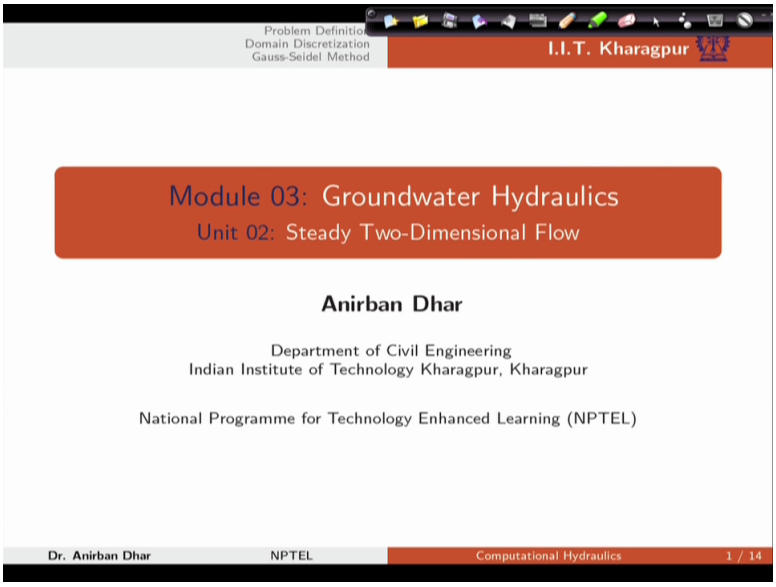
Computational Hydraulics
Professor Anirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Lecture 32
Steady Two-Dimensional Flow

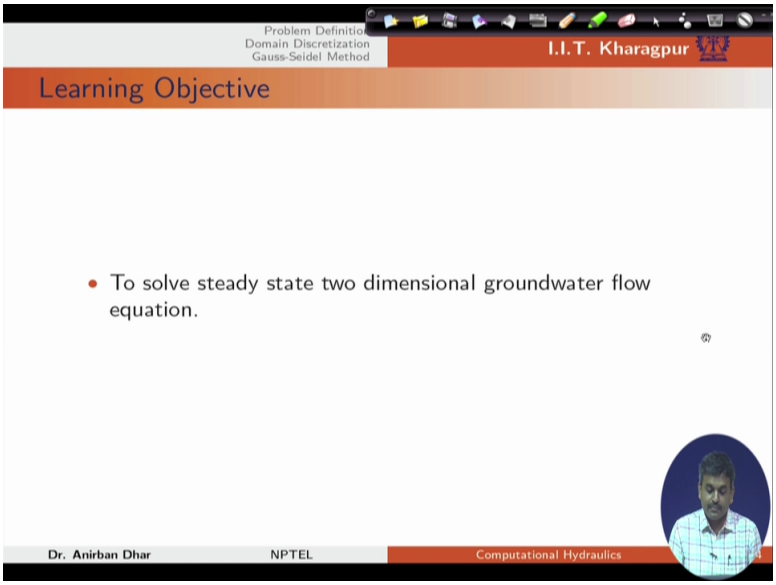
Welcome to this lecture number 32 of the course computational hydraulics. In our last lecture class I have discussed one dimensional steady groundwater flow equation. In this particular unit I will be discussing steady two dimensional groundwater flow.

(Refer Slide Time: 00:37)



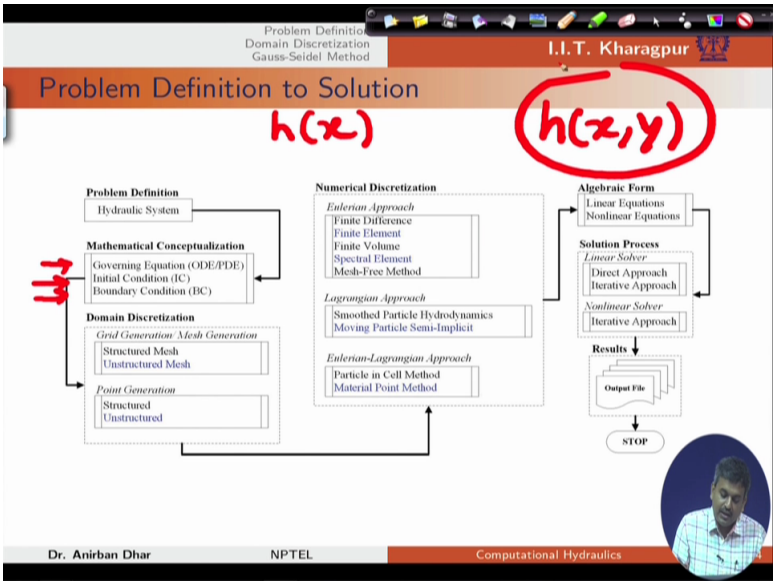
So learning objective for this particular unit. At the end of this unit students will be able to solve steady state two dimensional groundwater flow equation.

(Refer Slide Time: 00:54)



Again problem definition to solution, if we conceptualize our ground water problem as 2D problem then the governing equation will change. So obviously in our previous case we have conceptualize the governing equation in terms of one dimensional flow. But if we say that h is function of xy, there is variation in h over the xy domain or spatial domain but not in Z direction. In this case we are considering two dimensional flow. So we need to see what will be the variation in xy. So obviously we need to find out this h xy in this case.

(Refer Slide Time: 01:57)

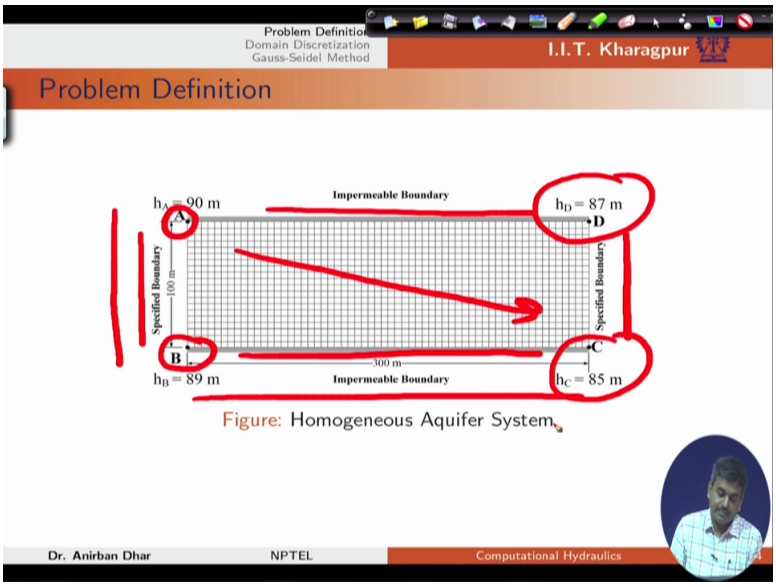


So let us see with our concepts that we have already covered. Let us define this problem. Problem, this is homogeneous isotropic aquifer system. In case of homogeneous isotropic

aquifer system obviously the differential equation will be Laplace equation. And in this case let us say that this is 300 metres length, a hundred metre on this side and left hand boundary is specified, right hand boundary is specified, top boundary is impermeable, bottom boundary is again impermeable.

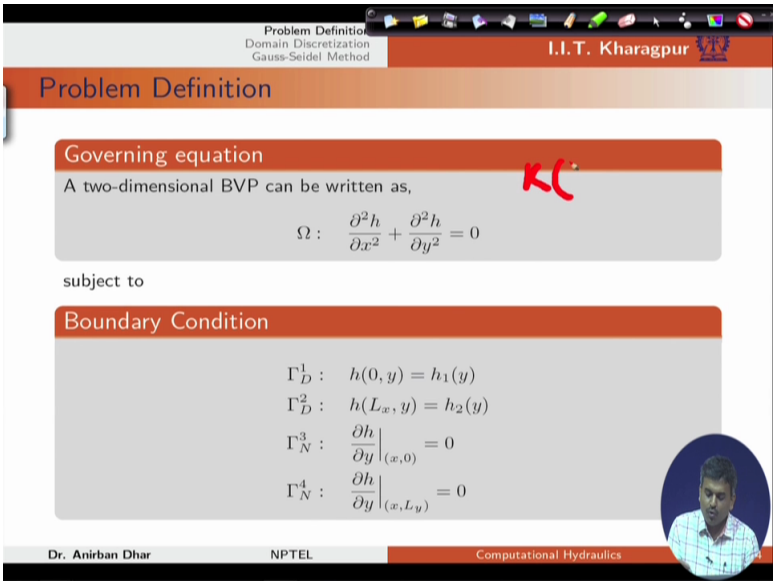
And this specified value at the end points point A is 90 metres, at point B it is 89 metres, point C this is 85 metres and hD which is 87 metres. So obviously the flow will be from this direction to this direction because this is the lowest point and this is the highest point. Obviously there will be some amount of flow variation in this case.

(Refer Slide Time: 03:31)



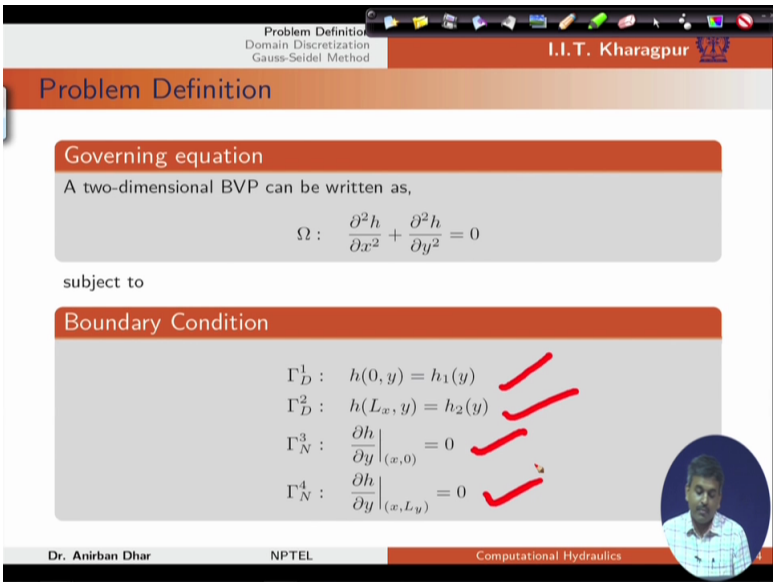
Now we should be able to (visua) visualise the problem based on over on two plots. Let us see what are the equations we need to use? This is two dimensional boundary value problem that we have discussed in terms of variable phi. This is same because it is homogeneous (itro) isotropic aquifer system so there is no variation in K. Obviously K is constant so we should not consider K in this case because of this conceptualization.

(Refer Slide Time: 04:04)



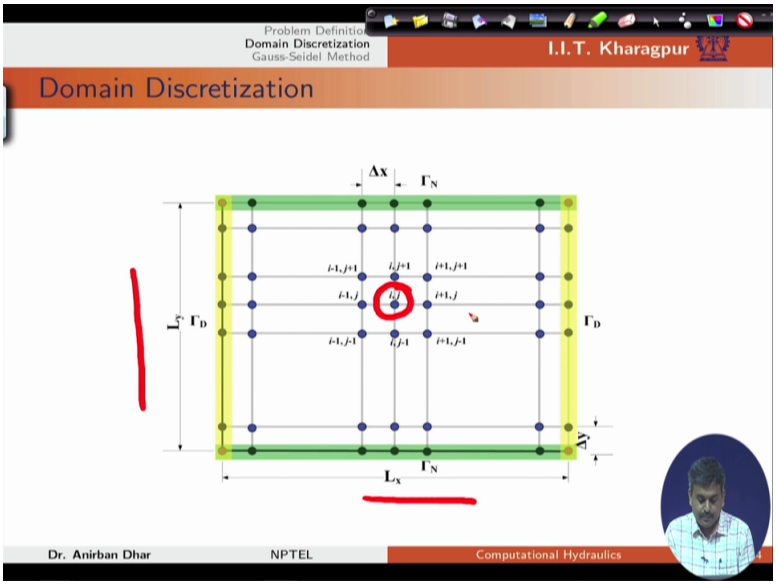
On the left hand side let us say there is specified, right hand side specified, top boundary, bottom boundary and top boundary we have impermeable boundary conditions.

(Refer Slide Time: 04:29)



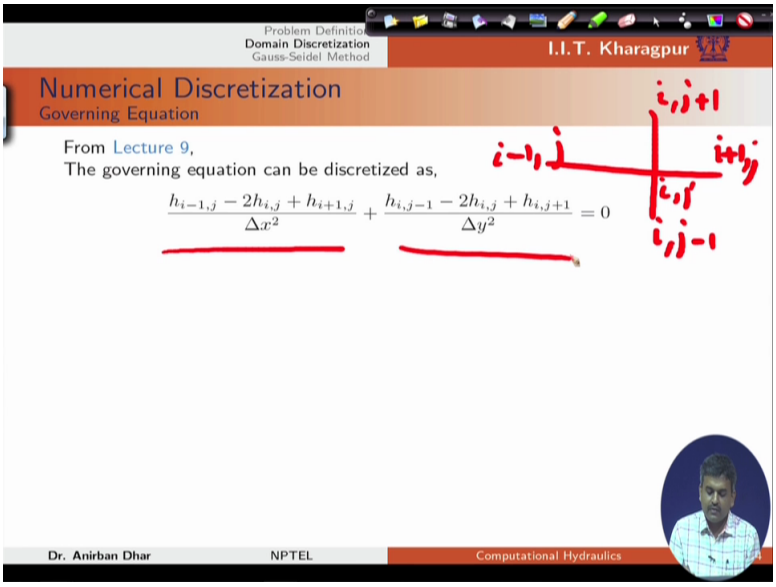
Domain discretization, this is Lx, this is Ly, this is del y, del x and if we consider any general interior nodes ij, these are the neighbouring nodes.

(Refer Slide Time: 04:51)



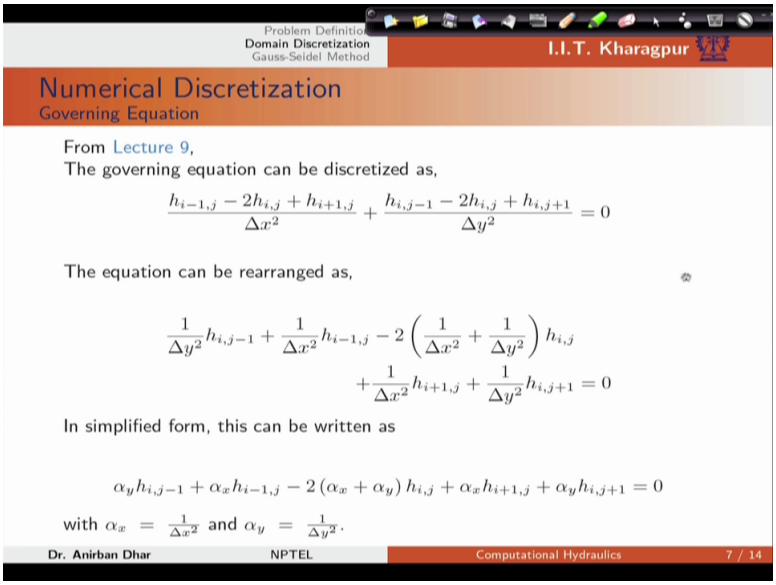
Now from lecture number 9 we know that with our discretization this is hi minus 1j, we are considering only this ij plus 1, this is ij minus 1, i minus 1j, i plus 1j. These points for this discretization.

(Refer Slide Time: 05:30)



So if we further simplify this, this is 2 into 1 by del x square, this 1 by del y square, this things are there. Further simplification if I write it into this format, this alpha x this is nothing but one by x square and 1 by y square. And this is alpha x, alpha y for our problem. So we have this problem where we have two index situation. Now for two index situation we already know that this can be further simplified using single index approach.

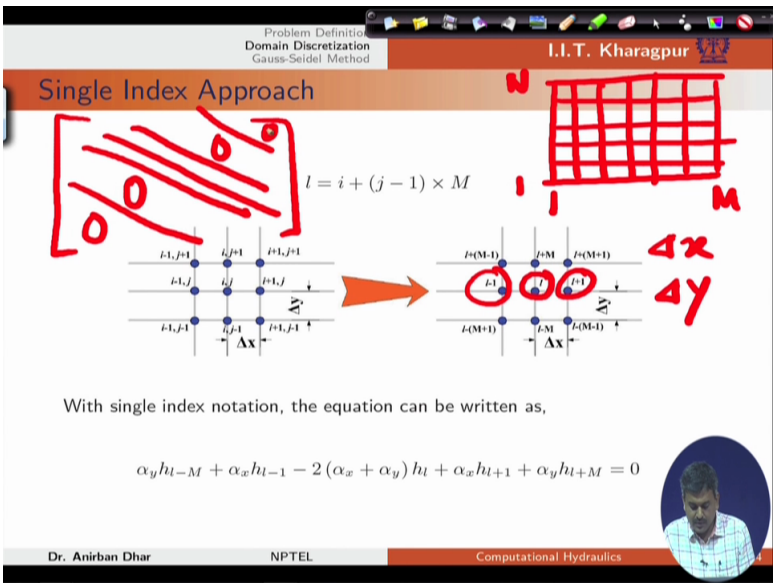
(Refer Slide Time: 06:24)



So if we start from point number 1 to M, again 1 to M and point 1 to N with equal spacing of grids then we can get the information about del x and del y. So with single index notation we can say that for any general node L, the top node is coming as L plus M, bottom one L minus M, this left one is L minus 1 and right one is L plus 1. So we can again generate our coefficient matrix.

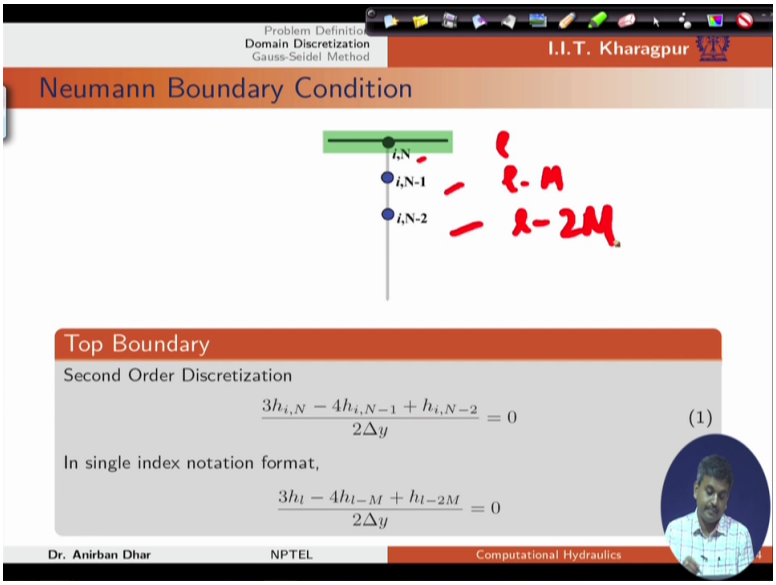
And the coefficient matrix we will be having penta diagonal structure because we have L, L minus 1, L plus 1, L minus M and L plus M, in between zero terms will be there. So with this penta diagonal structure we can solve this problem.

(Refer Slide Time: 07:55)



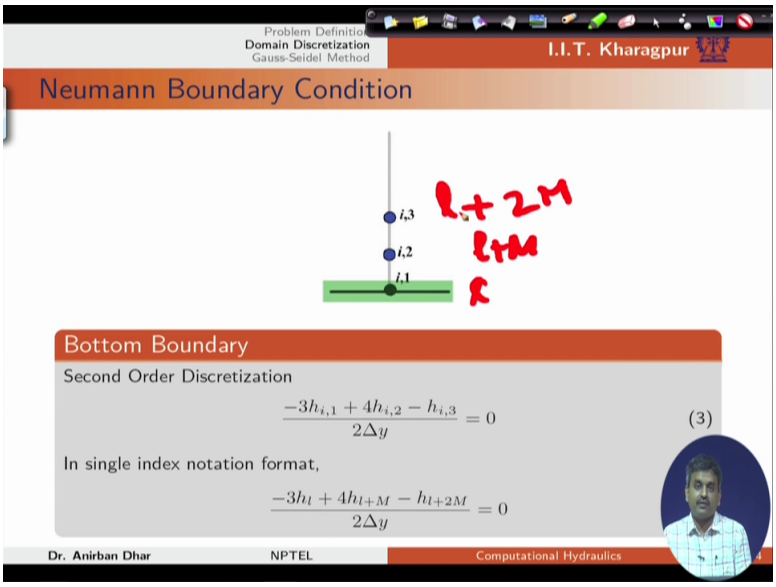
So if we consider the top Neumann boundary condition this considers N, N minus 1, N minus 2 points. And for our case we will have L, L minus M and L minus 2M notation.

(Refer Slide Time: 08:29)



Now if we further see our bottom boundary again that will be in terms of L, L plus M and L plus 2M points.

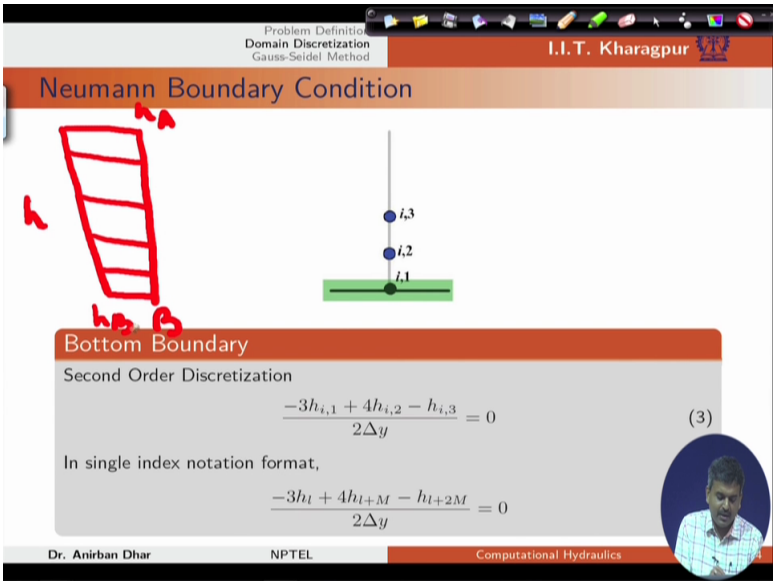
(Refer Slide Time: 08:48)



So we have completely defined our system because on the left hand boundary and the right hand boundary we have specified values. Obviously the specified values, this is hA and this is

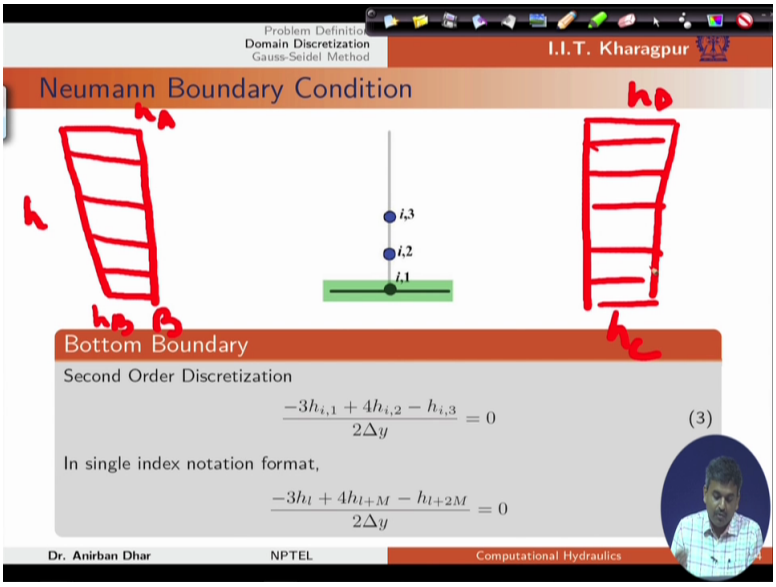
hB and this is varying like this. So we can specify the variation for different nodes. So for any intermediate node we can get the value of h at the boundary. This is hB.

(Refer Slide Time: 09:27)



On the right hand side similarly this is having higher value, this is lower this is hD and this is hC. We can get the intermediate value on the right hand side. So we have completely defined our boundary conditions.

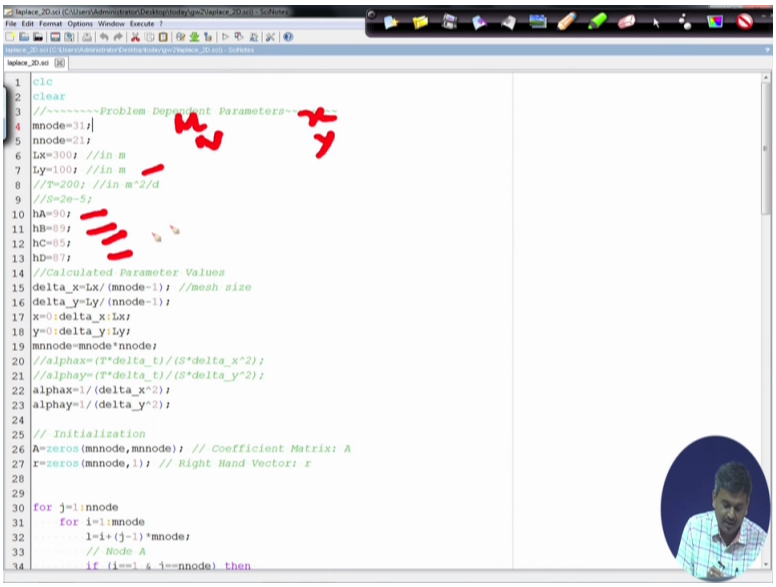
(Refer Slide Time: 09:43)



Now we should see how we can solve our problem using this mathematical structure. So again if I open this scilab. I am again opening this groundwater two. We have Laplace to 2D.

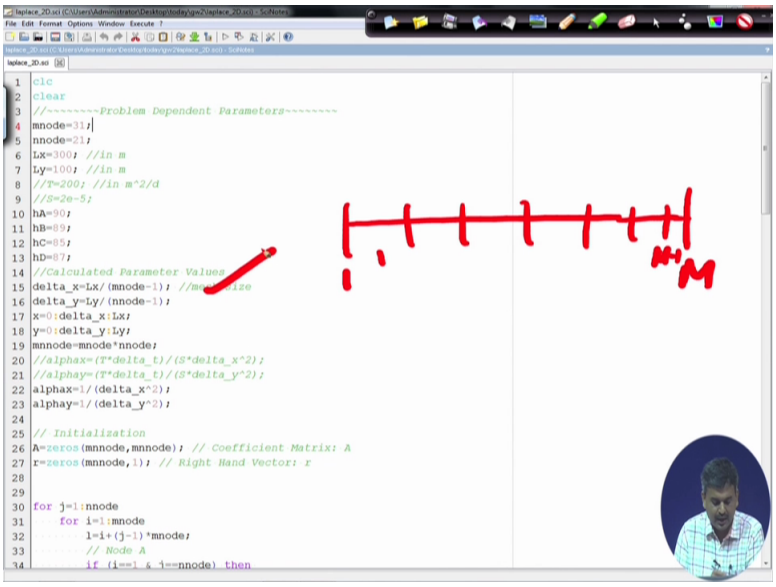
So Laplace 2D, let us say I have 31 nodes N nodes which is M on the other side at N on the y direction. This is x, this is y. Lx is the dimension of the domain, Lx is 300, Ly is 100. This hA is 90, hB is 89, hC is 85, hD is 87.

(Refer Slide Time: 11:05)



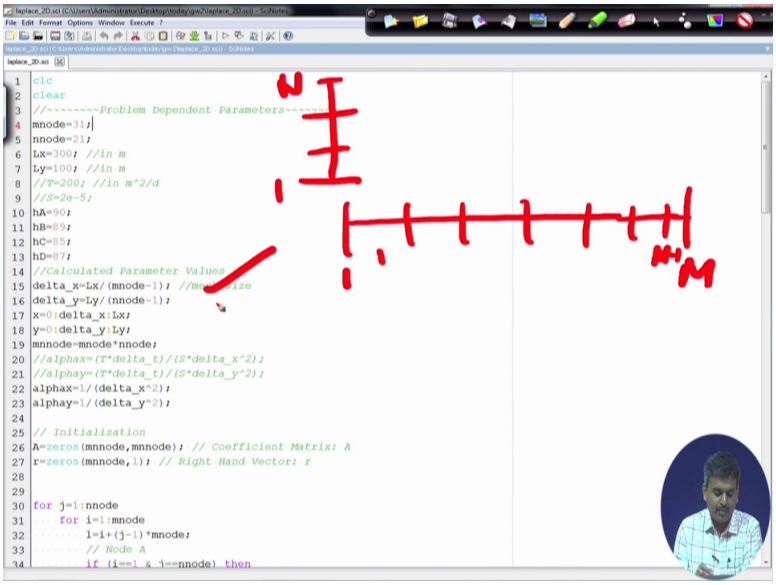
Now we can define this delta x. Delta x is Lx divided by M node minus 1. So the delta x size will be 1. We can get delta x by dividing Lx into 1 segment. Your segment number will be less than your node number. This is starting from 1 to M. So obviously you will have 1 to M minus 1 number of segments in between. So that is why I am dividing it with M minus 1.

(Refer Slide Time: 11:55)



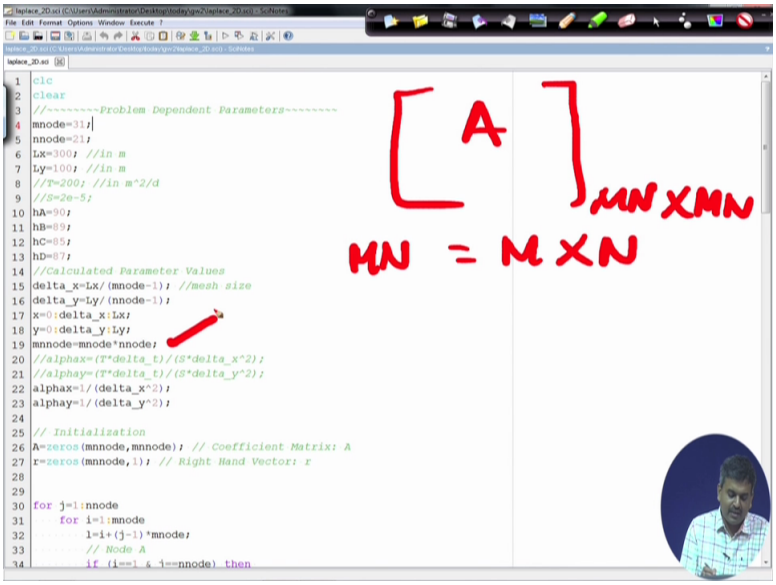
Again delta y on this direction, again I can divide this one into Ly divided by N node minus 1.

(Refer Slide Time: 12:07)



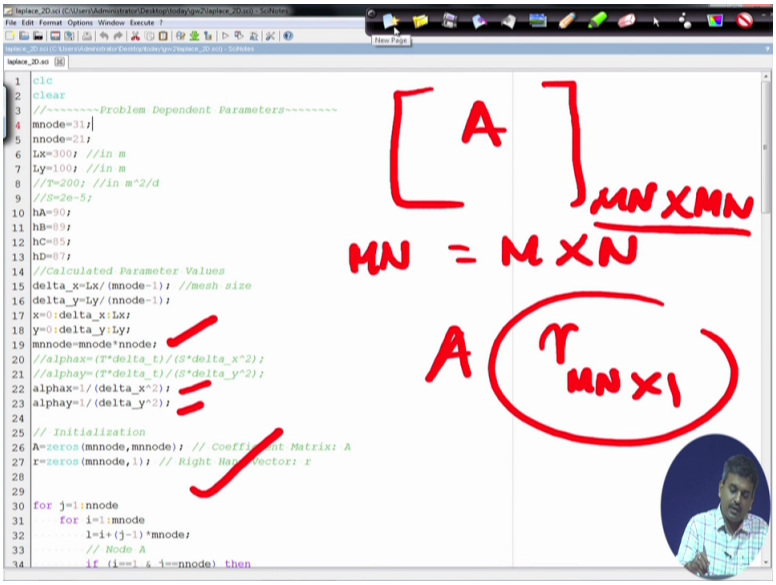
And I am generating the x values including the N points. X 0 delta x, Lx. Y 0 delta y, Ly. Now MN node, I need to know what is the MN value, M cross N because we need to define the full matrix. This is the A matrix and this side should be MN cross MN. So this is MN node means M node into N node.

(Refer Slide Time: 12:46)



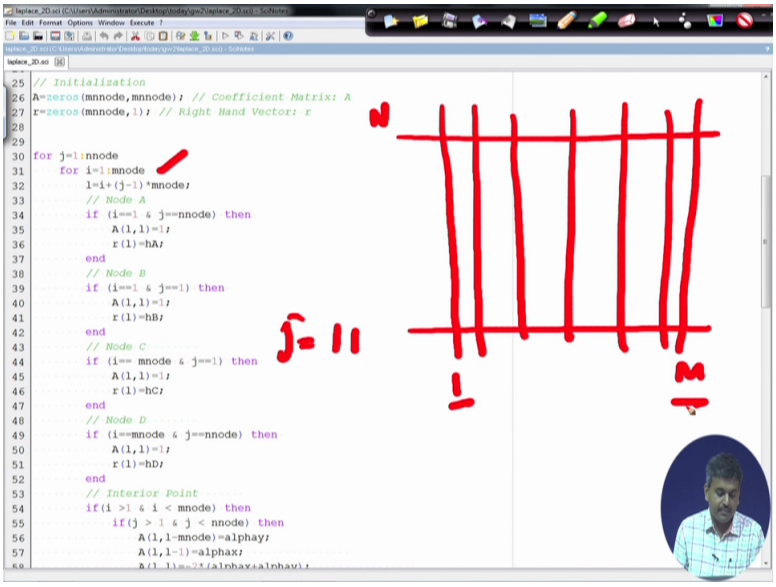
And this alpha x is 1 by del x square, alpha y is 1 by del y square, initialisation of the matrix, this is A zeros MN node into MN node. That means this one I am defining. And r, obviously r will be MN cross 1. So that is what I have initialised. Now I need to put some numbers based on the coefficient values.

(Refer Slide Time: 13:26)



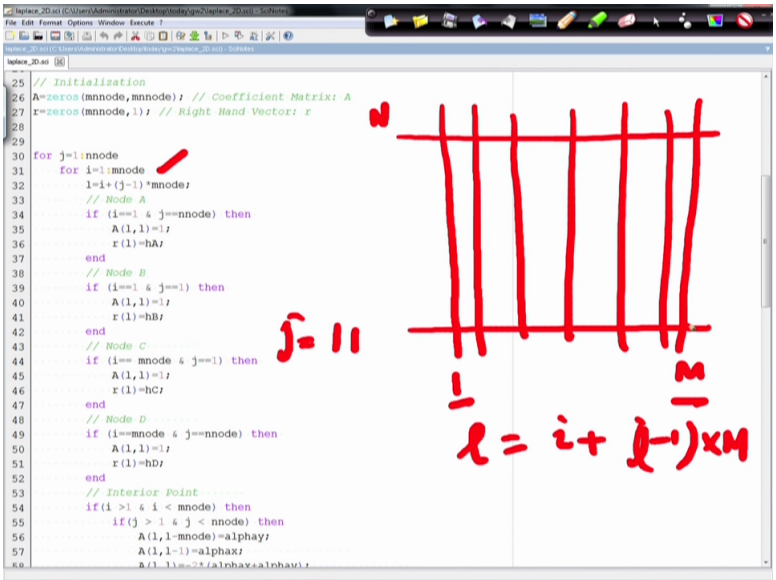
So let us start from our, this is our N level, this is 1 level, this is 1, this is M. So we are starting j equals to N node. So starting from j equals to 1 and running from 1 to M node. So in between, and we are running from 1 to M node here.

(Refer Slide Time: 14:19)



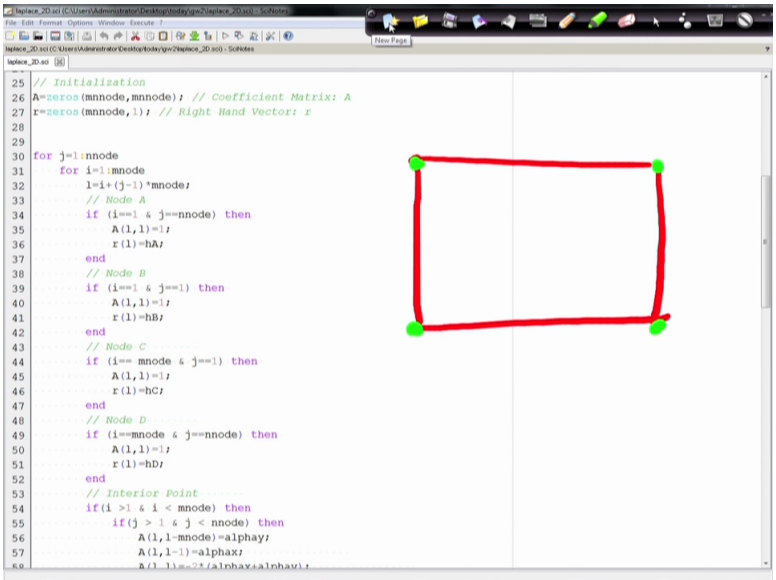
What is the combined index? 1, this is L equals to i plus j minus 1 into M. So if M equals to 1, in the first row we are designating the values as 1 to M. Then we are starting from M plus 1 to your 2M in this case.

(Refer Slide Time: 14:50)



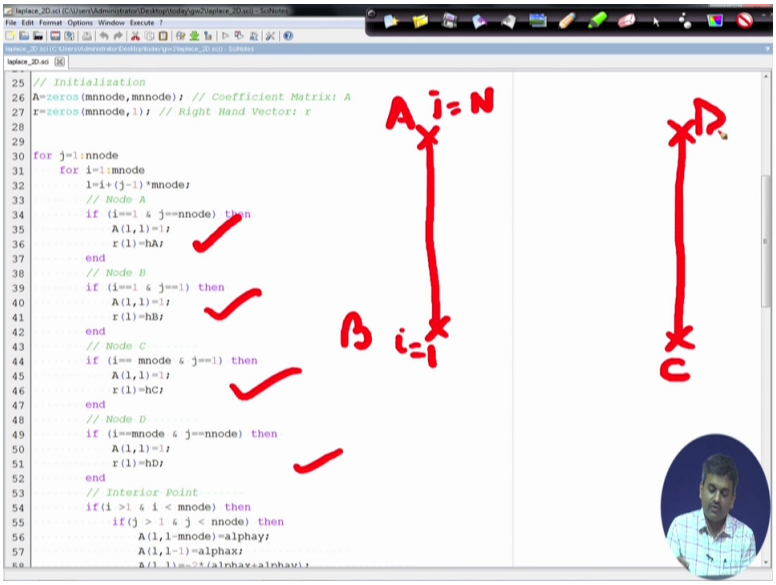
Then let us define first the corner values. Corner values important point is that we have four corner values. During calculation there maybe confusion that whether we should include it in our Neumann boundary condition or specified boundary condition?

(Refer Slide Time: 15:15)



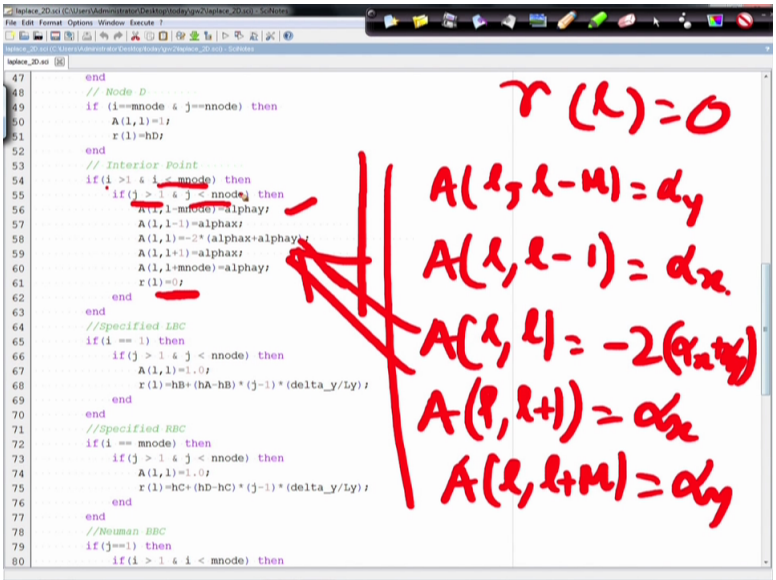
In this problem I have considered that we are considering the end point M specified boundary condition. So from node A, again it is a specified value that is why the coefficient is L equals to 1 and r L, this is hA because we have single index in this case. So if i equals to 1, this is i equals to 1 and on this line if j equals to N. So obviously in this case we have defined A. Similarly we can define B, then C, then D. So we have defined our corner nodes.

(Refer Slide Time: 16:22)



Next level is defining the interior nodes Interior nodes obviously we know that the coefficient of A L, L minus M, this is alpha y. This A L, L minus 1, this is alpha x. A L, L, this is minus 2 into alpha x plus alpha y. And A L, L plus 1 again this is alpha x. And A L, L plus M, this is alpha y. We can directly put it here. And the right hand side obviously that is zero or r L equals to 0 for all interior nodes running from i greater than 1 and i is less than M and j greater than 1 and j less than your N.

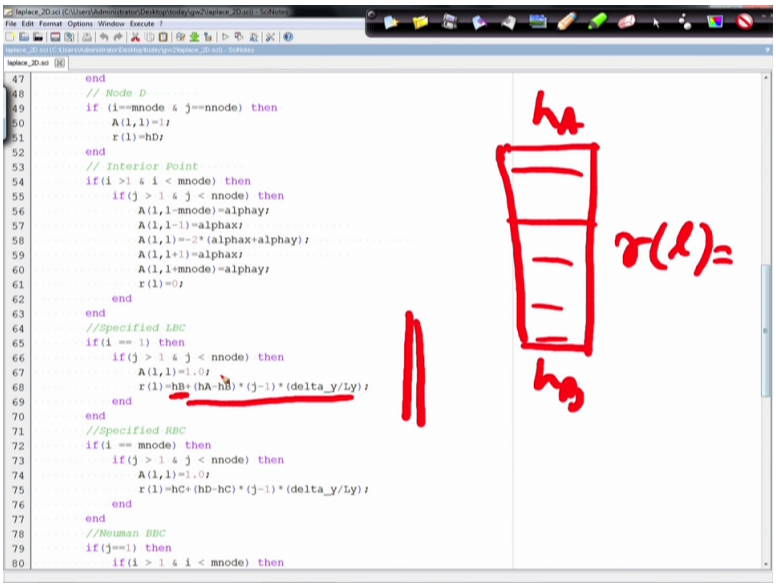
(Refer Slide Time: 17:41)



So we can define all the nodes here. Then comes your specification of left boundary condition. So the base value is hB because there is variation here hB and this is hA. So

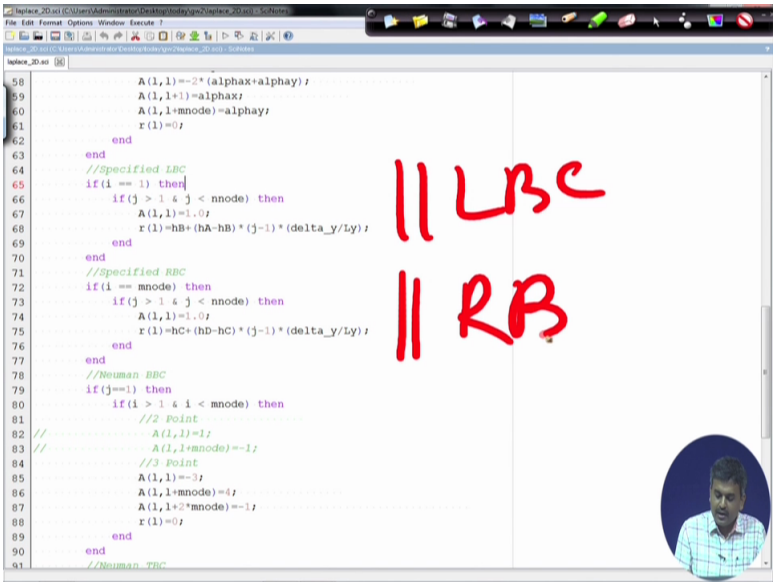
intermediate points we can define the specified value on the right hand side. RL equals to some specified value based on this linear interpolation between hB hA. And the coefficient is again 1 here.

(Refer Slide Time: 18:25)



So this is the case for left boundary. Same for right boundary. This is for left boundary condition, this is for a right boundary condition.

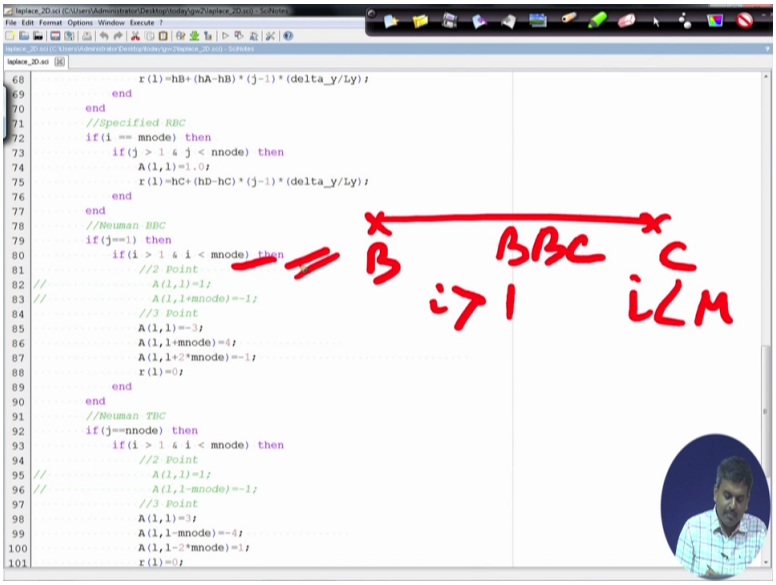
(Refer Slide Time: 18:43)



And for Neumann boundary we need to see the thing here. Neumann boundary condition, that means j equals to 1. But obviously we are excluding, this is our bottom boundary condition.

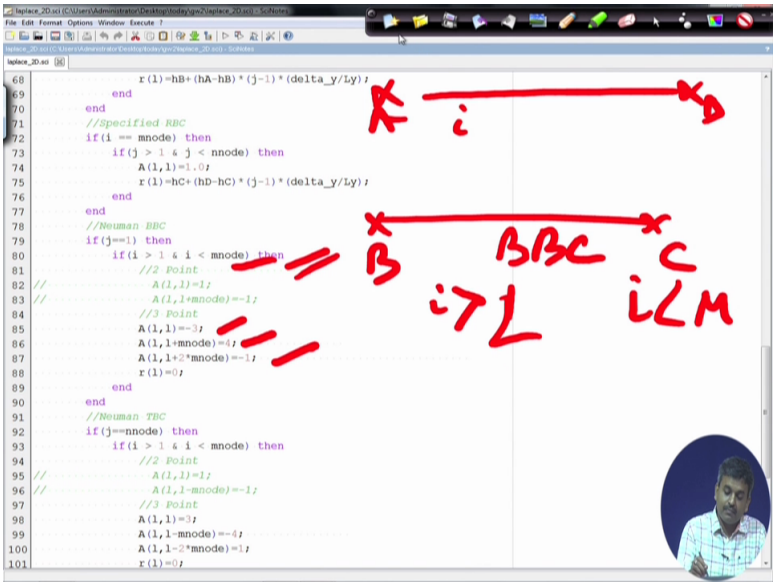
So we are excluding these two points. That means B and C. That is why i is running from 1 and i should be less than M. That is what we have implemented.

(Refer Slide Time: 19:15)



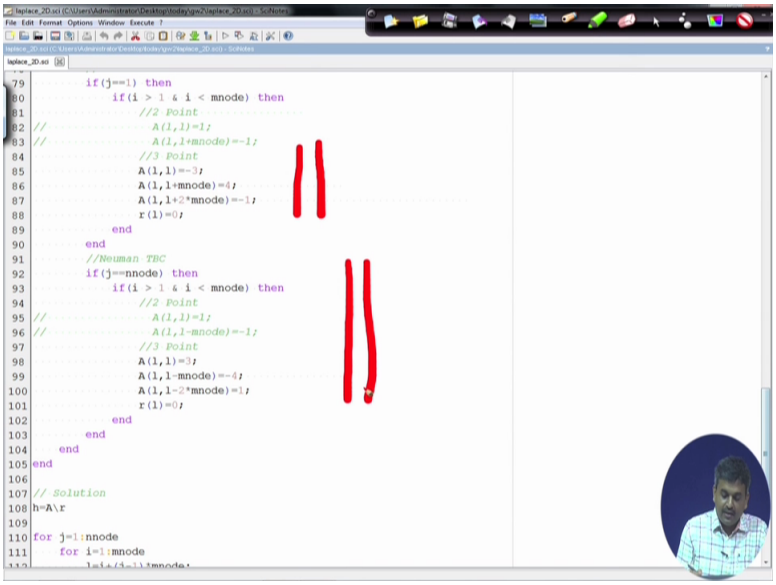
Let us see what is the case for 3 point case situation? 3, 4 minus 1 if you have top boundary condition similarly we can exclude this A point and D point. So i is again running from 2 to M minus 1. It is more than 1 and less than M.

(Refer Slide Time: 19:41)



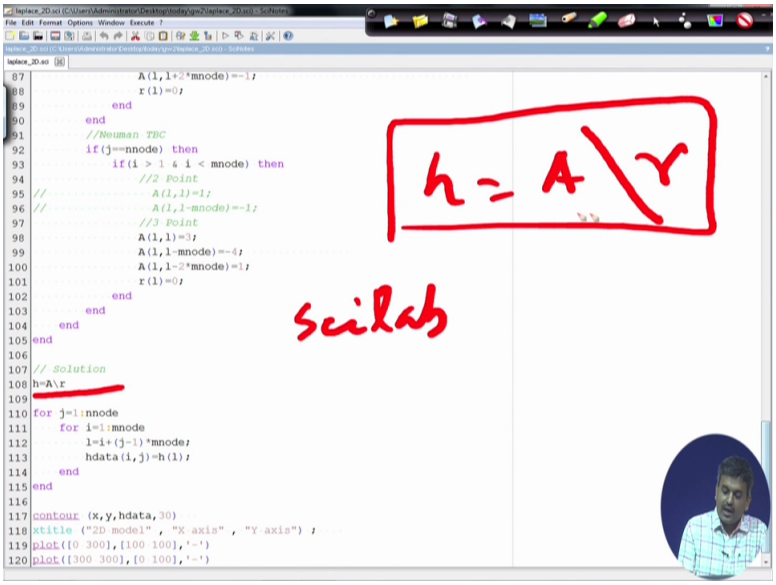
So in this case we can easily implement our boundary conditions whether it is bottom boundary or top boundary condition.

(Refer Slide Time: 19:56)



And finally in this particular solution I am not using the Gauss elimination, I am using some internal function of scilab. That is h equals to A backslash r. This will give you the solution h. This is same as Gauss elimination. We can also use Gauss elimination for this one and get the solution. But in this case to reduce the code size I have used this internal function here.

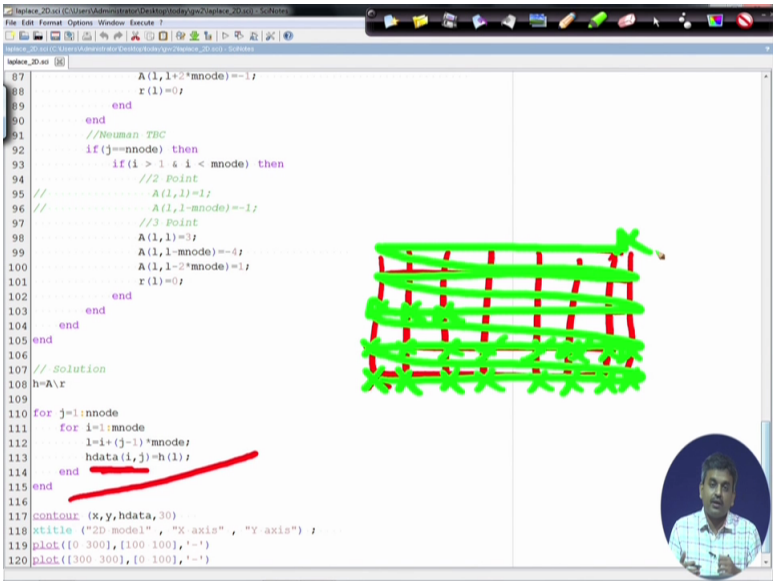
(Refer Slide Time: 20:33)



And if you use this internal function then again you need to transform it back. That means again I need to write it in terms of i j M. So h data is basically nodal data. Initially we have considered the single index notation and we have got the solution at this point starting from

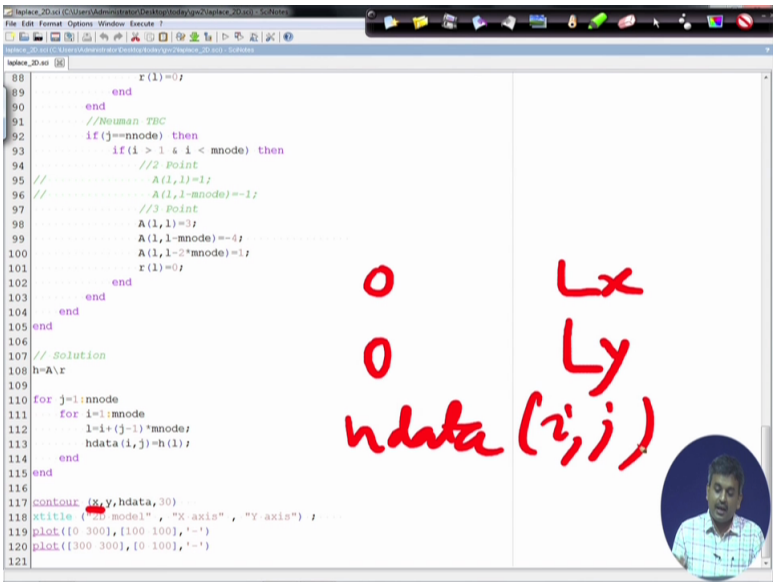
12345678. Maybe after that this one like this. Now again our movement was like this. And we got the solution up to this point.

(Refer Slide Time: 21:33)



So again we need to transfer it into 2D format. So if we transfer it into this 2D format, so using this h data, we can get the h variation with the varying ij. Now final thing we need to have contour. So we already specified this x value which is varying from 0 to Lx and y value which is varying from 0 to Ly, h data which is again function of i and j, essentially function of x and y in this case.

(Refer Slide Time: 22:18)

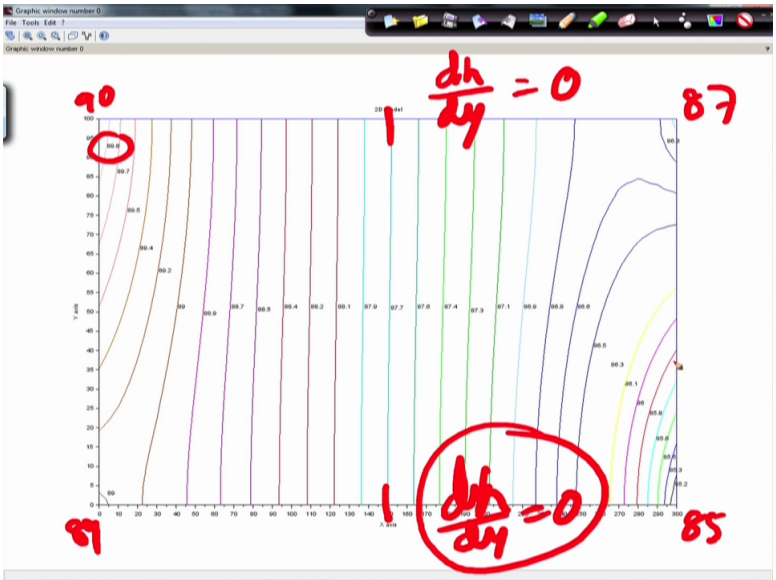


Now if I plot this, so I can select and run this one. Interestingly I am getting the contour plot. Whatever value we have specified in this case we are getting that. So near to this one we are getting 89 point 8. Obviously we have specified 90 here, 89 at this point. So near to this 89, 85, this is 87.

(Refer Slide Time: 22:56)

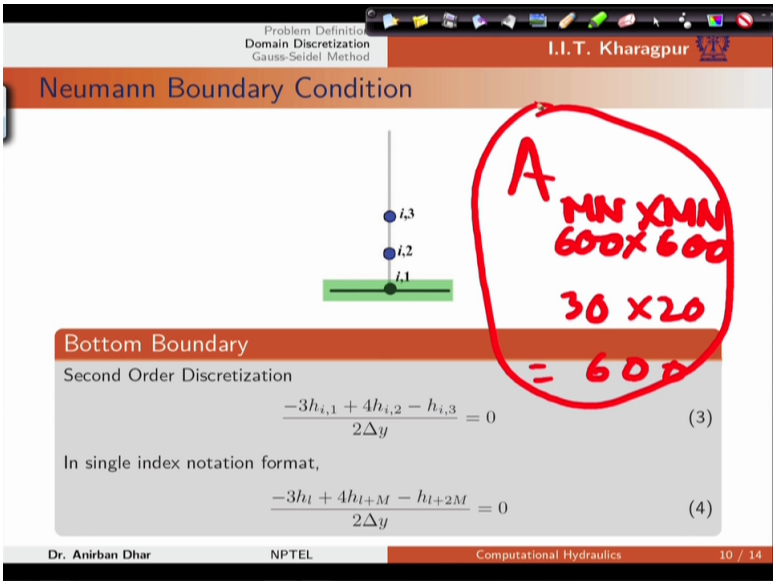
So another important point is this contour lines are perpendicular to this boundary and this boundary. Because we have dh by dy equals to 0 for these two boundaries. That means in that direction there is no variation or no flow condition. So the contour lines are perpendicular. Obviously these are crossing this right and left boundaries. Because we have specified values at that level. So from your contour plot also you can verify whether your solution is giving correct results or not.

(Refer Slide Time: 23:52)



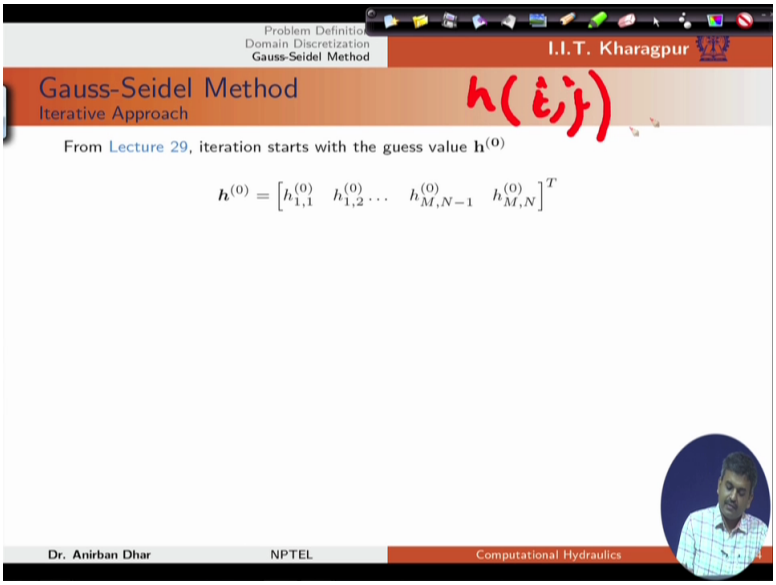
Now we have utilised this full matrix concept or single index notation for the solution of our problem. But the problem is that we need to store the full matrix A which is A MN cross MN. Let us say that we have 30 nodes on x direction and 20 nodes on y direction. Obviously this will be 600. So we need to store this 600 into 600 elements in this A. Now another approach we can follow and we can avoid this construction of A matrix.

(Refer Slide Time: 24:54)



It is basically from the concept of our Gauss Seidel. In Gauss Seidel what I can do, I can store my original values in ij. So in ij if I store, that is the natural way of storing the things. And I can start with the guess value or initial (val) guess value here for the particular case. So this is your ij.

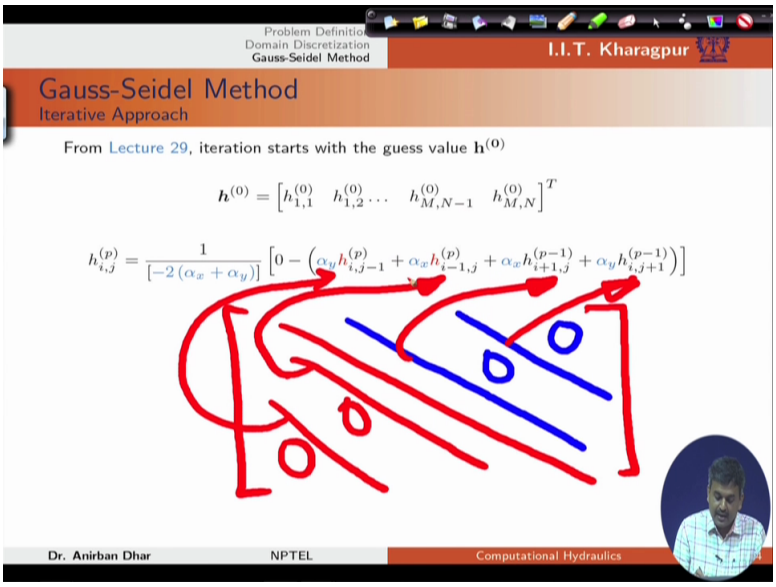
(Refer Slide Time: 25:46)



Now with this guess value I can write this one because anyway we are solving this problem as full matrix in our previous case. So whatever value is there on the upper side and we have the lower side values. So this one is corresponding to this j plus 1, this one is i plus one and this lower one is this one and finally this one is there. So if we construct our original matrix then this will look like this. But without constructing itself we can solve it.

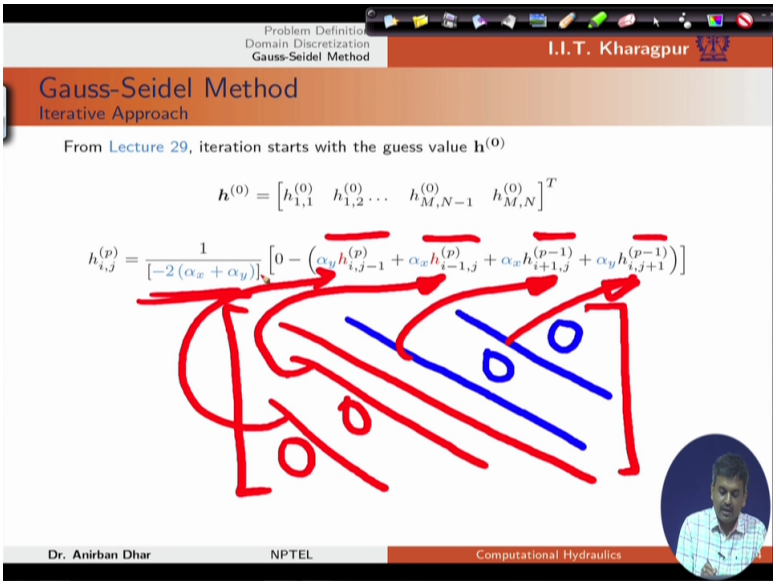
What we can do? We can transfer it on the right hand side. Again we will get updated values for our lower triangular one.

(Refer Slide Time: 27:32)



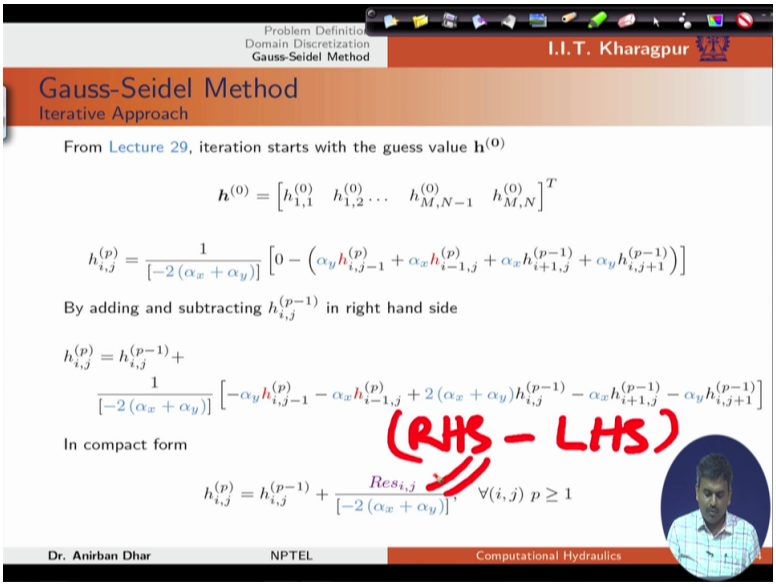
So in this case we have these two updated values and p minus 1, these are old values and this is our central coefficient that we will get.

(Refer Slide Time: 27:42)



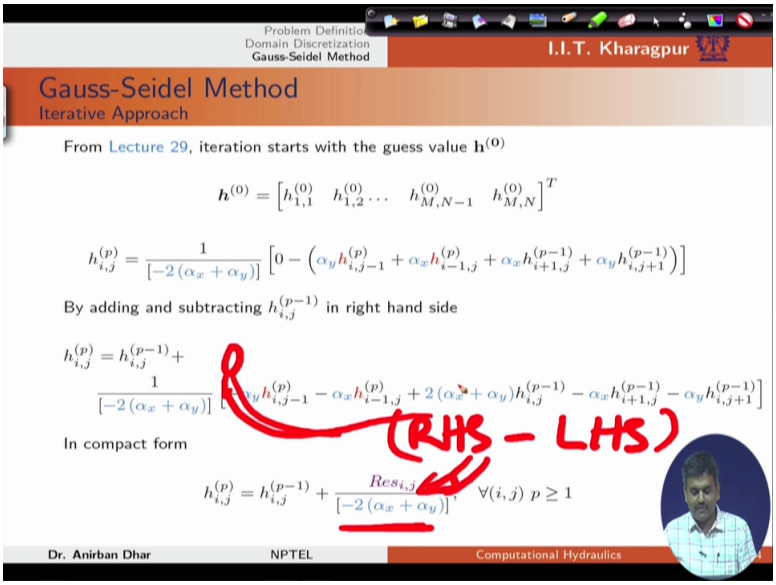
Now from this place if I write this one by adding and subtracting this term, what basically I am gaining? I am gaining in terms of the structure. What is there? This is nothing but right hand side minus left hand side structure. This is our residual.

(Refer Slide Time: 28:21)



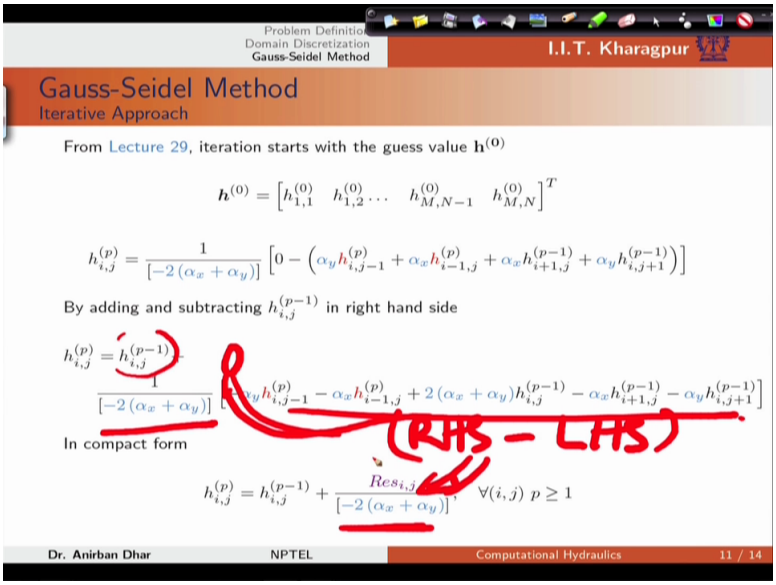
Residual divided by central coefficient. If we see our original problem right hand side there is nothing so we have zero. So right hand side is zero. Zero minus left hand side.

(Refer Slide Time: 28:37)



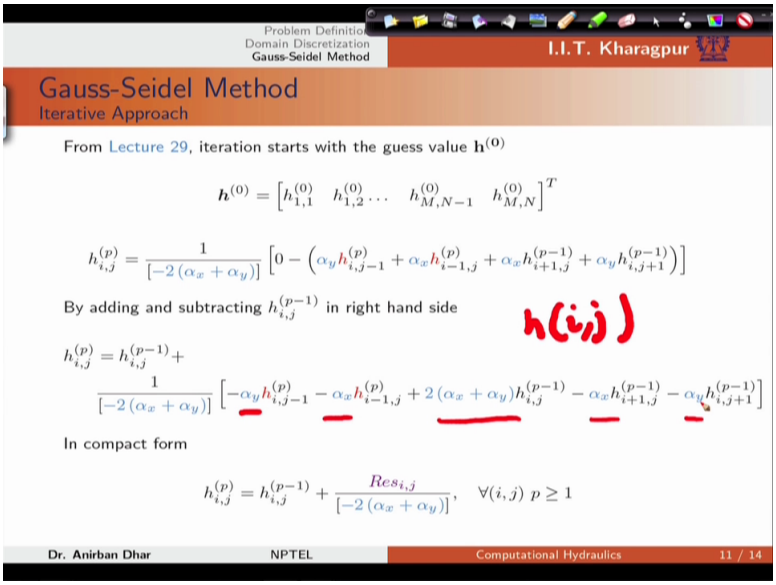
Whatever was there at left hand side we can take a negative sign of that and we can directly write that here and divide it by central coefficient plus our old value. And if we multiply some term here then we can say that this is (29:01) but from Gauss Seidel point of view this is same.

(Refer Slide Time: 29:08)



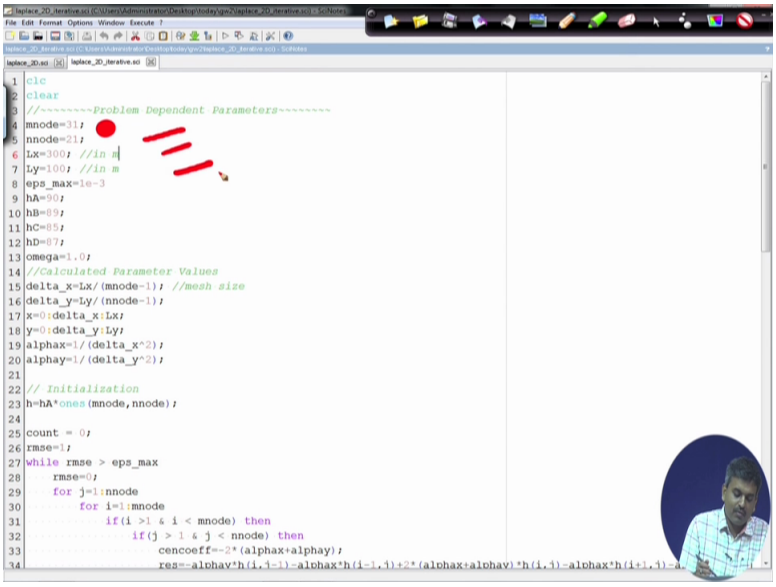
Now in this case it is interesting to note that with single storage. That means we have stored only ij values, h ij. But we have not constructed our A matrix. We have only multiplied this alpha x, alpha y terms here and without constructing our A matrix we can solve this thing.

(Refer Slide Time: 29:40)



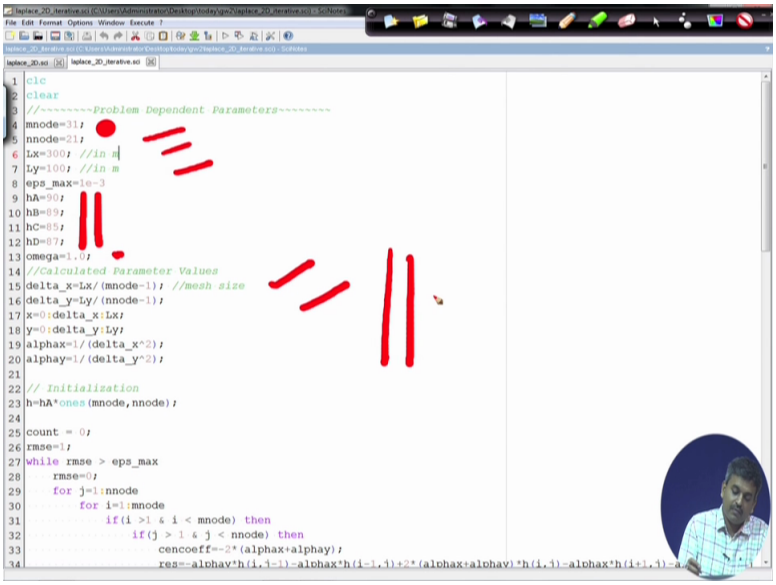
So if I show this thing, so Laplace iterative. We are using this two point method. Now in this case clc clear, this one is common, M node is 31, this is 21, Lx 300, Ly 300, eps max this is epsilon maximum. We can define this criterion for rmse calculation.

(Refer Slide Time: 30:58)



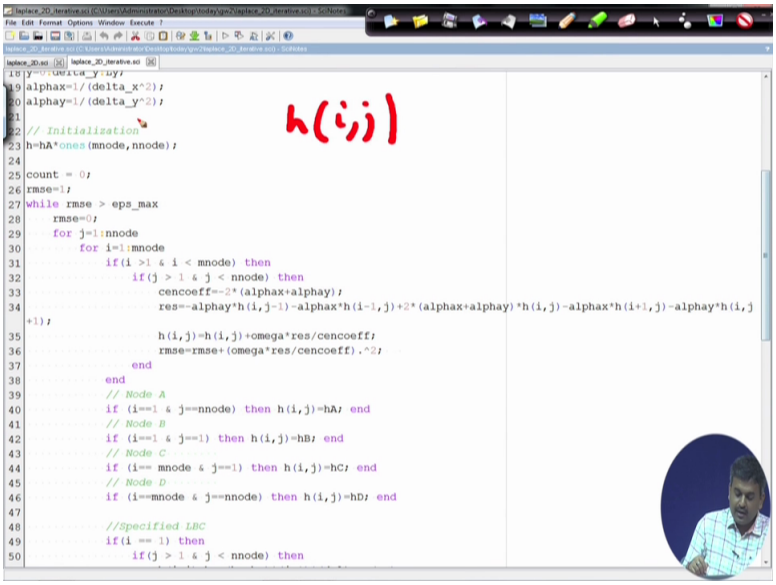
These are specified boundary values and omega because we are considering Gauss Seidel scheme we can consider this omega equals to 1. This delta x, delta y values up to this everything is same to our previous full matrix approach.

(Refer Slide Time: 31:24)



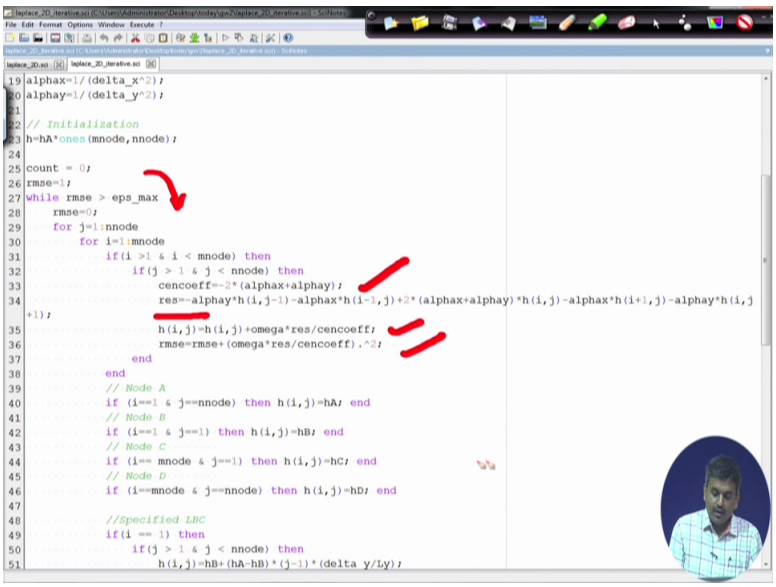
Now in this case initialisation of our original h. I am not defining any old and new vector or matrix in this case. So h ij is the only matrix that I am defining. So in this case initial value I have multiplied hA. That means point A value I am specifying for all the nodes and I can initialise this and I can get the solution here.

(Refer Slide Time: 32:13)



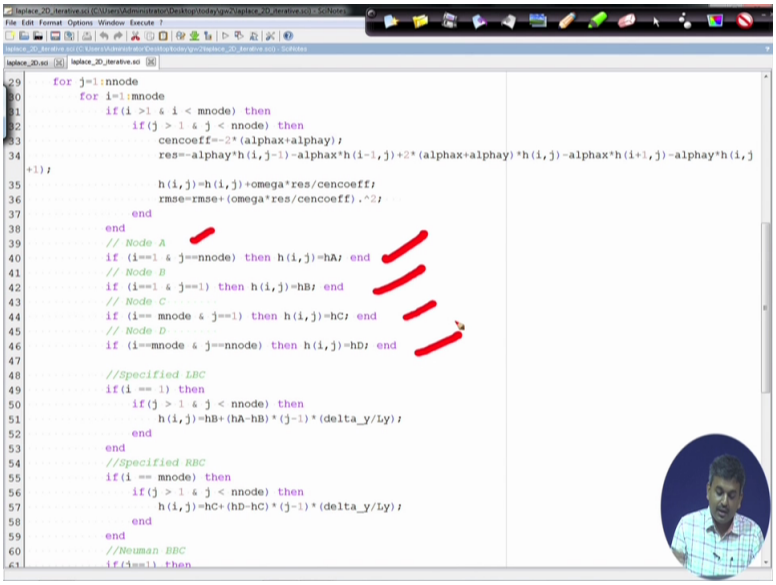
So in this case again the things are within our while loop. Count is counter, rmse is 1. So rmse greater than epsilon max. So rmse is 1 means it is less than 1 into 10 to the power minus 3. So obviously this will enter into this live. Again I am putting this rmse equals to zero and I am defining this central coefficient. Now central coefficient, this is my residual and I am dividing this residual by central coefficient and rmse equals to omega residual by central coefficient square.

(Refer Slide Time: 33:16)



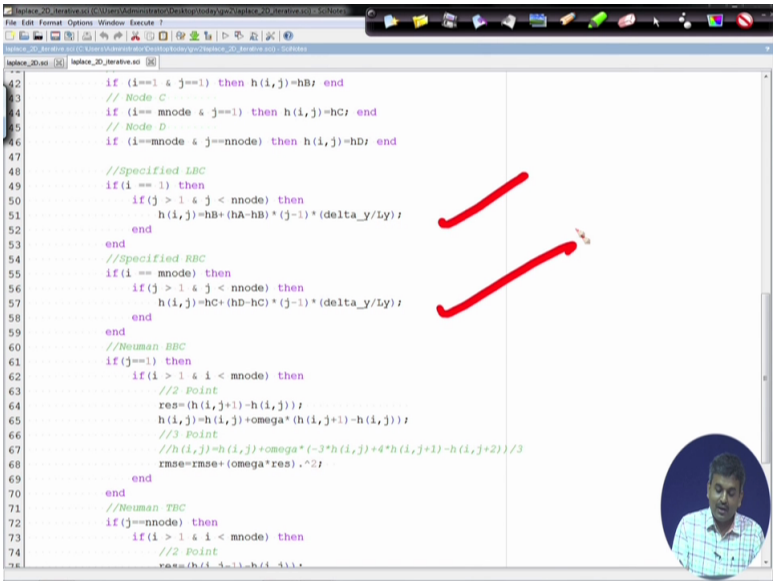
The same thing I am implementing for node A. Node A I can directly specify the value. Node B, node C, node D I can directly specify the values.

(Refer Slide Time: 33:35)



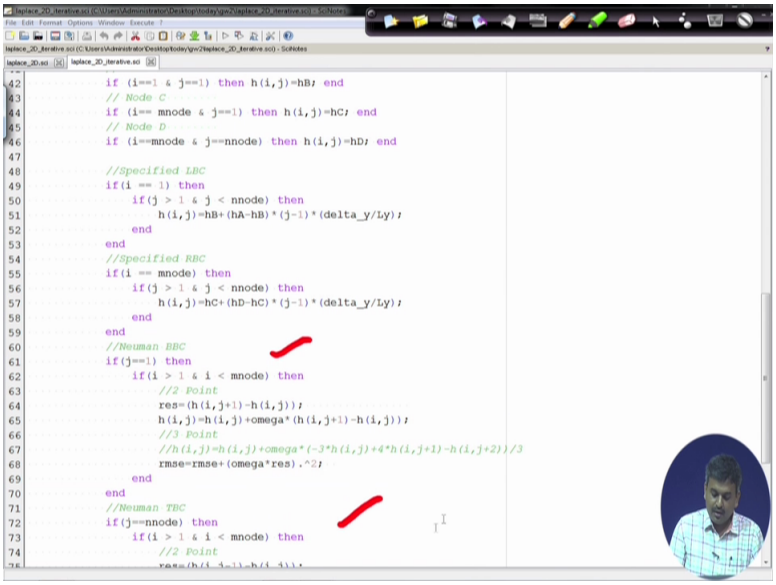
Then specified left boundary condition again h ij, I can directly specify the values.

(Refer Slide Time: 33:48)



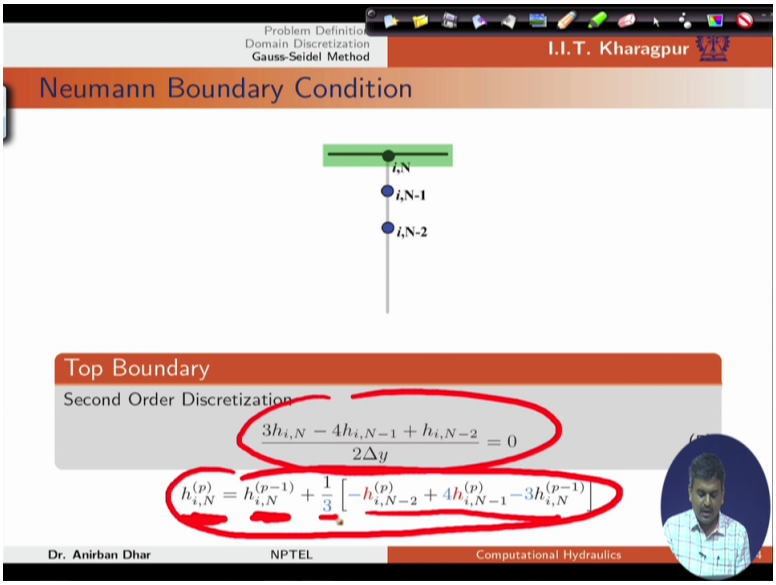
But what will be the condition for boundary case? Either it is a bottom boundary or top boundary. We need to see what will be the changes in our algorithm structure.

(Refer Slide Time: 34:04)



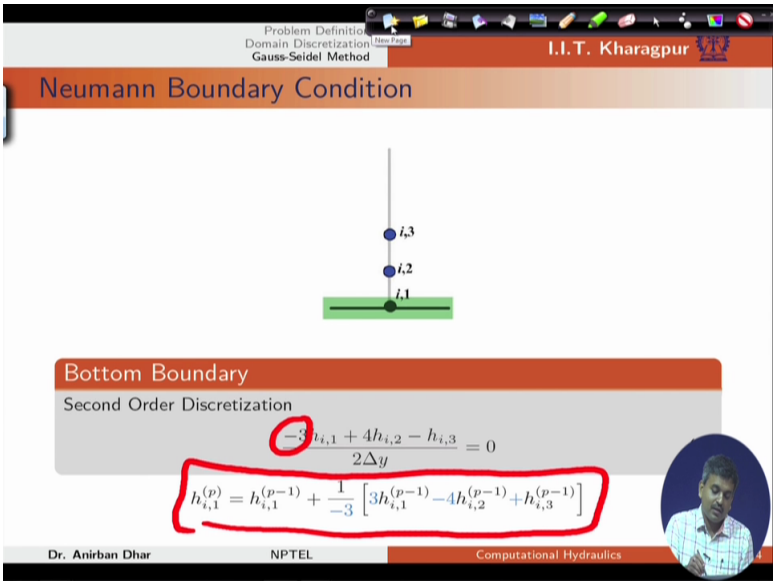
So for Neumann boundary which is top boundary, this was the condition in terms of two index. Still we are using the two index approach. But we can use this kind of simplification because 1 by 3 we are dividing it. This is again, this is updated one, this is old one, this is my residual divided by central coefficient.

(Refer Slide Time: 34:51)



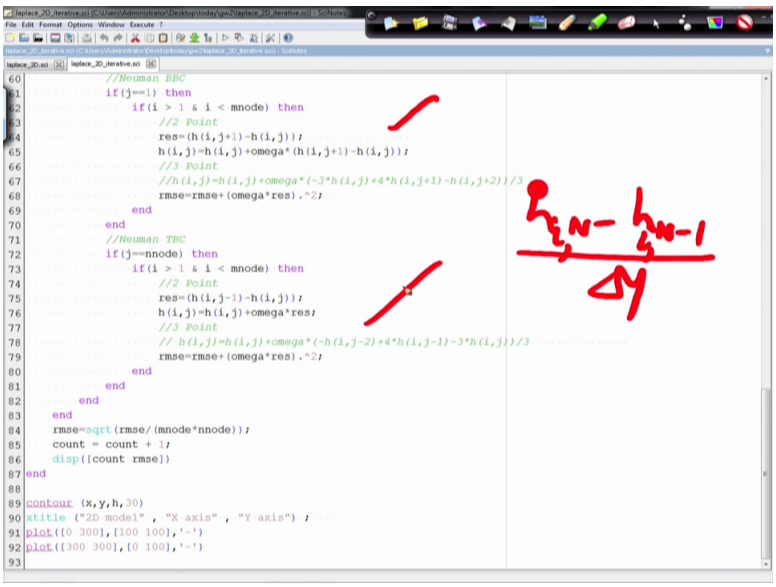
If I further use the same approach, this is my central coefficient. So 1 by 3, this is there if we consider our whole thing. So we can see that this iterative approach also we can solve the problem.

(Refer Slide Time: 35:23)



But in our algorithm I am just showing the values with two point system. So two point system means we are considering that we have our hN minus hN minus 1 for upper boundary. This is del y and for any i. So i N minus 1, we can directly specify this. So for top and bottom boundary we can implement this and ultimately we are updating this thing.

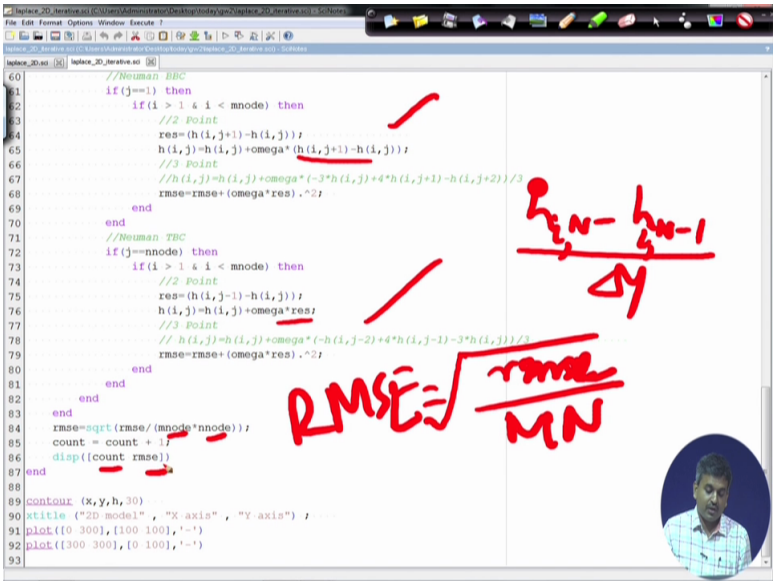
(Refer Slide Time: 36:20)



This is nothing but residual in this case. And finally after adding all rmse we can divide this rmse by M node into N mode. So rmse is nothing but the squared values. So rmse in this case is the summed value divided by M into N. This is the actual rmse value. And count is count

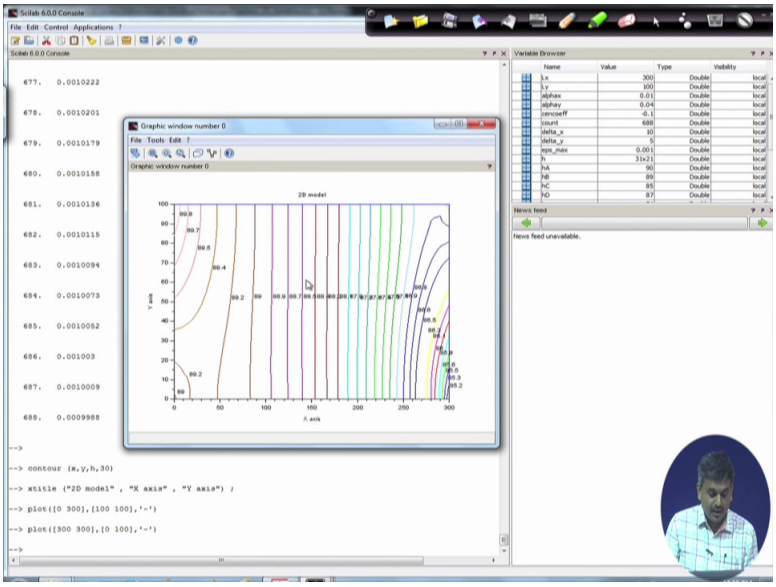
plus 1 and display count rmse means I am displaying the counter number and rmse for this particular solution approach.

(Refer Slide Time: 37:06)



Now in this case we have already got x information, y information. Now h information is directly available. Don't need to transfer it from single index to multi index case. Now I can run it and I should get some result for this one. So I have done it and this is a solution.

(Refer Slide Time: 37:38)



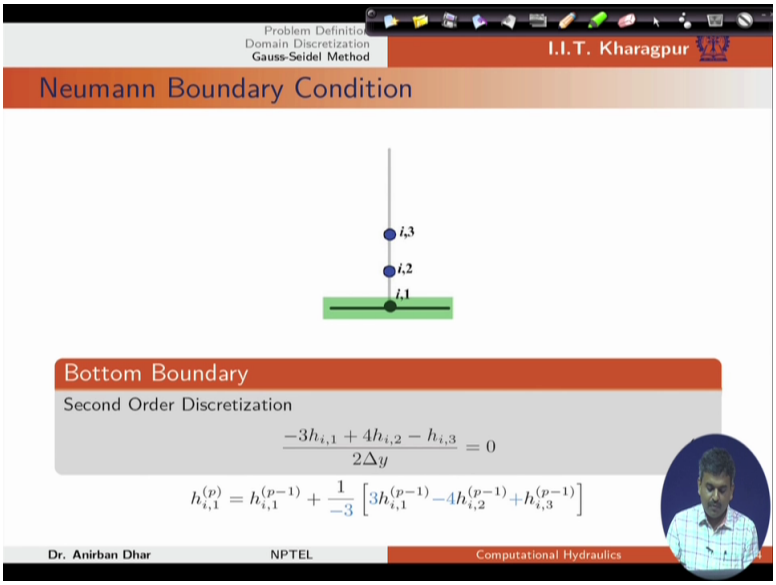
Again we can see that our solutions are matching here. It is 90. So 89 point 8, here 89, there 85, this side 87. So almost we are getting some variation and still for these two boundaries

are impermeable boundaries. We are getting our contour lines perpendicular to the boundaries. So obviously this criterion that is del h by del equals to zero. That is satisfied for these two boundaries. So we can say that our solutions we will get reasonable result from this iterative approach.

(Refer Slide Time: 38:47)

So in this case we have covered both our full matrix and iterative approach. Still I have included this three point case. During solution process I have shown only the solution with two point case. Please use the code that I will upload in the imperial website. You can use this three points system and verify whether this discretization is still valid for your problem or not.

(Refer Slide Time: 39:44)


Problem Definition
Domain Discretization
Gauss-Seidel Method
I.I.T. Kharagpur
Neumann Boundary Condition
i,3
i,2
i,1
Bottom Boundary
Second Order Discretization
$\frac{-3h_{i,1} + 4h_{i,2} - h_{i,3}}{2\Delta y} = 0$
$h_{i,1}^{(p)} = h_{i,1}^{(p-1)} + \frac{1}{-3}\left[3h_{i,1}^{(p-1)} - 4h_{i,2}^{(p-1)} + h_{i,3}^{(p-1)}\right]$
Dr. Anirban Dhar
NPTEL
Computational Hydraulics

Thank you.