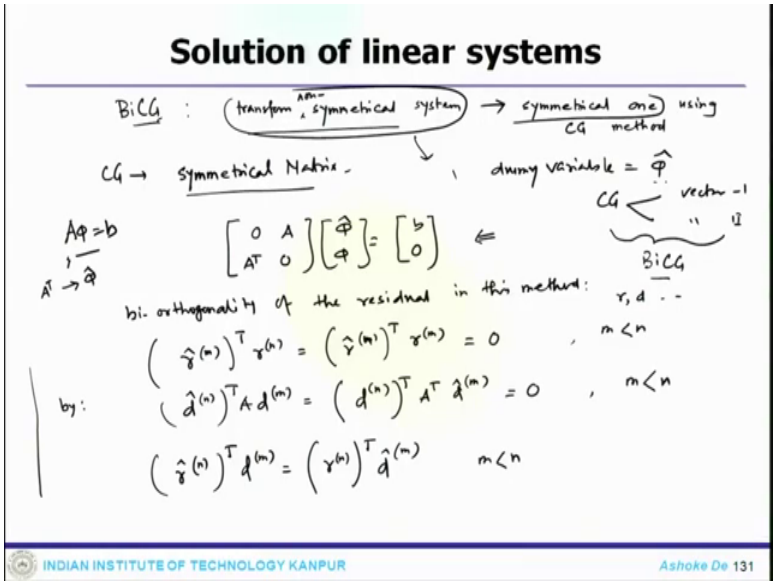
Introduction to Finite Volume Methods – II
Prof. Ashoke De
Department of Aerospace Engineering
Indian Institute of Technology, Kanpur

Lecture – 07
Linear Solvers – VII

Welcome back and we will still continuing our discussion on iterative methods. So, where we stopped in the last lecture is on the conjugate gradient and today, we are going to discuss on the slightly advanced version of conjugate gradient or a separate class of that kind of method call the bi-conjugate gradient and why it is required because as we have seen in CG method this is more for the symmetrical system, but practically the large systems that we solve as a linear system they are non-symmetrical.

So, once you have a non-symmetrical matrix you need to do the transformation using the technique that we have used in the CG and then the new process or the method is called as the bi CG.

(Refer Slide Time: 01:11)



So, what we look at today is that the in the bi CG approach essentially you transform the symmetrical matrix or transform the non-symmetrical system to a symmetrical one by using CG or conjugate gradient method. So, here the process or the underlying phenomena will remain pretty much similar, but what happens since all your realistic

problems are asymmetrical in nature; they are not going to give you the symmetrical matrix. So, symmetrical matrix only the CG can solve symmetrical matrix.

So, if you do not have a symmetrical matrix, like a non symmetrical or anti symmetric system which is essentially the outcome of your realistic system in that case CG cannot be applied. So, what you need to do? You need to this realistic system you need to transform them to a symmetric one. So, essentially what you do? You first transform them to a transform the non symmetrical system to a symmetrical one and then and the way it is done.

So, you have so, you introduce some sort of an dummy variable dummy variable here or original system of equation is A phi equals to b. Now, you introduce a dummy variable which is phi hat then the transformation takes place like 0 A A transpose with phi hat phi equals to b 0. So, that is what you do that and using these dummy variable you can through this particular process you can convert this unsymmetrical or system to a symmetrical one.

Now, when you apply to this kind of system, this kind of transformation process so, the actual CG process goes or results into two sequence of CG like vectors. So, this is vector 1 and this is vector 2. So, theoretically this particular one splits into two different vectors which are looks like an CG vectors and then the ordinary sequence based on the original system with the coefficient matrix A from A you calculate phi and from A transpose you calculate the dummy variable phi hat.

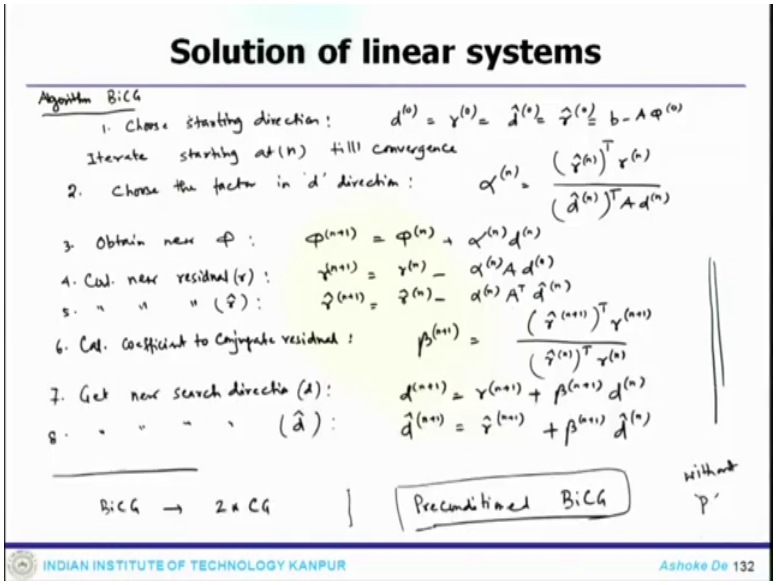
Now, because of these two series of vector the name has been coined as bi conjugate gradient. Now, you have the when you looked at the conjugate gradient you have all the series of residual vectors, you have the direction vectors like you have r, d and all sort of vectors. Now, once you form the bi orthogonality of the residual in this system in this method that can be achieved by this kind of equation like r hat at m transpose r n equals to r hat n transpose r m equals to 0, where m is less than n.

Now, the bi conjugacy of the search direction is fulfilled by the equation d hat n transpose A d m equals to d n transpose A transpose d hat m which is again 0, where m is less than n. Now, further the sequence of the residuals and the search directions are constructed such that the ordinary form of one which is orthogonal to the shadow form; so, this is the shadow form the A hat form is the shadow form the mathematically one

can represent like r hat at the level n transpose d m equals to r n transpose d hat m, where m less than n.

Now, this is how now several variants of this particular method which has irregular, convergence with the possibility of breaking down and those are developed. Now, if you this is the modification which come due to this shadow variable or the augmentation of this variable with the dummy variable phi hat and then accordingly it will be change in the algorithm.

(Refer Slide Time: 07:15)



So, if you look at the algorithm for bi conjugate gradient the algorithm gets now modified. So, what you do? You first choose starting direction. So, that is what the initial thing one has to do. So, choose the starting direction like d naught equals to r naught equals to d hat naught equals to r hat naught b minus A phi 0. Now, then you, obviously, you need to iterate starting at n till convergence. So, that one has to do. So, you have to iterate that till you get the convergence.

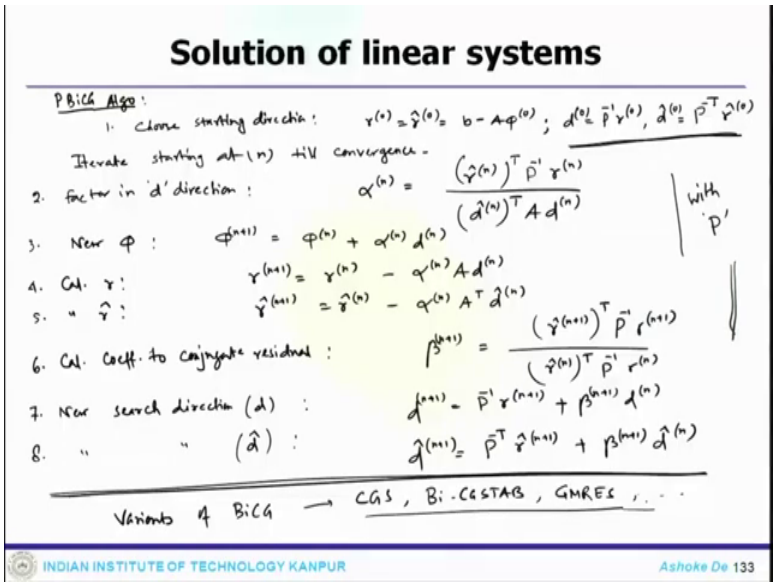
Then, next level you choose the factor in d direction. So, which will lead to calculation of the alpha. So, alpha n would be now r hat n transpose r n divided by d hat n transpose A d n. Now, then you obtain new phi like phi n plus 1 equals to phi n plus alpha n d n. Now, calculate new residual r like r n plus 1 equals to r n minus alpha n A d n. Similarly, you calculate the new residual r hat which is like r hat n plus 1 equals to r hat n minus alpha n a transpose d hat n.

So, using the dummy or shadow matrix A transpose you get the. So, now, it is time to calculate the coefficient to conjugate residual and that is through beta n plus 1 which is going to be now r hat n plus 1 transpose r n plus 1 r hat n transpose r n and then you can get new search direction that is d which will be d n plus 1 is r n plus 1 minus beta n plus 1 d n and similarly you get the new search direction along the shadow variable and that is d hat which is going to be d hat n plus 1 equals to r hat n plus 1 beta n plus 1 d hat n.

So, as you see the difference from the conjugate gradient here the search direction, residual direction these are now splitted into two different directions and that is why it calls the bi CG. So, immediately one can see bi CG process takes the time what the computational time is twice what CG is takes because of this.

Now, also as we have seen that for the conjugate gradient that one can have precondition conjugate gradient in this case also it is possible to have preconditioned bi CG which can actually improved the faster convergence or which can improve the rate of convergence and make it faster. So, that is why sometimes it is preferred to use some kind of a preconditioning and particularly for this case also you can have the precondition bi CG and the difference would come from the algorithm of the.

(Refer Slide Time: 13:19)



So, this would be the precondition bi CG algo and again you need to start with the choose with the starting direction. So, choose starting direction; so, that means, you

calculate r naught r hat naught equals to b minus A phi naught and d naught equals to P inverse r naught and d hat naught equals to P transpose r hat naught.

So, this is the difference one can see immediately it deviates from the standard bi CG algorithm because of this preconditioning matrix. Now, you can iterate starting at n till convergence, then the next level is to you choose the factor in d direction. So, factor in d direction. So, how do you find out? That it would be alpha n equals to r n transpose P inverse r n then d hat n transpose A d n. So, that is how you get it.

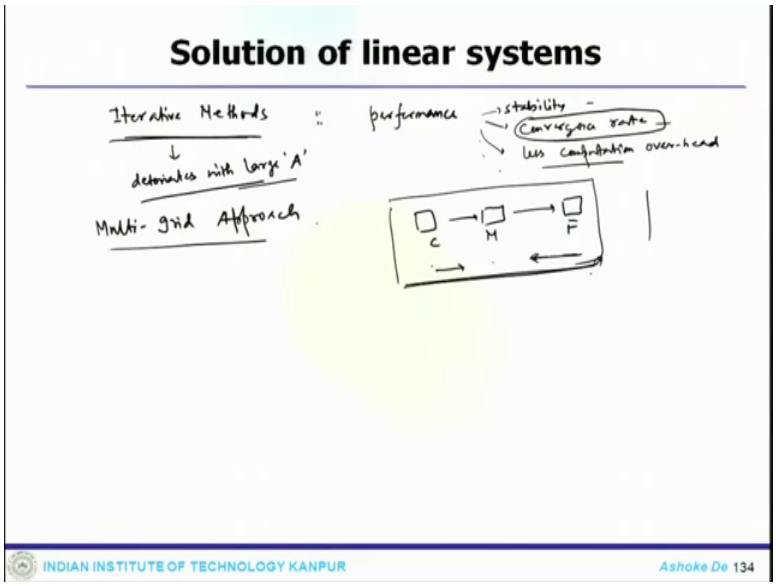
So, here if you see because of this inclusion of the preconditioning system it slightly deviate from the standard bi CG algorithm. Then you obtain the new phi such that phi n plus 1 equals to phi n plus alpha n and d n. Now, calculate r the residual factor which would be r n plus 1 equals to r n minus alpha n A d n and also similarly calculate the r hat which is the along the direction of the shadow variable. So, that will give us r hat n plus 1 equals to r hat n minus alpha n A transpose d hat n, ok. Now, then as usual you need to calculate the coefficient to conjugate residual.

So, which will get you beta n plus 1 equals to r hat n plus 1 that is transpose p inverse r n plus 1 then r hat transpose p inverse r n. Then you get the new search direction that is one along the original system d which would be d n plus 1 equals to p inverse r n plus 1 plus beta n plus 1 d n and the along the d hat also. So, this would be d hat n plus 1 equals to P transpose r hat n plus 1 plus beta n plus 1 d hat n. So, that is what you get with the preconditioned system and then it differs from the CG.

So, when you apply the preconditioning matrix. So, that makes the system better convergent. So, that is the idea of using preconditioning to now by this time if you look at it we have discussed and so, many iterative methods and always there are certain level of refinement with these process. Now, also there are some variants of bi CG algorithm. Those are called CG s which is conjugate gradient square it could be bi CGSTAB that is another algorithm or GMRES generalized minimum residual process.

So, these are there are different variants of bi CG algorithm which can be applied, but the baseline algorithm remain as we have discussed here and this is with precondition and this is without preconditioner. So, this one is without preconditioner and this one is with P. And, the other variant like CGS, bi CGSTAB, GM residual baseline algorithm remains pretty much same, now only the modifications are done as per the system.

(Refer Slide Time: 20:11)



Now, we will talk about the thing is that we are talking about this iterative process or iterative methods and we have discussed a different variants of iterative methods starting from the simple one and till the bi CGS algorithm with preconditioner.

Now, still the main concern remains the performance of an iterative method; performance in terms of stability, in terms of convergence rate. So, always people look for this iterative process how quickly it converges, so that you get the close to the exact solution; how stable is that, so that it does not lead to any spurious oscillation in the solution. Also the computation less computational overhead; that means, the cost or computing cost must be less.

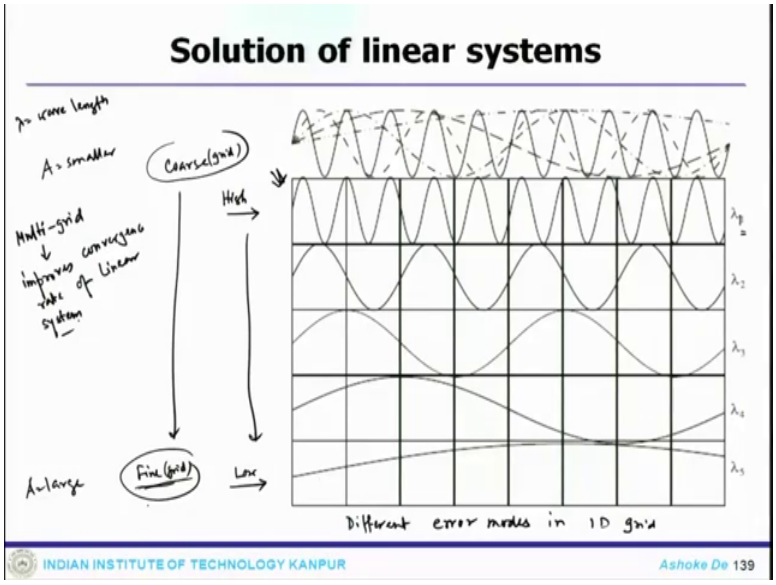
So, these are the some performance parameter, which always lead to the scientist and engineers to specially the applied mathematician to keep on working in this direction to get slightly improved methods on the iterative process. Now, in that context since we are talking about these convergence rate and these specifically the convergence rate not only has to do it which class of iterative method, one uses also it detoriates with the large A if the scale or the system size increases the conversion rate; obviously, detoriates. So, always there is an effort how to improve this convergence rate.

So, one approach is that one can adopt to some sort of an multi-grid approach. So, which means what you do here your computational grid is sort of get into a different level of refinement one could be coarse, then it could be the medium, it could lead to the finer.

So, you can have solution from here then interpolate here get a solution here or come back in these directions. So, that is how you can improve the solution of the overall system at a much faster rate.

So, what one can think about when you actually distribute the things in the different grid level and you get a solution and then from there you move to the other grid level so the convergence can be always improved. Now, when you say that, with that kind of approach you one can improve that now what happens when this kind of system you see an example some in a 1-dimensional grid what happens to that.

(Refer Slide Time: 23:33)



This is an picture which shows the error modes or different error mode modes in 1D grid. So, what it does is that, this is the different level of frequency or the wave frequency you could see, and these are the high frequency oscillation. This is high and then if you come this is much low frequency system or looks like an oscillatory system, low oscillatory system. Now, what does that do? It is essentially it is shown when you have two coarse; so, this is the high to low frequency oscillation which appear in the solution and that also increases from coarse mesh to coarse grid to fine grid.

So, this is one thing that we keep on talking that your grid refinement can have lot of impact on the solution, because not only from this kind of error point of view of the solution. Once you refine the grid your grid spacing reduces that leads to I mean that

reduces your truncation error which is one of the major error or leading error in any numerical approximation.

Now, in a coarse grid you can see the high frequency of short wavelength lambda; these are the wavelength. So, lambda here is the wavelength and we can see that the high frequency of short wavelength lambda to low frequency of long wavelength lambda phi. So, this is how high to low you have move and they are shown in these, where high frequency errors essentially appears oscillatory over an element and easily sensed by the iterative methods. So, these are the errors which are sort of sensed by your linear solver.

Now, frequency of the error also decreases as the lambda increases and you can see the error becomes slowly as your lambda increases the error becomes smoother. So, here it is much more high frequency oscillation slowly smoothening out then it smoothening out and finally, you get a much smoother profile and this gets worse. So, error becomes increasing which is smoother over the grid as a only one small portion of the wavelength live within any cell. So, these gets what is as the grid is further refined leading to high number of equations and explaining the degradation in the rate of convergence as the size of the system increases.

So, one hand we talk about that we should have a grid refinement because that reduces the truncation error, but once you start refining the grid the matrix system, this will be A would be much larger and here A would be much smaller. So, this will also become a problem. So, this is where the multi-grid approach becomes handy to kind of take care of these things. And, the so the multi grid approach actually what it does it improves the convergence rate convergence rate of linear system.

So, we will stop here today and will take from here in the follow-up lectures.

Thank you.