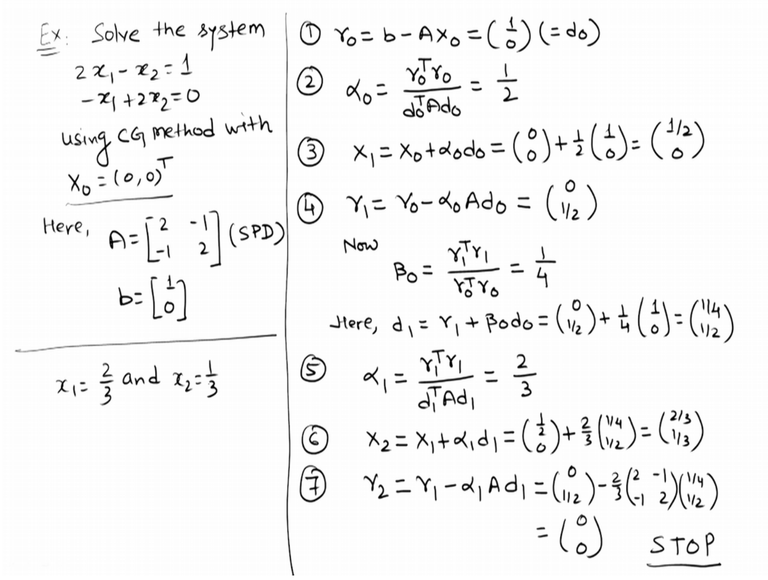
Matrix Analysis with Applications
Dr. Sanjeev Kumar
Department of Mathematics
Indian Institute of Technology, Roorkee

Lecture - 37
Krylov Subspace Iterative Methods
(CG and Preconditioning)

Hello friends. So, welcome to the 37th lecture of this course. As you remember in the last lecture we have discussed about Krylov subspaces based iterative methods. In last lecture in particular we have discussed about conjugate gradient method there we have seen the algorithm of this method. Now, in this lecture let us start with an example of conjugate gradient method.

(Refer Slide Time: 00:55)



Let us take a simple example solve the system 2 x 1 minus x 2 equals to 1 minus x 1 plus 2 x 2 equals to 0 using conjugate gradient method with initial solution x 0 as 0 0. So, here if we see the coefficient matrix A is a 2 by 2 matrix 2, minus 1, minus 1 and 2 which is symmetric as well as positive definite. So, it is a SPD matrix.

Now, the right hand side vector b is 1 and 0 and x 0 is given as 0 0. So, let us apply conjugate gradient method for solving this system. So, if you remember the first step of the algorithm of conjugate gradient method is to calculate residual initial residual that will be r 0 and it is b minus AX 0. Since AX 0 is a 0 vector. So, the second term will be 0 and it is equals to b and b is given as 1 and 0. Also in the algorithm we assume it as the initial such direction.

Now, in the second step what we will do? We will calculate alpha 0 that is the step length which is r 0 T r 0 upon d 0 T A d 0 and here you can notice d 0 and r 0 are the same vector. If I calculate it comes out to be 1 by 2, then in the third step since we are having d 0 as well as alpha 0 with us. So, I can calculate the next approximation of the solution that is X 1 which is X 0 plus alpha 0 d 0. So, here X 0 is 0 0 plus alpha 0 is 1 by 2 and d 0 is 1 and 0,so it becomes 1 by 2 and 0.

Now, in 4th step what we will do we will calculate the residual in first iteration. So, r 1 here r 1 will become r 0 minus alpha 0 Ad 0 which is different from the steepest descent method because in steepest descent method we used to take r 1 as b minus AX 1. So, if I calculate it I am having r 0 alpha 0 d 0 as well as a with us. So, what I will be having 0 and half. Now, once you are having r 1 calculate beta 0 which is r 1 T into r 1 upon r 0 T into r 0 and it comes out to be 1 by 4. So, here the new search direction d 1 is given as r 1 plus beta 0 d 0. So, r 1 is basically 0 and half plus beta 0 is 1 by 4 and d 0 is from the first line 1 and 0. So, it comes out to be 1 by 4 as the first component and 1 by 8 sorry 1 by 2 as the second component.

Once we are having d 1 with me I can calculate alpha 1. So, again alpha 1 will become r 1 T r 1 upon d 1 T A d 1 and this quantity becomes 2 by 3. So, if I am having alpha 1 I can calculate next step approximation of the solution that is X 2 it is X 1 plus alpha 1 d 1. So, X 1 is with me here which is half and 0 plus alpha 1 is 2 by 3 and d 1 is 1 by 4 1 by 2. So, this becomes 2 by 3 and 1 by 3.

Next I will calculate r 2. So, if you see here r 1 is r 0 minus alpha 0 A d 0. So, r 2 will become r 1 minus alpha 1 A d 1. So, here if I calculate it r 1 is 0 and half minus alpha 1 is 2 by 3 into matrix A is 2 minus 1 minus 1 2, and then d 1 is d 1 is 1 by 4 and 1 by 2. So, this becomes if I calculate it 0 and 0. So, now, r 2 is 0. So, our iteration will stop means method is converged to x 2 which is, solution is x 1 is 2 by 3 and x 2 is 1 by 3 which is also the exact solution of the system. So, this is the implementation process of the conjugate gradient method.

Now, let us talk about the convergence of the conjugate gradient method.

(Refer Slide Time: 08:21)

Theorem:- Let A be a SPD matrix of order n. The CG method for solving the system $Ax=b$ converges at most in 'n' iterations.
Moreover, the number of iterations for convergence will be propotional to $\sqrt{K(A)}$ $\left(K(A) = \frac{\lambda_{max}}{\lambda_{min}} = \|A\|_2 \|A^{-1}\|_2 \right)$

$\Rightarrow$ To have a better convergence, we want $K(A)$ must be small.

So, there is a result in the literature regarding the convergence of this method that is let A be a symmetric and positive definite matrix of order n. So, it is n by n symmetric as well as positive definite matrix. Then thus conjugate gradient method for solving the system AX equals to b converges at most in n iterations. So, CG method will not take more than n iterations. It will take less n or less than n iterations like in the case of previous example which we have taken in the beginning of this lecture the system was 2 by 2 and we have taken just 2 iterations for getting the exact solution.

Moreover the number of iterations for convergence will be proportional to a square root of conditional number of A. So, if conditional number of A is large then matrix CG method will take more number of iterations; if it is small then we will be having a faster convergence. So, it means to have a better convergence and if you recall from some of from the previous lecture the conditional number of A here can be given as lambda max upon lambda min which is just the product of norm of a with norm of A inverse. So, to have a better convergence we want the conditional number of the coefficient matrix must be small.

Now, the question arise if A is symmetric and positive definite matrix, but the conditional number of A is quite large then we will be having slow convergence of CG method. So, can we have some method to improve this convergence or to make the convergence faster? Yes, we are having such a method and that is called preconditioning. So, let us talk about preconditioning.

(Refer Slide Time: 12:24)

Preconditioning: The general idea of preconditioning for iterative methods is to modify the ill-conditioned system

$$AX = b$$

in such a way that we obtain an equivalent system $\hat{A}\hat{X} = \hat{b}$ for which the iterative method converges faster.

One of the approach for it to choose a nonsingular matrix M and rewrite the original system $AX = b$ as

$$M^{-1}AX = M^{-1}b$$

Here, M should be chosen in such a way that

$$K(M^{-1}A) << K(A)$$

So, the general idea of preconditioning for iterative methods is to modify the original system which is ill-conditioned, ill-conditioned means here the conditional number of the coefficient matrix is quite large. So, if AX equals to b is an ill conditioned system, so to modify this system in such a way that we obtain an equivalent system let us write it with the cap notations that is A cap X cap equals to b cap and here meaning of equivalent is the solution of AX equals to b is also the solution of A cap X cap equals to b cap or in a reverse way the solution of the new system which is denoted with caps equals to the solution of the original system. And for this new system we should have a faster convergence. So, for which the iterative method converges faster.

So, one of the approach for doing this, one of the approach for it to choose a nonsingular matrix m of the same size as the size of the coefficient matrix A and rewrite the original system which is ill-conditioned AX equals to b as M inverse AX equals to M inverse into b. Here m should be chosen in such a way that the conditional number of M inverse A should be very less when compared to the conditional number of A. So, in that way the convergence of this new modify system will be faster when compared to the ill-conditioned original system. So, this is the basic idea of the preconditioning.

Now, one of the problem here when you are going from the original system AX equals to b to a new system A cap X cap equals to b cap then for applying the conjugate gradient to the new system the matrix A cap as should A cap should be symmetric as well as positive definite. So, how to choose such a m that is the preconditioned matrix or preconditioner m such that M inverse a should be symmetric as well as positive definite. So, let us address this issue.

(Refer Slide Time: 17:32)

Issue: How to find $\hat{A}$, $\hat{x}$ and $\hat{b}$ in order to ensure the SPD property of $\hat{A}$.

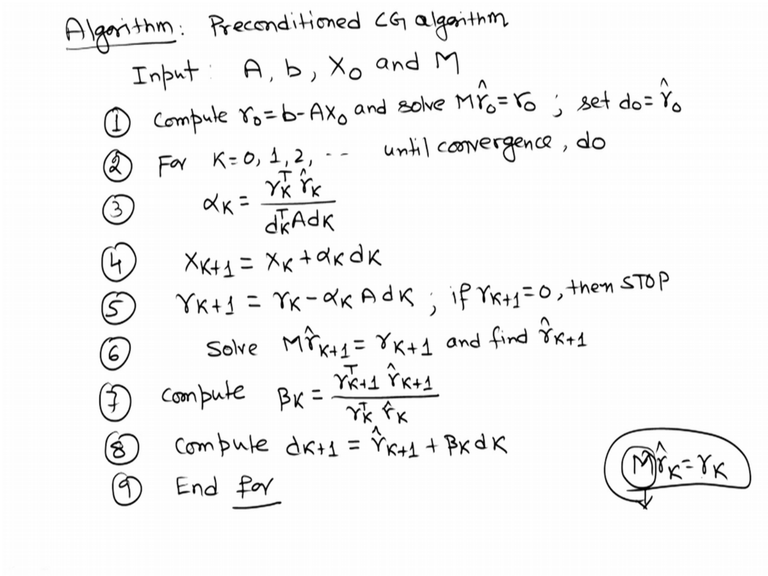
Soln: Let $M^{-1} = LL^T$ (L is a $n \times n$ lower triangular matrix) which is nonsingular)

$$
\begin{aligned}
Ax = b &\iff M^{-1}Ax = M^{-1}b \\
&\iff L^T A x = L^T b \\
&\iff \underbrace{L^T A L}_{\hat{A}} \underbrace{L^{-1}x}_{\hat{x}} = \underbrace{L^T b}_{\hat{b}}
\end{aligned}
$$

So, here issue is how to find A cap X cap and b cap in order to ensure the SPD property means symmetric and positive definiteness property of A cap. So, a solution to this issue is let M inverse equals to product of a matrix L with L transpose, where L is a lower triangular matrix. So, here what we will be having. So, here L is a m by m lower triangular matrix since we are taking a as n by n. So, better to write it n by n here which is nonsingular also, means none of the diagonal element of this lower triangular matrix is 0.

Then what we will be having? If we are having original system as AX equals to b then this system is equivalent to M inverse AX equals to M inverse into b which is equivalent to L transpose AX equals to L transpose into b which is equivalent to L transpose a L into L inverse x equals to L T b. Now, take this matrix L transpose a L as your A cap take L inverse x as your new variable X cap and take L T b as b cap. So, if you choose your matrix same A cap X cap and b cap in this way then we can ensure the symmetric and positive definiteness of matrix A cap.

(Refer Slide Time: 20:35)



Now, let us rewrite the conjugate gradient algorithm with preconditioning. So, the same algorithm which we have taken in the previous lecture we will rewrite it, but together with a preconditioner M. So, let us write it preconditioned CG algorithm. So, here input will be the coefficient matrix A right hand side vector b the initial solution X naught and the modification when compare to the classical conjugate gradient method is here we will be having a preconditioner M which is a matrix of the same size as your A.

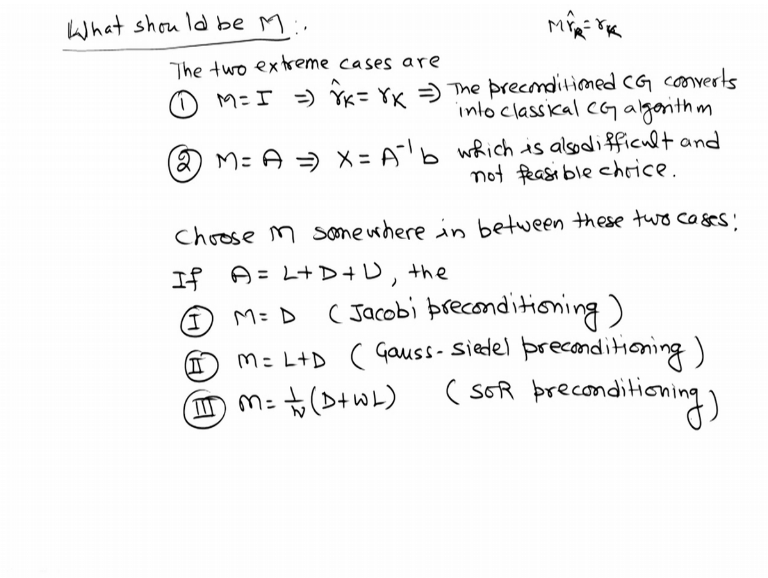
Now, the first step is compute r 0 equals to b minus AX 0 and solve Mr 0 cap equals to r 0. Once you obtain r 0 cap from here set d 0 equals to r 0 cap. If you recall the original algorithm there we do not have this step. And what we are doing? We are setting d 0 as my r 0. But here what I am doing? I am calculating a new r 0 cap which is nothing just M inverse r 0 which is due to preconditioning.

Now, in the second step for K equals to 0 1 2 until convergence do find out alpha K which is r K transpose into r K cap here now. So, this is another change here. Earlier we were having r K transpose into r K. Now, what we are having r K transpose into r K cap upon d K transpose A d K. The fourth step is if we are having update X as X K plus 1 equals to X K plus alpha K d K. Once you are having x K plus 1 calculate r K plus 1 which is r K minus alpha K A d K. Here if r K plus 1 equals to 0 then stop otherwise go to step 6 again solve Mr K plus 1 cap equals to r K plus 1 and from here obtain and find r K plus 1 cap with the help of m and r K plus 1. Then compute beta K which is r K plus 1 transpose into r K plus 1 cap upon r K transpose into r K cap.

Once you are having your beta K then compute d K plus 1 equals to r K plus 1 cap plus beta K d K and in that way the algorithm will run till convergence and this is the end of for loop which you are having in the second line. So, this is the preconditioned conjugate gradient algorithm.

Now, what we will see here what is the extra computation here when compared to the original conjugate gradient algorithm? If you see it in each iteration what we are having? We have to solve a system M r K equals to r K extra in each iteration for finding the r K cap. So, to have a faster algorithm to have a faster computation we have to choose preconditioner M in such a way that this system can be solved easily it should be there otherwise there is no use of using such kind of preconditioner because you are having in each iteration for solving a system in each iteration you are having an extra system. So, how to choose this M let us talk about it.

(Refer Slide Time: 26:33)



So, what should be M? So, the 2 extreme cases are number one, M equals to identity matrix. So, you can solve easily the system Mr 0 equals to r 0 or r K equals to this one because if this is an identity then your r K cap will become r K in each iteration and it means the preconditioned CG converts into classical CG algorithm. So, there is no preconditioning here.

The other choice is M equals to A. So, if you take M equals to A that is another extreme case then X will become A inverse into b which is as difficult because it is a direct method for solving the original system which is also difficult and not feasible choice. So, what we need to do? We need to choose m somewhere in between these 2 cases. So, there are few choices of M the first if the original matrix A can be written as the sum of 3 different matrices L, D and U like we have done in the beginning of iterative methods in Jacobi and Gauss Seidel method. So, where L is a lower triangular matrix, D is a diagonal matrix and U is an upper triangular matrix.

Then the first choice of M is if we take M equals to D it is called Jacobi preconditioned. So, it is called Jacobi preconditioning. The other choice of M is take M equals to L plus D then it is called Gauss Seidel preconditioning. The third choice of M is something like 1 upon omega into D plus omega L; this kind of choice of M is called successive over relaxation preconditioning.

(Refer Slide Time: 30:48)

then,

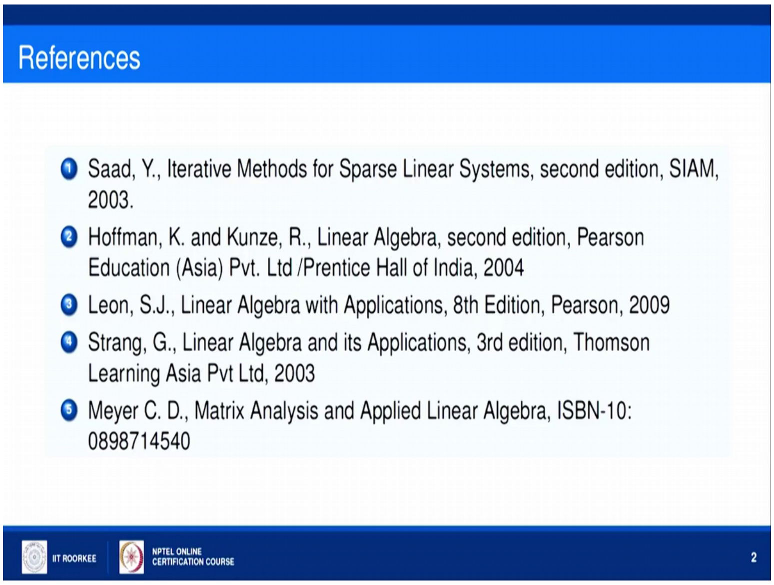
1. $\hat{A}$ will be symm. & positive def
2. $M\hat{r}_n = r_n$ can be solved easily.
3. $\cancel{\rho(I - M^{-1}A) < 1}$ ~~or~~ $\underline{\|I - M^{-1}A\| < 1}$

If you choose M in out of any way out of these 3 then we will have number 1 M will be symmetric and positive definite not M basically A cap, number 2 M r n equals to r n can be solved easily, because the earlier methods and the third one which is a guarantee for the convergence that is the spectral radius of I minus M inverse a must be less than 1 or norm of I minus M inverse a will be less than 1. I will prefer this one because this is necessary and sufficient condition. So, these 3 things will effect surely if you make the choice of m based on those 3 Jacobi, Gauss Seidel or SOR type of approaching.

So, with this I will end this lecture. So, in this lecture we have learnt that how can we make the convergence of the conjugate gradient method faster than the classical one using the preconditioning procedure.

So, with this I will end the iterative methods in this course and in the next lecture we will see some new properties of a matrix especially when all the entries of a matrix are positive.

(Refer Slide Time: 32:55)



These are few references for this lecture.

Thank you very much.