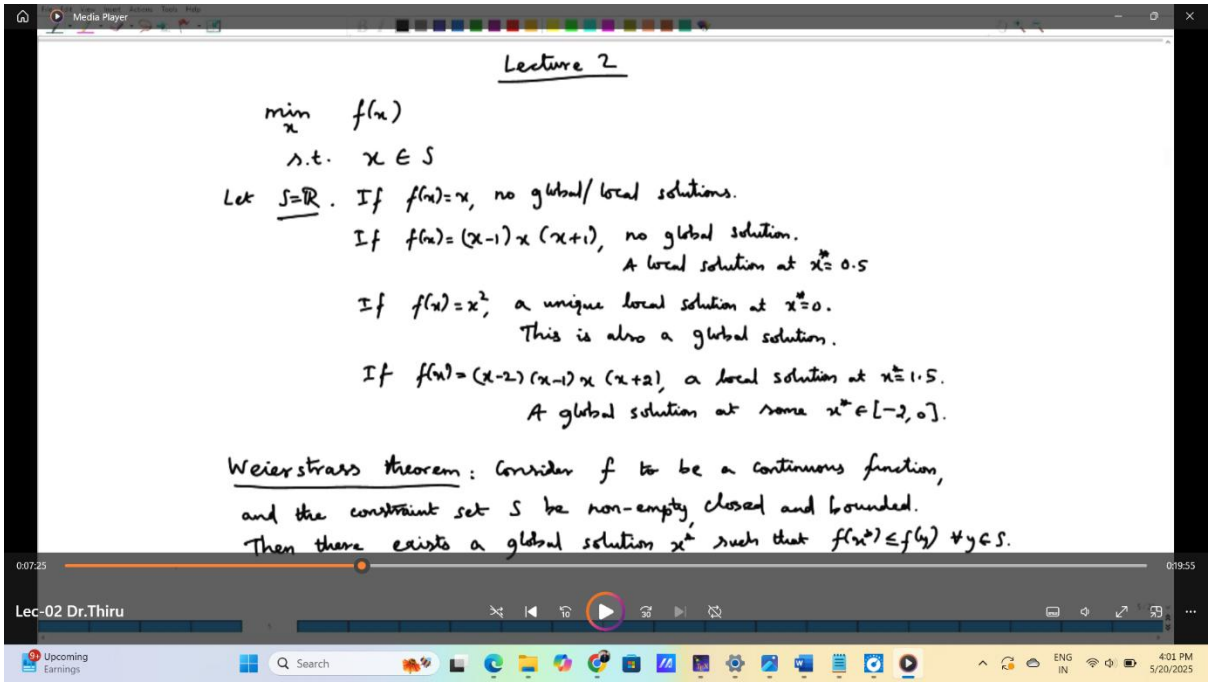
I am going to introduce to you a method called the damped Newton's method. This is a small tweak of the standard Newton's method. What is damped Newton's method? I will just write down the algorithm; it is exactly Newton's method except for a change in one of the lines.

First, initialize as usual: x_0, tolerance, and $k = 0$. While $\|g_k\| >$ tolerance, do the following:

1. Fix $d_k = -H_k^{-1}g_k$.
2. Choose α_k by backtracking line search.
3. Set $x_{k+1} = x_k + \alpha_k d_k$.
4. Set $k = k + 1$.

Finally, output $x^* = x_k$.

(Refer Slide Time 5:56)



The only difference between Newton's method and damped Newton's method is that you choose α_k by backtracking line search in the damped version, whereas in the standard method you blindly choose $\alpha_k = 1$. I claim that this would actually give us global convergence.

Let us verify it for the example where we had the issue of no global convergence. The example was

$f(x) = \sqrt{(x_1^2 + 1)} + \sqrt{(x_2^2 + 1)}$.

If you had $x_0 = (0.5, 0.5)$, you had convergence. If you had (1, 1), it was shuttling between (1, 1) and (-1, -1) repeatedly. When you had (2, 2), it blew up to infinity. We will use damped Newton's method to see if the issue is rectified.

We will implement this. We set the descent direction d as $-H^{-1}g$. We choose an initial $\alpha = 1$ and then, while the Armijo condition is not satisfied, we keep shrinking α by a factor of ρ. Let's set $c_1 = 0.75$ and $\rho = 0.8$. Then we update $x = x + \alpha * d$. Everything else is the same.

For (2, 2), when we ran the standard Newton's method, we got infinity in six steps. Let's see what happens here. In three steps, you can see that you are at (0, 0), which is the right answer. Let's try a bigger number, say (20, 20), to see if that is also taken care of. Fortunately, in four steps, you actually converge to 0.

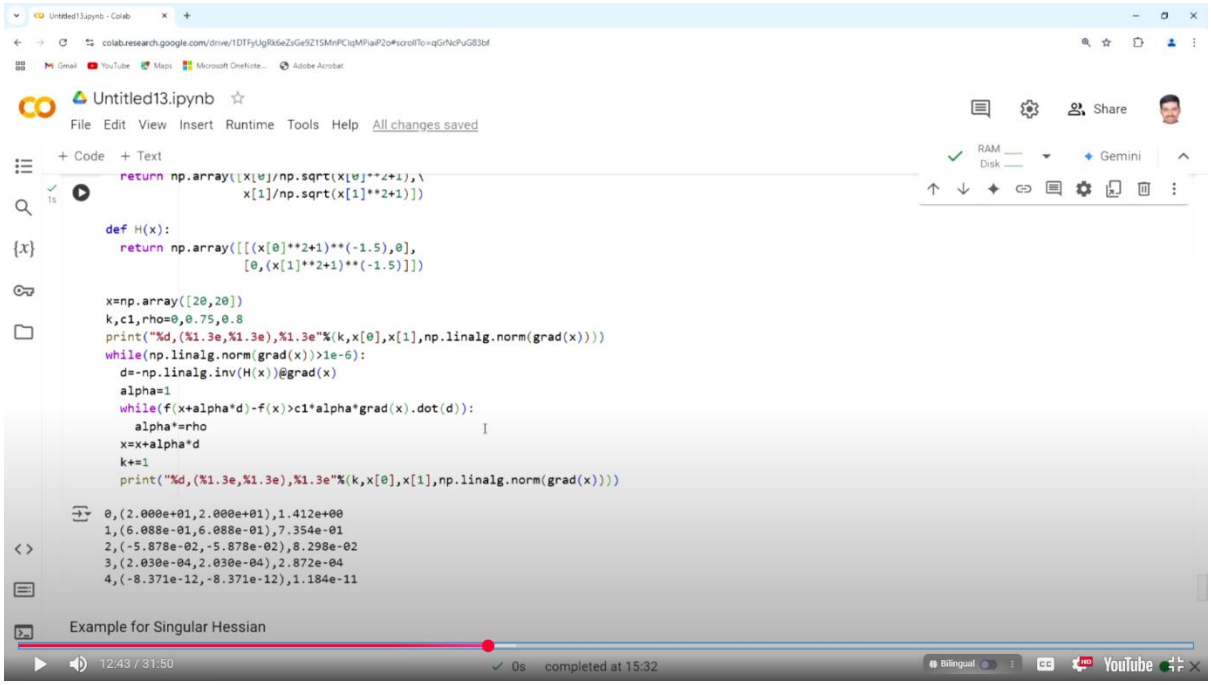
This means that if you choose the damped Newton's algorithm instead of the standard algorithm, it slows down a bit—that is correct—because you are not having $\alpha = 1$. The step size is being pulled down. But by doing that, the damped Newton's method actually achieves global convergence.

To understand this physically: suppose you have someone who runs very fast to the destination when you give them the direction, but they sometimes overstep. What do you do? You try to pull their speed down, telling them to go slow. That is exactly what the damped Newton's method does. Newton's method is going too fast, so the damped version damps the speed in each step by using a smaller step size. This takes care of the fact that it does not overstep in each iteration. That is basically the damped Newton's method. It is a very straightforward method; there is nothing very sophisticated happening compared to the standard method.

This takes care of the no global convergence issue. Suppose you start at a particular point; you do not know x*, so you will obviously start at an arbitrary point. Because of the no global convergence property, the answer might blow up. What would you do next? You don't know whether to increase or decrease the value; you might have to choose a point like (1, 3) based on the problem, but you don't know all these things. It is actually hard when the answer depends on the initial point. The damped Newton's method comes to our rescue. If the answer gets wrong because of the no global convergence property, you can just move to the damped Newton's method and you will get the answer.

One more thing about the damped Newton's method: you might be thinking whether it would take care of the other issues as well. For example, if $-H_k^{-1}g_k$ was not a descent direction (when H_k is not positive definite), would the damped Newton's method take care of that? Unfortunately, that is not possible because the issue there is not the step size; it is actually the direction. Even if you choose an α that is less than 1, the direction is still incorrect. The damped Newton's method does not work if the issue occurs because $-H_k^{-1}g_k$ is not a descent direction.

(Refer Slide Time 12:43)

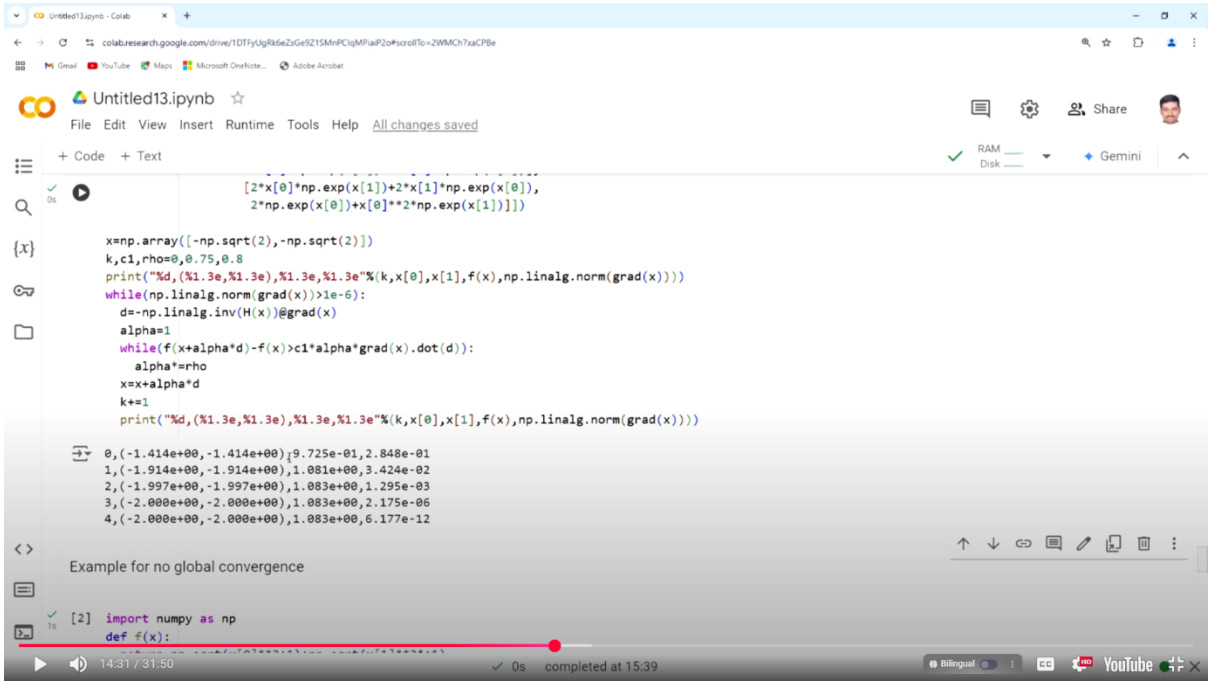


Let me illustrate that. Recall the example

$f(x) = x_1^2 e^{x_2} + x_2^2 e^{x_1}$.

When you start with $(-\sqrt{2}, -\sqrt{2})$, we had an increase of the f value, and the convergence was to a saddle point. Let's try the damped method here to see if it works. You can see that there is no change; the issue persists. So, the damped Newton's method works for no global convergence, but it does not rectify the other property. It rectifies no global convergence but does not rectify $-H_k^{-1}g_k$ being an ascent direction (the opposite of a descent direction).

(Refer Slide Time 14:31)



Now the question is, how do we rectify $-H_k^{-1}g_k$ not being a descent direction? What we do is we actually modify H_k to a slightly different matrix. I will call that the modified Newton's method. Before that, I will explain the theory behind it.

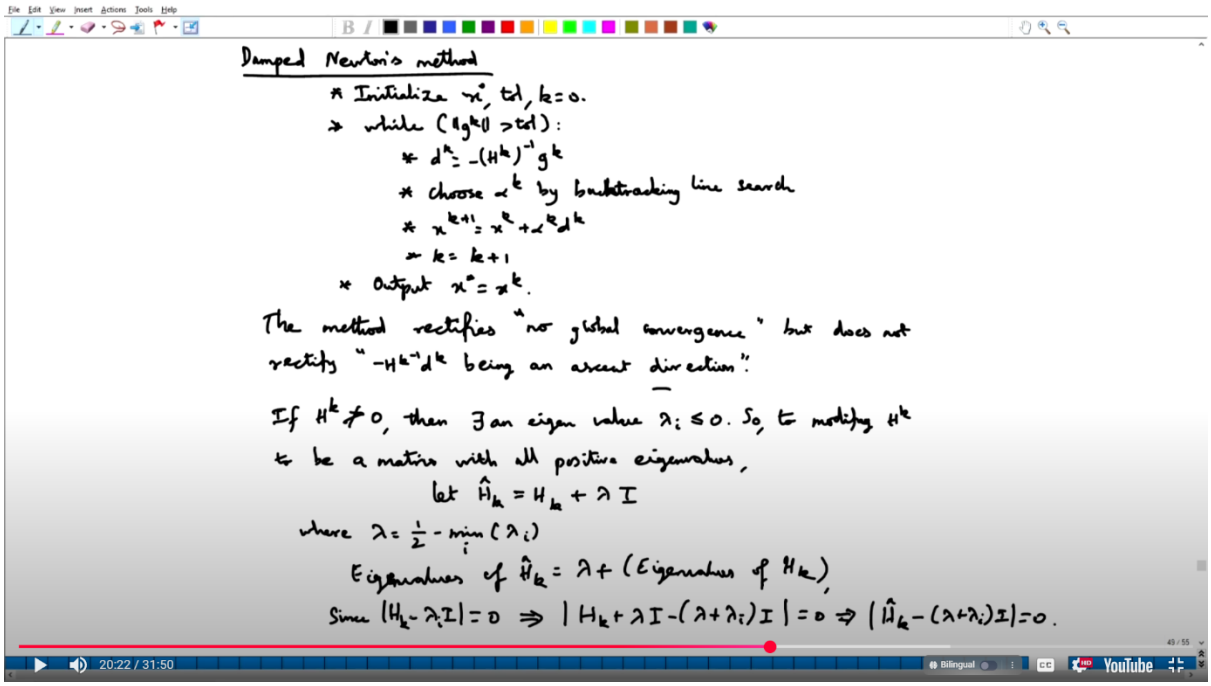
The issue, as you can see, is that if H_k is not positive definite, then there exists an eigenvalue λ_i that is less than or equal to 0. A way of characterizing a positive definite matrix is to say that every eigenvalue of that matrix is positive. For a negative definite matrix, every eigenvalue is negative. For a positive semi-definite matrix, every eigenvalue is non-negative. For a negative semi-definite matrix, every eigenvalue is non-positive. For an indefinite matrix, there exists at least one eigenvalue that is positive and one that is negative.

Since it is not positive definite, that means there should be at least one eigenvalue which is non-positive. The idea is to modify this matrix so that every eigenvalue is positive. Construct $\hat{H}_k = H_k + \lambda I$, where $\lambda = 0.5 - \min(\lambda_i)$. If λ_i's are the eigenvalues of H_k, find the minimum of those λ_i's. Since there exists at least one eigenvalue which is non-positive, you will have $\min_i(\lambda_i)$ to be either 0 or some negative number, and λ will be something greater than 0.5. I claim that $\hat{H}_k$ is a positive definite matrix.

Why is that? The eigenvalues of $\hat{H}_k$ are λ + eigenvalues of H_k. The proof is simple: if λ_i are the eigenvalues of H_k, then $|H_k - \lambda_i I|$ will be singular for all i. Now,

$|\hat{H}_k - (\lambda + \lambda_i)I| = 0$, implies that $|(H_k + \lambda I) - (\lambda + \lambda_i)I| = 0$, implies that $|H_k - \lambda_i I| = 0$. This proves that the eigenvalues of $\hat{H}_k$ are $\lambda + \lambda_i$.

(Refer Slide Time 20:22)



Since I have written $\lambda = 0.5 - \min(\lambda_i)$, you can see that the minimum eigenvalue of $\hat{H}_k$ is 0.5, which is positive. You might ask, what is the significance of 0.5? You can put any positive number. One thing is, if it is very close to zero, then inversion will make it too high. If the lowest eigenvalue is not 0 but is 0.00001 or 10^{-8}, then H^{-1} will blow up, which is not what you want. You also do not want λ to be too high, because then H_k and $\hat{H}_k$ differ by too much. We want to retain the properties of H_k as far as possible.

So, we choose $\lambda = 0.5 - \min_i(\lambda_i)$.

Instead of the descent direction being $-H_k^{-1}g_k$, I am going to make it $-\hat{H}_k^{-1}g_k$. That is the change. The **modified Newton's method** is as follows. Initialize as usual.

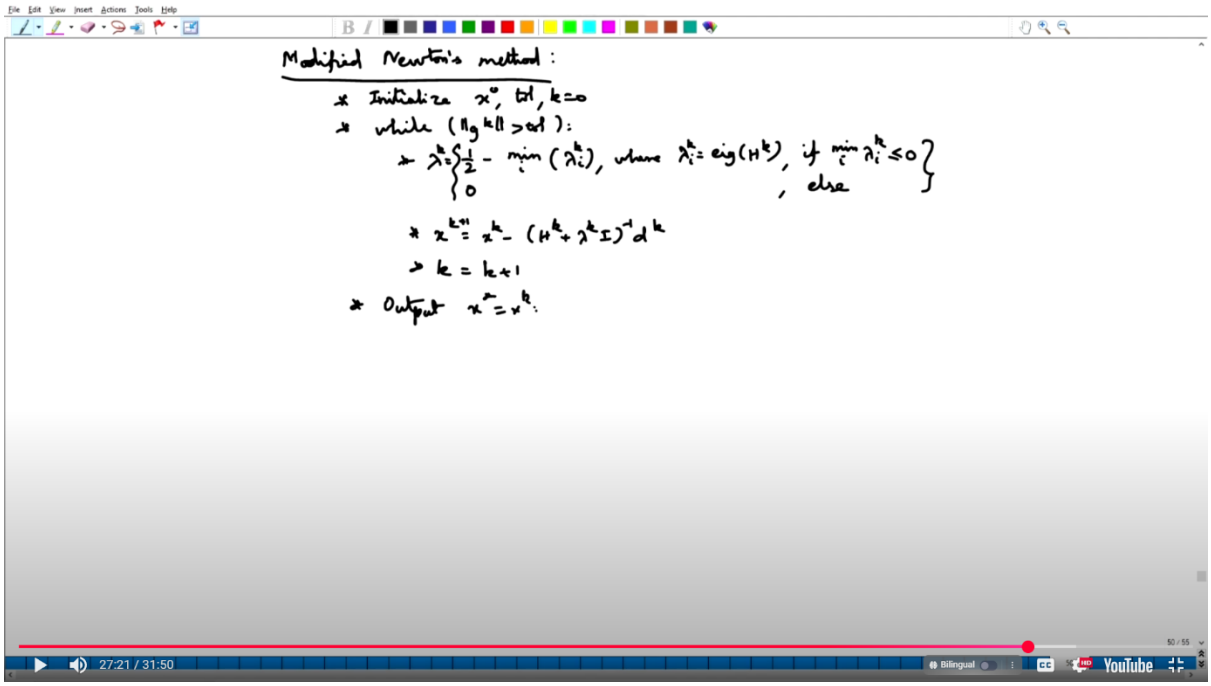
Then, define $\lambda = 0.5 - \min(\lambda_i)$, where λ_i are the eigenvalues of H_k. But this is only if $\min(\lambda_i) \leq 0$. Otherwise, if H_k is already positive definite, we set $\lambda = 0$ so we don't disturb the algorithm.

Then, $x_{k+1} = x_k - (H_k + \lambda I)^{-1}g_k$,

$k = k + 1$, and finally output

$x^* = x_k$.

(Refer Slide Time 27:20)



I claim that this will fix the issue of $-H_k^{-1}g_k$ not being a descent direction. The issue came because H_k was not a positive definite matrix.

Now $H_k + \lambda_k I$ is a positive definite matrix, so it should help us.

Let's verify if this particular method works in the example where we were failing. Let's implement this. We need to find λ first.

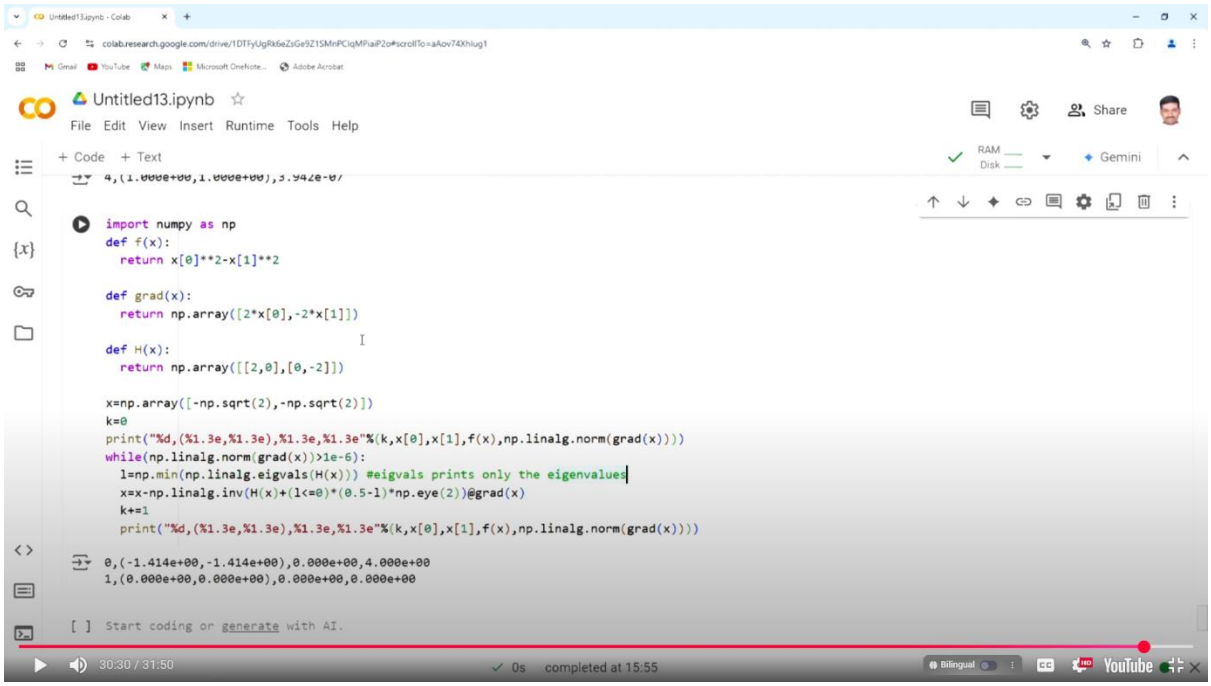
We can use np.linalg.eigvals(H(x)) to get the eigenvalues.

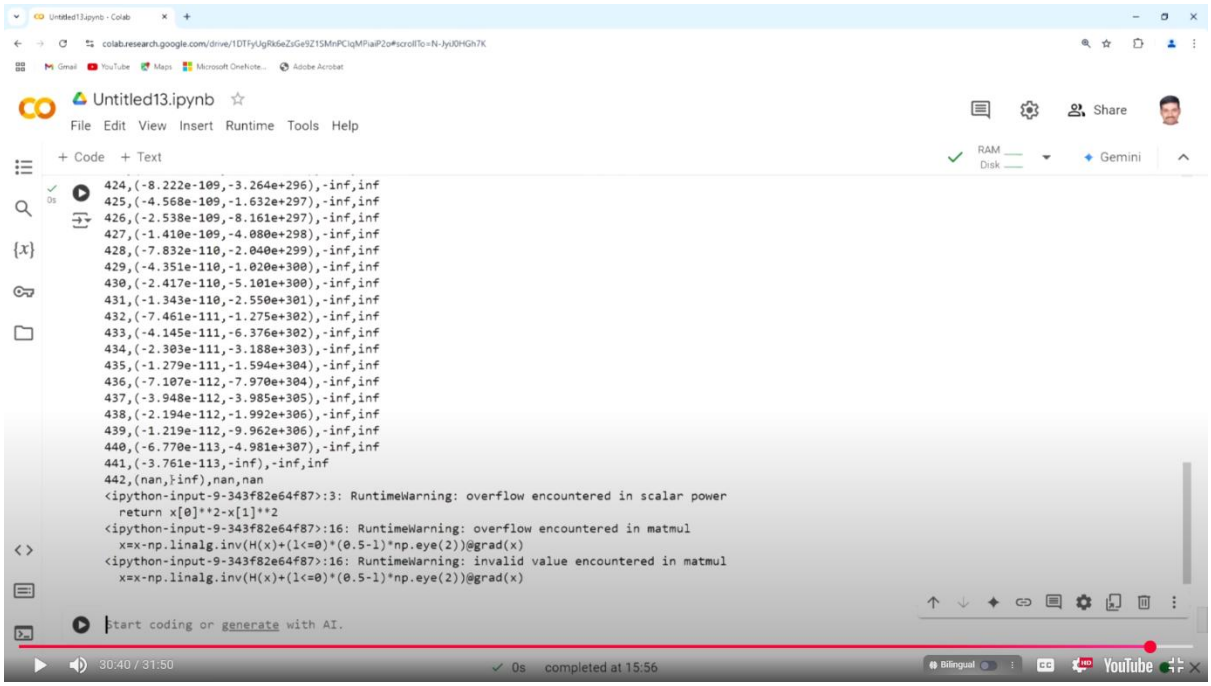
Let's call this array L. Then, l = min(L). If l <= 0, we set λ_val = 0.5 - l. Otherwise, we set λ_val = 0. Then, the descent direction becomes

d = -inv(H(x) + λ_val * I) * g.

Let's check if this corrects itself. I started with $(-\sqrt{2}, -\sqrt{2})$. You can see that we have converged to (0, 0). Not just for this example, but even for the example $f(x) = x_1^2 - x_2^2$. Without this modification, you can recall it converged to (0, 0), which is a saddle point. But if you implement this modified method, you can see that it actually goes to $(0, -\infty)$, correctly identifying the unbounded descent. This makes it clear how the modified Newton's method gets over the issue of $-H_k^{-1}g_k$ not being a descent direction.

(Refer Slide Time 30:50)





It also gets across the issue that H_k could be singular. If H_k is singular, it means at least one of the eigenvalues is 0. If that is the minimum eigenvalue, then you can use this method to modify it to have all positive eigenvalues (i.e., non-singular) and get over that issue as well.

We will end this lecture. In the next lecture, we will look at some more properties of Newton's method. Thank you.