Matrix Solvers

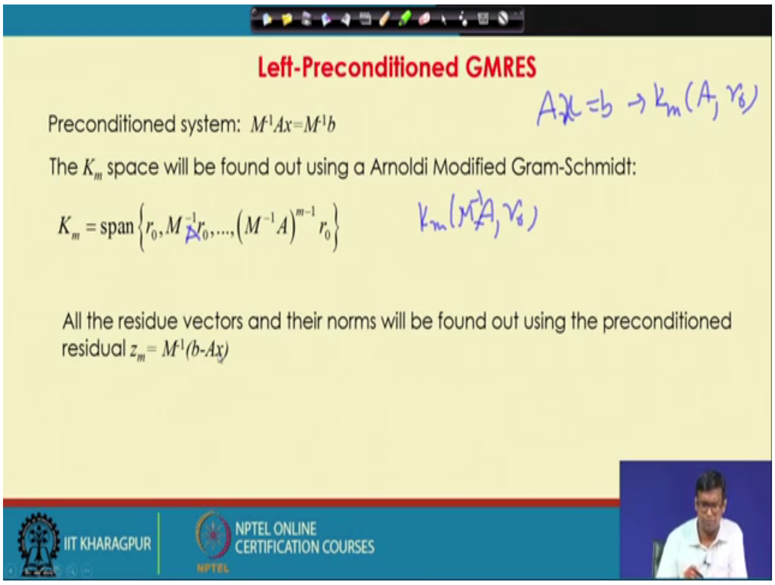
Prof. Somnath Roy
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 58
Preconditioned GMRES

Welcome. We are looking into preconditioning technologies, preconditioning techniques I should say. And we have looked into how precondition conjugate gradient can be obtained or for symmetric matrices how preconditioners can be obtained, so that the precondition matrix also remained as symmetric positive definite one. And looked into two different variants of conjugate gradient: the split precondition conjugate gradient and the left-precondition conjugate gradient.

We have not discussed in detail how the preconditional matrix is obtained till now, but we have seen that and different algorithm slightly different from the original algorithm can be proposed, if we think of a precondition system of equations, where the inverse the precondition matrix M inverse or part of the precondition matrix l inverse, l inverse transpose are already available with us how these algorithms will look like. Now, we will see for general matrices, where we cannot apply conjugate gradient, which are non-symmetric matrices, where still we can apply a Krylov space method like GMRES preconditioning can be done.

(Refer Slide Time: 01:33)



Left-Preconditioned GMRES

Preconditioned system: $M^{-1}Ax = M^{-1}b$

$Ax = b \rightarrow K_m(A, r_0)$

The K_m space will be found out using a Arnoldi Modified Gram-Schmidt:

$$K_m = \text{span}\left\{r_0, M^{-1}\cancel{A}r_0, \ldots, \left(M^{-1}A\right)^{m-1} r_0\right\}$$

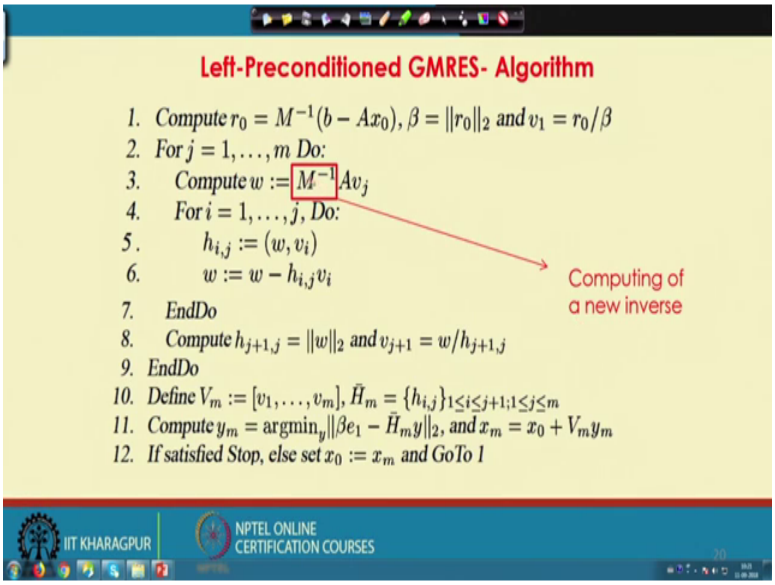
$K_m(M^{-1}A, r_0)$

All the residue vectors and their norms will be found out using the preconditioned residual $z_m = M^{-1}(b-Ax)$

So, we start with left-preconditioning of GMRES. The left-precondition system M inverse A x is equal to M inverse b. Now, instead of solving A x is equal to b, if we are solving M inverse A x is equal to M inverse b for GMRES as we have to evaluate the basis vectors of the Krylov subspace K m, we will find out the Krylov subspace for r naught, M inverse r naught, M inverse A r naught instead of finding sorry r naught M inverse A r naught M inverse A m minus r naught.

So, instead of finding K m, this will be K m of M inverse M inverse A r naught. For A x is equal to b, we use the Krylov subspace of A, r naught. Now, as we are solving M inverse A x is equal to b, we will use Krylov subspace of M inverse A r naught. So, this is this there will be an M inverse A r naught then up to M inverse A to the power m minus 1 up r naught. Now, Krylov subspace, we will change. And we will use something like a Arnoldi modified Gram-Schmidt to obtain the basis vectors for the Krylov subspace of this for this particular Krylov subspace. All the residue vectors and their norms will be found using the preconditioned residual Z is equal to M inverse b minus A x.

(Refer Slide Time: 02:59)



Left-Preconditioned GMRES- Algorithm

```
1. Compute $r_0 = M^{-1}(b - Ax_0)$, $\beta = \|r_0\|_2$ and $v_1 = r_0/\beta$
2. For $j = 1, \ldots, m$ Do:
3.     Compute $w := \boxed{M^{-1}} Av_j$
4.     For $i = 1, \ldots, j$, Do:
5.         $h_{i,j} := (w, v_i)$
6.         $w := w - h_{i,j}v_i$
7.     EndDo
8.     Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$
9. EndDo
10. Define $V_m := [v_1, \ldots, v_m]$, $\bar{H}_m = \{h_{i,j}\}_{1 \leq i \leq j+1; 1 \leq j \leq m}$
11. Compute $y_m = \text{argmin}_y \|\beta e_1 - \bar{H}_m y\|_2$, and $x_m = x_0 + V_m y_m$
12. If satisfied Stop, else set $x_0 := x_m$ and GoTo 1
```

Computing of a new inverse

So, the Krylov subspace vectors are found for a, the Krylov subspace vector are found for M inverse A and r naught, and the residue vector z where the new residue vector rather r naught is equal to M inverse b minus A x 0. This is the new residue vector. And that is how so instead of solving A x is equal to b, we are solving M inverse A x is equal to b in a GMRES. There is no other change only the idea is that solving M inverse A x is equal to b. um

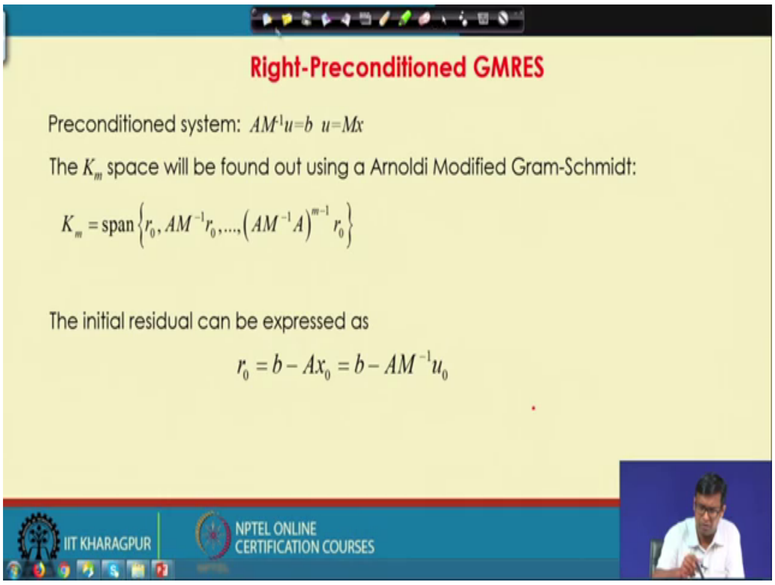
So, wherever we have we will be replaced by M inverse. And initial r naught finding is that then the orthogonalization step will require finding w, which was in GMRES finding A v j. Once you once v is on one of the basis of the Krylov subspace, we are orthonormalizing the other basis is w will be M inverse A v j. And it will follow the Arnoldi Gram-Schmidt type of steps. And then we will compute the elements of the Heisenberg matrix h, which is l 2 matrix of w, and then using the l 2 norm of w will normalize the v j, so it so that it becomes an the set becomes an orthonormal set.

And we will define the V m matrix H m. Finally, y m will be obtained by a minimization of beta e 1 minus H bar m y, these algorithms we have looked earlier. This is very similar as Jacobi algorithm, and x m will be updated such that. And one once we are satisfied then stop, otherwise you update x the gaze value x 0 with x m and continue with this loop. So, this is nothing but modifying the GMRES considering that we instead of solving A x is equal to b, we are solving M inverse x is equal to b.

While doing so we get two important steps here. One is a M inverse first M inverse b minus A x 0, which is one time step at least which you have to do it ever. So, M inverse not once time step at every iteration you have to do. So, at every and also M inverse A v 0. So, each every iteration requires a evaluation of the matrix M inverse or of the precondition matrix M inverse.

So, it is very similar to conjugate gradient, whether the problem is very similar to conjugate gradient method. In conjugate gradient method, we had the requirement of finding M inverse or l inverse transpose and multiplying it with certain vector for preconditioning, here also you have the requirement in the preconditioning steps itself. So, this can be thought of a finding M inverse A matrix, and this can be thought of finding M inverse. So, two operations are there actually.

(Refer Slide Time: 06:06)



Right-Preconditioned GMRES

Preconditioned system: $AM^{-1}u=b$ $u=Mx$

The K_m space will be found out using a Arnoldi Modified Gram-Schmidt:

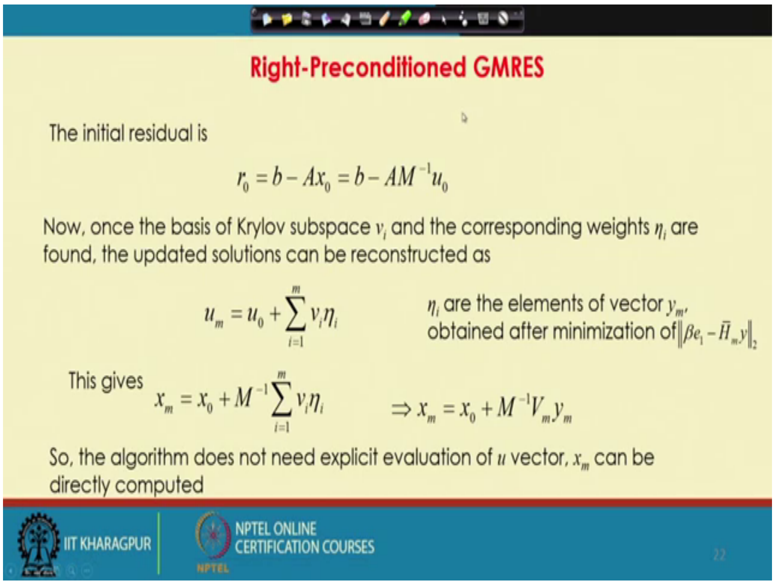
$$K_m = \text{span}\left\{r_0, AM^{-1}r_0, \ldots, \left(AM^{-1}A\right)^{m-1} r_0\right\}$$

The initial residual can be expressed as

$$r_0 = b - Ax_0 = b - AM^{-1}u_0$$

Right-preconditioned conjugate gradient, we have A M inverse u is equal to b and u is equal to M x. The Krylov subspace is now of A m inverse instead of A inverse. So, A it is r 0, A M inverse r 0, A M inverse this A is not here, A M inverse A M to the power minus 1 r 0, this is called Krylov subspace. We have to find out K m of A M inverse and r 0. And r 0 is the residual of this particular matrix system. r 0 is equal to b minus A x 0, which is b minus A M inverse a u 0.

(Refer Slide Time: 07:00)



Right-Preconditioned GMRES

The initial residual is

$$r_0 = b - Ax_0 = b - AM^{-1}u_0$$

Now, once the basis of Krylov subspace v_i and the corresponding weights η_i are found, the updated solutions can be reconstructed as

$$u_m = u_0 + \sum_{i=1}^{m} v_i \eta_i$$

η_i are the elements of vector y_m, obtained after minimization of $\left\|\beta e_1 - \bar{H}_m y\right\|_2$

This gives $x_m = x_0 + M^{-1}\sum_{i=1}^{m} v_i \eta_i$ $\Rightarrow x_m = x_0 + M^{-1}V_m y_m$

So, the algorithm does not need explicit evaluation of u vector, x_m can be directly computed

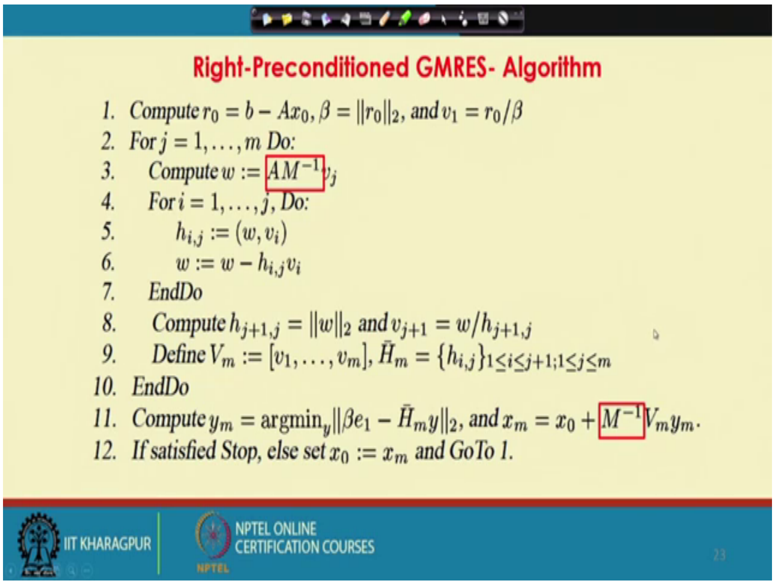
The initial residual is this. Now, once we get all the basis of Krylov subspace, and the corresponding weights eta i. So, we can find out the final solution of this equation of A M in rather we will find we have the equation A M inverse u is equal to b. So, once we find out all the basis of the Krylov subspace of A M inverse, and the corresponding weights to get the final solution we can write u m is equal to u 0 plus the basis into certain coefficient on that.

So, the final solution will be will lie u will be u 0 will be in an affine subspace of the affine Krylov subspace, which is u 0 plus this vector, and v i's are the basis of Krylov subspace of e 1 of A M inverse and r. So, this is v i's are the basis of Krylov subspace, and eta i are the weight which is eta i are the elements of the vector y m, which obtained after minimization of beta e 1 minus H bar m.

This will give that x m, so x is equal to we also know that x is equal to M inverse u. So, we will multiply M inverse here and we will get x, x m is equal to x 0 plus M inverse v i y i eta i. So, once we get the weights eta i, which are the elements of the vector y m, and the basis of Krylov subspaces are formed. We can directly find out x m without evaluating u. So, the algorithm will become x m is equal to x 0 plus M inverse V m y m. The original algorithm was x 0 x m is equal to x 0 plus V m y m, here it will be x 0 plus M inverse V m y m. Important thing is that we again have to go into a matrix inversion or a matrix solution of M inverse, we again have to find out M inverse, here we are stuck with the same problem.

Another important observation is that exactly same as our old method last discuss method, which is split precondition conjugate gradient or if we think of right precondition conjugate gradient. We do not have need to we do not need did not need to evaluate u there. Here also you do not need to explicitly evaluate u rather we can directly find out x, if we have find out the basis vectors of the Krylov subspace A M inverse and r. The algorithm does not explicit need explicit evaluation of u vector, x m can be directly computed using this.

(Refer Slide Time: 09:58)



Right-Preconditioned GMRES- Algorithm

```
Compute $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, and $v_1 = r_0/\beta$
For $j = 1, \ldots, m$ Do:
    Compute $w := \boxed{AM^{-1}}v_j$
    For $i = 1, \ldots, j$, Do:
        $h_{i,j} := (w, v_i)$
        $w := w - h_{i,j}v_i$
    EndDo
    Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$
    Define $V_m := [v_1, \ldots, v_m]$, $\bar{H}_m = \{h_{i,j}\}_{1 \le i \le j+1; 1 \le j \le m}$
EndDo
Compute $y_m = \text{argmin}_y \|\beta e_1 - \bar{H}_m y\|_2$, and $x_m = x_0 + \boxed{M^{-1}}V_m y_m$.
If satisfied Stop, else set $x_0 := x_m$ and GoTo 1.
```

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, look into the right precondition GMRES algorithm, it starts with the initial steps. And now the basis vectors of v is computed by A M inverse, so an A M inverse step is needed. And remaining all the steps are exactly same as the original GMRES, except when finding M x m we need M inverse. So, these two operators are required A M inverse and M inverse. So, we kind of every time we encountering the same problem that the preconditioned equation look very similar to the precondition algorithm, look very similar to the older algorithm. Only we have to find the operator, so at M inverse is included, there the preconditioning matrix is included. So, how will you find precondition matrix that becomes the important question here.

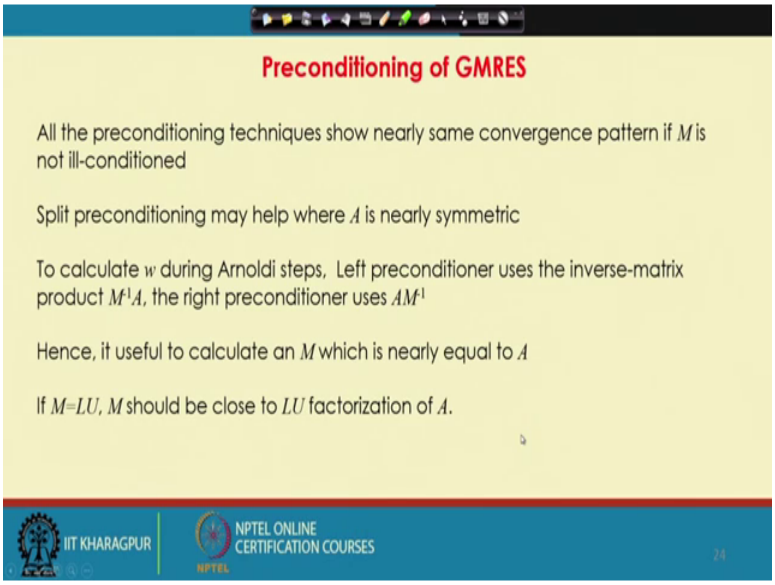
However, we have demonstrated that different preconditioning preconditioned algorithms can be obtained for both conjugate gradient and GMRES. And the algorithms are very similar to the older algorithm. So, if you already have a code, we can very same with very less effort, we can convert it to a precondition code. Only thing this A M inverse and M inverse are to be found out.

Now, one question is that that if A M inverse is very close to identity the problem becomes much simple and the condition numbers also improve, because i identity matrix has a condition number 1, so condition number also improves and we get faster solution. So, how can we find out M inverse, which is very close to a that is a one important thing. Another thing because we need to do M inverse in certain calculations not A M inverse.

A M inverse is a one issue, M inverse is a another issue. These two things have to looked in that we have to find out an A M inverse or M inverse say for the left-preconditioned GMRES or for conjugate gradient, which was a good condition number.

So, if M is close to A, the condition number will be in close to one, and the solution will converge faster. So, this is one issue that M should be very close to A, so that M A M inverse or M inverse is close to one. The other issue is that A M inverse is to evaluated in a form, so that this equation can be well solved. And we will see we already have some idea that M is of l LU if M is of LU form, the solution of M equation involving M inverse is much simpler.

(Refer Slide Time: 12:36)



Preconditioning of GMRES

All the preconditioning techniques show nearly same convergence pattern if M is not ill-conditioned

Split preconditioning may help where A is nearly symmetric

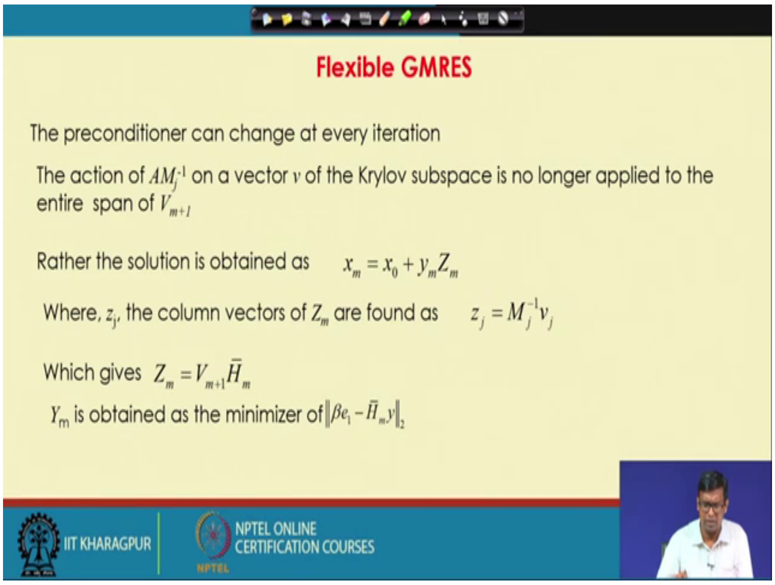
To calculate w during Arnoldi steps, Left preconditioner uses the inverse-matrix product $M^{-1}A$, the right preconditioner uses AM^{-1}

Hence, it useful to calculate an M which is nearly equal to A

If $M=LU$, M should be close to LU factorization of A.

All the preconditioning techniques show nearly same convergence pattern if M is not ill-conditioned for left right split preconditioning all these things. Split preconditioning may help, where A is nearly symmetric. To calculate w during Arnoldi steps, left-preconditioners use the inverse-matrix M inverse A, and write preconditioner use A M inverse. So, this is important that if M inverse is close to A M is close to A, M inverse is close to A M in A inverse, this will be identity and we will get very first the convergence of this method. It is useful to calculate an M, which is nearly equal to A. If M is equal to LU, m should be close to LU factorization of A.

(Refer Slide Time: 13:22)



Flexible GMRES

The preconditioner can change at every iteration

The action of AM_j^{-1} on a vector v of the Krylov subspace is no longer applied to the entire span of V_{m+1}

Rather the solution is obtained as $x_m = x_0 + y_m Z_m$

Where, z_j, the column vectors of Z_m are found as $z_j = M_j^{-1} v_j$

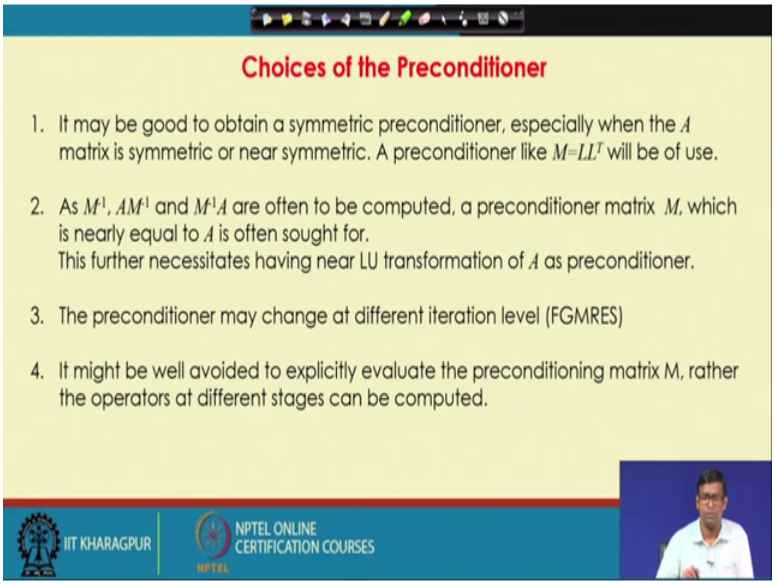
Which gives $Z_m = V_{m+1} \bar{H}_m$

Y_m is obtained as the minimizer of $\|\beta e_1 - \bar{H}_m y\|_2$

There is another method called flexible preconditioner, where the preconditioner can change in every iteration. Instead of going with a fixed M, I can changed M at different steps, so that the action M j inverse on vector V of Krylov subspace is now no longer applied to the entire span of V m j plus 1, because M j inverse is calculated at one particular step of the iteration. So, it is in application it is not applied over the over the entire Krylov subspace of vectors.

Rather the solution is obtained as x m is equal to x 0 plus y m Z m. And the Z m the column vectors of j is found out as M j inverse v j. So, it uses only one particular Krylov subspace vector and the preconditioner used at that particular step, which give Z m is equal to V m plus 1 H bar m. y m is obtained to as a minimizer of beta e 1 minus H bar m like any standard GMRES method.

(Refer Slide Time: 14:24)



Choices of the Preconditioner

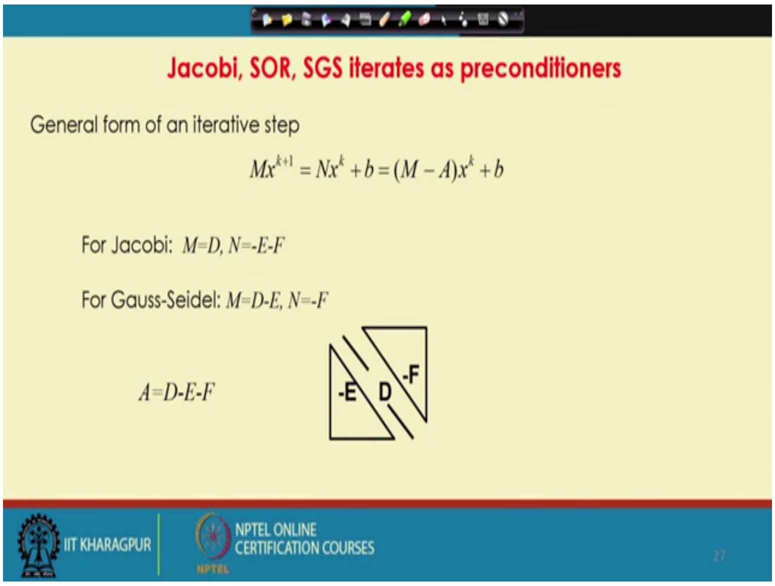
1. It may be good to obtain a symmetric preconditioner, especially when the A matrix is symmetric or near symmetric. A preconditioner like $M=LL^T$ will be of use.
2. As M^{-1}, AM^{-1} and $M^{-1}A$ are often to be computed, a preconditioner matrix M, which is nearly equal to A is often sought for.
   This further necessitates having near LU transformation of A as preconditioner.
3. The preconditioner may change at different iteration level (FGMRES)
4. It might be well avoided to explicitly evaluate the preconditioning matrix M, rather the operators at different stages can be computed.

Now, comes the issue that how the preconditioners can be chosen. It may be good to obtain a symmetric preconditioner, especially when A matrix is symmetric or near symmetric. A preconditioner like M L transpose will be of use we have seen that from our idea discussion in conjugate gradient method. As M inverse, A M inverse and M inverse A are often to be computed, a preconditioner matrix M, which is nearly equal to A is often sort for, because A M inverse and M inverse A will be close to identity there. And that is the driving part of the algorithm, because we find out Krylov subspace of A M inverse and M inverse A and the residual. If this is close to identity the problem becomes much simpler, this is straightforward.

This further necessitates having LU transformation of A as preconditioner, because we need some sort of LU transformation of to get M inverse, M inverse will be an LU. And A M inverse and M inverse A will be close to identity. So, a LU transformation of A, if it is obtained that can be the best preconditioner. And we can see if M is equal to l A of LU that means, M inverse is basically A inverse, the problem becomes an solution of an identity matrix and should be done in one step.

So, even if we cannot get exact LU transformation of u, because A because LU transformation is a costly operation, we should try to get a near LU transformation of A. The preconditioner may change at different iteration levels which you have seen from flexible GMRES. So, at different iteration levels we can try to recompute the preconditioner, which is also possible, if we are using flexibility GMRES. It might be well avoided to explicitly evaluate the precondition matrix M, rather the operators M inverse and A M inverse at different stages, and how are the effect of the operators in this solution that can be computed. So, instead of computing the M, evaluating the M storing it somewhere, we should rather see how is the matrix product of A M inverse into something or M inverse into something can be directly used.

(Refer Slide Time: 16:44)



Jacobi, SOR, SGS iterates as preconditioners

General form of an iterative step

$$Mx^{k+1} = Nx^k + b = (M-A)x^k + b$$

For Jacobi: $M=D, N=-E-F$

For Gauss-Seidel: $M=D-E, N=-F$

$A=D-E-F$

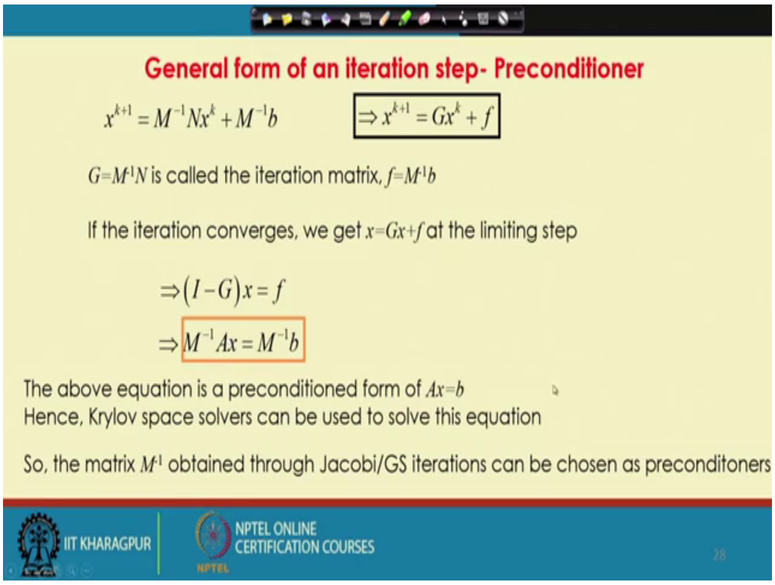
IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

27

So, now we will look into few preconditioning methods. And one is called a Jacobi type preconditioner, and that is that we look into how M can be evaluated using Jacobi or SOR iterations can they be used as they be used as preconditioner. The general form of an iterative step is the M x k plus 1 is equal to N x k plus b, which is M minus A x k plus b. For Jacobi M is equal to the diagonal matrix, N is minus E minus F. For gas Gauss-Seidel M is equal to D minus E, N is equal to minus F. So, D is split into upper the matrix A is split into diagonal upper triangular and lower triangular matrix. This is we have done in when long back when you are discussing about basic iterative methods.

(Refer Slide Time: 17:37)



General form of an iteration step- Preconditioner

$x^{k+1} = M^{-1}Nx^{k} + M^{-1}b$ $\boxed{\Rightarrow x^{k+1} = Gx^{k} + f}$

$G=M^{-1}N$ is called the iteration matrix, $f=M^{-1}b$

If the iteration converges, we get $x=Gx+f$ at the limiting step

$$\Rightarrow (I-G)x = f$$

$$\Rightarrow \boxed{M^{-1}Ax = M^{-1}b}$$

The above equation is a preconditioned form of $Ax=b$
Hence, Krylov space solvers can be used to solve this equation

So, the matrix M^{-1} obtained through Jacobi/GS iterations can be chosen as preconditoners

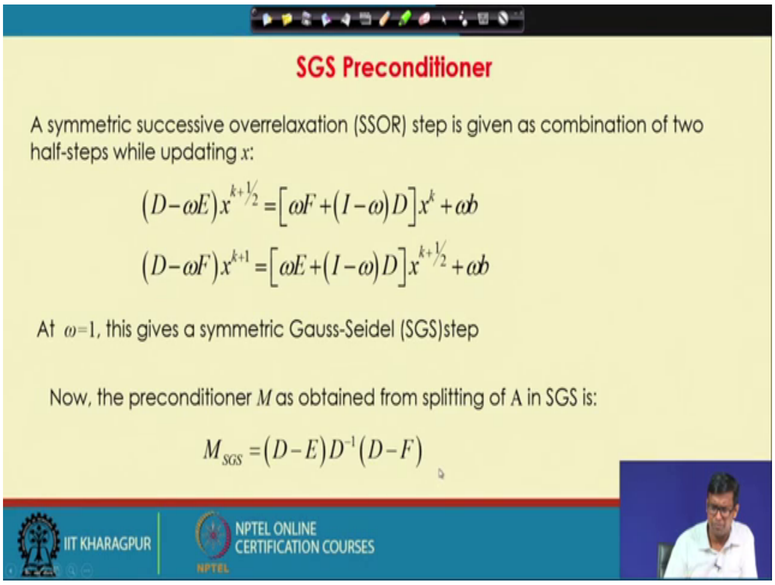
And iteration step is written as x k plus 1 is equal to G x k plus f. G which is equal to M inverse N is called the iteration matrix and f is equal to M inverse b. If the iteration converges, we get x at the converging step we will get x is equal to G x plus f or the equation will be I minus G x is equal to f, which is same as if we look into this splitting M inverse A x is equal to M inverse b. So, we got it that converging Jacobi or Gauss-Seidel iteration gives us exactly same what we are looking for as a precondition system of equation. So, if instead of solving A x is equal to b, if we can do few Jacobi iterates and get a M inverse A x is equal to b, which will be already a precondition system of equation. And we will probably have faster convergence.

The other thing is that how will M inverse look here. And specially if we can have a LU decompose form of M inverse M inverse or also you can see whether M inverse is close to A, if M inverse is same as A this is x is equal to A inverse B, we have already arrived at the solution, so that we will also look into. But, this equation is already a preconditioned equation, which is the converging step of Jacobi iteration or Gauss-Seidel iteration.

So, as we use Gauss-Seidel or Jacobi iterates, we are actually preconditioning the equation system. The above equation is the precondition form of A x is equal to b. Hence, Krylov space solvers can be used to solve this equation. So, one very is very smart very quick approach will be do few Jacobi or Gauss-Seidel iterations, get something like an M inverse A x is equal to M inverse b. This M inverse is not exactly same this is as your, I minus G M inverse A, because you one unless convergence you have not reached into x is equal to G x plus f, but this will be close to that. So, this will be a precondition system and solve it using Krylov space solvers.

So, the matrix M inverse obtained through Jacobi or GS iteration can be chosen as the precondition, and that is the another idea that M inverse that which is the Jacobi iteration matrix I minus G matrix that can be chosen as a preconditioner and can be applied in the Krylov space based solvers conjugate gradient or g GMRES. This M inverse, which is I minus G can be taken and applied there.

(Refer Slide Time: 20:36)



SGS Preconditioner

A symmetric successive overrelaxation (SSOR) step is given as combination of two half-steps while updating x:

$$(D-\omega E)x^{k+1/2}=\left[\omega F+(I-\omega)D\right]x^{k}+\omega b$$

$$(D-\omega F)x^{k+1}=\left[\omega E+(I-\omega)D\right]x^{k+1/2}+\omega b$$

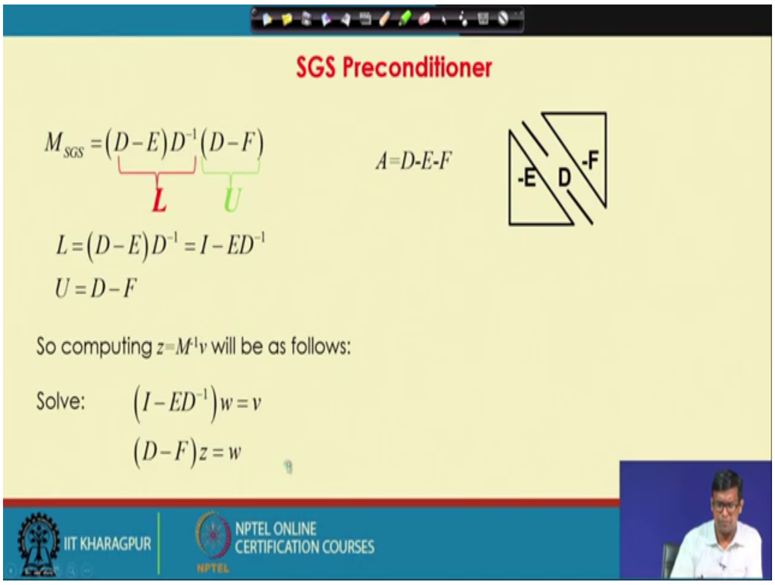
At $\omega=1$, this gives a symmetric Gauss-Seidel (SGS)step

Now, the preconditioner M as obtained from splitting of A in SGS is:

$$M_{SGS}=(D-E)D^{-1}(D-F)$$

Now, we have to see whether this is symmetric also, because symmetricity is recovered in some cases. A symmetric method can be thought of symmetric successive over relaxation steps. We have discussed about successive over relaxation, but this is successive over relaxation combining two steps. One is combining the lower triangular already updated part of the vectors. And another if updating the other with other elements of the vector, and then getting the final update using this updates here. So, this is the so SOR is broken into two half steps, which is called symmetric preconditioning. It w is equal to 1, this give a symmetric Gauss-Seidel or SGS step. And the matrix M obtained from SGS is D minus E D inverse D minus F.

(Refer Slide Time: 21:25)



SGS Preconditioner

$$M_{SGS} = \underbrace{(D-E)D^{-1}}_{L}\underbrace{(D-F)}_{U}$$

$A=D-E-F$

$$L=(D-E)D^{-1}=I-ED^{-1}$$

$$U=D-F$$

So computing $z=M^{-1}v$ will be as follows:

Solve: $(I-ED^{-1})w=v$

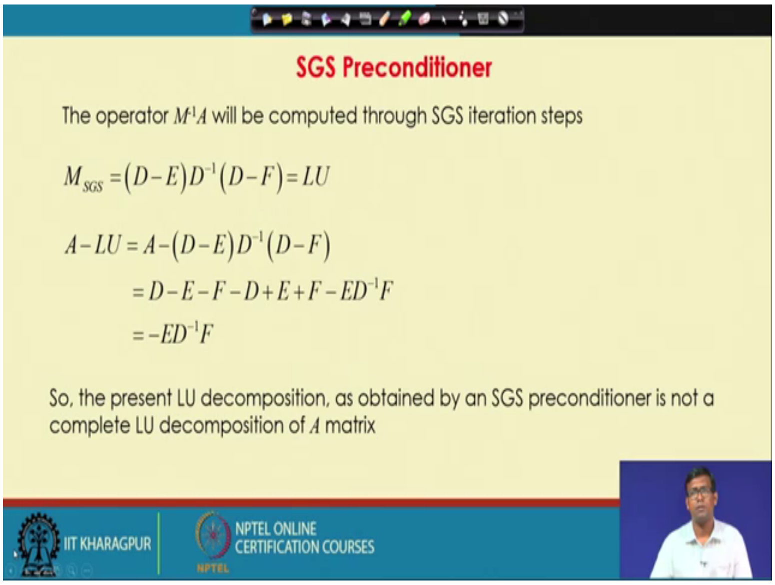
$(D-F)z=w$

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES

If we look into the M matrix the M inverse A that M matrix A is equal to m minus n, the M matrix obtained from SGS is D minus E D minus E, which is the lower triangular matrix. D is a diagonal, so D inverse is a diagonal, so this is a lower triangular into D minus F, which is an upper triangular matrix. So, M SGS if we use symmetric Gauss-Seidel that is symmetric successive over relaxation with omega is equal to 1, we get an M matrix which is of LU form. L is equal to D minus E D inverse, which is I minus E D inverse. U is equal to D minus F.

Now, if we have to find out z is equal to M inverse v, some M inverse operation has to be done. The first operation will be I minus, because this is a LU. So, L w is equal to v, which is I minus E D inverse w is equal to v, and because this is lower triangular matrix, the there will be exactly n steps to solve this equation. And the next step will be the U bar D minus F sorry D minus F z is equal to w, which will give us this is upper triangular, this will be again n steps. So, very quickly we can get solution of z is equal to M inverse a v, if we have the M SGS form the LU decomposition through SGS preconditioning.

(Refer Slide Time: 22:58)



SGS Preconditioner

The operator $M^{-1}A$ will be computed through SGS iteration steps

$$M_{SGS} = (D-E)D^{-1}(D-F) = LU$$

$$\begin{aligned} A-LU &= A-(D-E)D^{-1}(D-F) \\ &= D-E-F-D+E+F-ED^{-1}F \\ &= -ED^{-1}F \end{aligned}$$

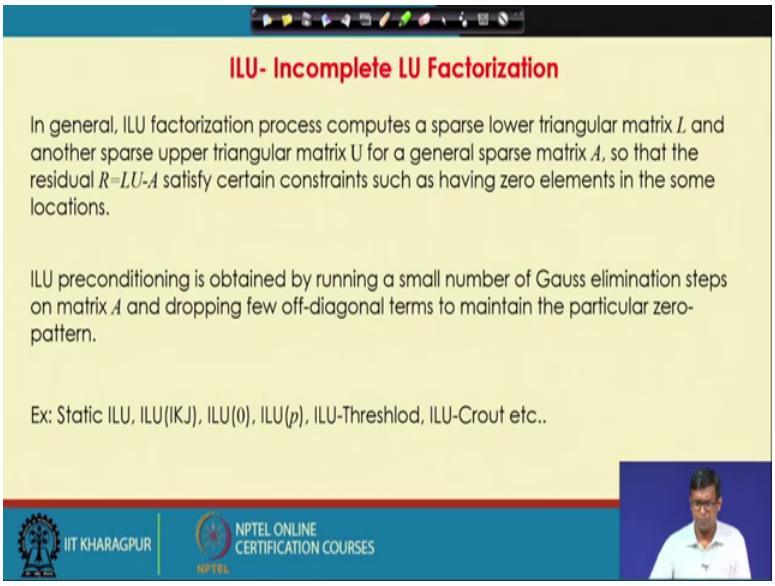
So, the present LU decomposition, as obtained by an SGS preconditioner is not a complete LU decomposition of A matrix

The operator M inverse A will be computed through the SGS iteration steps, M is equal to LU. And then if we write, so this will become that means, that as we will do some of the SGS iterations, we are actually using applying the operator M inverse S. We do not need to explicitly find out M inverse and multiply it with A. Some of the SGS preconditioning steps, we will take care of M inverse A operation multiplication or the operator is already updated.

Now, this M inverse is not exactly M inverse. So, if we find out M SGS and A minus LU this is a LU A minus LU, we get a solution which is E D minus E D inverse F. So, this is the error in the LU decomposition. The LU decomposition of M is not same as LU decomposition of A. There is some difference, which is minus E D inverse F. And if these as these A is a diagonally dominant matrix, if D is much larger, this is a small the values of this matrix are smaller. So, we can get small error here. And we will say that this is not a complete LU decomposition rather than incomplete LU decomposition of A.

However, a preconditioner matrix can be evaluated from this and applied to solve the equations. So, the present LU decomposition, as obtained by an SGS preconditioner it is not a complete LU decomposition of A matrix. So, we can use SGS as a preconditioning technique or symmetric Gauss-Seidel as a preconditioning technique for applying in conjugate gradient or GMRES type of solver. We can obtain the M inverse A steps as we brings few SGS iteration steps on the equation system. And M inverse if you have to find z is equal to M inverse w, that also can be very easily found out as M is of already in a LUr of l LU shape.

(Refer Slide Time: 25:08)



ILU- Incomplete LU Factorization

In general, ILU factorization process computes a sparse lower triangular matrix L and another sparse upper triangular matrix U for a general sparse matrix A, so that the residual $R=LU-A$ satisfy certain constraints such as having zero elements in the some locations.

ILU preconditioning is obtained by running a small number of Gauss elimination steps on matrix A and dropping few off-diagonal terms to maintain the particular zero-pattern.

Ex: Static ILU, ILU(IKJ), ILU(0), ILU(p), ILU-Threshlod, ILU-Crout etc..

So, and this we saw that this is an incomplete LU factorization. In general, and ILU factorization process computes a sparse lower triangular matrix L. So, we will come to a general definition of LU ILU or a incomplete LU factorization. In general, ILU factorization process factorization process computes a sparse lower triangular matrix L, then another sparse upper triangular matrix U all for a general matrix A.

Similar thing we obtained from the SGS preconditioning also. The exact form of lower triangular matrix and exact form of upper triangular matrix we obtained from a general sparse matrix A, so that the residual R is equal to LU minus A, which is a residual that certain satisfies certain constants. One constant is that we might, because this is a sparse matrix, similarly R will also be a sparse matrix. And we might like to have zero elements in some locations of our matrix. So, it will have certain advantage in storage of the matrix, and certain advantage in applying the matrix.

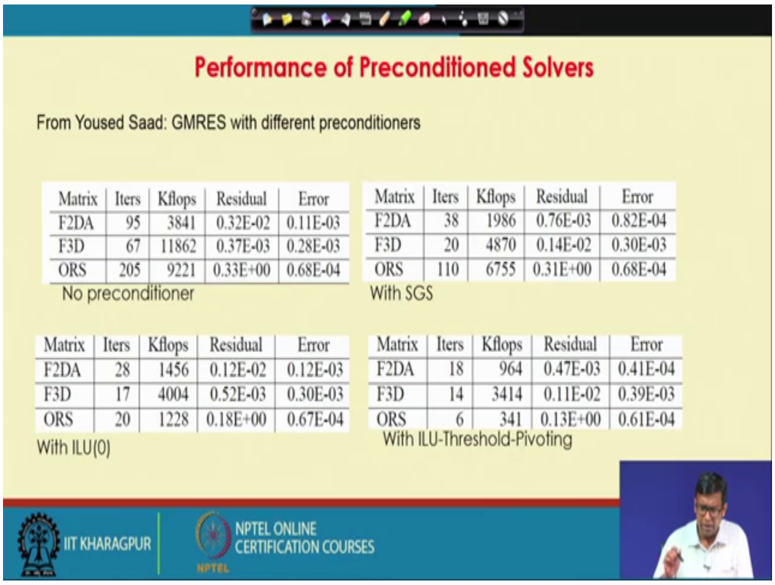
This is the general idea of LU incomplete LU factorization that complete LU factorization is difficult, it needs to run gauss elimination like steps for n queue operations. So, use some other method like a SGS preconditioning or like first few steps of Gauss elimination you run and get, few of the rows to be of LU form. So, some of these methods were an incomplete LU decomposition is obtained, and which will have a constant that some zero elements are obtained in some of the locations.

ILU preconditioning is obtained by running a small number of Gauss elimination steps on matrix A snd then dropping few off diagonal terms to maintain the particular zero-pattern. So, we said like few gauss eliminate and few gauss elimination steps, and drop the few off-diagonal terms to get particular zero- patterns, because we are not getting exactly LU decomposition of A.

But, we are getting two matrices which are of L and U pattern and A minus LU can have some error, and that because exact LU is a obtained we do not need to do anything the solution is obtained M inverse A is equal to identity. When exact LU is not obtained, still M inverse A is close to an identity matrix in a sense that the condition number is very close to one, so that will also work.

And that is the idea of ILU preconditioning runs something like a Gauss elimination type of method, and drop few of the off-diagonal terms to get a particular zero-pattern in the residual make matrix. And also to get low a LU form of the precondition matrix. And there are several methods like static ILU, which is basically running Gauss-Seidel for few steps ILU IKJ, which is reordering the LU decomposition method. ILU 0, where there will be no extra digits added in the final LU matrix. A and LU will have a almost similar sparsity ILU p, ILU-Threshold, ILU-Crout different methods are there.

(Refer Slide Time: 28:25)



Performance of Preconditioned Solvers

From Yoused Saad: GMRES with different preconditioners

Matrix	Iters	Kflops	Residual	Error
F2DA	95	3841	0.32E-02	0.11E-03
F3D	67	11862	0.37E-03	0.28E-03
ORS	205	9221	0.33E+00	0.68E-04

No preconditioner

Matrix	Iters	Kflops	Residual	Error
F2DA	38	1986	0.76E-03	0.82E-04
F3D	20	4870	0.14E-02	0.30E-03
ORS	110	6755	0.31E+00	0.68E-04

With SGS

Matrix	Iters	Kflops	Residual	Error
F2DA	28	1456	0.12E-02	0.12E-03
F3D	17	4004	0.52E-03	0.30E-03
ORS	20	1228	0.18E+00	0.67E-04

With ILU(0)

Matrix	Iters	Kflops	Residual	Error
F2DA	18	964	0.47E-03	0.41E-04
F3D	14	3414	0.11E-02	0.39E-03
ORS	6	341	0.13E+00	0.61E-04

With ILU-Threshold-Pivoting

And we can see from Yoused Saad book, what is the effect of preconditioner when we what is the advantage, when you apply preconditioner, GMRES is applied over three different matrices. One is a finite difference 2D matrix or finite difference over a 2D geometry matrix obtained from finite difference over the 2D geometry of a integral diagonal matrix. A septa diagonal matrix finite difference obtained over 3D geometry. And a ORS, which is the (Refer Time: 28:56) structured geometry.

In the first when we do not apply any preconditioner, the matrix this is a little larger matrix 100 by 100 type of matrix. It took 95 iterations for the first case, where number of flops were three 3841 kilo flops into 10 to the power 3. To get and residue all of the order of 10 to the power minus 2 and error is 0.1. F 3 D the other matrix took 67 iteration. ORS took 205 iterations to get certain value of residual and error we said that its close to convergence.

And the number of Kflops is thirty eight hundred 11862 9200. When we apply SGS preconditioner from 95, the number of iteration reduces to 238. The kilo flops um performance is almost half of that while maintaining a residue and error, actually of all almost of same order a little smaller here you can meet SGS. Similarly, F3D in the number of iteration from 67 goes to 20 to get similar order of convergence. From ORS from 205, it reduces to 110. So, almost half or more than half number of iterations are required. And also substantial improvement is obtained in terms of kilo flops. It does not exactly scale with the number of iterations, because some preconditioning steps some SGS iterative steps are already added.

So, exactly does not scale like that, but a high a significant decrease in the computational over step steps flops, and the number of iteration steps are improved. When you use ILU 0, the performance is further improved, especially from ORS from110, it goes to 20. So, instead of the SGS if you used Gauss elimination type of LU decomposition, the performance specially for ORS for non-structured sparse mesh sparse matrix is improved. And if you use further improved ILU, which is ILU-threshold pivoting, which was taking 205 steps in ORS, it now takes only 6 steps. And it was taking 9200 kilo flops, now it takes 341 flops.

So, a huge improvement in performance can be obtained. So, preconditioning if properly computed and properly applied over Krylov space base solvers, can give us extremely for solvers. However, we have now we have discussed some ideas about preconditioning, how the solvers are modified if preconditioners are obtained, and some very basic ideas how to obtain preconditioning. But, the evaluation of precondition are especially ILU tp type of recently developed preconditioners, need more involved mathematics and some of the graph partitioning type of approach that how you will keep the properties of LU at certain level, how will you minimize the residual etcetera, which you have not discussed.

But we try to demonstrate that applying preconditioners can heavily improve the performance of a Krylov space solvers. And you can see that what was taking 95 iterations take a 18 iteration, and what was taking 205 iterations for a very much sparser matrix takes only 6 iterations to solve. So, and magnificent improvement in performance can be obtained if preconditioners can be applied properly, because the conditioning number of the solution matrix is can be very well adjusted, so that it comes very close to 1. And in very less a few number of steps we get the right solution.

So, I try to give you a brief overview of preconditioners, but it actually requires much more discussion can be probably module itself in a course can come as preconditioner 2-3 weeks, one can discuss over the preconditioners. And however, I think this gives one route to a obtain very good performance of Krylov space based matrix solvers or accelerate the matrix solvers to get to show great performance one of the demonstration is through preconditioners.

Thank you.