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

Lecture – 53
Line Relaxation Method

Welcome. So, this particular lecture, we will work on Line and Block Relaxation schemes. These schemes are solved for gain solving matrix equations. However, instead of iterating for each individual component of the solution vector, we will take a block of solution vectors and we will try to iterate for them.

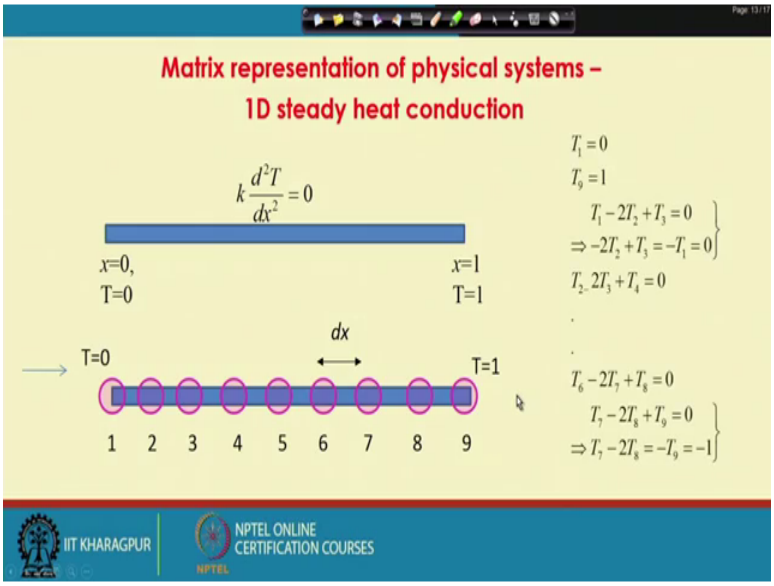
The advantage is this method in certain cases helps us to work on very large matrices, where we do not need to store the entire matrix rather we can store chunks of the matrix and do our work with that. And also there are some advantages in which we can break down the matrix into smaller sub matrices and solve each of the block independently, it can be ported into different computers and number of computers or number of computing course can perform in parallel.

We will see that application not exactly in this particular session; but in later sessions when we will discuss about parallelization and domain decomposition. And the third advantage comes in term of the fact that we will break the large matrix solver into smaller matrix problems into smaller blocks of matrices and in each block when we will solve, we can use something like a direct solvers which will be faster. We do not have to iterate within the small blocks.

However, you have to iterate for the different matrix blocks if the solution that is coming from the direct solver in one particular block, we have to check that whether it is a global solution. So, you have to iterate over different matrix blocks. However, we can use some of the direct solvers which can be useful especially if we can use something like ATDMA that will be useful, we will see these applications later.

So, we will look into line and block solver schemes instead of looking for each solution at each individual element of the solution vector, we will try to take a block of the solution vector and try to solve for it. And then, do it for the inter solution vector by dividing in into several blocks. I believe we have discussed something very similar to that when we have discussed about alternate direction implicit method.

(Refer Slide Time: 02:37)

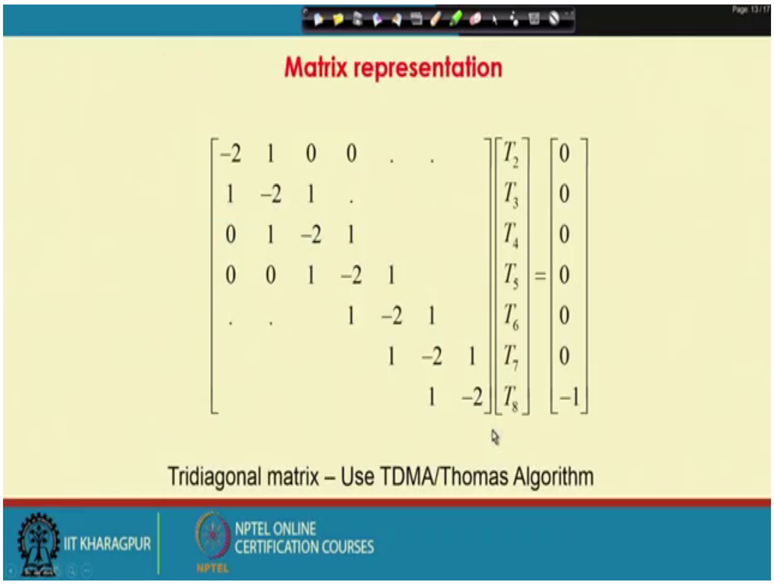


We will look into it. So, you go back to our older discussion on matrix representation of physical system, I have a one dimensional rod where I am solving conduction equation d 2 T d x square is equal to 0 with Dirichlet or essential boundary condition at both turns, that means, T is the solution variable which has to be found within the rod.

However, T at both the ends are specified as the boundary conditions. So, when we convert it into a matrix equation that use something like Taylor series expansion to convert d 2 T d x square differential term to a difference term, we get a set of matrix equation.

You can remember that these problems, we have this particular technique of analysis we have discussed long back, when we are discussing about how to convert a physical look into a physical system and get the matrix equation out of that and how what are the easy methods to solve that system of equations. So, we get a set of equations like that and for this particular case when you are using d 2 T d x square using a second order finite difference method.

(Refer Slide Time: 03:49)



We get a system matrix system of equation like this we call this as a Tridiagonal matrix. Because there is a main diagonal, there is a super diagonal and there is a sub diagonal in the matrix and remaining all the terms are 0 here in that particular matrix.

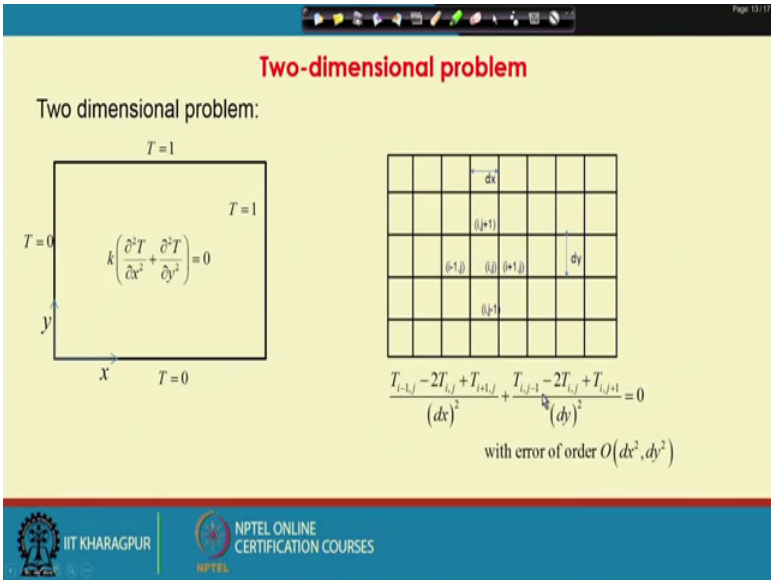
And there are algorithms like Tridiagonal matrix algorithm or Thomas Algorithm which can use as a recursive relation and can solve this equation if there are n such rows in this matrix, it can solve this system of equation in exactly n number of steps of order of n number of steps. And this Tridiagonal matrix algorithm is a very is important algorithm in a sense that not only for solving 1 d differential equations we can use TDMA, but we can use TDMA, we will see for ADI will which will see later.

But most importantly this TDMA type of algorithm we have used for some of the krylov subspace based solvers. By which we used Lenses Orthogonalization or lenses Biorthogonalization method and we got that inversion problem y m is equal to H m inverse beta e 1 that particular Heisenberg matrix inversion problem into a Tridiagonal matrix inversion problem for symmetric lenses algorithm also for lenses by orthogonalization.

However, when we obtained the Tridiagonal system of equation there which can be inverted using TDMA very efficiently and within very we with very less number of computational steps, we can get the solution that is why conjugate gradient or Biconjugate gradient has exploited the idea of TDMA.

They essentially obtained a tridiagonal matrix system and solved it and we got very fast solution. So, nevertheless so this is TDMA. We have discussed TDMA and few of its applications earlier. Now, the interesting thing which we are focusing more in today's class is for two-dimensional problem.

(Refer Slide Time: 05:52)



And also the same thing can be observed in three dimensional problem we will discuss it later. So, we have a similar problem, we have a rectangular plate all the boundaries are given with temperature constant boundary conditions.

So, value of the variable is known it at all the boundaries and we have to solve the partial differential equation which is Laplacian of T is equal to 0. When we try to convert it into the difference equation, we get a difference equation like that which contains different derivative in x direction as well as derivative in y direction, the difference expressions for that. This is a differential equation and this becomes a difference equation.

There is the journey from analytical to numerical one where from this equation can only be solved using certain analytical techniques like separation of variables. However, here we are getting a matrix equation and all the discussions on matrix solver can be of use for this type of equation. So, right now for the focus with the focus of matrix solution, so my idea is converting any differential equation into a difference equation like that which is one rho of the matrix equation.

So, we got an equation like that. If we try to put this equation in a matrix form, there will be 1 diagonal term and 4 of diagonal term which is a pentadiagonal matrix form. We can also see that this has 2 difference shell terms which are converted into difference terms; one is differencing in x direction i minus 1 i i plus 1 differencing in x direction. Another is differencing in y direction.

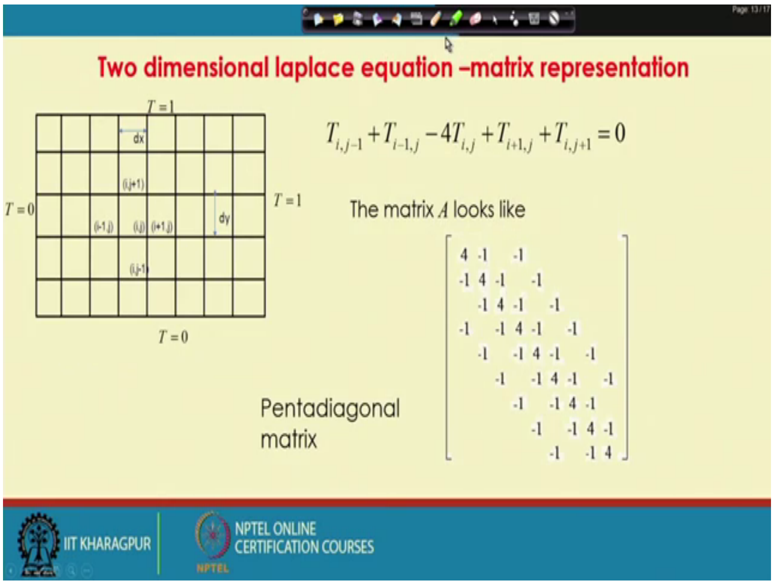
So, we will use an idea like that in case I use a guess solution in y direction, this is only a difference equation in x direction and this is a tridiagonal this will give a tridiagonal equation in x direction. So, if I have temperature known in this and this points within this particular line this is the tridiagonal matrix which can be directly solved using TDMA.

So, and this will be the idea of ADI will use some idea of iterative methods that will that is we will start with a guess solution and we will know assume that we know solutions here and here and we will find that tridiagonal matrix here; apply TDMA, get the solution here.

Now, this updated solution and the previous these solutions will be used in these 2 lines and we will try to find out the solution here. That is why we have finding solution along one particular line and we will call it as a line relaxation method. This will not converge to the right result because every time we are assuming going in one particular direction. So, solving in one particular direction and using guess value in other directions.

So, we have to also change the direction after one sweep of solutions, we will go as per this line and use the guess solutions in this lines and there will be a sweep in y direction. So, there will be a sweep in x direction and y direction and that is how the directions will be changed, we will do an alternating direction sweeping method and try to find out the solution. This is the basic philosophy of a d I method. Important is that in any step we assume that sorry we assume that we already know the solution here.

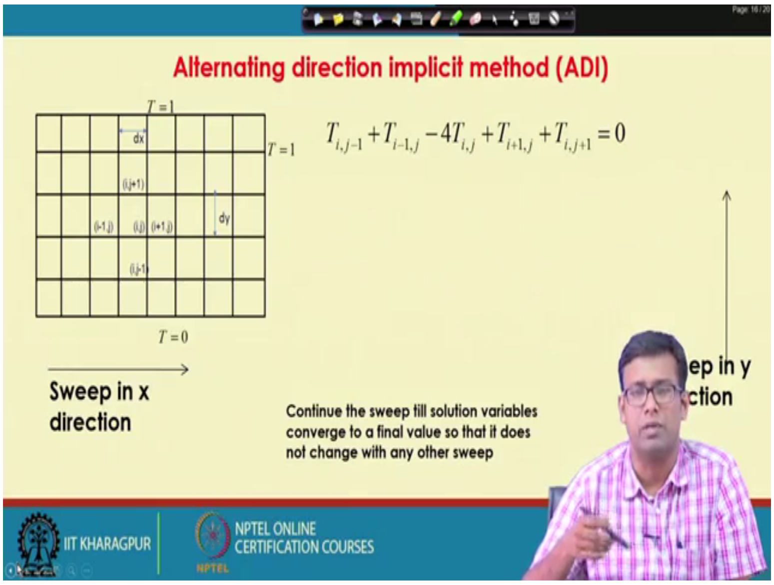
(Refer Slide Time: 09:22)



We already know the solution here which are the guess solutions and try to find out solutions here, say try to find out this particular equations to be solved here and that is why we called it line relaxation because we are relaxing; we are getting the solution, relaxing a line and getting the solution on that particular line. We are not considering going across the line in this steps.

So, let us go to the next slide that when we try to convert it into a matrix equation, it becomes a pentagonal matrix. Had it been a three dimensional problem, we would have got a pentadiagonal matrix here; we would have got a septa diagonal matrix there.

(Refer Slide Time: 10:10)



And the idea of ADI is that as we discussed assume a guess value here, put the boundary value here and solve for this particular line; solve for this particular line. For this line you solve the equation which will be a tridiagonal matrix equation here and you can use b star is the new guess value after substituting the values here and values here in the equation you get a new guess value b star solve it as a using a TDMA. And once you have solved for. So, for j is equal to 2, you solve all i i all a different I values using this TDMA. Once you have solved it, then use the updated value which is this last solved value here.

Assume a guess value here and solve for j is equal to 3 and so on you go for a sweep in x direction; that means, now once you this is done now use the updated value, we use the solve value here as the updated value for j is equal to 4 and use the guess value here and relax this particular line; you sweep in y direction.

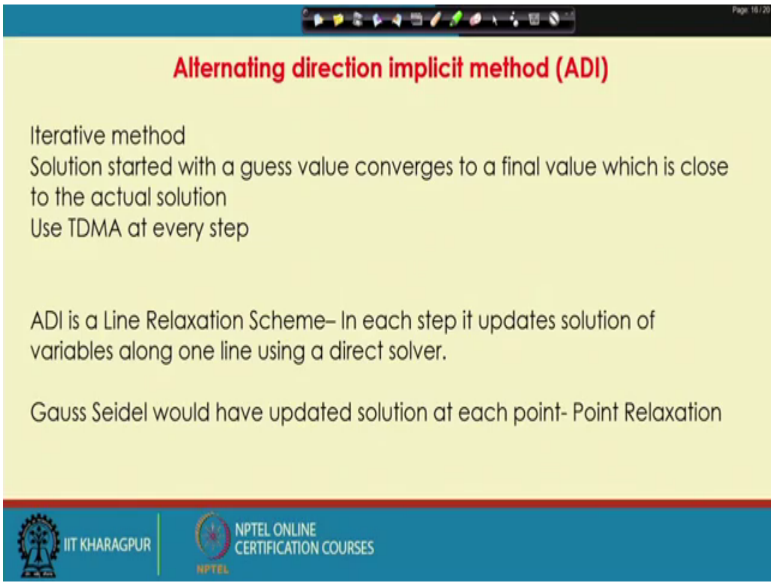
Similarly, you will have to after the sweep in y direction is complete, you have to do a sweep in x direction that is you start that this is already known as the boundary; use a use a guess value for this line and then, solve for this particular line where you will where this will be the guess, this will be the guess and go there and you will solve for T i j minus 1 minus 4 T i j plus T i j plus 1 is equal to some b star which is using these guess value. So, you will form this b star.

So, what you essentially do? You solve along all the j is equal to. So, along all the j points here j is equal to 1 to n, all the j points here you do a solution in this particular line and then, you move into another line.

So, every time instead of solving the complete matrix at every step, we are only solving the matrix along one particular line and assuming that rest of these things are known and putting it into the p part and that is why this is called a line relaxation method. Continue the sweep till solution variables converge to a final solution so that it does not change to any other, change in any other sweep.

So, essentially the idea of iterative methods, the solutions converging to a particular solutions all these things will be important here and it is it will be also important see the rate of convergence of this, considering it to be a basic iterative technique, which can be verified if you try to write it as a matrix equation you can verify that it converges and what are the rate of convergence if something can be done to increase the rate of convergence etcetera. We will try to explore some of this idea in block relaxation method.

(Refer Slide Time: 13:36)



So, this is essentially an iterative method, solution started with a guess value converges to a final value which is close to the actual solution and it uses TDMA at every step. Advantage is that TDMA is a very first solver; but this can only be done for a tridiagonal matrix.

So, we break down into iterative methods where iteration is done now over the lines and each line is solved as a line relaxation method using TDMA; each line is solved at in a moment using TDMA, it just a number of steps if at the points here and then, we are iterating over the lines and we are changing their direction of iterations etcetera direction of relaxation etcetera. So, it is an ADI method.

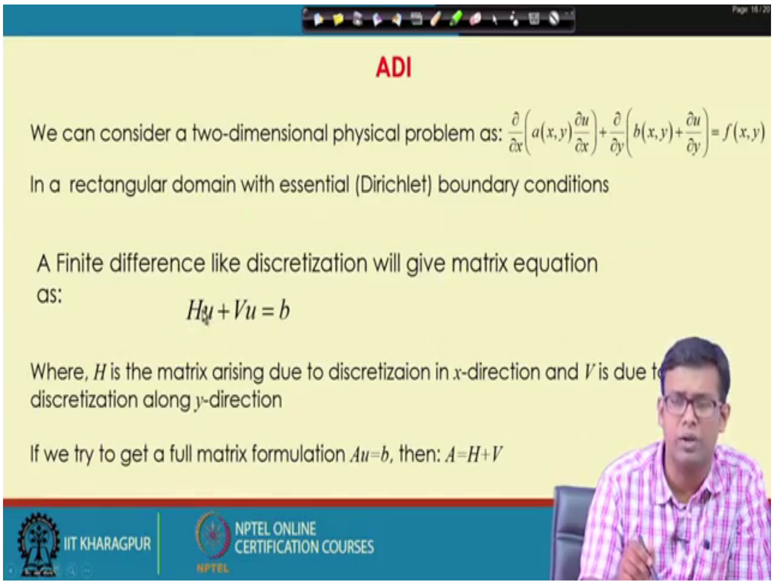
However, this is the first turn method consists compared to Gauss Seidel Jacobi because use of TDMA makes solutions in one line like updating the solutions will be much faster along one line due to its use of TDMA. ADI is the line relaxation scheme- in each step it updates solution of variables along one line using a direct solver and that is why we call it to be a line relaxation scheme because the values are relaxed in one particular line.

We get up smoothers we start with some solution and at the end we get a smooth function T across the domain. So, we start starting with any guess solution. So, we say that we relax the solution to the right solution through this method this. Then, that is why this particular term relaxation is being used here.

So, in less technical word relaxation this word when whenever this will come in matrix solution perspective, relaxation means basically iteration for one particular step. Gauss Seidel would have updated solution at each point that would have been a point relaxation. TDMA solution is updated at along one particular line.

So, it is a line relaxation scheme, if we had apply sorry not TDMA. ADI is the line relaxation scheme because it is updating solution using a direct solver at one particular line. Had we used Gauss Seidel, we would have used a point relaxation scheme because we would have we will be updating solution at if each point instead of a line will go to each node and update the solution.

(Refer Slide Time: 16:11)



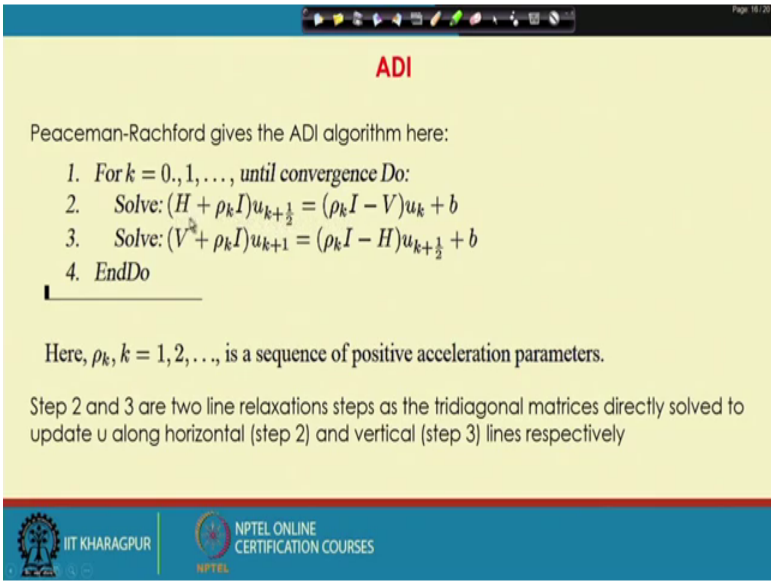
We will look little more into the theory of ADI and convergence we can consider a 2 dimensional problem del del x of a x y sum function of x y del u del x plus del del y b x y is equal to f x y.

In a rectangular domain with essential or Dirichlet boundary condition, we could be Dirichlet boundary conditions as essential boundary condition because at least at one boundary, the value has to be exactly specific. T is equal to T i something has to be specified for having a unique solution of the equations. So, this is essential to specify a Dirichlet boundary that is why Dirichlet is also called essential boundary condition.

So, finite difference like discretization scheme when applied over here, we will give H u plus Vu is equal to b; where, b is a matrix. V is a vector; u is the solution vector; H and V are 2 matrices. H matrix is arising due to the discretization in x direction; V matrix is arising due to discretization in y direction. H is a matrix arising due to discretization in x direction and V is due to discretization along y direction. If I try to write a full matrix equation, it will be A u plus is equal to b; A is equal to H plus V.

So, we can say that if we have to use an ADI, we will first try to assume u in the y directions and write an H u is equal to b star some equation like that which will be line relaxing along x along y constant line along x lines. Similarly the next step will assume u to be known along x direction and relax it along y direction. So, this will be Vu is equal to b b star prime or something.

(Refer Slide Time: 18:05)



This is called a Peaceman-Rachford algorithm for ADI, where k is equal to 0 to 1 until convergence solve H plus rho k I u k half instead of u k plus 1 instead of the new guess is u k plus 1 instead the new guess we said it is a half guess when we relax along one particular line. So, this is along x direction you solve H plus rho k I is equal to u k plus half plus rho k I minus Vu k plus b.

So, if this might be a question that what is rho k? Rho k is something like an over relaxation factor, it is called an positive acceleration parameter. Rho k is introduced here to increase the convergence rate of the equations of the iterative schemes. So, that the schemes converge faster and we will see what are the values of rho k etcetera later. Similarly, you solve a similar problem in y direction. So, there are two line relaxations; you do these two line relaxations and solve it.

Here rho k for k is equal to 1, 2 at different iterations, the values of rho k might changes is a sequence of positive acceleration parameters which is designed to increase the improve the convergence of this method. Because this is essentially a an iterative method very similar to the basic iterative methods we have discussed. We will have similar properties in where the convergence will be limited by the matrix and how this matrices has been split in H and V etcetera.

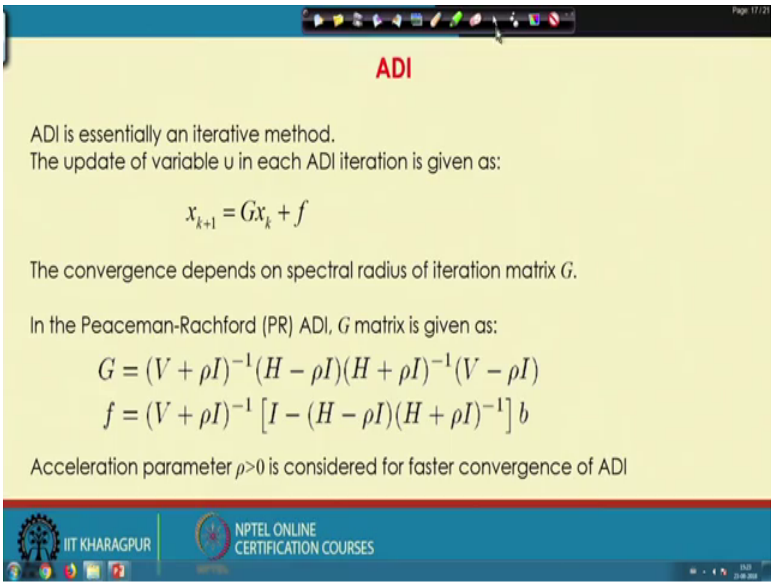
So, we need something to fast the make the convergence faster or accelerate; that means, scheme. So, rho k is used you can very well go ahead with rho is equal to rho k is equal to 0 which will be the ADI method that we have discussed before. Interestingly, if we see that the iterations is k is equal to 0, 1, 2 these are the iteration level.

In each iteration level, the iteration the scheme is broken into 2 semi iteration levels; the first semi iteration level is along x, second semi iteration level is along y. And in the first iteration semi iteration level, we are using the this is like a Jacobi scheme. So, we are using the last updated value u k. In the second iteration, we are using the updated value from u k plus half which came from the first semi iteration.

So, there are 2 step iteration; each step iteration uses something like a Jacobi method, where it uses the last updated value; updated value after the last iteration and updated value after the last semi iteration is used here. Step 2 and 3 are two line relaxation steps as the tridiagonal matrix directly solved to update u along horizontal in this that there is a tridiagonal matrix which can be updated along horizontal.

Why is the Tridiagonal? H is a tridiagonal as it comes from central difference of a second order derivative term and rho k is I will only add the diagonal component, will make it more diagonally dominant. So, the solution is faster. And the second step is a vertical line relaxation scheme.

(Refer Slide Time: 21:28)



ADI is essentially an iterative method. The updated variable in each a d I iteration is given like the basic iteration method it is x k plus 1 instead of x k, this will be now u k plus 1 what we are doing u k plus 1 is equal to G u k plus f. This instead of x, this will be u k. The variable is updated u because it would have been perfect if we are using x k, but instead of x we discuss that the entire discussion started with the fact that the variable solution variable is 0.

So, we are writing u k plus 1 is equal to G u k plus f. I will try to correct it in the in the slides that we will upload later. So, what is this? This is very similar. This is very similar as the basic iteration basic iterative step for Jacobi or for Gauss Seidel; this step is very very much similar to that.

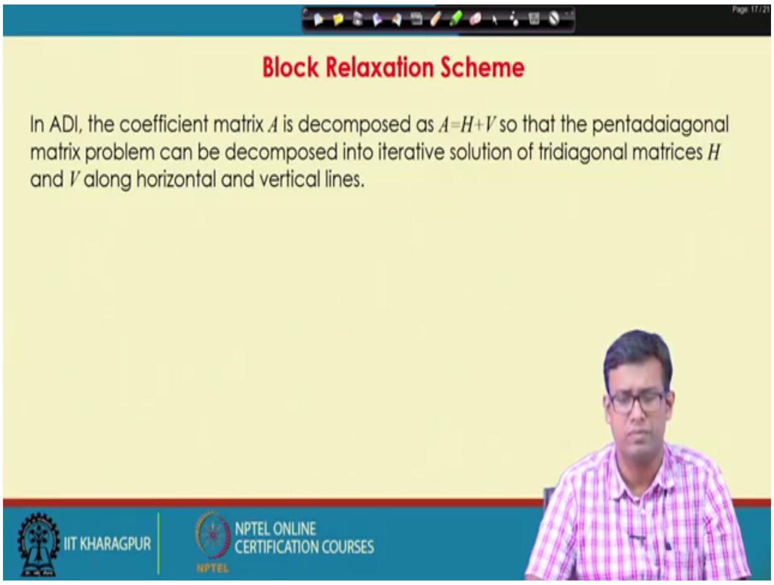
So, the convergence will depend on the spectral radius of the iteration matrix g in the Peaceman-Rachford scheme, the G matrix is given as G is equal to V plus rho I inverse H minus rho I H plus rho I inverse V minus rho I. So, if rho is equal to 0, this is V inverse H, H inverse V something like that which will be actually very difficult to converge because H H inverse will give us 1 and V V inverse will give us 1. So, it will be almost 1 will be the convergence parameter.

So, in a Jacobi type of algorithm, it will be very difficult to get it converge to you; that is why is ADI without no acceleration comes as the Gauss Seidel implementation, where we use the last updated value not the last total value.

So, now depending on value of rho, the spectral radius of G can be less than 1 and therefore, we can get a solution. (Refer Time: 23:50) As acceleration parameter rho is greater than 0, this considered for faster convergence of ADI as we increase the value of rho up to our optimal value this matrix G shows a smaller spectral radius.

So, it is required that rho G is less than one and as smaller as the spectral radius the convergence will be faster. So, based on the parameter rho, this is a different rho right. This is acceleration parameter rho and this is spectral radius of G less than 1. So, as we increase the value of rho, the spectral radius reduces and we get a faster convergence.

(Refer Slide Time: 24:48)



Now, the next idea will be a Block Relaxation Scheme. Instead of decomposing the matrix H into matrix A into the vertical sweep and horizontal sweep matrices, if we have to decompose the if A is if some form that it can be decomposed into small diagonal blocks and in most of the cases there is a sparse matrix with lot of diagonal 0. So, we can decompose it something like that and can introduce in a sense the granularity of the solutions.

So, the same idea can it be more generalized; can it be applied on not only for x and y sweeps, x y z sweeps or can be can it be can it be applied for a different location of the domain will be solved as a different block etcetera. So, that is called Block relaxation scheme and we will look into it in the next session.

Thanks.