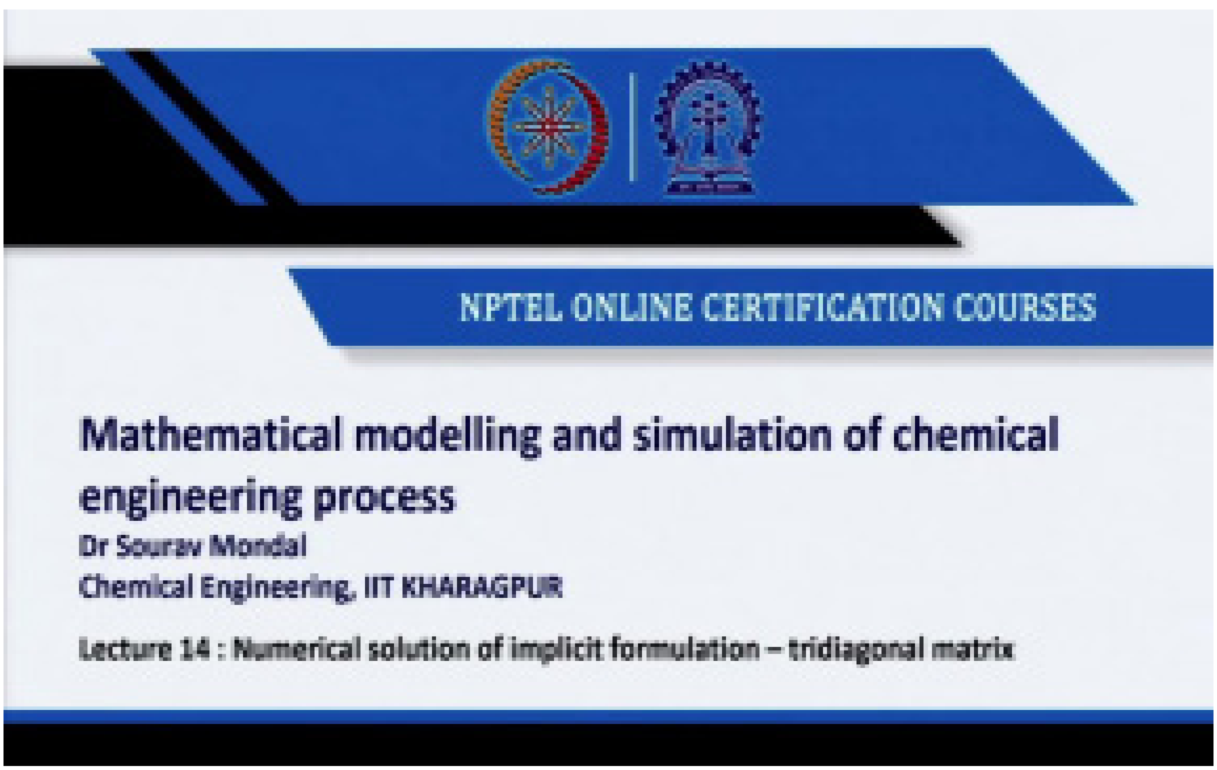
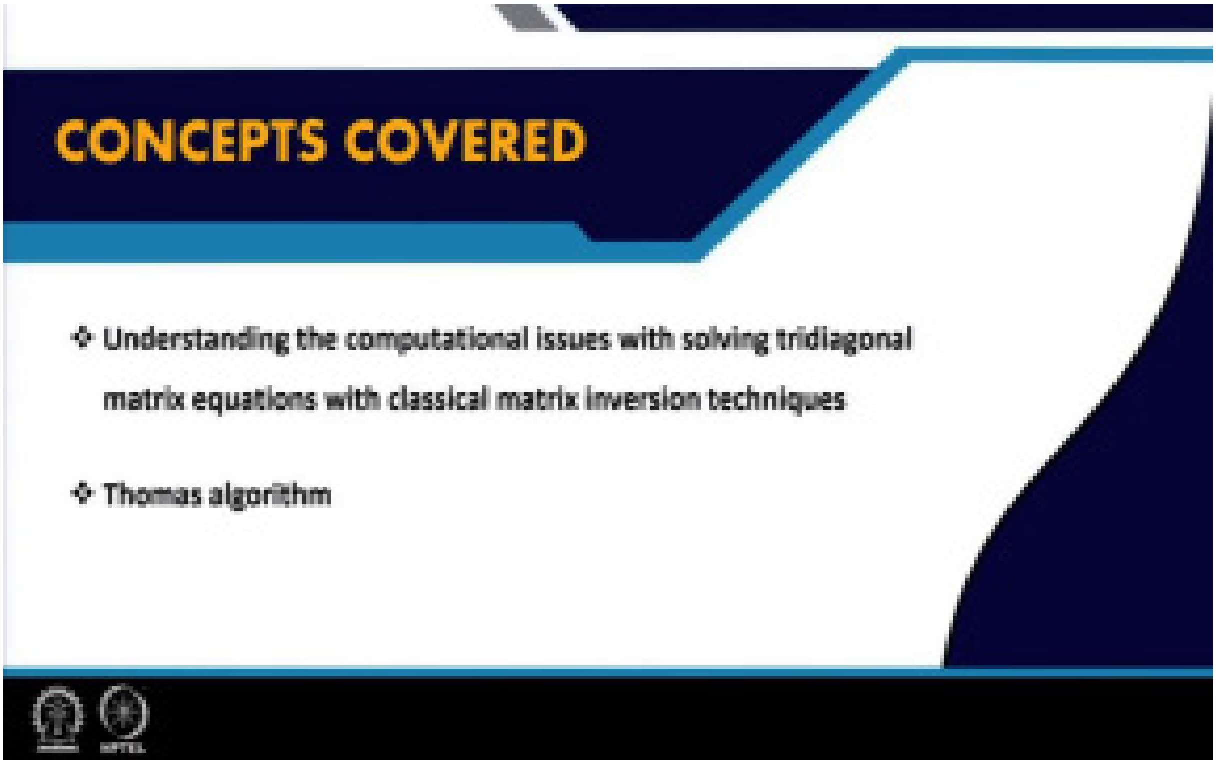
Mathematical Modelling and Simulation of Chemical Engineering Process
Professor Dr. Sourav Mondal
Department of Chemical Engineering
Indian Institute of Technology, Kharagpur
Lecture 14
Numerical Solution of Implicit Formulation-Tridiagonal Matrix

In this lecture, we are going to talk about the implicit formulation technique and the need of a specialized solution strategy or an algorithm to solve the implicit equation essentially these are tri-diagonal matrixes where only three elements in a matrix that is non zero and remaining and 0. So, to go for a full inversion of the matrix like that is how classically we solve using the Gauss Jordan and the Gauss-elimination, LU decomposition all these methods for matrix inversion. Here we need a possibly specialized method or a scheme so that these inversion or the solutions involving tri-diagonal matrix is much computationally faster.

And this has been proven to show that since the number of this matrix operation has reduced tremendously in this case of tri-diagonal matrix the special algorithm known as a Thomas algorithm here is proven to be quite quite faster in solving large systems.

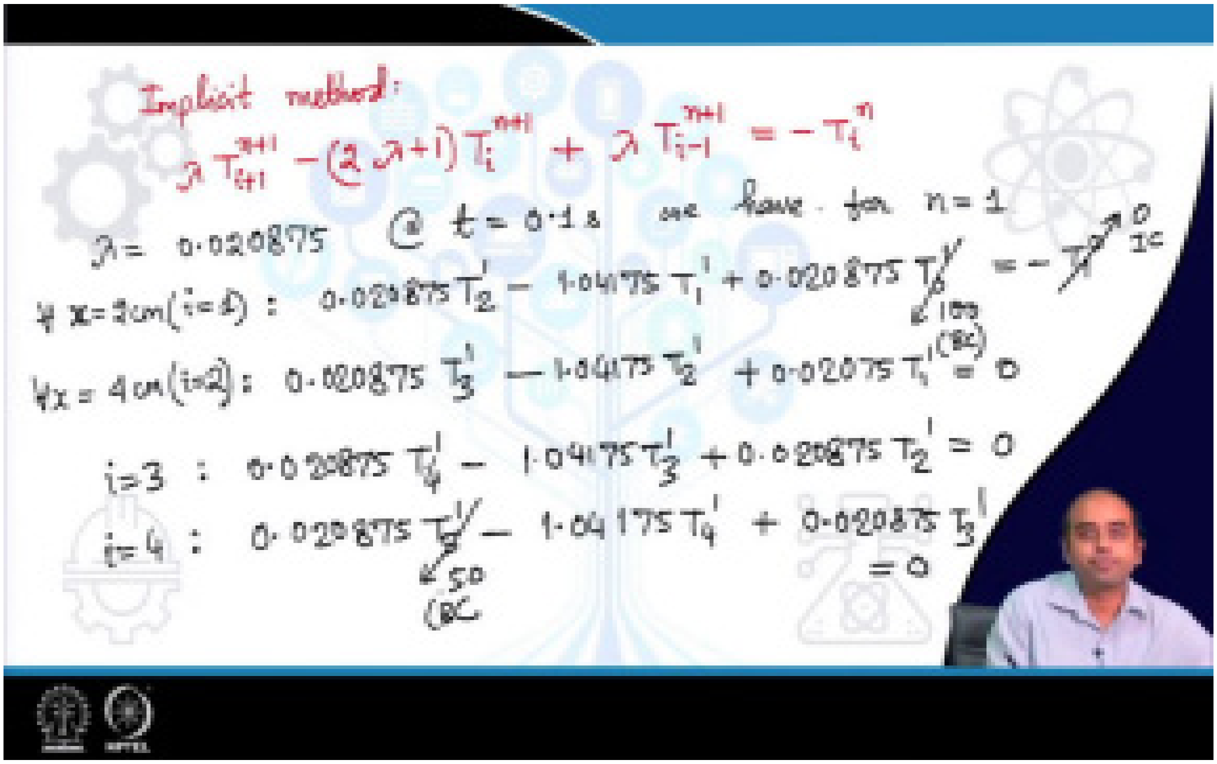
(Refer Slide Time: 01:37)





So, let us let us begin today's class and essentially here we will be talking mostly about the tri diagonal matrix operation and the background of the Thomas algorithm in solving these sorts of equation frameworks.

(Refer Slide Time: 1:57)



Okay. So, now, from the implicit methods, we will just have a start from the point where we did in the last class. So, the implicit method leads to the equation where you have three unknowns in each equation.

So, this is the generalized equation and for different i locations you will be having different equations, but essentially i plus 1 i minus 1 and i all three are unknown, because they are at the current step and essentially this will lead to an equation having three unknowns.

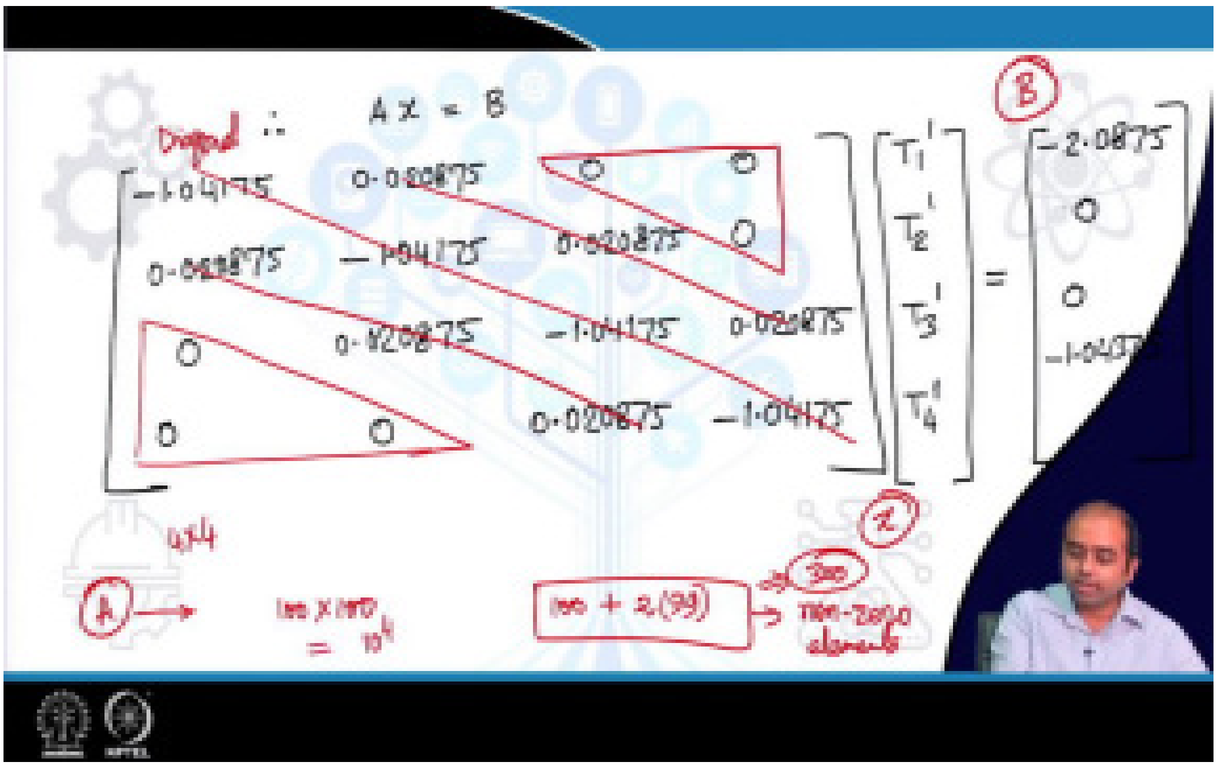
So, let us say we choose lambda the same value that we just used to solve the explicit problem also, I have intentionally chosen them to be all same, so that you can easily compare on solving them. So, we have for x is equal to 2 centimeter that is i is equal to 1 and here we are trying to consider the first time step.

So, in this case what we have is this 0.02875 remember T1 2 is an unknown and T superscript 1 for any interior point is unknown and that is what you see that and please note that this value is 100 because it is a boundary condition and this is 0 because of being an initial condition on the right hand side.

Similarly, for x is equal to 4 centimeter that is i is equal to 2 we can write down the similar equation and for the remaining i is equal to 3 I will also just write for the sake of completeness. So, what do you see here that if I tried to frame the matrix and try to represent this in the form of a x is equal to b where t is the dependent variable and a is the coefficient matrix and b is the matrix corresponding to the right hand side of this equation.

So, each row of that matrix will represent each of these equations are the coefficients corresponding to each one of the t values, is not it? This one is equal to 50 because it is a boundary condition.

(Refer Slide Time: 6:06)



So, with this equation framework, if I wish to write this in something like A T is equal to B sorry not A T let us write in a generalized way, where x is nothing but my temperature. So, it is like this let me write down so I hope I can fill this up properly. So, this is these are the individual coefficients 2T1, T2 and T3 in the first and T4, T5 all in the first equation. So, if you recall in the first equation we had T2 then T1 and nothing else was there.

So, so, this is for T1, this is for T2 and remaining are 0. So, this will be a 4 by 4 matrix similarly, for the second equation it is 0.020875 so, this is T1 this is T2 this is T3 and T4 is 0. Similarly, third equation T1 is 0 this is T2 in the third equation value of T2 then value of T3 in the third equation. Interestingly you see that all the diagonal elements are negative and this is and this is the 4th equation in the 4th equation T1, T2 both are 0 and T4 is non 0 and T5 is this value.

So, this is matrix A so, this is the coefficient matrix A. Next we write down what is the X so, X you can easily understand that it is T1 1, T1 2, T1 3 all at the n is equal to 1 that is the current timestamp and all are unknown. And the B side is minus 2.0875 this is 0 this is 0. So, this is how we frame this equation structure and of course, you can solve this by classical matrix inversion technique. But since you see here that some of the elements are 0 and it is essentially that the elements one after and one before the diagonal element position is non 0. So, this is the this is the diagonal element. So, this is the diagonal.

And you see that one the next value from the next diagonal element and one before the diagonal line is non-zero and rest everything is 0. So, in a 4 by 4 matrix, you can easily understand that there are these are the the the sort of the side elements which are 0 and essentially in a 4 by 4 matrix so, this is a 4 by 4 matrix, because delta x was 2 and we discretize the space domain into four parts.

So, this is a 4 by 4 matrix out of which total number of elements were 16 and out of them, we had the almost 3 plus 3, 6 elements to be 0, so, which may not be too much, but let us say if you discretize the space into 100 points if you discretized the space into 100 points, these matrix these A matrix will be having 100 cross 100 number of elements and from there you will easily you can easily calculate it out that approximately it would be the diagonal elements which would be non 0 and there is one set of the elements just after the diagonal and before the diagonal which is non 0.

So, approximately we will see that 100 plus something like 2 into I think 99 these are the number of non 0 elements. So, these are number of non 0 elements or key elements which needs to be tracked. So, approximately this is around what you see 300 and total number of elements is 10 to the power 4 or 10,000.

So, out of 1000s only 300 number of elements is non 0 and rest are all 0. So, any matrix operation that you want to do for example, when you try to find this inversion using this Gauss elimination or any of these Gauss or any of the techniques, you need to do element wise operation on all the points around all locations, and there you realize that since out of 10,000 only 300 are non 0 elements and rest 9700 around that number of elements are 0. So, it suggests that there (is) there are a lot of redundant computations that is needed because of the presence of these 0 elements. So, you will be essentially doing a lot of redundant calculations and that will make your computation slower. And this is the reason why a special technique is used here to save the computational time time and to make the calculation faster.

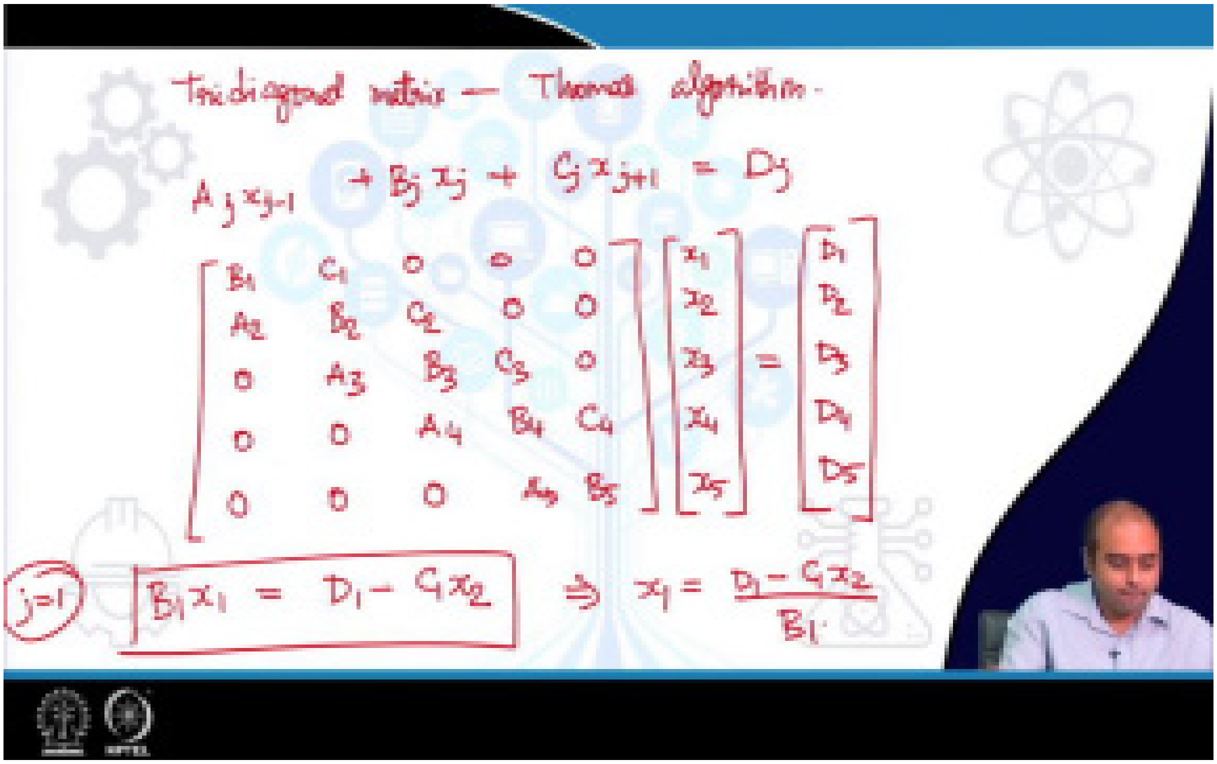
And you can easily realize that when we move this from 100 leads to 1000 grids, the situation changes drastically. And if you move to further higher number of grids, which is the case in most the sort of finite difference calculation and in involving large number of grid points, you are likely to get more number of this grid spacings and grid elements, you can easily understand the scale of the problem and the number of redundant calculation there is.

So, that is where you start to see the difference in the computation time. So, you may not see a realizable or physical difference or a realizable difference between doing 16 calculations and reducing and doing 10 calculations. So, in this case 4 by 4 matrix as opposed to doing 16 calculations for the classical method of matrix inversion, if you use these special techniques of Thomas algorithm, you will be doing 10 operations.

So, that difference in the computational time will not be observed, but when you change the scale of the problem to more number of grid points, this difference in the computational time will be really affecting your solution and please note that this is for each time instant. So, this will be also having a compounded effect when you go for several time steps to solve the problem. So, there you will see that each time step is really slowing down your system and then when you try to move much forward in time subsequent and the total time computational time will be significantly lower.

So, let us understand this Thomas algorithm. So, these metrics or this sort of matrix where only the diagonal elements and (one after) one before and one after diagonal elements are non this is called a tri-diagonal matrix.

(Refer Slide Time: 14:38)



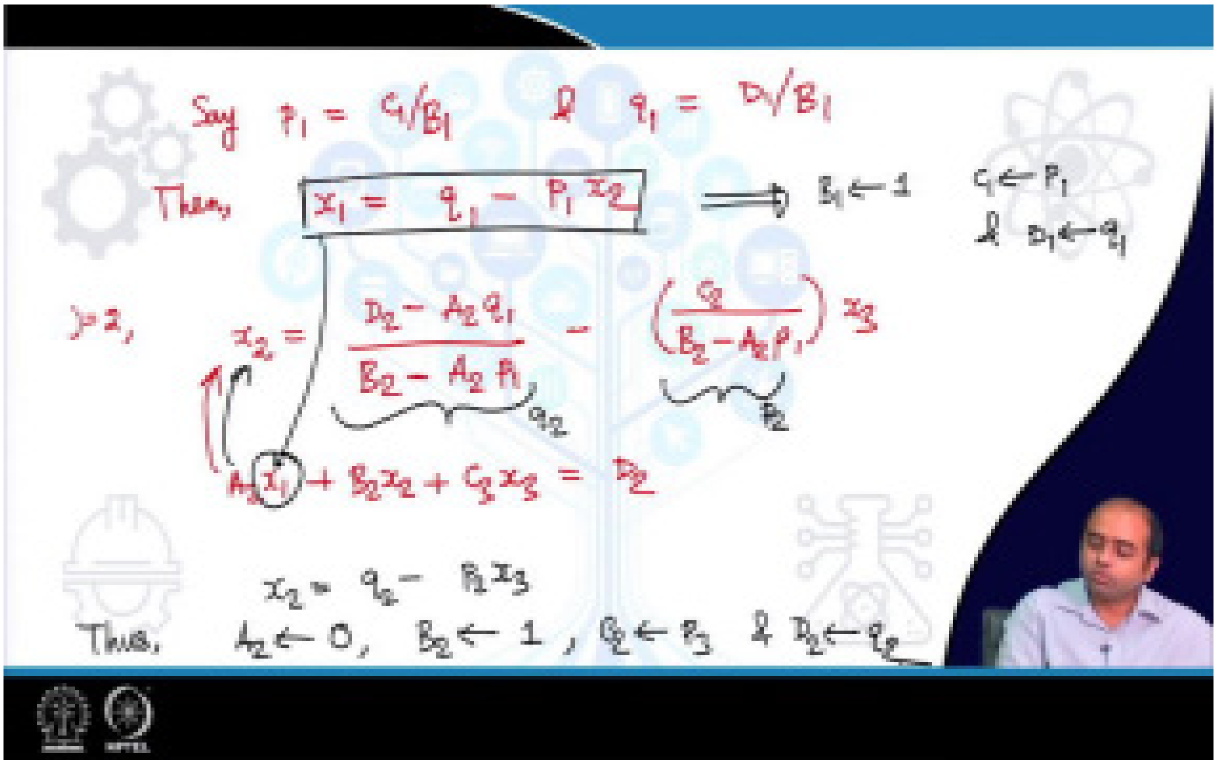
So, this tri diagonal matrix is generally it is not generally it is always actually it is always solved using Thomas algorithm for computational purposes. And what is this Thomas algorithm? Let us learn about this today. So, let us I write the coefficients of the different components in the matrix as Aij. So, this is the element corresponding to so, for simplicity I write it in terms of i and j and now let us remove the i component there is no need of i component here.

So, you can easily understand that A j are the components corresponding to the element before the diagonal and B is the diagonal elements and C is the element after the diagonal let us D is the things on the right hand side. So, let us take this small example of four elements in this case or five elements, let us say.

So, first one of course, we will start with B1, C1 because you do not have any A1 component here the starting one then we have A2, B2, C2, 0, 0, A1 is generally part of the boundary conditions similarly, C5 would also be part of the boundary condition so, it would be C3 this is how the general structure looks like, then we have 4; A4, B4, C4, and then we have the 5; B5 and let us write down these as x1, x2, x3, x4, x5 and this is D,1 D2, D3, D4, D5.

Now, let us look into the first term area or the first row. So, if you do the matrix multiplication for the first row, what equation do you get B1 x1 the same equation that you actually used to present this matrix formulation is not it. So, this was the equation from the first row or for j is equal to 1 this is what you get. So, from here, if you see I can rewrite my x1 as x1 is equal to D1 minus C1 x2 just rearranging by B1.

(Refer Slide Time: 18:15)



Now, say p1 is equal to C1 by B1 and we (have) just doing some redefinitions D1 by B1 then I can write my x1 as q1 minus p1 x2 I think all of you are following where q1 is C1 D1 by B1 and q2 P1 is C1 B1 just from this portion of analysis. So, I can write x1 in terms of x2 with just two coefficients q1 and p1.

Now, for stage for i j is equal to 2 similarly, I can also write you can also find out that for this j 2x2 you can write something just to find out that you can write something like this in terms of q1 and p1, I can always write down and how do I get it? It is from the same idea that so, how do I get it is that if I write the second equation, I will be writing A2x1 (is not it) plus B2x2 plus C3x3 is equal to D2. Is not it?

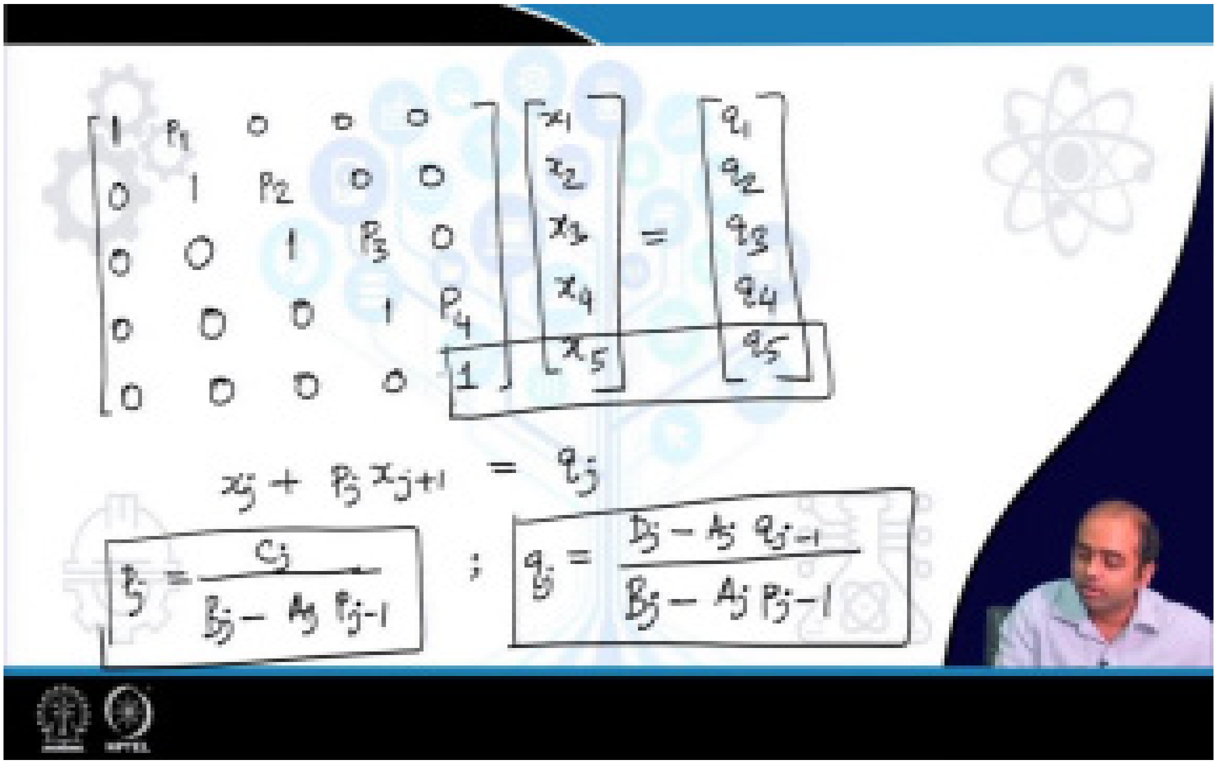
Now, here x1 I substitute x1, I substitute it from this equation from this equation you substitute your x1 and now your entire equation will contain only x2 and x3 this is what is rearranged here,

this is what is actually rearranged written down here here we can write this x2 in terms of x3. And like this we can write for any any generalized this p and j values is not it, but realize one thing now that now (the) from the previous equation where we had three unknowns x1, x2 and
x3 or any xj minus 1 xj and xj plus 1 three unknowns in each equation with this transformation, I can always get two unknowns in 1 equation.

So, in this equation only there are 2 unknowns. Now, in the same tri-diagonal matrix, if I want to reuse that the framework of the same tri-diagonal matrix I can consider like an analogous version where B1 is set to 1 and my C1 is replaced by p1 and D1 is replaced by q1. So, in the same formulation, if you recall the structure of this here in this now, I am trying to replace my B as 1 C1 as p1 and D1 as q1 and that will get me this this equation from here that x1 is going to q1 minus p1x2 is not it. Similarly, for this equation also, I can call this value as q2 and this value as p2 so, I can write x2 is equal to q2 minus p2x3.

So, in the same tri-diagonal matrix formulation, I will be setting my A2 here as 0, B2 as 1 because B2 stands for the coefficient of x2 and A2 stands for the coefficient of x1 and since, there is no x1 in this equation A2 will stand to 0 and C2 is the coefficient of x 3. So, I will write C2 as p3 and D2 is the coefficient on the right hand side. So, D2 will be replaced by q2.

(Refer Slide Time: 22:42)

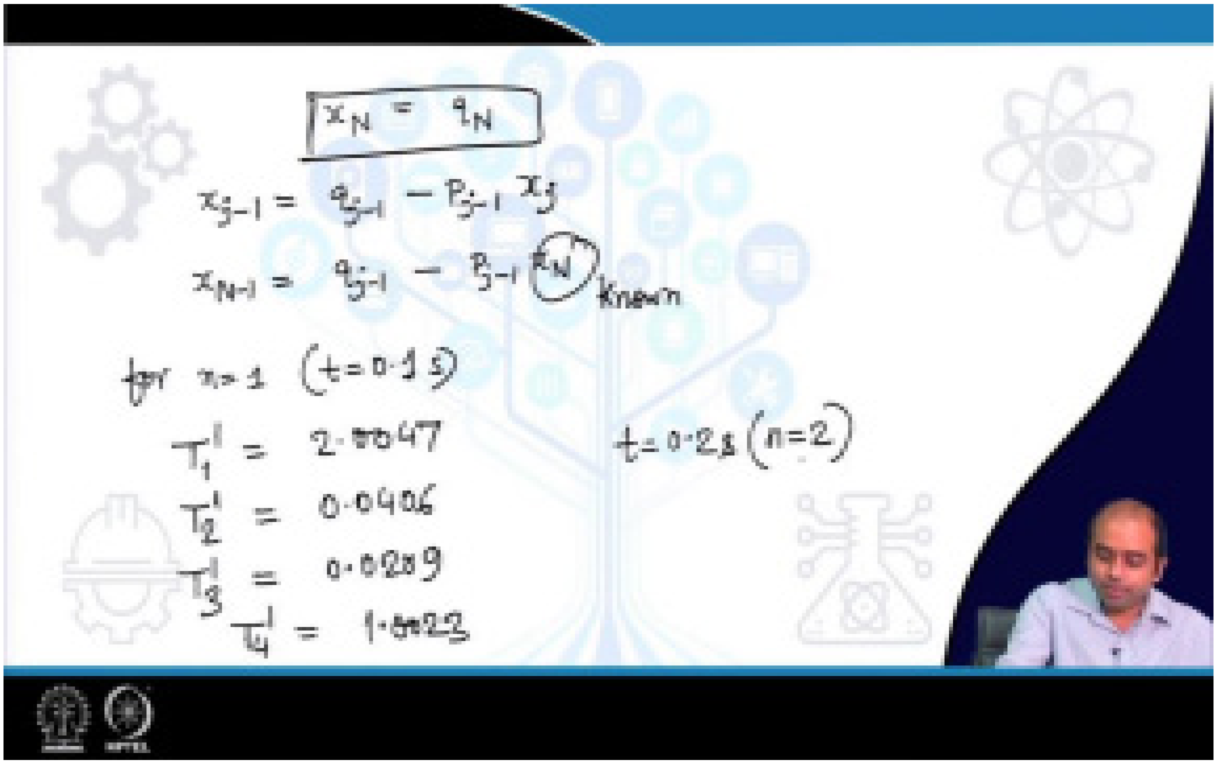


So, in the same in the same structure sorry I can fit my new set of p and the q. So, how does the new matrix formulation look like? So, here now, instead all the B elements would be equal to 1 and the one after that will are written as p1 remaining are of course, of course 0. So, all the elements are 0 here in this new framework, B elements are 1 and then we have p2.

So, this is the A3 which is 0 then B3 is set to 1 and C3 is p3, same goes with A4 and sorry this would be (()) (23:28) this is so this 0, 1, p4 this is B4 and C4 and finally, the last column would be only B5 because there is no C5 and here of course, we have x1, x2 these are same x3, x4 sorry x4 and this is x5 and these are q1, q2, q3, q4 and q5. So, what is the generalized structure we are following here that xj plus pjxj plus 1 is equal to qj and what is the generalized structure of pj and qj. So, pp is Cj by and this is also something you can work out Bj minus Aj pj minus 1.

So, pj depends on j (min) the value of p at j minus 1 and similarly qj also needs the value of pj as well as q sorry this is Aj qj minus 1 by Bj Aj pj minus 1. So, these are the generalized form of pj and qj and this depends on their previous values previous values that is how this p is formulated. Now, what is the immediate consequence of this equation? I think you can easily realize that immediately what you will be getting here is that this last row so, you do a backward substitution.

(Refer Slide Time: 25:48)



Now, so, the immediate consequences that xn is equal to qn so, you do not calculate from the top but from the end last row, because the last row you can easily see that my x5 is equal to q5 which means that this is the last and from the backward you try to do a recursive substitution a backward substitution recursively how from here we calculate like xj minus 1 is equal to qj minus 1 minus pj minus 1 xj and if you start from the backward so, if I try to if you know if I know my xn then I can calculate xn minus 1 which will be dependent on this value of xn and this is known from the last row.

So, like that you can proceed all the way up to x1 which will need the value of x2 and that is what you do I mean be getting before. So, if you do this calculation, if you use this matrix Thomas algorithm for n is equal to that that problem which we just framed this tri-diagonal matrix here if you use this Thomas algorithm and try to get a solution for n is equal to 1 that is t is equal to 0.1 second you will be getting T1 1 is equal to these values of course, these values are slightly different from the explicit calculation and that is understandable, because these are not the same techniques that we are applying.

So, similarly, I think I would suggest you to do it yourself what happens for T is equal to 0.2 second that is n is equal to 2 now, please note that in this case, you cannot have the same framework and the right hand side will be changed in this case the right hand side will be changing for n is equal to 2 value because that has the values based at n equal to 1.

So, the right hand side will change and again you do the same tri-diagonal matrix formulation and you get (the) use this Thomas algorithm and you can solve this problem using this implicit method. So, any implicit scheme that you see or that is used ever is based or always at the heart this Thomas algorithm is there and this significantly speeds up the solution.

So, I think a good exercise for you would be trying to work out that changing the number of grid points like once you try to have your own program for this calculation, using this Thomas algorithm, you can change the number of grid points it has from 4 you try to increase it to 50 100 or more and then try to see the difference when you use the Thomas algorithm and when you try to use the classical this matrix inversion techniques.

So, I hope all of you got a flavour of the numerical scheme that is used to solve partial differential equation, which is based on the explicit and implicit formulation, their differences as well as their utilities and the computational details behind each of these schemes. I hope all of you find this lecture to be useful.