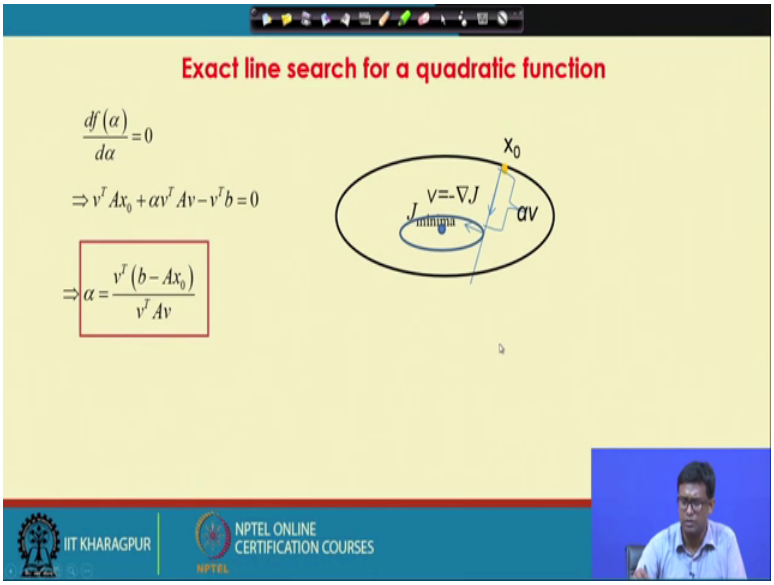
Matrix Solvers
Prof. Somnath Roy
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 45
Iterative Methods for Solving Linear Systems Using Krylov Subspace Methods

Welcome. So, we are discussing about the steepest descent algorithm; where we have seen that solving a problem Ax is equal to b is equivalent to find out minima of the functional J x, where J x is defined as x transpose Ax minus x transpose b. And in order to propose an iterative method for solving or for finding the minima of J x, we discussed about the gradient search algorithm that choose any value x 0 find J x 0 there, and then find out the gradient of J and move along minus gradient of J.

And move up to a certain distance, where this along with J reduces on that particular line, and this distance is measured by parameter alpha we discussed about how to find out alpha; till which J reduces along that line. And then when you see that J has reached a local minima along that particular line change the direction and go to the another take another direction.

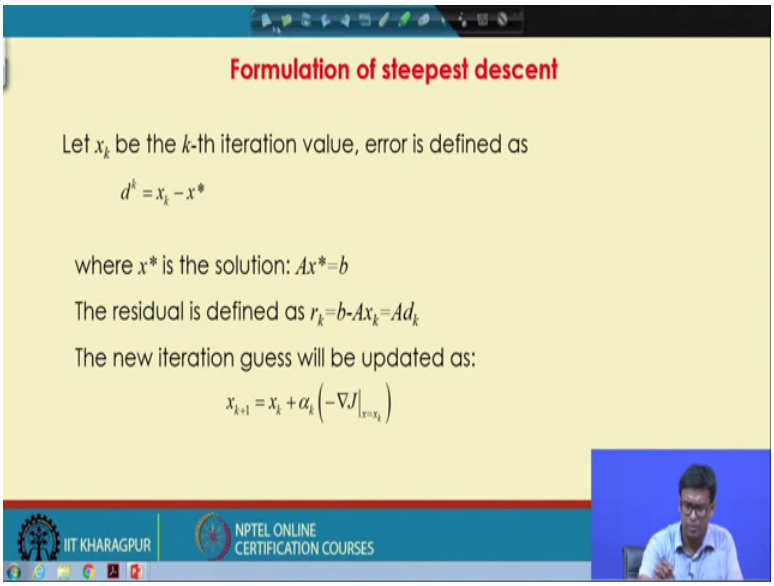
(Refer Slide Time: 01:22)



So, roughly it is this particular method that start with one particular x 0 move along one direction, and then see that this is a local minima of J. From here you take another direction and that is how you approach the global minima. And this is the parameter

alpha that distance should be covered in particular one particular such direction so that the search is optimum and in least number of steps we reach the local minima. Now we thought about taking this into a solver and writing and proposing an algorithm for iteratively solving x is equal to b or finding J minima using this.

(Refer Slide Time: 02:16)



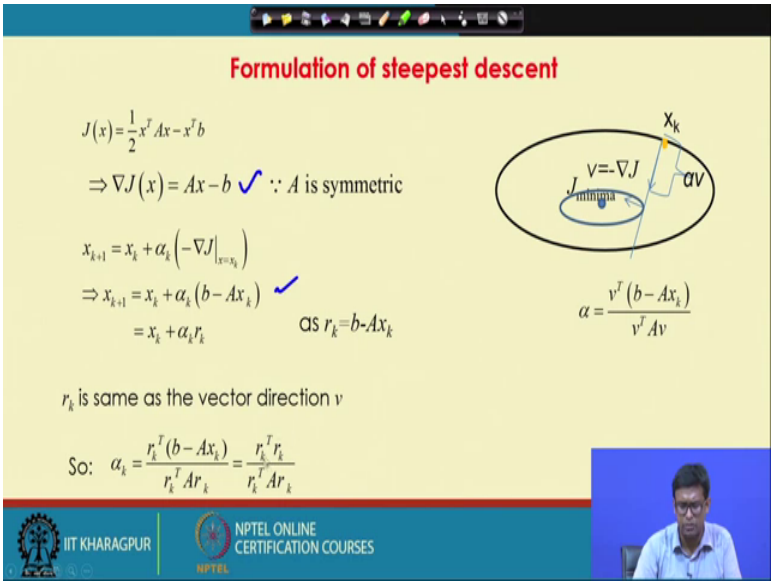
Which is called the steepest descent algorithm, let k be the kth iteration value. And the error is defined as d k is x x k minus x x star; where x star is the solution of x star is equal to b. X star is a where we have the solution x star is equal to b, that is the location at J is minima. So, this let x star be the point where J is minimum. And d k is the error; that means, we considered a guess value x k or kth iterations value x k d k is the error which is x k minus x star.

The residual is defined as b minus A x k, if x k is equal to x star we reach the right value b minus Ax is equal to 0. So, residual is 0, till it is not reached this is the non-zero value and we define residual is b minus Ax k or this is equal to b into A into d k. Why? Because b minus Ax k is equal to b minus A x k minus b minus Ax star, because this is 0. So, this is residual is if there is a 1 2 sorry (Refer Time: 03:45) (Refer Time: 03:53) there is a small sign convention, small sign convention issue r k is b minus A x k and this is; so let us define this is equal to x star minus x, this is minus d k, then it should come out.

So, this is minus of A into x k minus x star which is A into d k. So, let us define minus d k is x k minus x star. New iteration guess will be updated as x k plus 1 is equal to x k

plus alpha k into minus gradient of J x is equal to x k. So, it started with a value of x k, and we will update it to the new iteration x k plus 1. And this is how because we are also when we are thinking of solving x is equal to b, we are also trying to find out the minima of J. So, you should go along minus grad J x k and we found out there is a parameter alpha k by which we should go in that direction.

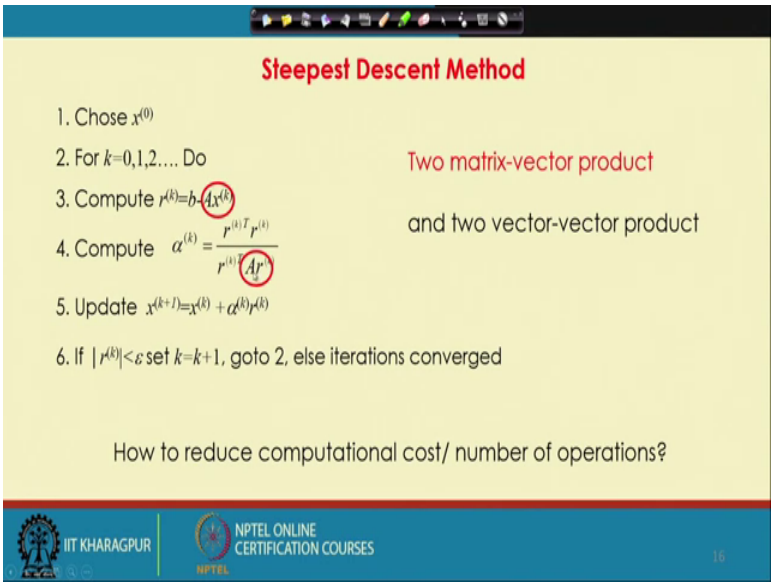
(Refer Slide Time: 05:08)



So, I have J x is equal to half of x transpose Ax minus b, grad J is equal to Ax minus b; as A is a symmetric matrix. X k plus 1 is equal to alpha k into minus grad J of x is equal to x k which is evaluated direct steps in that earlier. And minus grad J is b grad J is Ax minus b. So, minus grad J is grad J is Ax minus b therefore, minus grad J is b minus Ax. So, x k plus 1 into alpha k into b minus Ax k and we have defined b minus Ax k as r k. So, this is x k plus 1 is equal to x k into x k x k plus 1 is equal to x k plus alpha k r k.

R k is the same vector as the vector direction as the same vector in the direction v, r k and v are same. Because we have written earlier v is equal to minus grad J, now we can see that v is the same as r k. So, I have seen that alpha is equal to v if v is minus grad J now we have seen that minus grad J is r k. So, r k and v are same alpha is equal to v transpose b minus Ax k v transpose A v. So, we will get alpha is equal to r k transpose r k and v are same b minus Ax k by r k transpose A r k which is alpha transpose r k into r k transpose A r k.

(Refer Slide Time: 06:44)



And there is one observation which is r k and r k plus 1 are orthogonal. Why? Because, grad J is the direction in which J reduces fastest, grad J is perpendicular to the J contour where it has evaluated. Now, we move till J is minima or grad J this is tangential to J contour to tangential to J contour at x k plus 1. This is x k plus 1. The new direction is minus gradient of J x at x k plus 1 is perpendicular to the tangent. So, this grad J x k and grad J x k plus 1 must be perpendicular to each other. Therefore, r k and r k plus 1 are orthogonal vectors. So, nevertheless we found out that for one particular iteration how to find out alpha k.

So, the steepest descent method will be start with one guess value x 0, and then do an iteration k is equal to 0 1 2 until it converges. Compute r k is equal to b minus Ax k compute alpha k which is our k transpose r k by r k transpose A r k. Update x k plus 1 is equal to x k plus alpha k r k, and check if r k is less than an epsilon is the small value if. If the residual b minus A i have to see whether b minus Ax is equal to 0, if b minus Ax is a very small value then, sorry if not a very small value, if it is a very small value then it is iterated. If it is not a very small value, then set k is equal to k plus 1 and go to the 2 and do the try for the convergence of the iterations.

If else if this value is very small r k is a very small number, then you say that iterations are converging you have reached the right solution. So, this is roughly the steepest descent algorithm what we discussed here, and for a symmetric positive definite matrix,

we can utilize this the irrespective of the way it has been represented as a diagonal dominant matrix or not only, only we have to see that this matrix is symmetric as well as positive definite and we can utilize this method. This is in general a faster method for symmetric positive, definite matrix this is a faster method, then up to Gauss Siedel, Gauss Seidel SOR or Jacobi method.

However, when we look into this method we see that if we try to think do a computer program here, there are 2 matrix vector multiplication here. A into x and A into r. So, if I have a million by 10 to the power 6 by 10 to the power 6 matrix. Each of these steps needs multiplication with each element of the matrix with the vector each component of the vector. So, in a sense 10 to the power 6 into 10 to the power 6 multiplications are needed. Each row needs 10 to the power 6 multiplications and they are 10 to the power 6 (Refer Time: 10:48).

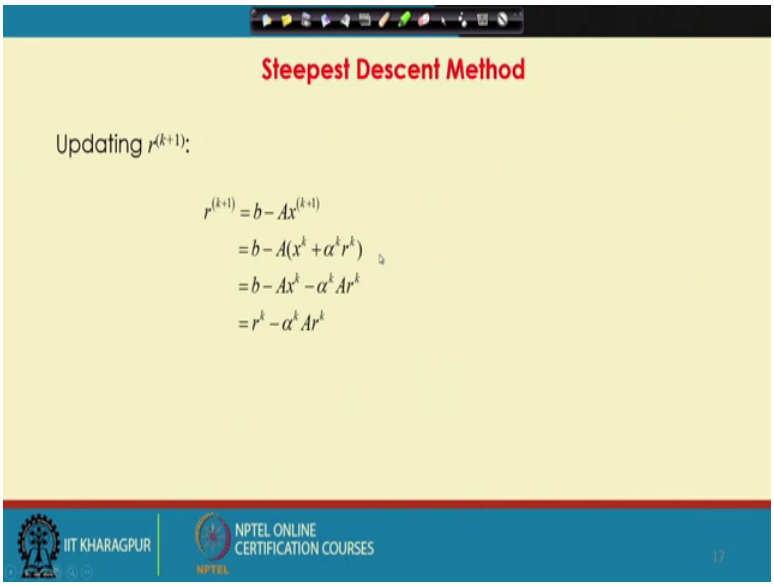
So, this particular method needs in Gauss Seidel in each iteration you only have to do one Ax multiplication; because x i is equal to b minus sum of A i ij x j at the older value except the diagonal term. So, there is only one matrix vector multiplication. There are 2 matrix vector multiplication. So, if I try to do a computer program; write a computer program out of it, though the number of steps will be less than Gauss Seidel; however, the calculations will be way high for large matrices because they are doing in each iteration the calculations will be very high for the large matrices. Because they are doing lot of the Gauss Seidel is doing only once matrix vector multiplication whereas, the steepest descent will do it twice. So, you need to modify this algorithm.

So, you seen that there are 2 matrix vector multiplications and 2 vector-vector product vector-vector is r k transpose r k and r k transpose A r k which is another again another vector is that vector-vector product. However, they are less time consuming. Because if there are 10 to the power 6 rows there then only 10 to the power 6 operations are there. Our matrix vector is 10 to the power 6 into 10 to the power 6 10 to the power 12 operations are; there is a very computationally costly operation if there is a matrix vector multiplication it is done twice.

So, we have to see how can we reduce the cost of the computation or number of operations. How can we improve the algorithm little bit better so that you can at least avoid one matrix vector multiplication here so that for million by million matrix 10 to the

power 12 floating point operations are saved, here we can do one matrix vector multiplications here? So, you have to look into some modifications was some possible modifications into the steepest descent method, before we proposed an algorithm for computer programs.

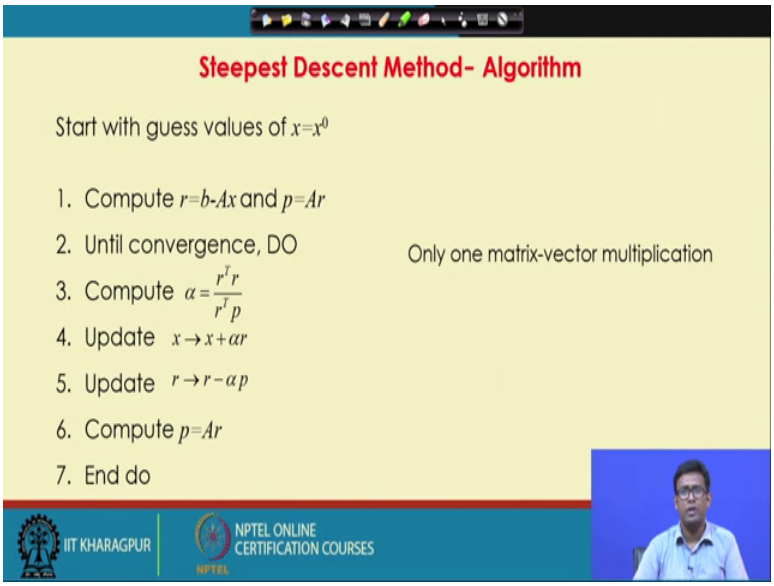
(Refer Slide Time: 13:00)



So, the idea is again if I go back to the previous slide, that every why I need 2 matrix vector multiplication, one is needed to find alpha another is needed to find that. And every time we need to do one matrix vector multiplication to find the updated value of r, r k is equal to b minus Ax. Once we update x in to a matrix vector multiplication here. That is actually replaced here updating r k plus 1 is equal to r k plus 1 is equal to b minus Ax k plus 1; which is b minus Ax k alpha k r k. This b minus Ax k alpha k A r k that is r k minus alpha k A r k. And if I again I go back to the previous slide, A r k is needed also for computing of alpha k plus 1.

So, if r k plus 1 can be computed using A r k then this is the only matrix vector multiplication which is common both to calculation of r k plus 1 as well as to the computation of alpha k. And in that light will try to modify it; will see that this is also needed for computing alpha k. So, while computing alpha k will store a A r k and you will utilize this for updating r k plus 1.

(Refer Slide Time: 14:53)



The final steepest descent algorithm method algorithm will be start with a gaze value of x is equal to x 0, compute b minus Ax and r is equal to b minus Ax and define a new variable p which is A r; p will be utilized later. And until convergence; that means, convergence means that until r is less than epsilon. Epsilon is a very small number; do compute alpha which is r transpose r into r transpose p.

So, alpha is equal to r transpose r r transpose A r, and now A r has been substituted by p. Update x is equal to this is because the same variable x which is overwritten as x is equal to x plus alpha at that is why we have written x arrow is equal to x plus alpha r. This is the way to write something which is getting overwritten. Update x is equal to x as x plus alpha update r. So, r k plus 1 is equal to r k minus alpha k A r evaluated at kth level. So, this is r k minus alpha k p k.

So, update r is equal to r minus alpha p compute p is equal to A r, and then if convergence is not done end do means you again come back here, and further with this the new value of p compute alpha, update x update r and repeat this loop until you see that r is less than mod r is less than epsilon or convergence has been achieved. So, this is the steepest descent method algorithm, and this is applicable only for A symmetric, what happens if A is not symmetric positive definite, then the problem we are solving in this problem what we are solving here is not finding x is equal to b solving or J is minima. The problem we are solving is not J is minimize not now solving x is equal to be

something different; is if it is not symmetric we are solving half of A plus A transpose x is equal to b a different problem.

So, we are trying to find out minima of something which is not Ax is solving x is equal to b; however, were convergence is based on mod r is less than epsilon. So, either it will not converge, but if it converges then it will take us to the same solution that mod r is less than epsilon; so, x is equal to b; so, in case the matrix is not symmetric or matrix is little not the asymmetricity is not very high.

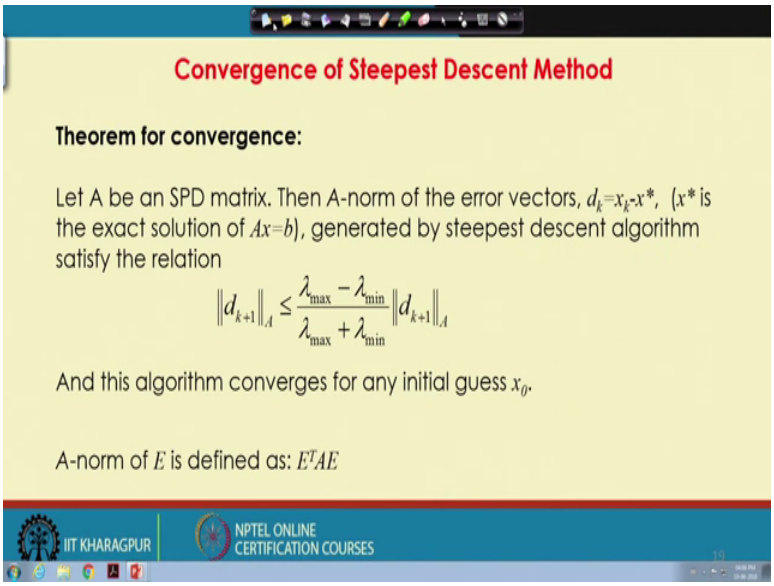
A i k is probably A k i plus a small number. In that case it still converges, but it takes lot of lot many iteration does not take less number of iterations. Because we are not using the right solution algorithm for that cases; however, if only if the matrix is symmetric positive definite, then using this method is advantageous in terms that this the formulation is complicated one program is also probably a little complicated than the Gauss Seidel method; however, this is advantageous because it is take times. The number of steps are small number of iterations are small calculations in each iteration is comparable to Gauss Seidel.

So, overall computational cost is less if we apply this method. Only the matrix if the matrix is symmetric positive definite, then application of this method is worthwhile. Otherwise it might give us the right result because we are looking for these criteria that mod r less than epsilon; that means, b minus perhaps Ax is less than epsilon. We are looking into this right in here. So, it might you if it converges it will give us right it shows that it will take as Ax is equal to b location. But it can take a very high number of iterations in the matrix is not symmetric positive definite.

So, this if I have a symmetric positive definite matrix, this is the algorithm through which will we will see later like Gauss Seidel have or SOR how a computer program can be developed. And, we can also demonstrate that the number of iterations and as well as computational time is much smaller than Gauss Seidel or Jacobi or SOR method for a symmetric positive definite matrix.

So, we will see in later class that how we can develop a computer program using this. And interestingly this is only one matrix (Refer Time: 20:09) in multiplication, earlier we had 2 matrix vector multiplication. So, one has been reduced there is only one matrix vector multiplication here.

(Refer Slide Time: 20:17)



Now we need to look into the convergence of this method; that is for 2-D I can give you some visualization that iteratively we if we change the such direction you should approach to the J minima or this method should converge. Now, there are certain theorems and analysis which can show that the error convergence means the error will be in smaller x minus x star is smaller than is the error x k minus x star this error will finally, reduce to a very small number given certain conditions.

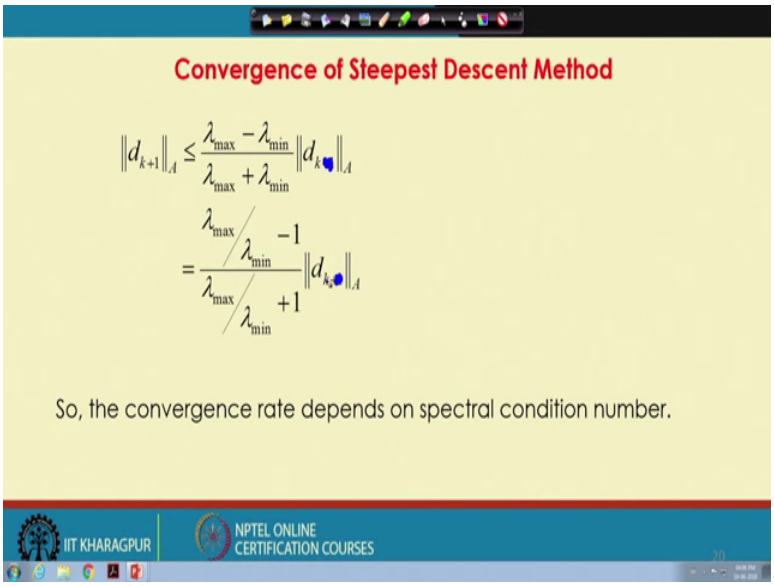
That let A be the theorem for convergence tells that let A be and SPD matrix, that is a symmetric positive definite matrix, then the A norm of the error vector which is x k minus x star, x star is the exact solution of x is equal to b. Generated by steepest descent algorithm satisfy the relation mod d k plus 1 A norm; A norm of any vector is defined as E transpose AE. So, a d k plus 1 transpose d k plus, this is basically d, I am sorry, where is the (Refer Time: 21:32) d k plus 1 transpose A A norm of d k plus 1, or let us take d k, it is A norm is basically given as d k transpose A d k.

This norm is satisfy the relation is less than equal to lambda max minus lambda min divided by lambda max plus lambda mean of d k; so, if we have any always d k plus 1 is less than d k A. The error x k minus x star at any iteration is always less than the error in the previous iteration therefore, it should converge to a finally, this error should be a small value and it should converge to a small number should converge to the right solution.

However, the rate of convergence here depends on not on the spectral radius rather the maximum and minimum lambda. So, the relation the difference of the maximum eigenvalue and minimum eigenvalue of this lambda max and lambda mean are there is no iteration matrix Are the eigenvalues of the matrix A only. So, it depends on the eigenvalues of the matrix A; so, this algorithm and this shows that this algorithm will converge for any Gauss value x 0, because this is always a number less than 1. Therefore, d k plus 1 is always error is always reducing.

So, finally, the error should go down it will with start with some value it will keep on the magnitude modulus of error is always reduced reducing. So, it will be a small number and this is this is a positive number right, because A is a SPD, this is always greater than 0 as A is SPD. So, this number will be greater than 0 and reducing; that means, that if this is d k A and this is k this will asymptotically approach 0. It will never be exactly 0 it will be always greater than 0, but it will be a very small number and then it will be finally 0. So, this algorithm will converge for any initial guess x 0.

(Refer Slide Time: 24:02)



This is a mistake here and this should be d k. I should correct it in the original notes also. This is d k. So, you can see d k plus 1, you know, means lambda max minus lambda mean of by lambda max plus lambda mean of d k; which is lambda max by lambda mean minus 1 by lambda max plus lambda min plus 1 of d k. And spectral condition number, I have defined it earlier of any matrix is the ratio of lambda max and lambda mean. Now

we have said that as small as the condition number therefore, lambda max and lambda means are closer it is easier good for matrix solver.

So, you are also seeing here, and this is always greater than; so, basically this is absolute all these are absolute values. So, this is always greater than 1 all this should be an absolute value. So, this is always greater than equal to 1. So, as this value is close to 1, this number is smaller; is 1.000, it is very small number. So, in very few steps the d k plus 1, the air should approach 0. So, if convergence spectral radius is the convergence rate depends on the spectral condition number, spectral condition number low, spectral condition number; that means faster convergence.

And when discussing about condition number, we have discussed that that if the condition number is small, then the convergence is faster. If lambda max and lambda min are closer the condition number are small convergence is faster. And we can see if this number is smaller we will reach the convergence faster. So, low condition number will give a faster convergence here. So, this is the first time we discussed earlier discussed about condition number we are seeing one example of condition number in first in the rate of convergence of the own matrix particular matrix solver which is the steepest descent method matrix solver, ok.

So, this method will converge; that means, we started with some geometric functional, now we can see that once we have derived the algorithm. This algorithm converts starting with any guess value x 0, this algorithm should take us to the convert solution of Ax is equal to b. And that is only for symmetric positive definite matrix. Now the question is that the entire exercise is only devoted for symmetric positive definite matrix. A matrix may not be symmetric in reality we deal with number of cases where we get asymmetric matrices. For example, if we think of a finite difference equation that we are discussing earlier, and if we use non uniform grid spacing the matrix will be a symmetric.

So, how can we modify this equation for an asymmetric matrix. As well as in a general case there can be negative definite matrix; for positive definite matrix, there is a solution if the matrix is not positive definite. If there is a negative eigenvalue, what is how to solve this matrix? This method does not cover those matrices therefore, it still now

restricted only to a closed class of method a small class of method we use symmetric positive definite.

Now, our next goal will be if we can extend this method for general matrices which are not symmetric and non-positive definite matrix. And what we will discuss for that there is called general projection methods. Instead of having a method for searching through gradient search approaching the minimum value of a functional, will see a g mode generalized method were probably the functionalism is little different; we are trying to find out minima of some other function, but solving a matrix which is not symmetric or which may not be positive definite also. Extension of this method will take us to general projection methods; which can solve a larger class of matrices. We will see that in the next classes.

Thank you.