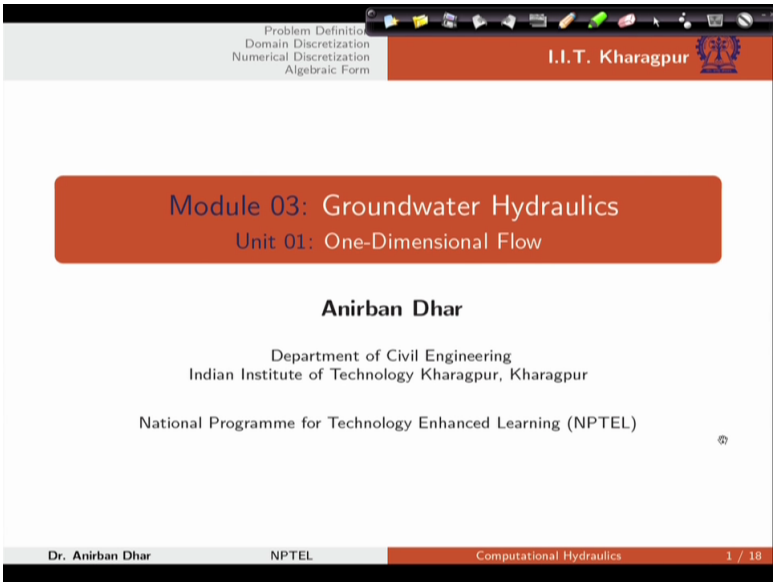
Computational Hydraulics
Professor Anirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Lecture 31
One-Dimensional Flow

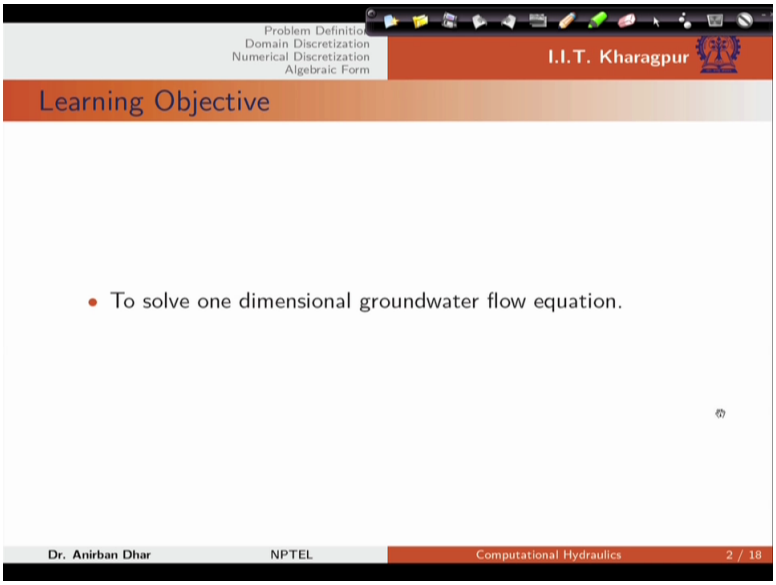
Welcome to lecture number 31 of course computational hydraulics. We are in module 3, groundwater hydraulics. And in this particular lecture I will be covering one dimensional groundwater flow.

(Refer Slide Time: 00:36)



Learning objectives for this particular lecture class. At the end of this lecture students will be able to solve one dimensional groundwater flow equation. Now we are considering the groundwater flow as part of our hydraulics thing. We can have surface water hydraulics or groundwater hydraulics. In this particular lecture class we will be discussing about the groundwater hydraulic part.

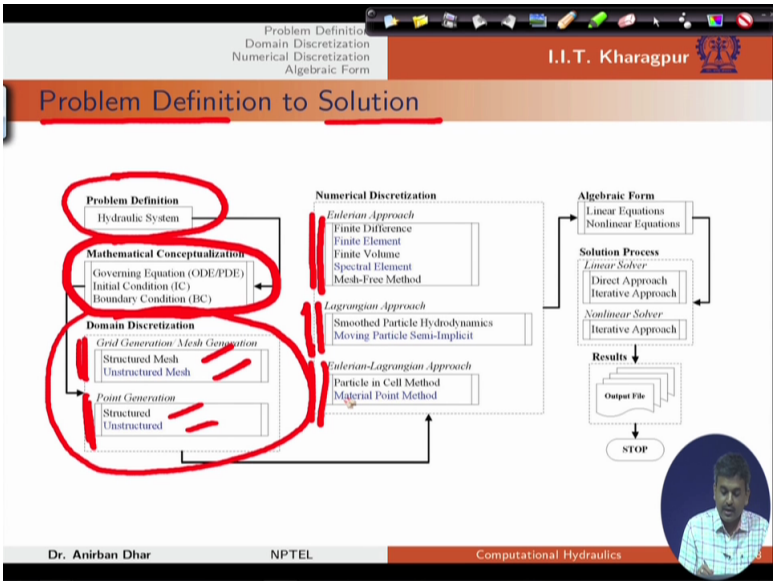
(Refer Slide Time: 01:17)



This flow chart you already know that from problem definition to solution. So problem definition is basically defining the actual complex situation or problem in the field. Then we conceptualize the problems in terms of differential equation, initial condition, boundary condition or depending on the mathematical conceptualization we can omit the temporal term. Then we will require only boundary condition because the time component not be there.

Then comes this domain discretization. So either we can generate grid or mesh or we can generate points. So this grid generation can be structured, unstructured and the point generation can be structured or unstructured. Depending on the type of grid we can select the numerical discretization. Now numerical discretization can be based on Eulerian, Lagrangian or mixed Eulerian-Lagrangian approach.

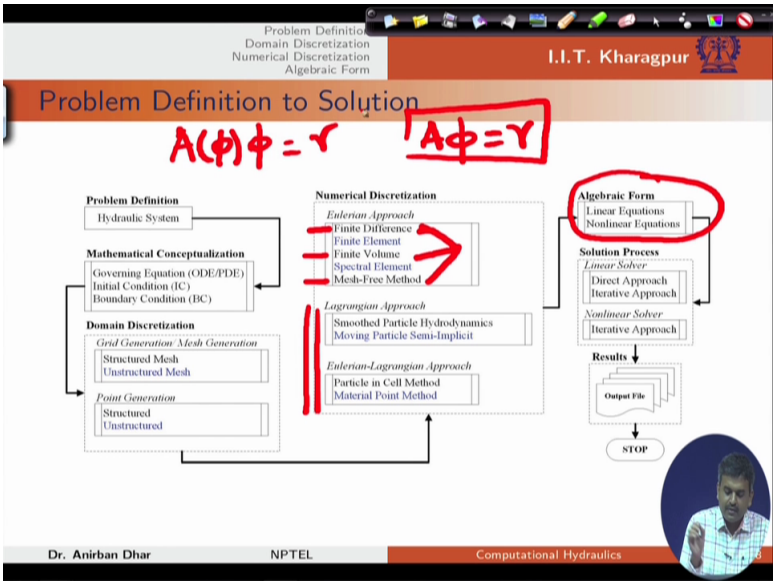
(Refer Slide Time: 02:50)



Now in this particular course we have already covered finite difference, finite volume and mesh free method. However we have not discussed these two approaches. So finite difference, finite volume or mesh free method, we discretize the differential equation under consideration and our ultimate objective is to get algebraic form that may be linear equation or non linear equation.

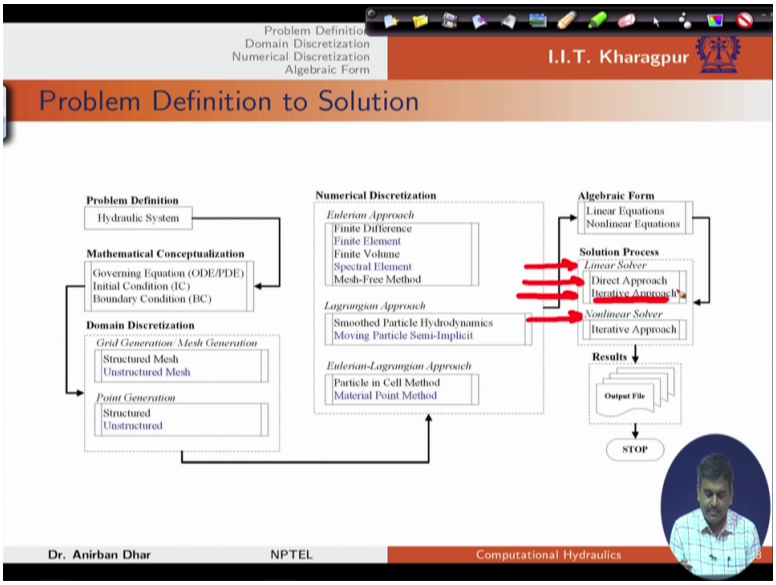
If A phi equals to r and A is constant coefficient matrix then we can say that this particular problem is linear in nature. Otherwise if A is function of phi so we can term it as our non linear equation.

(Refer Slide Time: 04:14)



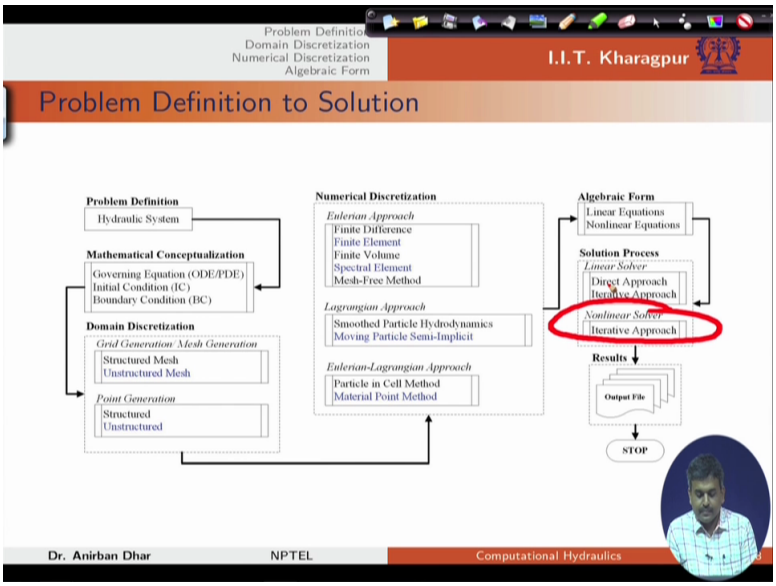
So with this we have already discussed about solution procedure. Solution procedure we can have linear solver or nonlinear solver. In linear solver we have discussed this direct approach, iterative approach. In direct approach we have covered Gauss elimination, tri diagonal matrix algorithm and LU decomposition. In iterative approach I have covered Jacobi method and Gauss Seidel approach and Gauss Seidel with (05:04).

(Refer Slide Time: 05:05)



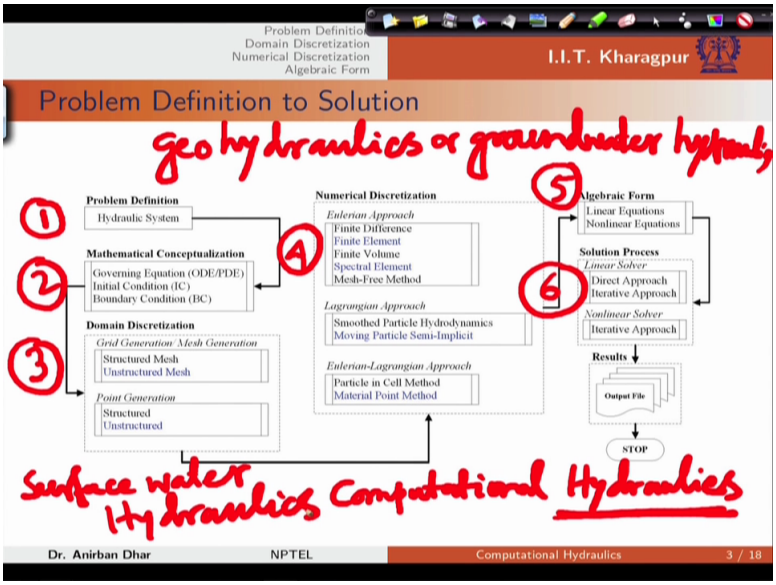
Now in nonlinear solver I have covered this iterative approach using Newton Raphson technique for multivariable case.

(Refer Slide Time: 05:20)



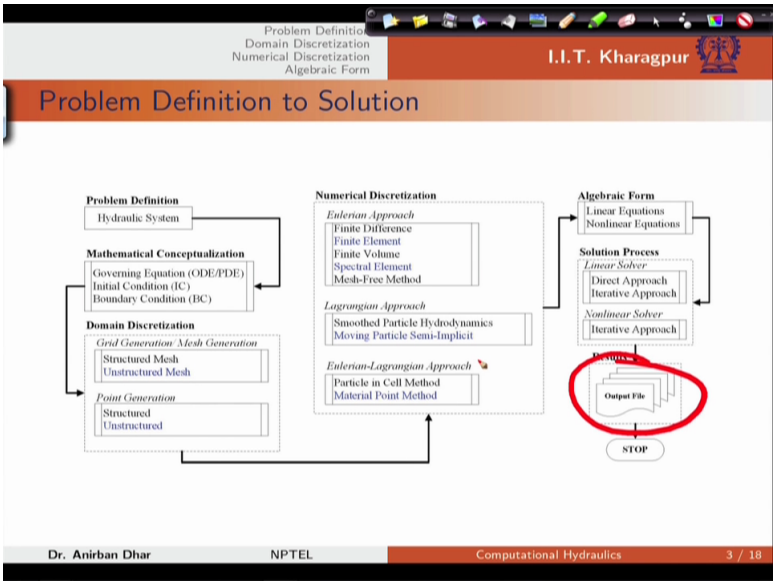
So as we have covered the problem definition, mathematical conceptualization, domain discretization, numerical discretization, algebraic form and solution process. So we have already completed these six steps. Now our objective is to apply this framework for specific problems. So this course is computational hydraulics. So in this hydraulics thing we will be covering geo hydraulics or groundwater hydraulics (lics) and surface water hydraulics.

(Refer Slide Time: 06:53)



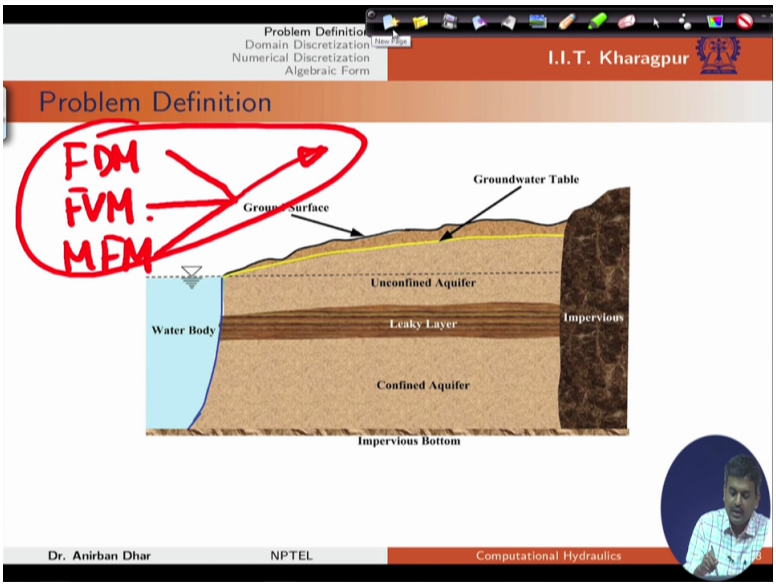
So in this particular lecture I will be covering groundwater hydraulic part and finally we can represent the results using some plots that maybe line plots or conclude plot and finally we can get the desired output.

(Refer Slide Time: 07:11)



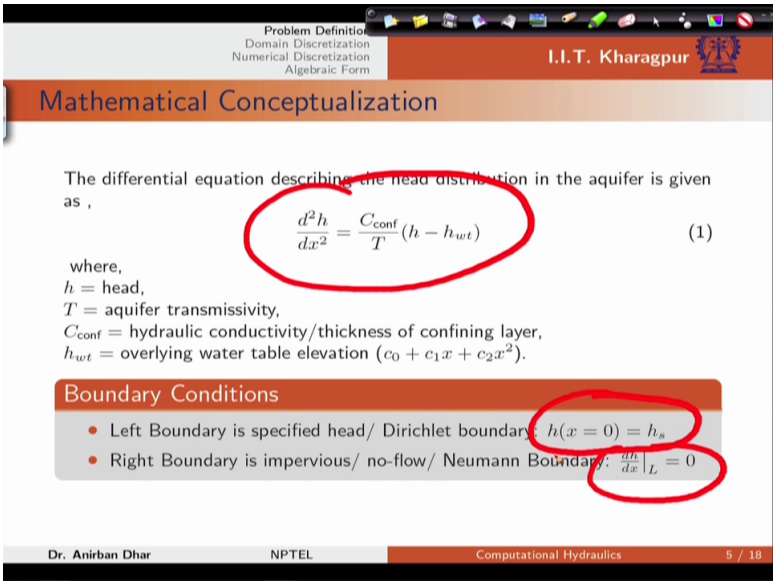
So this is our well known problem. We have discretized it using finite difference method, finite volume method and mesh free method. And what is the ultimate thing we got from our discretization equation? So ultimate discretization is same from all the methods. So we have got algebraic form which is almost same or equivalent in nature. Maybe in near boundary there will be some amount of deviation. Otherwise the discretization is same from all this techniques.

(Refer Slide Time: 08:20)



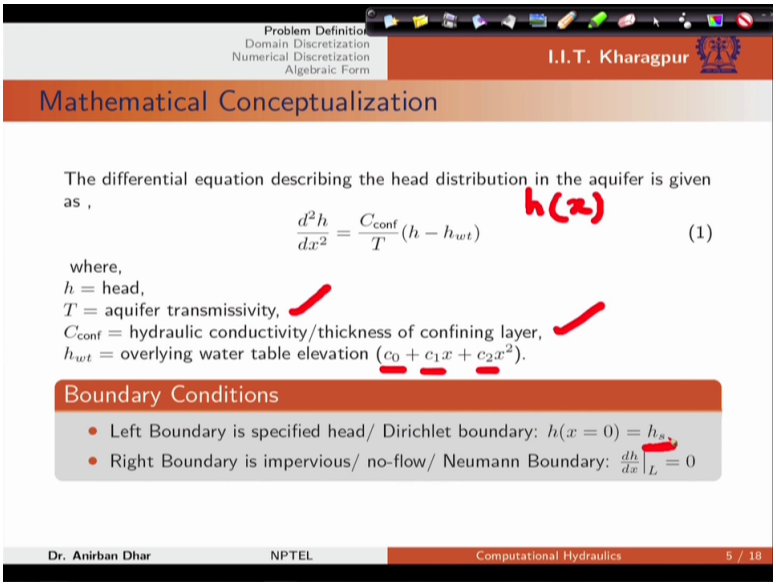
The basic problem is left side we have water body and we have water table in this case. And this is ground surface. So if we conceptualize this problem in terms of differential equation we can write this governing equation, left hand side we have this specified boundary condition and on the right hand side we have zero Neumann boundary condition which is representing the impermeable boundary condition.

(Refer Slide Time: 09:01)



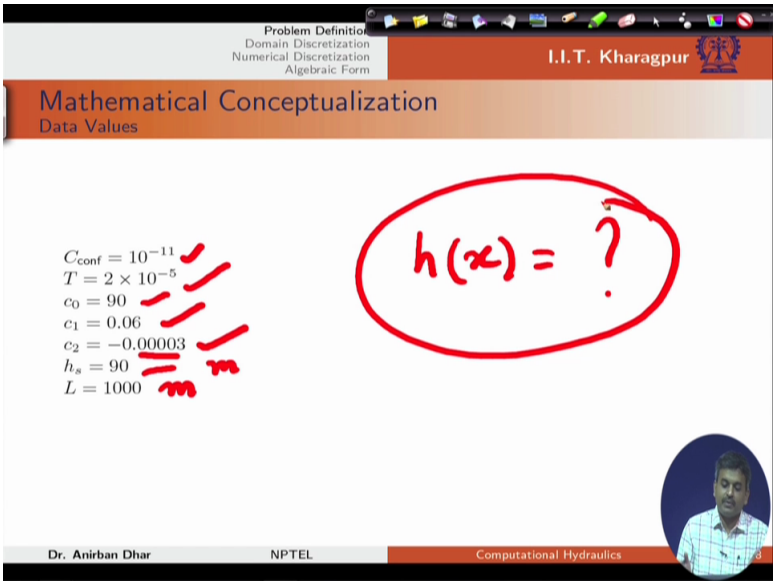
If you define the parameter values, ultimate objective is to get the variation of h as a function of x. So we need T Cconf, h wt. h wt is the function of x. Again C0, C1, C2, these are constants and we need information about this hs which is the condition on the left hand side.

(Refer Slide Time: 09:38)



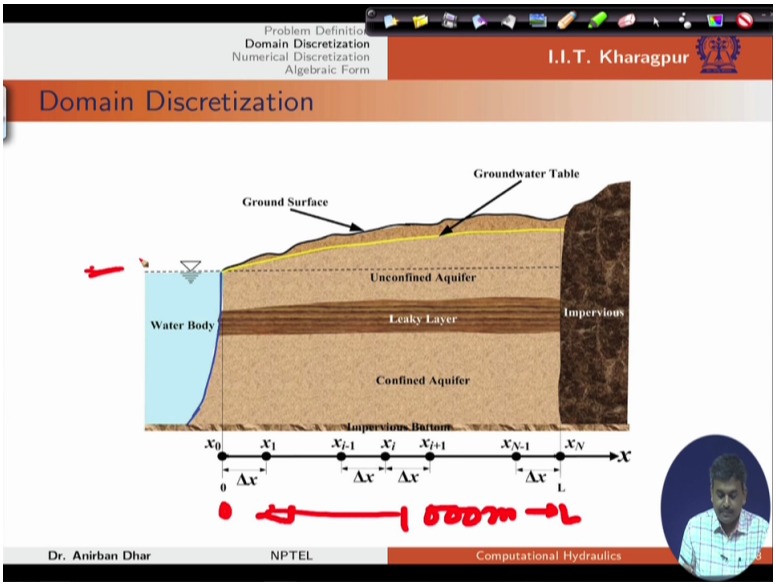
So let us say that Cconf is 10 to the power minus 11. T is 2 into 10 to the power minus 5 and C not is 90, C1 is point 6, this is minus 4 not 3 and hs this is metres. The elevation for the water body that is 90 metres and length is let us say this is 1000 metres. So with this we can solve the problem. Now all the parameters values are known. Ultimate objective is to get this solution. We have discretization available from our previous lecture classes.

(Refer Slide Time: 10:44)



So let us see. So this is our domain discretization from basic finite difference. We have nodes from zero to L. This is our 1000 metres distance and the water level in this case this is 100 metres or this is 90 metres.

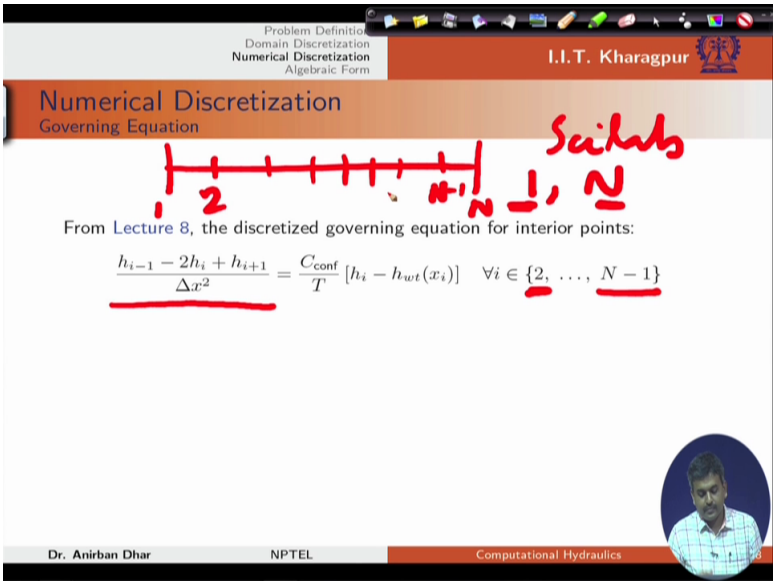
(Refer Slide Time: 11:26)



Now we can solve this problem with our discretization. From lecture number 8 we have this discretization for interior node. If we utilise interior node then from 2 to N minus 1, because in scilab we can store the array values running from 1 to N, not from zero. So that is why I have excluded this one. It is running from 2 to N minus 1. N is the right hand boundary and

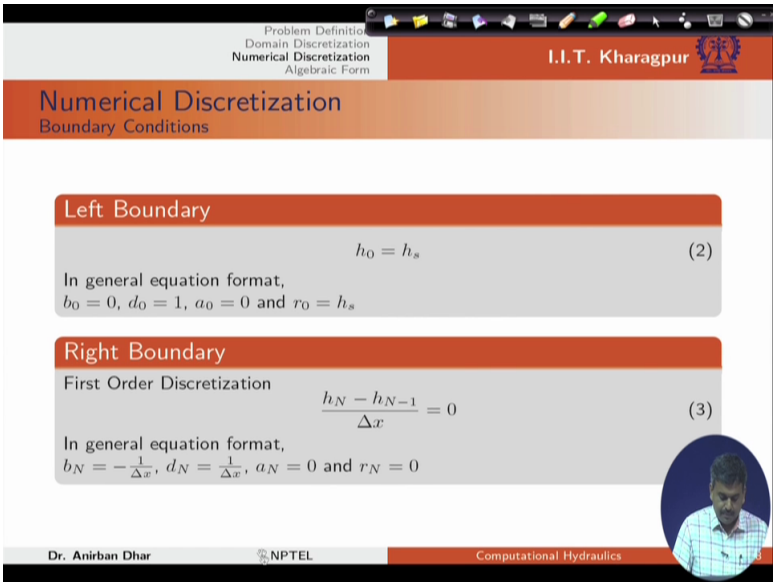
point 1 is the left hand boundary. So this is 1, N starting from 2 to N minus 1. There will be number of points in between and we can solve that using finite difference thing.

(Refer Slide Time: 12:27)



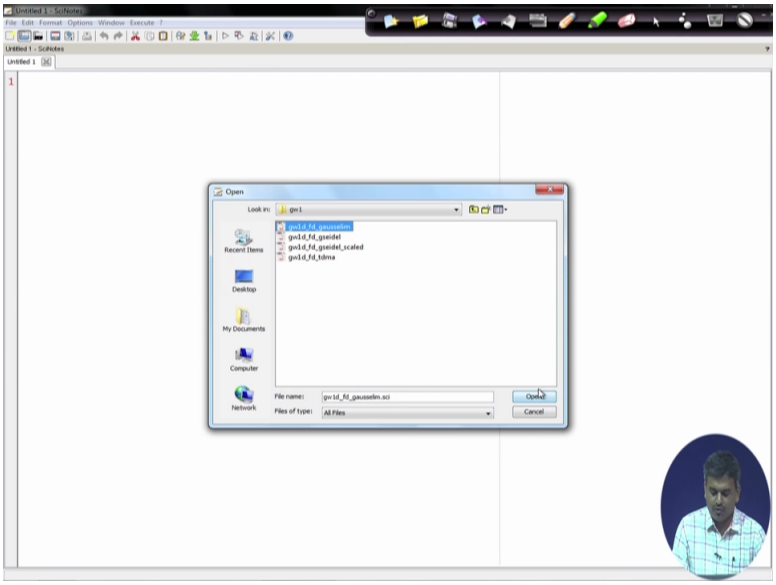
Equation can be further simplified. We can write it in this format. And let us start the problem with first order discretization of our left boundary. This is right boundary condition, because left boundary is already specified or Dirichlet boundary condition.

(Refer Slide Time: 13:04)



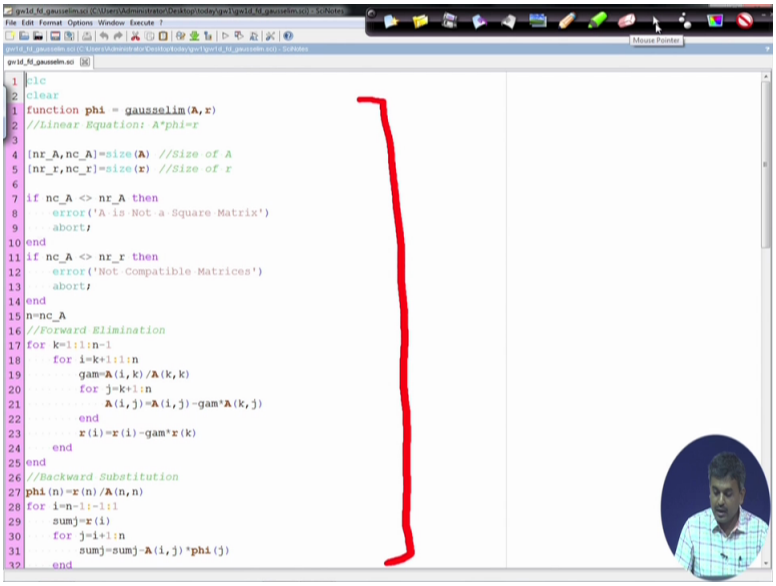
So if we consider our scilab we can open that. Now these are the four codes available. Now let us start with full matrix storage.

(Refer Slide Time: 13:35)



In full matrix storage we will be utilising Gauss elimination. So this is general Gauss elimination algorithm that we have discussed in the lecture number 25. So this is our Gauss elimination algorithm.

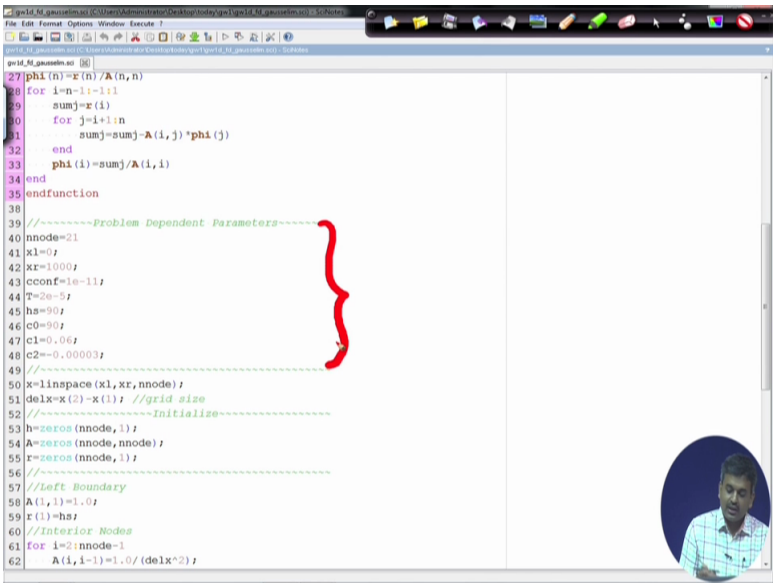
(Refer Slide Time: 14:07)



Here comes the problem dependent parameter. N node is the number of nodes or n. Number of nodes I have defined it as 21. XL, co-ordinate of the left boundary, this is zero. XR, right hand of the X co-ordinate, this is 1000 metres or 1000. Cconf, this is 1e minus 11.

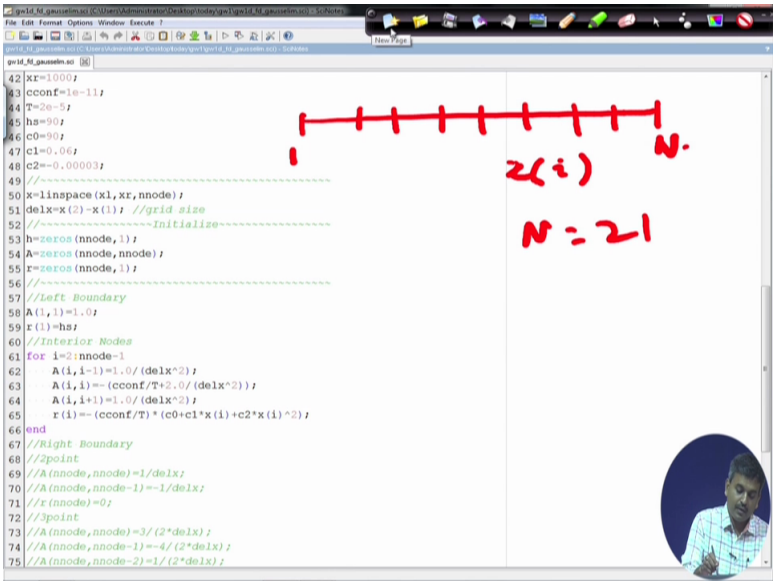
That is 10 to the power minus 11. T is 2e minus 5. Hs is 90, C0 is 90, C1 is point 06, C2 is minus 4 not 3. These are the parameter values we need to specify and based on these parameter values there will be generation of other values.

(Refer Slide Time: 15:12)



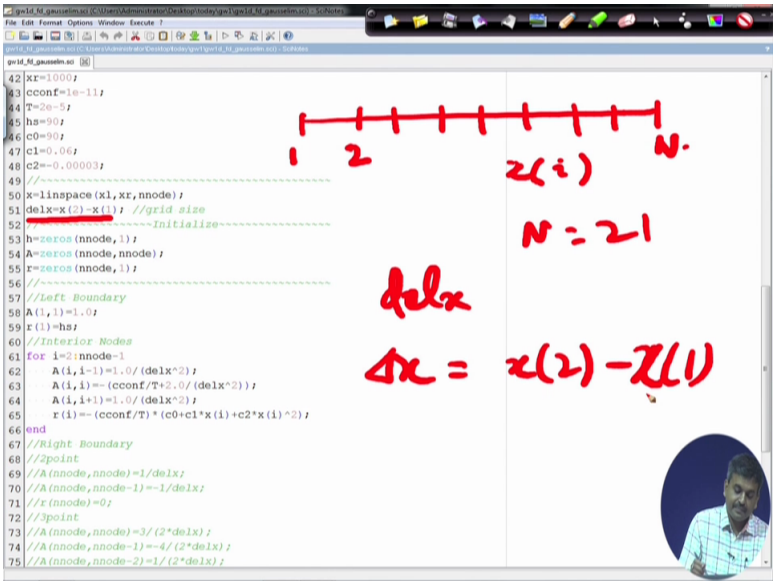
So first of that is x. That means we need x co-ordinate for all the points including the boundary points. So starting from 1 to N we need the x co-ordinate for any i. So x is equal to linspace xl, xr. So from x0 or xl to xr that means zero to 1000 we can generate N nodes or N number of nodes here. That means as we have considered N equals to 21. So including these end points, 21 nodal points will be generated and those will be equidistance.

(Refer Slide Time: 16:24)



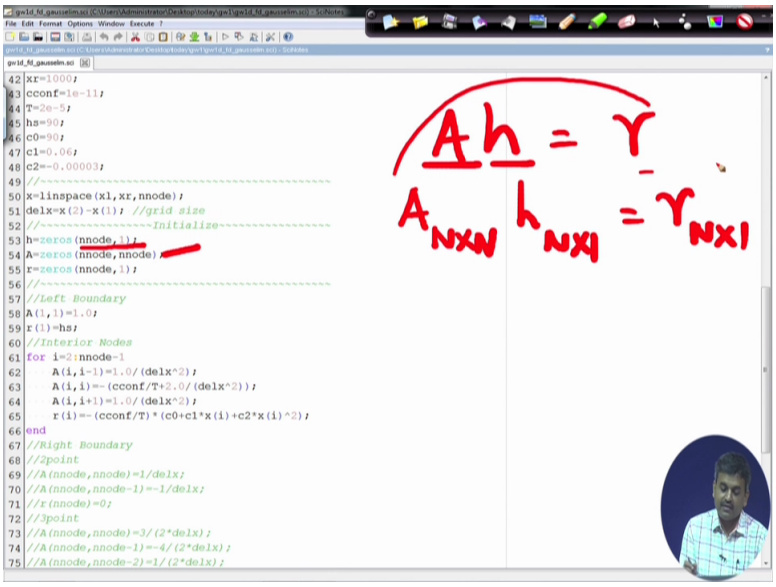
So del x here, if we consider this as point 2, so obviously del x or del x we can write it as x2 minus x1 because del x value is constant for this problem.

(Refer Slide Time: 16:47)



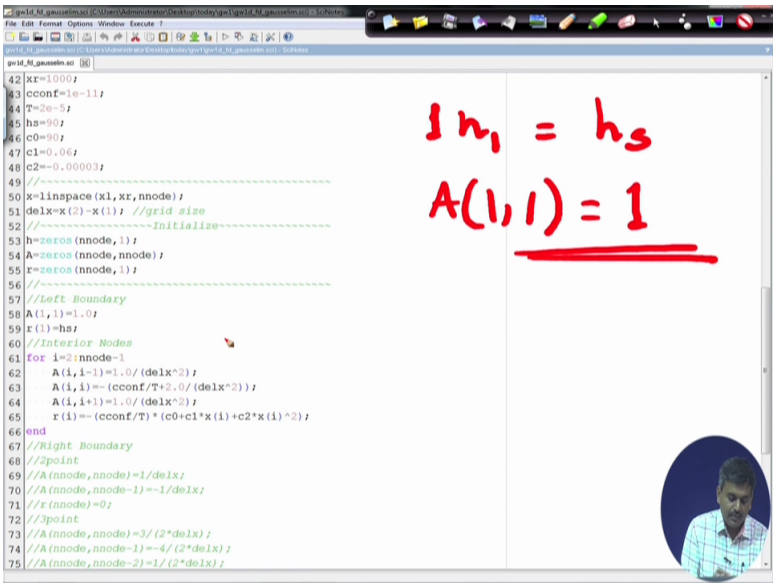
Now we need to define this structure, Ah equals to r. Ah is our desired value that we should get from this problem. A, this is N cross N. That means N nodes cross N number of elements will be there, h is N cross 1, r is N cross 1. So initially we need to generate these matrixes or for our problem. So h equals to zeros N node 1, a equals to zeros N node cross N node and r is zeros N node comma 1. So we have initialised these matrixes here.

(Refer Slide Time: 17:56)



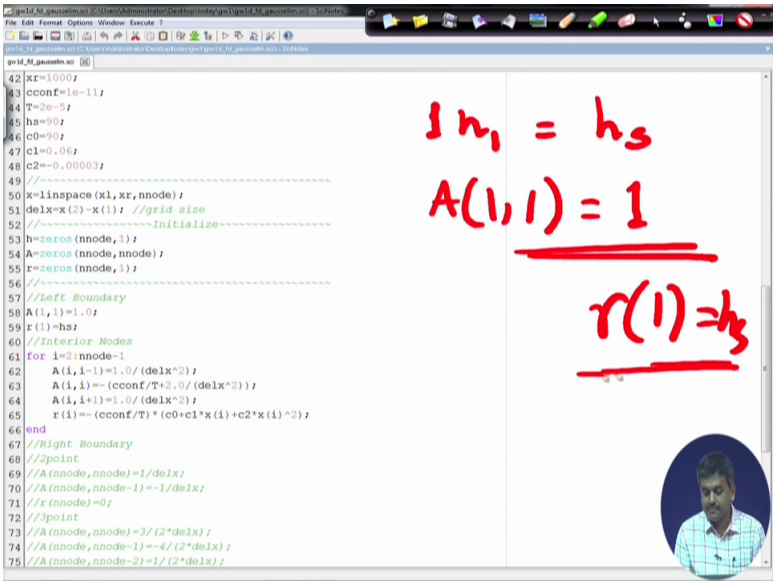
On the left boundary our value is directly specified. That means h1 equals to hs and the coefficient of this one is 1. So I can write it as a11 which is the first entry in the matrix and this is also the diagonal term I have entered at as 1.

(Refer Slide Time: 18:30)



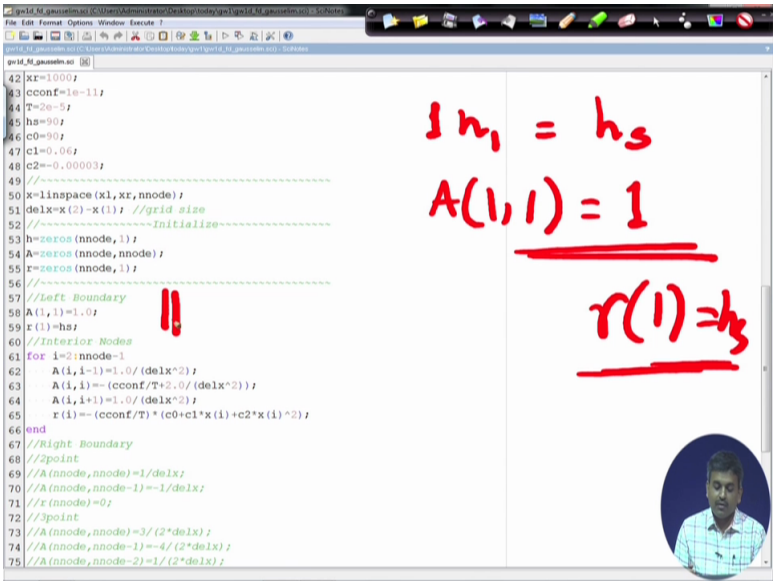
In the right hand side obviously the first entry will be hs as we have initialised the same matrix with zero values. So other countries will be zero. So if we multiply h1 with a11 so we should get this hs value.

(Refer Slide Time: 18:50)



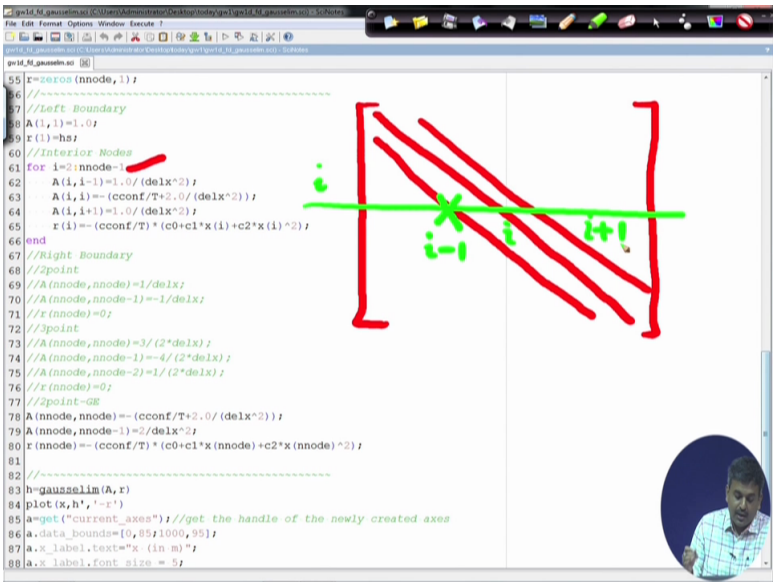
So this is our left boundary, this is the treatment for our specified boundary condition. The coefficient should be 1.

(Refer Slide Time: 18:59)



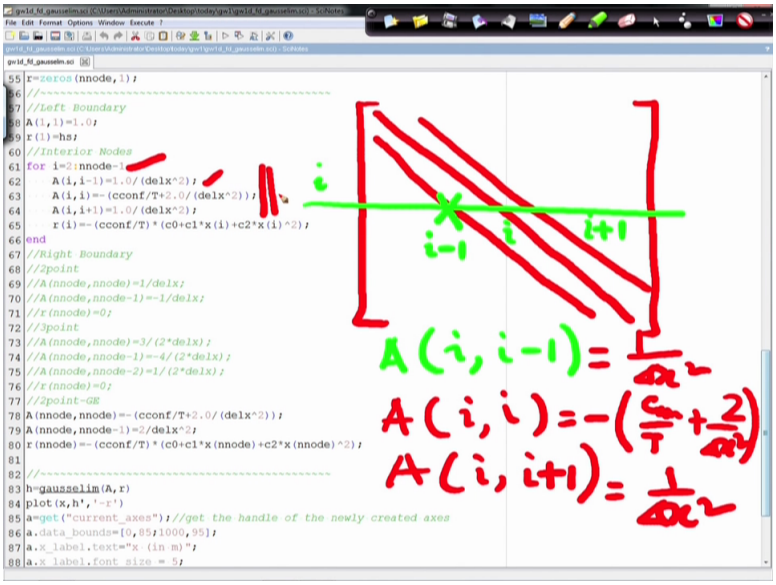
Now for interiors nodes that is starting from 2 to N node minus 1 again I need to define the different entries. So obviously this is having Banded structure but we are considering full matrix. So obviously in this case for any arbitrary row i which is the ith row, we will have i minus 1. This is ith 1 and this is i plus 1 column entry. In this case other columns will be having zero values.

(Refer Slide Time: 20:03)



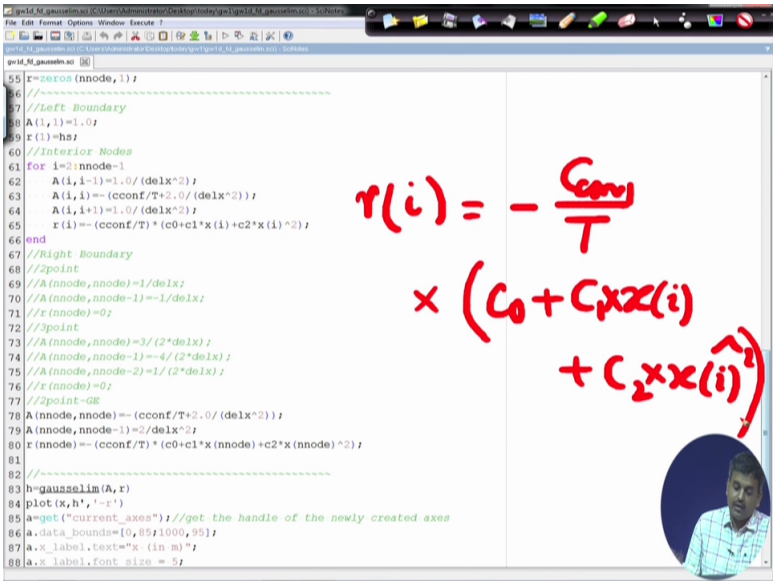
So in this case we will have A i, i minus 1. Others will be A i, i and A i, i plus 1 because we are considering this coefficient matrix with respect to a particular ith location. So as per our discretization on the left hand side this is equal to 1 by del x square, this is Cconf by T plus 2 by del x square. This is also available. Now and A i, i plus 1, this is again A by del x square. So we have defined our left hand side.

(Refer Slide Time: 21:13)



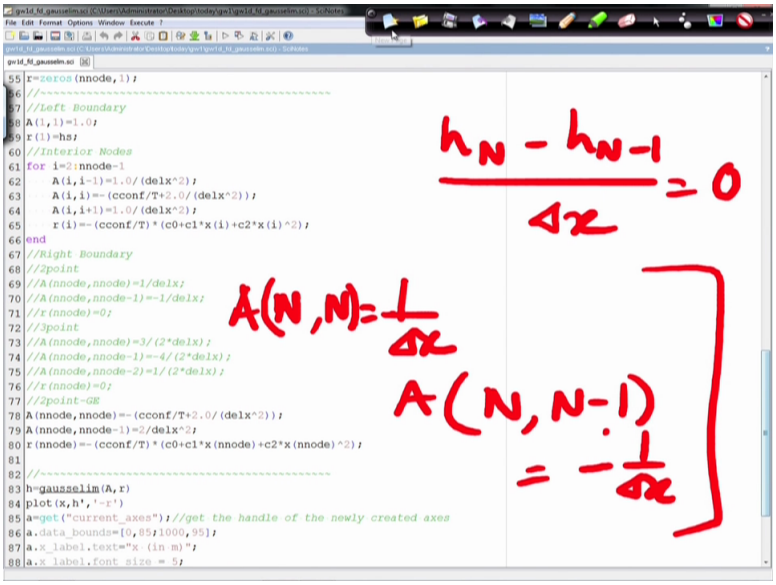
On the right hand side what will be there? Right hand side this entry will be minus Cconf by T into C0 plus C1, x i plus C2, this is x i square. So with this we have generated the information for intermediate nodes.

(Refer Slide Time: 21:59)



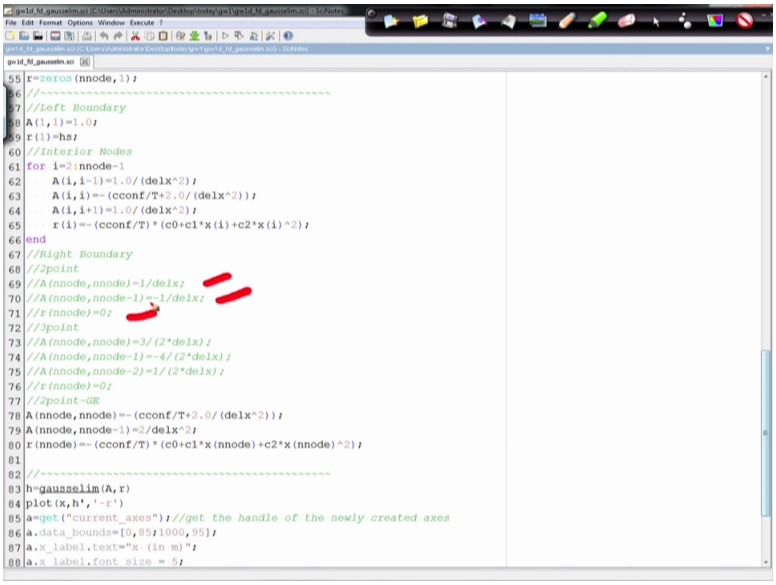
Now if we consider our boundary nodes we can use N, N minus 1 which is the simplest one. In this case this is equals to zero. Or the coefficient of h N, this is 1 by del x and the coefficient of h N minus 1 or I can say that this is N or N node, whatever we say, this is equals to 1 by del x. And A N N minus 1, this is minus 1 by del x. With this we can solve this problem.

(Refer Slide Time: 23:03)



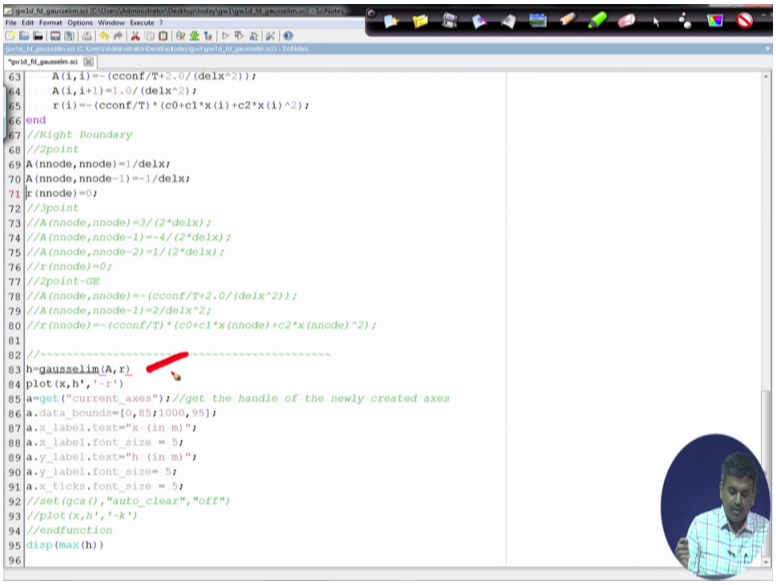
So this is the corresponding entries here. And the right hand side this is r equals to 0.

(Refer Slide Time: 23:12)



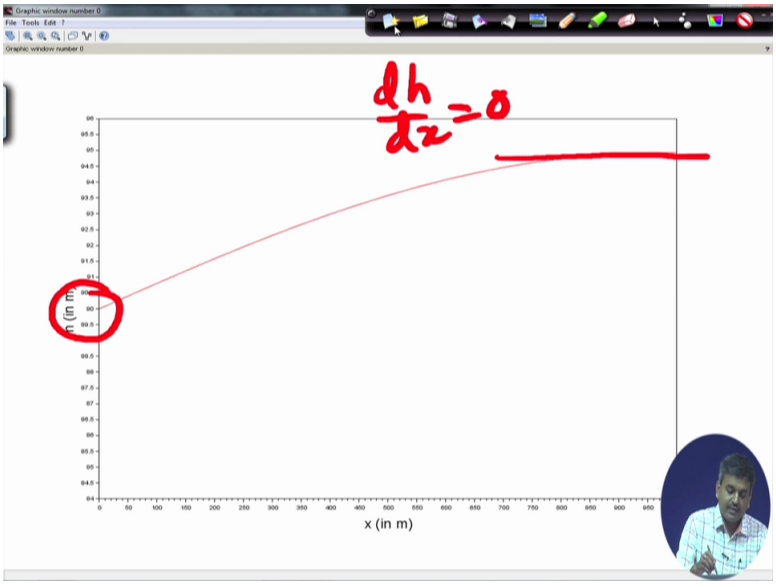
So I can come in this one. This is for other case. We will discuss that. So if these are the three entries on the last row and finally we will be using this Gauss elimination thing for solution of our problem, h equals to gausselim a, r. And finally we need to plot it. So plotting will be h x h transpose because we are converting it into row vector, x is row vector here.

(Refer Slide Time: 23:58)



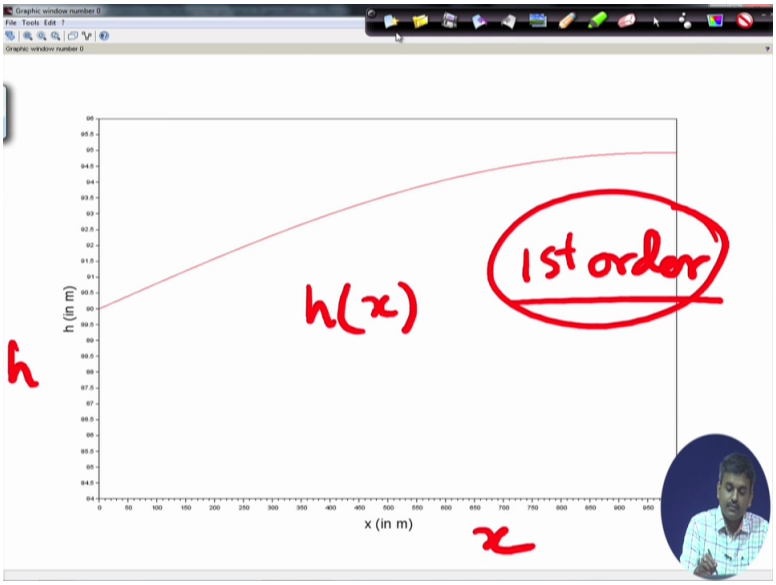
So let us run it, okay. So in this case you can see on the left hand side the value which is coming as 90. On the right hand side we don't know the exact value. We know it from the plot but we cannot verify it from our defined boundary condition. Defined boundary condition is dh by dx equals to zero in this case. If dh by dx equals to zero so obviously slope at this place is parallel to x axis. And obviously that kind of result we are getting in this case. So we can say that our solution is more or less correct. However we need to verify it.

(Refer Slide Time: 25:08)



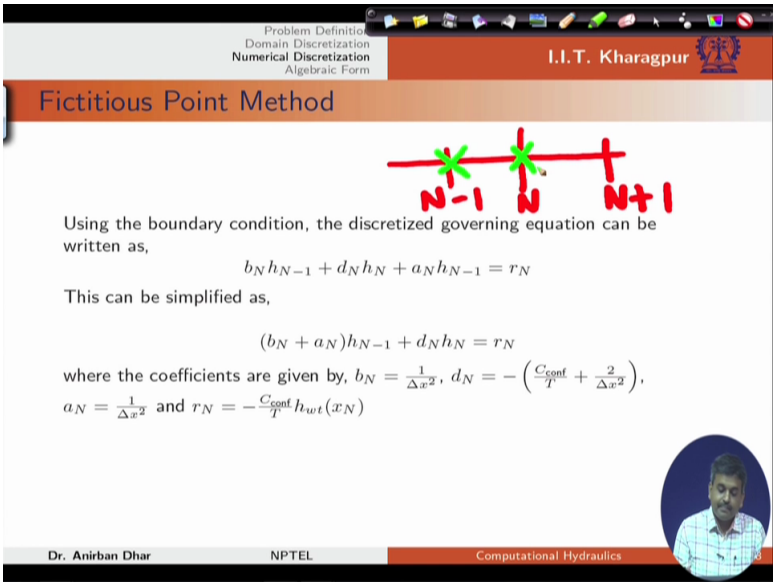
So we have got this variation of h with x. In this direction we have x, and in this direction we have h. So this is the variation of h with x with first order condition.

(Refer Slide Time: 25:36)



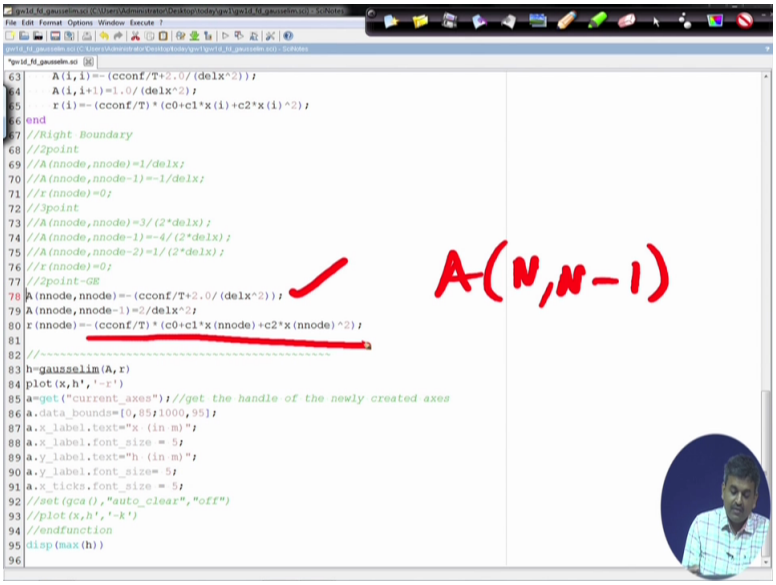
Now if we consider our original thing then we can have fictitious point method where we have considered N, N minus 1 and N plus 1 points. And we have seen that by defining the coefficient of these two points in the original matrix we can solve this problem.

(Refer Slide Time: 26:27)



So let us try to solve this problem with our approach. So two points, the coefficient of this matrix, central coefficient will be same as our original problem or for interior nodes. And the term which is corresponding to A N, N minus 1, that will be 2 by del x square because we have extra time here to represent the second order accuracy. And the right hand side x not zero. Again we need to calculate the value like our interior equations.

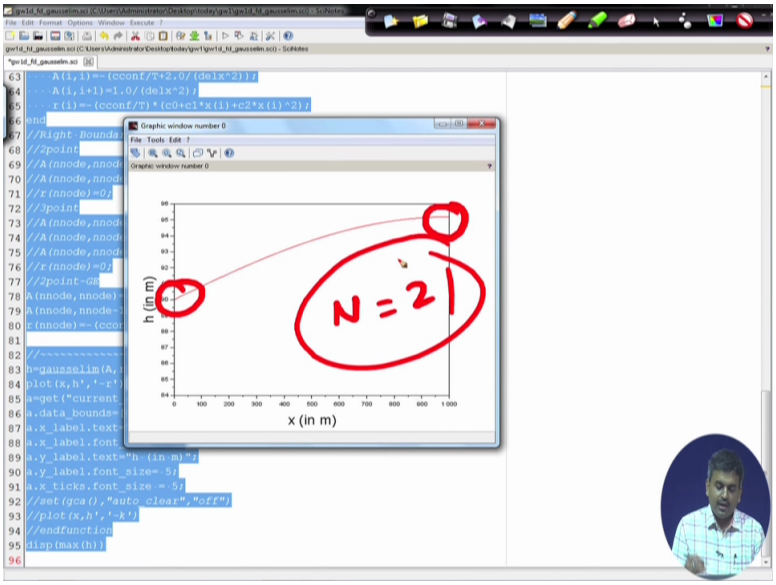
(Refer Slide Time: 27:48)



So let us see what is the solution out of this? So again we are getting the solution for this problem. Now one thing is clear from this problem that we are getting solution for both the cases but on the left hand side it is exactly (spe) satisfied because it is a specified boundary condition. On the right hand side there will be some amount of deviation with respect to our first order solution.

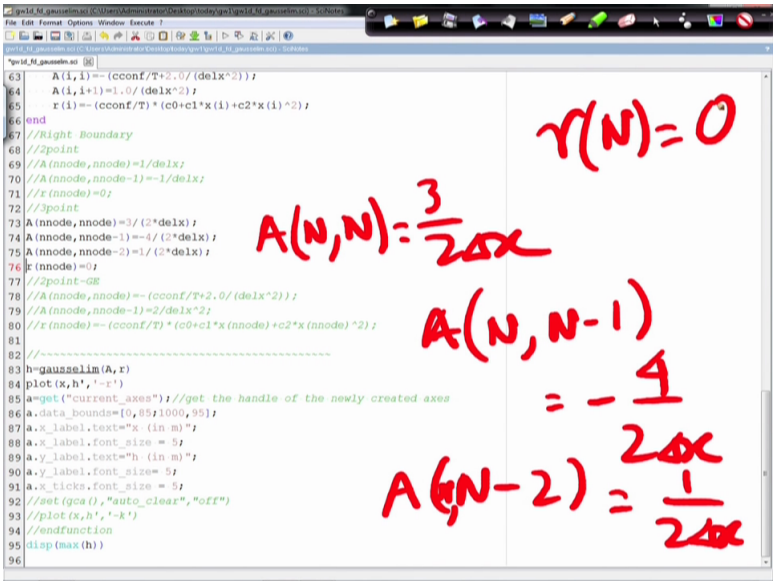
If we use less number of points, in this case we have used 21 points. Let us say if we use 11 points or 11 points in this case we can see there will be prominent deviation in the boundary. Because it is not directly specified, it is in terms of Neumann boundary condition.

(Refer Slide Time: 28:55)



Now if we check our discretization, accuracy of the boundary condition for 3 nodes case, again we can utilise the three node case for our problem. This is 3 by 2 delta x 3 by 2 delta x for the diagonal term A N, N, A N node minus 1. This is our minus 4 2 by delta x and A N, N minus 2, this is 1 by 2 delta x. Obviously if we specify this and if we put r N equals to zero on the right hand side then again we should get the solution.

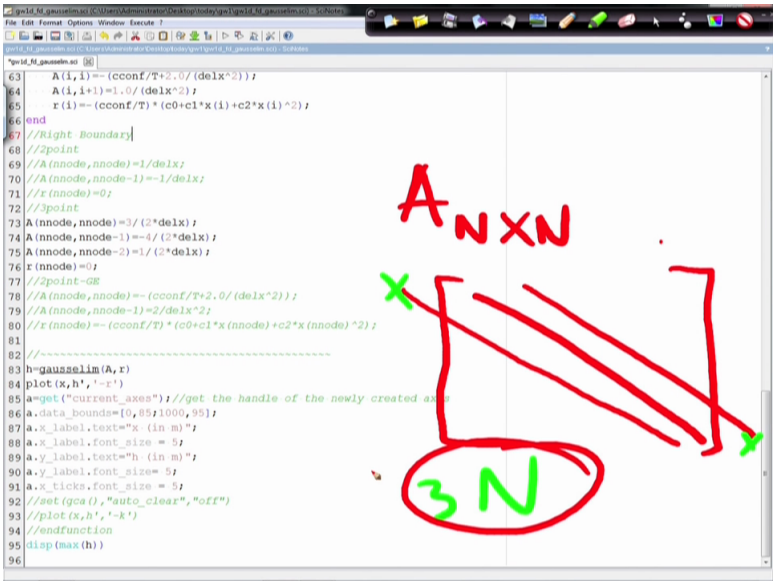
(Refer Slide Time: 30:13)



So again we are getting the solution using our Gauss elimination approach. Now let us see if we can reduce our computational burden by considering the Banded structure of the matrix. Because in this case we have considered A N cross N number of entries although those entries are having zero values, we have considered the full matrix.

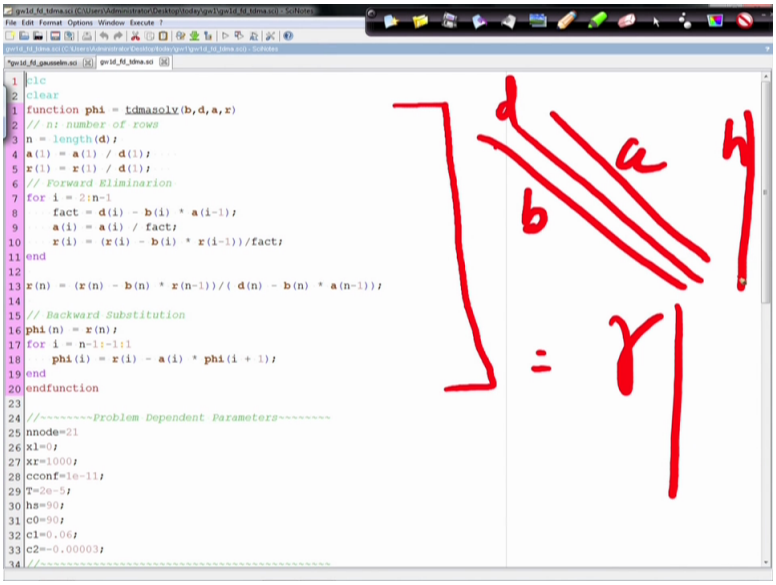
Now if we consider the part of the matrix, this is diagonal term, there will be one extra term on both sides, this side and this side because if we consider the column vector as N, so obviously we need to consider 3N number of entries in this case. So with this 3N number of entries we can solve our problem without having the full coefficient matrix. So that approach is tri-diagonal matrix algorithm.

(Refer Slide Time: 31:43)



Now if I open the tri-diagonal tdma ground water 1D finite difference tdma. So this first part is familiar one. This is tdma solver. We need to store the Banded that thing. This is diagonal, this is below diagonal, this is above diagonal, here it is h matrix and right hand side we have r matrix that is again column vector, this is again column vector.

(Refer Slide Time: 32:32)

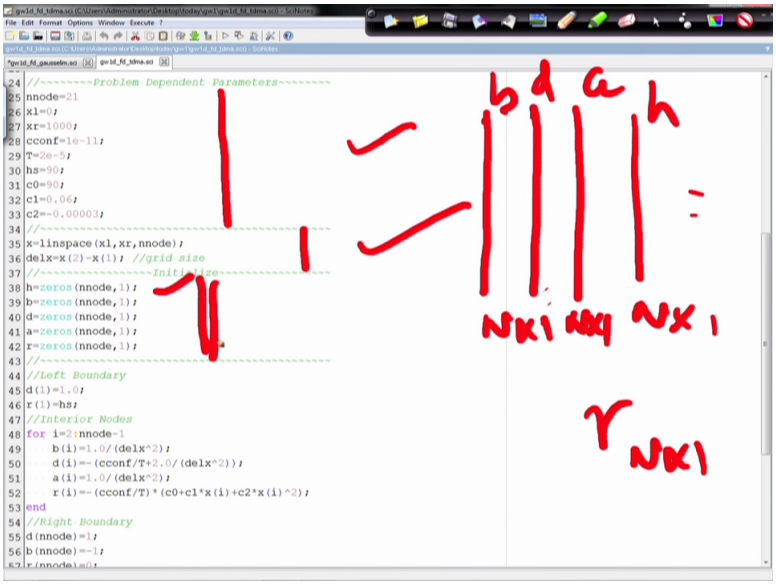


So these two are common for both full matrix and Banded matrix case. Only difference is here because we need to store only (three) 3N number of elements directly in this case. So let us see what are the changes in our problem section? So first part, these are the problem dependent parameters. There is no change here. Again the second level defining the x

coordinates and finding out the grid size. This is also same. Next initialisation is different because h, this is zeros N node 1.

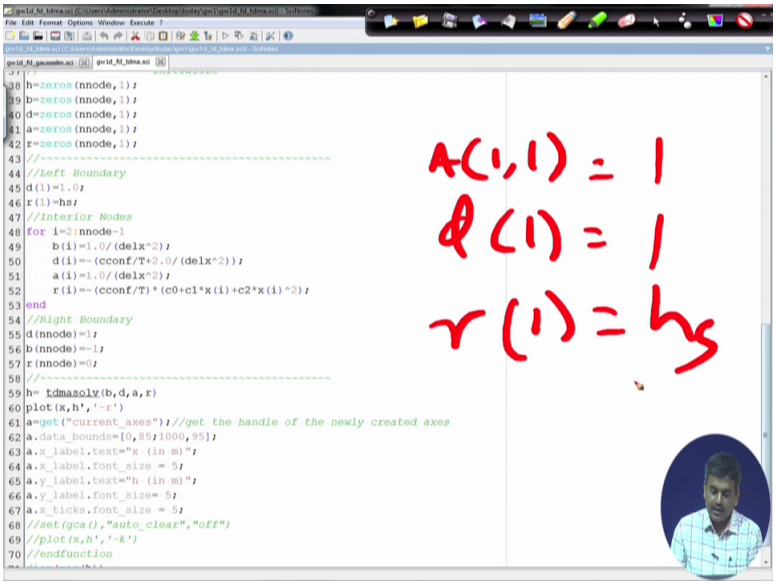
That means we are defining column vector for h, b that is below diagonal. We are defining one column vector. Again this is diagonal. For diagonal also we are defining this N cross 1 thing. Again for above diagonal we are again defining this N cross 1 structure. On the right hand side we will have this r, again N cross 1. So this is the initialisation of our column vectors.

(Refer Slide Time: 34:12)



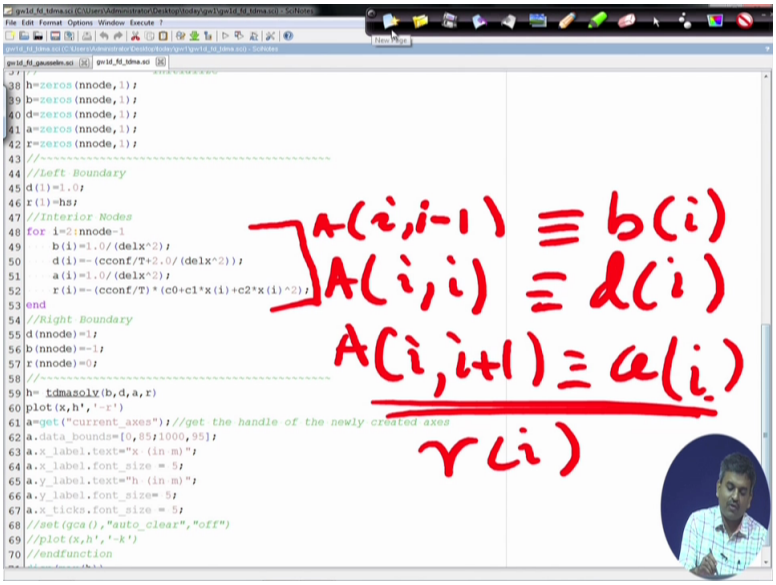
Now with this if I proceed obviously in our full matrix case we have seen that entry for the diagonal term in case of specified boundary condition is 1. So obviously here the equivalent thing is d 1 equals to 1. And the right hand side the entry will be same because hs that is for our full matrix case or this Banded matrix case. That is same because again we are considering this r vector here.

(Refer Slide Time: 34:57)



So in place of A i i minus 1 we are defining this is equivalent to b i. Again this A i i, this is equivalent to d i and A i i plus 1 in our full matrix case this is equivalent to a i and r i is same for these two cases. Now with this we can define the things for interior nodes. What will be the case for right hand boundaries with first order, let us say first order boundary condition.

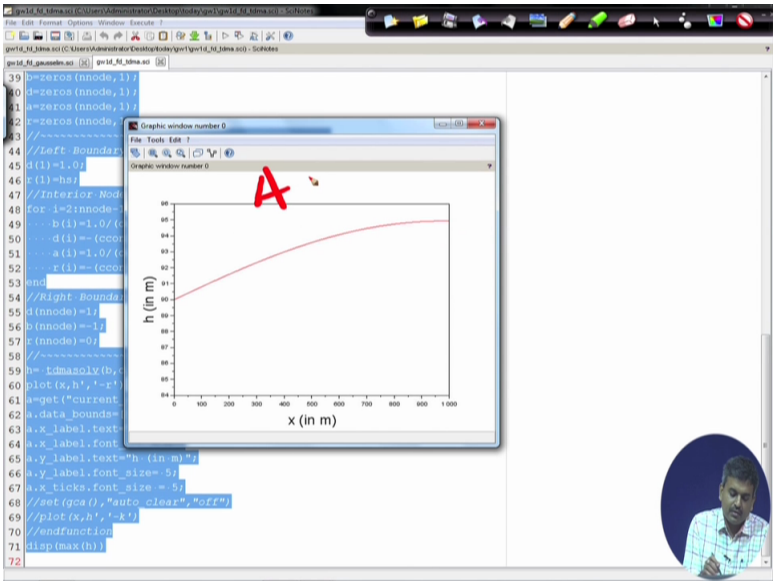
(Refer Slide Time: 35:49)



We have seen that with other boundary conditions also we can solve this problem. So let us test this Banded matrix structure with first order accuracy. So if I run this again we are getting some solution and if we increase the number of nodes N then we can see there will be

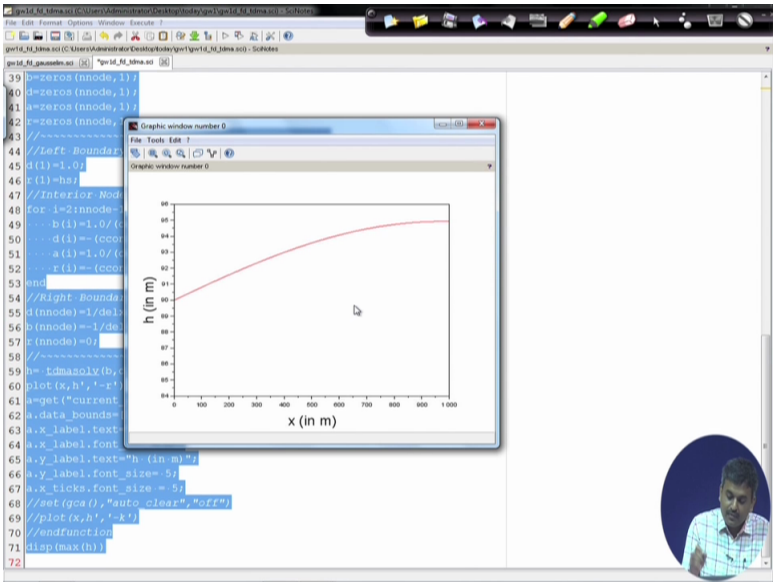
difference because the matrix case where there will be a full matrix structure, obviously the running time will be more compared to tdma structure.

(Refer Slide Time: 36:37)



Now we have solved this problem using our Banded matrix case. In this case we can also include the del x thing. We can divide it by our del x. This is first order condition. Without del x also this will work. So this is with del x or without del x whatever you do, it should work.

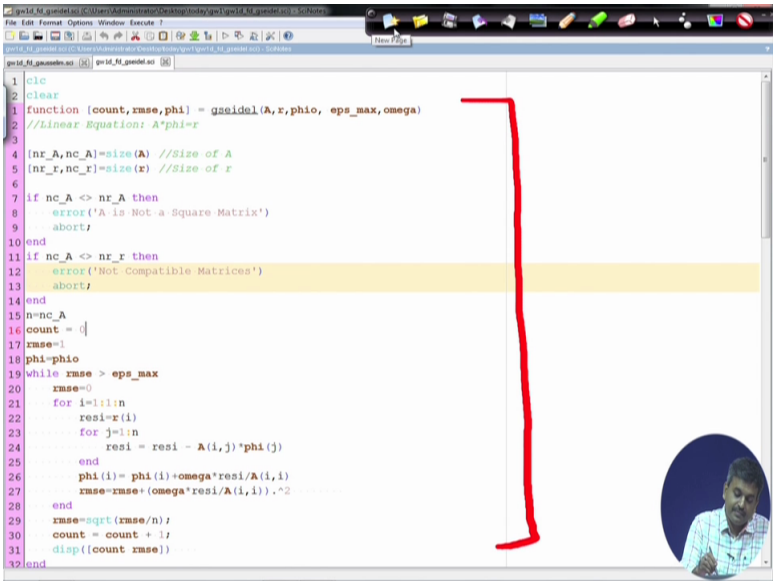
(Refer Slide Time: 37:28)



Now I will try to solve the same problem using iterative algorithm. Let us say that we have Gauss Seidel algorithm. This algorithm part, this is common because we have derived it in

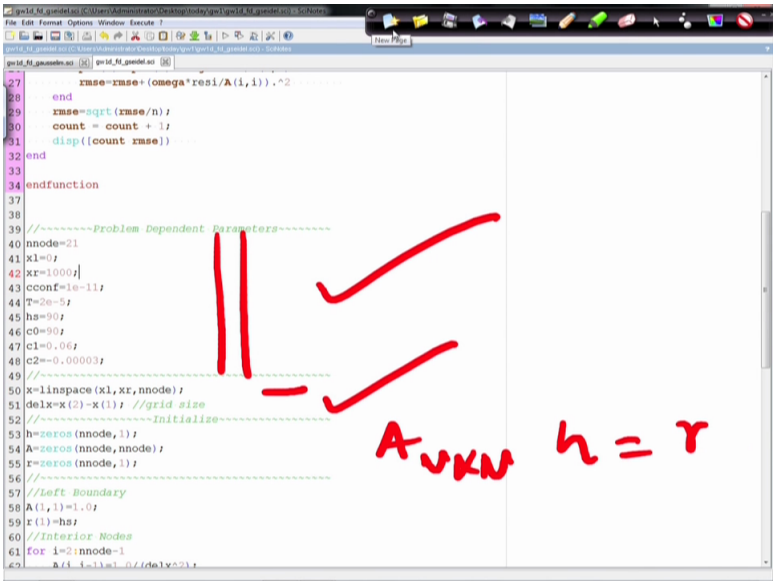
our lecture number 29 and we have discussed it there. In this case we are considering the full matrix.

(Refer Slide Time: 38:15)



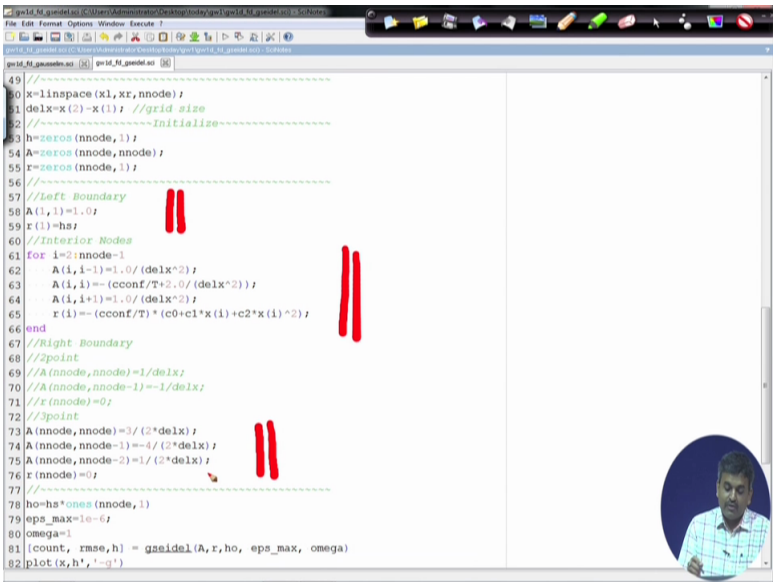
Let us see what will be the difference compared to the full matrix case that we have considered with Gauss elimination or tdma. Let us say we have maybe 31 or 21 points in this case. 21 points similar to our previous problem. X 0, xr 1000 these values are same like our previous problem. Now x is linspace, this part is also common. These two sections are common. Like our Gauss elimination we are considering full matrix. So A we need to consider again A N cross N, h and r.

(Refer Slide Time: 39:18)



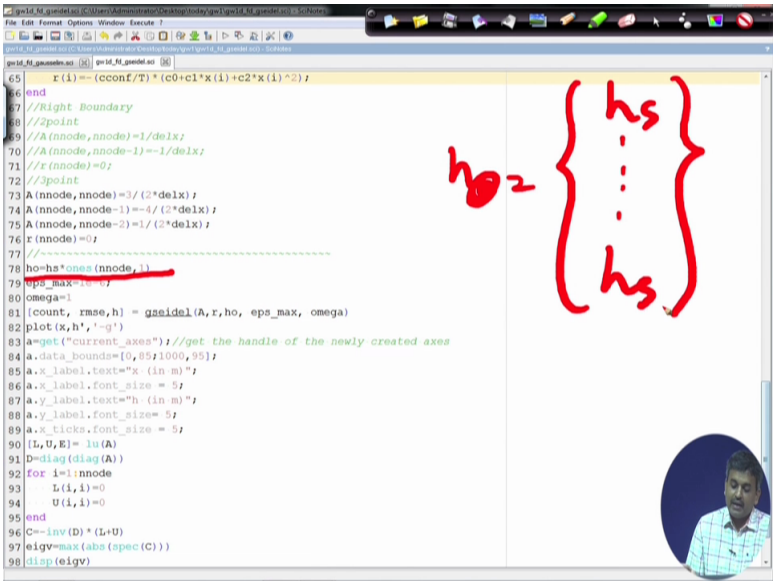
So this is the basic initialisation of our matrix structure. Now like our Gauss elimination we can create the boundary condition on the left hand side, interior nodes and in this case I will be talking specifically about this 3 point boundary condition. So in this case if we utilise this 3 point boundary condition what will be the situation.

(Refer Slide Time: 40:08)



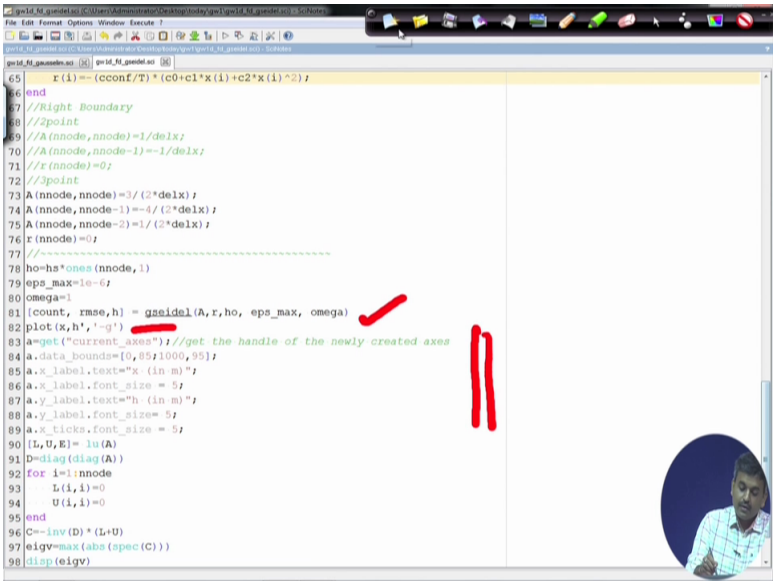
Let us define the initial value which is the guess value for the problem. Ho is the initial value. I have defined it as hs into once n node comma 1. That means for full hs this initial vector I have stored hs value for all the points. That means initial guess is equal to the value specified on the left hand boundary.

(Refer Slide Time: 40:56)



Now we can call our general structure, this Gauss Seidel one. And finally after getting this h again we can plot it using our plot command. So up to this, this is specifying h and x values.

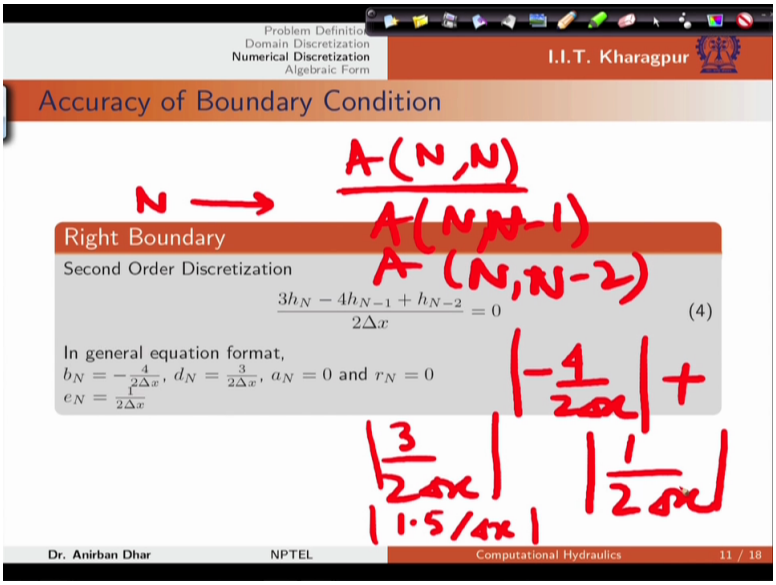
(Refer Slide Time: 41:28)



Now if we consider our right hand boundary, one interesting point is there. Interesting thing is for entry in the Nth row which is corresponding the Nth row, our coefficient term A N N is 3 by 2 delta x. 3 by 2 delta x is 1 point 5 divided by delta x. Now we can take mod of this. For off diagonal terms that is the coefficient of A N N minus 1 or A N N minus 2, we have minus 4 by 2 delta x.

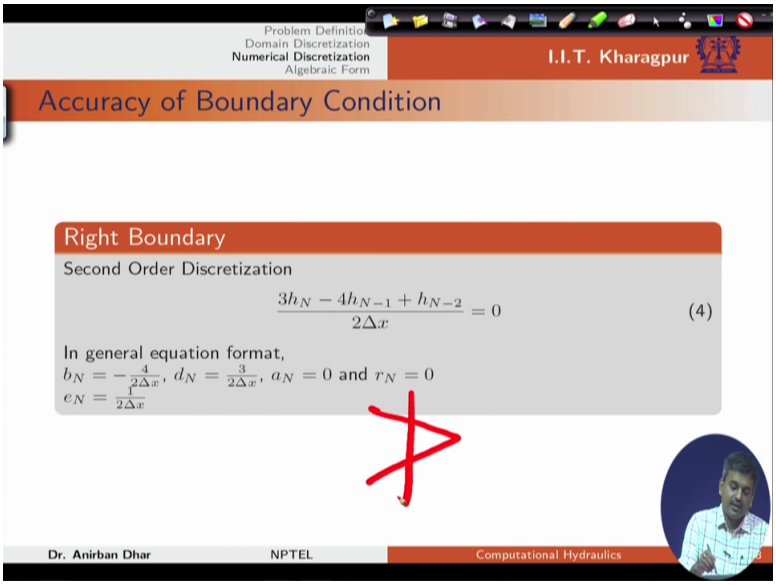
And this is 1 by 2 delta x. Obviously if we add the coefficients or absolute value of the coefficients of the off diagonal terms that is 2 point 5 divided by delta x.

(Refer Slide Time: 43:09)



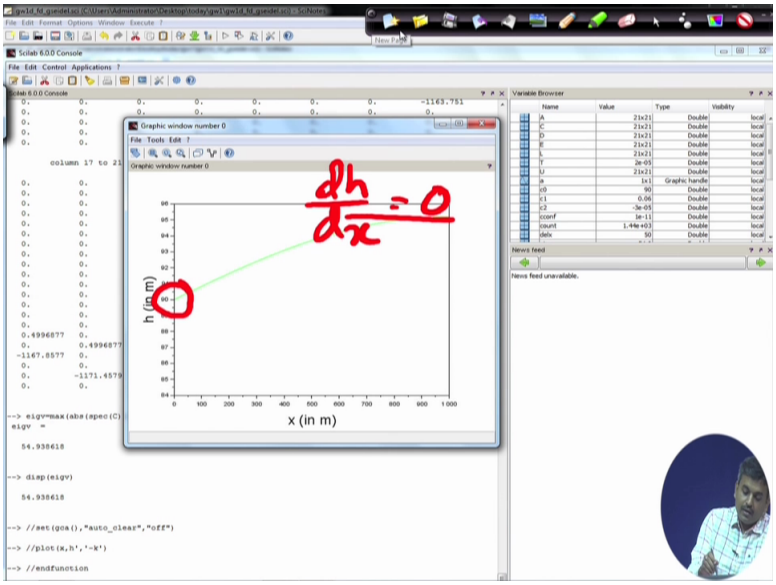
So obviously in this case 1 point 5 or coefficient of the diagonal term is not greater than the summation of the coefficient of the absolute values or coefficient with absolute values. So in this case our diagonal dominance is not satisfied.

(Refer Slide Time: 43:43)



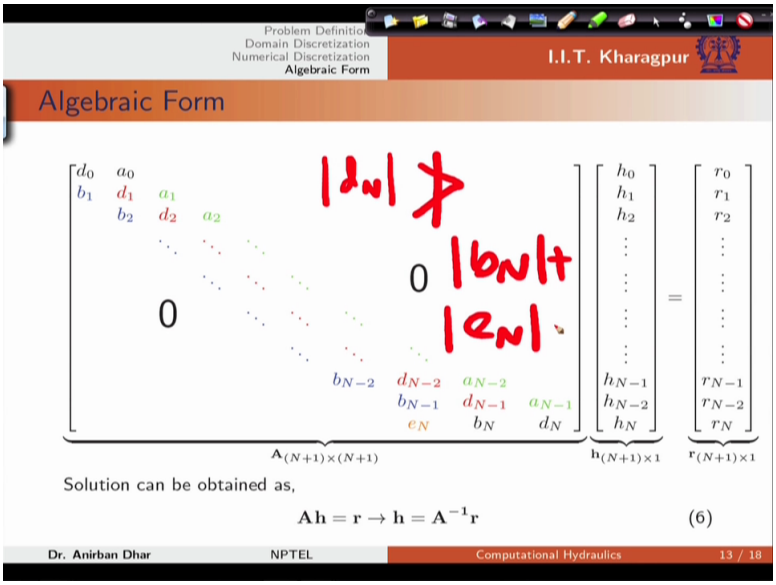
Now let us see what will be the solution from this case. Ideally this should not work for our problem. So let us run it using our algorithm. So if we run it we can see that we are getting some solution out of this. And the solution again this is 90 and again it is matching in terms of physical nature or the problem. It is interesting that by seeing the boundary condition itself we can test the nature of the solution or whether your solution is of correct type or not.

(Refer Slide Time: 44:54)



So in this case the things are working. But still the diagonal dominance criteria is not satisfied. So what is the reason behind this thing? If we closely look into our system then we have considered this two point Banded structure. If you have three point Banded structure and in this three point Banded structure we have seen that this dN is not greater than absolute value of bN plus absolute value of eN. So that should not be the case for convergence of the method but still we are getting solution out of this problem.

(Refer Slide Time: 45:50)

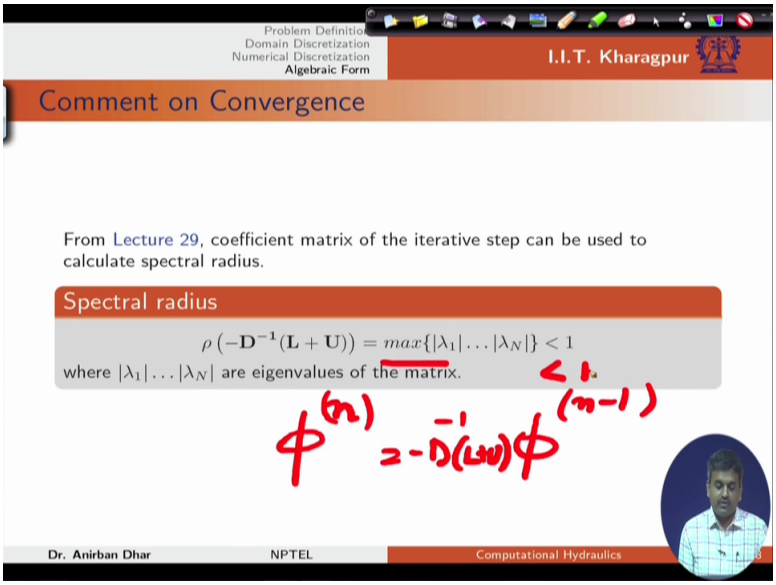


In this case we can consider this spectral radius. So what is this spectral radius or row? Row of the coefficient matrix we have considered phi n and phi n minus 1. And this is the

coefficient matrix which is D inverse L plus U in this case. And for this coefficient, if the spectral radius which is nothing but the maximum of the absolute values of the Eigen values.

So if we take modulus or absolute value of all Eigen values starting from 1 to N and if we select the maximum value out of that, this value should be less than 1. Then we can say that our algorithm, this one will converge in this case.

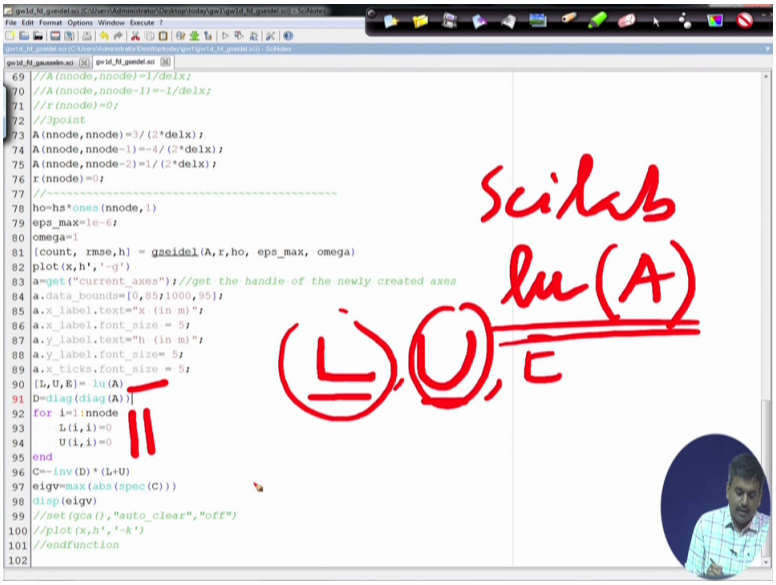
(Refer Slide Time: 47:16)



So let us examine from our problem whether the spectral radius is less than 1 or not. So what I have done? I have called this internal function of scilab, scilab internal function lu of A. A is our matrix or coefficient matrix. If I call this lu, it will automatically generate L U and another E matrix which is related to another information. But L U are not strictly upper or lower triangular matrixes. So it contains the diagonal term.

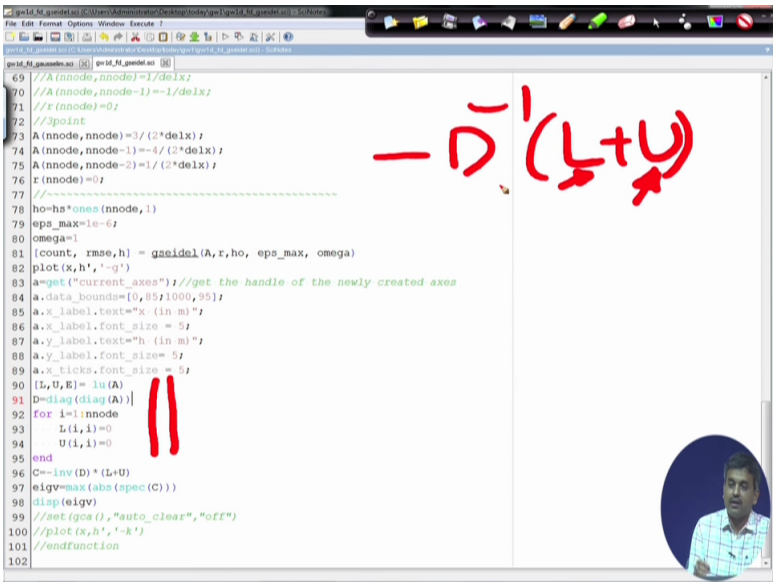
So what we can do, after calling this L U function I can specify L i i which is the diagonal term equals to zero, U i i the diagonal term equals to zero. So that way I can convert it into strictly lower and strictly upper diagonal matrix.

(Refer Slide Time: 48:55)



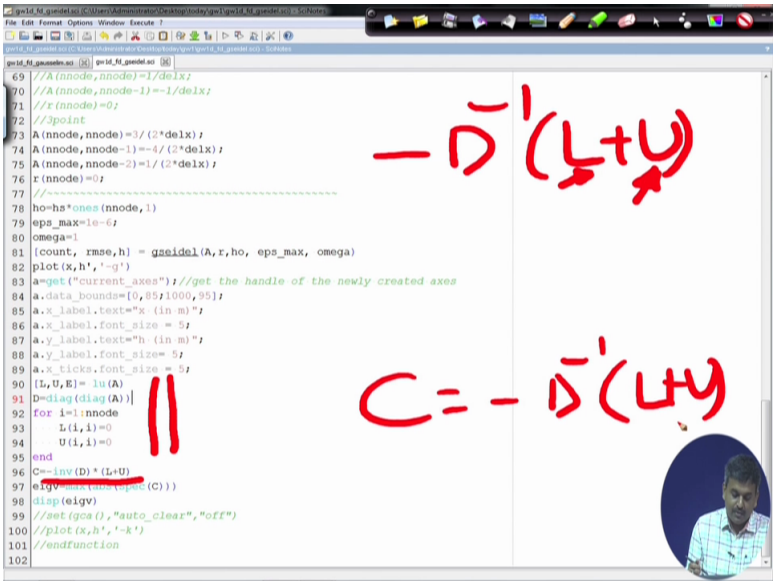
Then you will have diagonal matrix only continuing with the diagonal term. I have generated that using this direct command. Now let us see what is this D inverse L plus U? Now this is strictly lower triangular matrix, this is strictly upper triangular matrix and D is our diagonal matrix.

(Refer Slide Time: 49:22)



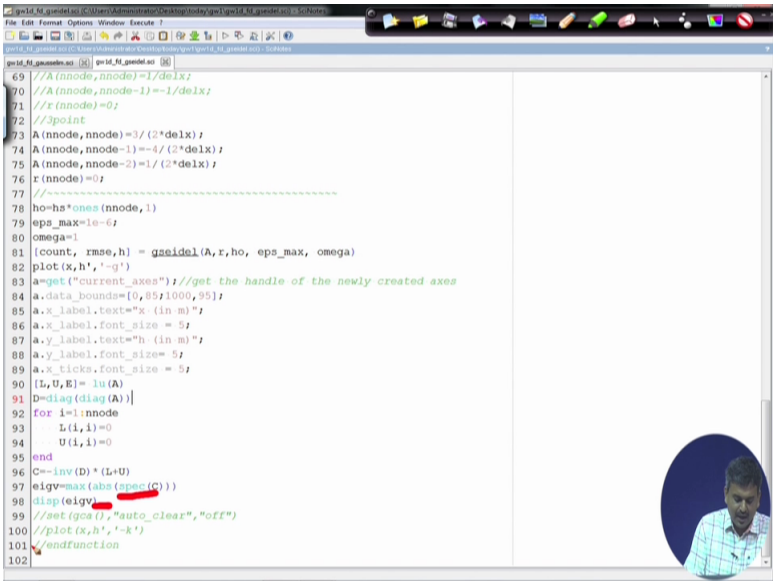
Now negative of inverse of this D into L plus U, this is C. I have converted this into this matrix C. Now if I can figure out what is the maximum Eigen value of this one.

(Refer Slide Time: 49:46)



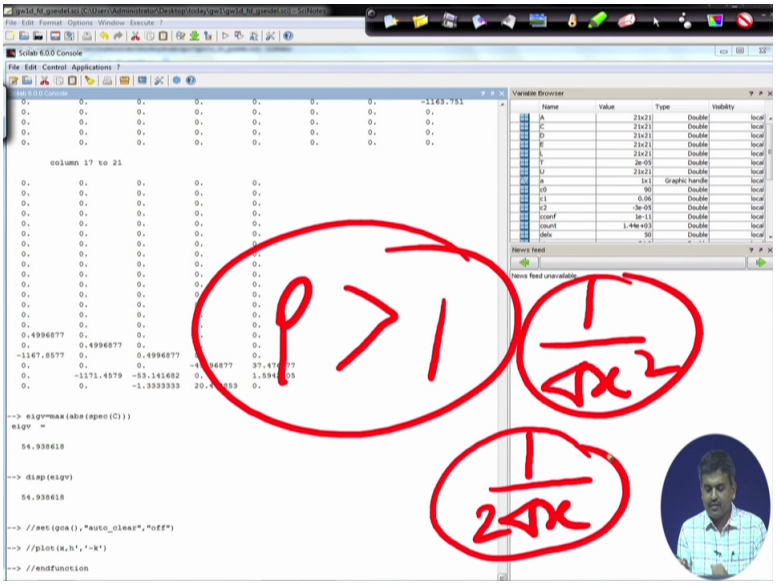
So I have taken spec. Spec will give me this Eigen values, absolute of this Eigen values, abs command and then maximum of this Eigen value and I can display this Eigen value.

(Refer Slide Time: 50:01)



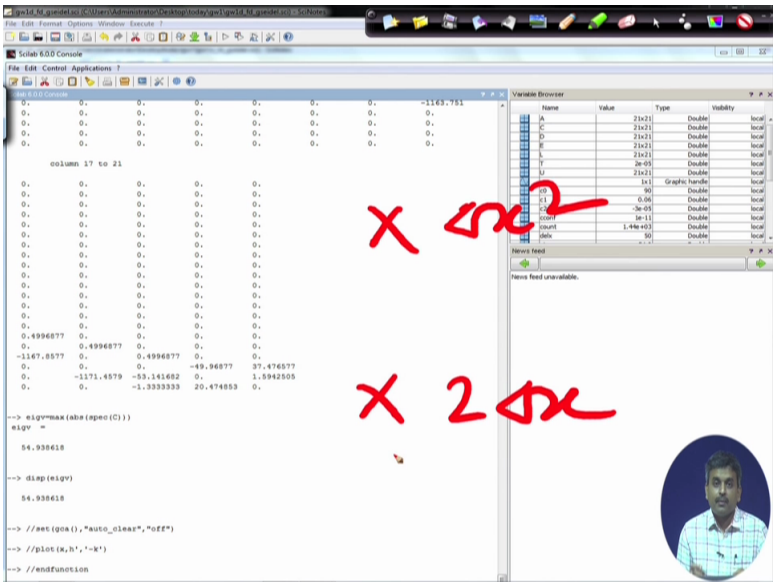
So from our previous problem if we see what is the Eigen value, the Eigen value coming here is 54 point 93816. Obviously this Eigen value condition is not satisfied in this case. But let us examine it from other point of view. If we can change something and get the desired thing because there will be some scaling problem. We are dividing our interior equation by 1 by del x square. That is why it is converting into some small numbers. Again we are dividing our the boundary condition on the right hand side by 2 delta x. Again there is some reduction in order.

(Refer Slide Time: 51:14)



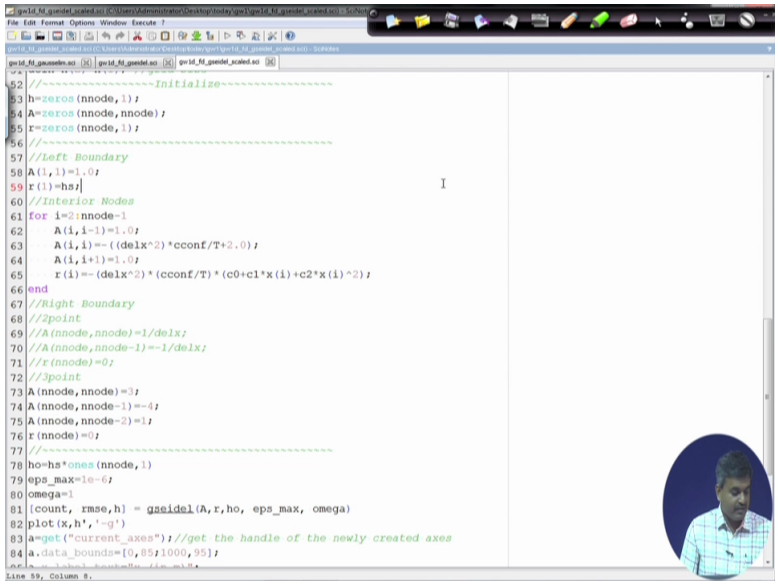
So by removing this, so what I will do, I will just multiply del x square with the interior equation and 2 del x with the boundary condition on the right hand side or with the last row. So I can get a scaled coefficient matrix and then I can examine what will be the case for this particular problem.

(Refer Slide Time: 51:46)



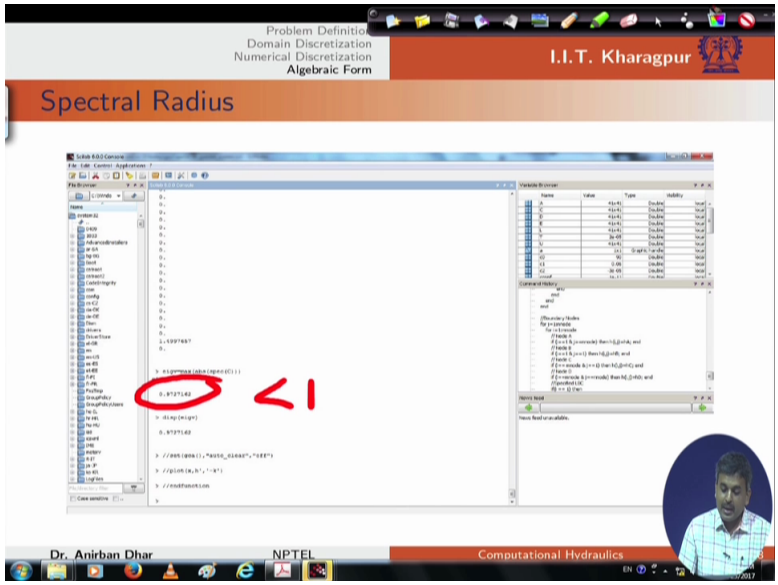
So what I have done here in scaled thing, this is same as previous one but I have multiplied delta x in all cases and 2 delta x for boundary conditions.

(Refer Slide Time: 52:18)



If I run this code so it will run it for convergence. If I see the actual value for this one, after convergence we will see that Eigen value is coming less than 1. So it is a scaled effect due to which we are not getting Eigen value less than 1. If we multiply it by delta x square obviously we are getting this Eigen value less than 1. So we can in for something out of this problem. The spectral radius is another important parameter in this case.

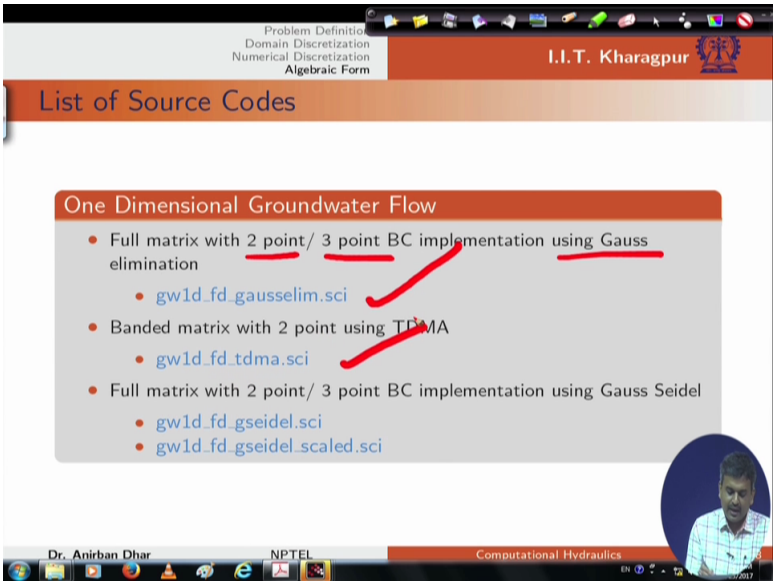
(Refer Slide Time: 53:12)



So if we closely look into this, this is point 9727162. Now in this particular lecture we have covered one dimensional groundwater flow, full matrix with the two point or three point boundary implementation using Gauss elimination. This is the code groundwater 1d fd

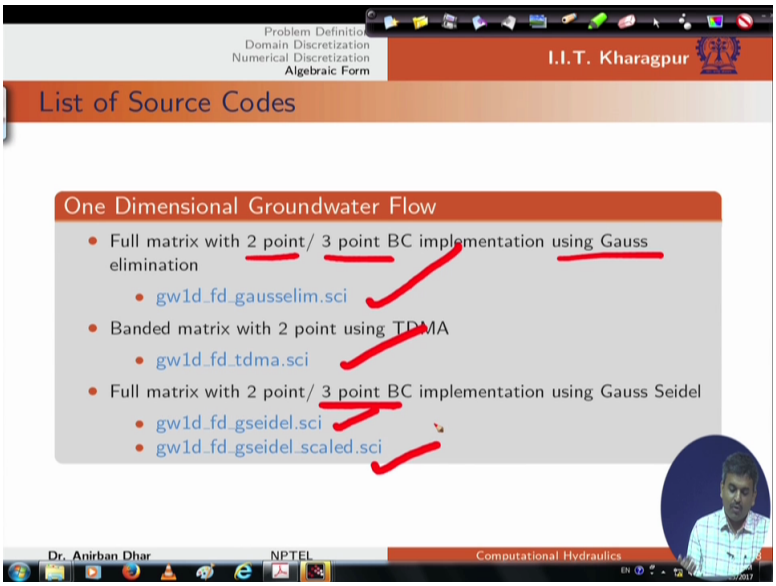
gausselim. I will provide this code so that you can verify the results that I have showed during this lecture class. Second code I have discussed that is groundwater 1d Fd tdma sci.

(Refer Slide Time: 54:02)



And third one is the full matrix with two points or three points boundary condition implementation. But I have discussed only this three point thing. You can check the two points thing also from this one. So Gauss Seidel, this is the normal one and this is the scaled one by multiplying del x square, the interior points and 2 del x with the right hand boundary condition.

(Refer Slide Time: 54:33)



So with this I can just end this particular lecture so that we can proceed to 2D groundwater flow equations. Thank you.