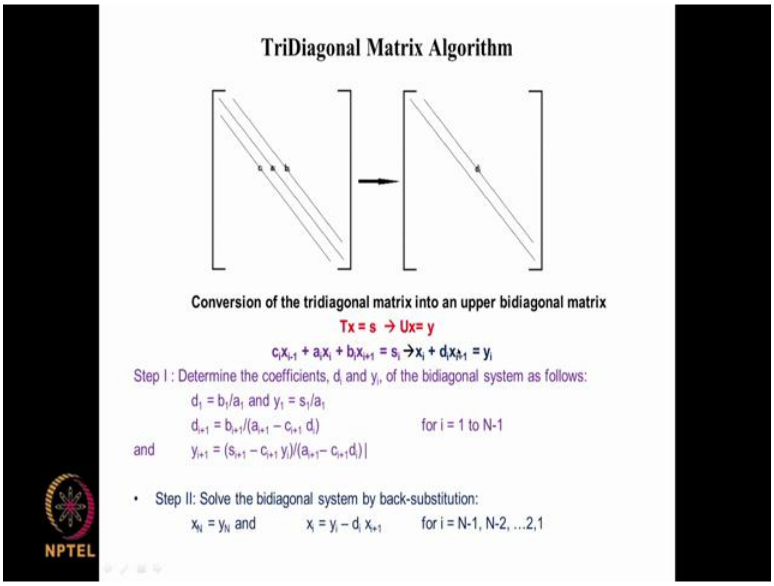
Computational Fluid Dynamics
Prof. Sreenivas Jayanti
Department of Computer Science and Engineering
Indian Institute of Technology, Roorkee

Lecture - 47
TDMA and other iterative methods

In the earlier lecture, on the board we tried to derive the Thomas algorithm for tridiagonal matrix it is also known as TDMA the TriDiagonal Matrix Algorithm. The idea is that we have system Tx is equal to s where matrix T is the coefficient matrix and this is the tridiagonal structure along the main diagonal you have coefficients a i.

(Refer Slide Time: 00:27)



And the super diagonal you have bi super diagonal, below this we have ci here and all the other coefficients as 0. So, that if you take any equation, any row here and then multiply by x to give you this you have an equation like ci xi minus 1 plus ai xi plus bi xi plus 1 equal to si. So, this is the equation that we get from for example, the one dimensional laplace equation or a precise equation. Discretize using central differences we get an equation of this particular form. What we do in the TDMA is to convert this tridiagonal matrix into an upper diagonal matrix, upper bidiagonal matrix in which you have only 2 non-zero diagonals and all the others has 0. And outer this non-zero diagonals the main diagonal is taken to have values of 1, every element on this has value of 1 and only di are non-zero and these are you have to be determined.

So, the transformation of Tx equal to s to Ux equal to y enables us to have an equation involving a coefficient matrix like this which can be done through success of substitution. And it is also a specialized substitution and which if you take any equation here it is of the form x i plus d i x i plus 1 equal to y i. So, there are only 2 variables. So, the ideas would be that you start here you get x n and from this you go to the next row here you have x n and x n minus 1, you already know x n you get this.

And then you go to the next row which involves x n minus 1 and x n minus 2 out of which you know x n minus 1. So, you can get x n minus 2 and so on. The back substitution always is an equation involving only 2 variables. So, the number of arithmetic operations immediate to get the solution from u x equal to y is very limited its one multiplication and essentially one multiplication and subtraction. So, you know x i plus 1. So, multiply that by d i and then y minus d i times x i plus 1 will give you x i. So, the evaluation of the solution after it is converted to u x equal to y is pretty straight forward.

Now, we have also derived steps by which we can obtain the values of d i and y i from known Tx equal to s, that is known values of a i, b i, c i and s i. And these are in to begin with we evaluate d 1 equal to d 1 by a 1 and y 1 equal to s 1 by a 1, and from then onwards we put d i plus 1 equal to b i plus 1 divide by a i plus 1 minus c i plus 1 d i. If you look at this for i equal to 1 you get d 2 is equal to b 2 which is known divided by a 2 plus c 2 times d 1 where d 1 is already evaluated here.

Similarly y i plus 1 which is coming on the right hand side of this transformed equation is s i plus 1 all the coefficients are known here minus c i plus 1 which is also known y i you have just calculated and then you have this which are known. So, again from known values of y 1 which is s 1 by a 1 and d 1 you can calculate d 2 here and y 2 here and then move on to d 3, y 3, d 4, y 4 all the way up to d n y n. In this way we can evaluate all d i y i and then you go through the bidiagonal system, solution by back substitution x n equal to y n and x i equal to y i minus d i times x i plus 1.

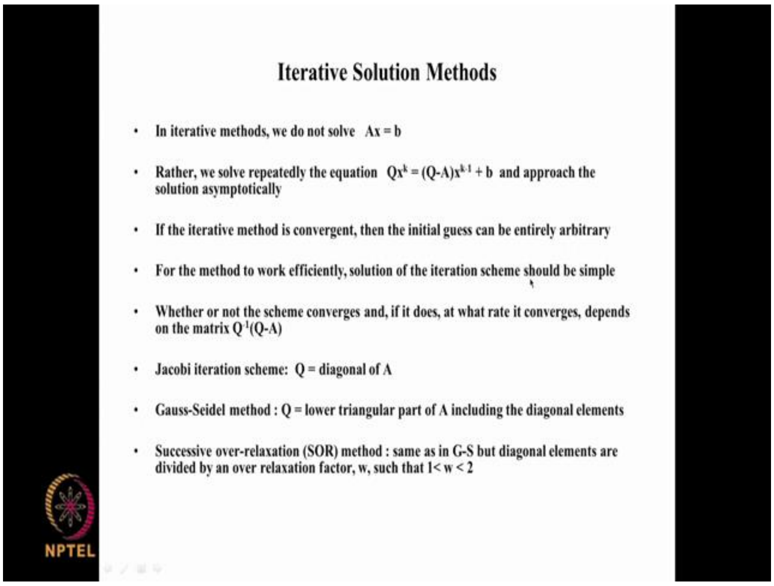
Evaluation of this involves one multiplication per variable. Here you have more multiplications you have to multiply this by this and then add this and then divide here. So, you have one multiplication, one division, again one multiplication, second multiplication is already done in a way, and then division here. So, in that way you can compute the number of arithmetic operations which are required for this.

This will be proportional to the number of unknowns. It is not something that to goes as n square or n cubed it is proportion to the number of unknowns, so that even if you have very large system of equations you can solve it very efficiently. So, this algorithm known as TriDiagonal Matrix algorithm or TDMA or Thomas algorithm is are very efficient way have solving Tx equal to s. Much more efficient then either Gauss-Seidel Gaussian elimination or LUD composition, in fact one could call this as special from of Gaussian elimination.

But the catch is that this is the applicable only when you have 3 adjacent rows and that to where diagonal dominance is guaranteed and this happens only in 1-D flows. If you have 2-D and 3-D then you do not have 3 point molecule we have 5 point molecular or 7 point molecule or 19 point molecule, in such cases this cannot be applied. But there are extensions of this method in more advanced methods for solution of Tx equal to or x equal to b which will see in the second part of this module.

So, far we have seen in detail 4 methods including the Crammers rule, but the 3 realistic methods that can be used for CFD type of solution is the Gaussian elimination method, LUD composition method and the TriDiagonal Matrix algorithm these have their applicability. But let us look at a couple of other basic methods which belong to the second category which is the iterative solution method not direct solution methods.

(Refer Slide Time: 07:19)



In iterative methods we do not solve Ax equal to b we solved different form of equation. For example, we solve repeatedly the equation Qx k equal to Q minus a times k x minus 1 plus b and approach the solution asymptotically. That means that instead of solving this equation we solve this equation here. What is the difference between this and this? This looks even more complicated than this, right. But the Q here is such that this equation becomes very easy to solve and you can see that when the solution is converged. So, that is when x k is almost equal to x k minus 1 or when its equal to x k minus 1 then Qx k equal to Qx k minus 1 minus Ax k minus 1 plus b. So, Qx k minus one and Qx k will cancel out and this minus comes here we get back this equation.

So, even though you are solving this equation which looks different from this equation converge solution is the same as this equation. So, in that sense we are going on a different way of solving this equation on in indirect way of solving this equation and that too we have solving this equation iteratively many many times. And despite this apparently roundabout we have doing this, this actually turns out to be more efficient then Gaussian elimination method or led composition method for a classic problems and this is very widely used for a number of reasons in CFD type of calculations.

We have already seen one such iterative method the Gauss-Seidel method, in the very first module and very first of part of it for the solution of flow through rectangular duct we made use of Gauss-Seidel method and that belongs to the class of iterative solution methods. If the iterative solution method is convergent then the initial guess can be entirely arbitrary, you can take any set of values for the initial guess and then have guaranteed converge solution finally. For the method to work efficiently, solution of the iteration scheme should be should be simple and whether or not the scheme converges and if it does converge at what rate it converges depends on the matrix Q inverse of Q minus A.

We will see this thing later on is a very possible that you have done the 2 basic iterative methods which are known as the Jacobi iteration scheme and the Gauss-Seidel iterative method. In seeking an iterative solution like this obviously, we want this to be convergent that is we have the first condition that this method should satisfy for it to work.

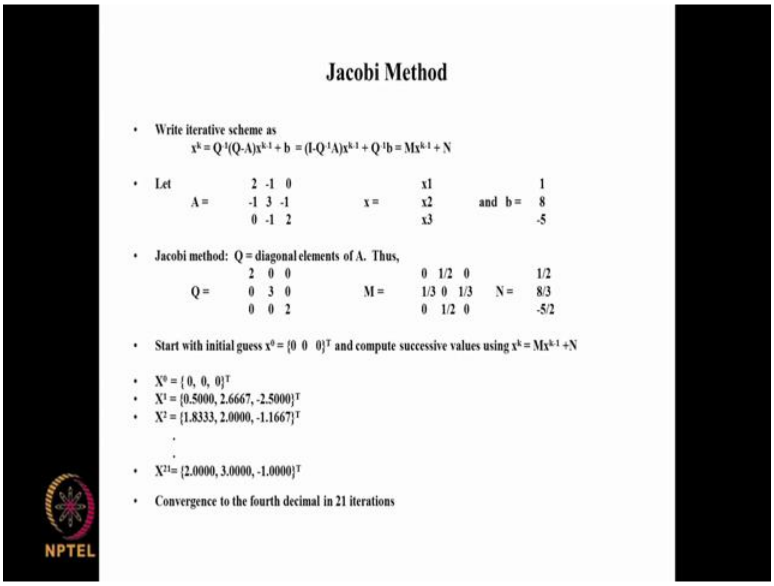
Secondly, it should be easy to solve because we are solving it many times and thirdly it should converge fast even though it is easy to solve if it takes a huge number of iterations to converge then the overall computational time becomes very large. Finally, finding cube it should not be expensive from a computational point, just like when you look at lu decomposition method once you resolve a into the product of l and u then the solution is pretty easy. But the decomposition of a and 2, l and u itself is computational expensive. So, in that sense although the solution of l a equal to b is easy, converting a into l u plus a into l u is so expensive that overall scheme is more expensive or more comprehensive intensive then the Gaussian elimination method.

So, in the same way here even if the solution of this is easy and even if it converges fast, if the decomposition of if finding Q which is not there in the original equation. If if we can find such a Q that it makes the solution converge and converge fast, but if in this is the process of finding Q itself is time consuming then the overall solution method is not good.

So, for an iterative solution method to be effective from a computational point of view finding Q should be easy, solving this equation, and how many such equations? There are n such equations, where n is the number of variables. Solving this, n times should be easy and solving the set of these n equations iteratively, the number of iteration steps should also be small or rate of convergence should be fast. So, if you have all these things then we can have an effective way of computing Ax equal to b even though it is going to approach the solution asymptotically is still have an effective method and that can be used.

So, what we are going to see is the 2 basic methods the Jacobi iteration method and the Gauss-Seidel method, and we see how in these 2 basic methods this idea of solving Ax equal to b in an iterative way by finding a cube which makes as iteration scheme convergent and all that works out.

(Refer Slide Time: 13:50)



So, in the Jacobi method we take Q to be the diagonal of a. So, what we mean by that? Is that lets say the matrix then we take Q to be equal to all this a i. And all the other elements of Q are 0. So, Q is a matrix and it has as many elements as A, but out of that only the diagonal elements are non-zero and all the other elements are 0 and not only that all the diagonal elements of Q are equal to the diagonal elements of A. So, in that sense there is now nothing much is (Refer Time: 14:35) find Q, it is just select the diagonal elements of A and put them into matrix q. So, in that sense the first idea that it should be easy to compute Q or it is easy to put together or construct Q is a conditional which is satisfied in the Jacobi iteration method.

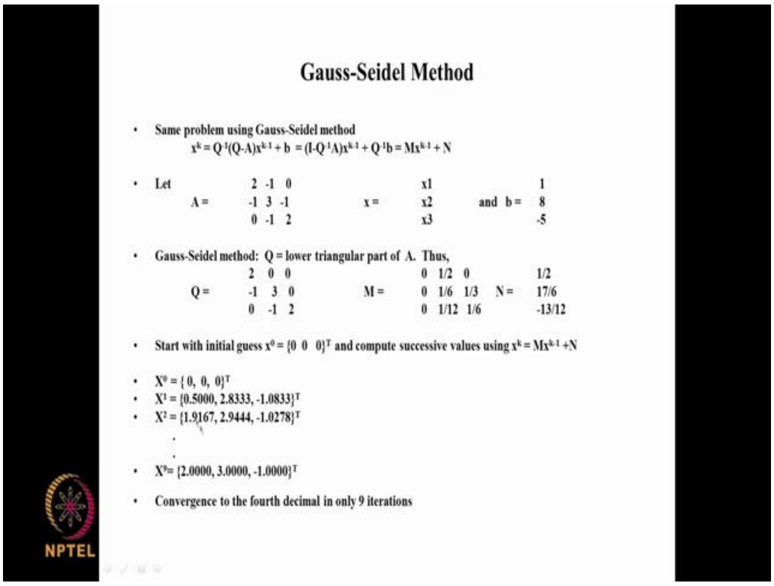
We can show that the Jacobi iteration scheme when constructed this way. So, that is taken Q to be the diagonal elements and all the rest that solution method using this particular recursive formula is convergent when we have the week form of diagonal dominance of the strong form of diagonal dominance is satisfied. Then we know that the Jacobi iteration method will work.

So, if that is there then we have a convergence scheme and we can make it work. So, exactly how it works is something that we can see for scheme like this. So, we are saying that x k is equal to Q inverse of Q minus a times x k minus 1 plus b or i where i is the identity matrix minus Q inverse of a times x k minus 1 plus Q inverse of b and we can also put this has m x k minus 1 plus n where m is this whole thing and n is Q inverse b. All though it is written like this, in actual case we do not really find the Q inverse and then find all this things we will see how it will be done.

So, if for example, a is given by this three things 2 minus 1 0, minus 1 3 minus 1, 0 minus 1 2, and x is x 1, x 2, x 3 and b is 1 8 minus 5. Then Q is the 3 diagonal elements so, that is 2 3 2 and all the others are 0 and now you can take Q inverse and multiply by i and subtract from identity matrix and then you can show that m is given by this 1 by 2, 1 by 3, 1 by 3, 1 by 2 and n is this. So, now, you know m here and n here we can start with some initial guess for example, 0 0 0 for the 3 variables and then substitute that here and then get x k that is x 1 and then you put the x 1 back here and then get x 2 and then put it back here get x 3, that is what you have. So, x 0 where the superscript indicates the initial guess, is 0 0 0 for the three elements; x 1 is 0.5, 2.667, minus 2.5.

We can substitute these things here and then we can get x 1, x 2, x 3 like that, and x 21 gives us 2.000, 3.000 and minus 1.000 and you can see that this is solution. Example, 2 times 2 is 4, minus 3 and that is equal to 1 and minus 2 plus 9 minus of minus 1, that is plus 8, and this is minus 3 minus 2. So, that is minus 5. So, it is a solution to this equation and we have got it in this particular way. So, by doing a solution iteratively we have gotten here.

(Refer Slide Time: 18:27)



The Gauss-Seidel method which we have seen earlier as a formula is also written in the same way x k equal to Q inverse of Q minus a times x k minus 1 plus b. So, it is exactly the same generic formula but the choice of Q is different, for the same set of equations here the same equations now Q is taken as all the elements which are on this main diagonal and below that. So, that is this one 2 3 2 minus 1 minus 1 and all this upper diagonal elements are 0 and then this is 0 because this is 0 here and we can substitute in this and we can get m and n to be like this.

So, we can start with the same initial guess and then we can compute successive values of x k using m x k minus 1 plus. So, you start with this and then you get this and then you get this, and in this particular case we find a solution in only 9 iterations. This is in a way the characteristic, way in which we can do this. Will do further in the next lecture, but I would like to recap on the basic iterative methods, in the iterative methods we solve Ax equal to b in a quirky way, in a strange way. In what seems to be much more difficult and roundabout way; by this was solving Ax equal to b we solve it has Qx k plus Q minus a times x k minus 1 plus b.

And Q is taken in a certain way and in such a certain way that when you go through this particular recursive step you can generate a series of improved guesses of the actual solution and if you generate substantially large number of these then it is expected to go to towards the final solution. So, that nature by which improved solution leads eventually to the final solution is known as a convergent scheme. So, if you have convergent scheme then you can start with some arbitrary initial guesses for these and you can march it forward in a sequential way and get to the final solution.

So, that will make it attractive and although we are doing all this inverses and all that you will see that when we are actually solving a large number of equations we do not have to construct the inverse we can have an explicit way of calculating successive values of x k from x k minus 1.

And not only that we a fairly simple robust rule for checking whether for a given Ax equal to b, whether the Gauss-Seidel method or Jacobi method will work, in the sense that whether and not they lead to convergence solution. So, because of the fact because of these 2 facts which is that; instead of solving a matrix equation we can solve readily some simple substitution type of equations in a sequential way makes this method very attractive these iterative, methods very attractive.

The second thing about them is that you do not have to worry about the initial guess and how you start and all that, you can start from many things and then you will eventually get a solution. The third is that at any point you have an estimate of the solution and that means, that you may have an estimate of the solution after 10th iteration it may be accurate only the second decimal place. And if you want more accurate solution then we can go for some or number of iterations and you get a more accurate solution. So, you can improve the accuracy of the solution of Ax equal to b by taking more and more step towards final solution and why this is important?

We have seen that when we are solving navier stokes equations, we have lot of non-linear terms and coupled terms coming from other equations and at the time of solving. For example, a u star equal to b in the discretized x momentum equation, the u star is provisional variable at all the grid points, its provisional because in evaluating this a u star equal to b we have making guesses of what is p and what is v in a case of 2 dimensions. That means, that this a u star equal to b is not going to be the final solution only when we have good value of p and a good value of v is this going to be correct.

So, we make use of the iterative method to solve a u star equal to b for some number of iterations we get an improved guess and then we move we move on to solving a prime v star equal to b we get v velocity field. And then for after set number of iterations we go and solve the pressure correction equation and then we come back and then we do this. So, in a scheme of solution method where we are solving Ax equal to b many times, but each time using approximate variables having this possibility of getting an approximate solution makes it worth well to perceive this method.

So, if you are using a direct method, direct method guarantees you the exact solution of a u star equal to b after so many numbers of arithmetic operations. But we do not really need the exact solution right in the beginning, because a u star equal to b is made approximately by assumed values of v star and p star. So, that is why in such cases we do not need to solve a u star equal to b exactly, is sufficient if we have a good solution for that, not the exact solution. So, in that sense these iterative methods are useful in the context of solving coupled non-linear equations, so that is one of the advantages.

In the next class we are going to work these out for some practice problems so that we know exactly how to construct Q and how to solve this equation Qx k equal to Q minus x k minus 1 plus b. And we will also device a method by which we can test whether this method is going to work, and we are also going to device a method by which we can stop this iterative scheme; exactly after how long, how many times do you have iterate this is something that we can monitor the rate of convergence. We can monitor the rate of goodness using certain measures and we discuss those methods so that we have a complete grip on these basic iterative methods.

In the next module, in the second part of this module we look at more advance methods which work faster than this Gauss-Seidel and Jacobi methods. And the faster we get to the solution the better it is and therefore, they are superior methods, but we understand right now the basic methods and basic direct methods and iterative methods and finally, we will go into the more advance methods.