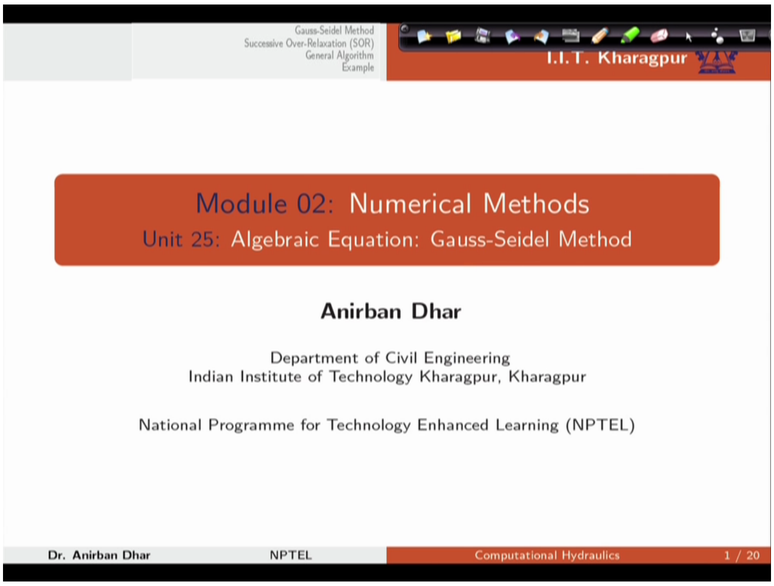
Computational Hydraulics
ProfessorAnirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Lecture 29
Algebraic Equation: Gauss - Seidel Method

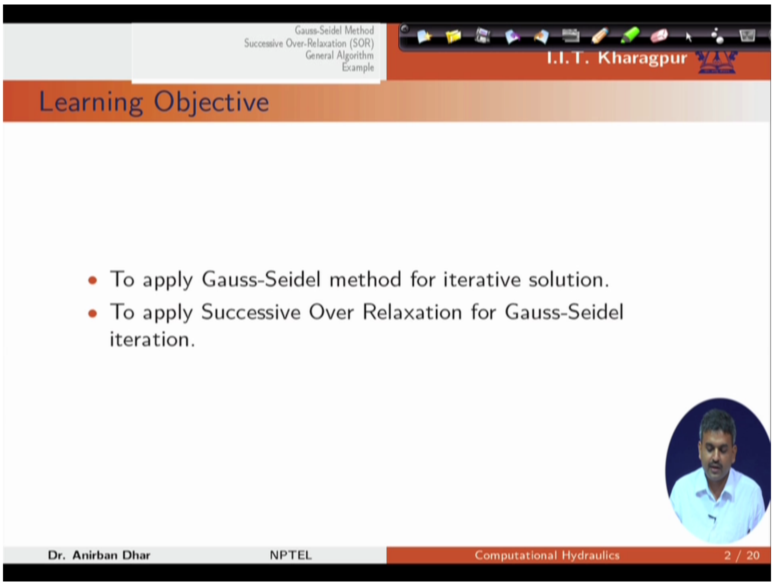
Welcome to thislecture number 29 of the course computational hydraulics.We are in module number 2, numerical methods. And in this particular lecture class I will be discussing algebraic equation, GaussSeidel method.

(Refer Slide Time 00:32)



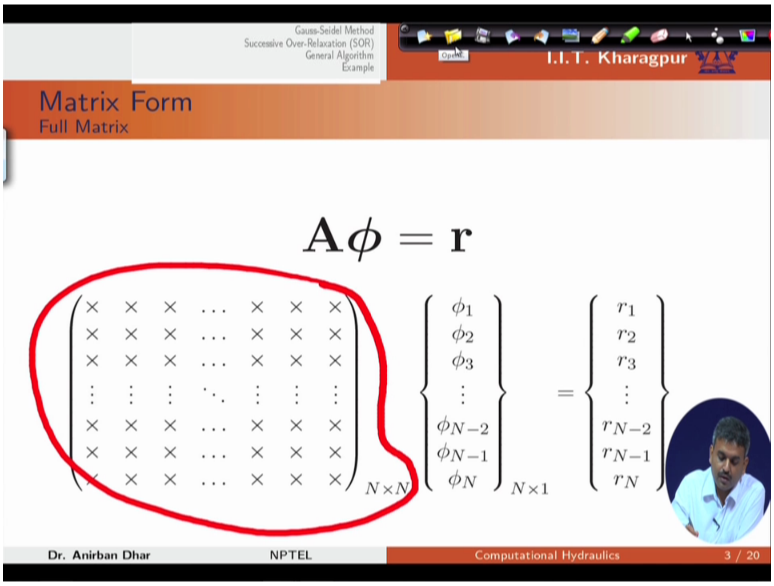
This is again another iterative method like our Jacobian method discussed in unit number 24. Now what is the learning objective? At the end of this unit students will be able to apply Gauss Seidel method for iterative solution. Andthey will be able to apply successive over relaxation for Gauss Seidel iteration.

(Refer Slide Time 01:04)



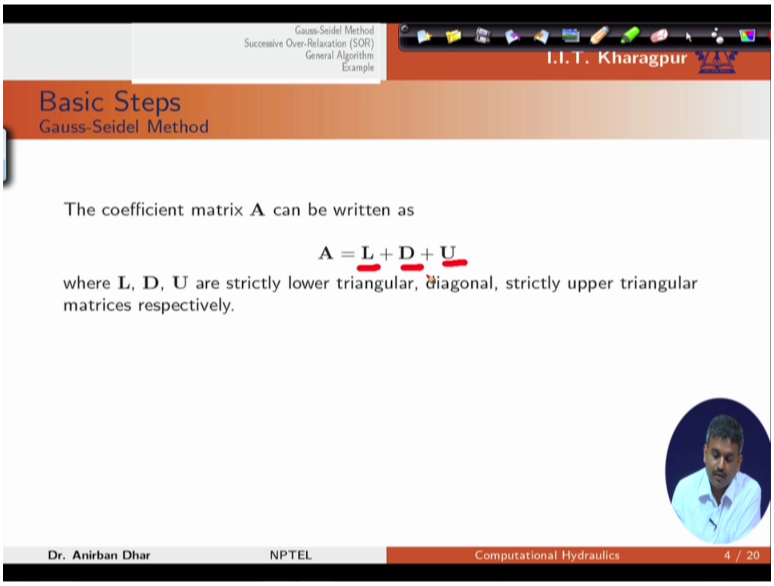
In general like our Jacobian methodwe have our general structure. We are considering the full matrix here. We are notdividing it, or we are not storing it in reduced form.

(Refer Slide Time 01:28)



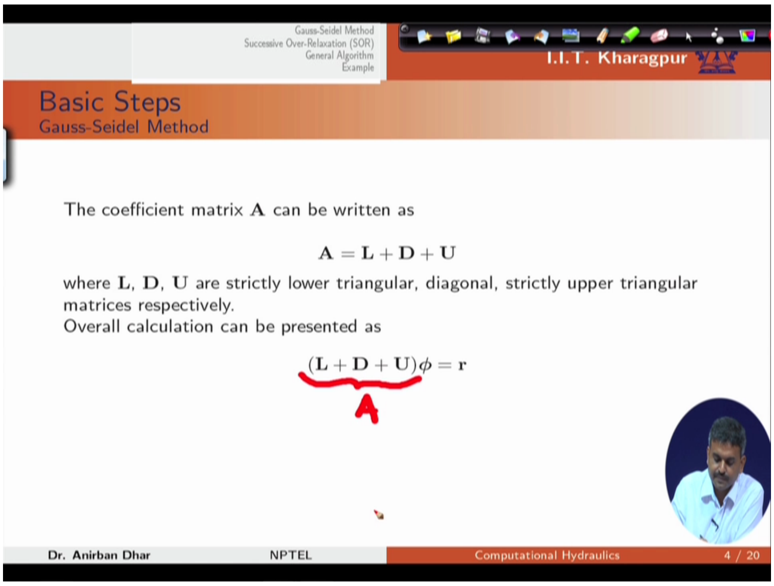
So let us start with the basic steps. Basic steps, this is similar to or first steps are exactly similar to ourJacobian method. We will divide our A matrix into lower, diagonal, upper. This lower one is strictly lower, diagonal is only diagonal terms and upper is with strictly uppermatrix.

(Refer Slide Time 02:03)



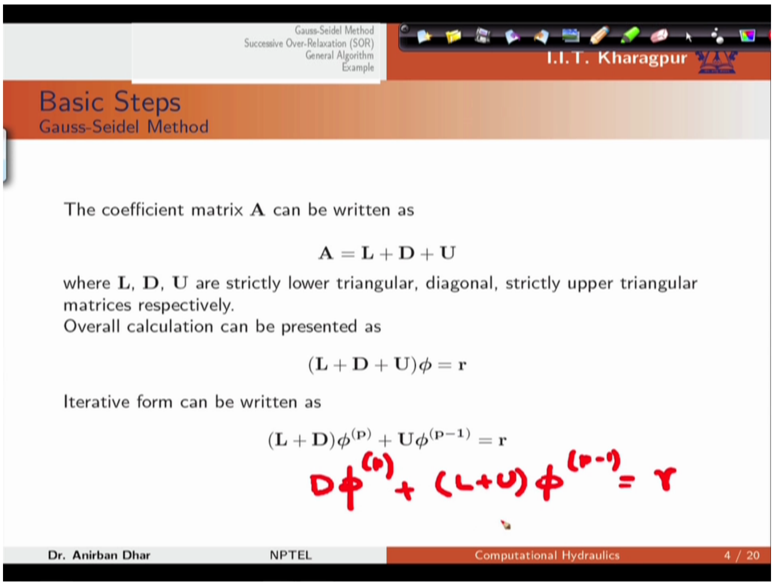
So now if we consider the overall calculation, again LDU this again can be multiplied because this whole thing is A.

(Refer Slide Time 02:16)



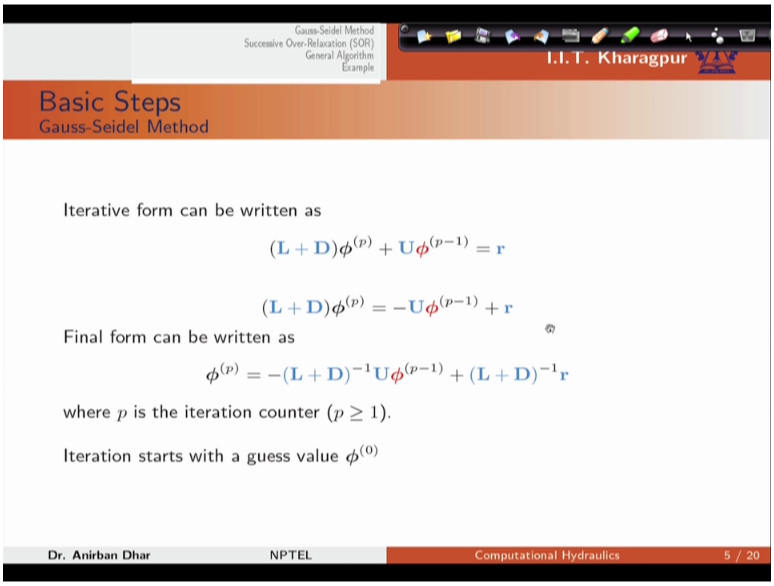
But in this process we can divide this into two parts. In case of our Jacobi iteration we have seen that this is D into phi p plusL plus U into phi p minus 1, this r. This was there.

(Refer Slide Time 02:54)



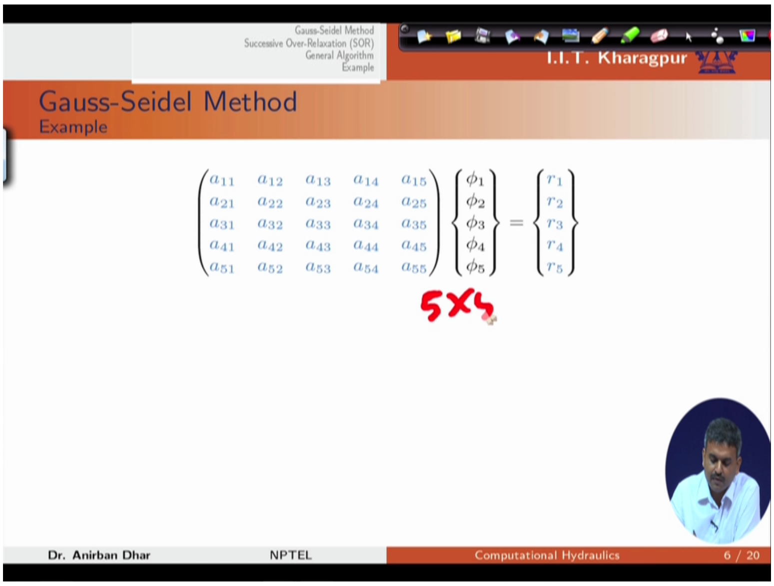
But in this casewe are considering the lower diagonal or lower triangular matrix here. And we are updating it. So L plus D and U is there. So phi again is the available value. And in this process we canwrite the final form, L plus D this inverse into U phi p minus 1, thisterm is there. Like our Jacobi iteration. But the difference is the inclusion of L term which is the lower part or the updated part of the matrix. Now this will be clear with example.

(Refer Slide Time 03:51)



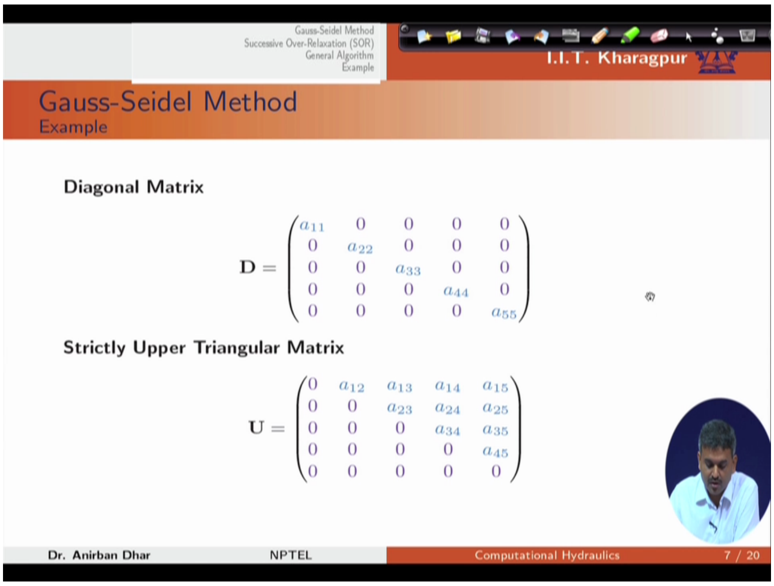
So in Gauss Seidel we let us say that we will apply the same example with 5 by 5 matrix.

(Refer Slide Time 04:07)



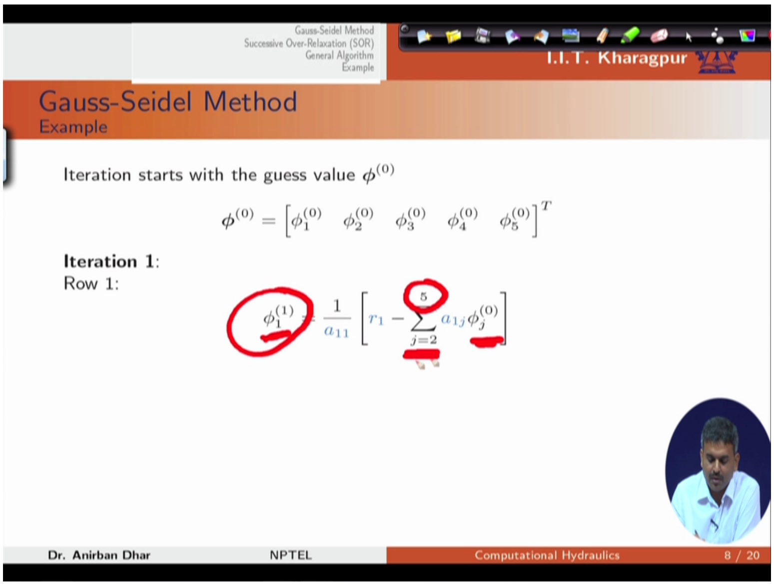
Now in this case again we can write our coefficient as lower, strictly lower, diagonal, strictly upper diagonal. Up to this we are not disturbing any element within the A matrix. And we will not disturb.

(Refer Slide Time 04:29)



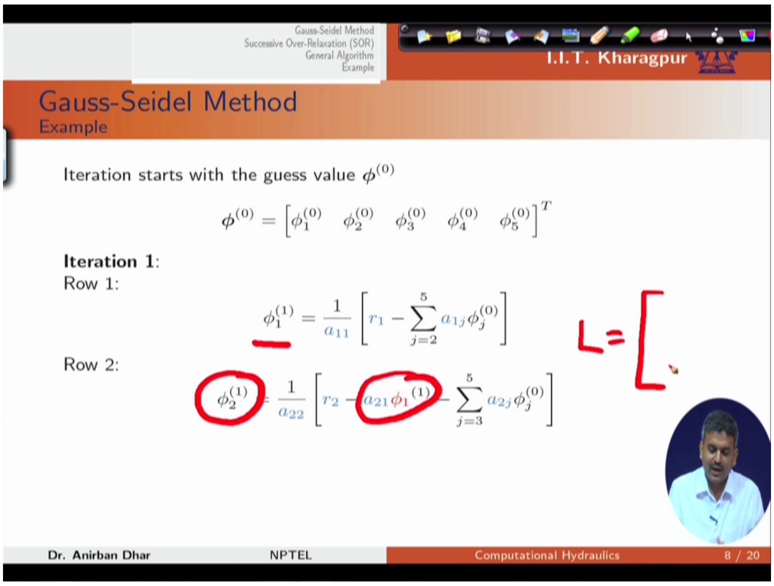
Iteration starts with guess values. Now in iteration 1,this is clear that for row 1, whatever available value starting from 2 to 5 is available from previous iteration. So except this diagonal term we are calculatingthis updated value based on previous time level value.

(Refer Slide Time 05:06)



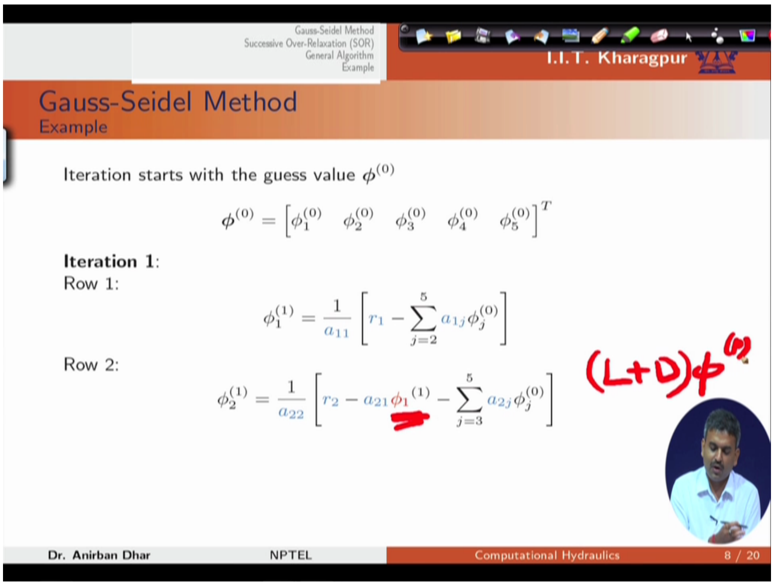
But interesting feature is there for row 2. Now we have this phi2 calculation and with phi2 calculation we can omit this 2 term, but we have already one updated value available for phi 1 from row 1. So this comes directly from the iteration 1 or present iteration level. So r2 minus a21. So a21 is the coefficient of our lower triangular matrix.

(Refer Slide Time 06:00)



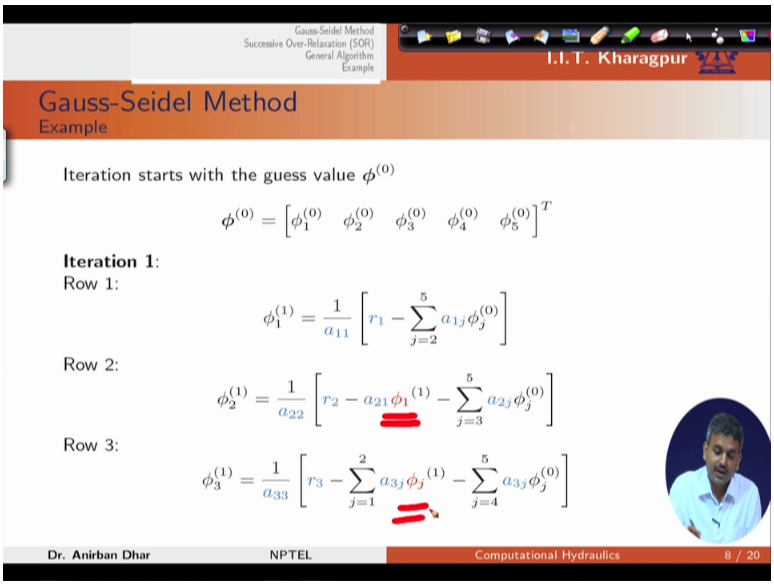
It's clear, that's why we are including the lower triangular matrix and you can see that we are writing it withpresent time level not from the previous time level. That's why L plus D into phi p was written in our calculation process.

(Refer Slide Time 06:26)



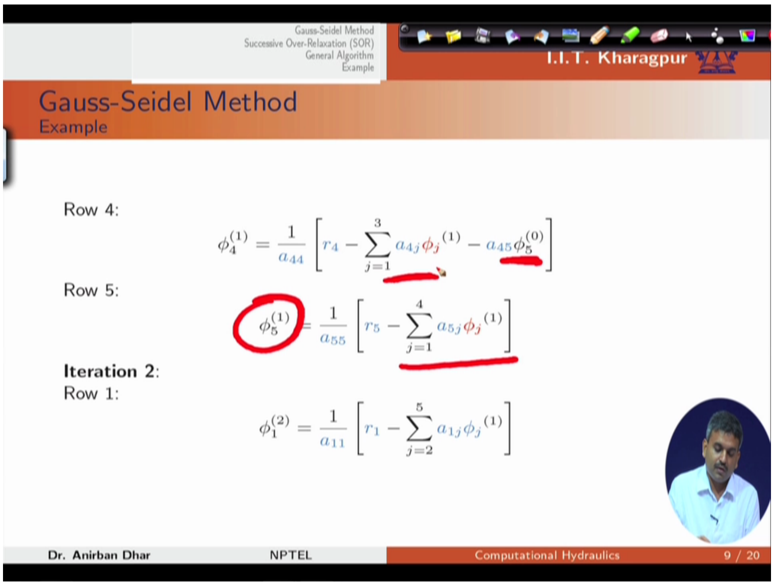
Now if we write the third one, again we have two values available for updating phi3. This is j 1 and j 2. And starting from j 4 to 5 we are utilizing previous time level value. Interesting part is that phi1 we have changed color, phi j we have changed color, because we are getting updated values and we are utilizing the updated values in the calculation. So red colored indicates that we are using updated values within the calculation process.

(Refer Slide Time 07:11)



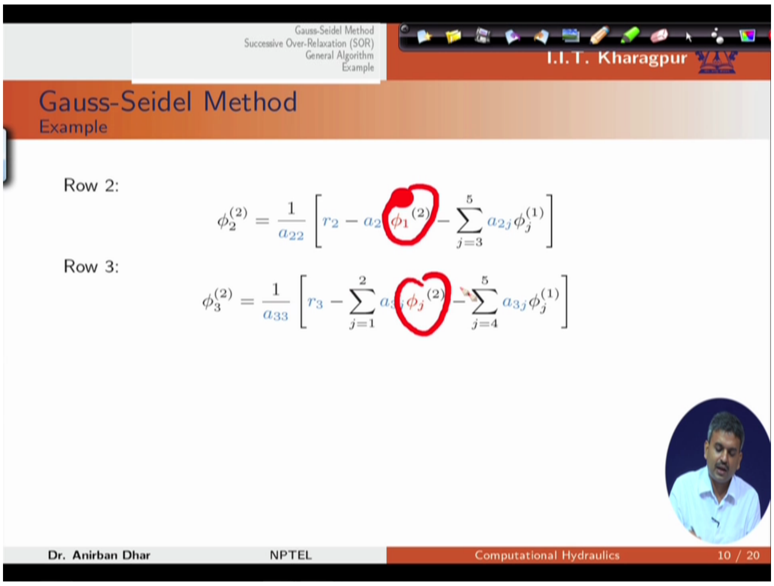
Butthe blue values we are not changing the coefficients on the right hand side vector.Gauss Seidel example again fourth, fifth. In the fifth one except this fifth one all values are updated ones. So starting from 1 to 4 we have all values updated ones. In case of fourth one, except one, all are updated ones.

(Refer Slide Time 07:52)



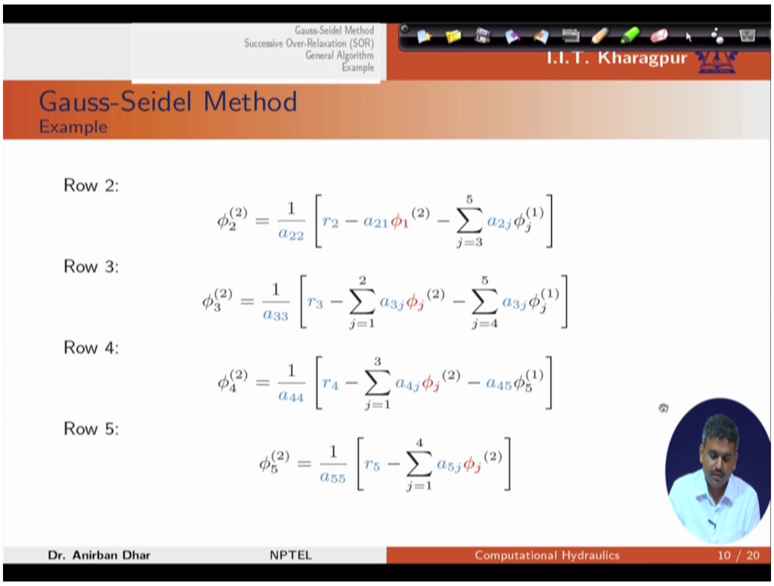
So in iteration 2 again if we consider the same thing you will find that this is same.We are using the presentavailable value at the present iteration level itself.

(Refer Slide Time 08:11)



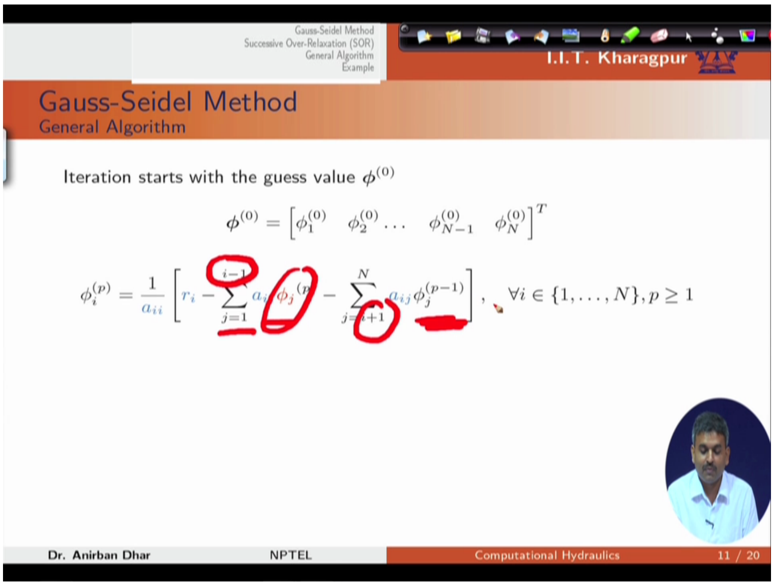
So anyway we are providing updated information to the algorithm. So that's why obviously we expect that the performance will be better compared to our Jacobi technique where we are utilizingthe previous iteration level value or guess valueat the beginning steps. So row 3, row 5 againthis approach is similar.

(Refer Slide Time 08:48)



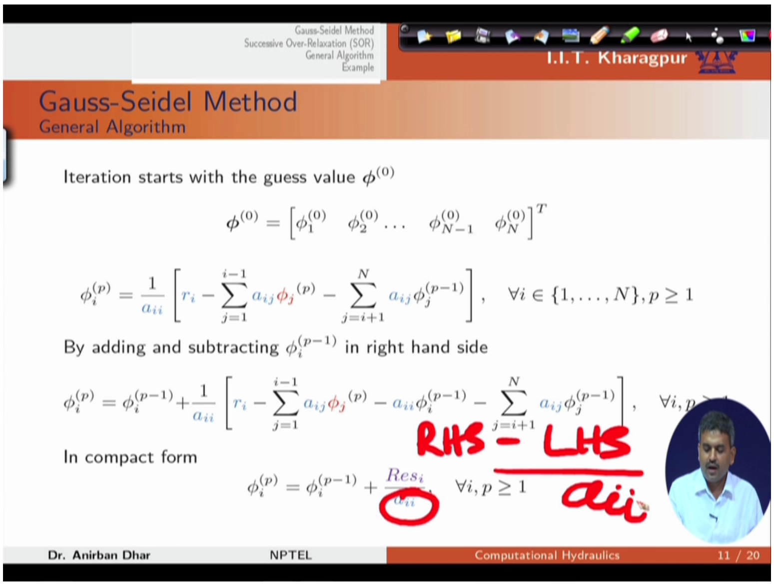
Now if we generalize the algorithm this becomes like this.Except that ith term that means up to i minus 1 we have already calculated the updated value. So j starting from I minus 1, we are using updated value. And from i plus 1 to N we are using previous time level value.

(Refer Slide Time 09:16)



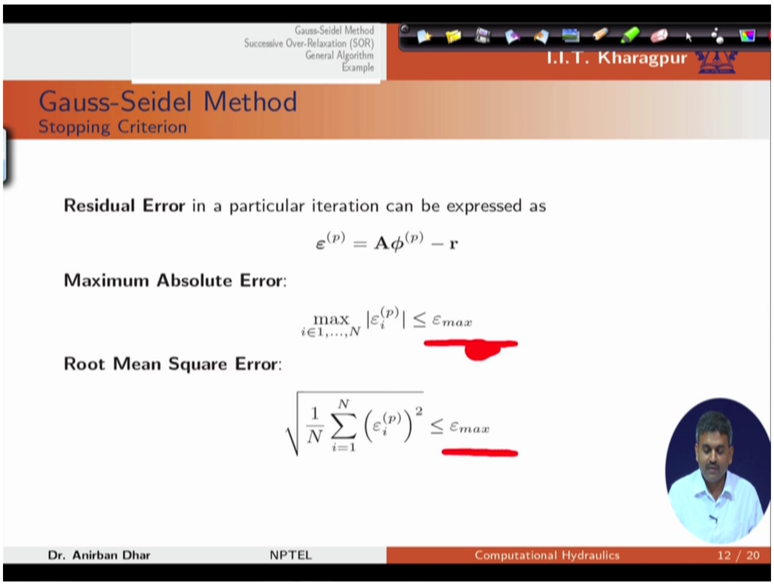
Now like our previous Jacobi method we can add phi i p minus 1. So we can just add phi i p minus 1. And we can write this as residual again. So whatever value is available either in updated form or in non-updated form, we can calculate the right hand side minus left hand side. So this is our residual and divided by coefficient of the diagonal term. This is a11.

(Refer Slide Time 10:01)



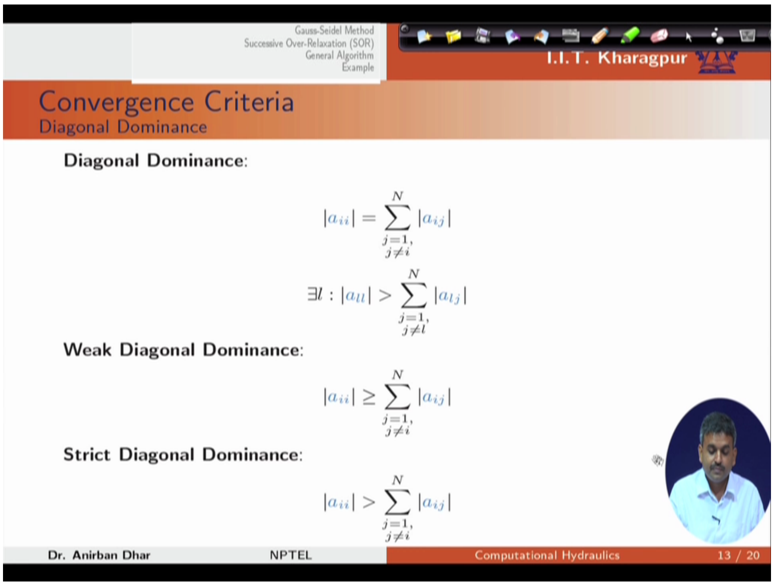
So we can use this concept where we are updating the variable valuesat the current time steps. So obviously convergence will be better. Like our previous Jacobi iteration we can utilize residual error for this one.Maximum absolute error or RMSE for this purpose where epsilon max, these are actually allowable epsilon values.

(Refer Slide Time 10:47)



Now in this case we need again this diagonal dominance. Without diagonal dominance these convergence are not possible.

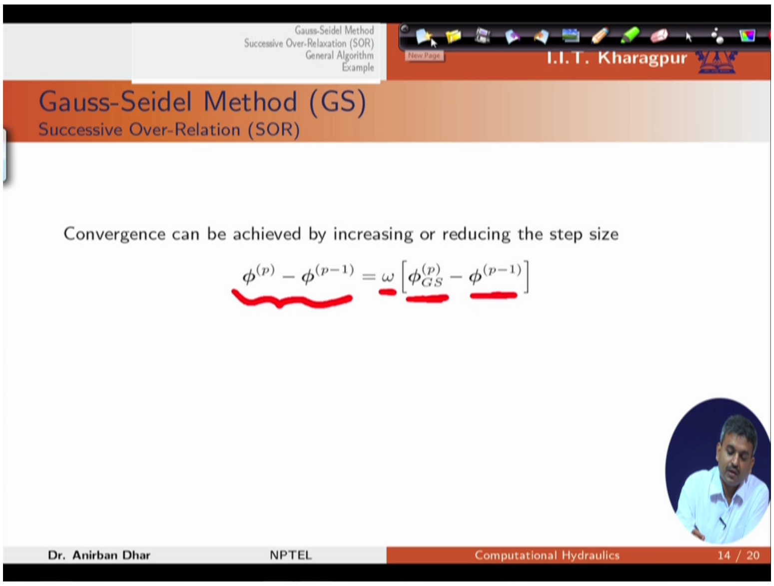
(Refer Slide Time 11:04)



Another method we can apply that is called as successive over relaxation. So we can apply this method and in successiveover relaxation we can control the update using Gauss Seidel step. So let us say that this is phi p minus phi p minus 1. So this omega is there, into this phi Gauss Seidel step whatever we are getting minus phi p that is old value, this step.

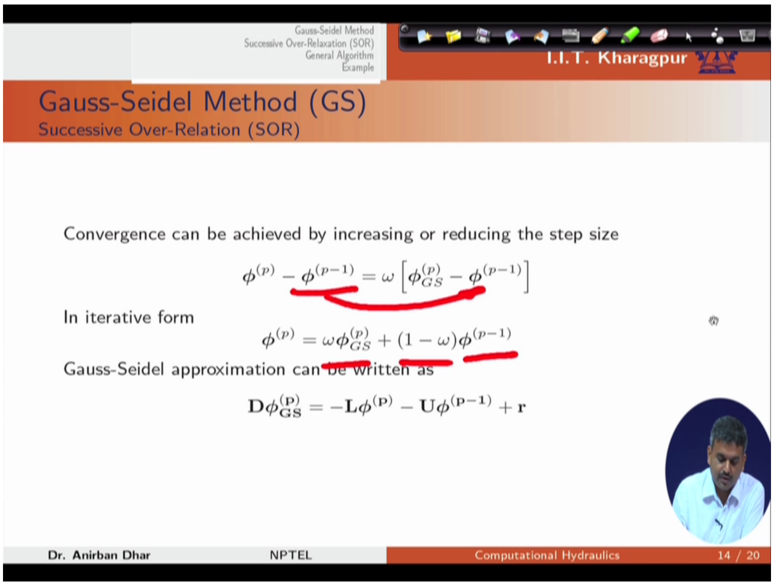
So we will take smallest step compared to theoriginal smaller or larger stepcompared to the proposed modification from the Gauss Seidel step.

(Refer Slide Time 12:24)



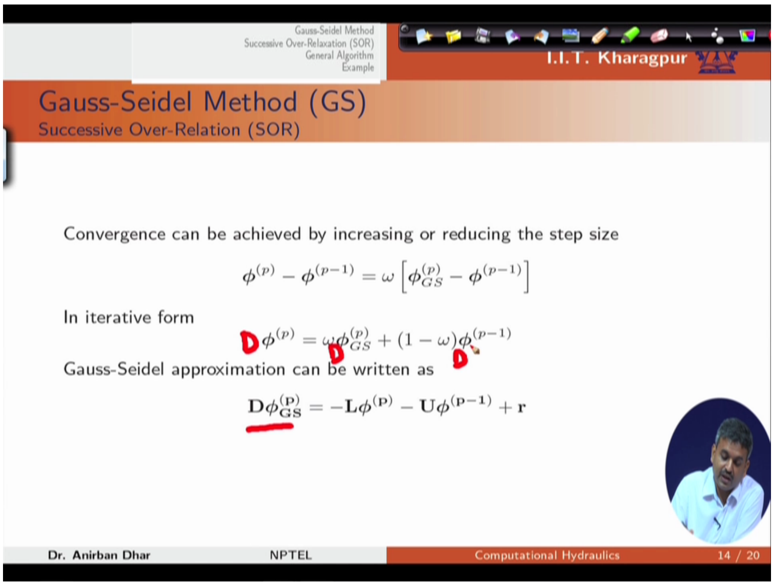
So if we directly write it in iterative form, so by transferring this part in the right hand side we can write omega into GS plus 1 minus omega into phi p minus 1. So in this case we can use the Gauss Seidel approximation.

(Refer Slide Time 12:55)



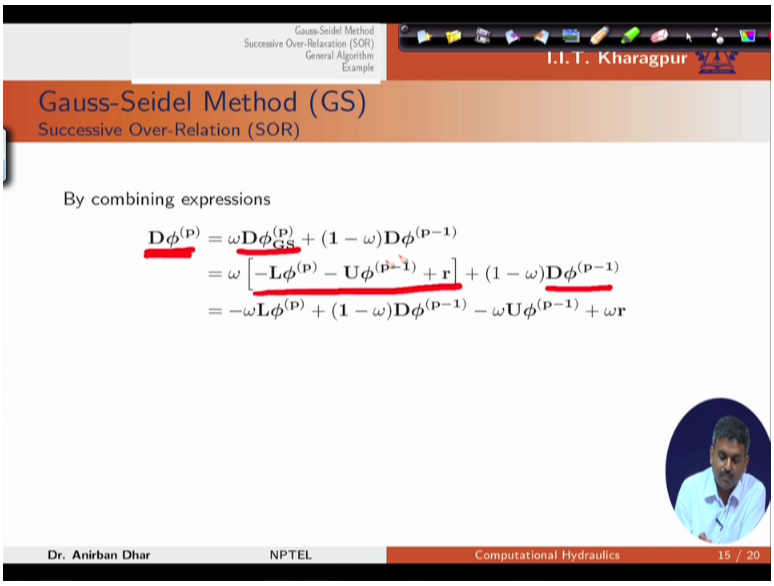
In Gauss Seidel approximation what we are getting? We are getting this D plus L. So right hand side we are writing this for lower triangular one, we are transferring. Now in this case if we multiply D on both sides.

(Refer Slide Time 13:37)



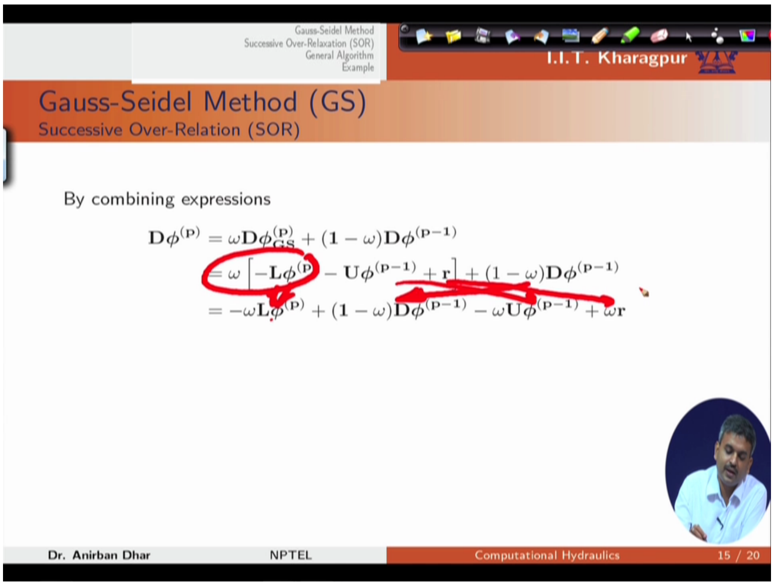
Then we can get this form. This is actually our desired level into D. But this D is our Gauss Seidel step like this. So omega into this Gauss Seidel step plus 1 minus omega D phi p minus 1. So if we rearrange this, what we are getting?

(Refer Slide Time 14:11)



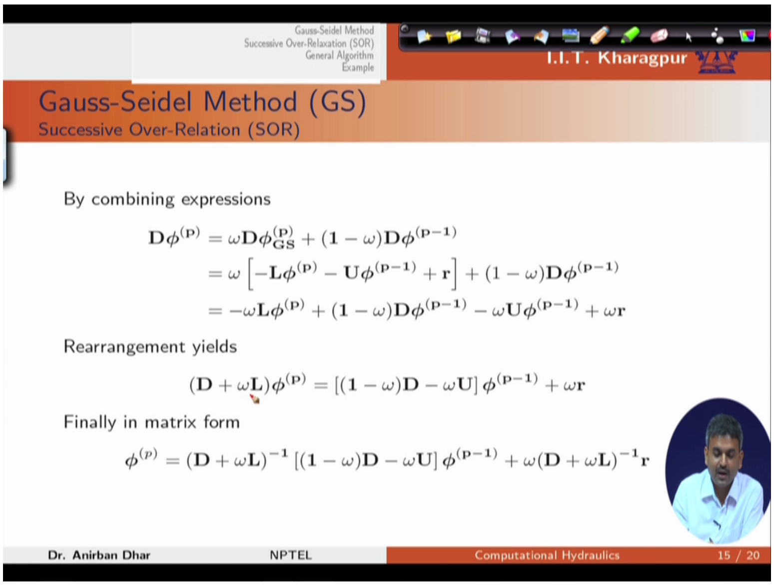
From this two terms we are getting this first term. This is our diagonal one and this part is coming from here and omega into r, this is our constant term. Although omega value may vary. Butwe can use a fixed value for iteration.

(Refer Slide Time 14:30)



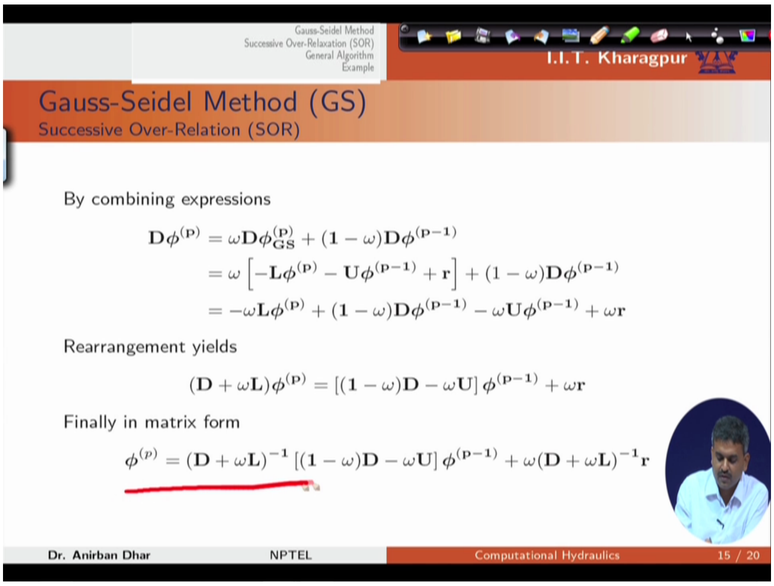
So rearrangement is this one. We can change the side as per our desired one. D plus L that inverse we need to utilize for Gauss Seidel. But in this case we are not using this one. We are multiplying this omega.

(Refer Slide Time 14:57)



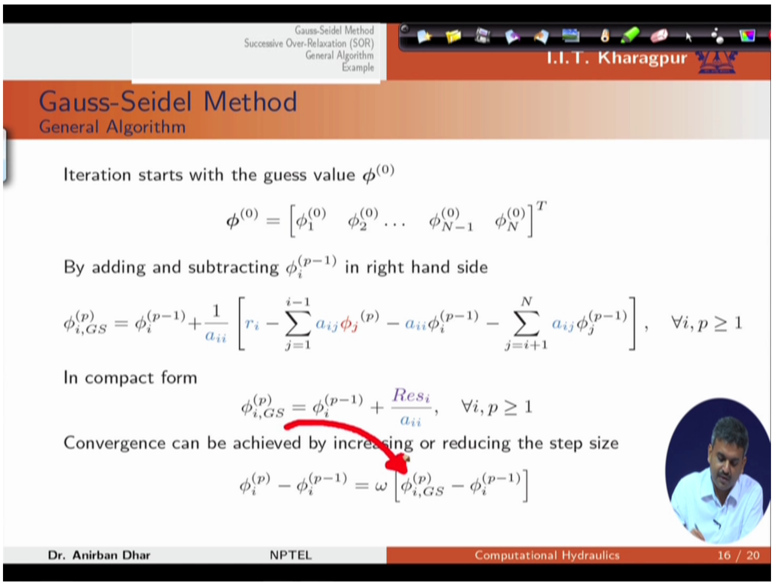
We are bit cautious, we are reducing or increasing steps. So finally in matrix form this can be written like this.

(Refer Slide Time 15:08)



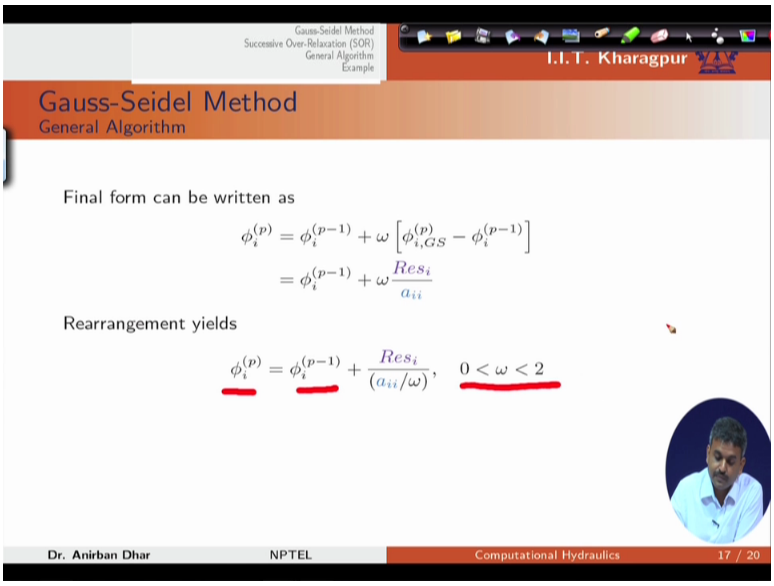
But implementation wisewe already have information about this system. Now this is in compact form, from our Gauss Seidel. This is our actually Gauss Seidel step. Now with this if we proceed and if we writein this format so that we can achieve convergence by increasing or decreasing the time step, we can simply input this value here.

(Refer Slide Time 15:52)



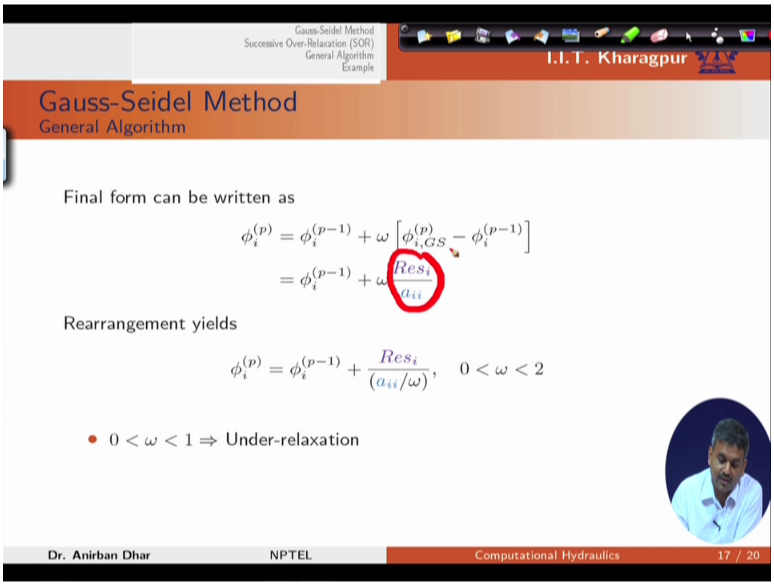
And rewrite this. So final form is omega into residual i. So a ii, this is our general form and omega is written here. So rearrangement is that phi i equals to old value plus residual i, a ii, omega. So in a way this value omega varies between zero to 2.

(Refer Slide Time 16:40)



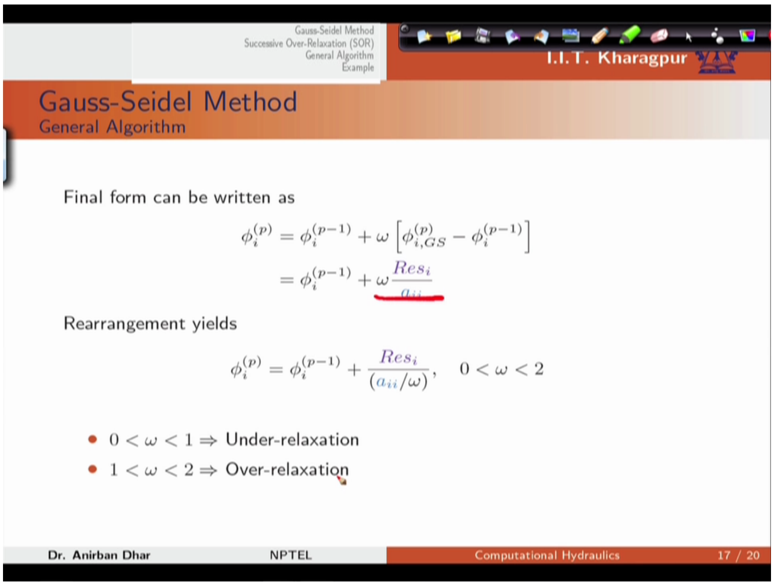
Now if we have zero to 1, we call it as under relaxation. That means whatever value is coming here, we are reducing that valueunder relaxation.

(Refer Slide Time 17:00)



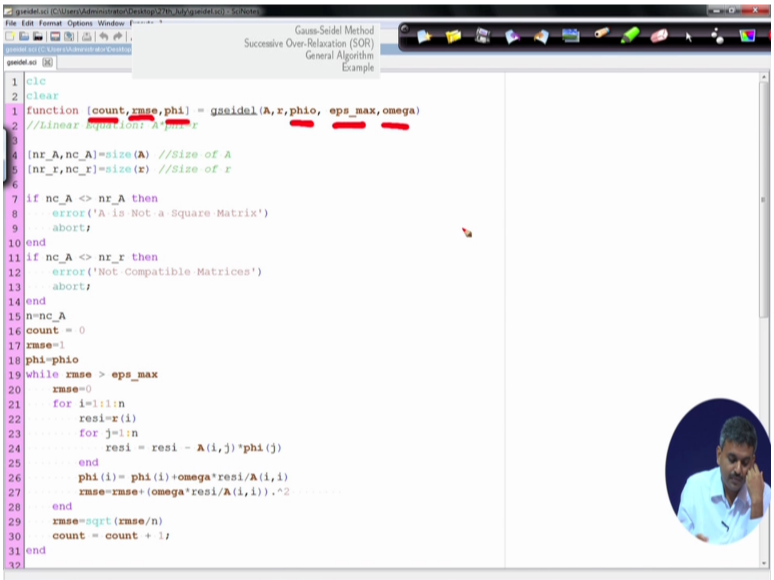
And over relaxation if it is between 1 to 2. But virtually what is happening, we are playing with the diagonal term. If we rewrite this thing like this, this is a ii divided by omega. So in case of under relaxation we are actually increasing the diagonalterm. And in case of over relaxation we are reducing the diagonal term. So by changing this omega values we can control the convergence for Gauss Seidel over relaxation as our method.

(Refer Slide Time 17:48)



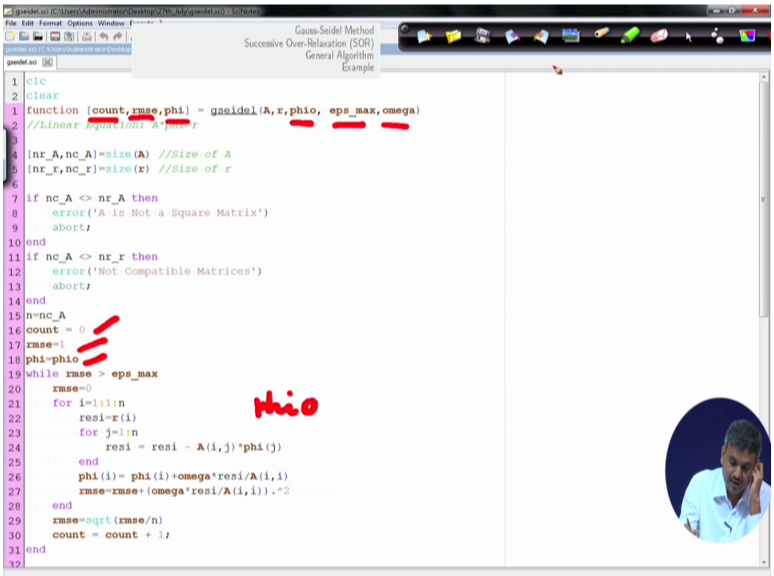
Now let us consider our standard example in this case. In this caseif we have Gauss Seidel up to this. This is general, we are again utilizing count,rmsc, phi, phi initial guess. So epsilon max, omega. Omega means now we need to provide this Omega value for this case.

(Refer Slide Time 18:33)



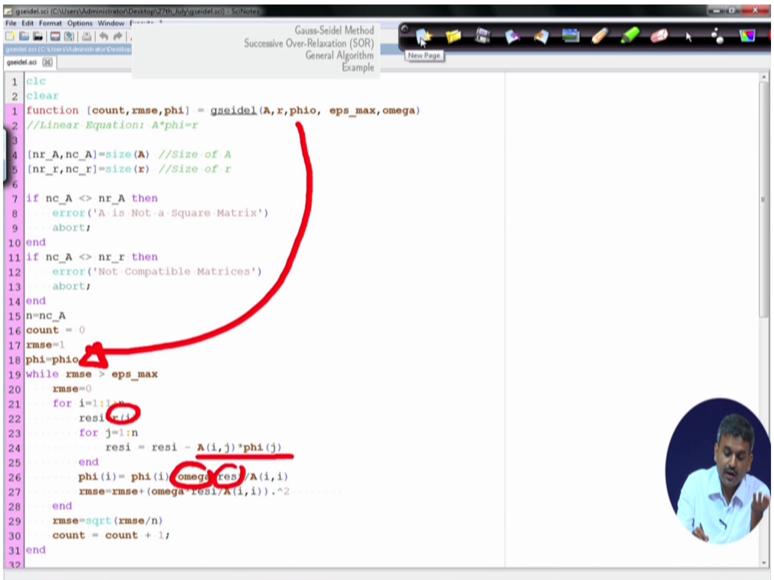
Now this is count equals to zero, rmsc equals to 1 to execute this while loop. And phi is equals to phi o. That means previously in Jacobi iteration we were using phi o or old values only.

(Refer Slide Time 18:51)



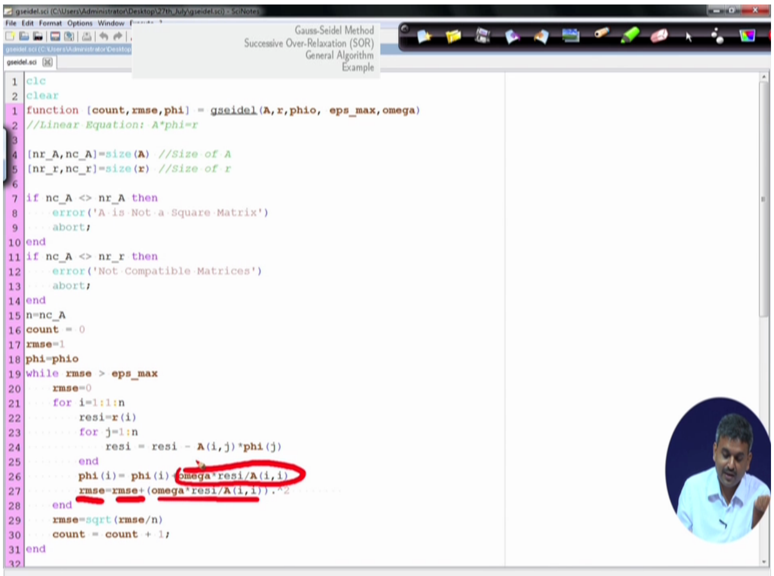
Now whatever value is updated we will directly try to utilize it. So that's why I have transferred these values here directly in phi. Now I am directly updating this phi. So updating and we are multiplying this omega, this residual, residual i again starting with ri minus aij and phi j. And this is for all j. That means whether it is for lower or upper triangular matrixwhatever maybe the coefficient. Whatever value is availablewithpresent or pastiteration level, we are utilizing that information.

(Refer Slide Time 19:39)



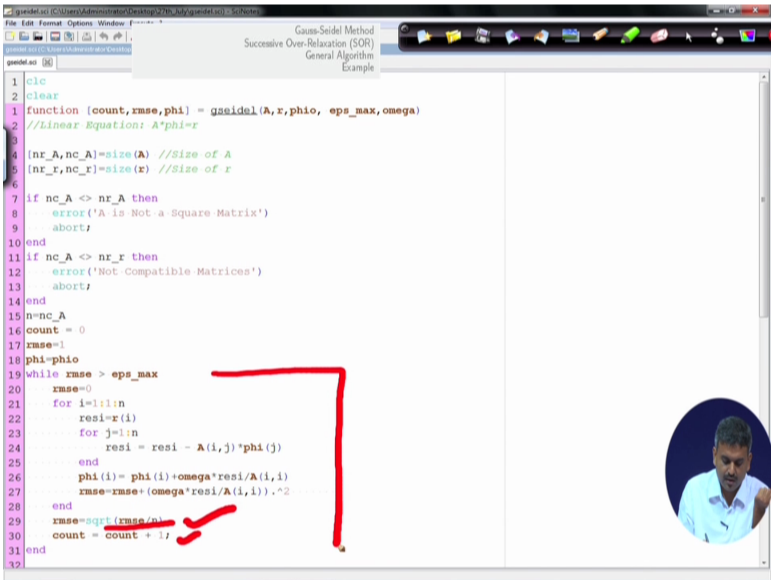
Nowwith this I am also calculating rmsc. Rmsc again, omega because this is actuallythe increment that we are giving between 2 iterations.

(Refer Slide Time 19:56)



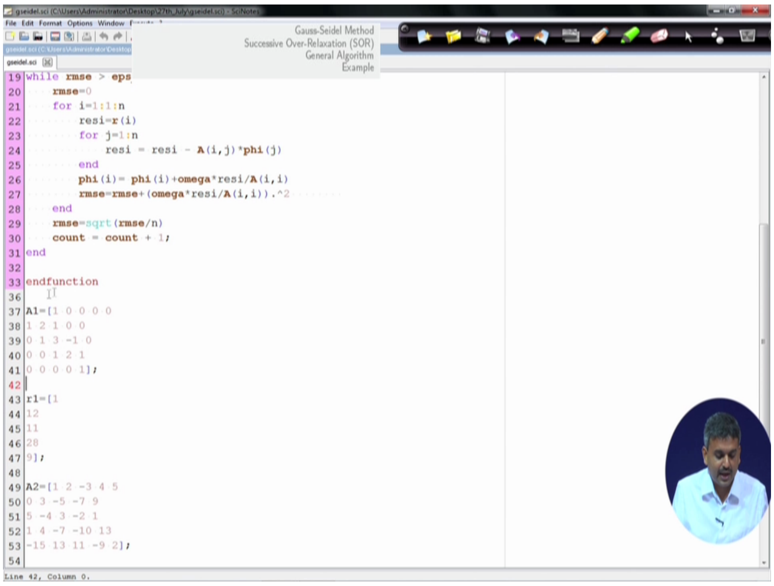
So again thisrmsc equals to rmsc divided by n and we're taking square root. So this is actual calculation and count equals to count plus 1. And with this, this while loop ends.

(Refer Slide Time 20:13)



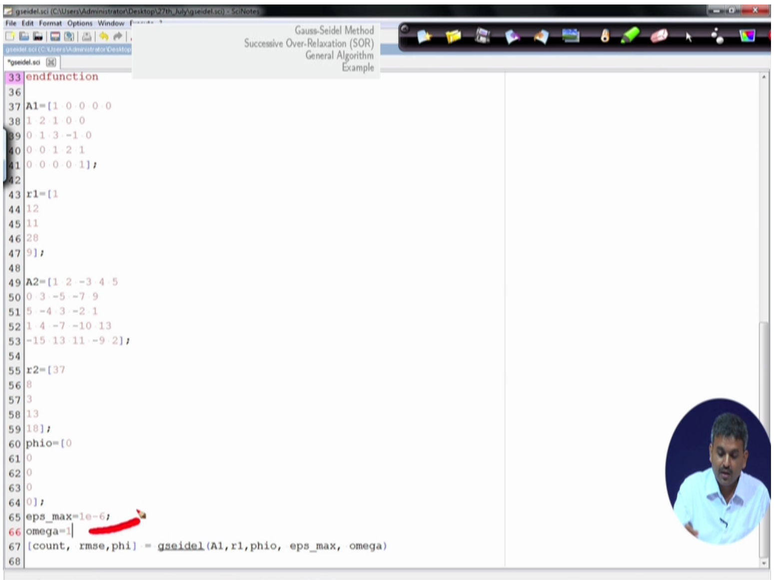
And this is our algorithm. Compared to our direct solvers like LU decomposition or Gauss elimination, we need to write less numbers of lines for this one.In case of this our standard problem, we canuse this concept of our case. And in this case let us say that A1, this is for diagonally dominant our standard matrix.

(Refer Slide Time 21:01)



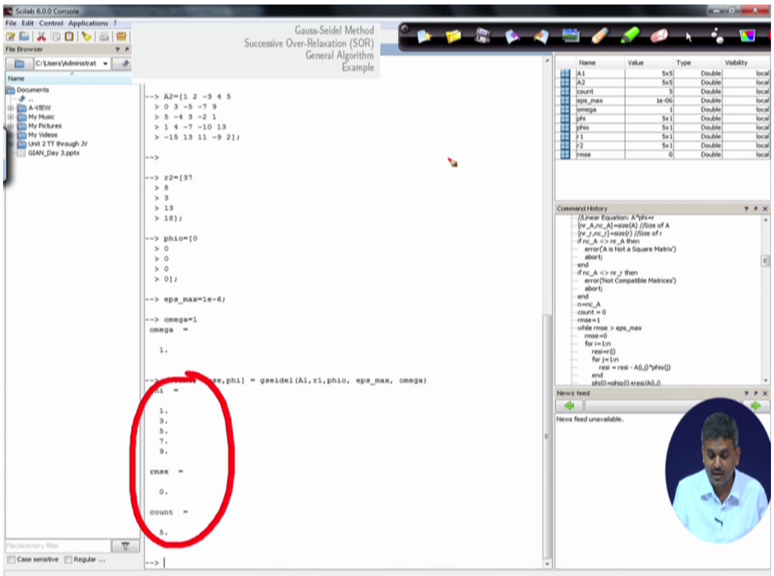
So let us start with A1 and r1. Now omega equals to 1. Omega equals to 1 means we are at Gauss Seidel step. We are not using SOR that means over relaxation or under relaxation. So in this case omega is 1, this means we are at exactly Gauss Seidel step.

(Refer Slide Time 21:40)



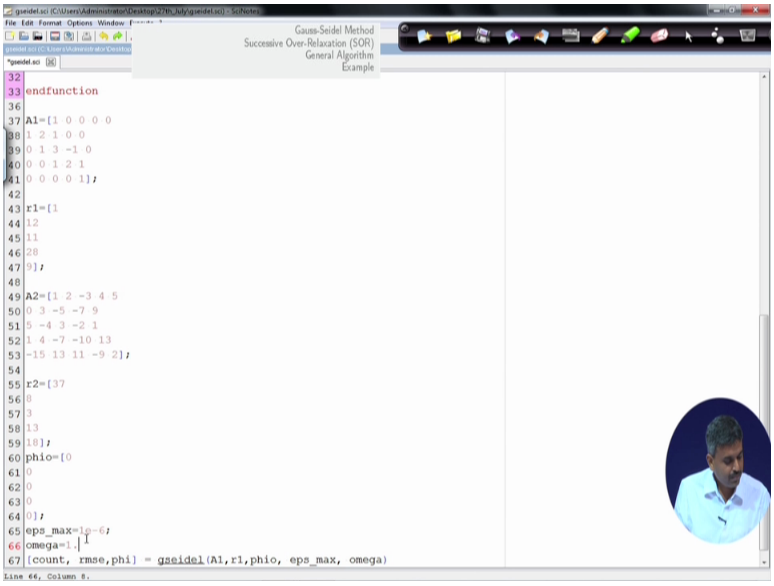
So let us take this phi o is our initial value zero. Epsilon max is 1 into 10 to the power minus 6. If we calculate thiswe will exactly get this 1, 3, 5, 7, 9. This is our exact solution. And interestingly rmsc value is coming close to zero. And with five iterations only we are getting this thing.

(Refer Slide Time 22:20)



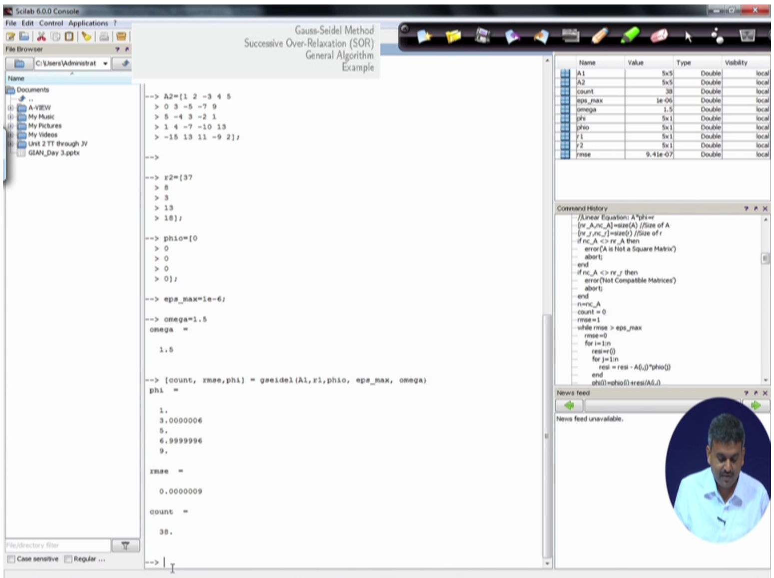
And if we consider our under relaxation, let us say or over relaxation this one, point 5.

(Refer Slide Time 22:40)



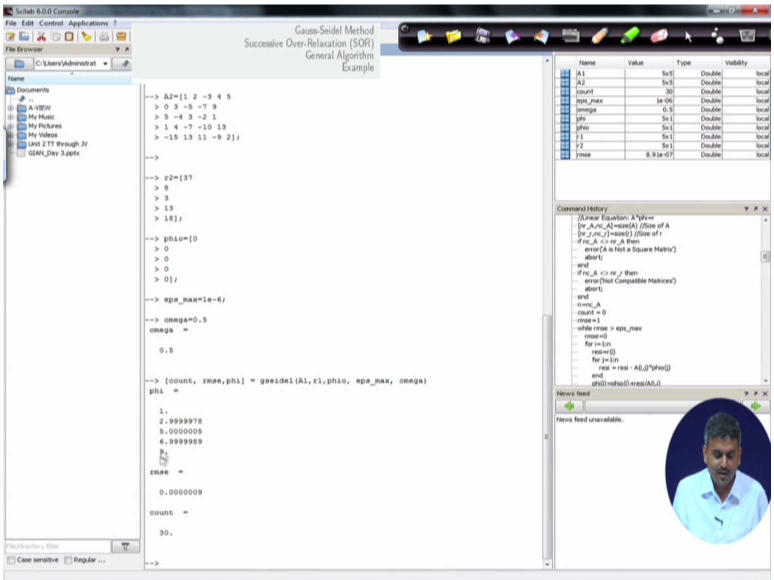
Then what is happening here? If 1.5, we need 38 iterations.

(Refer Slide Time 22:53)



And if we havepoint 5 that is under relaxation,then we need 30 iteration for this one.

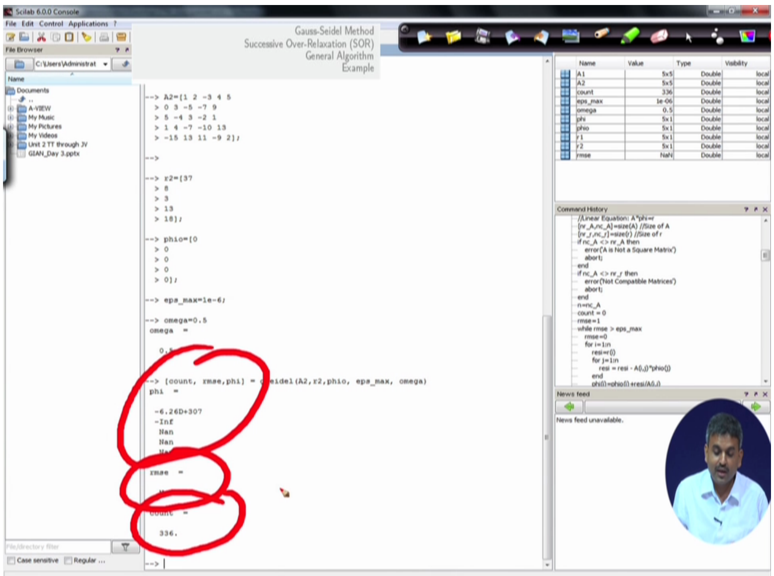
(Refer Slide Time 23:10)



And this values are closed. For this one within five iteration with Gauss Seidel we are getting the solution. But if you are having large matrix and sparse matrix we need then the utility of this approach will be very much visible. Now if we consider our second matrix which isnot diagonally dominant and we utilize this algorithm. That means this is a2 and r2. Only change is in terms of a2, r2.

We are keeping the initial value epsilon max and omega same. Soif we do that we can easily see that solutions are not converging. Rmsc equals to Nan. Although these many iterations we are not getting converge solution for this one.

(Refer Slide Time 24:26)



So for that our diagonal dominance is very much required for this iterative system. Otherwise we will not get solution from this system of equations. So with this we can end our successive GS SOR methods. And next lecture we will devote with nonlinear iterative techniques. Thank you.