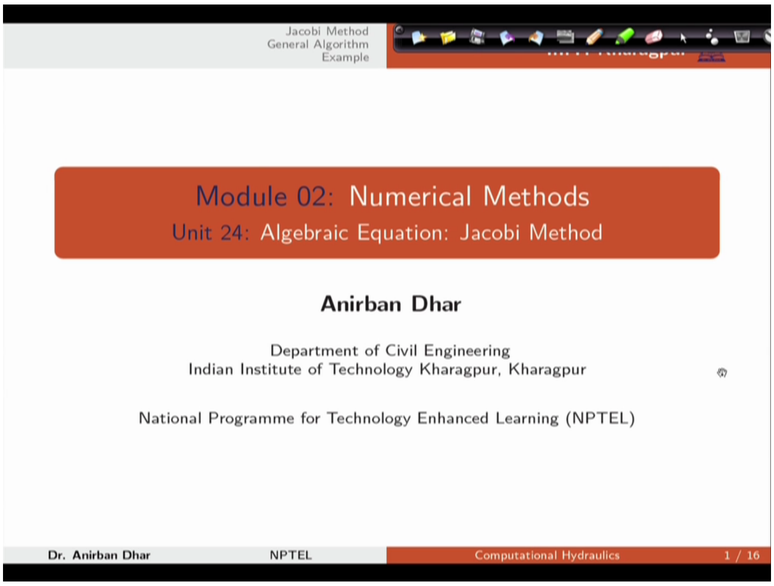
Computational Hydraulics
ProfessorAnirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Lecture 28
Algebraic Equation: Jacobi's Method

Welcome to lecture number 28 of the course computational hydraulics. We are in module 2, numerical methods. And in this particular unit number 24, I will be talking aboutalgebraic equation,Jacobi method.

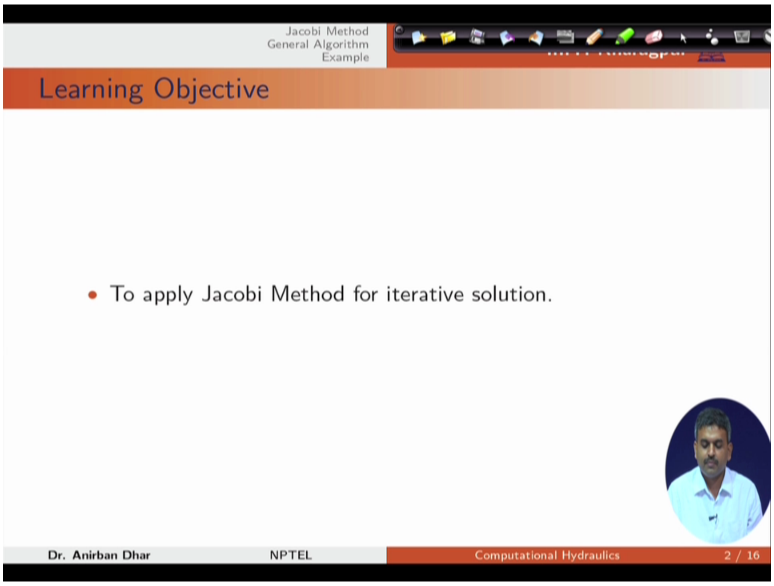
(Refer Slide Time 00:40)



Jacobi method, essentially this is iterative technique to solve any algebraic system with constant coefficient. We can utilize either direct approach. We have seen that Gauss elimination,LU decomposition or tridiagonal matrix algorithm can be utilized to solveour direct our matrix to get our phi value. But in Jacobi's method or Jacobi method we will talk about iterative technique to solve the problemstarting from a guess value.

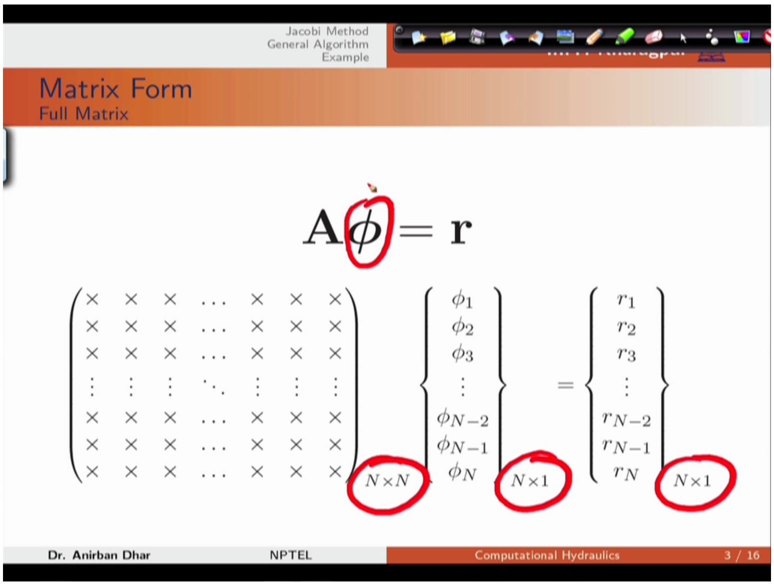
Learning objective,at the end of this unit students will be able to apply Jacobi method for iterative solution of our algebraic system.

(Refer Slide Time 01:45)



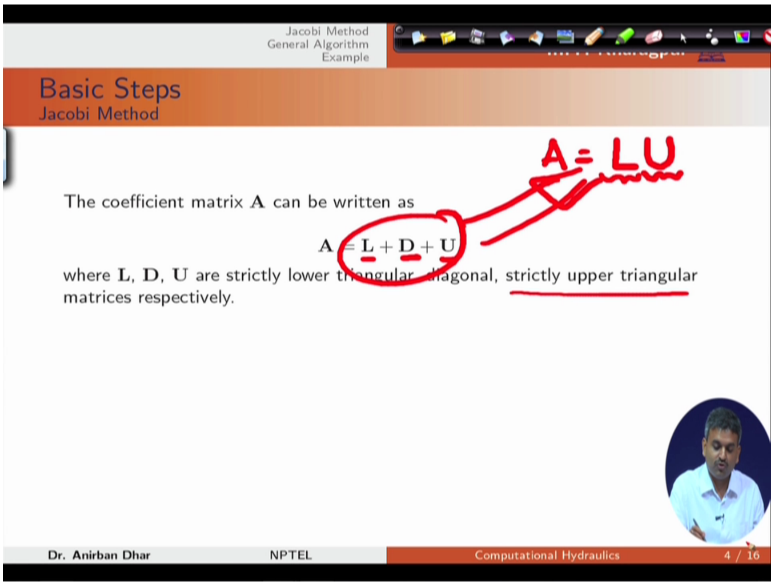
Matrix form,full matrix. If we consider full matrix then we have N by N, N cross 1,N cross 1. In this casewe are defining the algorithm for general purpose matrix. That means it's having full coefficients. That means all nonzero coefficients or maximum number of nonzero coefficients available within the system. Phi is our desired variable.

(Refer Slide Time 02:29)



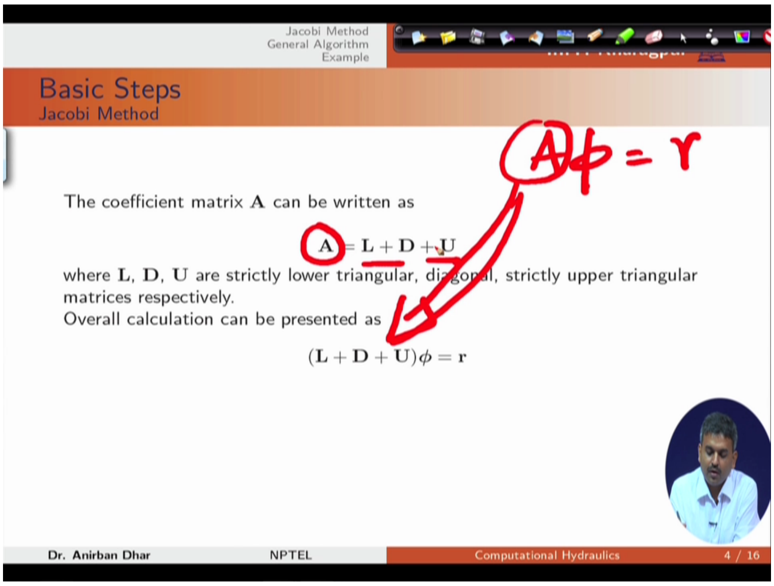
Now in this case what are the basic steps?In case of LU decomposition we have divided or decomposed the matrix into LU format. That means one lower and one upper triangular matrix. In that case this decomposition was valid for multiplication. But in this case we are essentially considering L plus D plus U, where L is strictly lower triangular matrix, this is strictly diagonal matrix and this is strictly upper triangular matrix. This has got no connection with our LU decomposition discussed in our direct solution approach.

(Refer Slide Time 03:41)



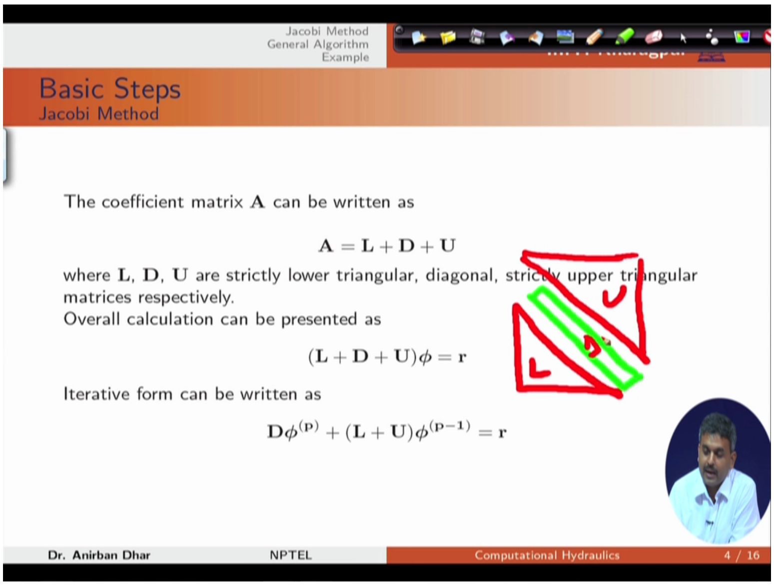
Now in this case overall calculation can be presented like this. As we have decomposed our A, our general structure was A phi equals to r. So in place of A we can replace L plus D plus U. That means lower, diagonal, upper.

(Refer Slide Time 04:13)



Now for this one we can write it like this. In this case we have divided it into diagonal parts and off diagonal parts. So if we say that this is our matrix so we will divide it into one lower part, one diagonal part and one again upper triangular part. These are strictly. That means this diagonal term is not included either in ourupper or lower triangular. This isonly applicable for diagonal matrix.

(Refer Slide Time 05:15)

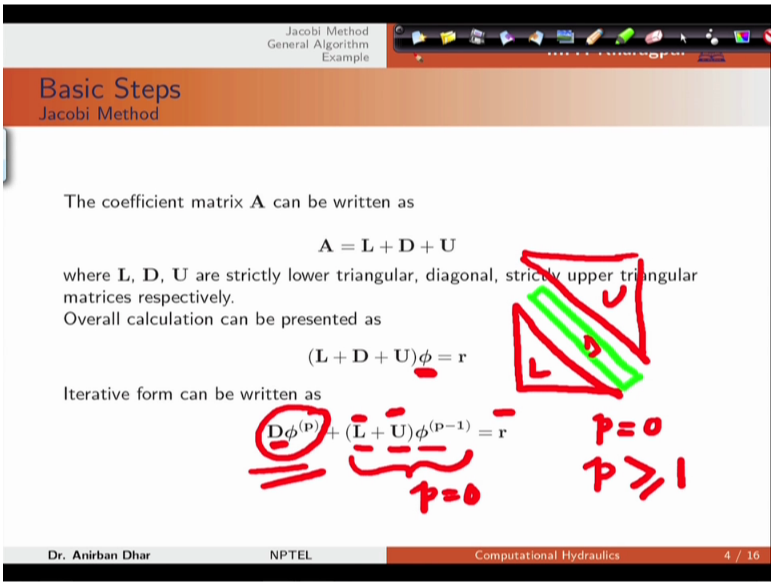


Now let us decompose this phi like this. Phi can be multiplied with our D. And L plus U this is multiplied with previous iteration value P. P stands for the iterationcounter. So if we start from zero which is a guess level, this equation or expression is valid from P greater than equal to 1. That is why P equals to 1 that means this is for first iteration. This L plus U into phi P minus 1. That means this value will be evaluated for guess value.

That means P equals to zero level. Now in this case four things are fixed. That means D matrix is constant, L matrix is constant, U matrix this is constant and r is also constant. The only varying thing is phi. Every iteration phi is varied. We are not changing this L U. That means like our direct approach we have decomposed our matrix into different forms and we have changed those values.

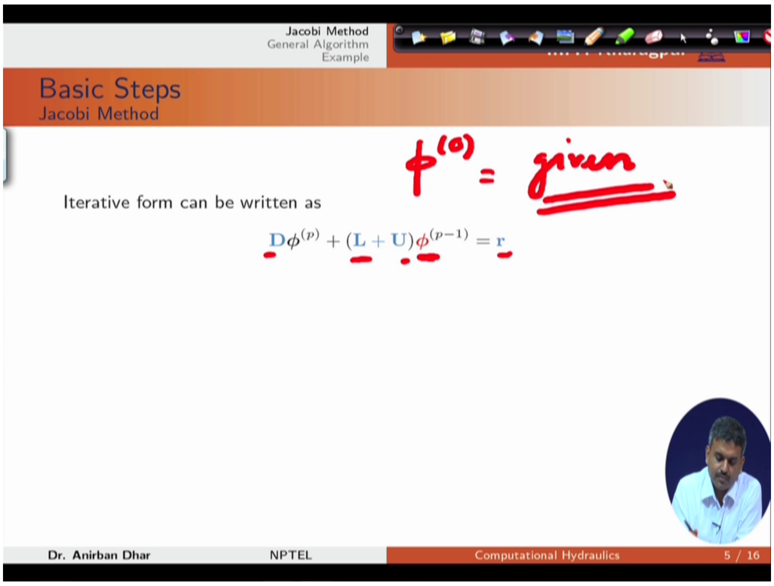
But in this case we are not changing the values directly. We are keeping this structure intact and we are just dividing it into lower, diagonal and upper matrix, in additive form.

(Refer Slide Time 07:22)



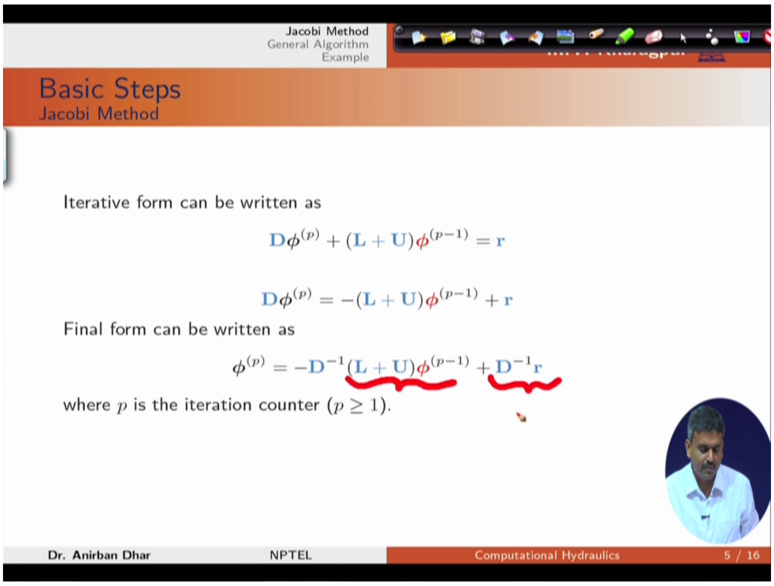
So iterative form, because we are not disturbing this diagonal, lower, upper value and r value. So I amusing the blue color. So that means we are not changing these values. And this phi value at previous iteration P minus 1, this is available. If we start with zero this should be given.

(Refer Slide Time 08:03)



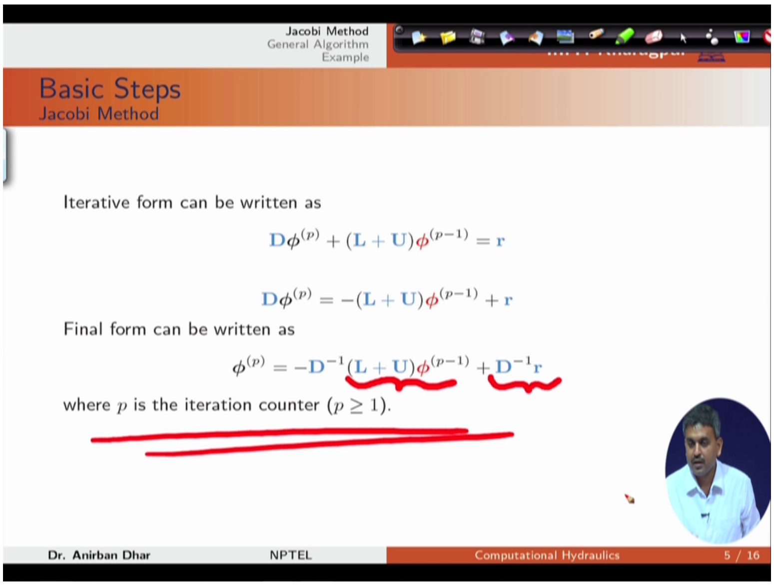
Now in iterative form if we further write this, so we transfer it on the right hand side. Now this is necessarily inverting the diagonal matrix and multiplying itwith the,first one is our previous iteration value. And this is our right hand side vector multipliedby this diagonal term inverse.

(Refer Slide Time 08:36)



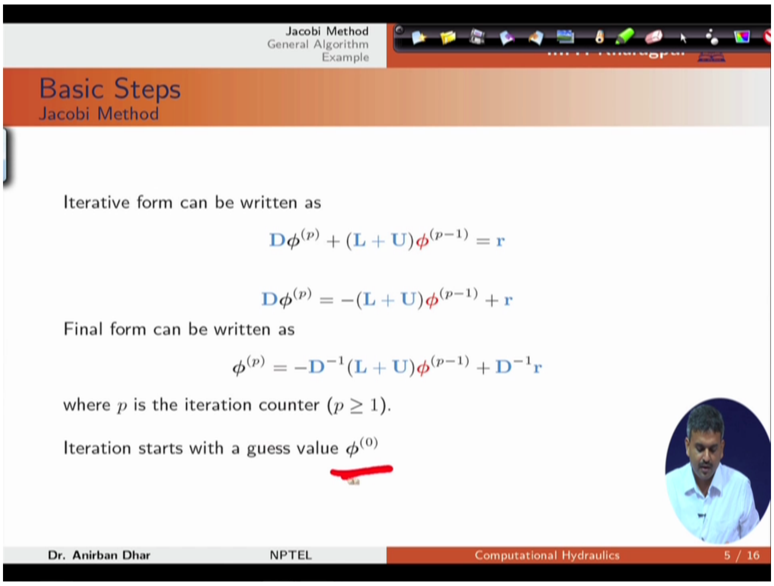
Now this iterative process,this is valid for P greater than equals to 1, because we cannot apply this for zeroth level which is the guess level for any iteration.

(Refer Slide Time 08:50)



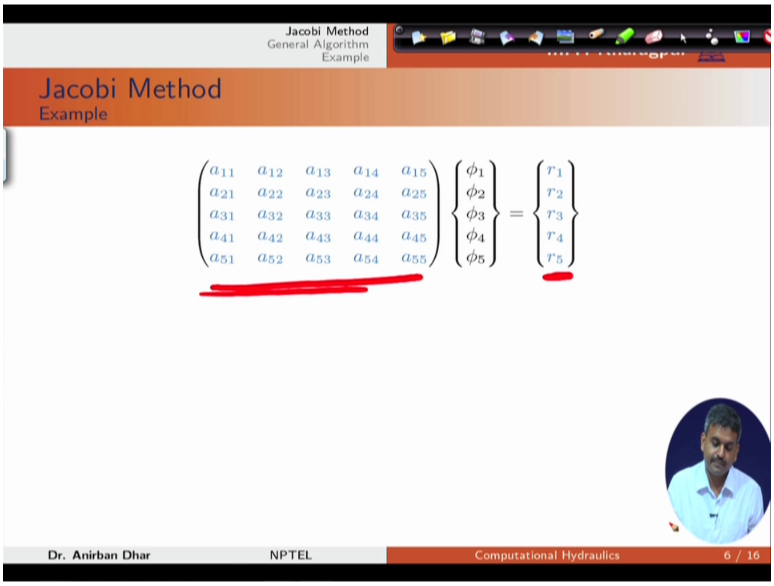
Now iteration starts with a guess value phi0.

(Refer Slide Time 09:00)



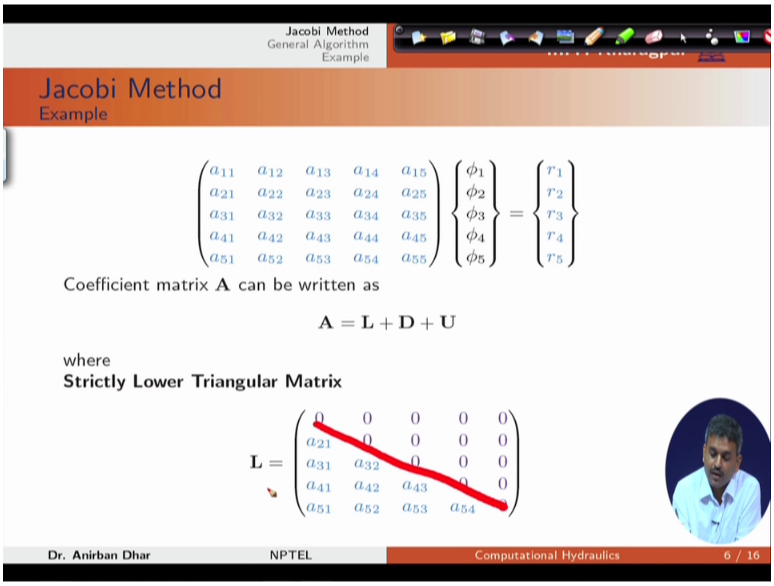
Now in this process if you want to explain the whole thing, blue values are undisturbed values, r values are also undisturbed values, like over directapproach where we are changing different rowsto get the desired solution.

(Refer Slide Time 09:30)



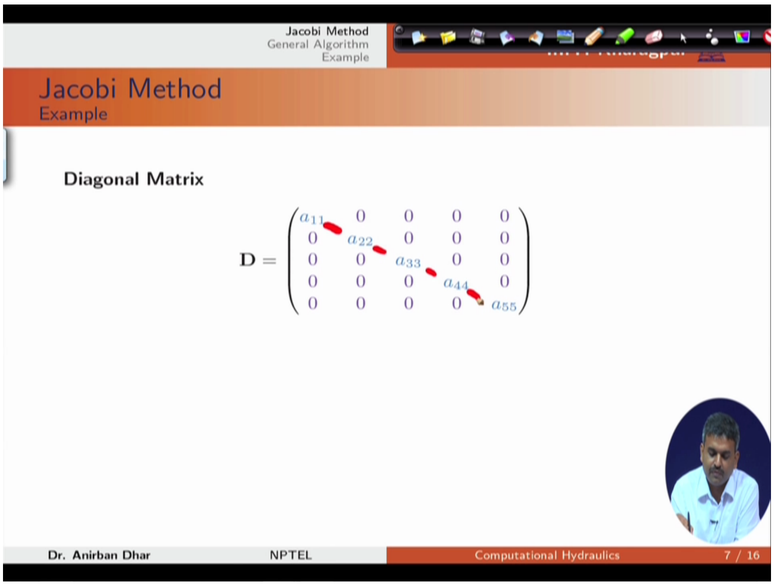
So in this case the coefficient matrix A can be written as Lwhich is strictly lower triangular matrix. Noticethat we are not including the diagonal term here. This is strictly lower. The coefficients are exactly same.

(Refer Slide Time 09:55)



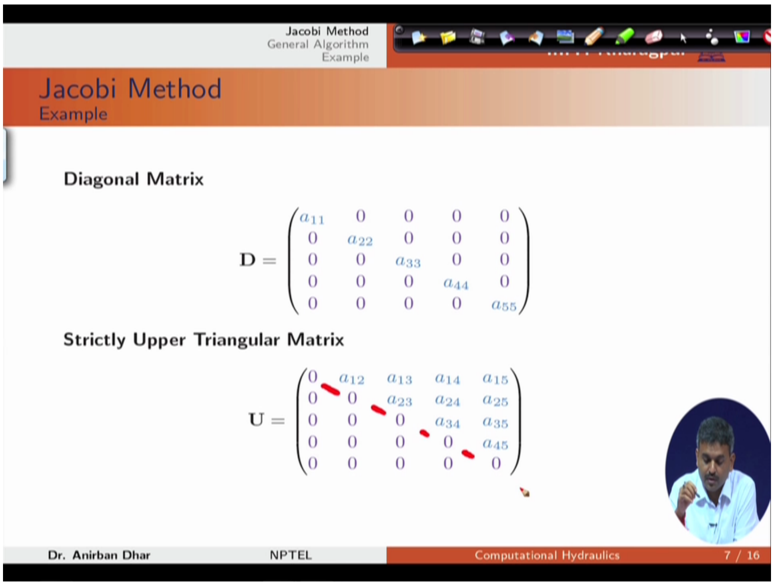
Next level iswe are dividing it with diagonal term. So that means only diagonal term is there.We have 5 by 5 matrix. So 5 terms.

(Refer Slide Time 10:09)



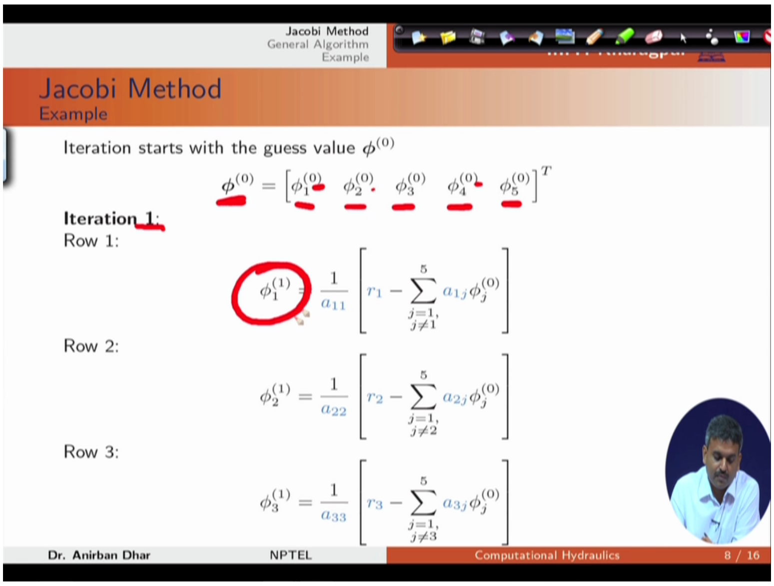
Our next levelwe have strictly upper triangular matrix.This also is without diagonal term because we are not including diagonal term in this process.

(Refer Slide Time 10:25)



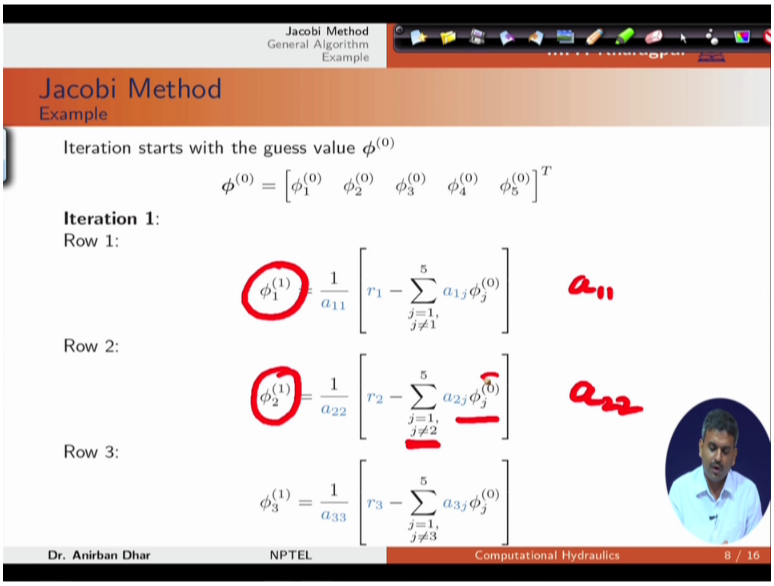
So iteration, this starts with phi0. Now phi0 in this case this is a column vector. Now this is phi1, phi2, phi3, phi4, this is valid for zeroth level. These values are given values. Now iteration 1, we are calculating the iteration 1 values and this is we are just transferringoff diagonal coefficients including the off diagonal variables. So for 1 this is a1j and phi j at zero. That means whatever value is available at the previous time level we are utilizing that for calculation of this off diagonal thing. So if I divide it by a11, I should get guess for from this iteration 1.

(Refer Slide Time 11:38)



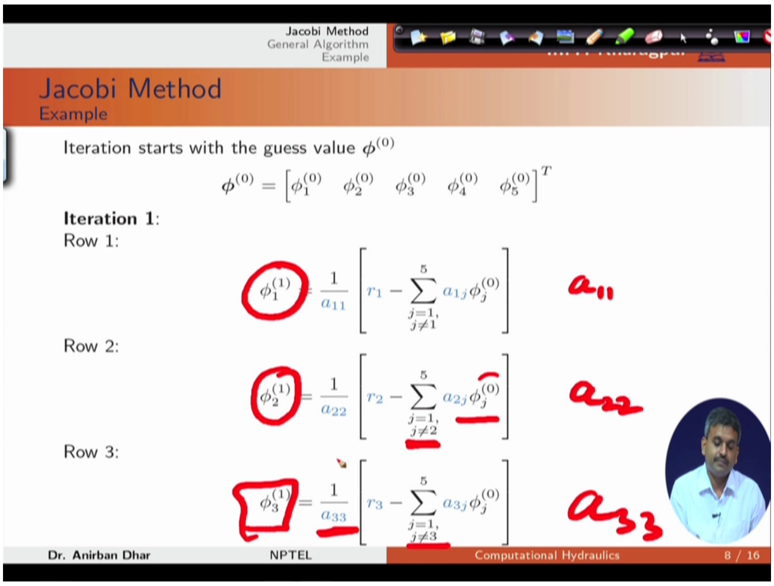
Now with this case I am not calculating anything. For a particular iteration I have calculated this. Now this is valid for row 1. Now for row again I am calculatingwithout including the diagonal term. So in this case a11 is the coefficient of the diagonal term. In this case a22 is the coefficient of diagonal term. So without including j equals to 2, j starting from 1 to 5, we are calculating this using our old value or guess value.

(Refer Slide Time 12:23)



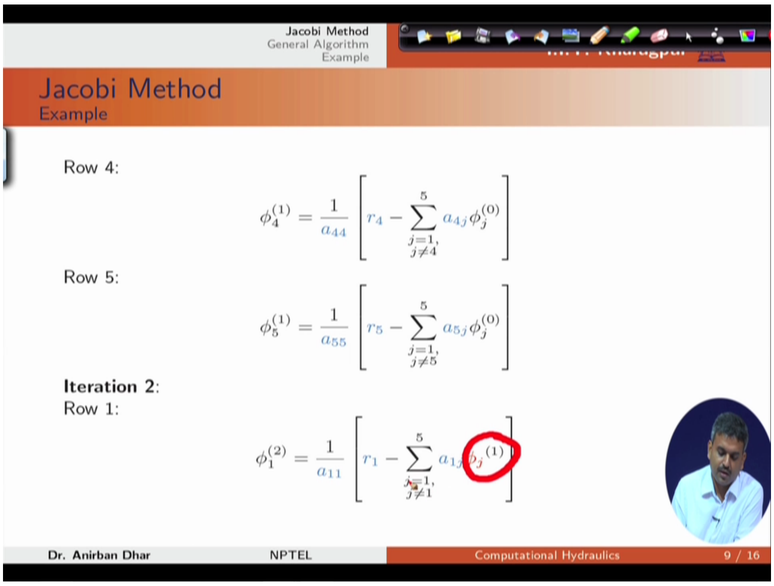
Row 3, again a33 and without including j equals to 3. That means without including the diagonal term we have again calculated this phi3.

(Refer Slide Time 12:43)



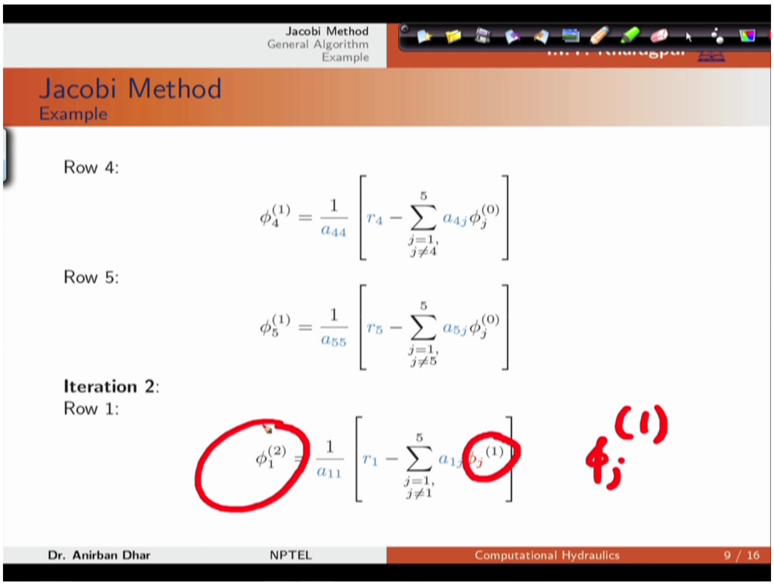
Now in this process we can again calculate a44 by omitting this fourth term which is diagonal term. Then fifth one, by omitting that fifth one. Now in iteration 2 we will utilize this updated value from one.

(Refer Slide Time 13:24)



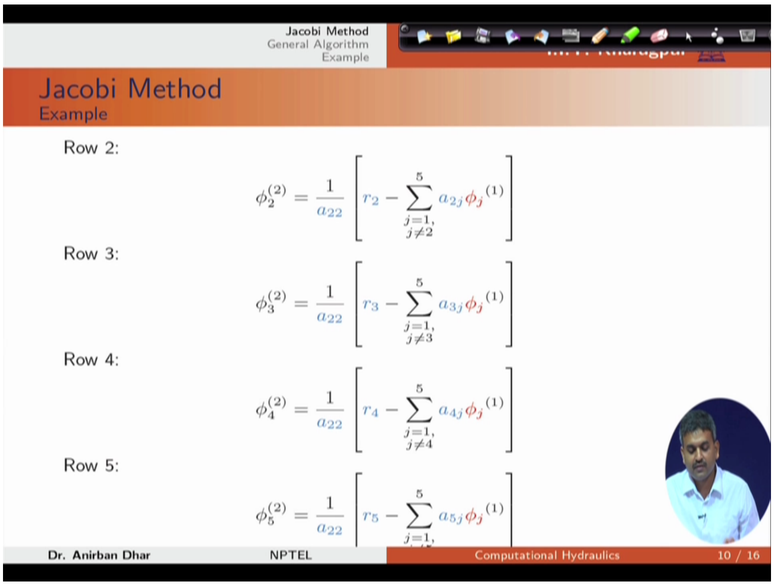
So whatever value we have got from our iteration 1, this stands for iteration level 1 because iteration level zero was our guess level or initially provided value. So iteration level 2 value calculated based on iteration level 1.

(Refer Slide Time 13:48)



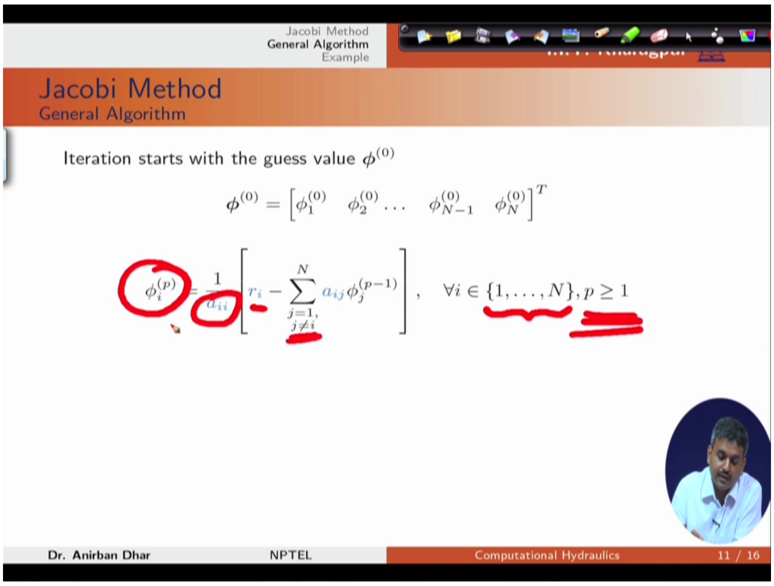
Now in this case row 2, row 3, row 4, row 5, these values can be calculated based on our this straight forward process where we calculate the values on previous iteration level value. We are not considering time level here, we are simply considering the iteration level.

(Refer Slide Time 14:22)



So if we talk about the general algorithm, again if we have N numbers of variables. So phi 1, 2 to N minus 1, N. Againthis one we can calculate ith row at pth iteration. That depends on ri minus the summation of j equals to 1 to N and by omitting ith term only. And aij, blue terms because we are not changing these coefficients or right hand terms. So that's why aij, phi j p minus 1. And this is valid for all rows. And this is applicable for P greater than equals to 1. So we are dividing it by a ii.

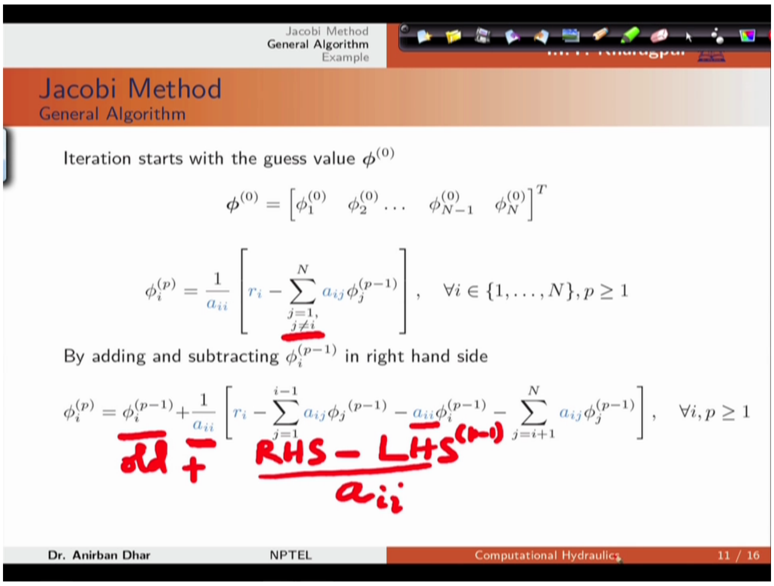
(Refer Slide Time 15:33)



Now in this process we can get the value for pth iteration level. Or by adding and subtracting this phi i p minus 1 we can get that, let us say that we are adding phi ip minus 1. We have omitted here. Let us add that value within the system. So if we add this value we have this efficient a ii, phii,i p minus 1. All values are calculated as p minus 1 and divided by a ii. So old value, this is old value plus this is right hand side minus left hand side. Because we have calculated all values here.

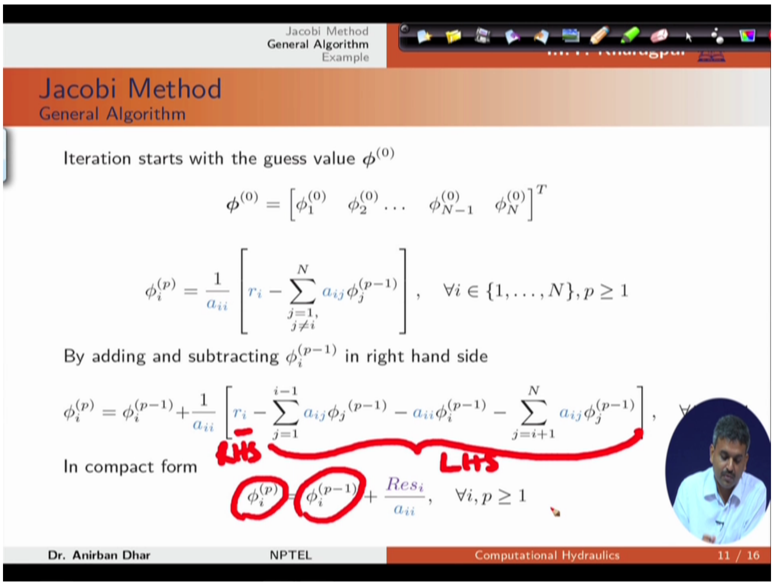
So this is at p minus 1 level and a ii. So we can divide it by the diagonal term. So we can say that in this case the Jacobi iterative technique can be generalized like this.

(Refer Slide Time 17:08)



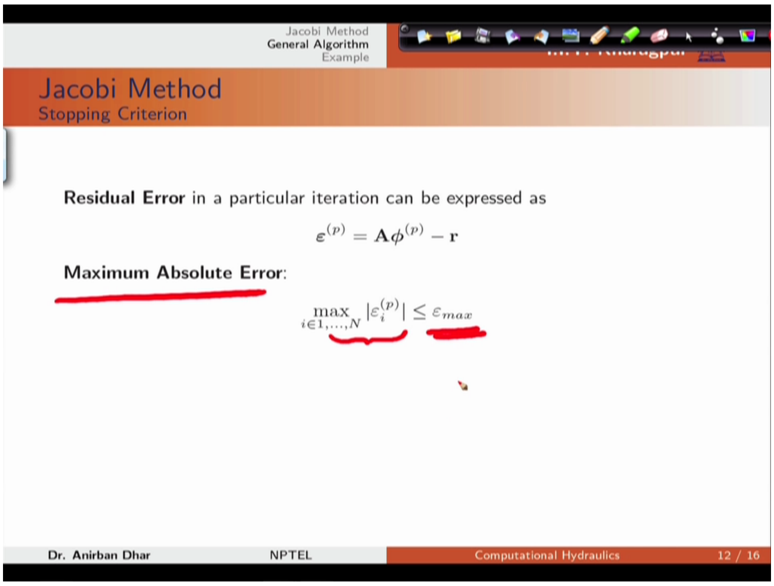
Thatif we have any row ithrow, updated level value that should depend on the previous time level value plus residual. The residual means you are right hand side minus your left hand side. This is your right hand side minus left hand side. This is residual for ith row divided by a ii. So we can get a simplified form of our Jacobi iteration or iterative technique.

(Refer Slide Time 17:45)



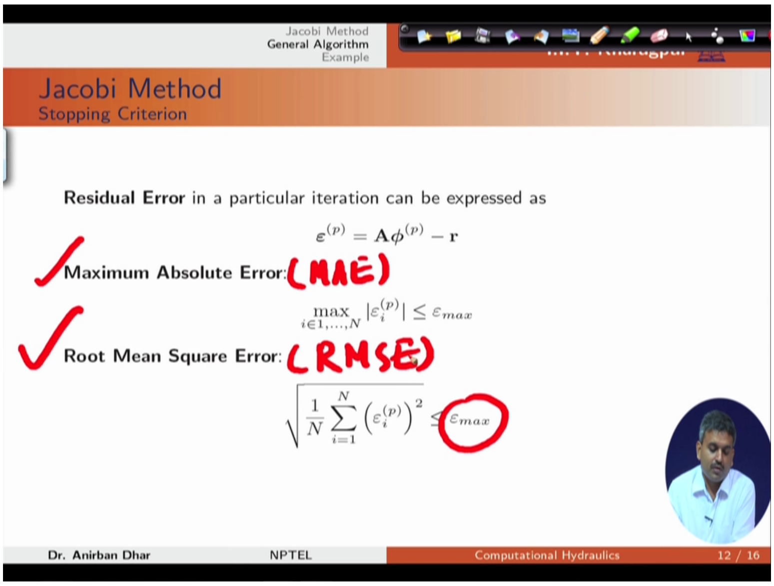
Nowresidual error is important because every iteration there will be error associated with that. Now what should be the stopping criteria? Stopping criteriashould be based on this residual error. Now residual errorwe can utilize to calculate this maximum absolute error. So this is maximum for all i in i to N. We can calculate the maximum value of thisresidual error for different rows and we can get the maximum values. That should bebelow the desired error. This can be one criteria.

(Refer Slide Time 18:47)



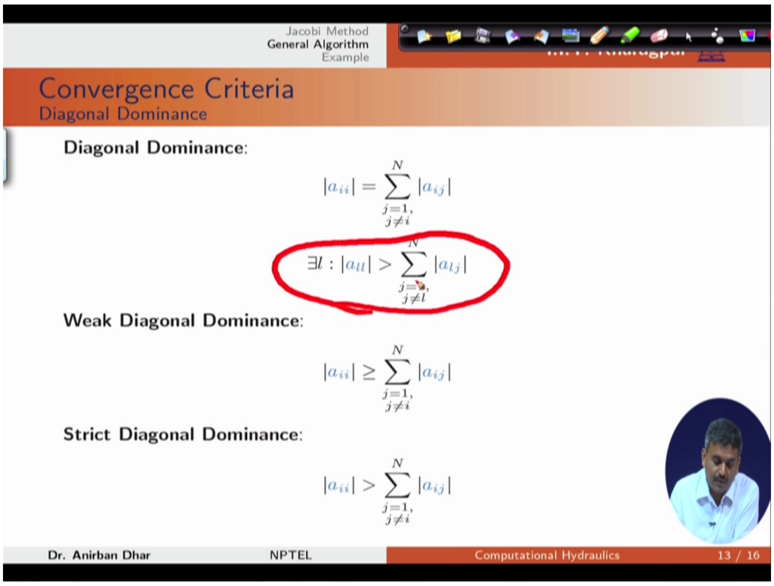
And other one can be root mean square error that we have difference, we can take the square, sum it up,divide it by N. And we can take the squareroot of that. And this should bebelow the desired level. So we can use either this one or this one or we can use the combined one that maximum value of root mean square error. This can the maximum absolute error or RMSE value.

(Refer Slide Time 19:33)



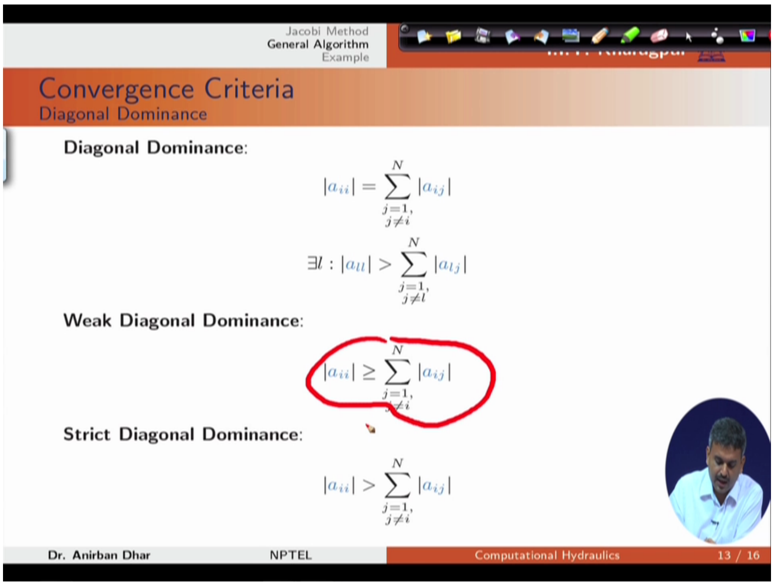
Now these values can be utilized for our coding purpose in scilab. Then comes this diagonal dominance. For convergence of iterative schemes we need these diagonal dominance. That means ifthe modulus value of diagonal term is a ii. If it is equal to sum of absolute value of off diagonal terms and there exist one L for which it is greater, thenwe can call this as diagonal dominance.

(Refer Slide Time 20:31)



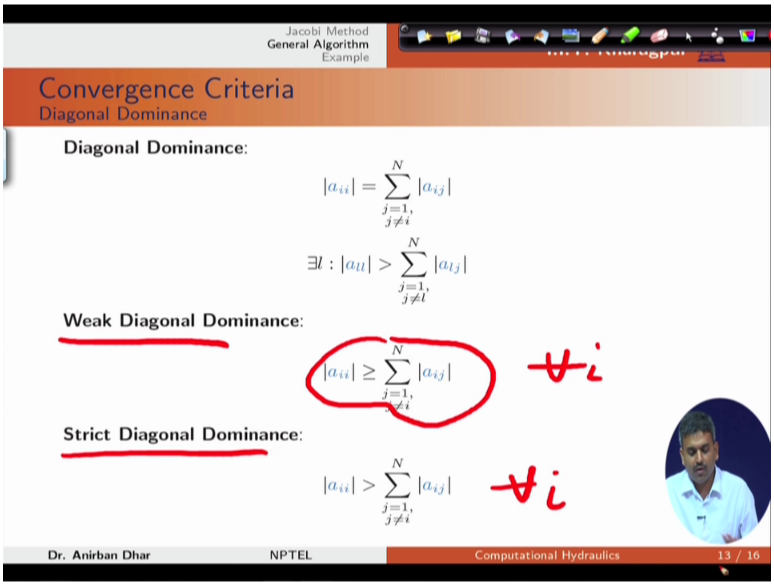
Andif we have weak diagonal dominance then there can be multiple number of rowswhich can bewhich can satisfy this particular criteria.

(Refer Slide Time 20:49)



And for strict diagonal dominance, this strictly greater than condition should be (satis) satisfied for all rows. This is valid for all rows and this is alsovalid for all rows. That means for all i this conditionshould be satisfied. So we can call it as weak diagonal dominance or strict diagonal dominance.

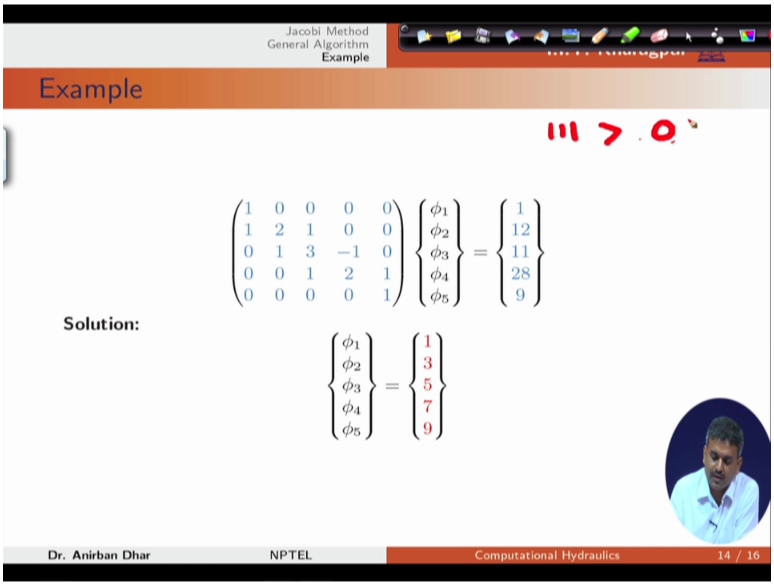
(Refer Slide Time 21:19)



Now in iterative scheme diagonal dominance is important. Diagonal dominance criteria should be satisfied otherwise we will not get convergence in case of iterative schemes.

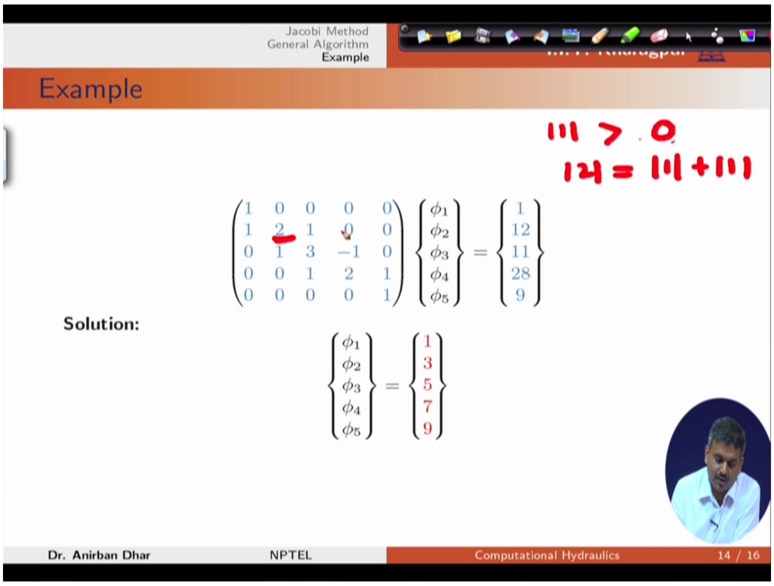
Nowlet us consider example.We already know that we have utilized this problem for our Gauss elimination, LU decomposition and TDMA or tridiagonal matrix algorithm. Now in this case interesting part is that if we consider first row,this first row is greater than mod 0. Because all values are zero, has got no meaning. So we can say that this is greater than zero.

(Refer Slide Time 22:21)



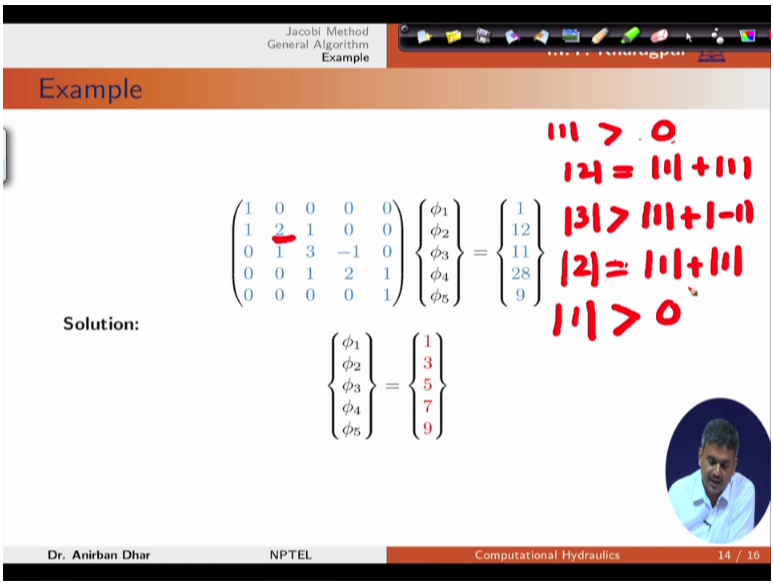
Then for second row the diagonal term is this one, a ii. So a22 term is mod 2. This is equals to mod 1 plus mod 1. So this is equal.

(Refer Slide Time 22:41)



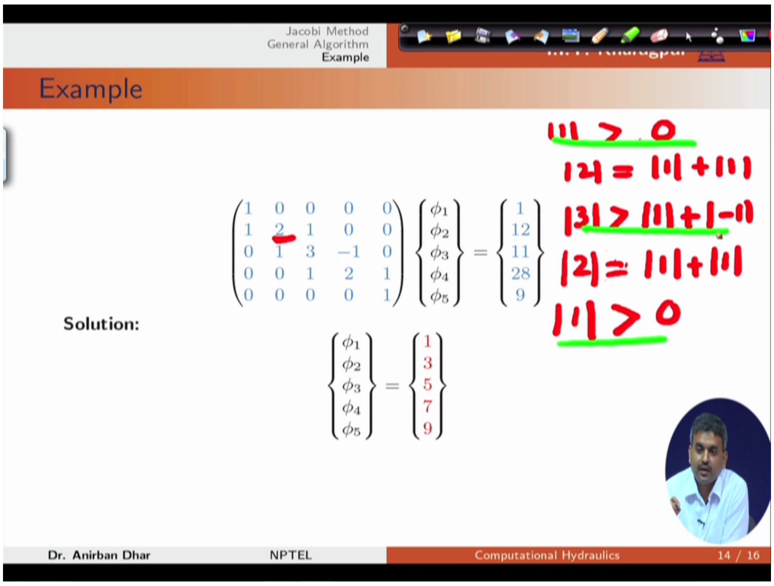
Next row, this is 3 and next condition is 1 plus minus 1. That means 3 is greater than 2. That is why this diagonal dominance is there. It is not strict because one case we have equality sign. Then this is for fourth row. Again we have this equality condition. Equality condition 1 plus mod 1. And 5th row again we have diagonal dominance in this case.

(Refer Slide Time 23:35)



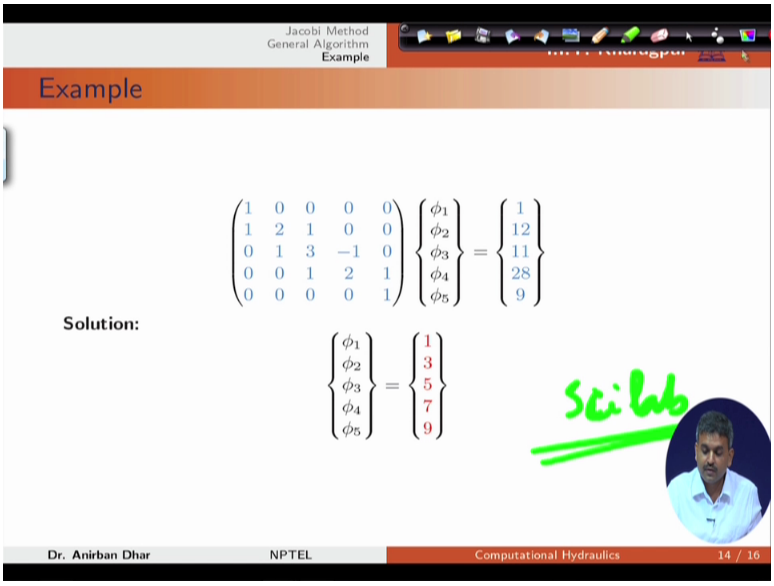
So we can see that for 3 rows 1, 2, 3, we have diagonal dominance or strict dominance. And for two rows we have equality signavailable.

(Refer Slide Time 23:55)



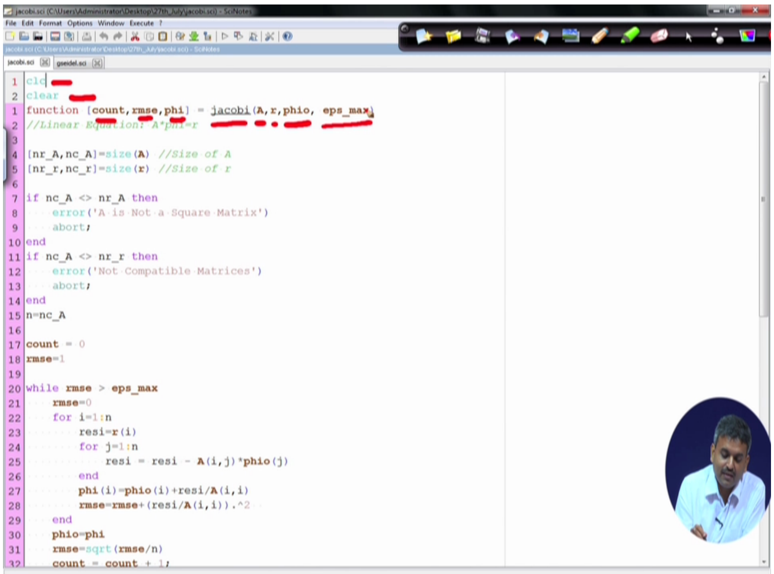
Now in this case if we write ourconditions thenwe can write our code in scilab. So let us use it for this one.

(Refer Slide Time 24:20)



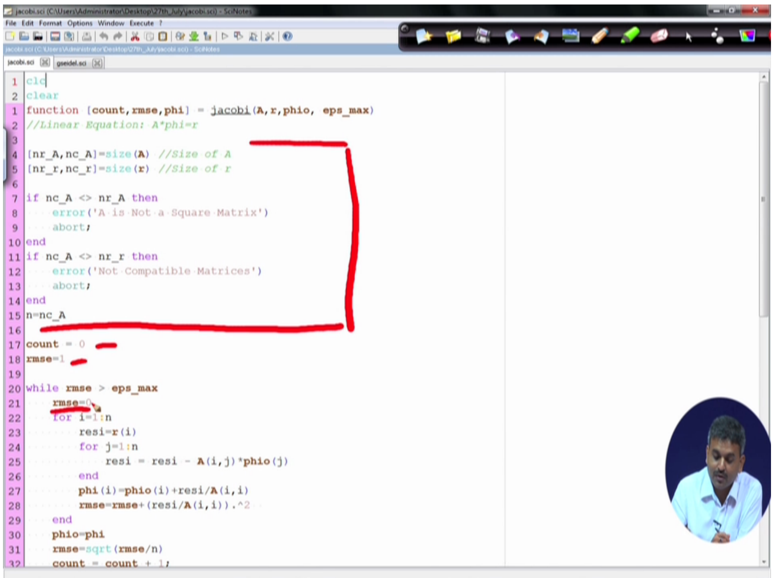
We can use the structure here. The clc again this clear screen, this is clearstructure or clear memory. This count is for number counter for iteration. Rmse is root mean square error and phi. Phi is the variable. Now in Jacobi iteration we need it A. A is the matrix. R then phi o. Phi o is the initial guessthat needs to be supplied and epsilon max. This is maximum allowable error which needs to be specified for this case.

(Refer Slide Time 25:22)



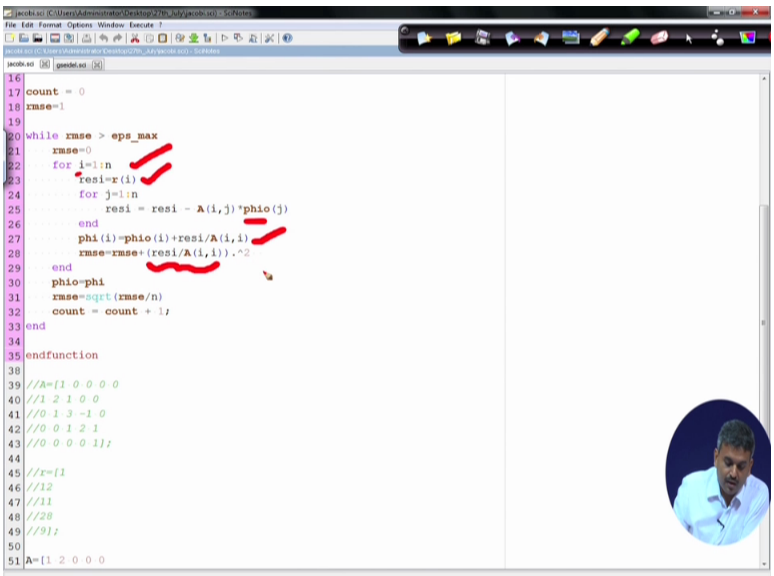
Now counter we already know this part from our Gauss elimination, LU decomposition and other course. So we are starting with count equals to zero. So counter is zero. Initially I am specifying rmse equals to 1. Why? Because otherwise this while loop will not be executed here. So if rmse is greater than epsilon max then only this willbe executed. So let us say that rmse equals to 1 then automatically the process will start within this. And at the beginning itself, I am specifying rmse equals to zero.

(Refer Slide Time 26:13)



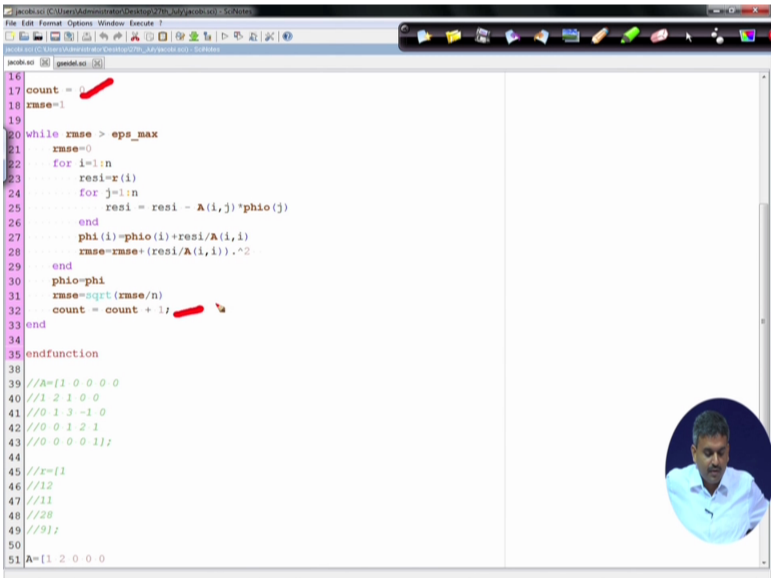
So I am initializing everything. Now in this processI am running it for all rows.So i starting from 1 to N. This residual let us say ri, this is right hand side minus for all j. So resi equals to resi i minus aij into phio that is initial guess minus jfor jth element, we are calculating this. Now this is nothing but right hand side minus left hand side and we can divide it by the diagonal term a ii. So rmse is nothing but the difference in old value and new value. So old value and new value difference is resi divided by a ii.So I am taking square of that.

(Refer Slide Time 27:33)



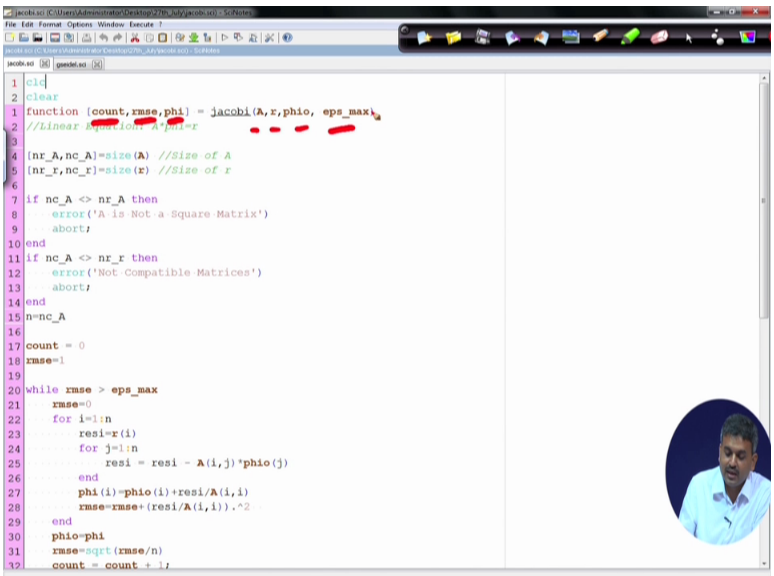
So after this now I am specifying this old value as new value. So in place of old value I am changing this thing. So and finallyin this process I have calculated rmsefor all rows and finally its square root sqrt of rmse divided by n. So taking some of that we are dividing it by n. And then we are taking it as square root. And count is equal to count plus 1 that every iteration. So we have started with counter zero and this process we are counting.

(Refer Slide Time 28:27)



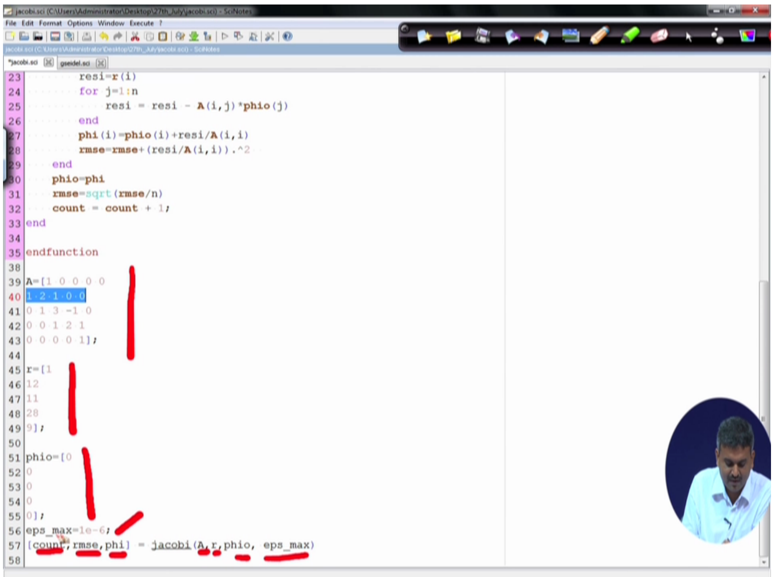
So count equals to count plus 1. So we have calculated rmse, we have calculated count. Nowlet us use this as function for calculation of our case. So function is count rmse phi and a, r, phi o, epsilon max, these are the values.

(Refer Slide Time 28:57)



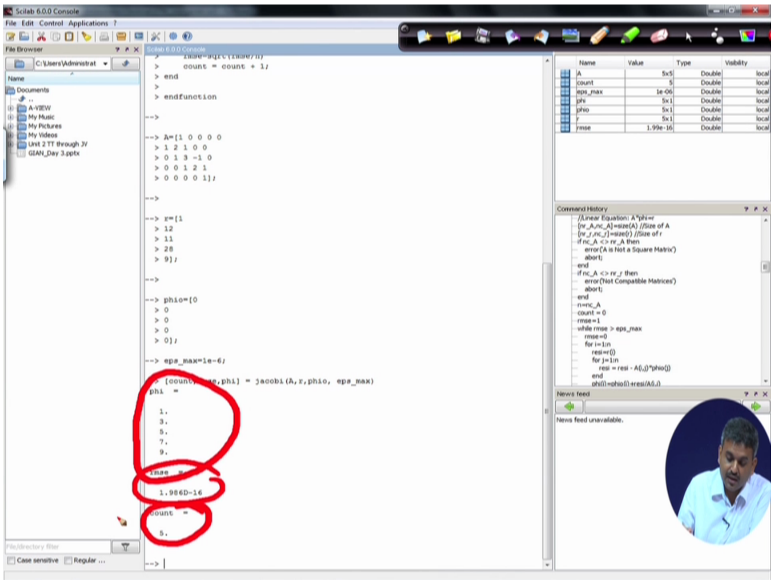
Now in this process let us say that we have ouroriginal tridiagonal matrix available. So 121, this is 131. Now in this case if we calculate this. In this case let us say this is our definition regarding A, this is r vector, this is phi o. Weare starting from zero-zero value. We don't know what is the value. So epsilon max, we have specified 1 e 10 to the power minus 6.So 1 e minus 6 means 1 into 10 to the powerminus 6. So we are calling this Jacobi function with count,rmse, phi and we are providing a, r, phi o in terms of zero-zero and epsilon max as 1 e minus 6.

(Refer Slide Time 30:26)



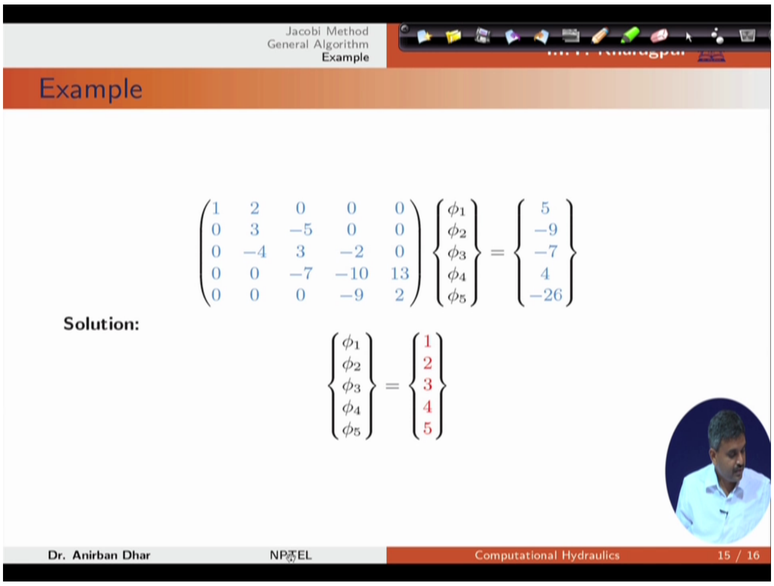
So if we run thisprogram by selecting it and then executing it. So you can see that we are getting the exact desired value 1, 3, 5, 7, 9. And in this case this rmseisvery less. It is close to zero 10 to the power this is minus 16 andcount 5. That means with 5 iteration only we are getting this solution.

(Refer Slide Time 31:15)



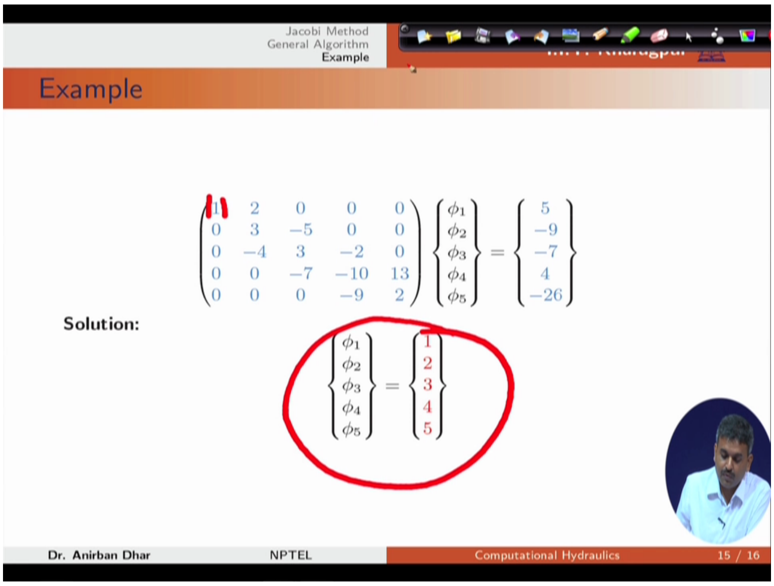
So with thiswe have another example. This is again1, 2, 3, minus 5, like that.

(Refer Slide Time 31:50)



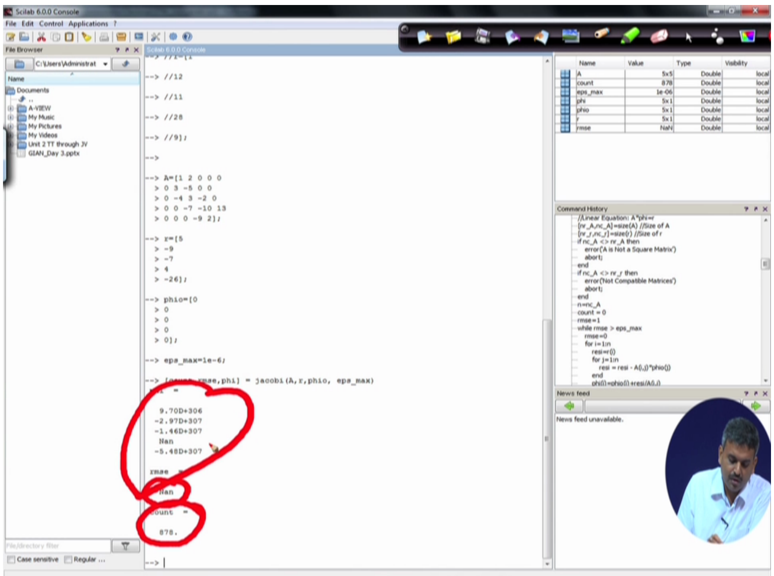
So this is 1, 2, 3, minus 5, 0, 0, minus 3. So in this case clearly it's visible, this is our diagonal term, this is not greater than the off diagonal term. So already we know that the solution is 1, 2, 3, 4, 5. But let us see what value we are gettingout of our calculation process.

(Refer Slide Time 32:35)



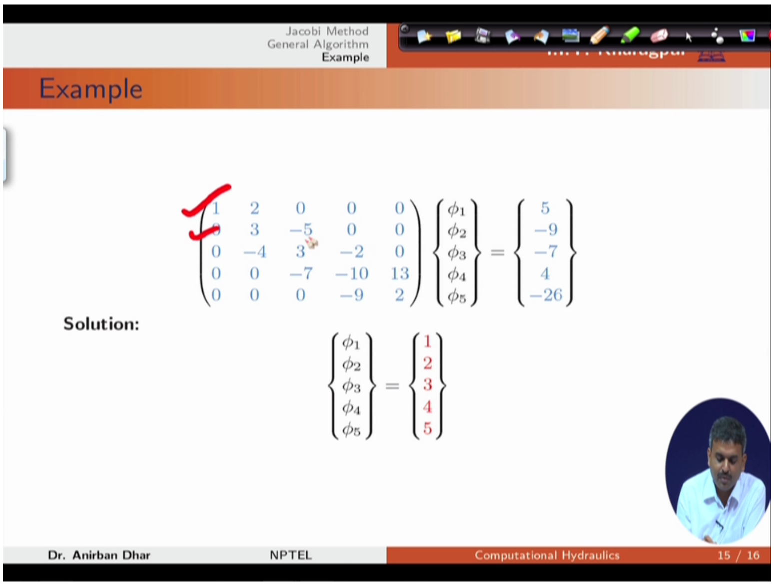
Now if we consider this for calculation, again we run this with evaluate selection option.So we will see that absurd values are coming. These are not physical values and rmse is Nan, not a number. And counter is also 878. Although we are getting some value but these are not meaningful.

(Refer Slide Time 33:15)



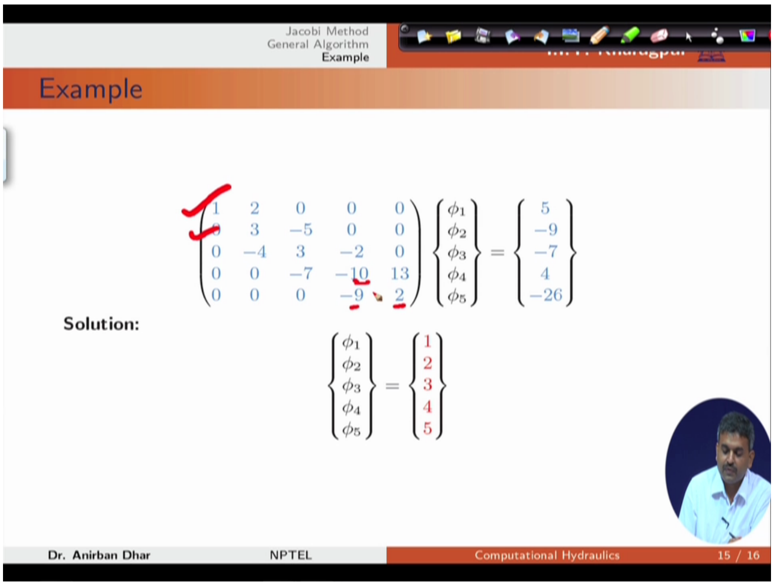
So it's clear thatto get a solution using iterative technique we need to have diagonal dominance for our solution for our matrix. Otherwise this iterative technique is not applicable. We can see that there is clear violation for this one, this one. Because 3 which is less than minus 5.

(Refer Slide Time 33:49)



In this case also this is 4 to 6 which is greater than 3. 3 is our diagonal term. In this case also 10 which is 7 and 20. 20 is greater than 10. Again 2, 9 all the cases we have strict violation.

(Refer Slide Time 34:15)



Sothis iterative method is not applicable for this kind of matrixes. Thank you.