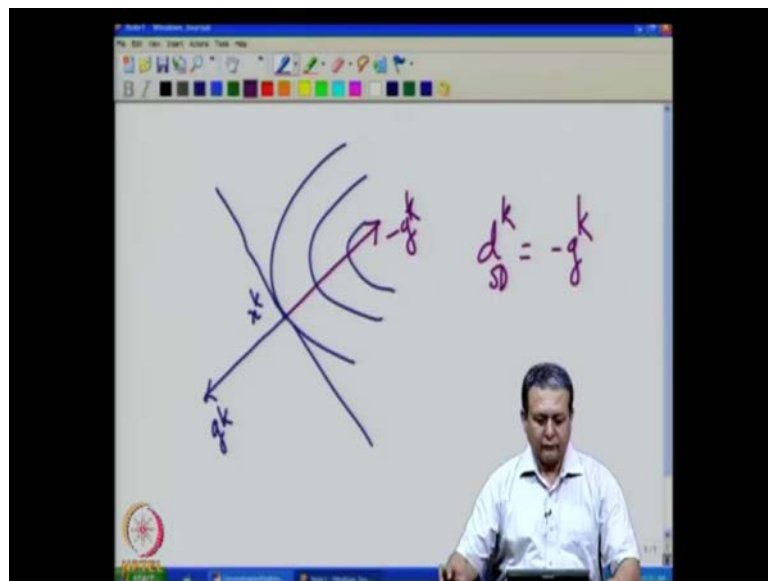


**Numerical Optimization**  
**Prof. Shirish K. Shevade**  
**Department of Computer Science and Automation**  
**Indian Institute of Science, Bangalore**

**Lecture - 13**  
**Steepest Descent Method**

Hello, welcome back to this series of lectures on numerical optimization. So, in the last class we started studying about Steepest Descent Method. So, in the Steepest Descent Method the idea is to move along the direction of Steepest Descent. So, we use the first order approximation of the objective function and that the Steepest Descent direction is negative of the gradient direction.

(Refer Slide Time: 00:49)



So, we saw that if these are the contours of the objective function and if you are currently at a point which we call it as  $x^k$ . Now, this is the first order approximation, this is the gradient at that  $x^k$  and with respect to the first order approximation. The maximum decrease in the objective function is possible when we move along the negative of the gradient direction.

So, if you move along this direction, then we get the maximum decrease in the first order approximation of the objective function. So, the direction  $d^k$  is equal to minus of  $g^k$  is called the Steepest Descent direction. So, it will be denoted by a subscript SD. So, at the  $k$ -

th iteration, the Steepest Descent direction is nothing but the negative of the gradient direction, then along with that direction. If we use line search then we have the steepest descent algorithm with lines search.

(Refer Slide Time: 02:15)

**Steepest Descent Method**

- Uses the steepest descent direction,  $d_{SD}^k = -g^k$

---

**Steepest Descent Algorithm**

- (1) Initialize  $x^0$  and  $\epsilon$ , set  $k := 0$ .
- (2) **while**  $\|g^k\| > \epsilon$ 
  - (a)  $d^k = -g^k$
  - (b) Find  $\alpha^k (> 0)$  along  $d^k$  such that
    - (i)  $f(x^k + \alpha^k d^k) < f(x^k)$
    - (ii)  $\alpha^k$  satisfies Armijo-Wolfe conditions
  - (c)  $x^{k+1} = x^k + \alpha^k d^k$
  - (d)  $k := k + 1$
- endwhile**

**Output :**  $x^* = x^k$ , a stationary point of  $f(x)$

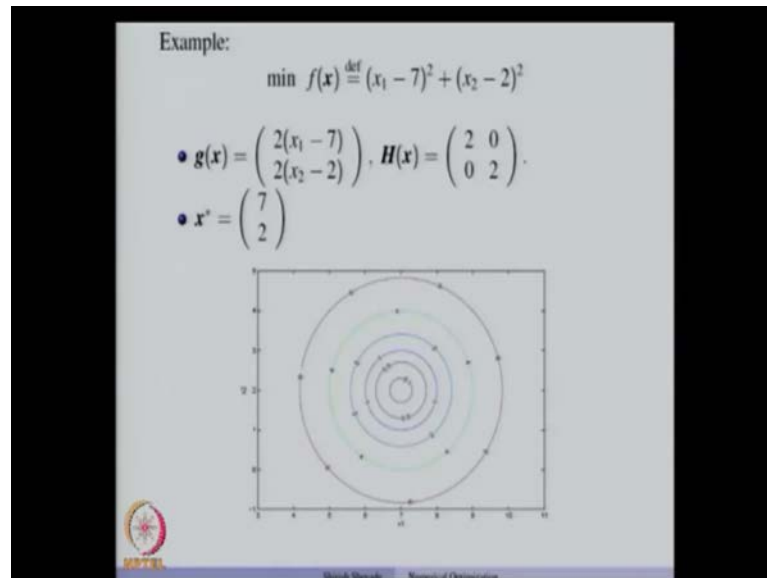
- Exact or Backtracking line search can

So, let us continue our discussion on the Steepest Descent Method. So, as I mentioned the Steepest Descent direction is at the k-th iteration is nothing but the negative of the gradient at that point and the Steepest Descent algorithm started with the usual initialization of the  $x^0$ . The tolerance for the norm of the gradient for a used for stopping and iteration number is set to  $\epsilon$ . So, while the norm of the gradient is greater than  $\epsilon$ , which is the direction  $d^k$  to be minus  $g^k$  and this is the thing but the Steepest Descent direction and then  $\alpha^k$  can use,  $\alpha^k$  can find the step length of positive length. Such that, the value of the objective function at  $x^k + \alpha^k d^k$  is less than  $f(x^k)$  and  $\alpha^k$  satisfies Armijo-Wolfe conditions. Remember that, these conditions are necessary to ensure convergence of an optimization algorithm these conditions along with the condition that the direction is a descent direction.

And then  $x^k$  is moved to  $x^{k+1}$  by adding  $\alpha^k d^k$ . The iteration counter is incremented by 1 and then the whole procedure is repeated till the norm of the gradient at a given point  $x^k$  is less than or equal to  $\epsilon$  and at that point the algorithm stops and as an output. We get  $x^*$  which is nothing but the  $x^k$  which is a stationary point of  $f(x)$ .

And then we started looking at some of the examples, note that the exact or backtracking line search also can be used in step two be rather than the two steps which are mentioned here. We can use the Backtracking line search or exact line search, we studied Backtracking line search in the last class. So, those ideas can we used in step two be of this algorithm.

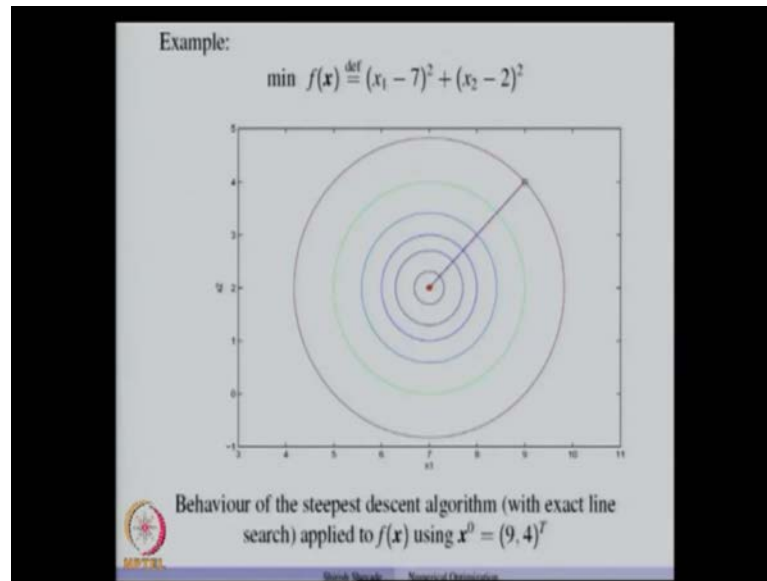
(Refer Slide Time: 04:31)



Now, then we started looking at some of the examples. So, the first example we took was of a function  $f$  of  $\mathbf{x}$ , which is defined as  $x_1$  minus 7 square plus  $x_2$  minus 2 square. Now, the gradient of the function is 2 into  $x_1$  minus 7 and 2 into  $x_2$  minus 2 and the point at which the gradient vanishes is 7 comma 2 or whose  $x$  coordinate is 7  $y$  coordinate is 2. The Hessian matrix is also given here. Now, this is a Quadratic function those. So, Hessian matrix is independent of  $\mathbf{x}$ .

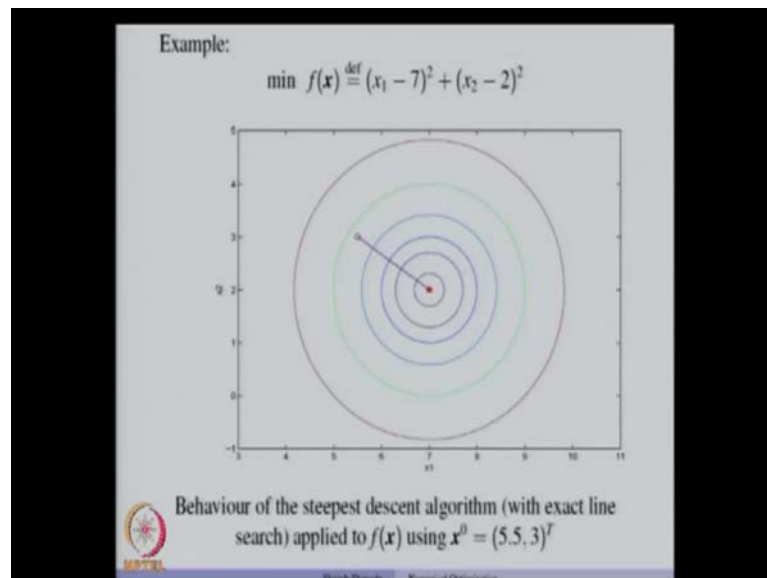
So,  $\mathbf{x}^*$  is a local minimum and if you look at the contours of this objective function. So, here the objective function value is 8. So, this contour, this curve colours corresponds to the objective function value of 8 and then objective function value comes down to 4 to 1.5 and finally, at the 0.7 comma 2 we have the minimum of the objective function and let us apply Steepest Descent Method to this objective function.

(Refer Slide Time: 05:52)



Now, suppose if you start from a point which is 9 comma 4 and applies Steepest Descent Method with exact line search to the objective function, then you will see that in 1 step, we have reached the solution. Now, were we lucky to start with this particular initial point let us find out, let us take another.

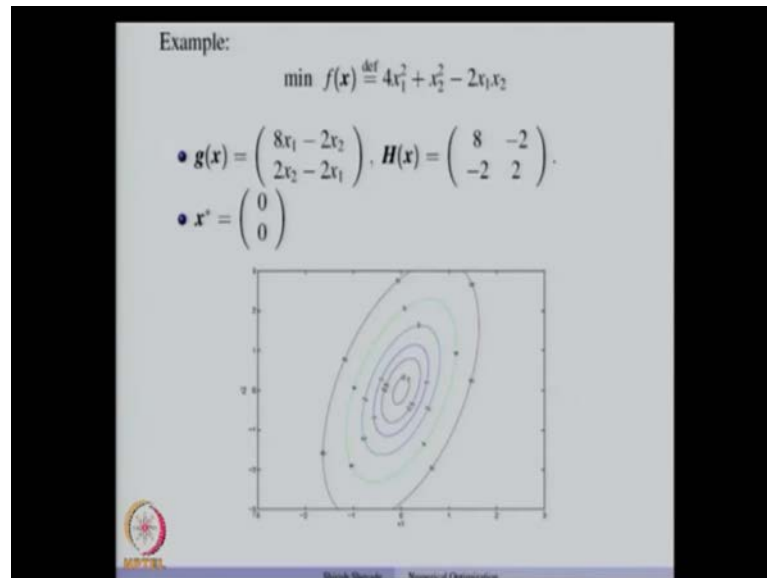
(Refer Slide Time: 06:20)



Let us take the same function but another initial point. So, we have taken the initial point whose x coordinate is 5.5 and y coordinate is 3. And you will see that again in 1 step the Steepest Descent algorithm converges to  $\mathbf{x}^*$ . So, if we have circular contours of a

Quadratic function then the Steepest Descent Method would converge to  $x^*$  in 1 iteration. Now, let us see when these contours of an objective function which is Quadratic are not circular but they are elliptical.

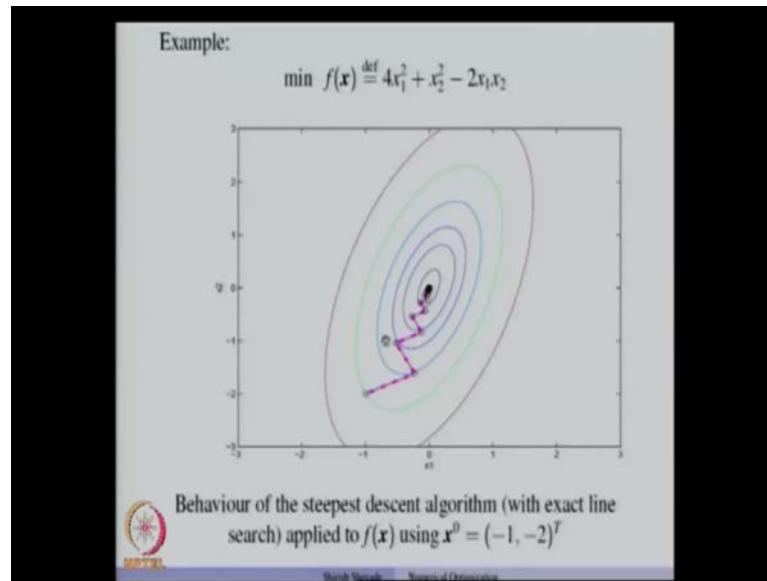
(Refer Slide Time: 07:04)



So, let us look at another example. So, we have a function  $4x_1^2 + x_2^2 - 2x_1x_2$ , the gradient is  $8x_1 - 2x_2$  and  $2x_2 - 2x_1$ . Now, the gradient of this objective function vanishes at the origin. So, that is the stationary point and you will see that the Hessian matrix is all positive definite matrixes. So, the origin is a strict local minimum of this problem. So, let us look at the contours of this objective function so on.

So, we can see this contour plot of  $f$  of  $x$ . So, this is the plot corresponding to the objective function value  $f$  of  $x$  equals to 8 and as you move inside the objective function value decreases. So, from 4 to 1.5, 0.1 and finally, we get a 0 objective function value at the origin. So, origin is a strict local minimum of  $f$  of  $x$ , which is given here. Now, let us apply the Steepest Descent algorithm to this objective function with exact line search.

(Refer Slide Time: 08:30)

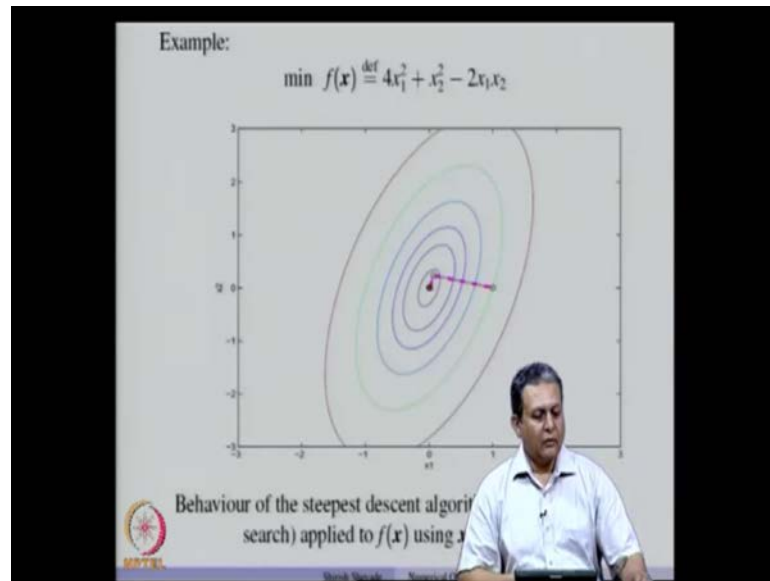


Now, if you start from this point x coordinate is minus 1 and y coordinate is minus 2. So if you apply exact line search. So, the first point that, we reach from this initial point and then we move along this direction and so forth. So, more than 20 iterations were needed in this case to reach from minus 1 to from minus 1 minus 2 to 00. If you use Steepest Descent Methods with exact line search.

Now, you will see that the behaviour, if we apply the Steepest Descent Method in this case, we will see that iterates are found in the zigzag directions, before the convergence to the optimal point x plus. So, this zigzagging behaviour is very typical of Steepest Descent Method for Quadratic functions, where the contours of the Quadratic functions are not circular.

Now, why does this zigzagging takes place? So, we will study that in today's lecture. Now, compare this with the 1 step convergence for the circular contours. So, if the contours are circular starting from any point, we could converge to the minimum in only one iteration while here, if the contours are elliptical then we require more number of iterations.

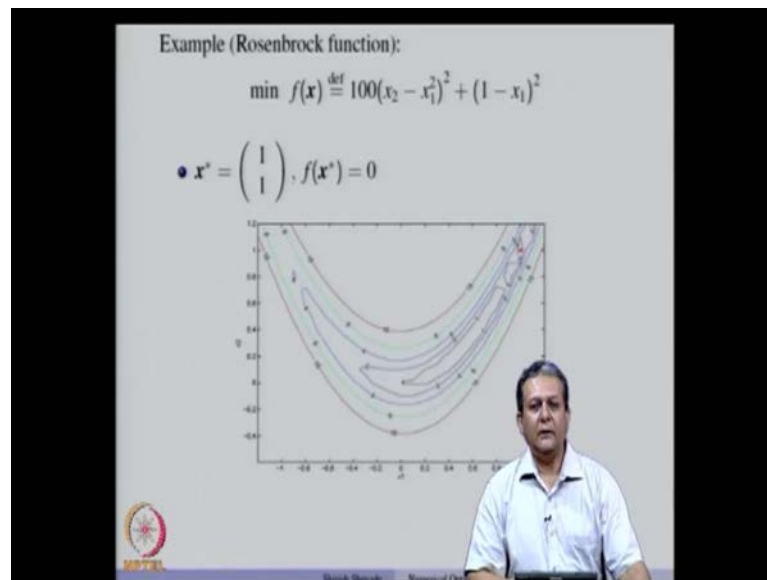
(Refer Slide Time: 10:20)



And we will see that if we start with another point again in this case about 4 iterations where needed, if we started from this points say whose x coordinates is 1 and y coordinates is 0 and about 4 iterations h were needed before the Steepest Descent Method converge to the minimum of this objective function.

So, what is so special about these elliptical contours which make the Steepest Descent Method with exact line search behave in this particular way that is what is the reason behind these zigzagging directions that we get from Steepest Descent Method applied to the quadratic functions which have elliptical contours. So, we will study that now.

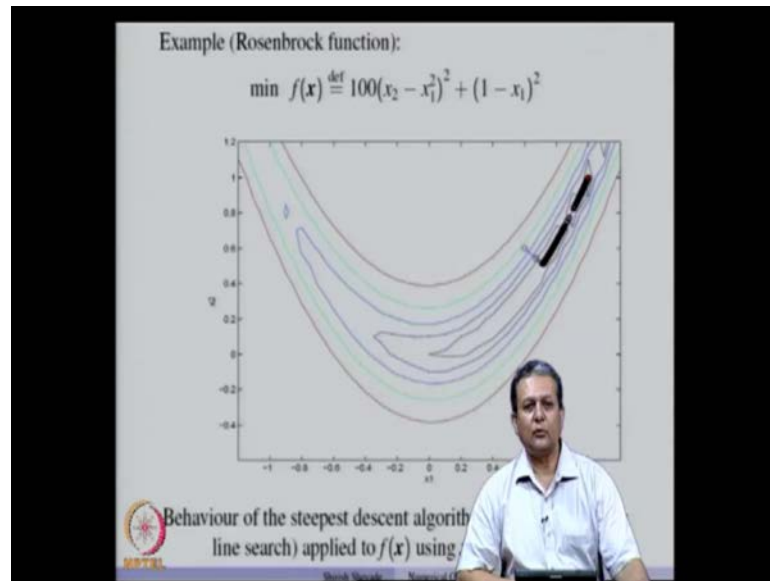
(Refer Slide Time: 11:09)



Before that let us take another example. So, this is standard function which is used in optimization literature to study the performance of different optimization algorithms. And so, the Rosen Brock function which we studied in 1 of the earlier classes is given here and we know that the minimum of this objective function occurs at 1 comma 1 and the minimum objective function value is 0. Let us look at the contours of this. So, the outer most plots correspond to the function value 16 and then come on to 8, 4, 2, and 1. This is the point which is shown by the red star, that point is a minimum of this function. Now, let us applies Steepest Descent Method to the function  $f$  of  $x$  and this time we will use the Backtracking line search.



(Refer Slide Time: 12:18)



So, if we start from a 0.6, 0.6. So, you will see that the first step is to this point and then to this point and then there are lots of small steps. So, there are lots of small steps which correspond to the circles along these lines and then finally, it goes and converges to the minimum which is 1 comma 1. So, lots of small steps are taken before the algorithm converges to the minimum.

Now, here I have used Backtracking line search, I can use any other in exact line search method the number of steps required could reduce but this is just to demonstrate that how the Steepest Descent Method applied to Rosen Brock function with Backtracking Line search behaves when we start from particular 0.6, 0.6. Now, let us take another initial point and see how is behaviour of Steepest Descent Method with Backtracking Line search to Rosen Brock function.

(Refer Slide Time: 13:31)


Example (Rosenbrock function):

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

•  $x^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, f(x^*) = 0$

$k$	$x_1^k$	$x_2^k$	$f(x^k)$	$\ x^k - x^*\ $	$\ g^k\ $
0	0.6	0.6	5.92	0.5657	75.59
10	0.72	0.52	0.0792	0.5601	0.3938
100	0.78	0.61	0.0465	0.4414	0.2451
1000	0.9914	0.9828	$7.45 \times 10^{-5}$	0.0193	0.0069
2028	0.9989	0.9978	$1.81 \times 10^{-6}$	0.0019	$9.97 \times 10^{-4}$

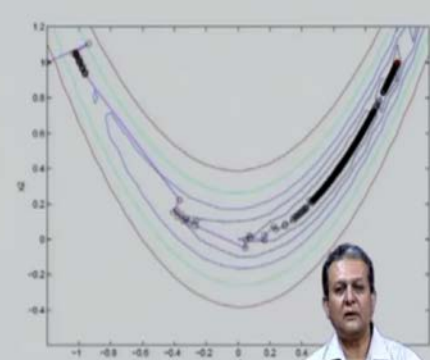
Table: Steepest descent method (with backtracking F  
applied to Rosenbrock function, using  
 $x^0 = (0.6, 0.6)^T, \hat{\alpha} = .5, \rho = .3$  and  $\epsilon_1 = 1.0$




So, this is the Rosen Brock function.

(Refer Slide Time: 13:36)

Example (Rosenbrock function):

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


Behaviour of the steepest descent algorithm  
(line search) applied to  $f(x)$  using



And so, suppose we start from 1 minus 1.2 and 1. So, the first step that is taken is to this point, then the second step comes back to this point and then you will see lot of small steps before there is the algorithm takes a big step to this point and then again lot of small steps, then again a reasonably big step and then lot of small steps. So, you will see that here the steps number, the step size are really small before the finally, the algorithm

converges. So, a lot depends on initial point special the number of steps needed to reach the final point depends a lot on the initial point.

Now, let us go back to the previous example, where we started from 0.6, 0.6 and reach the minimum and let us look at the behaviour of the algorithm for this particular problem.

(Refer Slide Time: 14:40)

Example (Rosenbrock function):

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

•  $x^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, f(x^*) = 0$

k	$x_1^k$	$x_2^k$	$f(x^k)$	$\ x^k - x^*\ $	$\ g^k\ $
0	0.6	0.6	5.92	0.5657	75.59
10	0.72	0.52	0.0792	0.5601	0.3938
100	0.78	0.61	0.0465	0.4414	0.2451
1000	0.9914	0.9828	$7.45 \times 10^{-3}$	0.0192	0.0069
2028	0.9989	0.9978	$1.81 \times 10^{-6}$	0.0024	$9.97 \times 10^{-4}$

Table: Steepest descent method (with backtracking line search) applied to Rosenbrock function, using  $x^0 = (0.6, 0.6)^T, \hat{\alpha} = .5, \rho = .3$  and  $\epsilon_1 = 1.0 \times 10^{-4}$ .

So, here is a table which denotes the iteration number in the first column, then the coordinates at that particular iteration. Then the value of the objective function is the distance between  $x^k$  and  $x^*$ , the Euclidian distance and the norm of  $g^k$ . So, initial  $k$  equal to 0 and we started with the 0.6, 0.6. The objective function value is 55.92. The distance between  $x^k$  and  $x^*$  is 0.5657 and norm of  $g^k$  is 75.59. Now, the Steepest Descent Method with Backtracking line search required 2028 in this case to reach an optimal to reach very close to the optimal point which is 1 comma 1.

So, we will see that, we have reached here to a 0.99, 0.99 which is reasonably close to 11 and the value of the objective function is very close to 0. The distance between  $x^k$  and  $x^*$  is of the order of point 0024 and norm of  $g^k$  is 9.97 into 10 to the power minus 4. So, in this case I have used epsilon to be 10 to the power minus 3 in the Steepest Descent algorithm. So, the algorithm will continue till norm of  $g^k$  becomes less than 10 to the power minus 3. So, you will see that at this step 2028. The algorithm has indeed gone to a stage where the norm of the  $g^k$  goes below 10 to the power minus 3.

So, I showed here some iteration. So, you will see that from 0 from the initial point to the tenth iteration there is quite a bit of reduction in the norm of the gradient from 75 to 0.39 and then from 10 to 100 iteration. There is a gradual in decrease in the norm again from 100 to 1000 there is a gradual decrease. So, the Steepest Descent algorithm in this case required about 2000 to go to the minimum of this. Rosen Brock function starting from 0.6, 2.6 and every time you would see that, the function value has been decreasing in a consistent way although there could be a the rate of decrease could be very small but there has been a consistent decrease in the objective function and the distance for  $x_k$  to  $x^*$  also comes down at a reasonable rate.

So, for the Backtracking line search, we need some parameters like alpha hat and rho. So, in this particular case I had set alpha hat to be point 5 and a rho to be point 3 and the constant c 1 which corresponds to Armijo-Wolfe conditions was set to 10 to the power minus 4. So, this was with respect to the initial 0.6, 0.6. Now, if we turn to the other example, where the initial point was minus 1.2 and 1 and as we have seen here it requires lot more steps to go to the minimum.

(Refer Slide Time: 18:20)

Example (Rosenbrock function):

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

•  $x^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, f(x^*) = 0$

k	$x_1^k$	$x_2^k$	$f(x^k)$	$\ x^k - x^*\ $	$\ g^k\ $
0	-1.2	1.0	24.2	2.2	232.87
10	-1.00	1.01	4.02	2.0042	7.69
100	0.57	0.32	0.1867	0.80	0.84
1000	0.99	0.97	$1.99 \times 10^{-4}$	0.0314	0.014
2300	0.9989	0.9979	$1.11 \times 10^{-6}$	0.0024	$9.63 \times 10^{-4}$

Table: Steepest descent method (with backtracking lines search) applied to Rosenbrock function, using  $x^0 = (-1.2, 1)^T, \hat{\alpha} = .5, \rho = .3$  and  $c_1 = 1.0 \times 10^{-4}$ .

So, let us study the behaviour of this algorithm. So, you will see that, the algorithm required about 2300 iterations before it reach close to the minimum. Remember that minimum is 11 and the point that we have reached this 0.9989, 0.9989, 9979 which is reasonable close to 1. So, the distance between the  $x_k$  and  $x^*$  is about point 0024.

And as in the previous case, these two columns denote the coordinates of points at the  $k$ -th iteration, this column denotes the distance between  $x_k$  and  $x^*$  and this column denotes the norm of  $g_k$ . So, you will see that initial norm of  $g_k$  when we started with this point the norm of  $g_k$  was 232.87 and that came down to 7.69 and then 0.4 finally, when the algorithm terminated the norm of  $g_k$  was less than  $10^{-3}$  and in this particular case the value was  $9.63 \times 10^{-4}$ , you will also see, that the distance of  $x_k$  from  $x^*$  gradually came down to 0.0024 and the value of the objective function finally, was almost close to 0 which is the desired objective function value.

Again in this Steepest Descent Method, I have used Backtracking line search there  $\alpha$  hat again was chosen to be 0.5 row was 0.3 and then the constants  $c_1$  was like in the previous case said to  $10^{-4}$ . So, you will see that, the number of iterations needed depend a lot on the initial  $x_0$ . So, you can also try applying Steepest Descent Method to different objective functions and study its behaviour.

(Refer Slide Time: 20:32)

**Convergence of Steepest Descent Method: Quadratic case**

Consider the problem:

$$\min_{x \in \mathbb{R}^n} f(x) \stackrel{\text{def}}{=} \frac{1}{2} x^T H x - c^T x$$

where  $H$  is a symmetric positive-definite matrix.

- $g(x) = Hx - c. \therefore x^* = H^{-1}c.$
- How does steepest descent method perform when applied to  $f(x)$ ?
- Assume that *exact line search* is used in each iteration.

Now, let us look at the Quadratic case or Quadratic function case and see how the Steepest Descent Method converges to the minimum. Now, this particular case is very important, because even if a function is non-quadratic then and somewhere near the optimal point  $x^*$ , the behaviour of the function is quadratic. So, it is important to

study the behaviour of the Steepest Descent Method for Quadratic case first before we move on to non-quadratic case.

So, let us consider the problem to minimise  $f$  of  $x$  where  $f$  of  $x$  is defined as  $\frac{1}{2} x^T H x - c^T x$  and  $H$  is the symmetric positive definite matrix. So, this is a convex Quadratic function and. So, the reason for studying these Quadratic functions is that for many functions even if they are non-quadratic near  $x^*$  or near their minimum, the behaviour of the function is like a Quadratic function. So, it is important to study the convergence of Steepest Descent Method for a Quadratic case. So, that is why I have chosen a Quadratic function here first. Now, we know that the gradient of this function is  $H x - c$  and if we set the gradient to 0. So,  $H x - c = 0$  gives  $x^*$  to be  $H^{-1} c$ . So, this is other optimal point and the since it is a positive  $H$  is a positive definite matrix the inverse of  $H$  is possible and we get  $x^*$  to be  $H^{-1} c$ .

Now, you want to see how does Steepest Descent Method perform? Apply as applied to this particular function  $f$  of  $x$ . So, in the Steepest Descent Method we have seen that in every iteration the direction chosen is a negative of the gradient direction. Now, for a Quadratic function like this we can also try using exact line search, because one can work out the exact formula for  $\alpha_k$  at every iteration. So, let us assume that we use exact line search in each iteration. So, this is an important assumption that we make that exact line search is done in every iteration and we uses Steepest Descent directions in every iteration. Now with this let us study the behaviour of Steepest Descent Method as applied to this convex Quadratic function.

(Refer Slide Time: 23:30)

What is the step length  $\alpha^k$  at iteration  $k$ ?

$f(x) = \frac{1}{2}x^T Hx - c^T x. \therefore g^k = g(x^k) = Hx^k - c$

Define  $\phi(\alpha) = f(x^k + \alpha d^k) = f(x^k - \alpha g^k)$ .

Exact line search:

$$\alpha^k = \arg \min_{\alpha > 0} \phi(\alpha)$$
$$\phi'(\alpha) = 0 \Rightarrow \nabla f(x^k - \alpha g^k)^T (-g^k) = 0$$
$$\Rightarrow (Hx^k - \alpha Hg^k - c)^T g^k = 0$$
$$\Rightarrow (g^k - \alpha Hg^k)^T g^k = 0$$

Therefore,

$$\alpha^k = \frac{g^{kT} g^k}{g^{kT} H g^k}$$
$$\therefore x^{k+1} = x^k - \left( \frac{g^{kT} g^k}{g^{kT} H g^k} \right) g^k$$

As I mentioned that it is easy to calculate the step length alpha k at every iteration k, if we use exact line search for Quadratic function. So, let us see how to do that. So, we have f of x to be half of x transpose H x minus c transpose x and the gradient at the point x k, we are going to denote it as denoted by g k and that gradient is H x k minus c. Now, the Descent direction is d k is minus g k. So, once we chose the direction, we need to find out what is the value of the step length. So, for that purpose we define a function phi alpha which is f of x k plus alpha d k and we want to find out. What is the value of alpha? Or what is the value of positive alpha such that, these functions are minimised. So that positive value of alpha which minimises this function will be the alpha k for this case.

Now, since we are talking about the Steepest Descent Method d k is nothing but minus g k. So, phi alpha is nothing but f of x k minus alpha g k and in the exact line search the alpha k is determined by minimising phi alpha. Now, in our case f is a Quadratic function. So, phi alpha also will be a Quadratic function of alpha and its minimum can be easily found out by setting phi dash alpha to 0 and setting phi dash alpha to 0. So, what we get is the gradient of f at x 2 minus alpha g k transposes minus g k equal to 0 and if you rearrange the terms.

So, gradient of  $f$  of  $x^k$  minus  $\alpha g^k$  is nothing but  $H x^k$  minus  $\alpha H g^k$  minus  $c$  transpose  $g^k$ . Now, because remember that, the gradient of  $x$  at  $k$  is  $H x^k$  minus  $c$ . So, we can use this same formula here with  $x^k$  replaced by  $x^k$  minus  $\alpha g^k$ .

Now, if we combine  $H x^k$  minus  $c$ . So, what we get is that is nothing but the gradient at  $x^k$  and which we can write it as  $g^k$ . So, this quantity can be written as  $g^k$  minus  $\alpha H g^k$  transpose  $g^k$  equal to 0. So,  $H x^k$  and minus  $c$  are combined to get  $g^k$  and this gives us the formula for  $\alpha^k$  which is nothing but  $g^k$  transpose  $g^k$  divided by  $g^k$  transpose  $H g^k$ . Now, remember that  $H$  is a positive definite matrix which is symmetric. So,  $g^k$  transpose  $H g^k$  is greater than 0, if  $g^k$  is not equal to 0 and  $g^k$  transpose  $g^k$  is a norm square that is also quantity which is greater than 0 if  $g^k$  not equal to 0. So, if  $g^k$  is not equal to 0  $\alpha^k$  is greater than 0 and then we can use this  $\alpha^k$  to determine our  $x^{k+1}$  the new point. So, the new point  $x^{k+1}$  is nothing but the old point minus  $\alpha^k g^k$ . So, the  $\alpha^k$  is the quantity which we derived just now into  $g^k$ . So, that gives us the new point  $x^{k+1}$ .

(Refer Slide Time: 27:31)

At what rate does  $\{x^k\}$  converge?

Define  
 $E(x^k) = \frac{1}{2}(x^k - x^*)^T H(x^k - x^*)$ . ( $E(x^k) > 0$ , if  $x^k \neq x^*$ )

Note that  $E(x^k) = f(x^k) + \underbrace{\frac{1}{2}x^{*T} H x^*}_{\text{constant}}$ .

Define  $y^k = x^k - x^*$ .  $\therefore H y^k = g^k$ . Using

$$x^{k+1} = x^k - \left( \frac{g^{kT} g^k}{g^{kT} H g^k} \right) g^k.$$

Relative decrease in  $E$ ,

$$\frac{E(x^k) - E(x^{k+1})}{E(x^k)} = \frac{(x^k - x^*)^T H(x^k - x^*) - (x^{k+1} - x^*)^T H(x^{k+1} - x^*)}{y^{kT} H y^k}$$

Now, once we find the direction  $d^k$  which is minus  $g^k$  and find  $\alpha^k$  using close form solution that we have seen. Now, we are interested in finding out at what rate does  $x^k$  converge. So, let us define a function  $e$  of  $x^k$  to be half of  $x^k$  minus  $x^*$  transpose  $H$  into  $x^k$  minus  $x^*$ . Now, this function you will see that, this is function is greater than 0 if  $x^k$  is not equal to  $x^*$ , that is because  $H$  is a positive definite matrix. So,  $e$  of



$x_k$  is always greater than 0. So, that is 1 important point about this function  $e$ . We can also call it as an error function because we can write  $e$  of  $x_k$  as  $f$  of  $x_k$  plus some constant.

So, as we have seen earlier that sometimes the behaviour of an optimization algorithm can be determined using some error function and an error function can be the objective function that we want to optimize. So, minimisation of  $f$  of  $x$  will be same as minimisation of  $e$  of  $x$  because  $f$  of  $x$  is just added to this constant quantity to get  $e$  of  $x$ . So, let us now study the behaviour of Steepest Descent algorithm with respect to this function  $e$  of  $x$  and remember that, we are not losing anything, because  $e$  of  $x$  is nothing but  $f$  of  $x$  plus constant the reason why  $e$  of  $x$  is chosen is that because of the nice form it has. So, this function is positive when  $x_k$  not equal to  $x^*$  and when  $x_k$  equal to  $x^*$ , the function is 0. So, we will see what happens to  $e$  of  $x_k$  when the Steepest Descent Method is applied to minimise  $f$  of  $x$ .

Now, to study that let us first define a variable  $y_k$  to be  $x_k$  minus  $x^*$  and. So, we can write  $H y_k$  to be  $H x_k$  minus  $H x^*$  and  $H x^*$  is nothing but  $c$ . So,  $H x_k$  minus  $c$  is nothing but the gradient of the objective function and  $x_k$  and that is nothing but  $g_k$ . So, we will use this property that  $H y_k$  equal to  $g_k$  in deriving this expression. Now, we have already seen that the new point  $x_{k+1}$  is obtained by  $x_k$  minus  $\alpha_k g_k$  and  $\alpha_k$  has a close form solution which is  $g_k^T g_k$  divided by  $g_k^T H g_k$ .

Now, if we use this then we are interested in finding out what is the relative decrease in the function  $e$  when we go from  $x_k$  to  $x_{k+1}$ . So, in other words we are interested in finding out what is  $e$  of  $x_k$  minus  $e$  of  $x_{k+1}$  by  $e$  of  $x_k$  remember that we are trying to minimise the function  $e$ . So,  $e$  of  $x_k$  is greater than  $e$  of  $x_{k+1}$ , because we are minimising the function  $e$  secondly  $e$  of  $x_k$  is always greater than 0 when  $x_k$  not equal to  $x^*$ . So, assuming that  $x_k$  is not equal to  $x^*$  this quantity is going to be always positive and we want to see how it approaches towards the how the sequence  $x_k$  approaches towards the optimal point  $x^*$  now.

So, if you use this definition of  $e$  of  $x_k$  and since, the factor half is common both in the numerator and denominator. So, we can get rid of that factor and what we get is  $x_k$  minus  $x^*$  transpose  $H$  into  $x_k$  minus  $x^*$  minus  $e$  of  $x_{k+1}$  is nothing but  $x_k$

plus 1 minus  $x^*$  transpose  $H$  into  $x^k$  plus 1 minus  $x^*$  and in the denominator I have replaced  $x^k$  minus  $x^*$  by  $y^k$ . So, what we have is  $y^k$  transpose  $H$  into  $y^k$ . Now, let us simplify this expression further.

(Refer Slide Time: 32:20)

$$\frac{E(x^k) - E(x^{k+1})}{E(x^k)}$$

$$= \frac{(x^k - x^*)^T H (x^k - x^*) - (x^{k+1} - x^*)^T H (x^{k+1} - x^*)}{y^{kT} H y^k}$$

$$= \frac{2\alpha^k g^{kT} g^k - \alpha^{k2} g^{kT} H g^k}{y^{kT} H y^k}$$

Substituting  $\alpha^k = \frac{g^{kT} g^k}{g^{kT} H g^k}$ , we get

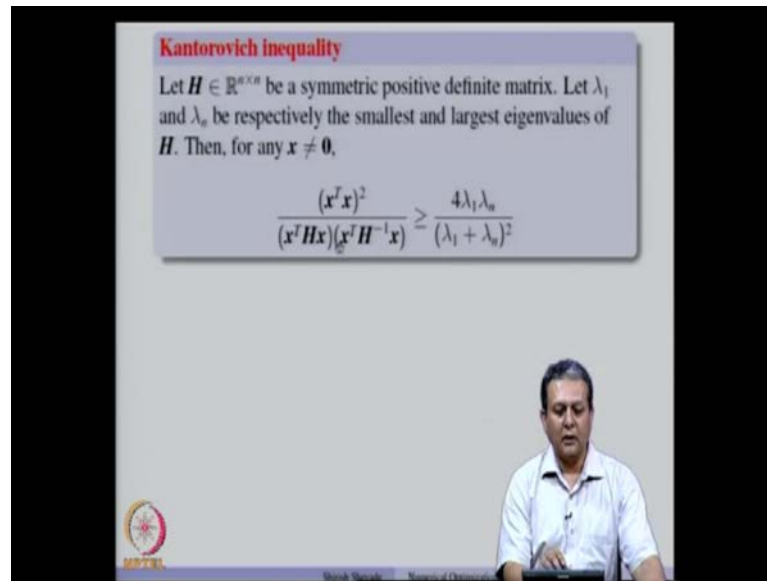
$$\frac{E(x^k) - E(x^{k+1})}{E(x^k)} = \frac{(g^{kT} g^k)^2}{(g^{kT} H g^k)(g^{kT} H^{-1} g^k)}$$

So, if you simplify this what we get is  $2\alpha^k g^k$  transpose  $g^k$  minus  $\alpha^k$  square  $g^k$  transpose  $H g^k$  divided by  $y^k$  transpose  $H y^k$  and. Now, we can plug in the value of  $\alpha^k$  that we have already found for this Quadratic function and if you plug in this value of  $\alpha^k$  that we have already found here what we get is  $e$  of  $x^k$  minus  $e$  of  $x^k$  plus 1 divided by  $e$  of  $x^k$  which is a relative decrease in  $e$  is nothing but this quantity.

Now, this quantity is still dependent on the matrix  $H$  and  $H$  inverse. So, if you look at this expression this is  $g^k$  transpose  $H$  inverse  $g^k$  so, this relative decrease in  $e$  depends on  $H$  inverse and  $g^k$  now can we get a bound on this relative decrease which is independent of this  $h$ .

Now, 1 possible thing that we can do is that we can chose  $g^k$  to be a norm 1 vector. So, that  $g^k$  transpose  $g^k$  is 1 now can we get some bound on this which depends on  $H$  and  $H$  inverse now if you can do that then this we can bound the relative decrease in the objective function now for that purpose we will need what is called Kantorovich inequality.

(Refer Slide Time: 34:03)



**Kantorovich inequality**

Let  $H \in \mathbb{R}^{n \times n}$  be a symmetric positive definite matrix. Let  $\lambda_1$  and  $\lambda_n$  be respectively the smallest and largest eigenvalues of  $H$ . Then, for any  $x \neq 0$ ,

$$\frac{(x^T x)^2}{(x^T H x)(x^T H^{-1} x)} \geq \frac{4\lambda_1 \lambda_n}{(\lambda_1 + \lambda_n)^2}$$

So, let us look at Kantorovich inequality. So, let  $H$  be a symmetric positive definite matrix and let  $\lambda_1$  and  $\lambda_n$  be the respective the smallest and largest Eigen values of  $H$  then for any  $x$  which is non-zero this inequality holds. So,  $x$  transpose  $x$  square divided by  $x$  transpose  $H x$  into  $x$  transpose  $H$  inverse  $x$  is greater than or equal to  $4$  into  $\lambda_1 \lambda_n$  divided by  $\lambda_1 + \lambda_n$  square where  $\lambda_1$  is the smallest Eigen value of  $H$  and  $\lambda_n$  is the largest Eigen value of  $H$ .

Now, if you compare this left hand side expression with the expression that we got here. We will see that they are same where  $x$  is replaced by  $g_k$  and remember that our algorithm converges only at the point when  $g_k$  is close to  $0$ . So, as long as  $g_k$  is not  $0$  we can write this quantity to be greater than or equal to  $4 \lambda_1 \lambda_n$  divided by  $\lambda_1 + \lambda_n$  square.

(Refer Slide Time: 35:12)

**Kantorovich inequality**

Let  $H \in \mathbb{R}^{n \times n}$  be a symmetric positive definite matrix. Let  $\lambda_1$  and  $\lambda_n$  be respectively the smallest and largest eigenvalues of  $H$ . Then, for any  $x \neq 0$ ,

$$\frac{(x^T x)^2}{(x^T H x)(x^T H^{-1} x)} \geq \frac{4\lambda_1 \lambda_n}{(\lambda_1 + \lambda_n)^2}$$

Using this inequality,

$$\frac{E(x^k) - E(x^{k+1})}{E(x^k)} = \frac{(g^k)^T g^k}{(g^k)^T H g^k (g^k)^T H^{-1} g^k} \geq \frac{4\lambda_1 \lambda_n}{(\lambda_1 + \lambda_n)^2}$$

Therefore,

$$E(x^{k+1}) \leq \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 E(x^k)$$

Now, this inequality can be derived in different ways there exist different proofs to prove this inequality. So, 1 of the proofs what it assumes is that  $x^T x = 1$  and then it tries to maximise this  $x^T H x$  into  $x^T H^{-1} x$  subject to the concern that  $x^T x = 1$ . So, for that proofs we requires some knowledge about how to solve a constrained optimization problem. So, let us postponed the discussion on Kantorovich inequality till be study something about constant optimization problem. So, for the time being let us assumed that this inequality holds and we can use it for our purpose. So, if you use this inequality then the relative decrease in function  $e$  which is nothing but  $\frac{(g^k)^T g^k}{(g^k)^T H g^k}$  by  $\frac{(g^k)^T H^{-1} g^k}{(g^k)^T H g^k}$  into  $\frac{(g^k)^T g^k}{(g^k)^T H g^k (g^k)^T H^{-1} g^k}$  which same as these when  $g^k$  is not 0 and that is bounded by  $\frac{4\lambda_1 \lambda_n}{(\lambda_1 + \lambda_n)^2}$  now the  $H$  on the right side of this equation are replaced by the Eigen the smallest and the largest Eigen values of  $h$ .

So, therefore, we can rewrite this expression as  $e$  of  $x^k + 1$  is less than or equal to  $\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \frac{e(x^k)}{E(x^k)}$ . Now,  $H$  is a positive definite matrix. So, all its Eigen values are positive. So,  $\lambda_1$  and  $\lambda_n$  both are positive and  $\lambda_n$  is greater than  $\lambda_1$ . So, if you look at the expression here  $\lambda_n - \lambda_1$  is a positive quantity  $\lambda_n + \lambda_1$  is a positive quantity and more over  $\lambda_n - \lambda_1$  is less than  $\lambda_n + \lambda_1$ . So, that is why  $\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}$  is a positive fraction. So, you will see that  $e$  of  $x^k + 1$  is less than or equal to this fraction square

into  $e$  of  $x^k$ . So, that means that  $e$  of  $x^{k+1}$  is a decreasing sequence and  $e$  of  $x^{k+1}$  is less than  $e$  of  $x^k$ .

(Refer Slide Time: 38:01)

$$E(x^{k+1}) \leq \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 E(x^k)$$

Therefore,  $E(x^k) \rightarrow 0$  and  $x^k \rightarrow x^*$  ( $H$  is positive definite).  
 With respect to  $E$ , the steepest descent method

- converges linearly with convergence rate no greater than  $\left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2$
- Actual convergence rate depends upon  $x^0$
- Define the *condition number* of  $H$ ,  $r = \frac{\lambda_n}{\lambda_1}$
- Convergence rate of the steepest descent method depends on the condition number of  $H$ 
  - $r = 1$  (circular contours)  $\Rightarrow$  convergence in one iteration
  - $r \gg 1$  (elliptical contours)  $\Rightarrow$  convergence is slow
- For nonquadratic functions, rate of convergence to  $x^*$  depends on the condition number of  $H(x^*)$

So, therefore, we have a decreasing sequence and  $e$  of  $x^k$  is a quantity which is bounded below by 0. So, we have decreasing sequence of  $e$  of  $x^k$ , that sequence will tend to 0 and  $x^k$  will tend to  $x^*$ . So, when this  $e$  of  $x^k$  goes to 0 we have  $x^k$  which goes to  $x^*$  because  $H$  is a positive definite matrix. So, with respect to  $e$  this Steepest Descent Method converges linearly. So, you can see that the definition of linear convergence holds with respect to  $e$  when Steepest Descent Method applied is applied to a Quadratic convex Quadratic objective function.

Now, the convergence rate is not is no greater than this quantity which is  $\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}$  square. So, that means the convergence rate depends on the smallest and the largest Eigen values of the Hessian matrix of the Quadratic function and the rate is not greater than this quantity. The actual convergence rate it does depends upon  $x^0$ . So, we saw earlier that different  $x^0$  for elliptical contours will result in different convergence rates for a Quadratic functions. Now, if we define the condition number of the matrix  $H$  to be the ratio of the largest and the smallest Eigen value of the matrix  $H$ . So, let us denote this ratio by  $r$  then the convergence rate of a Steepest Descent Method it depends on  $\frac{r-1}{r+1}$  or the condition number of the matrix  $H$ .

So, this is the very important observation that the condition number of the hessian matrix of a Quadratic function decides the convergence rate of the Steepest Descent Method. Now, when this ratio is 1, then  $\lambda_n$  equal to  $\lambda_1$  then what we have is this quantity is 0 and then  $e$  of  $x_k$  plus 1 will be 0. So, irrespective of your initial point 1 as the goes to the minimum in 1 iteration and when is this ratio 1, when  $\lambda_n$  equal to  $\lambda_1$  that means when we have circular contours. So that is why we saw that, when we considered circular contours and wanted to minimise the objective function using Steepest Descent Method the convergence took place only in 1 iteration irrespective of the starting point.

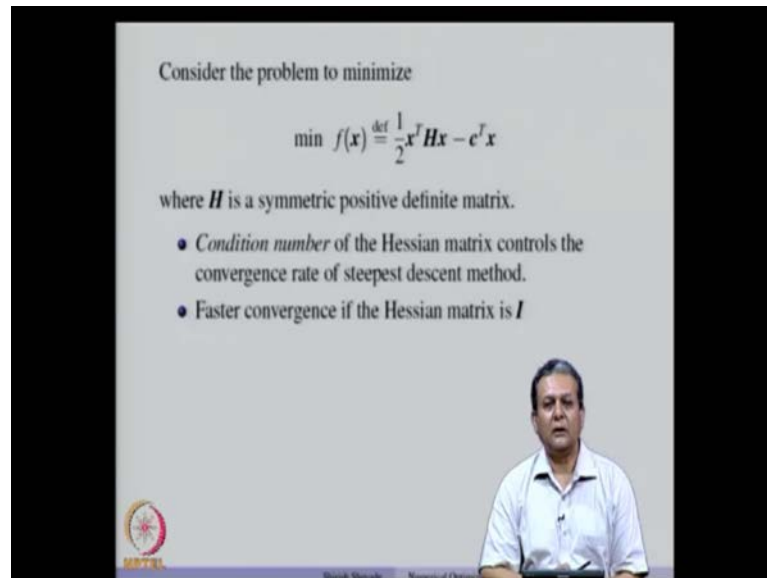
So, if we have  $r$  equal to 1 which corresponds to circular contours, then the convergence of Steepest Descent Method required only one iteration on the other hand if  $r$  is much greater than 1. So, which corresponds to the elliptical contours and in that case we saw that the convergence required many more iterations before the algorithm reach the point  $x^*$ . So, in that case the convergence is very slow. So, the convergence of the Steepest Descent Method although it is a linear convergence, the rate of the convergence depends on the nature of the contours of the Quadratic function. So, if we have circular contours, we have convergence in one iteration irrespective of the starting point and if we have elliptical contours then the convergence is slow lot depends on the initial point.

Now, for non-Quadratic functions as I said that at this minimum  $x^*$  the behaviour of the function is typically like a Quadratic function and therefore, the rate of convergence of Steepest Descent Method to  $x^*$  depends upon the condition number of the hessian matrix evaluated at  $x^*$  so, this is a very important point for non-quadratic function and the theory that we saw with respect to the Quadratic functions typically holds near the point  $x^*$ . So, that is why it was important to study this theory so that the result can be used for non-quadratic functions as well.

Now, if we consider the same example that we saw earlier. So, we have taken the function  $x_1$  minus 7 square plus  $x_2$  minus 22 square and we have applied Steepest Descent algorithm with exact line search and you will see that, it converge to the optimum point in exactly 1 iteration and then we also saw other example, where if we start from some other point it again reaches the convergence in 1 iteration.

Now, if we look at this objective function, whose contours are elliptical, we saw that the zigzagging phenomenon occurs and the Steepest Descent Method. In this case required many more iterations before it converge to the minimum. Now, what is so special about the circular contours which made the Steepest Descent Method converge in 1 step, while in elliptical contours, it required more iteration. So, let us try to analyse that.

(Refer Slide Time: 44:25)



Consider the problem to minimize

$$\min f(x) \stackrel{\text{def}}{=} \frac{1}{2} x^T H x - c^T x$$

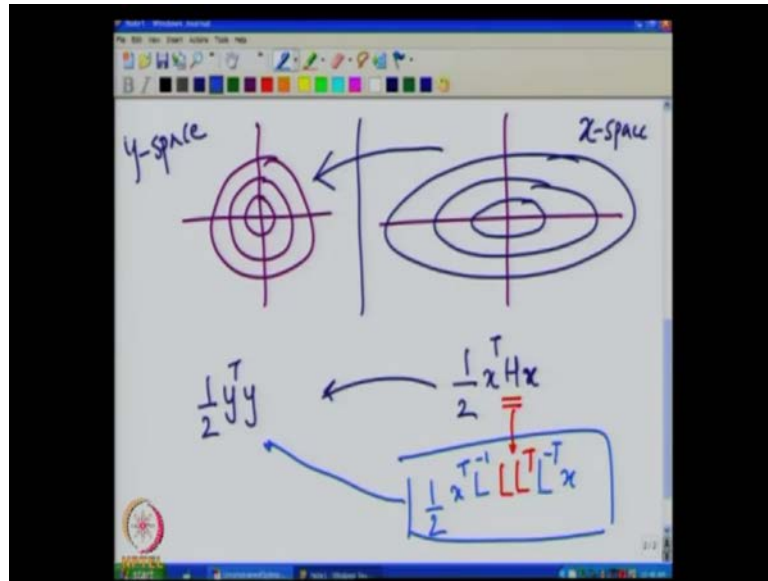
where  $H$  is a symmetric positive definite matrix.

- Condition number of the Hessian matrix controls the convergence rate of steepest descent method.
- Faster convergence if the Hessian matrix is  $I$

The slide also features a logo in the bottom left corner and a speaker in the bottom right corner.

So, let us consider the problem to minimise  $f$  of  $x$  to be half  $x$  transpose  $H$   $x$  minus  $e$  transpose  $x$ , where  $H$  is a symmetric positive definite matrix and we saw that the condition number of the Hessian matrix  $H$  which is a positive definite matrix. In this case controls the convergence rate of Steepest Descent Method and we also saw that if the hessian matrix is identity matrix, that means that the condition number is 1 then the convergence of the Steepest Descent Method is fast. Now, can we use this fact to derive a method which will be faster even if we use Steepest Descent Methods. So, let us consider a simple case.

(Refer Slide Time: 45:27)



So, suppose what we have here in the first case, we had circular contours and in the second case, suppose we have elliptical contours. Now, if we have for a Quadratic function, if we have elliptical contours like this. We know that the Steepest Descent Method does have slow convergence especially when the contours are elongated. Now, can we convert or transform these contours to the circular contours? So that if you use, a Steepest Descent Method we will get the convergence in 1 iteration in other words. So, suppose we have a function which is half of  $x^T H x$ .

Suppose we have a simple function like this. Now, can we convert it to a function where we can write it as  $y^T y$ ? So, suppose we have done some transformations. So, let us call this as  $x$ -space and this is the contours in the  $y$ -space. So, if we transform  $x$  to  $y$  in such a way that for the same objective function the hessian matrix in the  $y$ -space is an identity matrix. Now, if you are able to do that, then we can apply Steepest Descent Method. In this case and get the convergence in 1 step. Now, how do we do this transformation? So, for that purpose we need to look at this matrix  $H$ .

Now, suppose if we can write  $H$  to be  $L L^T$  transpose, using the Cholesky decomposition of  $H$  and then to convert  $x$  to  $y$ . Suppose we use a transformation, which is suppose the following. So,  $L^T x$  here and then we have  $x^T$ , then suppose if we define  $y$  to be  $L^T x$ , then what happens? So, then this expression. So, what we get is a transformation to the  $y$ -space. So, because this  $L^T$  and  $L$  minus



transpose get cancelled and L inverse L will get cancelled and what we get is a circular contour. So, we will see more about this now.

(Refer Slide Time: 49:50)

Consider the problem to minimize

$$\min f(x) \stackrel{\text{def}}{=} \frac{1}{2} x^T H x - c^T x$$

where  $H$  is a symmetric positive definite matrix.

- *Condition number* of the Hessian matrix controls the convergence rate of steepest descent method.
- Faster convergence if the Hessian matrix is  $I$
- Let  $H = LL^T$  be the Cholesky decomposition of  $H$
- Define  $y = L^T x$ . Therefore, the function  $f(x)$  transformed to the function  $h(y)$ .

$$h(y) \stackrel{\text{def}}{=} f(L^{-T}y)$$

So, let us consider the Cholesky decomposition of the matrix H, which is L L transpose and it is defined y to be L transpose x. Now, if we define y like this, then the function f x gets transformed to the function h of y and the h of y is defined as f of x and x is nothing but L minus transpose y.

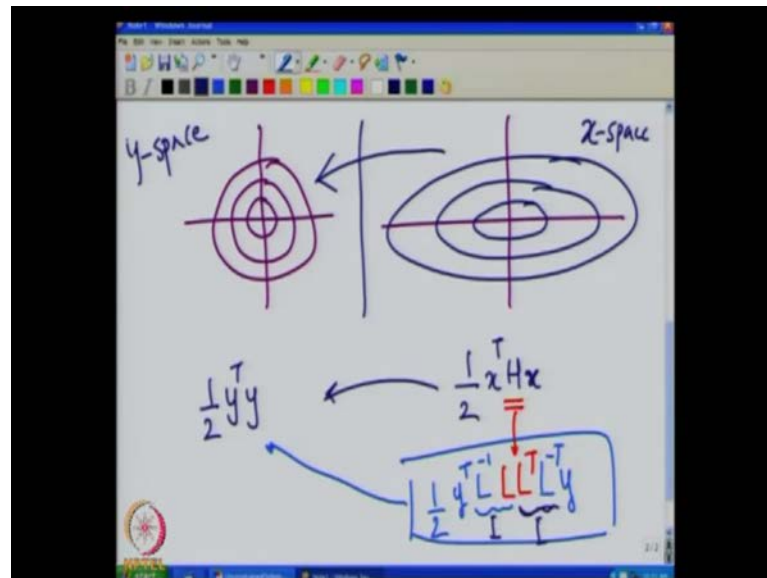
(Refer Slide Time: 50:22)

$$h(y) = f(L^{-T}y)$$

$$= \frac{1}{2} y^T L^{-1} H L^{-T} y - c^T L^{-T} y$$

Now, if we do this then  $h^T y$  is nothing but  $f$  of  $L^{-T} y$  and. Now, if we apply the definition of usual definition of  $f$  of  $x$ . So, what we get is  $y^T L^{-1} H L^{-T} y$ .

(Refer Slide Time: 50:44)



So, this should have been  $y$  and this also should have been  $y$  and then what we get is  $L^{-1} L$  is identity and this is identity and therefore, what we get is half  $y^T y$ . So, if  $x$  is written as  $L^{-T} y$  and  $H$  is decomposed to  $L L^T$ , then because of the nature of this factorisation. What we get is that, these terms get cancelled and finally, what we get is  $y^T y$ . So, because of this transformation we were able to transform the points here in the  $x$ -space to  $y$ -space such that the original contours which were elliptical they get converted to the circular contours or the spherical contours in the high dimensional space circular contours in the two-dimensional space and that was possible because of the way  $H$  was decomposed and the transformation  $x$  was done like  $x = L^{-T} y$ .  $x$  was transformed to  $L^{-T} L^{-1} L^T L^{-1} y$ . So, because of this transformation the things become easier and then I can apply the Steepest Descent Method here.

(Refer Slide Time: 52:31)

$$\begin{aligned}
 h(y) &= f(L^{-T}y) \\
 &= \frac{1}{2}y^T L^{-1} H L^{-T} y - c^T L^{-T} y \\
 &= \frac{1}{2}y^T L^{-1} L L^T L^{-T} y - c^T L^{-T} y \\
 &= \frac{1}{2}y^T y - c^T L^{-T} y
 \end{aligned}$$

- The Hessian matrix of  $h(y)$  is  $I$
- Let us apply steepest descent method in  $y$ -space

$$\begin{aligned}
 y^{k+1} &= y^k - \nabla h(y^k) \\
 &= y^k - L^{-1} \nabla f(L^{-T} y^k) \\
 \therefore L^{-T} y^{k+1} &= L^{-T} y^k - L^{-T} L^{-1} \nabla f(L^{-T} y^k) \\
 \therefore x^{k+1} &= x^k - H^{-1} \nabla f(x^k)
 \end{aligned}$$

So, this is the transformation that took place. So, here we have also included term the  $c$ . term minus  $c$  transpose  $x$ . So,  $L$  inverse transpose  $y$  will replaced  $x$ , in this expression and therefore, what we have is something which is very nice as I mentioned here, earlier that  $L$  inverse,  $L$  will get cancelled,  $L$  transpose,  $L$  inverse transpose,  $L$  transpose inverse will get cancelled and finally, what we get is a Quadratic in  $y$ , where the Hessian matrix here is the identity matrix. Now, that is very important because now, if we apply the Steepest Descent Method in the  $y$ -space. So, what we get is  $y^k + 1$  is nothing but  $y^k$  minus gradient of  $H$  of  $y^k$  that is ours standard Steepest Descent Method with  $\alpha^k$  equal to 1. So, we are not using any line search in this case or in shot we are just setting  $\alpha^k$  to 1.

Now, if we expand it further, if we take the gradient of  $H$  with respect to  $y$  it will be  $L$  inverse into gradient of  $f$  of  $L$  transpose inverse  $y^k$  and now, what happens in the  $x$ -space? So, if you want to consider that, then what we have to do is that we have to pre multiply the whole quantity by  $L$  transpose inverse. So, if we pre multiply this expression by  $L$  transpose inverse, then what we get is  $L$  transpose inverse into  $y^k + 1$  is nothing but  $L$  transpose inverse into  $y^k$  minus  $L$  transpose inverse into  $L$  inverse gradient of  $f$  of  $L$  transpose inverse  $y^k$ . So, now what happens here? So, what is this quantity  $L$  minus  $L$  transpose inverse into  $L$  inverse? Now, if you look back the way  $H$  was defined was  $L L$  transpose. So, if you take an inverse of this quantity  $L L$  transpose, what we get is  $L$  transpose inverse into  $L$  inverse, which is a quantity here. So, this

quantity is nothing but the inverse of the Hessian and what are these quantities? These quantities are nothing but  $x^{k+1}$  and  $x^k$  and suppose, if you replace  $L$  minus  $L$  transpose inverse  $y^k$  by  $x^k$ , then what we get is  $x^{k+1}$  to be  $x^k$  minus  $H$  inverse the gradient of  $f$  of  $x^k$ .

So, the usual gradient direction gets deflected the usual negative gradient direction gets deflected by the matrix  $H$  inverse and that is used to move from  $x^k$  to  $x^{k+1}$  with the step length of 1. So, we have come out with a new method, where instead of the usual negative gradient direction. We are trying to deflect the negative gradient direction by the matrix  $H$  inverse.

Now, if you do that then that corresponds to the Steepest Descent direction in  $y$ -space and this steepest Descent Direction in  $y$ -space would converge to the optimal point in 1 step, because the Hessian of the objective function  $H y$  which also corresponds to  $f$  of  $L$  transpose inverse  $y$ . The Hessian of that objective function is an identity matrix. So, if we apply the Steepest Descent Method to the function  $H y$  we get is the solution in 1 step and that corresponds to deflecting the negative gradient direction by  $H$  inverse. Now, this method is called the Newton method and we will study Newton method in the next class.

Thank you.