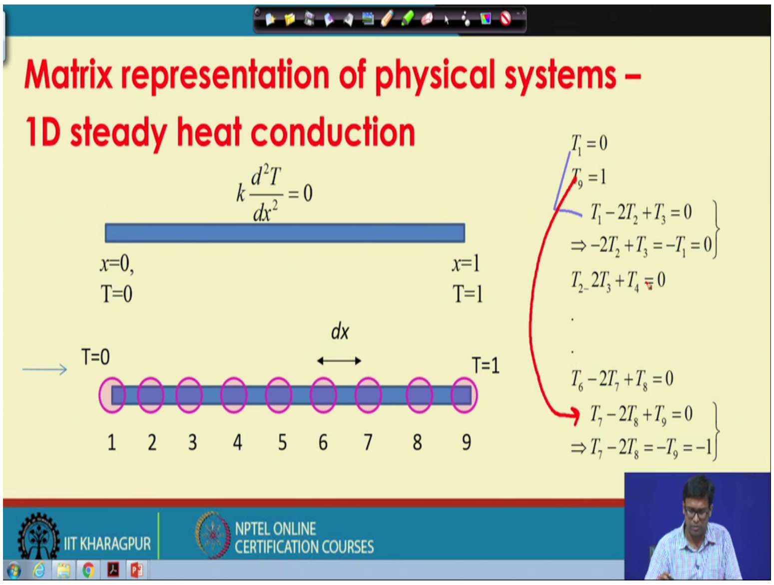
Matrix Solvers
Prof. Somnath Roy
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 11
Tridiagonal Matrix Algorithm

Hello. In last class, we have seen how Laplace equation can be converted into a difference equation and from there we got a tridiagonal matrix, which is basically we are trying to solve heat conduction equation in an one-dimensional one steady heat conduction equation.

So, this class we will see how this equations can be solved; what is the most efficient way to solve this equations. And we look into an algorithm named Tridiagonal Matrix Algorithm.

(Refer Slide Time: 00:47)

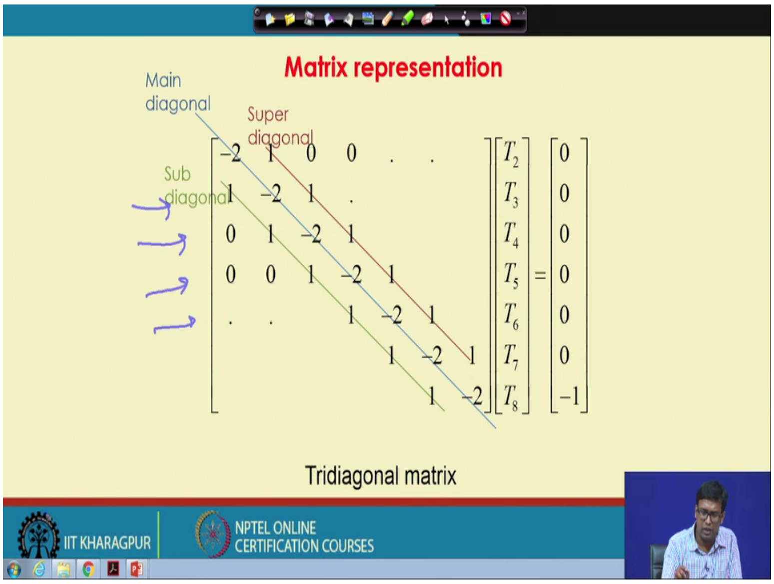


We had this problem. When we were looking into matrix representation of physical systems – 1D steady conduction rod we where the governing equation is k d 2 T d x square is equal to 0.

And the boundary conditions x is equal to 0, x is equal to 1. And we convert discretize these into number of finite points, where we finite number of discrete points, where we got the difference equation for each point, and the boundary condition at x is equal to 0; T 1 is equal to 0; at x is equal to 1; T 9 is equal to 1.

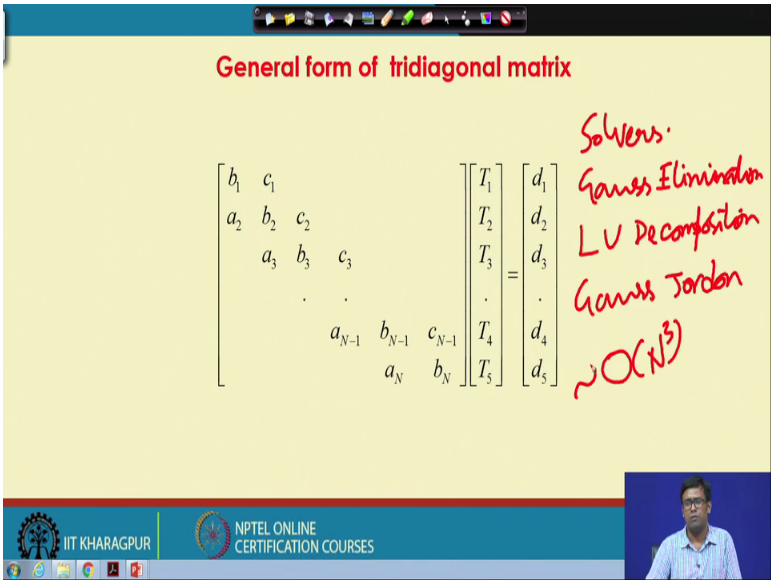
And the finite points discretized equation and first equation contains a T 1, so we substituted T 1 in the first equation for example and we also substituted T 9 in the last equation, and these two equations have changed actually. But we may in between equations remain 1 minus 2 and 1 for coefficients of the diagonal term and the next two terms. And what we got the we discussed it in last class; what we got is called a tridiagonal equation system.

(Refer Slide Time: 02:01)



The matrix representation is this; this is a tridiagonal matrix. The diagonal there is a main diagonal term; there is one sub super diagonal term, which is the coefficient for the next points in the mesh. And the sub diagonal term which contains the coefficient for the previous points in the mesh.

(Refer Slide Time: 02:21)



General form of tridiagonal matrix is we if this particular form it was a symmetric matrix with minus 2 1 1 minus 2 1, and the coefficients because we considered an uniform mesh the coefficients were repeating in each row.

However, in a general form, it might not repeat or we can get a general form of tridiagonal matrix where the there are tridiagonals from row 2 to row N minus 1. And there are only two elements, because the a 0th element is not there or N plus 1th column is not there, there are only two elements in for first and last row, and remaining all these terms are 0.

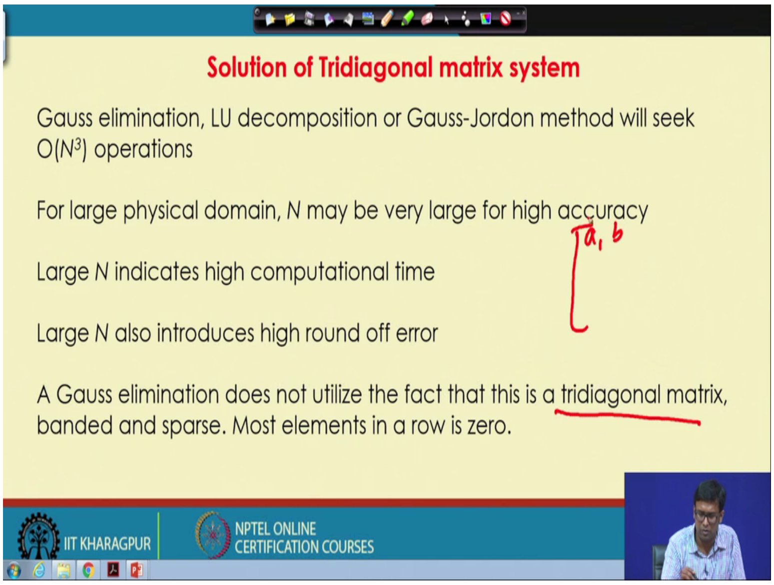
Now, this is a very frequently arising matrix in lot of scientific computing problem, and we have to see how we can solve this. Interestingly this is not only in not only comes from approximation of differential equation to difference equation.

When we will do (Refer Time 03:28) method, we will see that some of the functional spaces, we will also transform to a tridiagonal system. There is a very important solution algorithm called generalized minimum residual algorithm where or biconjugate stabilized biconjugate gradient method, where we will see tridiagonal matrices are coming in between the solution space.

However, we have to see how we can solve this equation. And we have right now went through three different solvers one is Gauss elimination; one is LU decomposition; and the other is Gauss-Jordon. This solvers all these solvers for Gauss elimination, Gauss-Jordon, and also for doing the decomposition, LU decomposition the number of operation for solving the matrix is of the order of N cube.

So, if we have a 10,000 row matrix, the number of solution will be 10 to the power 12, number of operations will be 10 to the power 12, which is quite high. So, we will see is there any other method for this particular type of solution; and this is important, because many times we will interact with matrices of similar form.

(Refer Slide Time: 05:01)

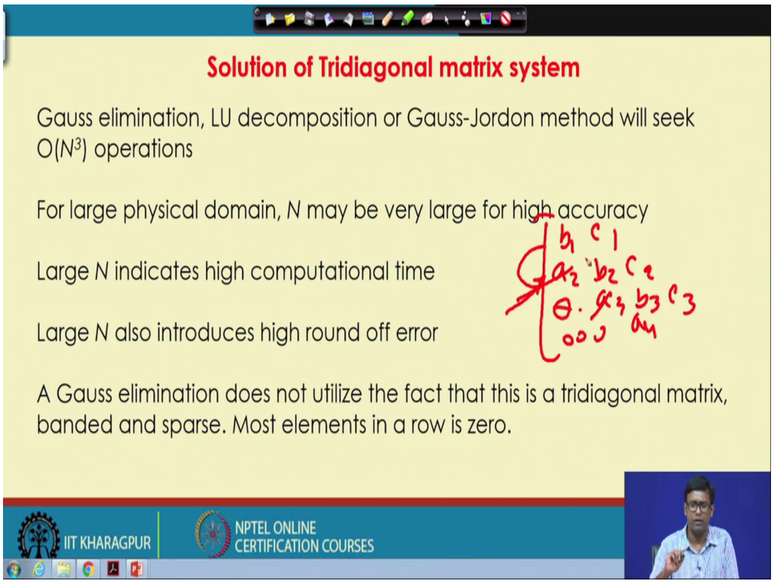


So, solution of tridiagonal matrix system; Gauss elimination, LU decomposition or Gauss-Jordon typically six order of N cube operation. For large physical problem, N may be very large for high accuracy. We have seen in the last class that as we increase N, d x reduces and the error reduces. So, for a getting good high accurate highly accurate solution, N has to be large, so that d x is small.

However, there is one problem, which we have also seen in last class; that there are round of errors. So, as we in increase N, there will be high round off error, because each grid there will be some round; each operation is associated with sound some round of error. Also as we increase N, the number of operations are of the order of N cube, which is very high.

A Gauss elimination is a method for a general matrix, but it does not utilize the fact that here for this particular tridiagonal matrix form, which is this matrix is banded and sparsed. And most elements in a row is zero.

(Refer Slide Time: 06:15)

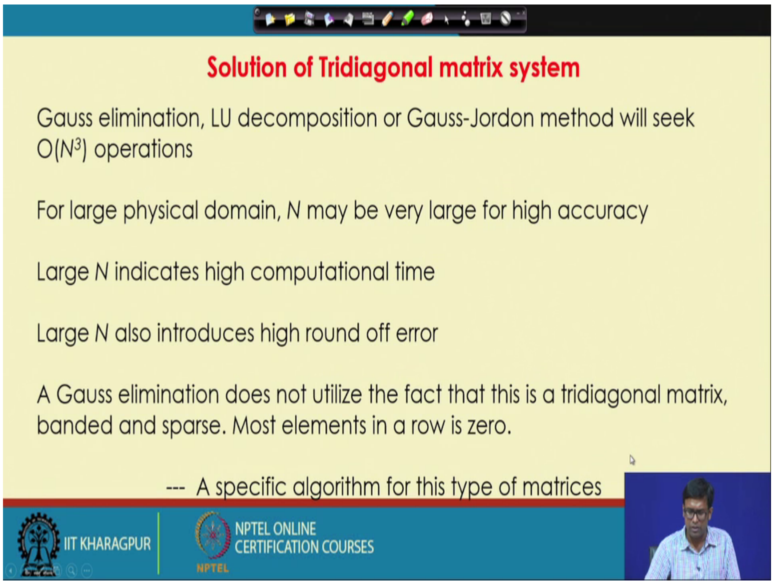


For example, if we have if we apply Gauss elimination a 1 sorry if we apply Gauss elimination b 1, c 1 and then a 2 b 2 c 2 a 3 b 3 c 3 , if we try to apply Gauss elimination here, in first step this a 1 will be this a 1 will be eliminated. But we do not need to eliminate anything here, because this is already 0.

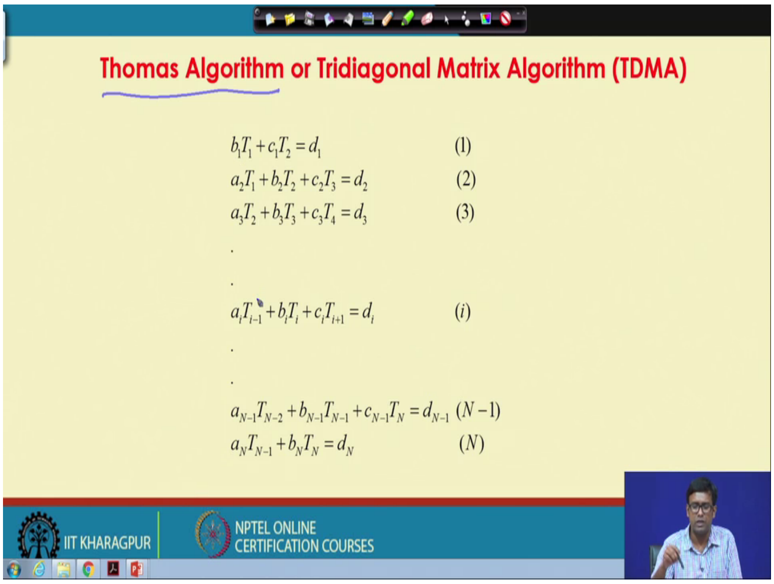
Similarly, all this are already 0. When we will do the next step using b 2, we can eliminate a 2, but we do not need to eliminate anything here, it is already 0, it starts from a 4 here. So, most of the elements in a row is zero, and we really do not need to eliminate row all the terms below one particular pivot.

Only the next row below the particular pivot can be eliminated, and we can get through with it. So, this few properties we can utilize. And then have a specific algorithm for this type of matrices.

(Refer Slide Time: 07:06)



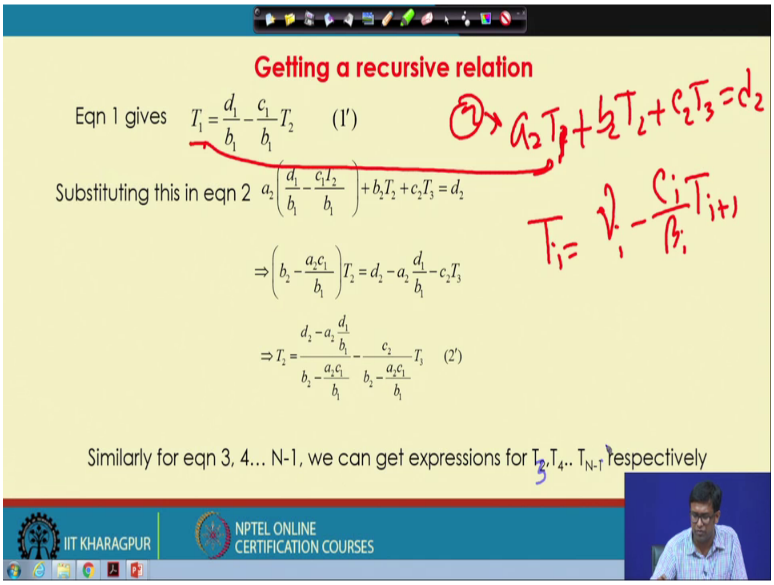
(Refer Slide Time: 07:13)



And this algorithm is known as a tridiagonal matrix algorithm of TDMA. This is also known as Thomas algorithm, because there was a mathematician name Thomas, who proposed this algorithm. So, we write down all the equations starting from equation 1 to equation N.

And now, and then what we will do, we will do something like a Gauss elimination step for us, so we will eliminate the T 1 from the second equation, and then we will eliminate T 2 from the third equation and so on.

(Refer Slide Time: 07:58)



However, we will see that there is a recursive nature of this elimination process, when we will do that. So, we go for the elimination round; and get a recursive relation. That equation 1, if we go back to equation 1 again, equation 1 gives us T 1 is d 1 minus c T 2 by b 1. Now, if we substitute this into equation 2, we will get a 2 into d 1 minus this plus equation 2 was basically a 2 T 2 a 2 T 1 plus b 2 T 2 plus c 2 T 3 is equal to d 2, this is equation 2.

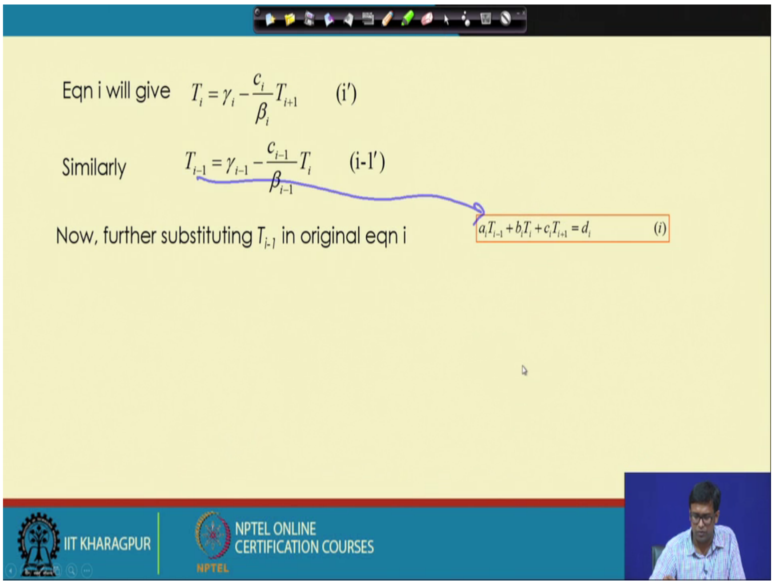
So, if we substitute this T 1 into equation 2, we will get a 2 into this particular term plus b 2 d 2 plus c 2 T 3 is equal to 0. So, this becomes an equation between T 2 and T 3. So, we can write something into T 2 is d 2 minus something minus something into T 3. And T 2 is equal to d 2 minus a 2 b 2 by b 2 minus a 2 a c 1 by b 1 minus c 2 by b 2 minus a 2 c 1 by T 3.

So, few things we can see, which are interesting here, that there is a general form that T i oh sorry. There is a general form that T i is equal to some constant say gamma i minus C i divided by some constant beta i by T i plus 1, this is a general rule for form we are getting for this two equation.

Similarly, we can get similar expressions for equation 3, 4 to N minus 1, where we can get expression for T 2 involving T 3. For equation 4, we can get expression for sorry sorry we can for equation 3, you can get expression for T 3 involving this will be T 3. T

4, for equation 4, we will get expression for T 4 involving T 5, and up to equation N minus 1 where we will get expression for T N minus 1 involving T N.

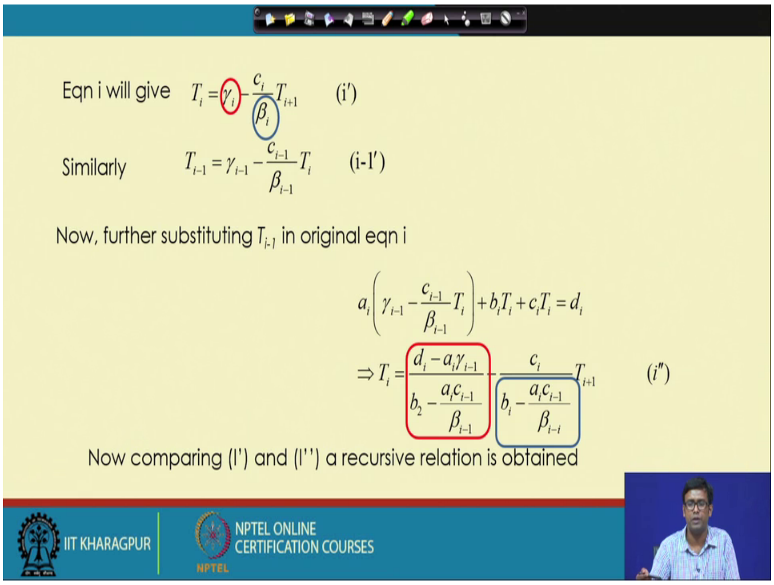
(Refer Slide Time: 10:24)



So, equation i will give a general form, T i is equal to gamma i minus c i by beta i T i plus 1; this gamma and beta will change for each of the terms. So, equation i is converted to equation i prime. Similarly, equation i minus 1 will also have similar form T i minus 1 is gamma i minus 1 minus c i minus 1 beta i minus 1 T i.

Now, if we further substitute this T i minus 1 into the original equation i, we will get the original equation is a i T i minus 1 plus b i T i plus c i T i plus 1 is equal to d i. So, this T i minus 1, we the expression for T i minus 1, we will substitute this into this particular equation.

(Refer Slide Time: 11:17)

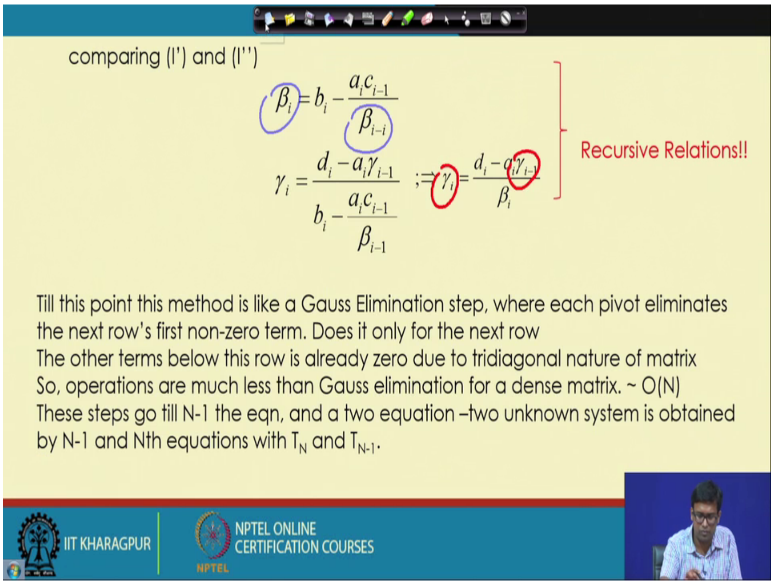


And what we will see that a i into the expression for T i minus 1 here plus b i T i plus c i is equal to 0. So, we can rearrange it and write T i is some constant minus something c i by minus c i by another constant into T i plus 1. Now, if I compare equation i prime and equation i double prime, there basically same equation T i is expressed in terms of T i plus 1.

We will see that the first term matches with this term or you can probably see it here. This term matches, this particular term; this term matches this term; and this term will match this term. So, I will see the that beta i is b i minus a i c i by beta i minus 1. So, if I know beta i minus 1, I can find beta i; that is called a recursive relation that one term is dependent on the previous term in that particular series.

Similarly, once I know beta i minus 1 and I also know gamma i minus 1, which is from the previous equations, we can find out gamma i. So, gamma i is d i minus a i gamma i minus 1 b 2 minus a i c i minus 1 by beta i minus 1. And comparing 1 I prime and I double prime, we get what is called a recursive relation. So, if we know 1 particular we know gamma 1, beta 1, we can find out gamma 2 beta 2; we know gamma 2, beta 2, we can find out gamma 3, beta 3 so on we can do it till gamma N, beta N.

(Refer Slide Time: 13:11)



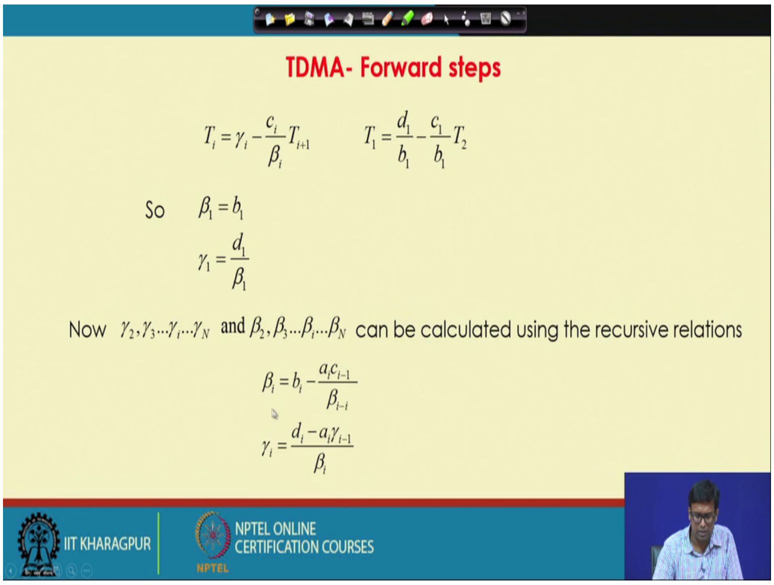
Comparing 1 and 1 prime, we can write beta i is equal to b i minus a i c i minus 1 beta i minus 1 and. So, beta i is, if we know beta i minus 1, if we know beta i minus 1, we can find beta i; and if we know sorry and gamma i, where this term is nothing but beta i here. So, if I know I have earlier found beta i, so if I found beta i or I know beta i minus 1 and no gamma i minus 1, then I can find gamma i. And this is what we are seeking for a recursive relation. We will see why is it important.

Till this point this method is very much like a Gauss elimination. We are eliminating to only the below pivot, next row below pivot element and where each pivot elements the next rows first non-zero element. However, it does it only for the next row; it does not do it for the later rows, because the later rows have zero at that element.

The other terms below this row is already zero due to tridiagonal nature of matrix. So, operations are much less than Gauss elimination operation for a dense matrix, and we will see this is the number of operations are of the order of N; if N is a total size or number of rows.

Thus these steps will go till N minus 1 equation, we will eliminate we will express T N minus 1 as a function of as a constant plus something into T N. And when we will get 1 equation for N minus 1, and we already have the last equation which relates T N and T N minus 1. So, you will get two unknown system, two equation two unknown system by N T N minus 1 and T N using N minus 1 and Nth equation, and we can solve that.

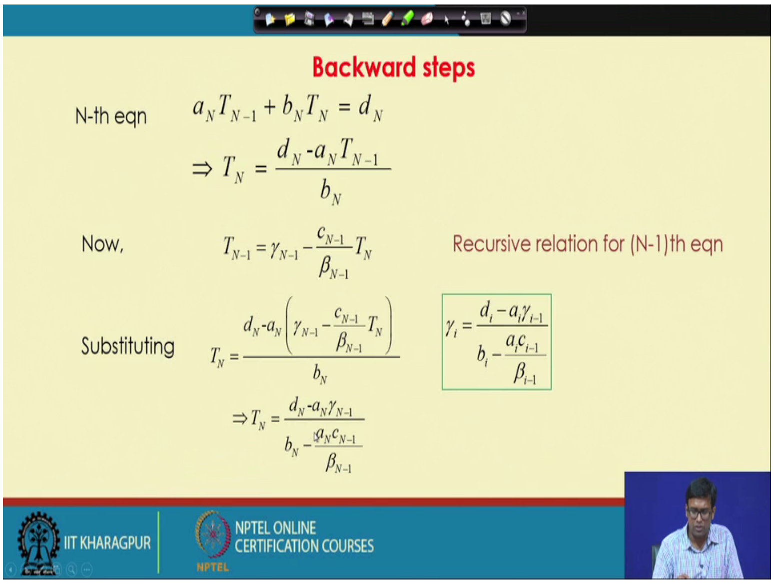
(Refer Slide Time: 15:19)



However, we will try to find out a more elegant algorithm here. Sorry so, sorry, so this is the forward step in TDMA. What is you got T i is equal to gamma i minus c i by beta i T i plus 1; and get what is T 1.

Here, we can find out beta 1 is equal to b 1; and gamma 1 is equal to d 1 beta 1. And then we have the recursive relations, then we will have the recursive relations b beta i and gamma i dependent on beta i minus 1, and gamma i minus 1. And we can calculate gamma 2 beta 2; gamma 3 beta 3 so on up to gamma N beta N.

(Refer Slide Time: 15:57)

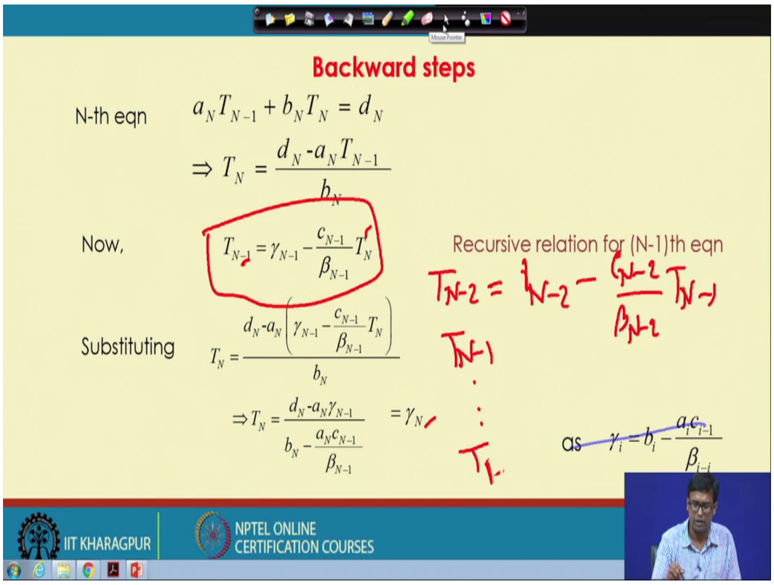


Now, we will do the backward steps. The Nth equation, we will start with the Nth equation. Nth equation is the equation of N minus 1 and N we can write, T N in terms of T N minus 1 in the Nth equation. Now, T N is equal to gamma N minus 1 minus c N minus 1 beta N minus 1 by T N, which is using the recursive relation for N minus 1th equation which we got there.

If we substitute this, if we substitute this T N minus 1 into as this T N minus 1, which is coming from N-th equation, we will get the value of T N. We will get an equation involving T N only, and we will get the value of T N as T N is d N minus an gamma N minus 1 b N minus an c N minus 1 by beta N minus 1.

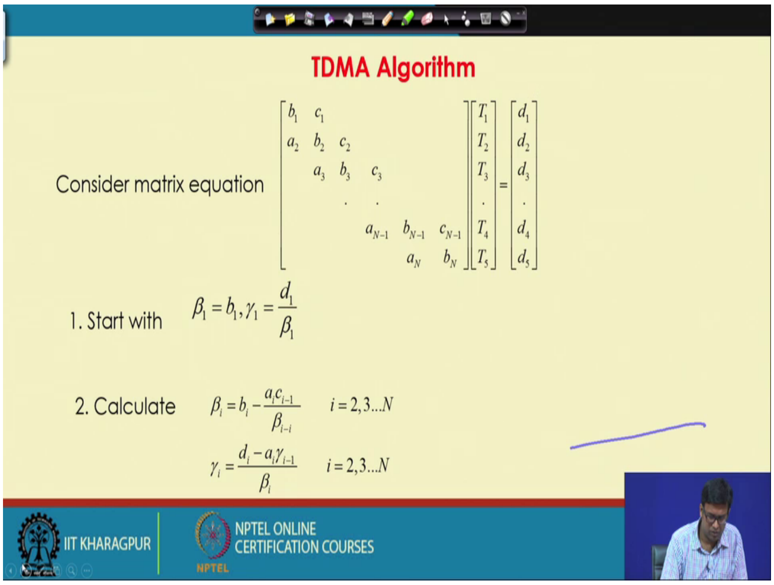
And gamma i is given as d i minus a i gamma i b i minus a i c i minus 1 b i minus 1, which is the generated recursive relation for gamma i. If we compare these two things, T N is nothing but gamma N.

(Refer Slide Time: 17:09)

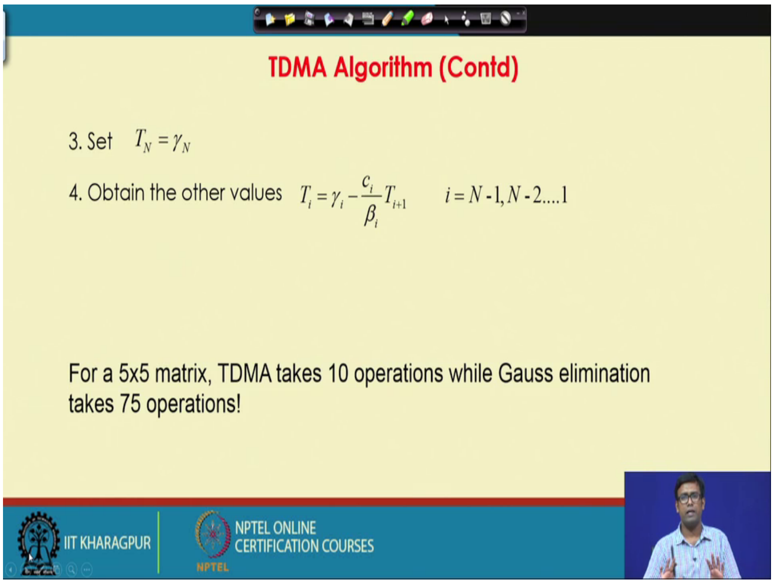


So, as this is not needed this is not needed at this stage. So, T N directly comes to gamma N, if we compare this two relations. So, what we have done in forward steps, we calculate using the recursive we started with gamma 1 and beta 1 from first equation using the recursive relationship; we came up to gamma N and beta N. And when we came up to gamma N T N is nothing but gamma N. So, now we have to do the backward step starting from T N is equal to gamma N.

(Refer Slide Time: 17:49)



(Refer Slide Time: 18:01)

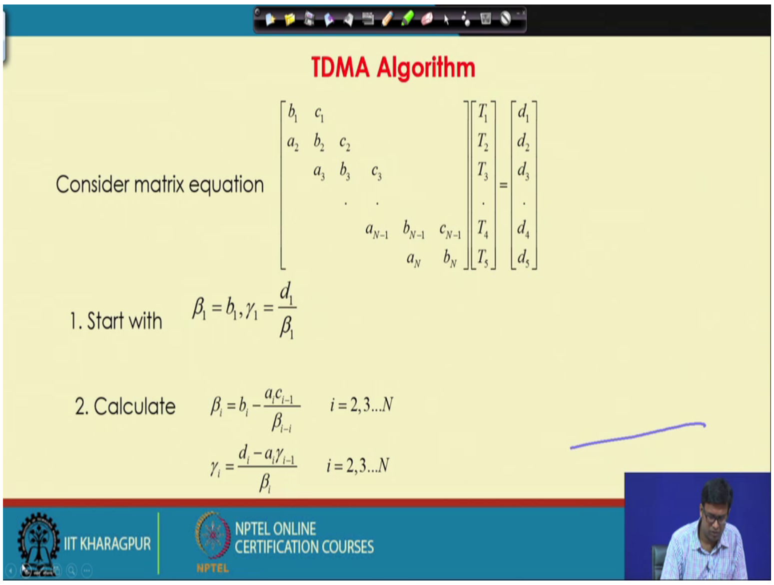


The backward step will follow nothing but; the backward step we will follow nothing but this particular relationship, that T i minus 1 is gamma i minus 1, which is already calculated c i minus 1 beta i minus 1 into T i plus 1, so I will get T N minus 1. Similarly, I will get T N minus 2 once I for I got T N, I got T N minus 1 using T N.

Now, I will get T N minus 2 is gamma N minus 2 minus c N minus 2 by beta N minus 2 T N minus 1. Similarly, I will get T N minus 1 and so on I will get up to T 2. This T 1 I already know, T up I will get up to T 1 sorry, T 1 is not known to me; I will get up to T 1.

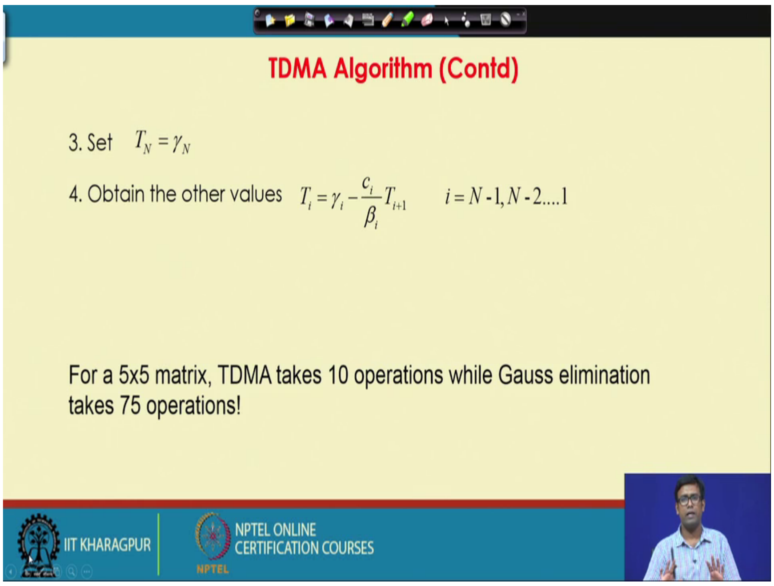
So, now if we can summarize this algorithm, this will be probably of more help we looked into derivation of the algorithm if we can summarize it in form of the algorithm.

(Refer Slide Time: 19:11)



This is the matrix equation, which is a tridiagonal matrix equation, we are up in which we are applying TDMA. We will start with beta 1 is equal to b 1 gamma 1 is equal to d 1 by beta 1. We will calculate beta and gamma for all other term all other 2 to N using the recursive relationship.

(Refer Slide Time: 19:33)



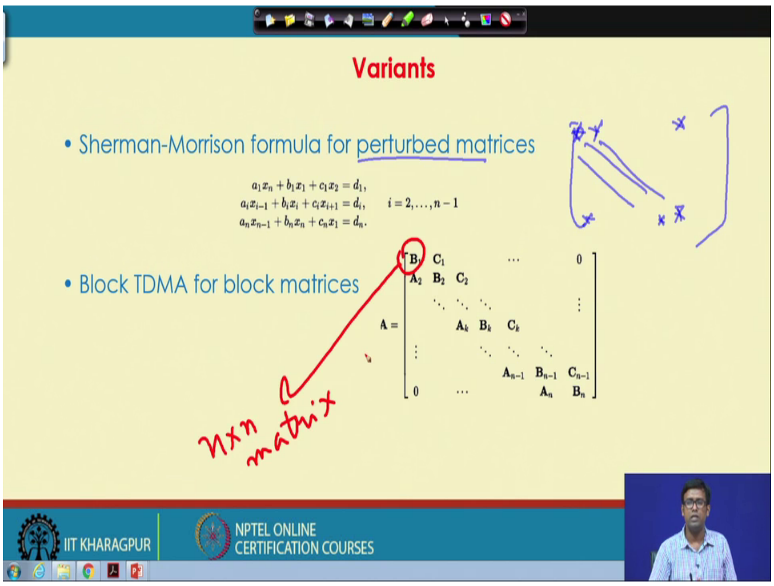
Now, I will set T N is equal to gamma N, which is the actual solution of N-th point. And then we will use the relationship, which is again a recursive relationship T i is equal to something T gamma i minus c i by beta i T i plus 1 for N minus 1 N minus 2 and up to 1 and we will solve the equation.

And this is the entire algorithm, which is a very straight forward algorithm only if we try to write a computer program, we have to write the expressions for gamma i and recursive relation for gamma i and beta i. There will be one sweep calculating wall gamma and beta from 1 to N, and there will be one backward swim calculating all T's from T N, T N minus 1 to T 1. For a 5 into 5 matrix, TDMA takes this is the order of N; So, just 10 operations, 5 operations forward step; 5 operations backward step.

While a Gauss elimination takes 75 operations, so it should be any computer program it should be more than 7 times faster than a Gauss elimination process. For a larger problem, it will be the efficiency can be much more evident, because, say for a 100 into 100 matrix, TDMA will take few 100 operation 200 operations.

Where Gauss elimination will take order of 1000 operations, so it will be 10 times faster in that case; Not 1000, 100 cube 10 to the power 6 operations, so it will be much much faster, TDMA will be much much faster. When TDMA is very useful algorithm used for tridiagonal matrices which occur heavily in very frequently in different scientific computations cases; specially in cases where one-dimensional problems are involved.

(Refer Slide Time: 21:27)



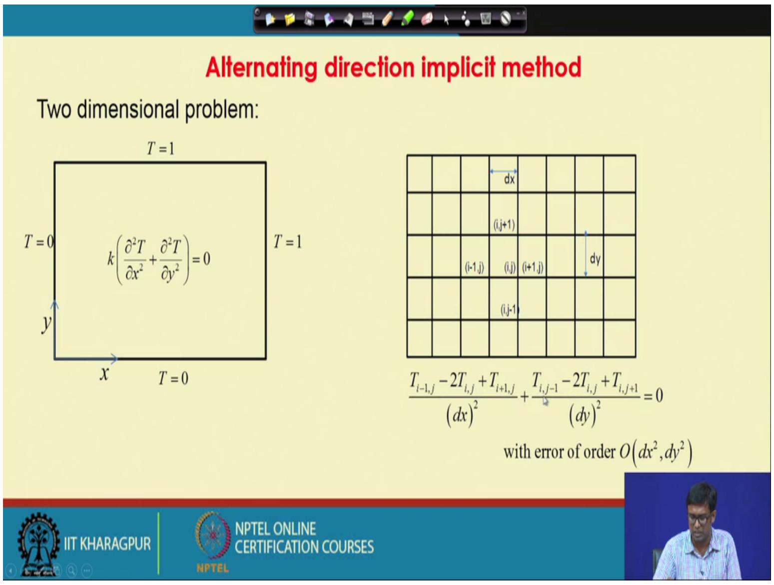
There are variants of TDMA, one is that instead of just being the it tridiagonal the central point the diagonal point, the next point and the previous point, there is a something like a periodic boundary condition is there. There is one band, which is away from the diagonal band; which is called the perturbed matrix is associated here.

For example, this is of the you will see like these are the general diagonal forms. The first term, we will have one term here 1 1 0 here; 1 1 0 here; and 1 1 0 here. Similarly, the last term, we will have 1 1 0 here; 1 1 0 here; and 1 1 0 here, which is near tridiagonal matrix, but it is parted from the tridiagonal space.

And that is Sherman-Morrison formula, we will look it in detail the formula is available widely in internet resources, which is a variant of tridiagonal matrix. There can be block tridiagonal matrices, instead of having a tridiagonal matrix there can be block matrices arranged in tridiagonal form like A is combination of several matrices in arranged in tridiagonal space. We will look into block algorithms later of this course, later in this course.

So, like b 1 this is probably an n into n matrix. So, there are number say large matrix and number of matrices arranged in block, and this block arrangement is tridiagonal. And using TDMA algorithm variant can be this obtained where the block matrices can be solved. Is this these are not very difficult very straight forward implementation of TDMA algorithm for this specific problems can help here.

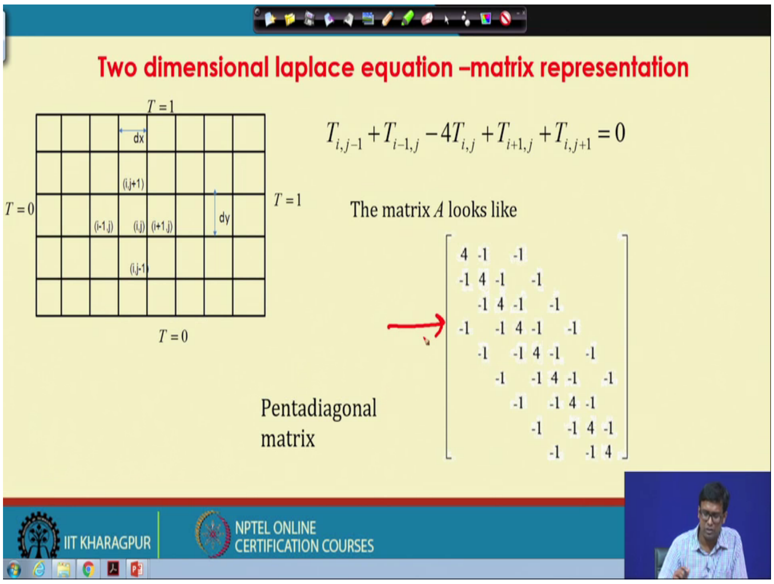
(Refer Slide Time: 23:15)



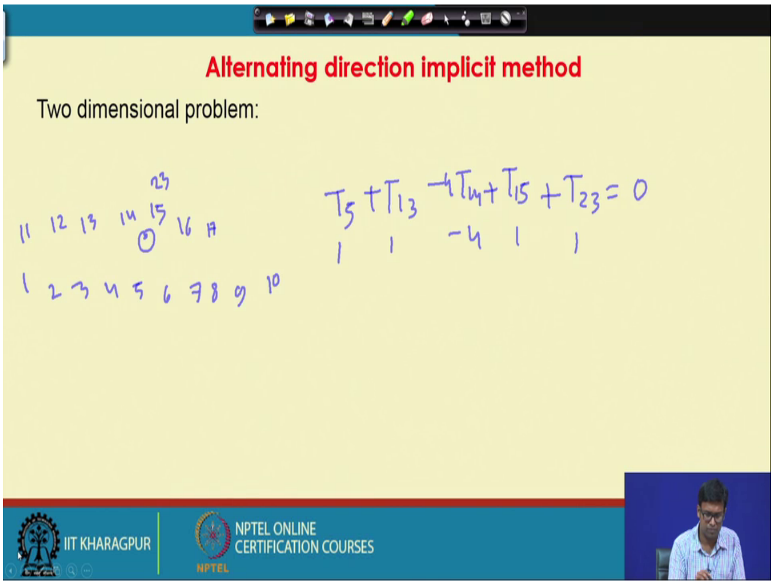
There is another interesting method, which is called alternating direction implicit method. And this method is applicable for two-dimensional problems. What is in a two-dimensional problem, we have seen it earlier that this is a Laplace equation in 2D, which is converted into a we will get a matrix equation out of it into an equation involving variation of temperature along the points in both x and y direction.

So, it has T i minus 1 j T i i plus 1 i j T i plus 1 j by d x square plus T i j minus 1 T i j minus 2 T i j plus T i j plus 1 by d y square is equal to 0. So, it is not of tridiagonal nature, if we try to look into the matrix, and looking to the matrix actually.

(Refer Slide Time: 23:50)



(Refer Slide Time: 24:18)

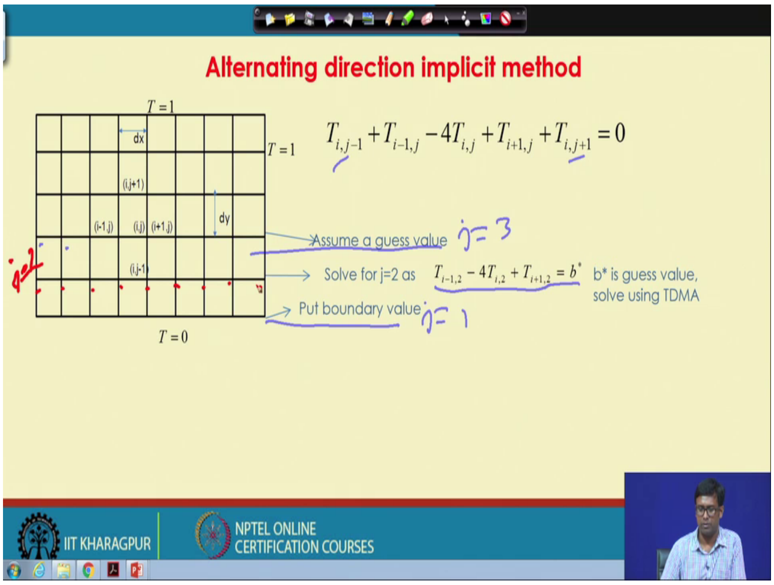


So, this is the equation for each point. And now if I start numbering this points, like this is a boundary condition, so this number 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 so on. So, we will get that for this particular point, which is my 14 point now. We will get T minus 4 T 14 plus T 15 plus T 13, and this is T 5 and this is this is 5 this is 14, so this will be 14 plus 9 - 23 plus T 23 is equal to 0.

So, therefore, we can get a matrix equation with 1 1 minus 4 1 1 - 5 nonzero bands, and this will be like this oh sorry. We can get a matrix equation with this 5 nonzero bands sorry, see this is our. So, we will get a matrix equation with this 5 nonzero bands, which is 4 minus 1 1 minus 1 4 minus 1 1 minus 1 minus 1 4 minus 1 minus, this is a general row of this matrix; and this is called a Pentadiagonal matrix.

So, we can see, it is not a td tridiagonal matrix, so TDMA will be directly applicable here. So, what we have to do here, there is a variance an application of TDMA, which is called alternating direction implicit method; which will be applicable here.

(Refer Slide Time: 26:32)

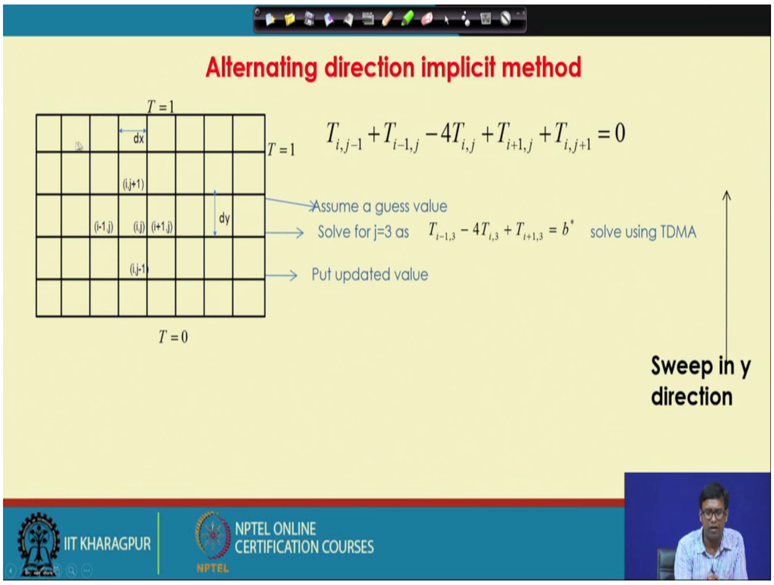


And this starts with this particular equation, and it will try to make it TDMA in a sense that it assume the there is a boundary condition for j is equal to 0, j is equal to 1 row. It assumes a guess value for j is equal to 3 row; for j is equal to 2 row, so there is a boundary value for j is equal to 1 j is equal to 1.

There is a guess value which is assume for j is equal to 3. So, for j is equal to 2, this these values, the j plus 1 and j minus 1 values are already have obtained from guess or boundary value. It the unknowns remains only the as of for tridiagonal form, which is which now all the points in this particular row or say the here we are solving in this particular row, all the points in j is equal to 2 row this all these points can be solved using a TDMA algorithm.

And once this is done, now I have updated values for all these points what is the present value of in this points. Instead of guess, I will use this value here, I will have a guess value for this set of points, and solve for solve a TDMA, and j is equal to 3. So, what I will do, I put an updated value here, I assume a guess value here, and solve for j is equal to 3 using TDMA.

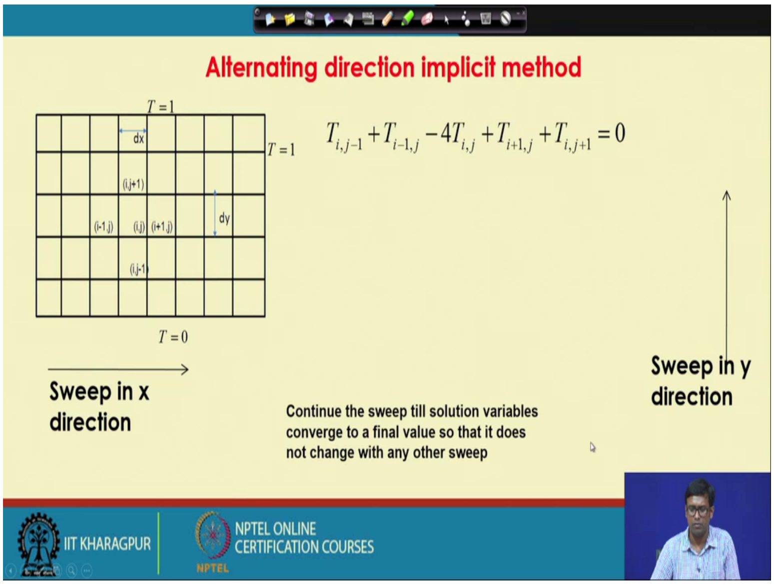
(Refer Slide Time: 27:52)



And similarly, we will do a sweep in y direction. So, we will solve for j is equal to 4 j is equal to 5, and each line this is called a line solver, because each line can be solved using a TDMA. Once this is done, I have some updated value, which are not a final value, because we started with some guess value at certain points, we have some updated value of all the points.

And then I will do a sweep in x-direction. Similarly, I will assume guess value for one particular i one guess value for one particular i here, one particular i here, one particular i here, and do a sweep calculate all the values in the j direction.

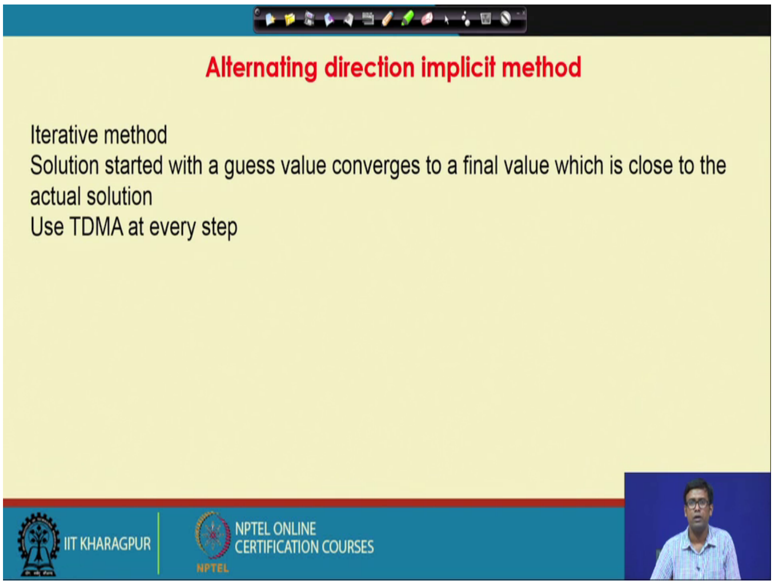
(Refer Slide Time: 28:23)



And then we will do a sweep, similarly value one particular i here, one particular i calculate all the values in j direction, we have sweep go sweep in x direction. And we will alternate this directions, continue the sweep with alternating the direction ones in x direction, and ones in y direction alternating it, until it convert just to a final value when the solutions do not change.

And this calls an this is very useful very quickly we can get solution much quickly than Gauss elimination or LU decomposition method, this is called an alternating direction implicit method. This is also an application of TDMA.

(Refer Slide Time: 29:14)



This is an iterative method; Solution started with a guess value, and convert this to a final value, which is close to the actual solution, and it uses TDMA at FD step. However, this rises two interesting question; what is an iterative method, we will start with the guess value and update the values, and we will do T lid converges to the right solution.

So, iterative method and convergence is very interesting part in matrix solvers, and as said in the introductory lecture that a plethora of matrix solvers are they are based on iterative method and which look into first convergence of the solvers, which we will discuss at much later stage.

Thank you.