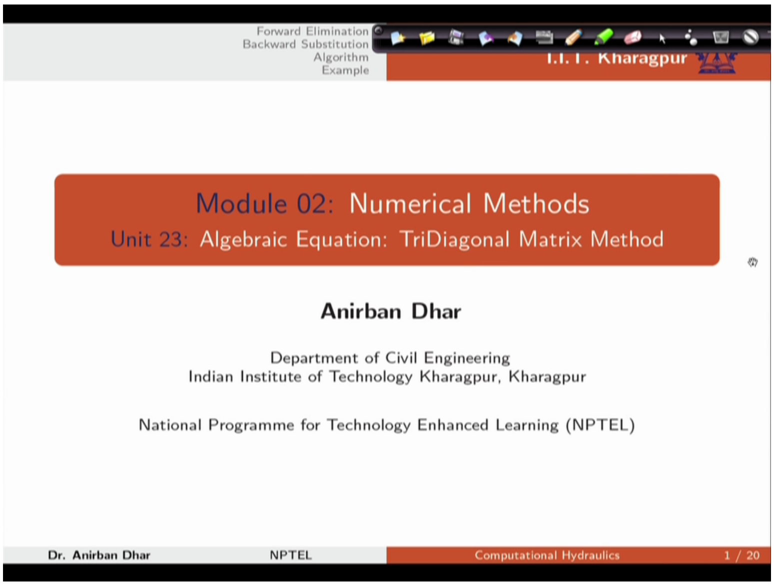
Computational Hydraulics
ProfessorAnirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Lecture 27
Algebraic Equation: Tri-Diagonal Matrix Method

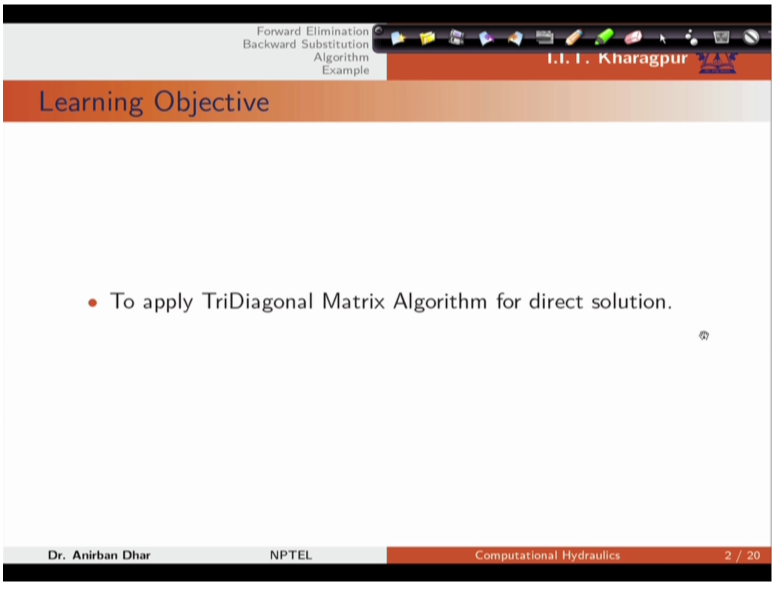
Welcome to this lecture number 27 of the course computational hydraulics.We are in module 2, numerical methods. In this particular unitnumber 23we will be discussing tridiagonal matrix method. This comes under algebraic equations.

(Refer Slide Time 00:28)



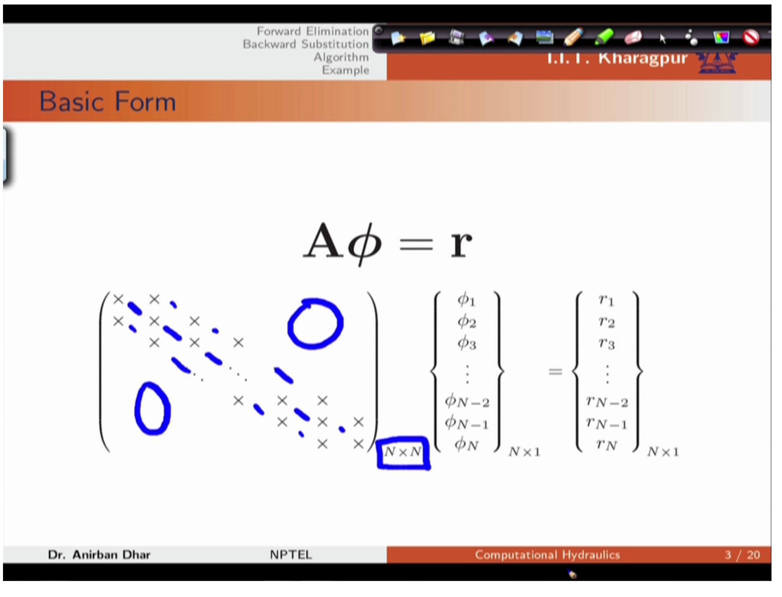
Now what is the learning objective for this particular unit? To apply tridiagonal matrix algorithm for direct solution of matrixes. Specificallywe will be discussing special kind of matrix that is tridiagonal banded matrix for thisparticular lecture class.

(Refer Slide Time 01:13)



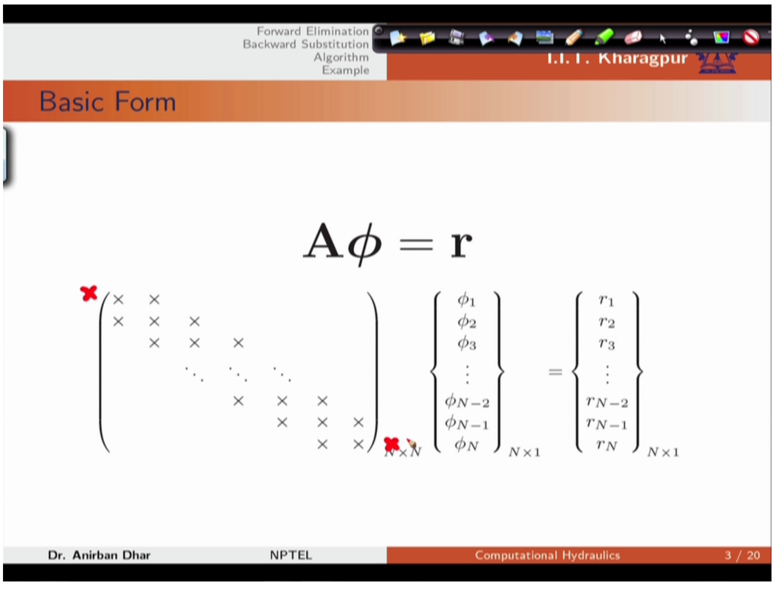
We have a basic formof tridiagonal matrix. In this case, tridiagonal structure we have one diagonal term and two off diagonals. Off diagonals are adjacent ones. So other terms are with zero values. Now if we have a large matrix, large N by N matrix and we need to store this extra zero values where we know thatif we multiply phiwith that zero value, obviously it will not add any value to our calculation process.

(Refer Slide Time 02:11)



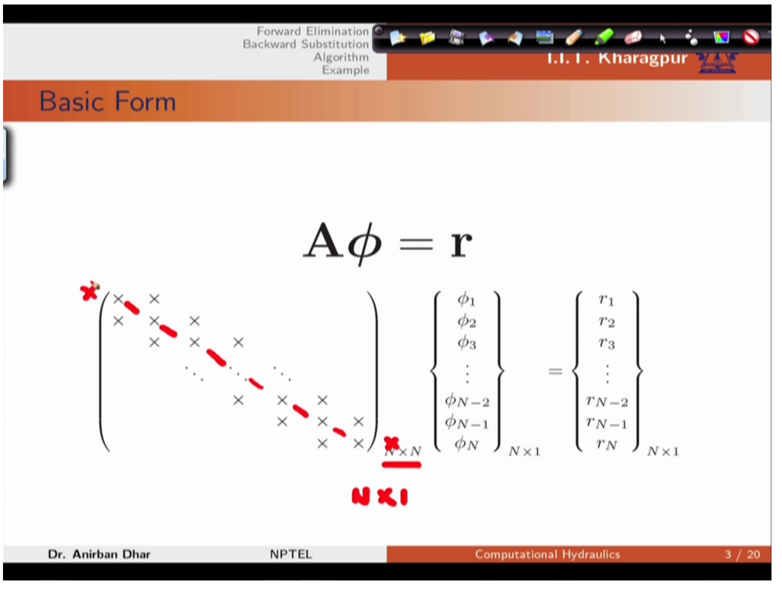
So what we can do, we can extend this up to one extra point. And on lower side also we will consider another point here.

(Refer Slide Time 02:32)



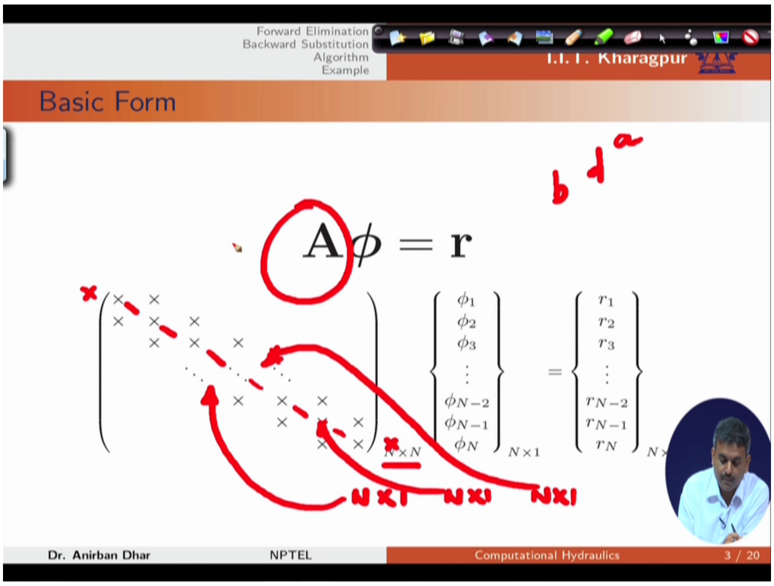
So if it is N by N total matrix. Now we can divide this A matrix into three differentcolumn vectors. So how is this possible?In diagonal term we will have N values. So let us say we have N cross 1 column vector. Another one is for lower one. Although we know that this value is dummy value.

(Refer Slide Time 03:19)



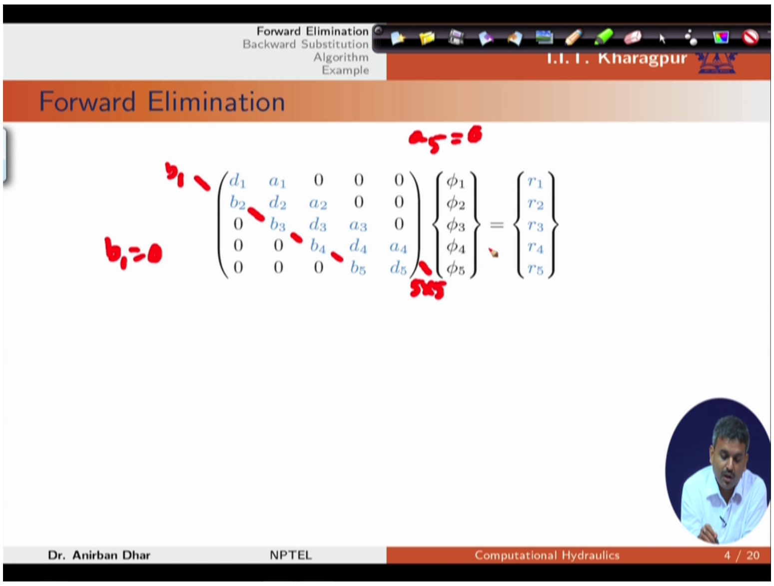
To keep the size consistent with the diagonal term we will store this one extra value here, N cross N, N cross 1. So maybe if we consider this for lower one, this is for middle one and this is for upper one. So let us say we are defining it with a diagonal term and below diagonal term and above diagonal term. And now we need to store this phi and r as usual because these are required forour calculation process. Only we will concentrate on the matrix A and we will try to reduce the storage requirements for matrix A in this case.

(Refer Slide Time 04:29)



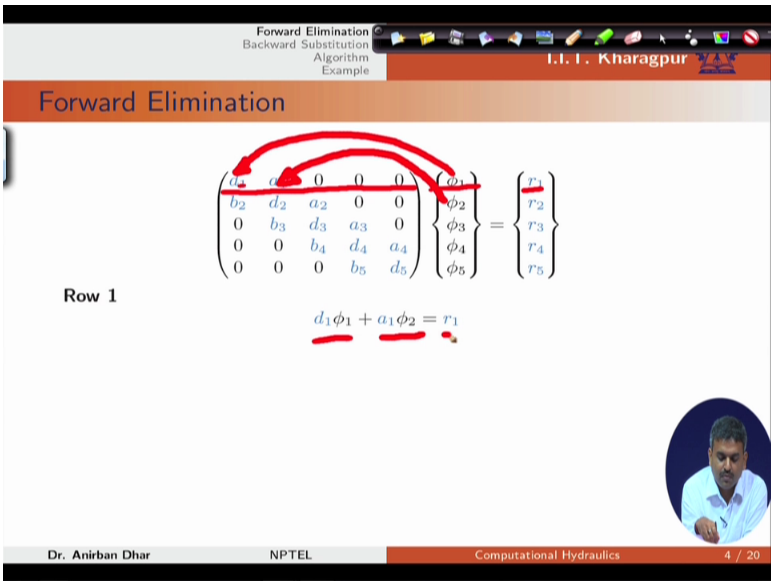
Now like our Gauss eliminationwe will have this 5 by 5 matrix, sum example one. This is below term. So essentially if we write this b1, b1 is zero. At the same time if we consider a5, obviously a5 will be zero in our calculation.But d1 to d5 will be there for our calculation process. So phi, these are as usual.

(Refer Slide Time 05:29)



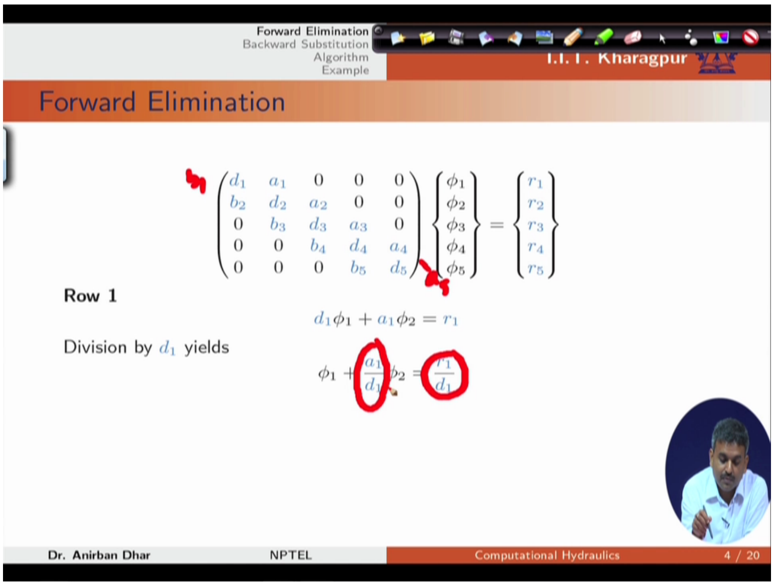
Now let us say that we are starting our calculation with row 1.Row 1 what is required? We have d1, phi1, d1 multiplied with phi1, a1 multiplied with phi2 and r1 is there. So with this if we write d1 phi1 plus a1 phi2equals to r1.

(Refer Slide Time 06:10)



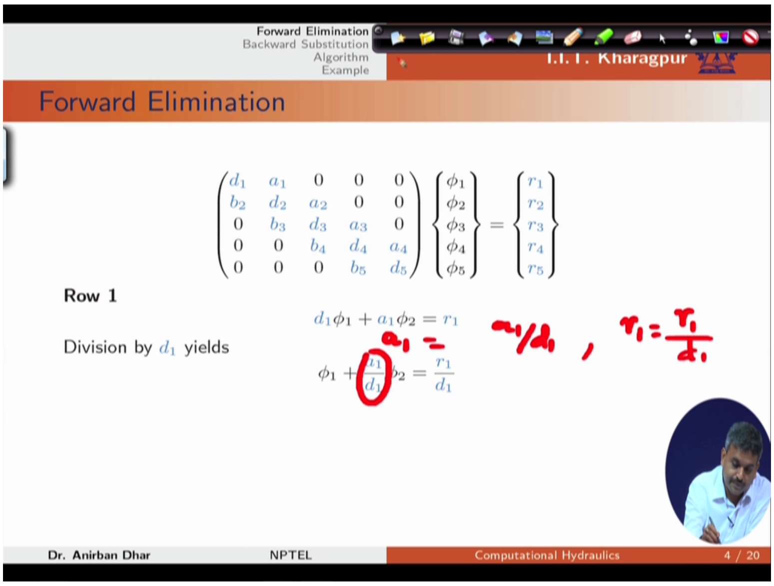
Then we can straight way divide this d1, this is phi1 plus a1 by d1 and r1 by d1. These are some initial values. But interesting thing is that we are using the storage structure for b1 to b5, d1 to d5 and a1 to a5. So in first case we can define itwith a new variable.

(Refer Slide Time 07:13)



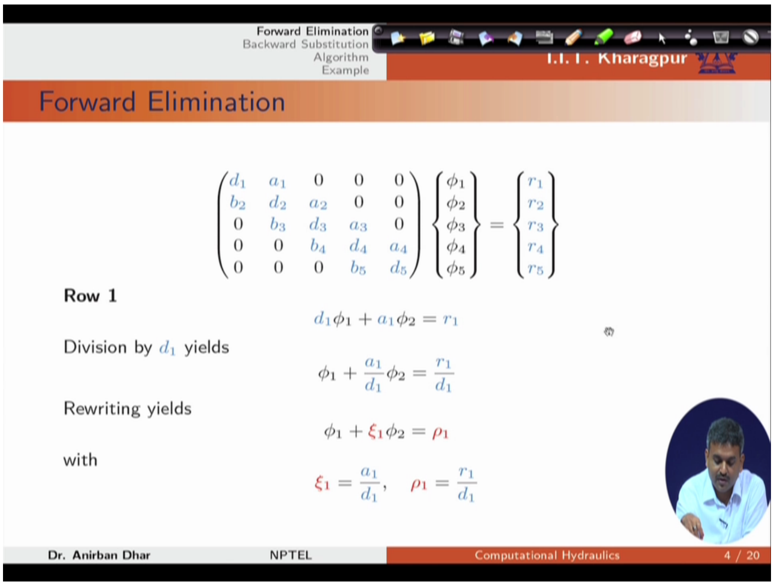
Otherwise we can store a1, this equals to a1. So a1 equals to a1 divided by d1 and r1equals to r1 divided by d1.

(Refer Slide Time 07:48)



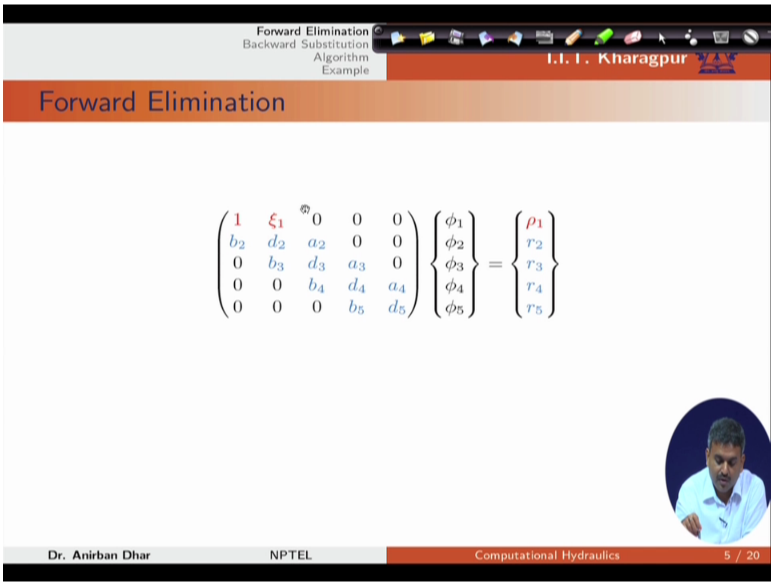
Now in this case rewriting we can see that if we define new variables xi1 and rho1, we can rewrite our firstrow.

(Refer Slide Time 08:04)



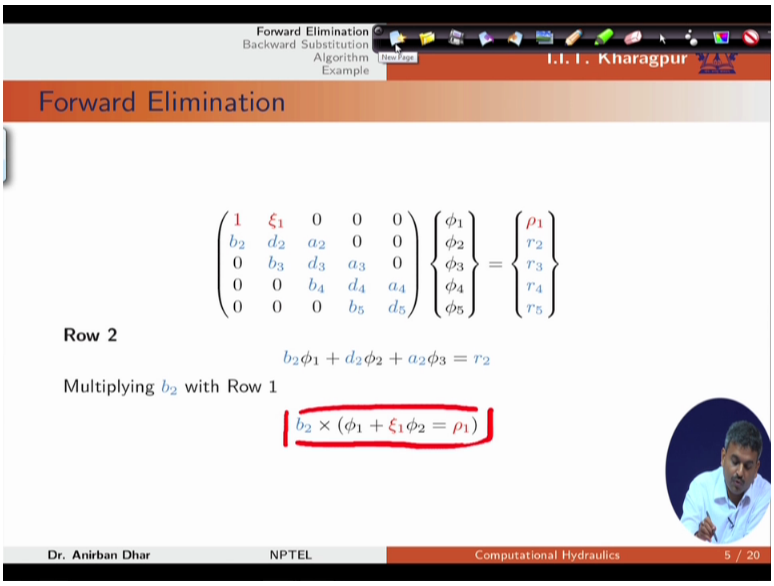
So this thing we have changed. So we have converted our diagonal termin the unity and we have changed our off diagonal term. Obviously we cannot change b, we have changed only a and rho.

(Refer Slide Time 08:25)



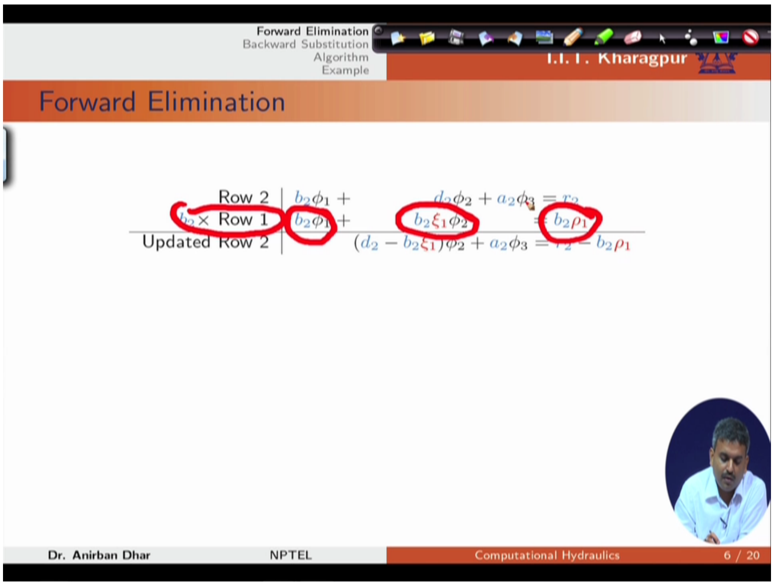
Now in this case if we consider rho by multiplying b2 with row 1. Now remember that in Gauss elimination we have used unchanged or undisturbed row1 in our calculation process. But in this tridiagonal algorithm we are using same forward elimination, but we are multiplying this quantity with disturbed one because we have changed our first row. So we are multiplying row 1 with b2.

(Refer Slide Time 09:27)



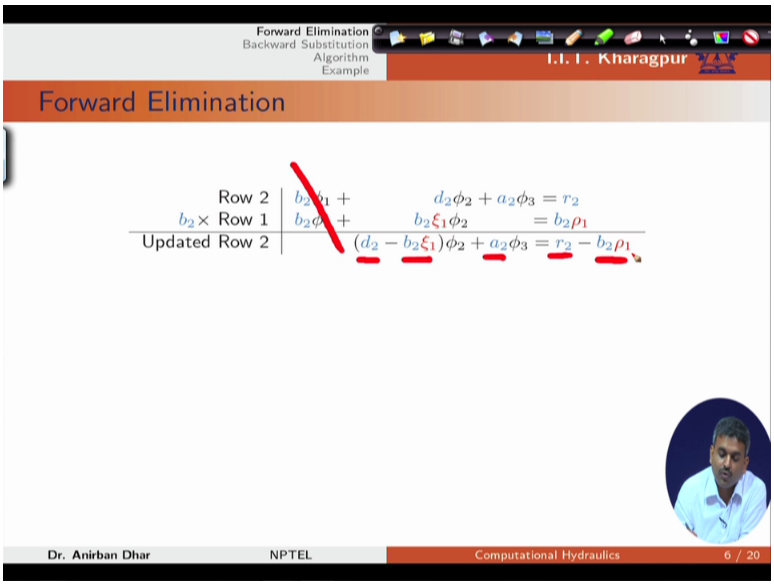
And again we can write b2 into phi1. This is d2 phi2, a2 phi3, r2. And this is our b2 into row 1. This is b2 because diagonal term is 1. So no coefficients are there.So simply we are multiplying b2. And b2 into xi1, this quantity is there as coefficient of phi2, equals to b2 into rho1. Now b2 into rho1, this is important term.

(Refer Slide Time 10:15)



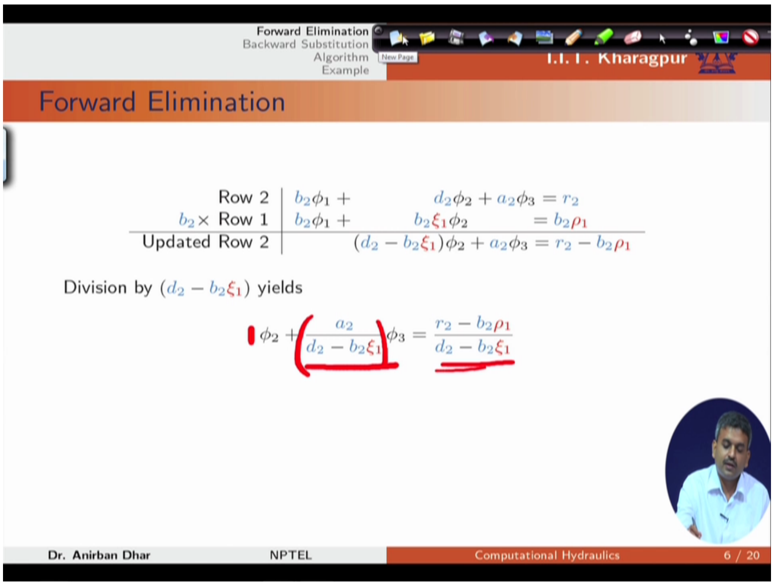
Now if we subtract this, we can cancel this part. So only thing that will be left is d2, b2 into xi1, a2, r2 and b2 rho1.

(Refer Slide Time 10:35)



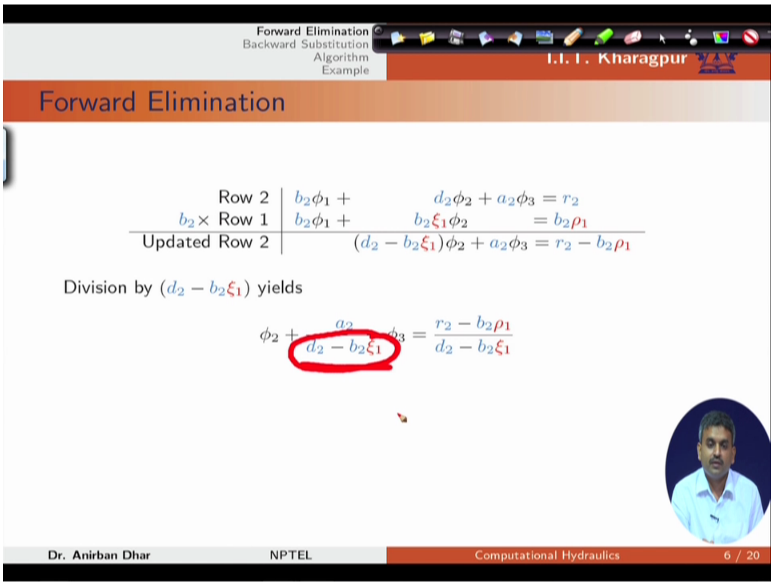
Now in this case if we divide it by d2 minus b2 into xi1 and right hand side also with the same quantity, we can get the transformed form of our row 2. Our objective in this is that to convert the diagonal term with the value of 1 as coefficient. And we are storing this thing.

(Refer Slide Time 11:31)



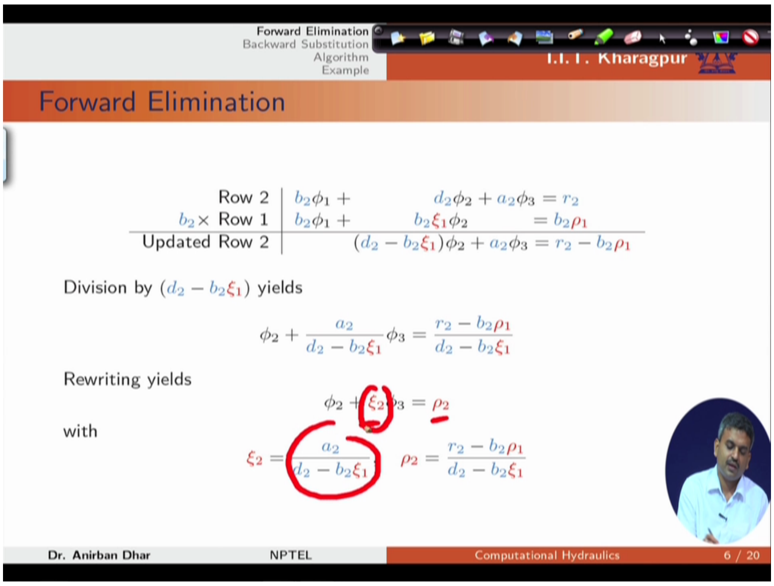
Please remember that whenever we are dividing any quantity, that d2 minus b2 xi1, this should not be too small. Otherwisethis may create numerical error if it istoo close tozero value. So it will introduce error within the system.

(Refer Slide Time 11:57)



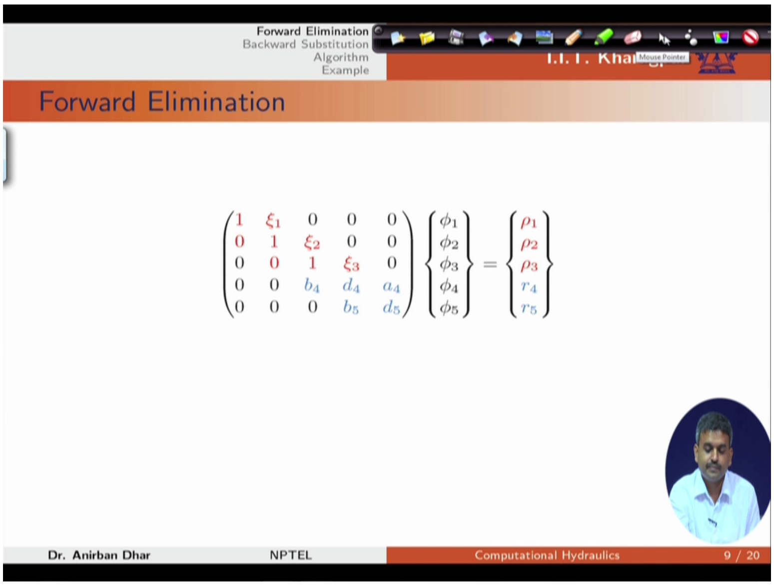
Rewriting, if we write it in terms of xi2 and rho2, we can write the same thing but in other form. But during implementationwe are not going to utilize this xi or rho, because our objective is to reduce the storage structure. So we will utilize our A vector for storage of xi2 directly, this quantity.

(Refer Slide Time 12:46)



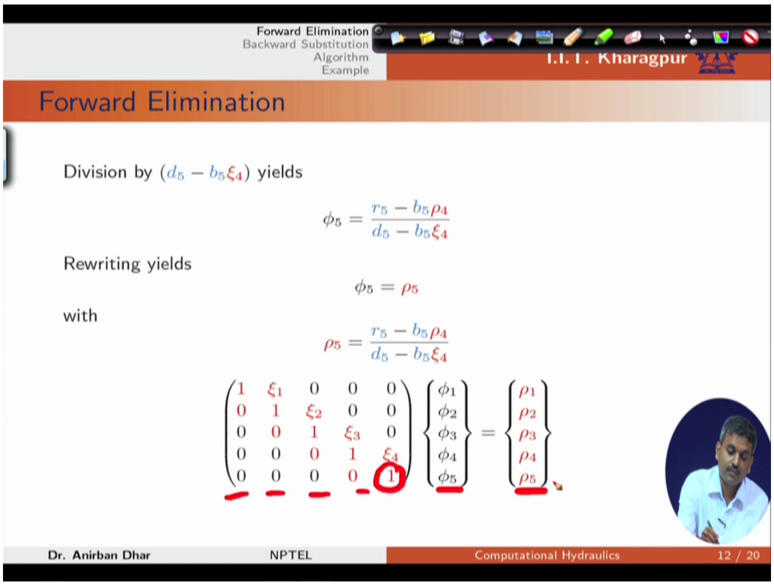
And for r vector we can store this rho2 directly. So similarly if we reduce our row 3, we can repeat the same thing here and we can reduce our row 3. Obviously you can see that unchanged coefficients are represented with blue color and change coefficients or changed vectors are represented with red color here. So up to row 3 we have converted.

(Refer Slide Time 13:35)



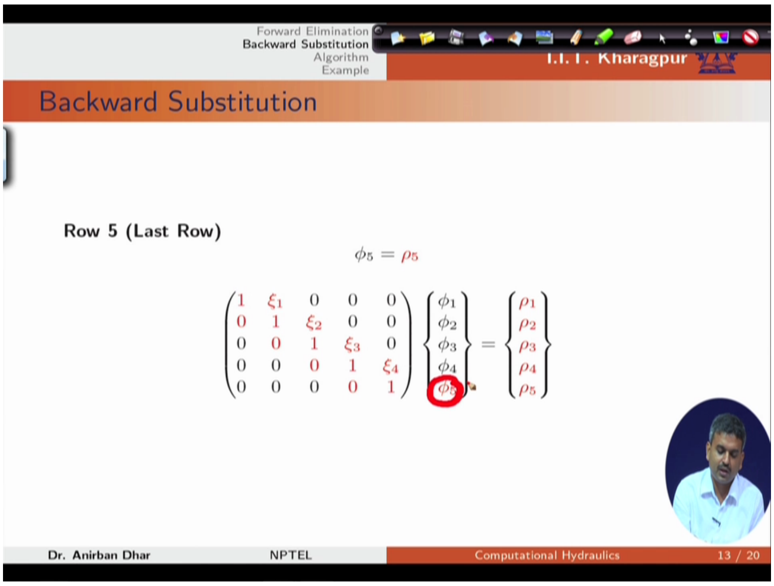
Now row 4 and row 5,same process will convert thistotal thing into upper matrix.I will not say this is upper triangular because we are getting only diagonal term and upper or above diagonal term. So now we can utilize this form for our backward substitution. Because at the last row it is clear that the coefficient is 1. It is multiplied with phi5 only, other coefficients are zero. So obviously this phi5 equals to row 5.

(Refer Slide Time 14:37)



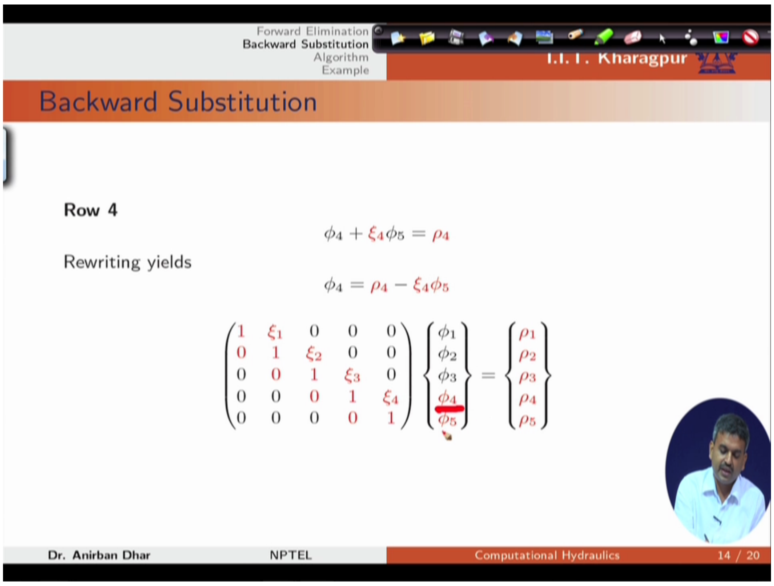
Now in this process we are starting from last row. So from last row if we start,after first calculation we have got the value of phi5. So I have changed the color of phi5.

(Refer Slide Time 15:04)



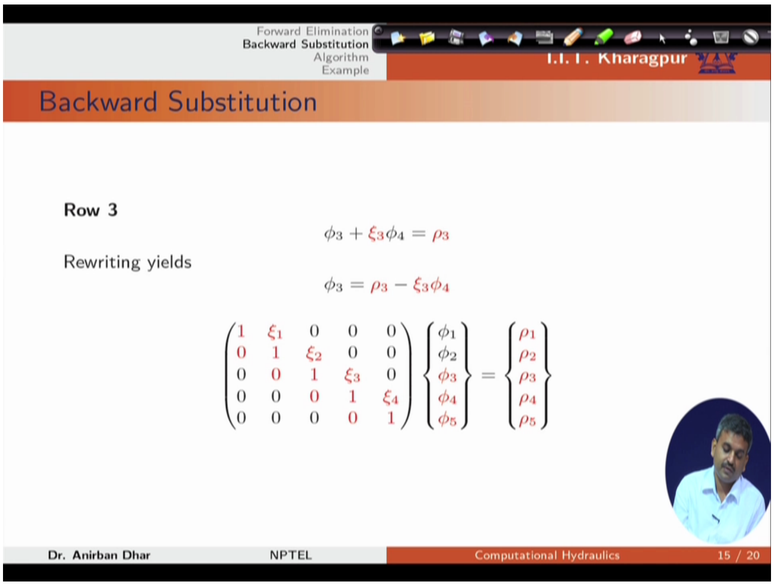
Now in the next step row 4, we will get the value of phi4. So step wise we are getting different values.

(Refer Slide Time 15:17)



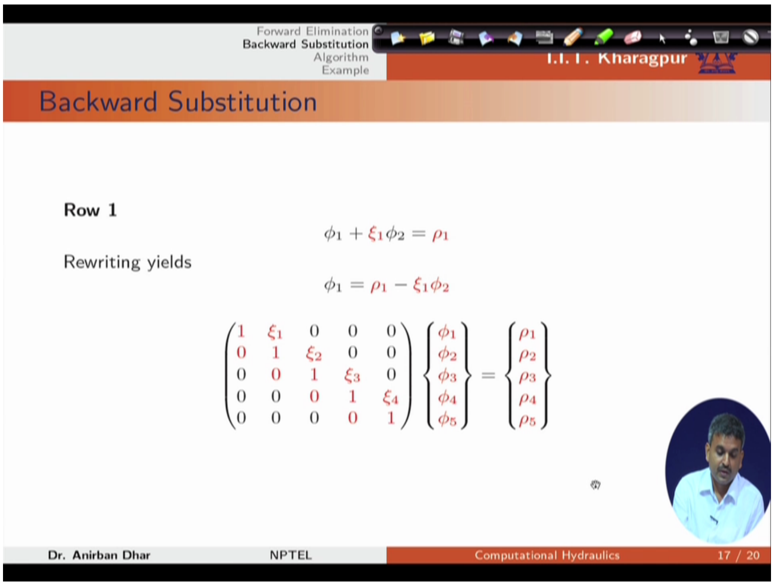
Now in this case there is no change in a and r vector. So although this A matrix is not a complete matrix, we have reduced it into diagonal and off diagonal level column vectors. We can utilize it directly for calculation.

(Refer Slide Time 15:52)



So if we start with row and finallywe will get this phi1, phi2, phi3, phi4 and phi5 values.

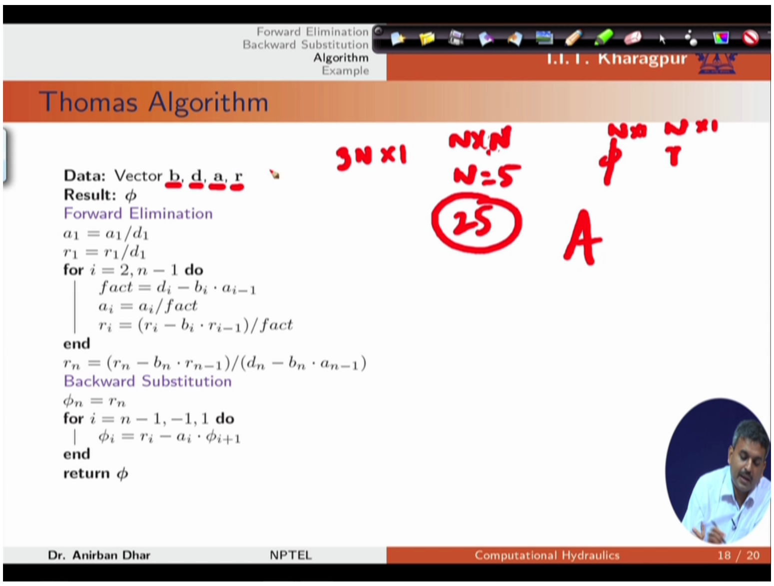
(Refer Slide Time 16:06)



Now we need to implement this in algorithm format. So first what is input that is required? We need that below diagonal, this is diagonal and above diagonal vectors. And r is another right hand side vector. Now the size of these vectors, this is N or 3N.3N for b, d, a and r is as usual we are utilizing this N for phi andN for r. These are column vectors, N cross 1, N cross 1. So 3N cross 1 that is required storage structure. So instead of N into 1 N into N into our N.

That means if we have N equals to 5 we need to store 25 elements of our A matrix. But instead of that we are storing only below diagonal, diagonal and above diagonal values or coefficient values in a column vector format. So these are inputs.

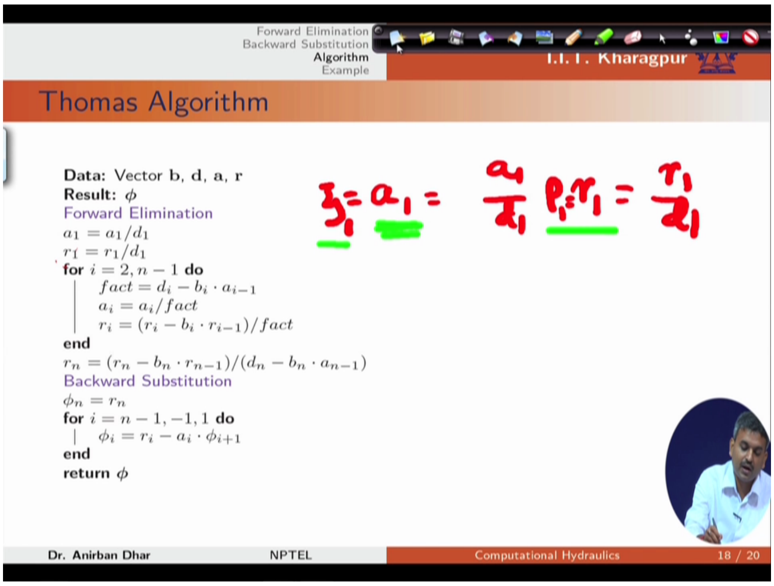
(Refer Slide Time 17:46)



Now output or result is 5. So first is forward elimination. As I have told earlier that we are not utilizing xi and rho as our separatevectors.Because that will require more storage. So we are using a1 and r1 only. So a1 equals to a1 divided by d1, r1 equals to r1 divided by d1. This means that in the first step when we have converted this into xi1 and this into rho1. Actually this is not a required.

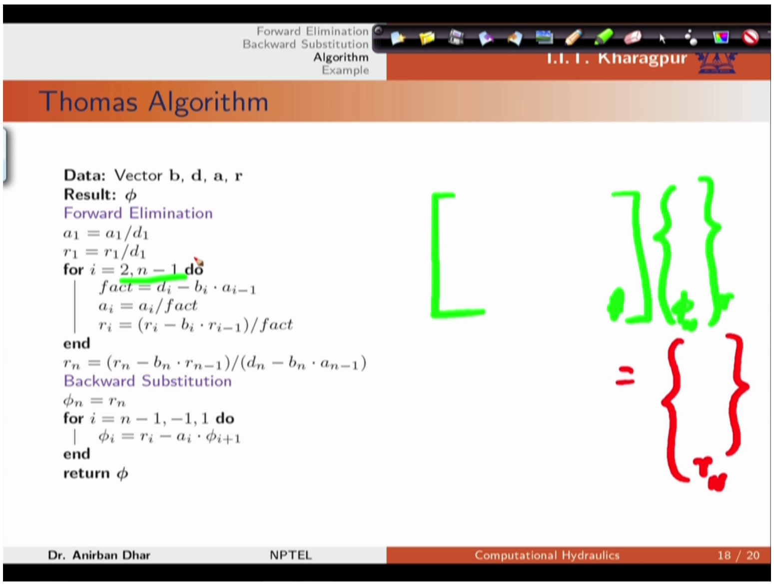
We can store the same value in a, because value is further not required for our calculation. In case of Gauss elimination we have seen that we need undisturbed first level for our calculation process. Here it is not required.

(Refer Slide Time 19:24)



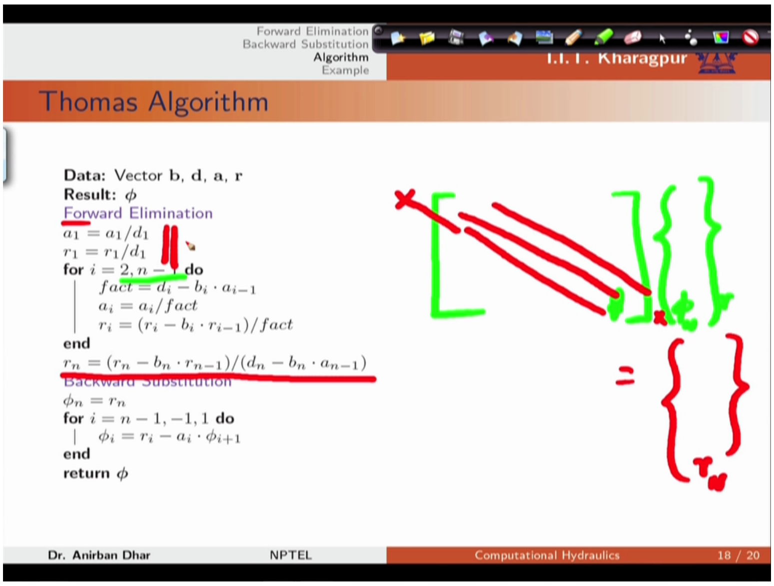
Nowwe are starting from 2 to n minus 1. Why n minus 1? Because for nth row we will have only our r term. Because diagonal term will be 1, so r term. Andthis is phi n. So phi n equals to rn. So in this processwe are calculating up to our ai equals to ai divided by factor. It is for 2 to n minus 1. So only for interior values.

(Refer Slide Time 20:27)



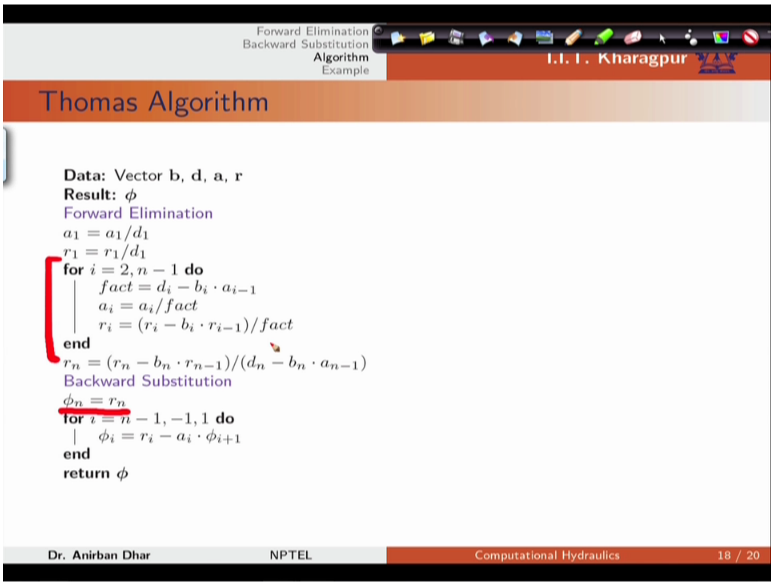
And at the end points we are explicitly calculating rn. We are avoiding the calculation of aN because aN is already zero value. We already know because we have this kind of structure. So lower value is zero. For b1, this is not required. That is why at the first place we have calculated a1. We already know that converted d1 will be our 1. So we don't need to write that value here.

(Refer Slide Time 21:06)



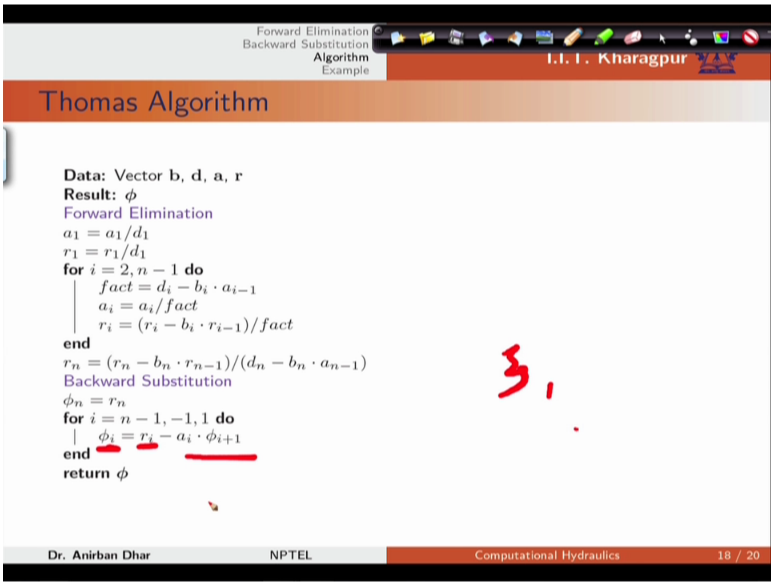
And backward substitution process, we can directly write it as phi equals to rn. Because coefficient is 1. So directly rnwe can write. Because within rn only we are storing the values. Now in this case one thing is clear. In Gauss elimination we have two step process. We are fixing a particular row and then operating on other rows. In this case we are utilizing the information of the previous row and we are operating on the next row.

(Refer Slide Time 22:03)



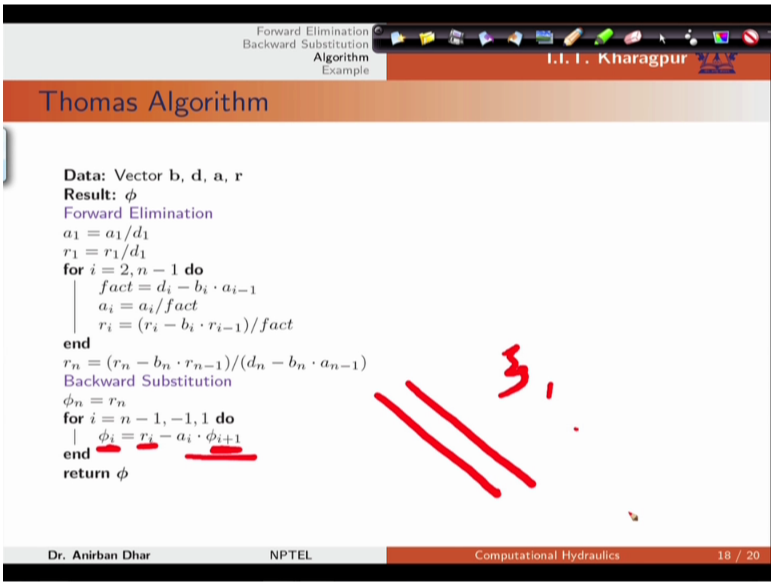
So only one loop is required. No two loops for this one. Similarly for backward substitution this is clear because coefficient of phi, this is 1. So r1 minus a1. So a1 is storing the xi1 2 values. So this will givethe actual value for this one.

(Refer Slide Time 22:42)



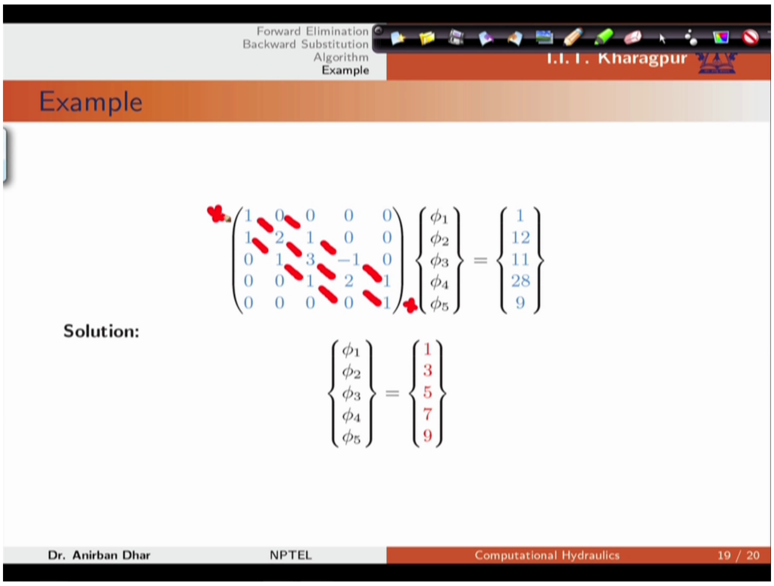
Because we need only phi i plus 1 value. Because we have only upper diagonal available. Lower diagonal, we have eliminated that and we have calculated it as zero value. So we don't need to utilize it within our calculation process.

(Refer Slide Time 23:08)



Now in this case we can again utilize our standard example. Because this is having diagonalstructure. So in this case we can utilizethe concept. This is not available, this first one is not available.

(Refer Slide Time 23:37)

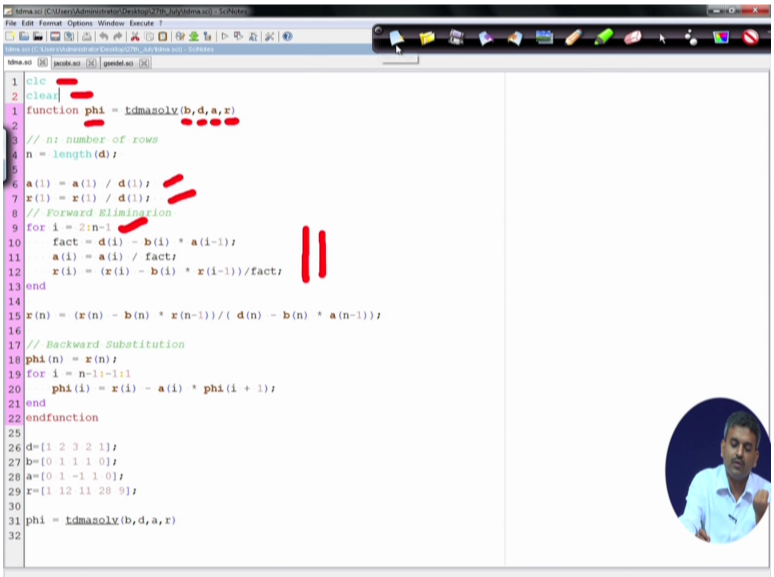


Now we already know what is the solution for this problem? We have utilized Gauss elimination and our LU decomposition for solution of this problem. Now in this casewe can use this tridiagonal matrix thing. This is compatibly a smaller code. It is not a lengthy one. Clc, clear, these are basic lines that are required to clear the console and clearthe memory in console. Then function required is phi. We need to supply b, d, a and r. This is tdmasolv.

Now n is the number of rows. Nequals to length of d. D is the diagonal that we are transferring. Now directly we are writing a1 equals to a1 divided by d1, r1 equals to r1 divided by d1. And in the forward elimination processwe are calculating this factor, i equals to 2 to n minis 1.Because we don't need to calculate the a.Because the nth value of a is not available. So fact equals to di minus b into ai minus 1. So it is the previous value.

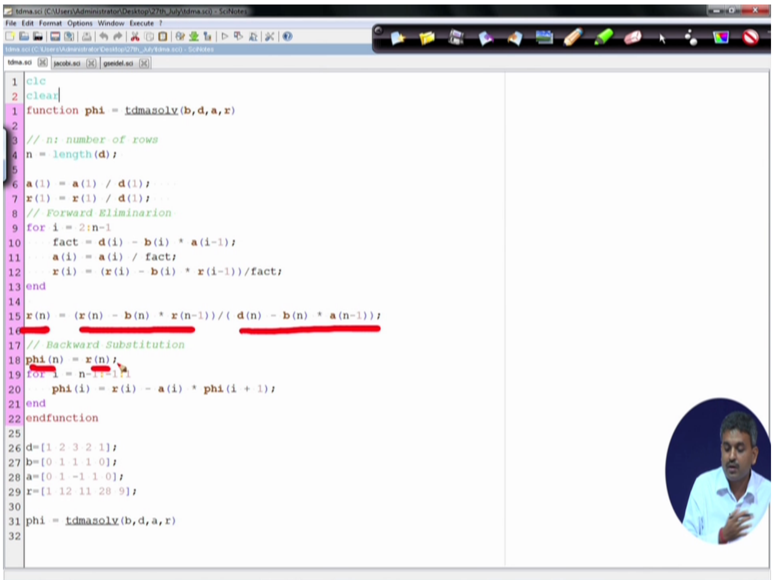
And with this factor we are just multiplying these to get a and r values, updated one. Because diagonal will be automatically converted into 1, unity.

(Refer Slide Time 25:40)



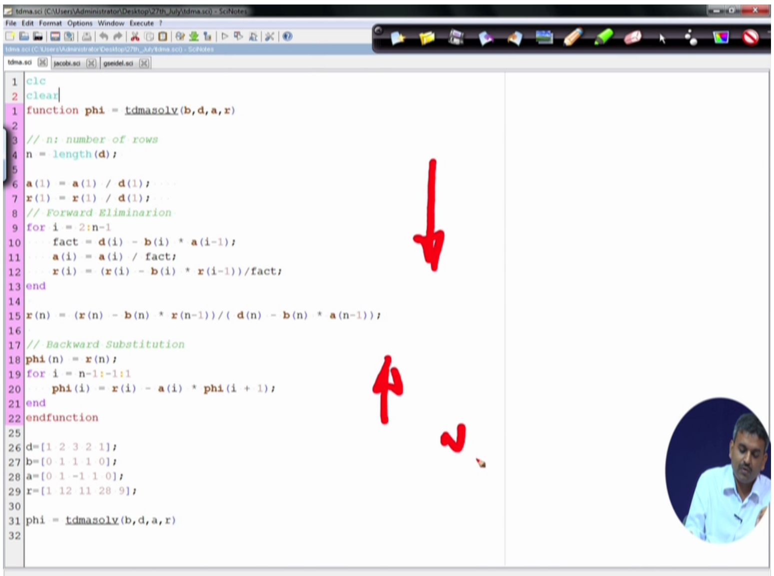
In this case rnequals to rn minus bn. We are directly calculating this. Now in backward substitution process phi n equals to rn. Now phi n equals to rn that is similar to our Gauss elimination.

(Refer Slide Time 26:04)



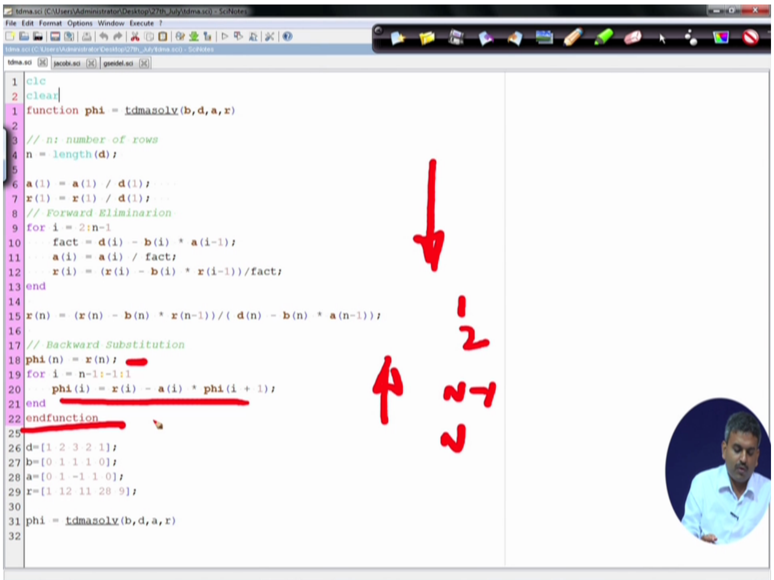
But in Gauss elimination you need information about all previous variables. But in this case only the immediate previous variablethat is for phi, phi i plus 1 is required. Because we are moving in the backward direction. This is forward elimination, backward direction. We are starting from N.

(Refer Slide Time 26:34)



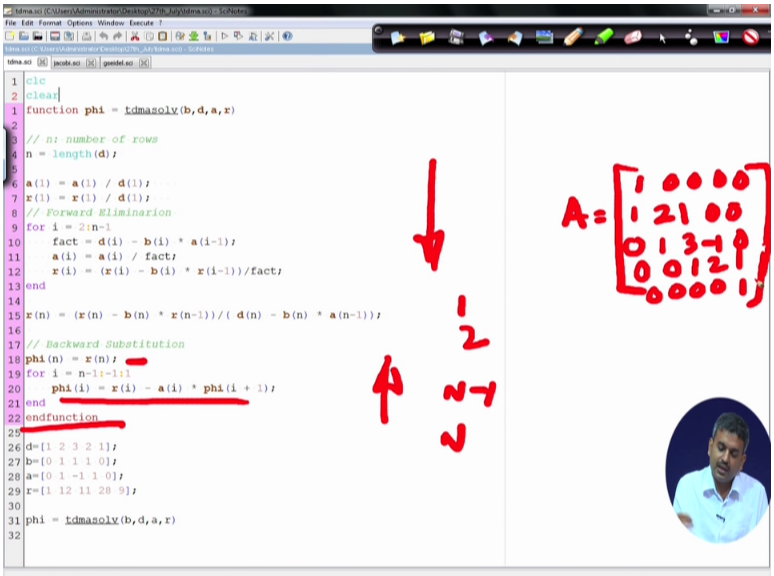
So we already have N value here. Now for N we are starting calculation of N minus 1. To calculate N minus 1 we need value of N. So to calculate 1 we need value of 2, like that. So finally we can get the solution.

(Refer Slide Time 26:57)



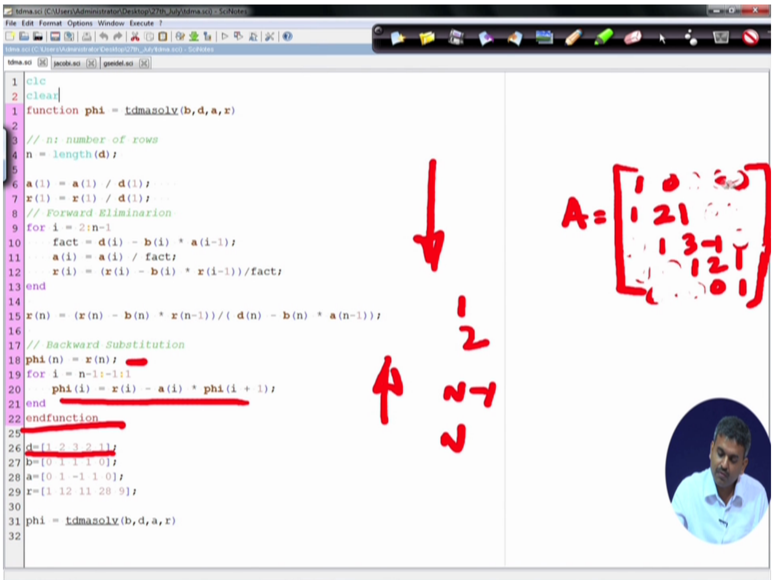
So now how to store? Because in case of our LU and Gauss elimination we have seen that we have stored the matrix like this, 10000, 12100, 013 -10, then 00121 and four 0 and 1. This was the structure for A.

(Refer Slide Time 27:35)



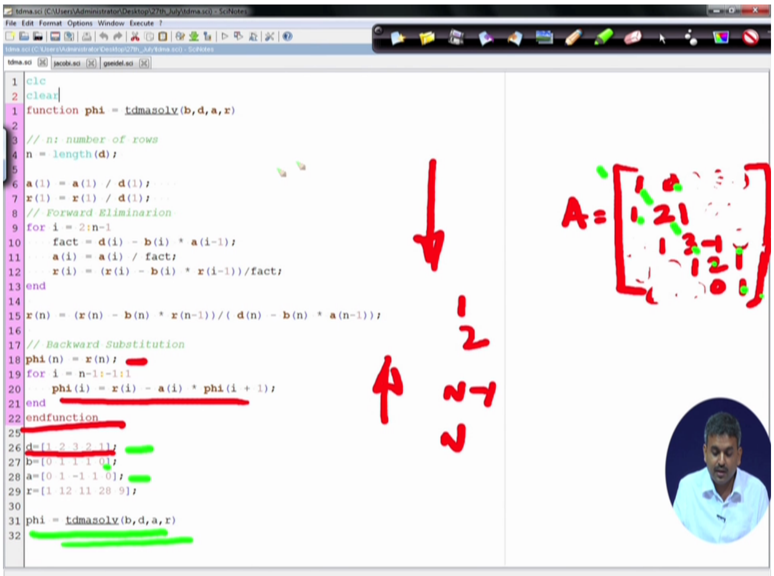
Now in this case no need to supply the matrix that way. So we can directly supply this d. So d is actually 1. So we can eliminate these values directly and we can directly supply this 0.

(Refer Slide Time 28:10)



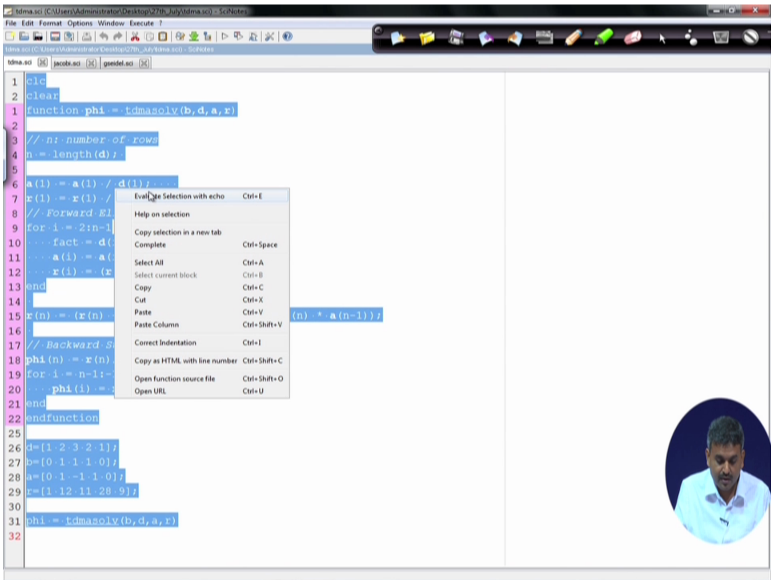
So this is 1 this is 112321. So this is 12321. So next is below diagonal. This is b is 01110. So this is 01110. Then above diagonal this is starting from 01 -11 and last one is 0. And r which is on the right hand side this is 1, 12, 11, 28 and9. Now we can call this tdmasolv to get the solution.

(Refer Slide Time 29:07)



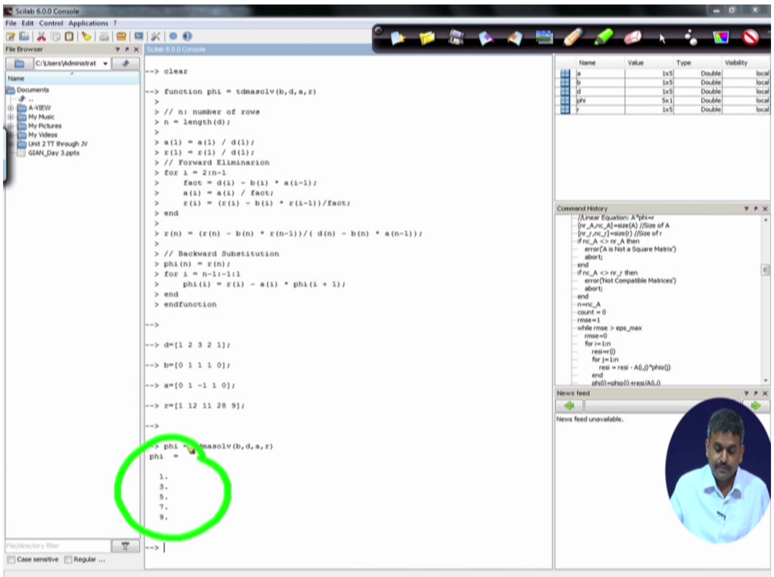
So let us see what solution it is giving. Now we can select it andevaluate selection.

(Refer Slide Time 29:20)



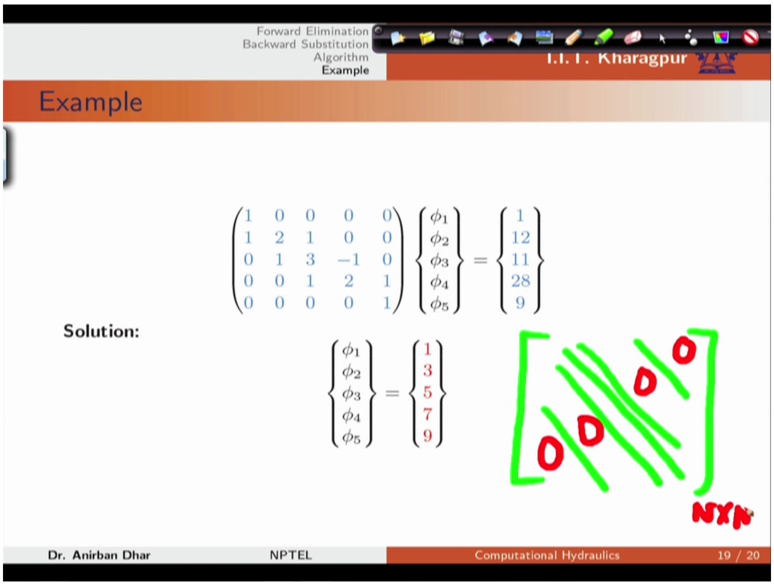
We can see that it is directly providing the desired solution. Desired solutions is our 1, 3, 5, 7, 9,odd numbers. Now we can see that with a minimum storage we are getting maximum output. So a matrix structure is important during calculation process and we can utilize the matrix structure to frame a particular algorithm.

(Refer Slide Time 30:09)



Even for penta-diagonal structures, if we have penta-diagonal matrix structure then penta-diagonal matrix structure like this that we have seen in our one of the two dimensional problems. That if you have zero values in between and you have 5 diagonals. not diagonals, one diagonal and two off diagonals on each side available then we can frame our algorithm in such a way that it not needmaximum storage required that is N cross N. We can reduce our storage structure by introducing different algorithms.

(Refer Slide Time 31:03)



Next lecture onwards we will start the iterative techniques to solve the same kind of problems but the approach is somewhat different. Thank you.