Computational Fluid Dynamics

Prof. Sreenivas Jayanti

Department of Chemical Engineering

Indian Institute of Technology, Madras

Module No. # 05

Solution of linear algebraic equations

Lecture No. # 29

Topics

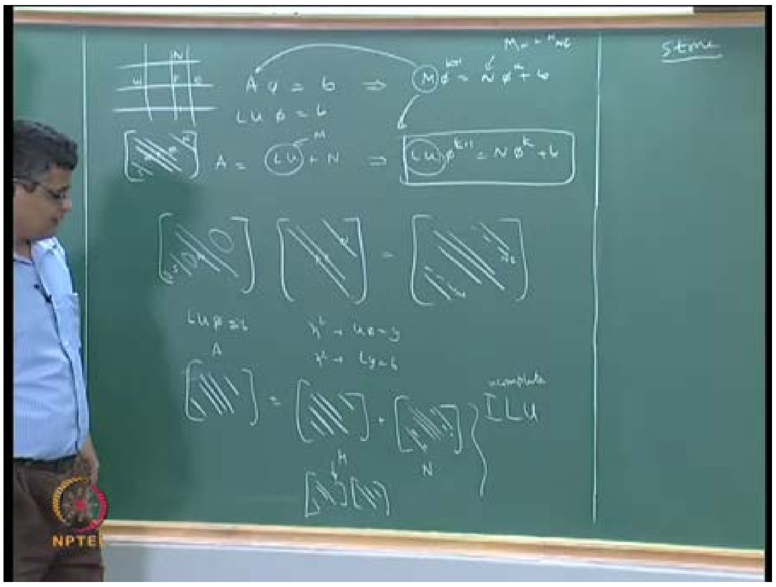
Advanced iterative methods

Strongly implicit procedure

Conjugate gradient method

Multigrid method

(Refer Slide Time: 00:19)



Let us see, how we can bring in implicitness into the solution. If we take a A phi equal to b, we know one way of introducing implicitness, we break it as L u phi equal to b, and then we go through the solution as u phi equal to y, and L L y equal to b type for a approach. This is a fully implicit solution, but we know that the decomposition of A into L u itself is time consuming process, it consumes too many operations. So, we want to write this as say M phi equal to N phi plus b, which is the standard way of the iterative procedure, and we put this as k plus 1, and k here; and we for good convergence, we would like this M here to resemble A as much as possible. So, and that is since, M is being evaluated k plus 1; that means that this M represents the degree of implicitness in this iterative solution, and if M is equal to A, then N must be equal to 0, and we have fully implicit method. So, we cannot afford a fully implicit method, because the best methods have N cubed number of operations, for the general case.

So, we want to make M to closely resemble A, but at the same time not take up as much time for computation. So, the basis for this is that we we put A equal to some L u plus say N, where Lu here is M; and therefore, write this as this equation as L u phi N plus 1 phi k plus 1 equal to N phi k plus b. Now the advantage of this approach is that we know that the solution of this is easy that is to go from k to k plus 1 is efficient, because the solution of Lu decomposition kind of method is relatively easy, and it requires the only the back substitution and forward substitution.

And if we further impose this condition that in the matrix that we are looking at, the L has non-zero values only taking a two-dimensional case, we have five diagonals. So, A has a five diagonal structure, corresponding to point p, this is point p, we have east, west, north and south. So, this is the p diagonal, and this is the west, and this is the east, west, south and north. So, there are five diagonals here. And therefore, we say that the Lu decomposition here must have only this, and wherever we have the west and south, and similarly the upper triangular matrix of this must only contain three diagonals here; this is the p diagonal, and east, and north. So, and the advantage of this is and rest are 0's.

The advantage is that even the Lu decomposition back substitution will not require N square number of operation, because this one has only one variable, and this one has only two non-zero variables. So, the back substitution is also not very costly, and the forward substitution is also not… This is a forward substitution and back substitution here sorry this is the back substitution and forward substitution here. So, the typically for a common Lu type of phi equal to Lu phi equal to b type of problem will require N square number for the solution of u phi equal to y, and N square number of operations for Ly equal to b, that is the case, when u is filled up and L is filled up. But by imposing the condition that u and L have non-zero values, only in the along the diagonals, where we have these non-zero things here; we are reducing the number of mathematical operations that are required here, so that the solution of this is even faster and it requires much less number of back substitutions.

So, the key to making something like this, this kind of approach will work provided we have an easy way of determining this L and u; the components of this L and these diagonals here and these diagonals here; and provided L u is as close to A as possible or provided N as 0 values, so that the rate of convergence is also faster. And and provided the solution of this L u equal to L u equal to this whole thing is evaluated fast. So, we are taking care of the fast process of forward substitution, back substitution by requiring these to be only very few diagonals, just as those diagonals in this A here. And by taking advantage of the sparsity and the structure of this, it is possible to arrive at efficient ways of evaluating the components of both L and u matrix like this. And if you if you want to put this, if you now evaluate the product of L u here, then you not only have these three diagonals, these five diagonals, but you also have a diagonal here and the diagonal here.

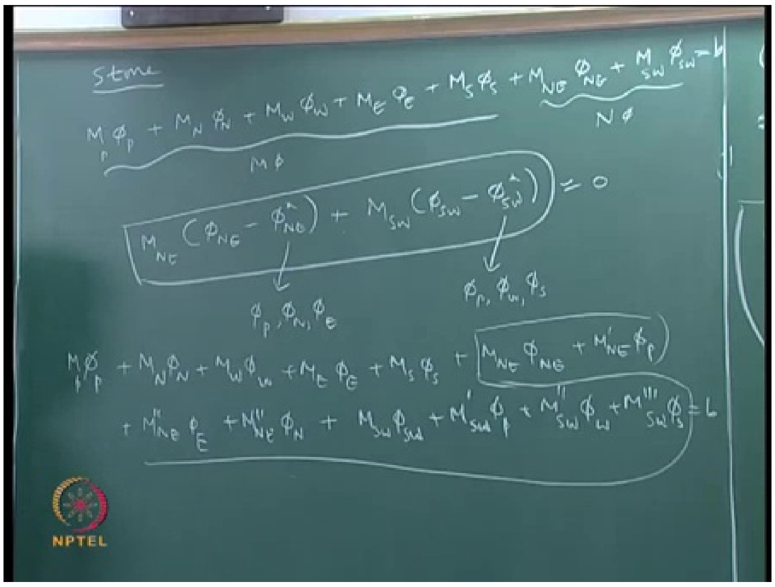
So, this corresponds to south west and north east. So, this is p, east, north and west and south. So this, so the true product of the L and u that we have, that we can find out from in this decomposition will have not five diagonals, but will have seven diagonals. So, the way of writing this thing in in this way is to say that these five diagonals here constitute M, and this extra diagonals constitute N. So, the N contains the M of S W, and M of N E, and the rest of the five are what are actually there in this. So, if we do it like this, then we can come up with an iterative scheme, but that type of approach where you break the break up the A into N M plus like this.

So, this is our M, and this is our N, this contains the south west and north east, and this contains the main diagonals; and this is further broken up into L here and u here, so that we can now come up link with this. This is known as Incomplete L u decomposition - I L u incomplete L u decomposition. And this the rate of convergence of this particular method is not very great, because we are still having significant number of these things and that information is being added only at the k eth iteration, and not at the k plus 1th iteration. So, there is a significant amount of explicitness in in decomposing A into M plus N in this way, the five diagonal A is being made into a five diagonal, which is decomposed into a three diagonal L and u, and the two additional diagonals like this. This incomplete Lu L u decomposition method although it is implicit, there is more implicitness into the overall iterative solution here, it is not implicit enough that the rate of convergence is significantly enhanced to give you overall number of reduced number of mathematical operations.

So, it is in this context stone, I think in 1964 proposed 1968 proposed a reallocation of this A influence into M and N; what his idea was that just putting only the true elements here, and putting all the other five elements is is too straight forward, and its obviously, not not leading to this. Is it possible to convert to select further elements? To bring in more elements into the N, in exactly the same points as were A also has these non-zero elements; so, that is the elements, the structure looks similar. So, the coefficients of these diagonal values are chosen such that the overall contribution of N becomes negligible.

So, the so what the variation that stone has brought into it into the incomplete Lu decomposition is to not only make N have those extra two diagonals, but also have the full extra five diagonals. So, N is in fact, more populated is less sparse than either A or N. But the values of these are chosen in such a way that the overall contribution of N is is made to be 0; and we can understand it by taking this thing here.

(Refer Slide Time: 12:06)


Stone

$$\underbrace{M_p\phi_p + M_N\phi_N + M_W\phi_W + M_E\phi_E + M_S\phi_S}_{M\phi} + \underbrace{M_{NE}\phi_{NE} + M_{SW}\phi_{SW}}_{N\phi} = b$$

$$M_{NE}(\phi_{NE} - \hat{\phi}_{NE}) + M_{SW}(\phi_{SW} - \hat{\phi}_{SW}) \approx 0$$

$$\hat{\phi}_{NE} \to \phi_p, \phi_N, \phi_E \qquad \hat{\phi}_{SW} \to \phi_p, \phi_W, \phi_S$$

$$M_p\phi_p + M_N\phi_N + M_W\phi_W + M_E\phi_E + M_S\phi_S + M_{NE}\phi_{NE} + M'_{NE}\phi_p + M''_{NE}\phi_E + M'''_{NE}\phi_N + M_{SW}\phi_{SW} + M'_{SW}\phi_p + M''_{SW}\phi_W + M'''_{SW}\phi_S = b$$

So we can since this has these components we have…

(No audio from 12:06 to 12:43)

So, we let us say that some discretization involving all the seven elements can be written like this, we have the p east, west, north, south, and then these these things here. So, this is what is effectively being done in in this particular way, because this Lu product will actually have seven diagonal; so we are recognizing these things. And this is what the contribution of N phi is, and this is the contribution of M phi, and we can see that in the N phi contribution, we have these things. So, what stone has suggested is that we bring in contribution from not only from the N E, but also from the other components here, which we can put as N E star. And similarly the contribution of M SW phi SW, which is coming from the fact that we are decomposing into L and U; and the contribution from the rest of the components here phi star such that this whole thing is equal to 0.

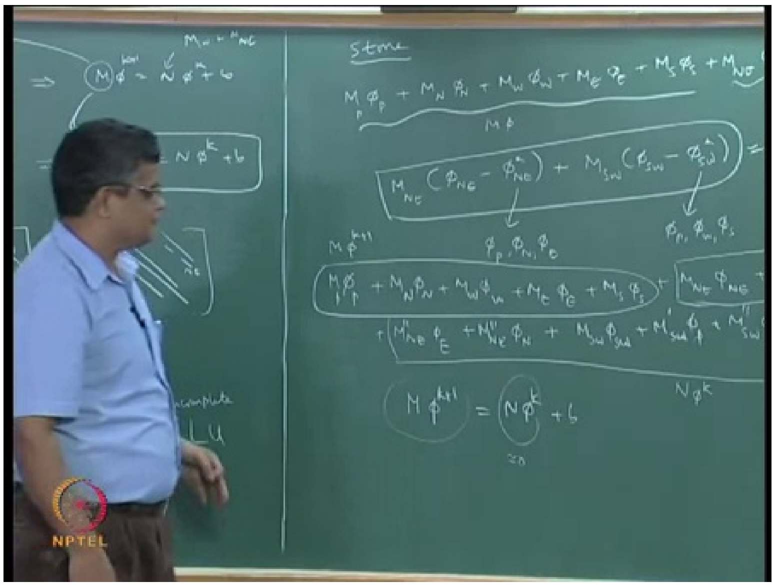
In the regular, in the incomplete Lu method, N contribution is this, what stone has suggested is that the contribution of N should be this, such that this whole thing is almost equal to 0. And if you say that this is equal to… So, we can put this here, and now what is this NE star and SW star, if these are expressed in terms of phi P, phi N and phi E; and this is expressed in terms of phi P, and phi west, and phi s. Then we will have we will have certain values here, certain coefficients. So, those coefficients will constitute the extended N n diagonals.

So, the way that is solved, this is the way that we are solving in ILu; in the way the way that we are solving in the stones method is this, M N phi N plus M West phi West plus M East phi East plus M S phi South plus M NE phi N E plus M NE phi phi P plus M NE double dash phi East plus M NE triple dash phi North plus we have M South West phi South West plus M South West prime phi p plus M prime double prime South West phi W plus m triple prime South West phi South is equal to b. So, what we have is that the essential equation, the discredited equation in the case of stones method is this, where this is M, and this is N, and these things are used to determine the L and u components.

So, the overall solution to go from k to k plus 1 will involve two… One forward and backward substitution, which are rendered East. But instead of having these N phi being equal to this, we now have we are enabling N phi to be given by this whole thing, all the way up to this. And we choose the values of M M prime, M double prime and M triple prime NE; and M prime M triple double prime and M triple prime of SW in such a way that this whole thing is equal to 0.

And it is possible to come up with those kind of approximations, assuming that the distribution the phi itself is a smooth function, which is what is we have for a typically elliptic type equations; elliptic equations is by definition or nature, it is a smooth solution. So, if it is smooth solution we can express the value at a particular point NE, in terms of the neighboring points P NE; and similarly is the expression this phi star SW can be expressed in terms of P, W and S. There can be any number of approximations; Stone has proposed some approximations by which these can be related to that. And so that overall you have you have a means of evaluating each of these coefficients and not only that, the evaluated coefficients are are evaluated subject to the condition that the contribution of all these things is equal to 0.

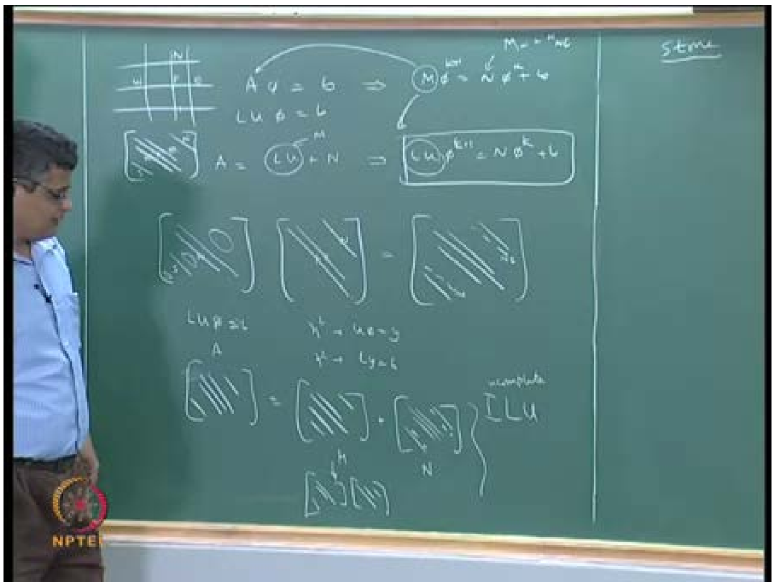
(Refer Slide Time: 18:37)



So, in the revised stones method, this is what is going into M phi, and this is being evaluated at k plus1; and this is what is going into N phi, this is being evaluated at k; and what we have on the right hand side is is there. And now what we see here? The same right hand side has a contribution from at k plus 1 coming from this, and the rest of the contribution is made almost equal to 0, in choosing this primed coefficients. So, that means that the contribution if you were to write the overall method as N phi k plus b like this, the contribution of this is 0, almost 0 and so, most of the contribution is coming from this. So, this is made more strongly implicit, because the contribution is being evaluated k plus1. So, the So, the most of the contribution is coming only from the M components; the N components are designed, so that they are roughly equal to 0.

Making use of the property of the elliptic equations that you have a smooth variation of phi, if you have a hyperbolic equation and if you have a shock, then you can have a very strong variation there; and for those kind of things, this approximation of representing this phi and star in terms of the local values here is will not be valid, but for electric equations, it is valid; you take advantage of that to rewrite this, and then we can come up with formulas for evaluating this.

(Refer Slide Time: 20:19)

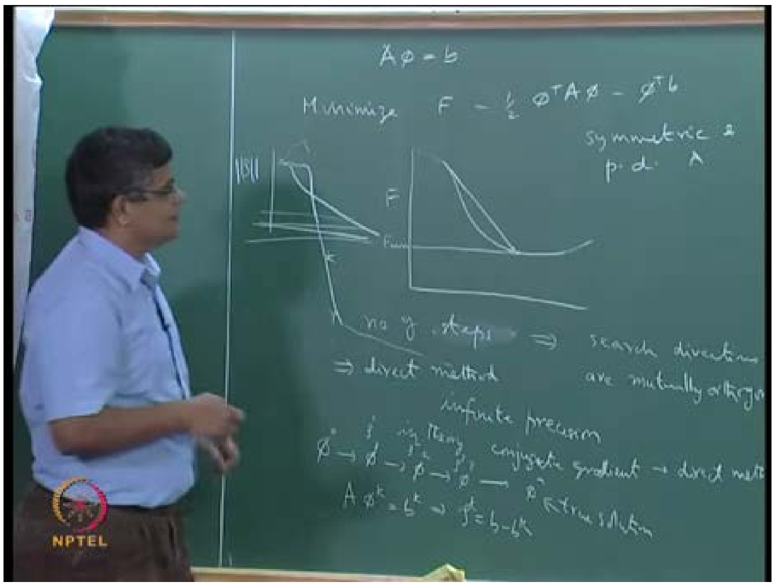


So, in this method, we are enabling a stronger linkage between M phi to with this and this; by making M to much closely resemble the value of the A, the contribution of A, because we are making N to be the contribution of N phi k to be almost equal to 0. So, and the resulting equation is also posed in terms of the resulting… So, this this thing can be written now as M phi equal to b prime, this is now being solved using a direct method, using the Lu method; and L u also have only sparse matrices, so the forward substitution backward substitution is will will not involve much computational time.

So, it goes from k to k plus 1 will involve only forward and backward substitutions of two upper or lower triangular matrices involving a few diagonals. So, that computational effort is less; and the overall solution, the iterative solution going from k to k plus 1 to k plus 2 k plus 3 is rendered faster, the convergence is faster, because we are making it more implicit by making the contribution of N phi equal to 0. We cannot look at all the details of how to evaluate this, those are available in in books and research papers, but this method has proved to be a very good method in its own right for the solution especially, of the Laplace or Poisson type equations, where we can expect a smooth solution. And especially this is useful in solving the pressure correction equation, which is the Poisson type equation.

So, this is one example of a fairly complicated thinking that goes into improving the rate of convergence of an iterative method by bringing in more implicitness, and bringing in a direct method in the as part of the overall iteration. So, this this solution of going from k to k plus 1 is direct, but by having to go to k to k plus 1, and then k plus 2, k plus 3 like that, you are enforcing iteration. So, this is one successful example of a an advance method; we look at two new methods, two new ideas for the same solution of A phi equal to b, these are totally different from the approaches that we have looked at so far. These are also iterative methods, and one of them, the first of these is the conjugate gradient method; conjugate gradient method is it has an entirely different philosophy for the solution of A phi equal to b.

(Refer Slide Time: 23:21)


$A\phi = b$
Minimize $F = \frac{1}{2}\phi^T A\phi - \phi^T b$
symmetric & p.d. A
F
no of steps ⇒ search directions are mutually orthogonal
⇒ direct method
infinite precision
conjugate gradient → direct method
true solution
$A\phi^k = b^k \Rightarrow r = b - b^k$

The problem is not forced as A phi equal to b, but a minimization of a function F in order to solve A phi equal to b, we would like to minimize function F, which is created from this is defined as half phi transpose A phi minus phi transpose b. So, this is a function, which is created and one can show that the point at which F is minimum corresponds to the solution A phi equal to b. And so the problem is now transformed into solution of search for the minimum of this function F. So, we are not solving any any methods like the simultaneous equations, but you may have a F here, and you want to find that point of minimum, at which this is.

Now, what is the advantage that we gain by this; what is a what is the difference between… What is the character of this particular thing? Now, when we want to find the minimum of a function of a function like this, we normally make use of a method, which is known as method of steepest descent, that is this F is a multi dimensional surface, because you have n number of equations. So, the function F here is an n dimensional surface, and we want to find out that in that n dimensional surface, what is the minimum value of F? It can be shown that when you have the matrix A as a symmetric positive definite matrix, then it is possible to come up with with a very efficient search algorithm, in which we look for the minimum of this F, in those in n projections of this n dimensional surface as it were.

Trying to minimize the value of F in each projection, and then if if we do so, it can be shown that at the end of n such projections, n such looking at trying to find the minimum value, we will get the exact solution. So, if you have, but the that guarantee of reaching the exact solution in n number of solutions is possible only if the search directions are mutually orthogonal. So, the that is the basic idea of this particular method, that is the problem of A phi equal to b is converted into finding the minimum of the function F, which is created from A phi and b in this particular way. And so we would like to use the for example, the application of the steepest descent method to find in each direction, what is the minimum value? For example, if you are here at this particular point, this is a slope, and slope is pointing in this direction.

So, you say that looking at this, this seems to be the minimum point. So, you can make use of the slope information at this location to see, what is likely to be the minimum value of this, at which F is minimum here. So, it is not always that you will get the minimum like a drawn here, but based on the slope, you know if you are trying to find out the minimum value of this, whether you want to go in this direction or in this direction. So, that depends on the slope value here, and that slope in an n dimensional surface, which gives you the minimum the steepest descent is the direction, in which you would go at that particular point of time.

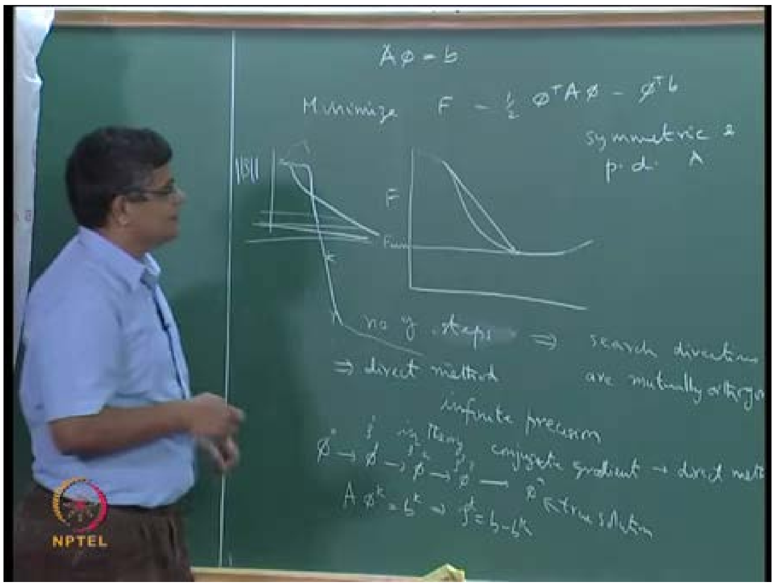
Now in that particular projection, now you choose another projection, which is orthogonal to it; and then you try to minimize in that projection where you should what you should should be on next step. And then you go to another direction, which is orthogonal to not only the previous direction that you have taken, but also to all the previous directions. So, the problem is is the minimization problem is reduced to one of constructing a series of search directions, which are mutually orthogonal; each new direction is perpendicular to every other search direction. And if you go through like this, then you you will reach the minimum point in exactly n number of steps, n number of steps.

Each minimization in each direction has a certain number of mathematical operations. And you know that you you will get this to the solution at the end of n steps; so, that makes this a direct method; in the sense that the solution is found out at the end of n number of n at an finitely calculable numbers of mathematical operations, because we know that in each step we know, how many mathematical operations are needed to go to the next step; and we also know how many steps are involved in this. Now this is a case, only when we have infinite machine accuracy, infinite precision of mathematical operations. So, because infinite precision is not really there, in theory the conjugate gradient method is an is a direct method.

And this is how it is proposed in 1952 by Hastens and Magnus and Stifle, they proposed this in 1952, and it created ripple at that time in the mathematics community, but the ripples soon died down, because each step will have large number of mathematical operations, we have to do so many steps; and this method is applicable only for symmetric and positive definite positive definite matrix A. It was a revived, not as a direct method, but as an iterative method in 1970 by Reed, who who suggested that this method can be used to find an approximate solution for A phi equal to b, because this has the characteristic of of of reducing the residual of this iterative solution of a A phi equal to b very drastically, very rapidly.

So, if it is in the right ballpark area, then it can reduce the error by even an order of magnitude in a single step. So, the what we are looking at is that we have phi 0, that is initial guess, and we get phi 1, and then phi 2, and phi 3 and so on up to phi n; and phi n is the true solution. This is and so each of the solution is obtained by going through a search direction to minimize the value of F in that particular direction, and that step procedure is known to us; although we will not discuss it. So, we are generating a series of solutions, and what makes it a good iterative method is that at each point we can calculate the residual. So, if you substitute the value of phi, then A phi for example, k should be ideally equal to b, if it is the exact solution, but this is only going to be equal to b k. So, the difference between b and b k is called the residual. So, you can compute the residual at each each step rho 1, rho 2 and rho 3 like this.

(Refer Slide Time: 23:21)



$A\phi = b$
Minimize $F = \frac{1}{2}\phi^T A \phi - \phi^T b$
symmetric &
p.d. A
$\|\rho\|$
F
no of steps ⇒ search directions
⇒ direct method
are mutually orthogonal
infinite precision
conjugate gradient → direct method
$A\phi^k = b^k \Rightarrow \rho = b - b^k$
true solution
NPTEL

So now, you can put that as the norm of this residual, like we do for any iterative scheme verses k; in the case of usual iterative methods, then it goes to something, and then it reaches a study value on a log plot, and that gives you the rate of conversions. But in a case of gauss seidel in a case of conjugate gradient method, it may be that the residual changes like this, suddenly it comes comes drastically reduces, and then it goes through the bottom in a way.

So, it is this attraction instead of going through large number of iterations to reduce the residual slowly from this level to this level to this level like this; in a few iterations, you can reduce the errors by large amount. And then may be it may take further number of reductions like this; and it is this characteristic coupled with the fact that in a typical CFD solution, we do not want to we do not need to reduce the residual to 0. So, why is that the residual is sufficiently small? It is the solution is good enough.

So, it is this nature, this requirement in a CFD that we can deal with an approximate solution not necessarily the exact solution is good enough for us, enables this conjugate gradient method used as an incomplete iterative method. So, its incomplete in the sense that we would not be taking all the n steps, we will be taking only a small number of steps, and hopefully within those large small number of steps, we the we come to the point of drastic reduction in the in the residual. And so, in that sense, it was revived, and there been lot of studies and lot of improvements of this, which have improved the method in various aspects; for example the general conjugate gradient method, you can have an residual, which may be even increasing, before it starts decreasing. At some point, it has to decrease, and it may be increasing and so there have been some modifications to make sure that in each step, it is decreasing.

And this condition of symmetric and positive definite A is severely restricting case for CFD solution, that is possible only when you have Laplace equation on a uniform grid with constant coefficients. But when you have for example, convective diffusion equation typically which is what we have in CFD type of things, and since we want to use some sort of upwind differencing for the advection term or the convection term, in terms of stability and all that. So, in such a case we will not have a symmetric matrix. So, they have been variants of this called bi-conjugate gradient method gradient method, and then further stabilizations, and squared methods, and all these things proposed in 80’s and 90’s, which have made it really an attractive method for the CFD type of applications; attractive, because it can it has the potential to reduce the error very drastically, very fast.

And secondly, because it is not necessary to have the Scarborough type of condition in order to apply this method. So, it is not necessary to have diagonal dominance in A phi equal to b, in order to apply the conjugate gradient method. So, it can be used even when the diagonal dominance, its condition is not there. So, if you are looking at a solution in a non orthogonal grid, in which case you may not have diagonal dominance; in such cases or when you have severe difficulties arising from source terms, in such cases this conjugate gradient method is very useful. And some of the properties of this residual reduction and all those things can be improved by preconditioning.

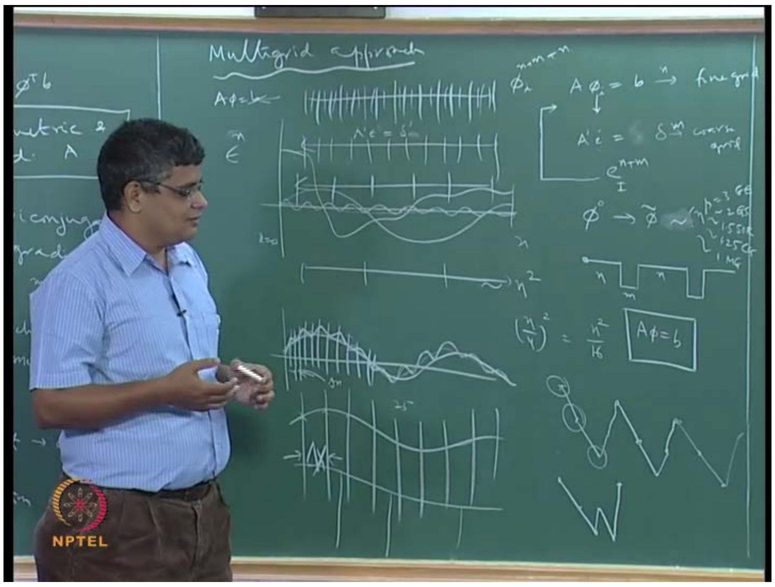
So, instead of the as usual the convergence of this method the residual reduction of this method depends on the Eigen values of A here and if the Eigen values are spread out over a large interval. So, that is if the ratio of the maximum Eigen value to the minimum Eigen value is large, then it can show erratic behavior, and it can it may not be very useful. So, preconditioning is done to reduce that ratio of the maximum to minimum Eigen value and that is known as the condition number. So, and one of those preconditioning methods is essentially, you have C inverse is the is multiplied one can multiply by preconditioning matrix, and to reduce the spread of these Eigen values, and one such preconditioning method is something that we have already seen, which is incomplete Lu decomposition of of the matrix A.

So, those kind of methods are brought into play here to to manipulate the Eigen values of the matrix A, so that the convergence of this conjugate gradient method is improved. So, they have been lot of implements of this method, and this method is also used for example, in the nuclear industry, when we are dealing with three-dimensional solutions of the thermal hydraulics of a nuclear reactor. There you typically do not have large number of points, but because of the constitutive equations, and additional equations that come into picture; and the source terms, you almost never have the condition of… It is almost impossible to have diagonal dominance.

So, in such a case, the conjugate gradient method has been used very successfully, where the number of iterations is usually of the order of 10 or 20 or even 5 instead of the 100 or 1000 or 10000 that we normally used with Gauss-Seidel method. So, the number of iterations is reduced, but we must note that in each iteration, what we mean by iteration, is that successive minimization of this function here; is the number of these things is reduced, but each computation, each reduction here will involve matrix computations and vector matrix multiplications like that.

So, the each step will have large number of multiplications, but overall when you compare overall bi-conjugate gradient method applied for a convection diffusion equation, with the corresponding thing from a simple solver will be much better with the bi conjugate gradient method, but it requires lot of prior preparation and intelligent programming in order to get the best out of this. So, this is one approach, where we have converted the solution of A phi b is equal to b into a minimization of a function.

(Refer Slide Time: 41:27)



Another very different approach is is the multigrid approach; this method depends on an entirely different philosophy; and the philosophy is that there are some methods we have seen the Jacobi method, and we have seen the Gauss-Seidel method, and SOR method, strongly implicit method. So, these all these methods have different ways in which they are trying to reduce a residual; these are all iterative methods, and in each iteration, you have a residual, and we want to the objective of these methods is reduce that residual to a reasonably small value. And what you mean by residual - is the degree of dissatisfaction of the current estimate of phi, when applied in to that A phi equal to b.

So, residual reflex by how much the current estimate of phi fails to satisfy A phi equal to b, the original equation that we try to solve. So, the way that the residual is reduced, and the error is reduced depends on the specific methods. And there are some methods which damp out high frequency errors, and some damp out low frequency error first. So, if you say that we have already come across the idea of the spatial variation of error in looking at the Von Neumann stability analysis; and we have a variation of error as a function of x, which is the space dimension at a particular time say n is like this.

So, this this varies from x equal to 0 to x equal to l in this particular, and this can be decomposed into as we have said earlier into low frequency variation like this, and high frequency variations like this. So, the overall contribution of the low frequency, and the high frequency terms will depend on the particular shape of the error function at that particular time. So now, if you are looking at an iterative method; and the idea of the iterative method is to reduce this error and bring it down to very small value. And in the process it has to damp out the high frequency error, and it has to damp out the low frequency error.

Jacobi method and Gauss-Seidel method are two contrasted methods, which damp out the high frequency errors in a different way. Jacobi method is not very good at damping out the high frequency areas whereas, Gauss-Seidel method is a good smoother, so that means, that it will damp out these things, and if you consider if we take just these two errors, and then come up with something like this. If you add these two, it may be that we have a variation like this. And we can see a low frequency component and high frequency component; so if you start with this as the initial guess, and then you apply on this the the Jacobi method and Gauss-Seidel method.

Gauss-Seidel method will soon give you a variation like this, in which this high frequency component is more or less damped out, whereas the Jacobi method may still retain a significant amount of this high frequency error. So, it is this property that there are some iterative methods, which will damp out the high frequency error very quickly, and convert this frequency distribution, which has the thing into something like this enables us to replace that high frequency curve, which which is something like this with a low frequency approximation. Now what is the difference between this one here, and this one here? If you want to represent this accurately, then you need to know the function at every small delta x, only then you can show that is increasing like this, decreasing like this, increasing like this. But if you have this kind of curve, a smooth kind of curve, and if you want to represent it, it is sufficient to have at this delta axis, so that I know this value, and I know this value, I can pass a smooth curve through this.

So, if this is my function, which is varying rapidly, then if I want to represent it correctly, I need to have a delta x, which is very small. And in this, I can have much larger delta x. So, this enables us to solve for the error equation now which is here, with a larger grid spacing than in this particular case. And this is the idea of the multi grid approach that is you are trying to reduce the error. So, you do some computations with with for example, a smoothing kind of iterative method like Gauss-Seidel method, and reduce this this error distribution to something like this; now you have got a smooth error distribution, try to reduce the error in this, using a bigger grid.

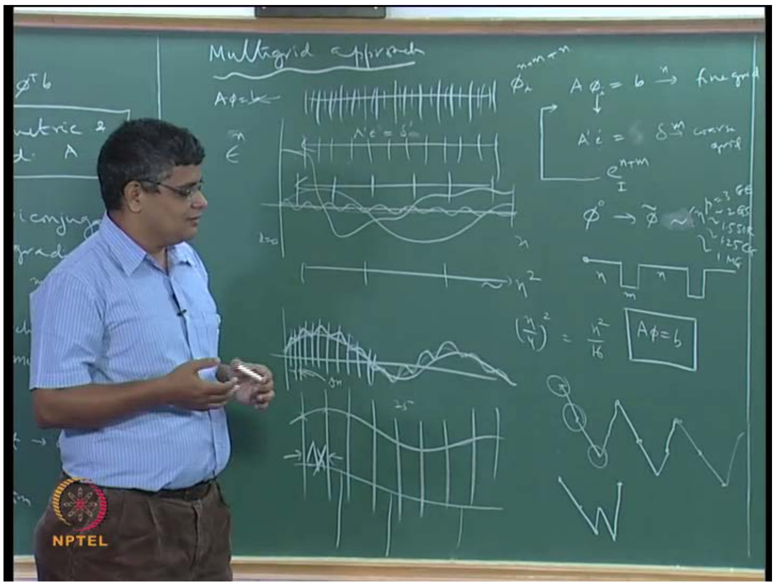
And then this error will be reduced to something like this; now you can use much bigger delta x here, because this become even smoother function, so that means that even larger delta x can be used. So, in that sense and we know that typically the computational time, number of mathematical operations varies as n square for Gauss-Seidel method or for a typical method that we are looking at. So, where n is the number of points, and here we have we require may be 100 points to represent something like this whereas, here we may require only 25 points to represent this to the same accuracy, because it is now smooth.

So, here to do a computation on this on this on this function with a small delta x will require n square number, and here we need only n by 4 square numbers, so that is n square by 16 number of operations to reduce the error in this. So, in order to the computational time required to work on this fine grid is much more than the computational error required to do the on to work on this course grid. And not only that, we know that for example, for a typical Laplace type of equation, the larger the number of points the larger will be the the closer will be the largest Eigen value to 1. So, that mean this is spectral radius becomes even close to 1. So, that means, that the rate of convergence rate of reduction of error in with a fine grid is less than with a coarse grid.

So, when you go from fine grid to coarse grid, you have two advantages; one is that the actual number of computations that are required will be reduced; and the rate of error reduction is also residual reduction is also increased. So, because of this, it is preferable to work with the coarse grid wherever possible. But if you have a coarse grid like this, then we cannot represent a fine variation, and we know that using the coarse grid will be will give us more inaccuracy.

So, the in a multi grid what we do is that we solve A phi equal to b on a fine grid; for n number of steps using a smoothing iterative method like a Gauss-Seidel method. And from this, we can evolve the error equation some A prime e equal to some b prime, and this error equation in the form is in the form of residual, so residual. And then this is solved on a coarse grid, because now as a result of n number of smoothing operation on this the error is reduced. So, that we can solve for this on this coarse grid and then we can get after another m number of operations on this we get error at for example, if you say that if you say that this is small i represents the fine grid and capital I represents the course grid. So, we we get the value at a error at n plus m.

(Refer Slide Time: 41:27)



So, using this error, which is now reduced as compared with error at n, we make use of this estimate of error to guess the new value of phi I at n plus one n plus m. So, and we now work with this fine grid here, for another n number of equation steps iterations. So, now, we have got further reduction of the fine scale error in this, with this thing we come back on to the coarse grid, and then we have we try to reduce the error in this by doing further number of m operations on this.

So, in that sense what we are doing is, we are going from a a fine grid solution for n number of steps here, and then we go for a coarse grid number of coarse grid for m number of steps, and then we go to fine grid, and then we do like this, n number, m number like that, we go through these kind of steps. And the advantage in the process is that these m steps on the coarse grid will take a small time, but they are reducing the error involved at these points first. So now, when you put this back into the coarse grid this coarse grid will now try to reduce the error in between these things on this, and then it can continue to smooth out, and then get to a faster rate of, overall rate of error reduction.

So, in a typical multi grid approach you have for example, for the same domain here, you may have a very fine grid like this, may be even smaller. And you may have another grid, which is twice the grid size; and another grid, which is twice of this, another which is twice of this and so on. And then you you solve for phi on this A phi equal to b, on this and in each of these things you solve only for error in the form of residual. So, you solve it on this, and then you go to this, and then you go to this, now you have got the error estimation on this. So from this you go back here, and then you may from here you may go back to this, and then you may go back to this like this. So, you go you solve for sometime on this, and you solve for some more time on this, some more time on this, some more time on this. So, you reduce some error here, more error here, more error here, more error here; and then, but in the process since you are coarse grid, some of the error in between is not estimated, then you can go back in this way.

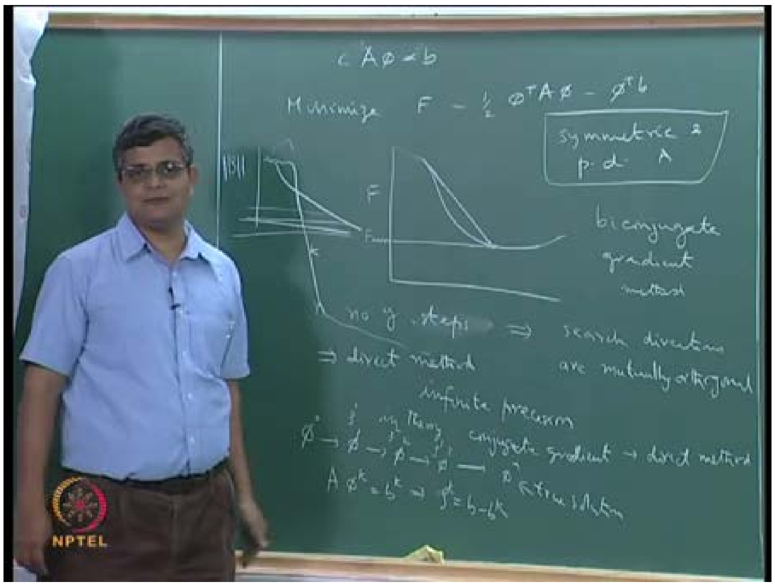
So, you can go from fine grid to coarser grid to coarser grid, and then back to this and then back to this; so you can go through in these steps or you can start with this, we can go to this step, you can go to this step, come back to the coarser grid, go back to the fine grid, and then you can do like this. So, you can patterns in in different ways, the way that you see if it that so that you can have for example, a W type pattern or V type of pattern, which is repeating like that. So, if you look at the actual computation that is done, there is some computation, which is done on the finest grid, more computation on the coarser grid, more on even the coarser grid coarser speed like that.

And we know that the number of iterations on the coarsest grid required for certain error reduction will be a small fraction of what is required for on the finest grid, so that the overall residual reduction is now made on coarser grids more of it is now made on coarser grid and only we depend only on a fraction of the total error reduction to be done on the finest grid. Therefore, the total number of mathematical computation that are required to reduce the reduction to reduce the residual by a certain factor is only partly reflective of the effort needed on the finest grid, most of it will be done here therefore, the overall mathematical operations required will be significantly less in the multi grid approach.

If one were to express phi 0 to phi tilde the effort required from an initial guess to close to the exact solution in terms of n rise for p as n rise for p, we know that p is equal to 3 for Gaussian elimination, and is 2 for Gauss-Seidel method, and 1.5 for SOR optimal, it is 1.25 for conjugate gradient, and it is almost 1 for multigrid method. So, in that sense as so you go to more and more sophisticated methods involving newer and fresher ideas, involving not the simple way of doing it, but more complicated way like this or in the conjugate gradient method, you can gain in terms of the efficiency by which you solve your A phi equal to b.

So, the solution of A phi equal to b, which is necessary in the case of CFD solution, it is necessary, because we have do it for so many times, and because we have to do it for so, many equations. So, the the efficiency of the solution of A phi equal to b can be increased greatly if we go for more and more methods, there are some methods which are most suitable for certain cases than some other methods. There is always the most sophistication that you try to bring into your solution method the more efficient will be the method. But there is a lot of detail that has to be developed in trying to fully understand and implement methods like conjugate gradient and a strongly implicit method and multi grid method and all that.

(Refer Slide Time: 1:00:00)


Minimize F = ½ φᵀAφ − φᵀb
symmetric & p.d. A
biconjugate gradient method
no of steps ⇒ search directions are mutually orthogonal
⇒ direct method
infinite precision
conjugate gradient → direct method

So, if you want to drive the benefit from this, it is necessary to do lot more work than what can we done in a course like this; one has to take specialized courses or do lot of reading of advanced text books, and go through the literature to not only implement these things, but also to fully understand the theory behind all this; for example the the effect of the spectrum of Eigen values on the efficiency of of the conjugate gradient method, and how by doing certain manipulations one can ensure that in each step of this, the function f is minimized.

So, the the thought process behind the development of this methods must be really understood and appreciated, before you can think of implementation. As far as the initial trials for cfd is concerned, it is better to stick to simple methods like the Gauss-Seidel method, which is good enough for many applications. So, in the next lecture we will now try to look at the complete template for the solution of the Navier Stokes equations that we have so far seen.