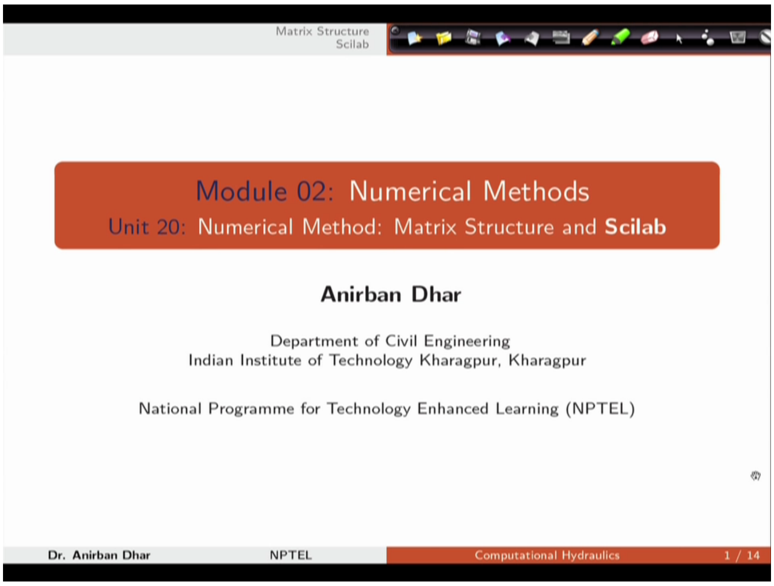
Computational Hydraulics
Professor Anirban Dhar
Department of Civil Engineering
Indian Institute of Technology Kharagpur
Lecture 24
Numerical Method: Matrix Structure and Scilab

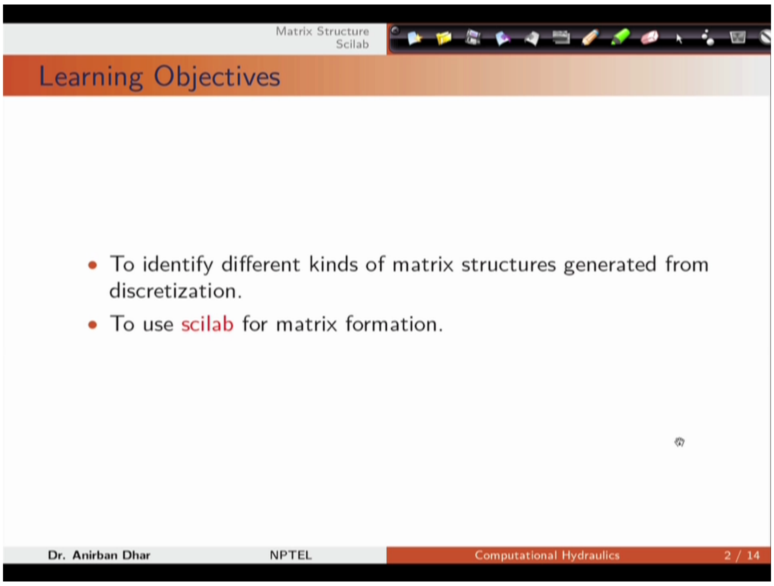
Welcome to this lecture number 24 of the course, computational hydraulics. We are in module 2, numerical methods. And in this particular lecture class I will be covering numerical method part and mostly matrix structure and introduction of scilab software.

(Refer Slide Time 00:45)



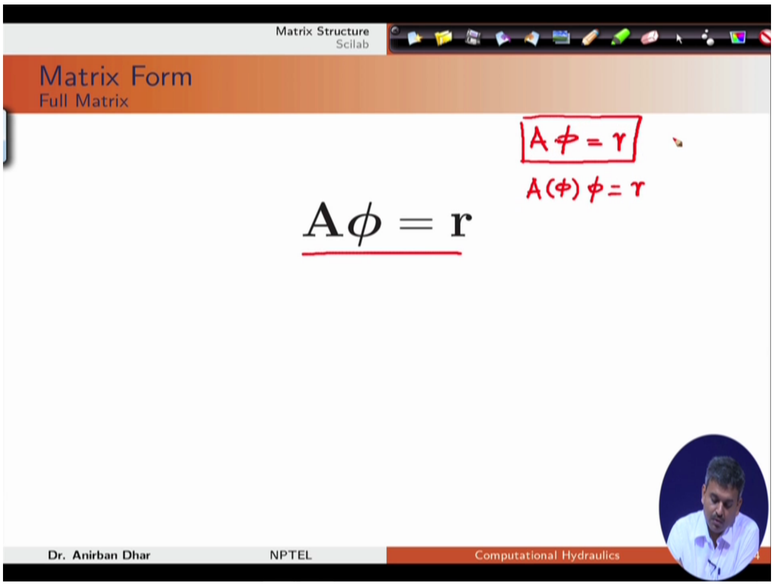
Learning objectives. At the end of this unit students will be able to identify different kinds of matrix structures generated from discretization. And students will be able to use scilab for matrix formation.

(Refer Slide Time 01:12)



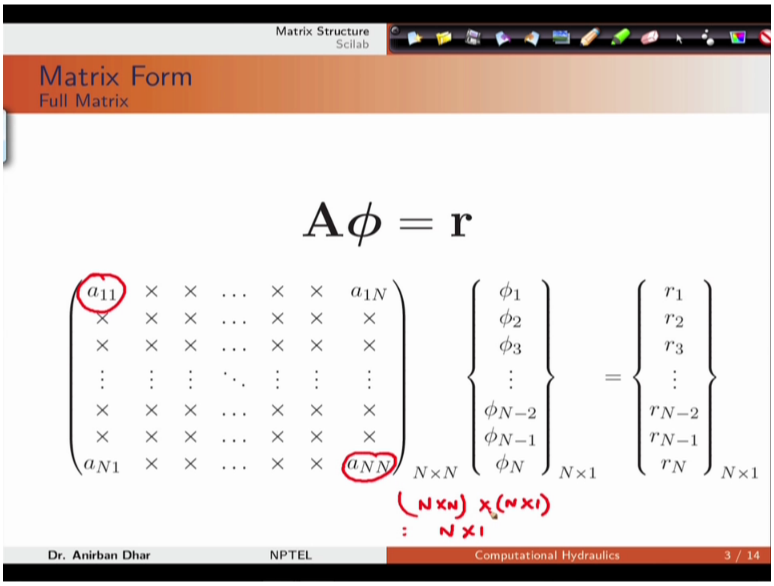
We have utilized finite difference finite volume and mesh free methods for discretization of our partial differential equation or ordinary differential equation. Now ultimate structure for that one is A phi equals to r where A is some matrix which is of constant coefficient and phi is the variable vector and r is the right hand side vector. Now in this case we can solve the thing directly. However if A or coefficients present in the A as function of phi then the whole equation or this form, this becomes nonlinear in nature.

(Refer Slide Time 02:30)



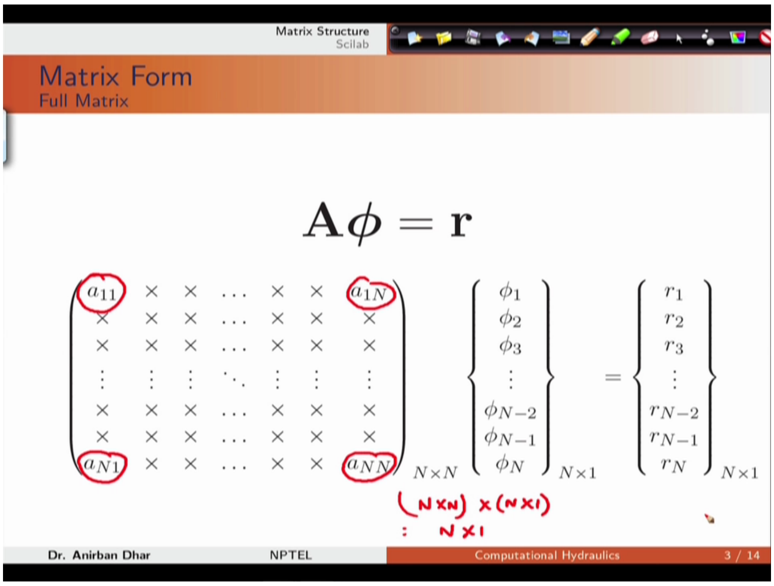
Now we have a solution process available for linear equations and nonlinear equations. Now linear or nonlinear equations if we consider we need to follow different approaches for these two forms. So if we explicitly write it in terms of matrix, A 11 is the first entry in the A matrix and A NN is the last entry. Obviously A matrix is the square matrix. So compatibility should be there in terms of multiplication. That means N into 1, obviously output will be N into 1 structure.

(Refer Slide Time 03:27)



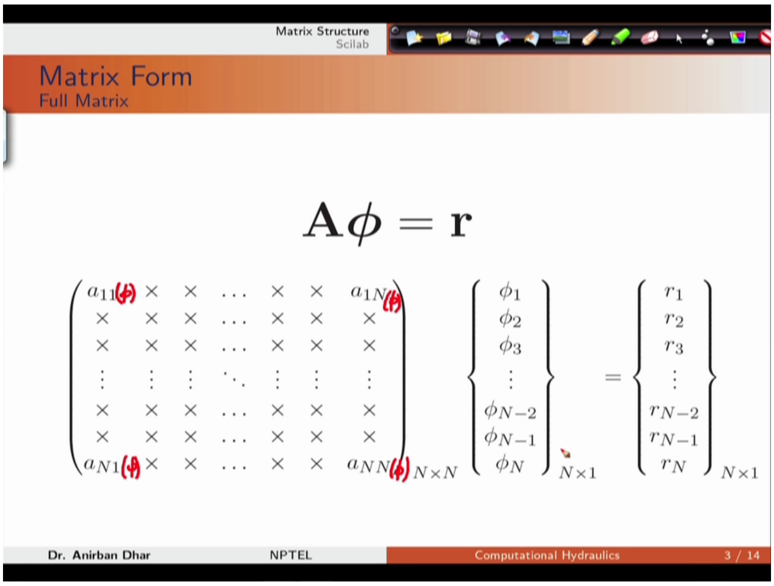
So in this case we can see that A 11 or A 1N, A N1 or A NN, these are represented as constant coefficients. These are not functions of phi.

(Refer Slide Time 03:56)



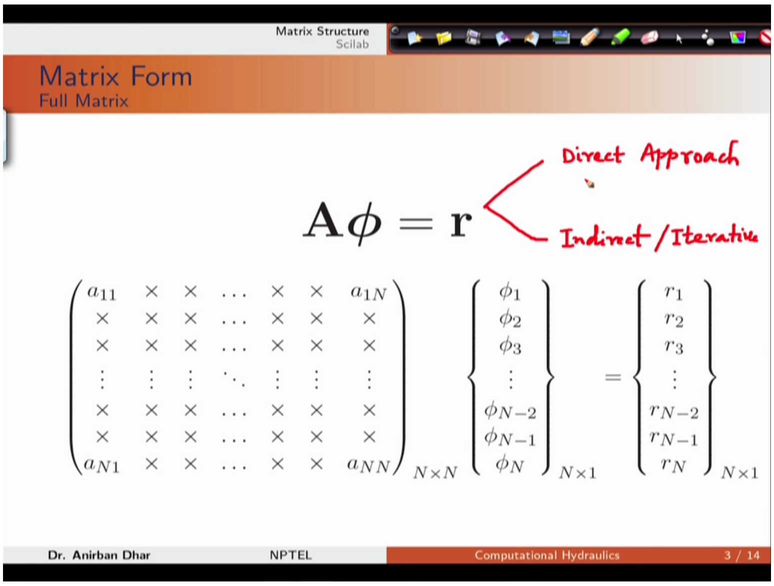
Now if we have this consideration that A is a function of phi, again A 1N is a function of phi. A phi, this is also the function of phi. So the whole thing becomes nonlinear in nature because we will be multiplying again phi1 phi2 phi3 and phi N minus 2 phi N minus 1 and phi N, with these coefficients.

(Refer Slide Time 04:34)



So the equation set will be nonlinear in nature. So for linear equation we have two distinct approaches available. One is called direct approach and another one is called indirect approach. Indirect approach is also known as iterative method. Now under direct approach we can use Gauss elimination or LUD composition and under this indirect or iterative approach we can consider Jacobi's method or Gauss with successive over relaxation.

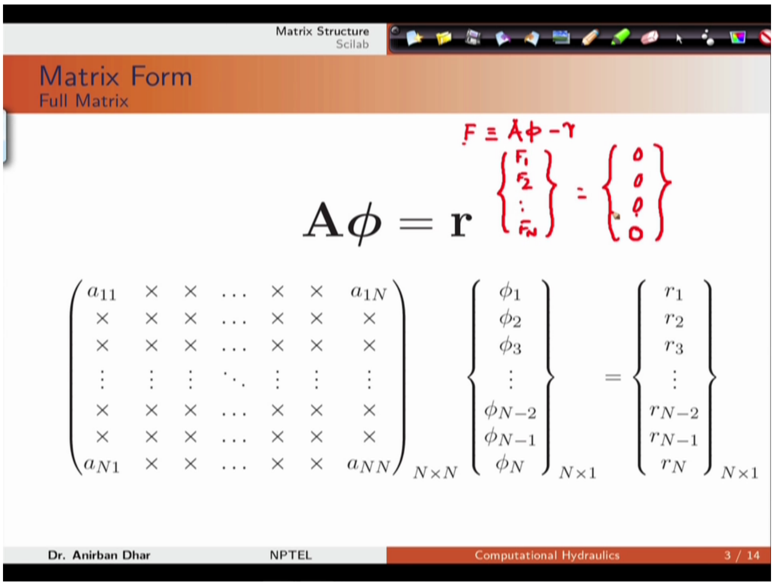
(Refer Slide Time 05:49)



However for solving nonlinear equations we should use a different approach. In that one we can transfer the whole thing in the left inside and we can define sum F. So if we consider it in factor form, there will be F1, F2 to FN, number of terms in this F. So obviously this equation or expression should be zero. And interesting in this case these equations are nonlinear in nature.

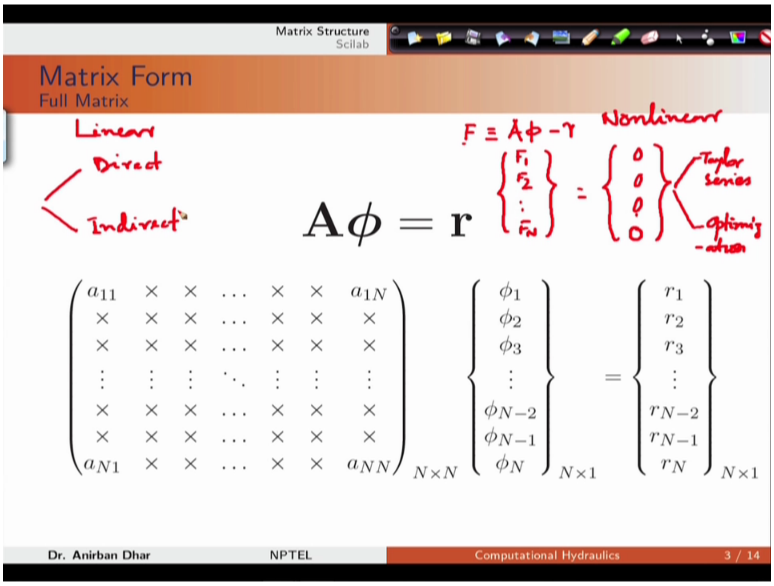
One thing should be clear that is, if one of these equations either F1, F2 or FN is nonlinear and others are linear in nature, then we have to use approach for nonlinear equations. That means the presence of one nonlinear equation in this system that will convert the whole system into nonlinear system and we should follow the approach for nonlinear equations.

(Refer Slide Time 07:40)



Now in this case again we have two approaches. Then first one is based on Taylor series another one is based on optimization approach. One is based on Taylor series, another one is based on optimization method. Now this is a summary of the approaches. So we can say that inside nonlinear, Taylor series or optimization based approach. And for linear we have direct approach and indirect approach.

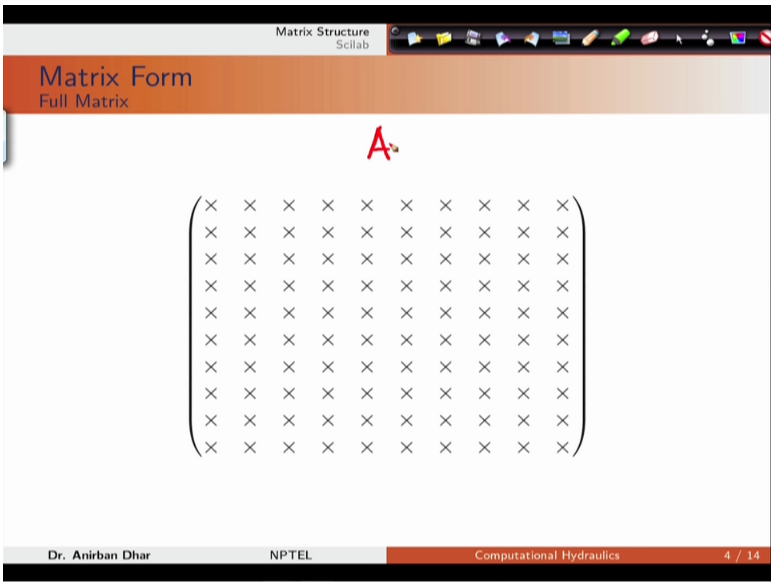
(Refer Slide Time 08:57)



So let us start with the matrix structure. Because we have seen in our finite difference approximation or finite volume approximation that we need to consider only the points or nodes or cells available in the neighborhood. So whenever we are discretizing a particular governing equation basically we are considering the points available in the neighborhood of the point under consideration or cell under consideration or node under consideration.

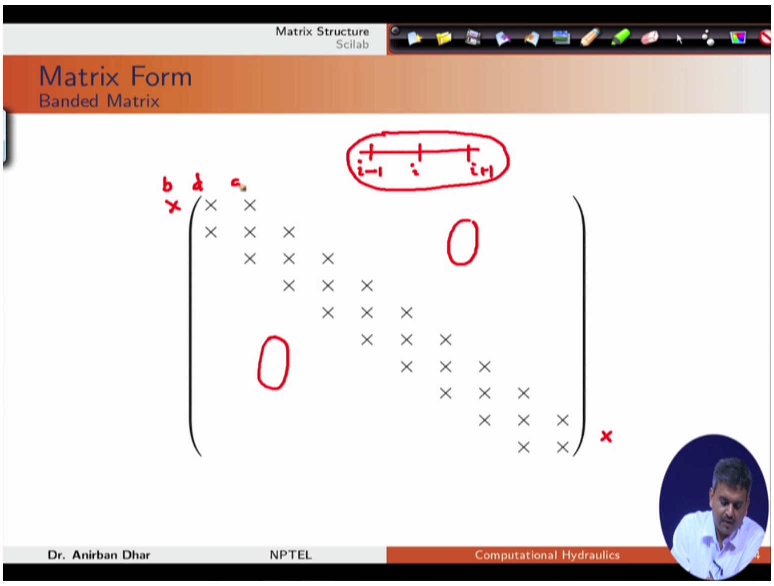
So if we consider the case where the support domain size is equal to the size of the domain, then we should consider all the points. Then the structure of this matrix A, this will be having all the entries or all the non zero entries.

(Refer Slide Time 10:22)



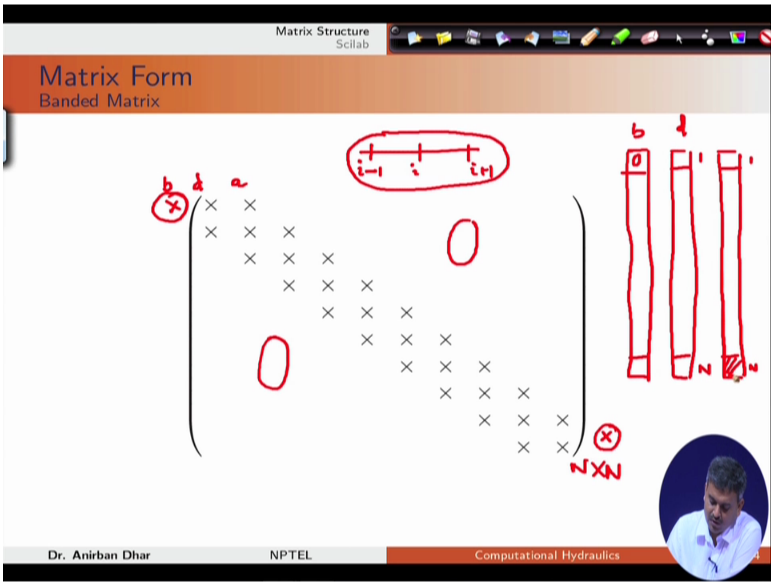
If we have a case where we are having only i plus 1, i, i minus 1, then we have these three nodes available for the support domain. So we can say that we have only three entries available for the matrix. And other entries shall be zero. So we can store it in single column vector like structure. So let us say that if we have three structures. One is for diagonal, this is d, this is below and this is above.

(Refer Slide Time 11:32)



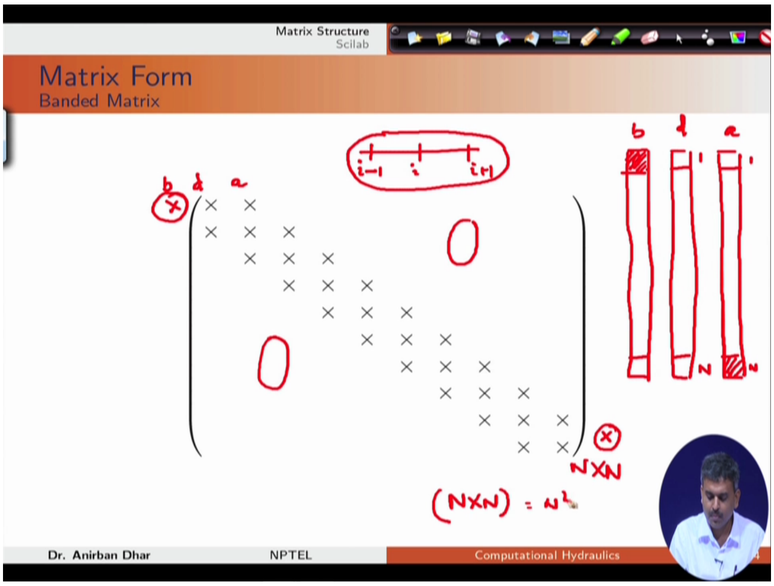
So for diagonal we will have N entries starting from 1 to N. Again for below, the diagonal we will have N entries but this first entry is not there. So either we can put zero or any value. But ultimately we will not be using that value during calculation process. So this is for b. And if we have another column vector like structure where we have entries up to N, starting from 1 to N. But in this case this Nth entry, if we are considering N by N, the size of the matrix, then Nth entry, this is not available.

(Refer Slide Time 12:47)



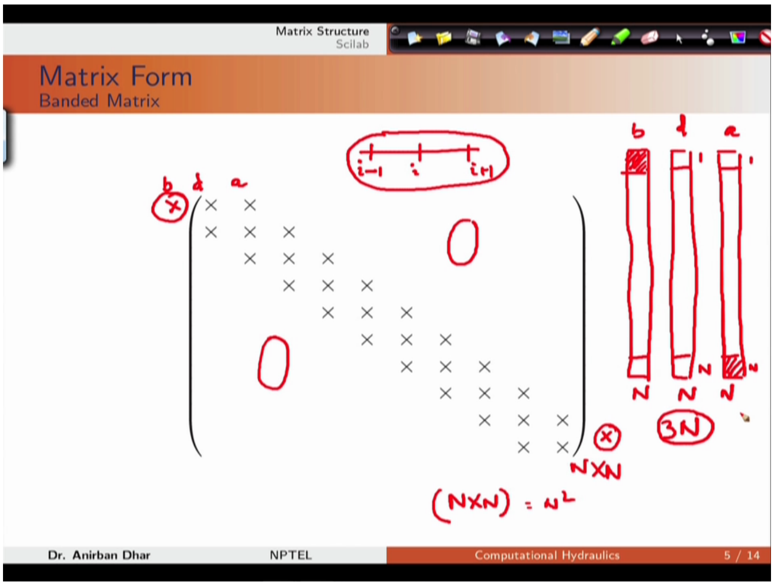
In this case the first one, for above case we do not have the Nth entry available. So we can put any value there. But during calculation process we will not be using that value, okay. For uniformity in the structure definition we can store the matrix information using only three column vectors. So if you have to store it in full matrix format, obviously we need these many entries N into N or N square.

(Refer Slide Time 13:31)



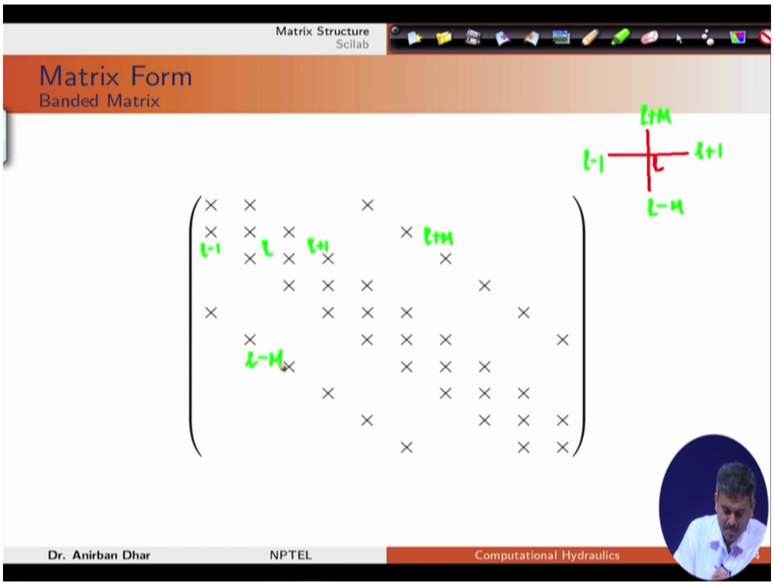
But in this case we have N, N, N. So with 3N number of entries we can represent the same information or we can supply the same information to the (calcu) calculation process.

(Refer Slide Time 13:56)



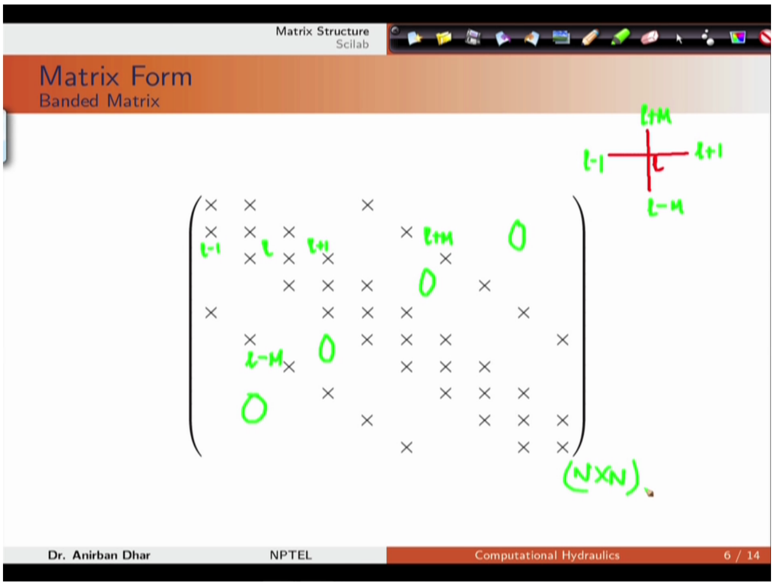
Now in case (penta) if we have penta-diagonal structure, in case of our two dimensional algorithm we have seen that this this is Lth point and we have L minus N number of entries there, then L plus M, this is L plus 1, this is L minus 1. So we can say that this is L minus 1. This is for L and this is for L plus 1 and for further our L plus M or L minus M entries, this is for L plus M, this is for L minus M.

(Refer Slide Time 15:12)



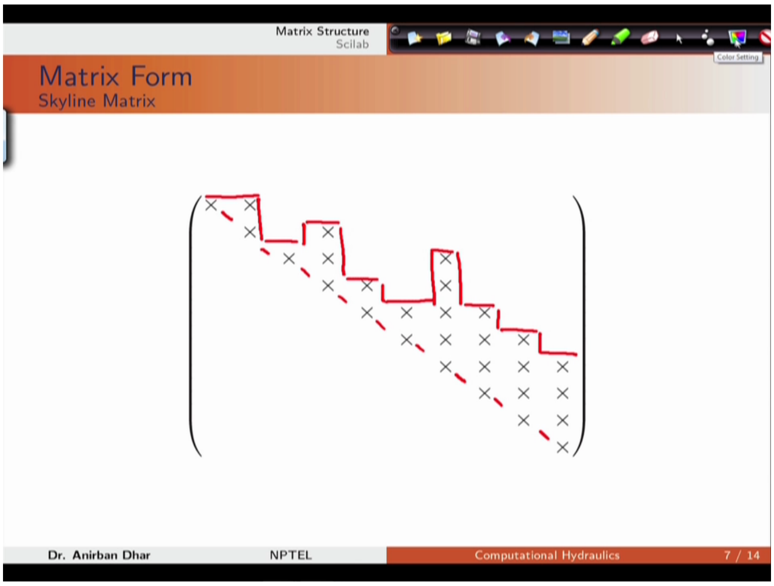
So we can see that without storing this zero values in between we can represent it in terms of column vectors and finally we can store these values efficiently during our calculation process. This is again banded matrix structure.

(Refer Slide Time 15:42)



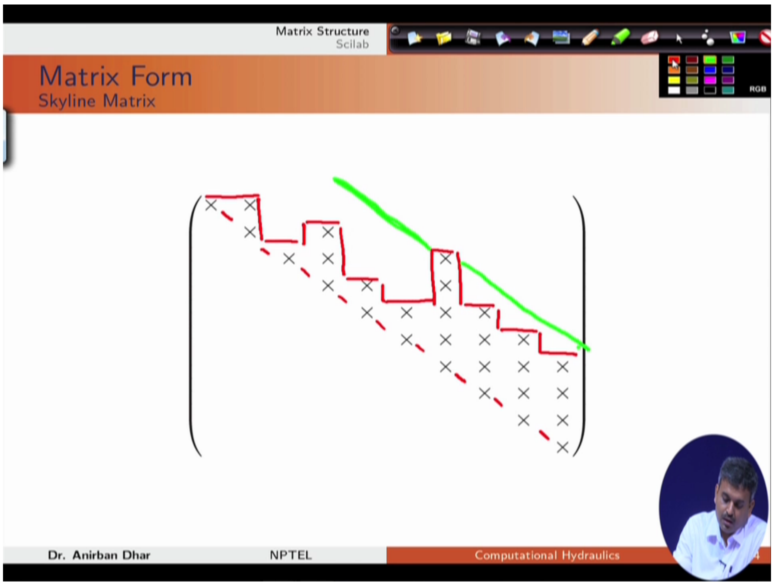
In our previous case we have seen tridiagonal, this is having penta-diagonal structure. 1, 2, 3, 4 and 5. So we have penta-diagonal structure available for this case. Now another kind of skyline matrix can be there. If we consider these entries then we have this kind of skyline structures available for our calculation. And interesting part is that if the points are symmetric in nature or information that is available for the lower portion of this diagonal. If that is symmetric in nature we can represent our information with this skyline structure only.

(Refer Slide Time 17:10)



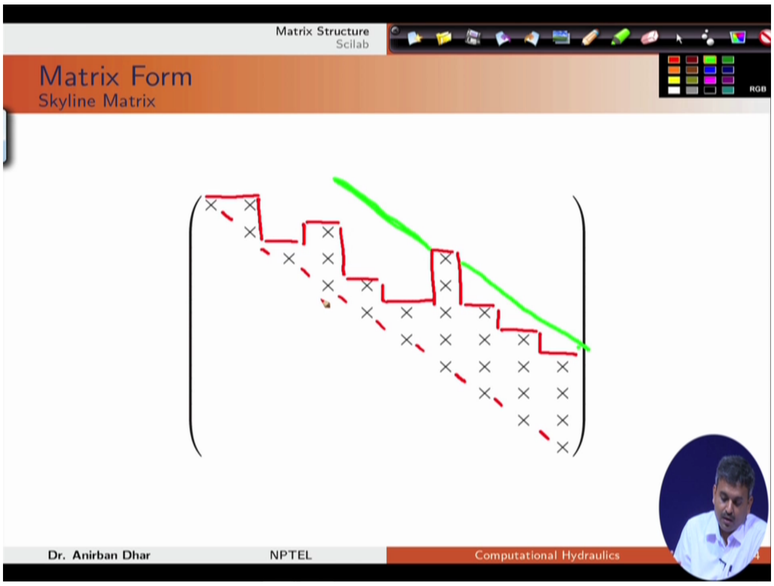
Now in this case interesting part is that if we consider our banded structure, then we should consider this point for banded matrix structure because in between zero entries will be there. But still you need to store this banded matrix for storage purpose.

(Refer Slide Time 17:39)



But we can also store it in the form of like vertical skyline kind of structure. So individually starting from diagonal we can store it in a single column vector. Or depending on the requirement of the algorithm we can store it in different small column vectors.

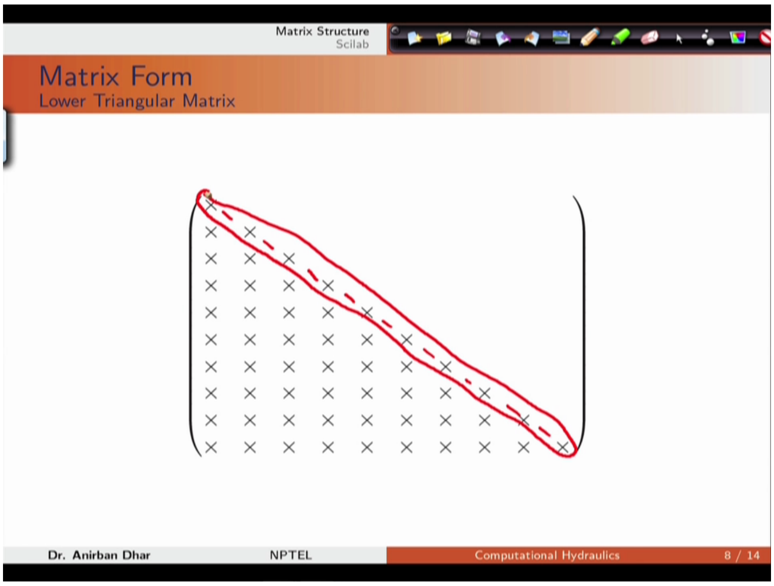
(Refer Slide Time 18:08)



So this is known for all structure of the matrix. Now special kind of structures we need to discuss because these will be there during our LU decomposition process. So lower triangular matrix. So in this case lower triangular matrix means including the diagonal term, if we have only entries available for the lower portion of the triangle. That is our lower triangular matrix.

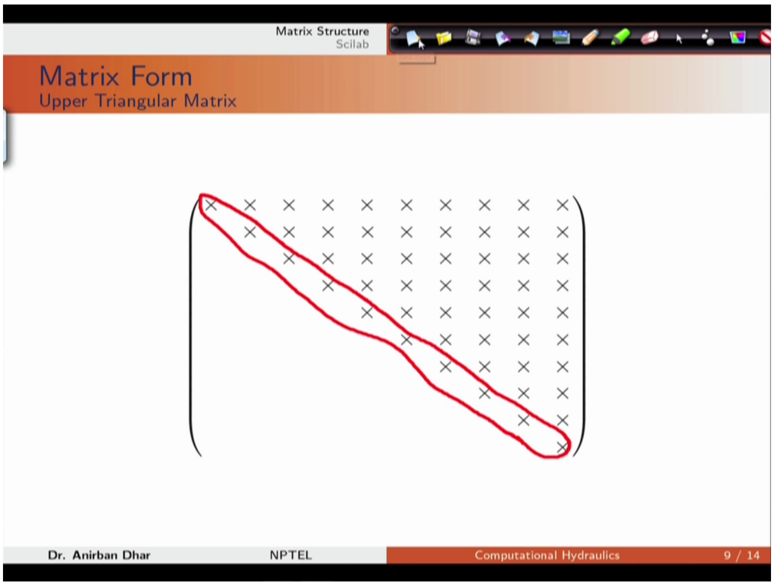
If let us say that this matrix is strictly a lower triangular, then we should exclude the entries in the diagonal term. So only the entries in the lower portion should be considered for strictly lower triangular matrix.

(Refer Slide Time 19:27)



Similarly if we consider upper triangular matrix then we should consider all the entries including the diagonal term. But if we need to consider strictly upper triangular matrix then we should exclude this diagonal term from the representation or our calculation process. So we will be representing only the strictly upper triangular matrix.

(Refer Slide Time 20:10)



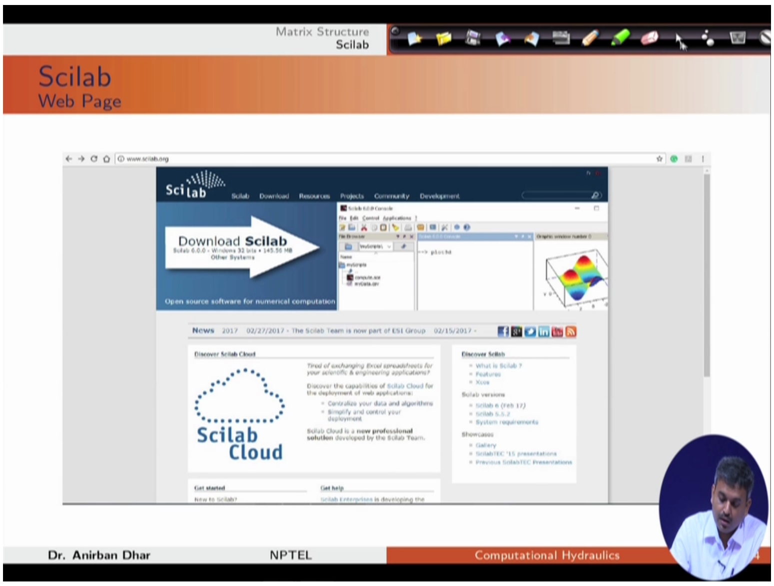
Now we have seen different kind of structures for the matrixes and these matrixes will be will be generated out of the discretization process. Whether we use finite difference, finite volume or mesh free method, we will get finally the matrix structure either in linear or nonlinear format, either it can be banded or full matrix structure will be available for any particular problem. Now we need to represent it or we need to calculate the values.

Ultimate objective of this particular course is that at the end of this course you should be or students will be able to write small quotes so that they can solve the practical problems. So what we can do as part of this course, we can write small problems and we can utilize the discretized form of the equation to get our matrix form and finally we can get the solution and we can represent it using some plotting tool or plotting software.

(Refer Slide Time 22:21)

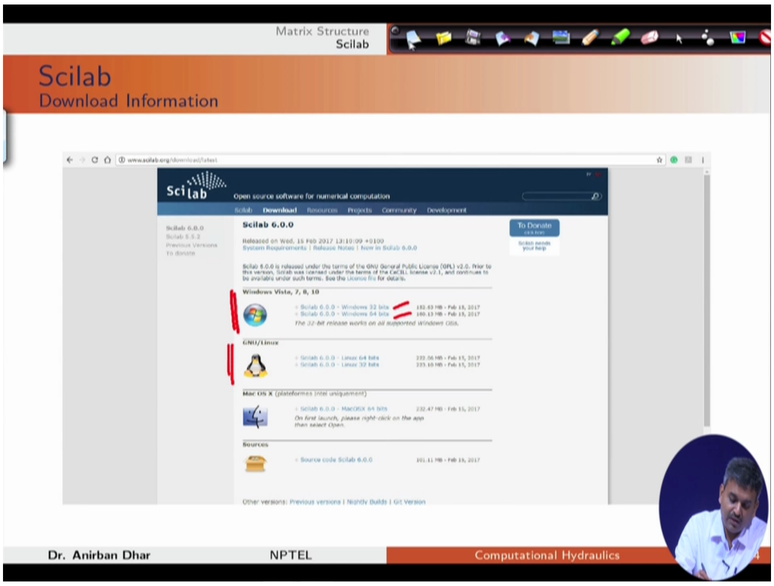
So all of this we can do in this scilab which is very much similar to matlab. But it is open source software. Now this scilab, we can directly download from web address scilab, www.scilab.org. Now from this download button we can directly go to the download portion.

(Refer Slide Time 22:47)



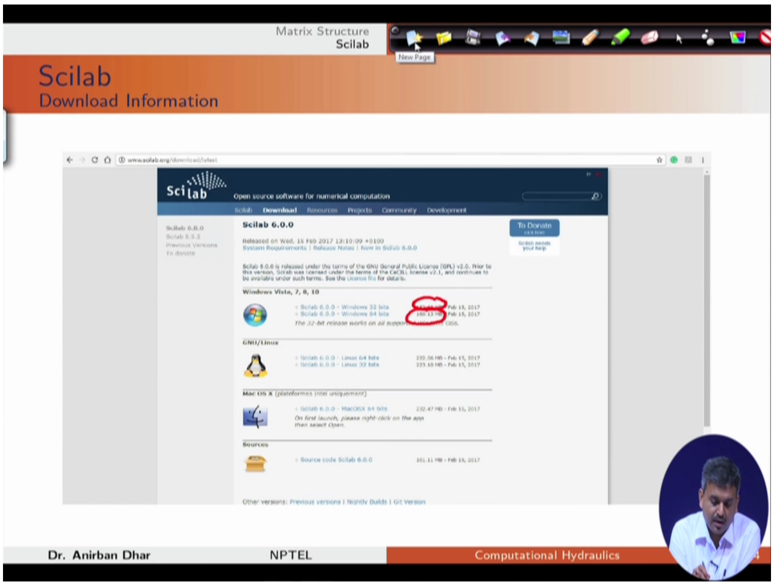
And from the download (pa) page you can see that either you can download it for Windows or Linux versions. Similarly depending on your machine configuration either 32 bit or 64 bit, you can download the software.

(Refer Slide Time 23:15)



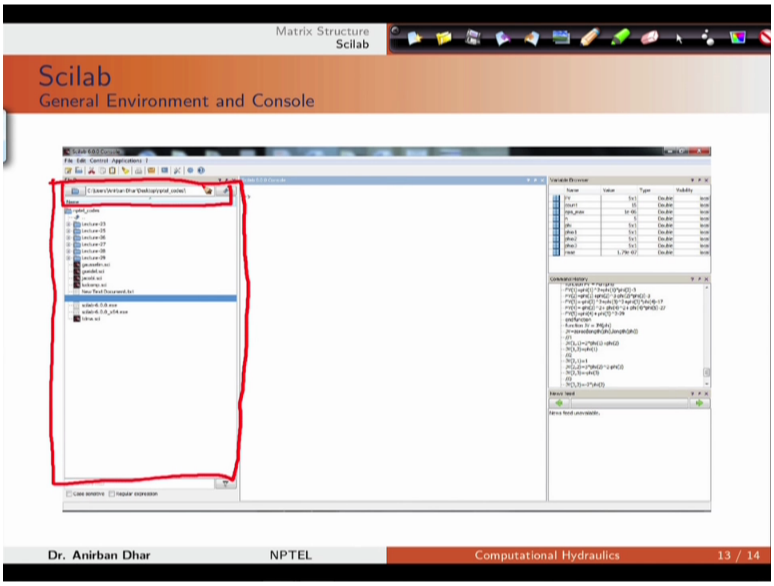
The advantage of using this particular software is that you need to download only 153 MB or 160 MB for 64 bit machine. So you don't need much space for your computer to run this particular program.

(Refer Slide Time 23:39)



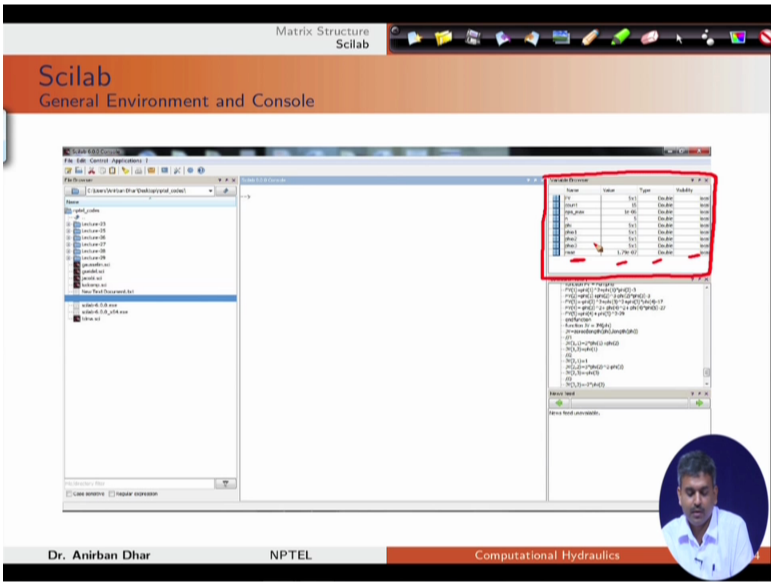
And at least you can test your concepts that you have learnt from this particular course and you can directly translate those concepts into small codes using this scilab platform. Now if we open the scilab general environment and console, on the left hand side you will find file browser. You will get all information regarding the file and at this place you will get information regarding directory. Now directory information is required.

(Refer Slide Time 24:29)



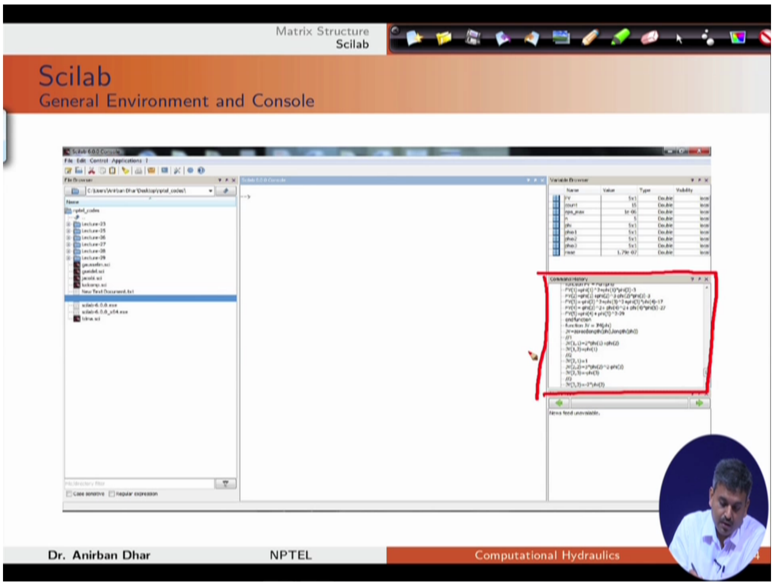
Now on the right hand side we have variable browser. Variable browser means that whatever variable you are using for a particular program or calculation its value and type and visibility, whether it is local or global, that will be available in this variable browser. And the first column you will have the name of a particular variable.

(Refer Slide Time 25:16)



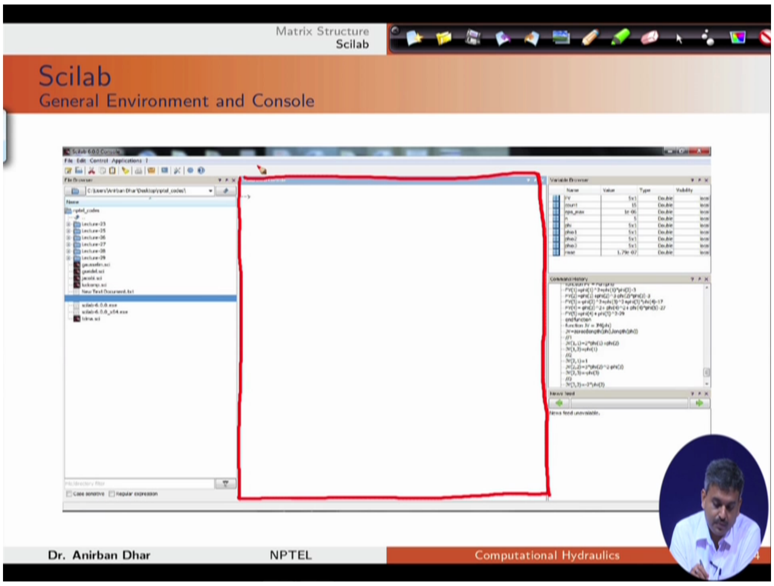
Below you will have command history, whatever command you have utilized you can directly get that from the command history.

(Refer Slide Time 25:33)



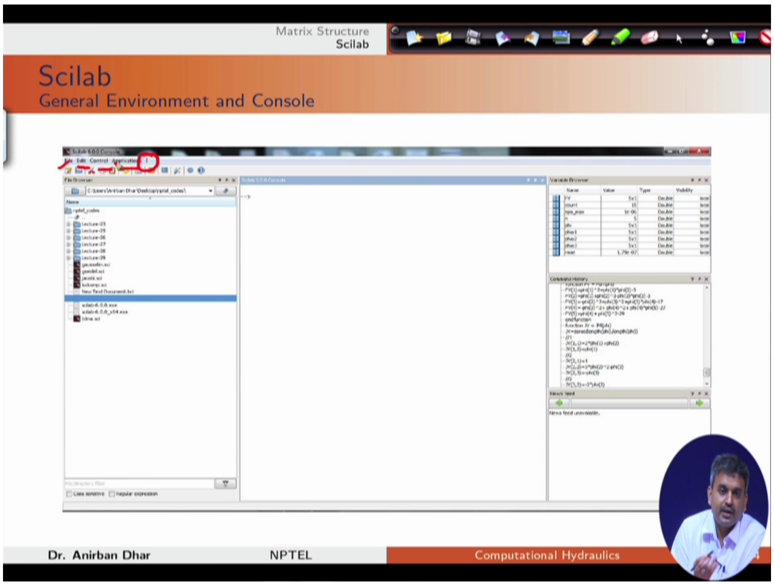
And most important part is the central portion where we have console. And this console part is the important portion because we perform all calculation in this console portion.

(Refer Slide Time 25:52)



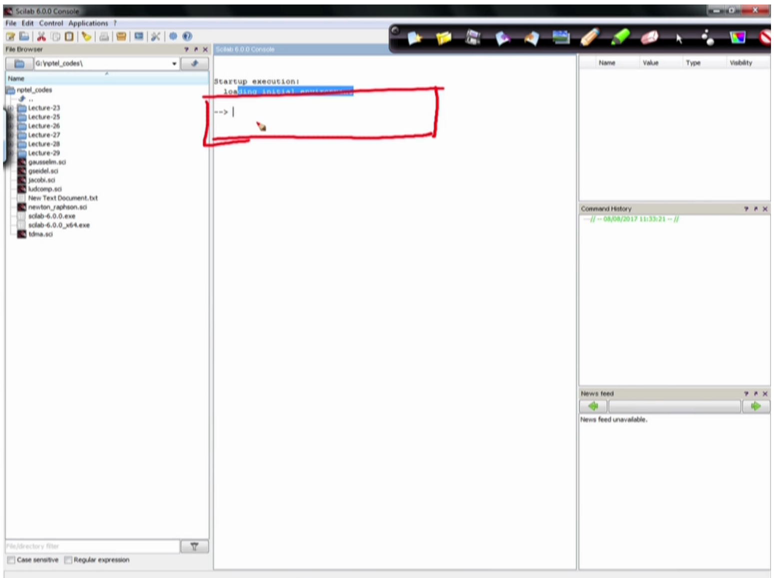
On top you will have file, edit, control, application and this question mark is the help. You can click on this question mark button and get all command related information from this help button.

(Refer Slide Time 26:13)



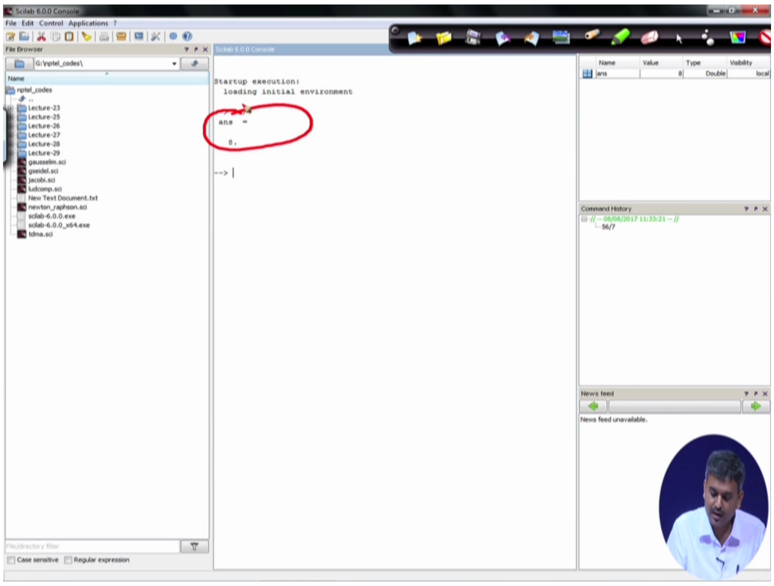
Now if we open this one, so let us see that this is our console. This is our console environment. So we can write anything in this console for calculation.

(Refer Slide Time 27:01)



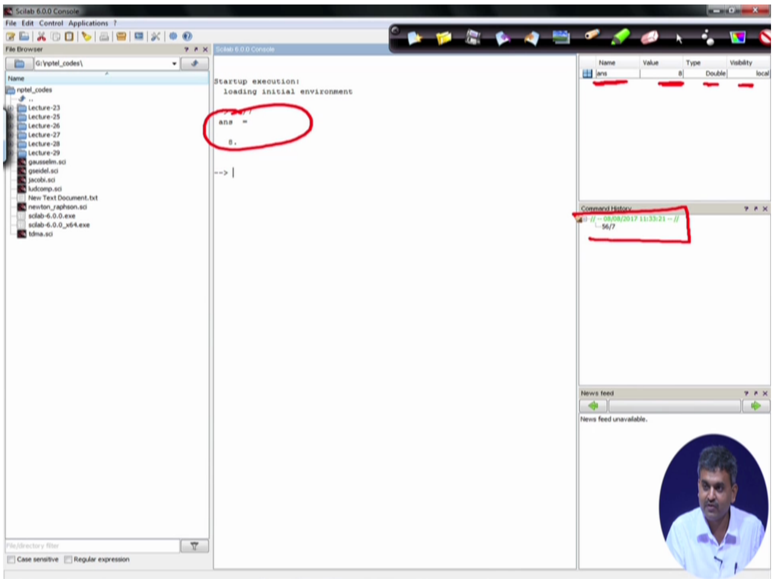
Let us say that we want to use this console as calculator. Let us say 56 divided by 7. This should be 8. So we can directly get the answer at this position.

(Refer Slide Time 27:35)



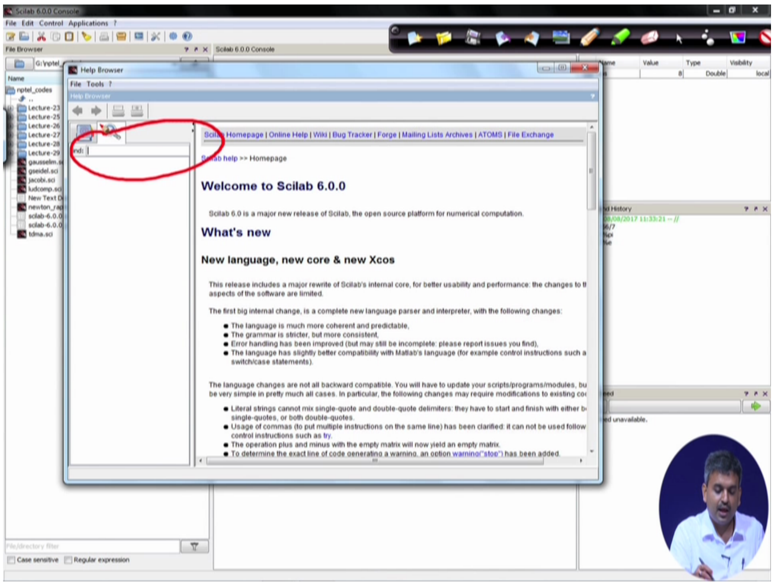
And on the right hand side, output is answer, value is 8 and type is double and visibility is local. At the same time we are getting command history on this right portion.

(Refer Slide Time 27:50)



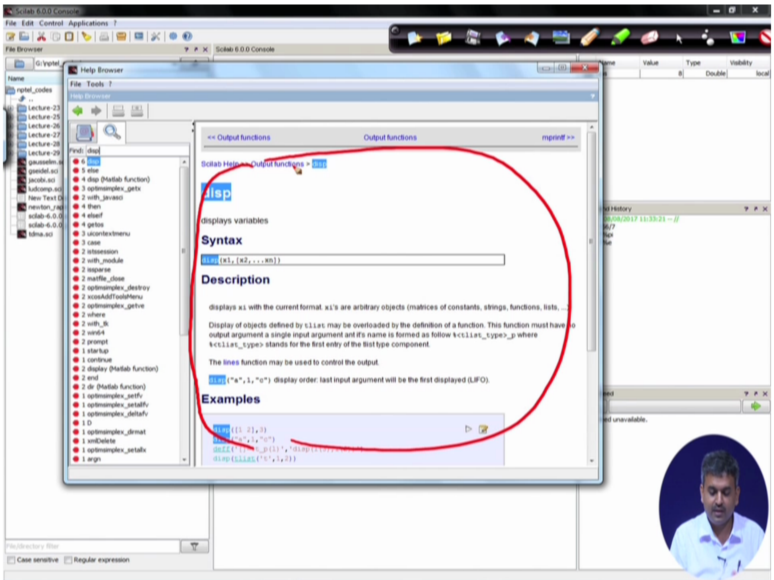
So we can easily utilize this thing for our calculation purpose. Now interesting thing is that we can use percentage pi to get value of pi. Or percentage e to get value of e. Again if we open this scilab environment help then you can search anything here. You can write anything.

(Refer Slide Time 28:47)



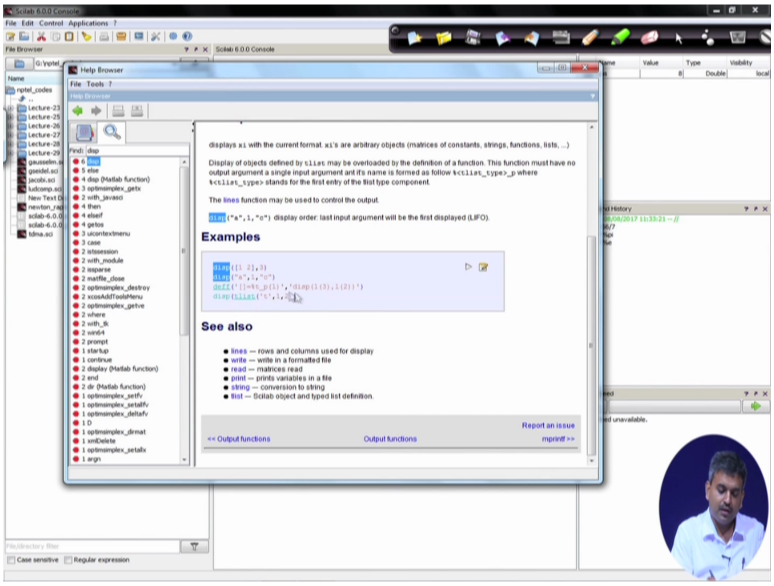
Let us say if I want to write some display command here. So disp, so if I search it, so on the right hand side it will show the output. Output means whatever command related information is there.

(Refer Slide Time 29:12)



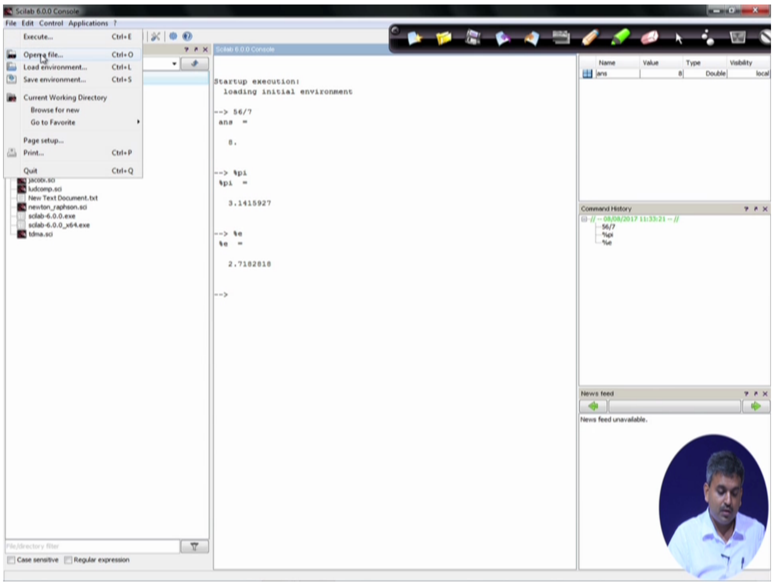
And on the bottom of that help or examples are available. So we can directly show this thing.

(Refer Slide Time 29:25)



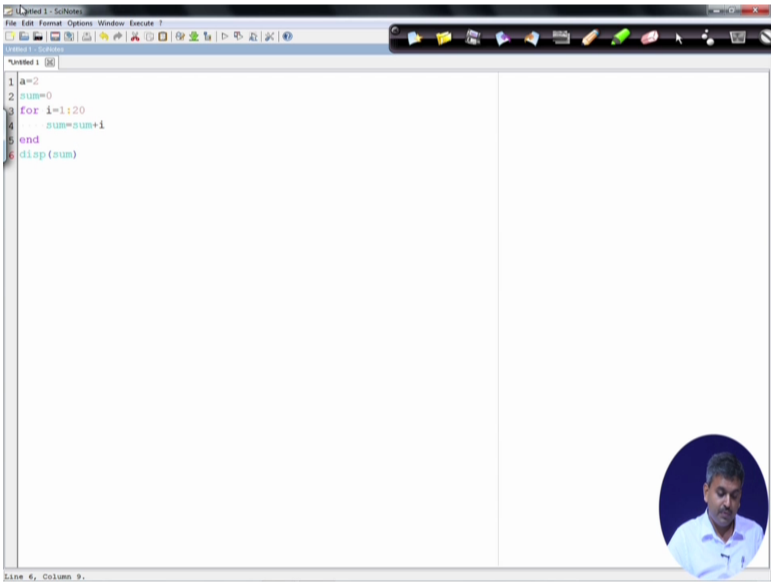
Another information is that we can use this file. So we can use file command. So this is opening a file or opening sci notes. Now what is the advantage of sci notes? Advantage of sci note is that we can store the commands in this particular sci note for future use.

(Refer Slide Time 29:34)



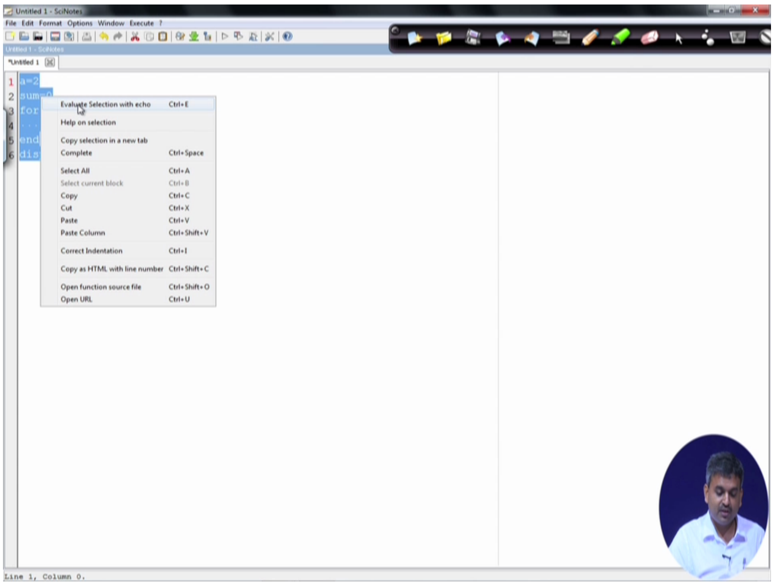
Let us say that if A equals to 2, then we are writing for loop for I equals to 1 to 20. This colon thing is utilized represent the for loop starting from 1 to 20. And in this case let us say that we want to sum it up. So some outside I am writing it as sum equals to zero and inside sum equals to sum plus I. That means I continuously we are adding. And finally let us display this sum.

(Refer Slide Time 31:12)



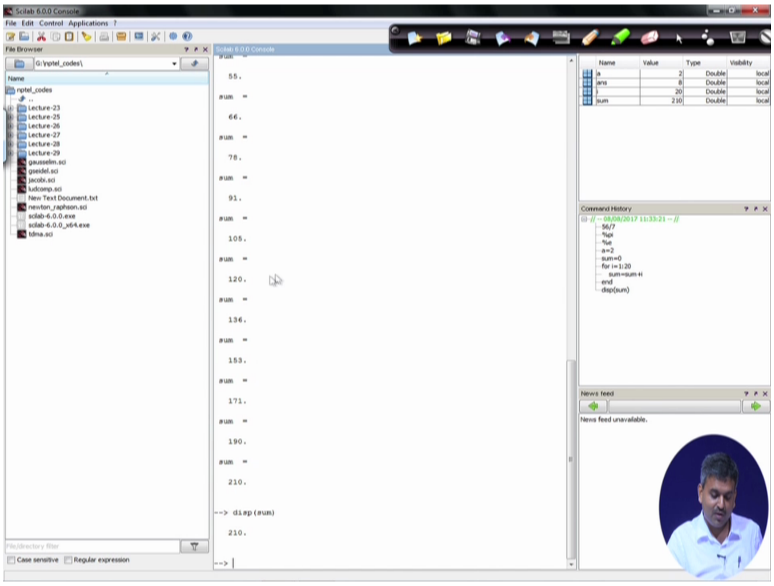
Now what we can do, we can select it, right click, evaluate selection with echo.

(Refer Slide Time 31:18)



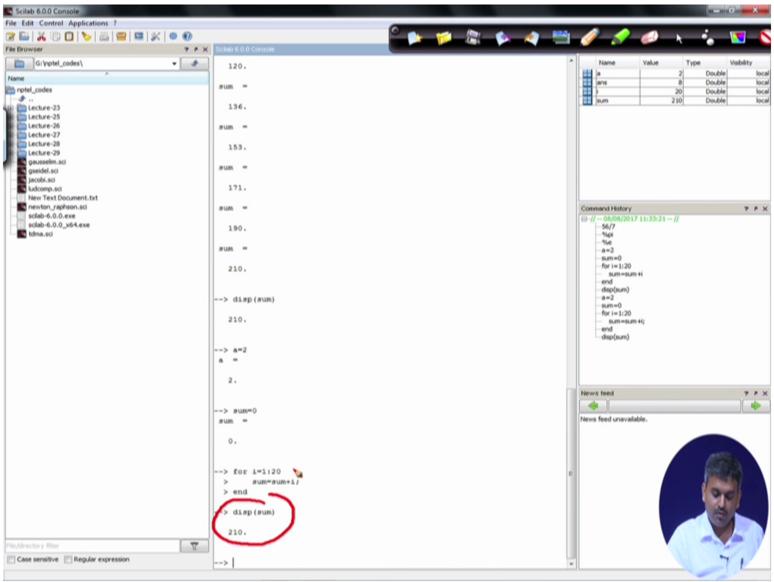
So we can directly get the output here, this is 210.

(Refer Slide Time 31:28)



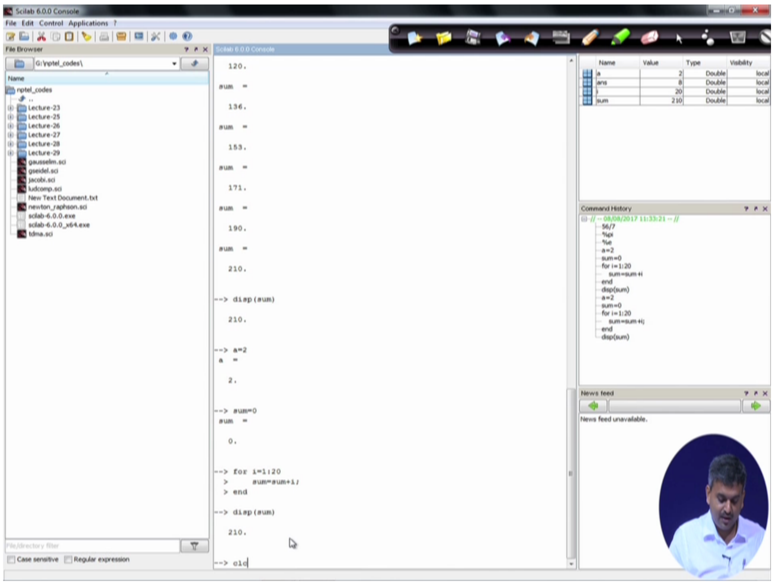
And continuously we are getting update using this one. Now we can see that all the steps and value of the sum at all steps displayed in this process. Now if we use this thing, so we can get only final output.

(Refer Slide Time 32:04)



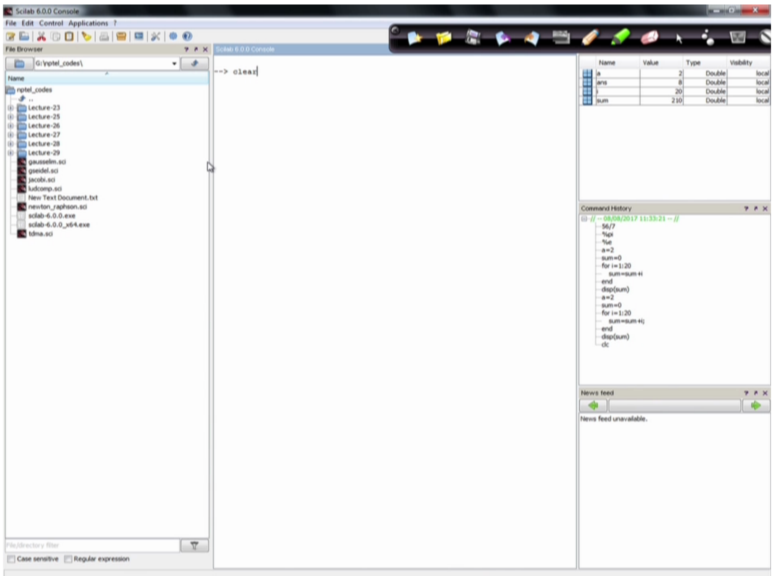
So what is used in this case? In this case we have utilized the semicolon. So semicolon after any line that will stop the display of output in your console window. Now in this case what we are getting is clear. However at the starting itself we should clear the screen. So CLC is used. Otherwise we can directly write in our console window CLC. So that will clear this console.

(Refer Slide Time 32:48)



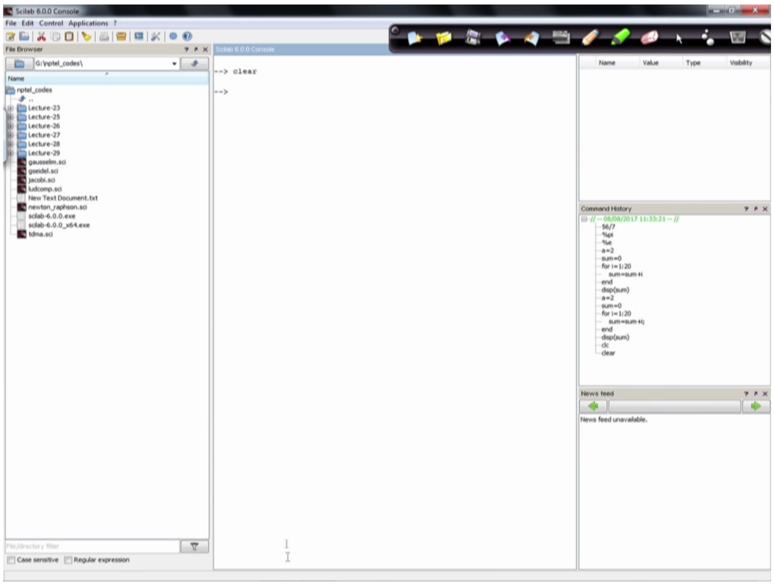
But still variable related information is there on the right hand side. Now to clear this variable related information, we should write clear command.

(Refer Slide Time 33:06)



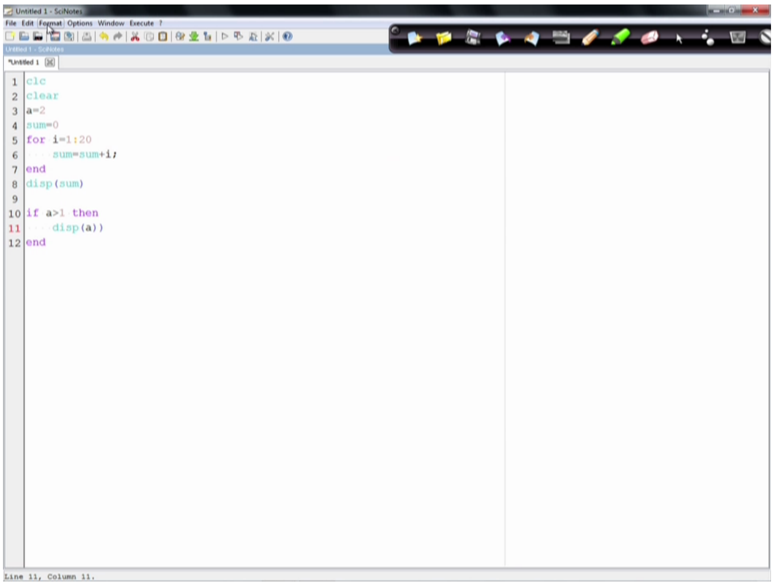
So after using this clear command, I think you can see that this variable related information is not available. That means in memory the variable values are not stored.

(Refer Slide Time 33:23)



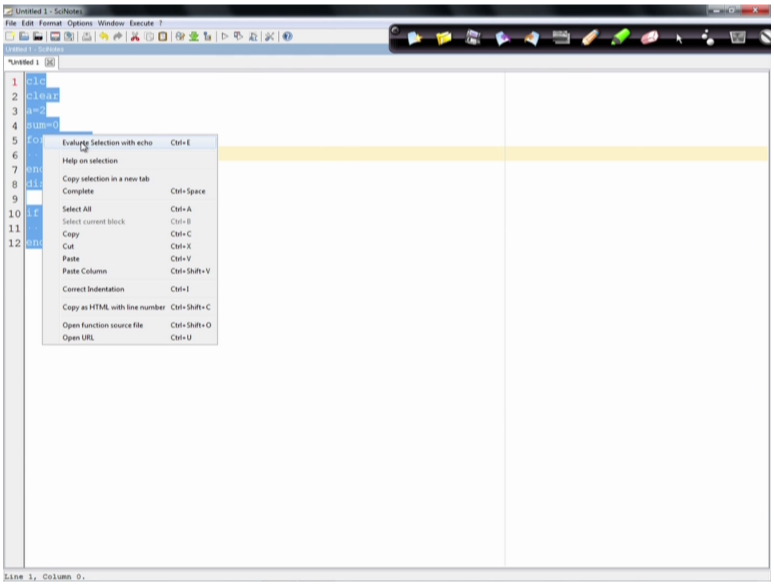
So to start any scilab code or script we can uses two commands that is CLC and clear, to clear the console and clear whatever values are stored in our memory. Now we can also use some while loop. Or if then else condition. If then if A greater than 1, then display A.

(Refer Slide Time 34:18)



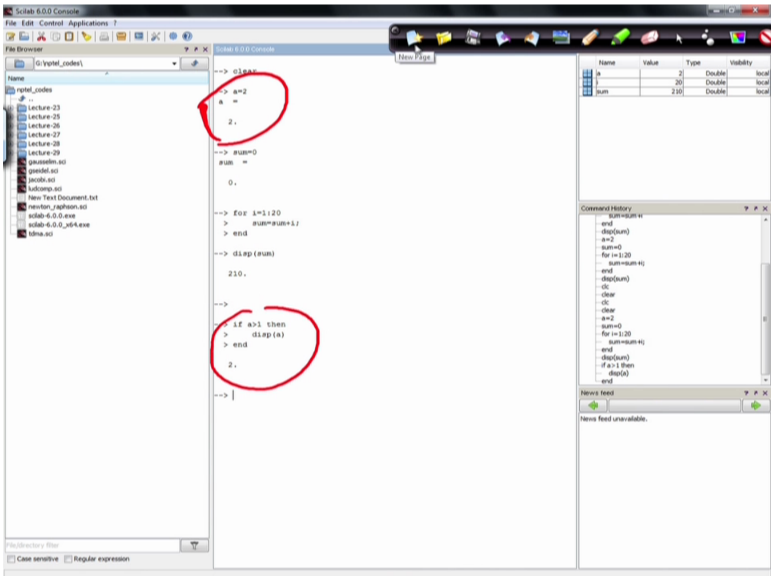
If you want to execute small portion of the code then you can select it and evaluate it. But the A related information is not there so we will select the whole thing because the value of A equals to 2 is written on top. So that should be available otherwise the code will not display any value here.

(Refer Slide Time 34:50)



So we can see that we have got information about the displayed value of the sum. And if A greater than 1 because we have defined A equals to 2. So obviously it will print.

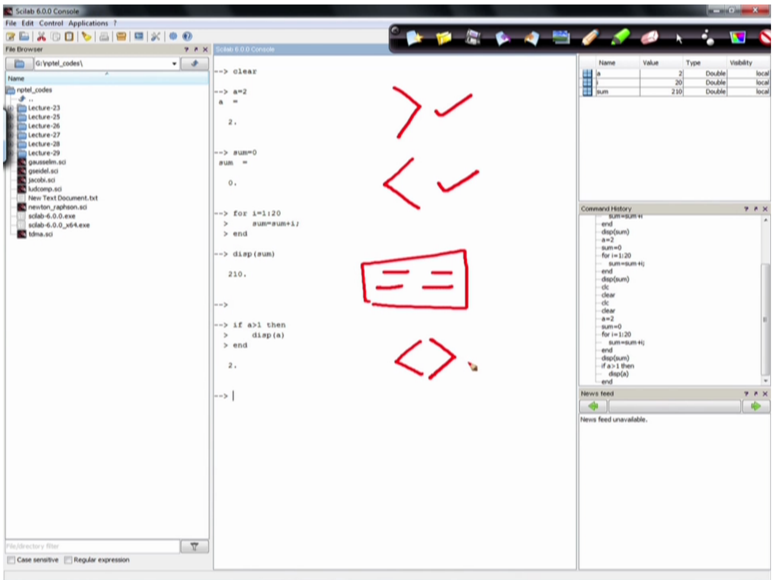
(Refer Slide Time 35:16)



That means things should be cleared that greater than command is applicable here or less than command we can use. And similarly we have other commands available in this case. So those commands are directly utilized for comparing. If you want to compare two values we need to use two equal the little sign. And this compares two values. Obviously it should be clear that we can compare interior values.

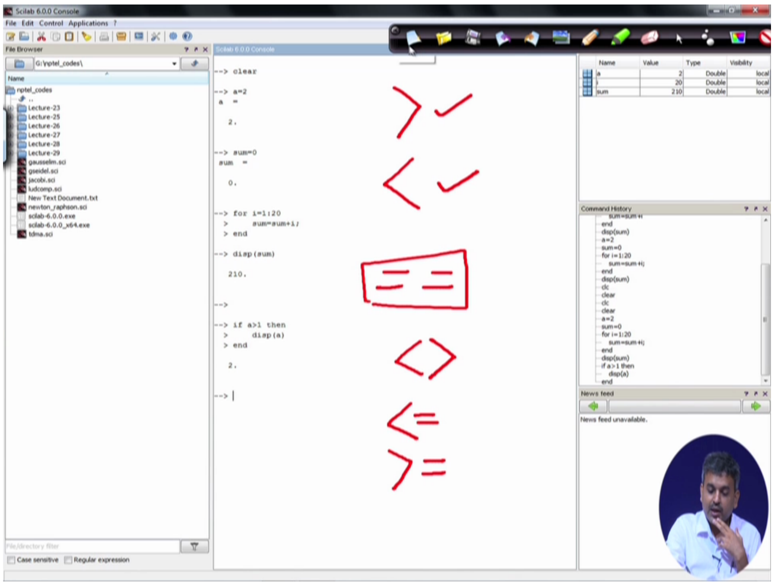
But if you have a double value after decimal place there will be some disturbances. So we cannot get the equality sense in case of double variables. So this is greater than, this is less than, this is for equality and for not equal or different we can use this combined symbol.

(Refer Slide Time 36:43)



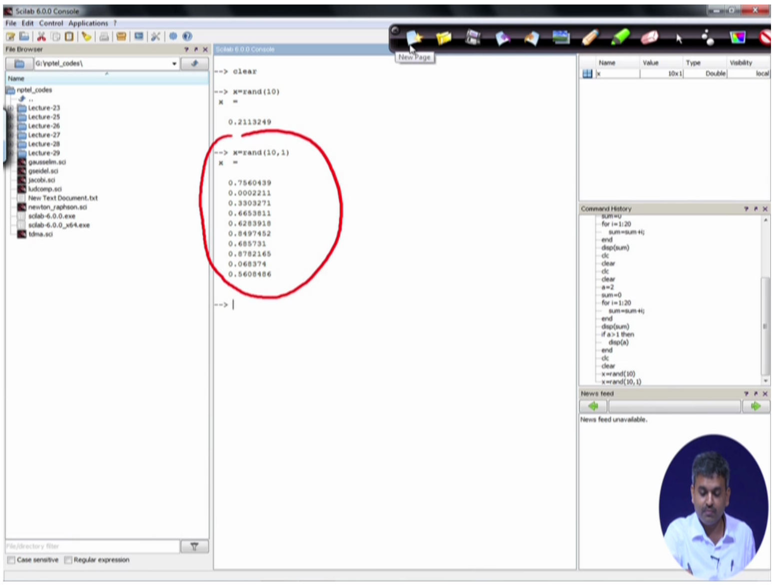
If it is less than equal to or greater than equal to we can use these operators directly for our calculation purpose.

(Refer Slide Time 37:04)



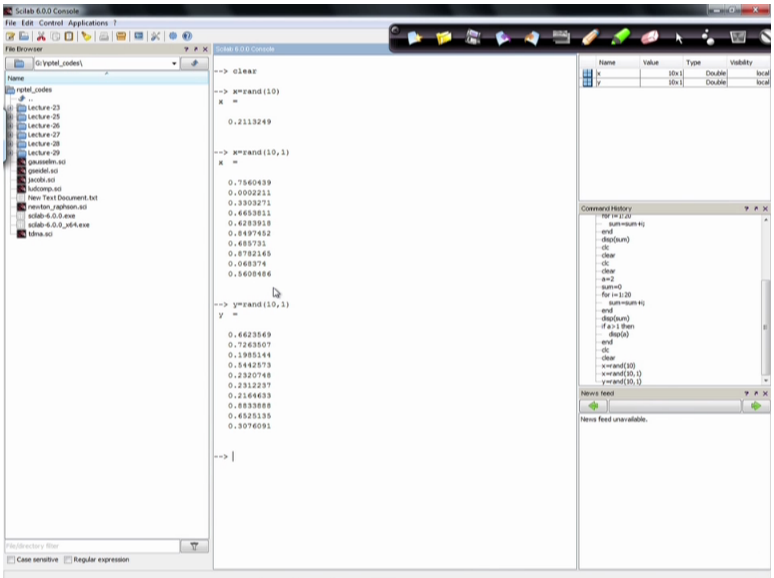
Now this thing is okay because we can write small codes. Again another advantage is there for this scilab that is our plotting thing. Okay. So let us say that x equals to rand. Rand is random. Random 10 values. So we have written or we have used this command x is equal to rand, to generate random values here.

(Refer Slide Time 38:04)



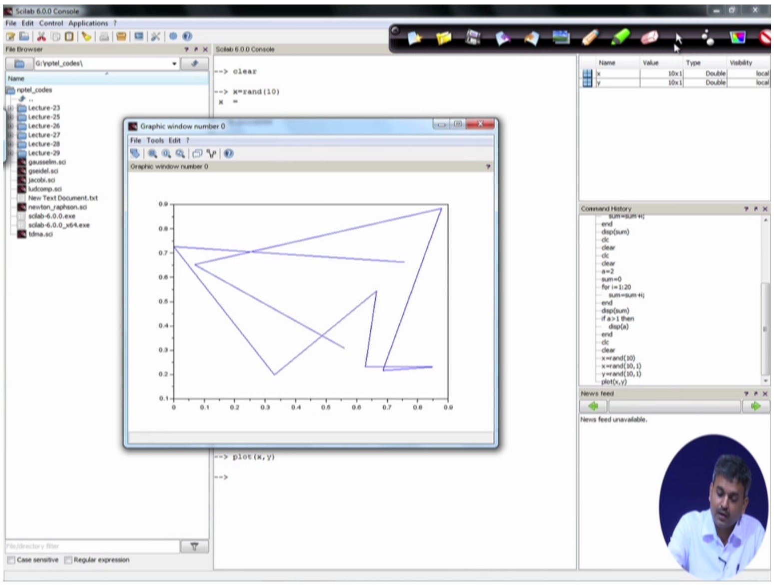
Now we can generate y here with same rand for plotting purpose. Y equals to rand 10 1. 10 1 means 10 rows in this case. So we have x and y values available. So this is for demonstration purpose only.

(Refer Slide Time 38:34)



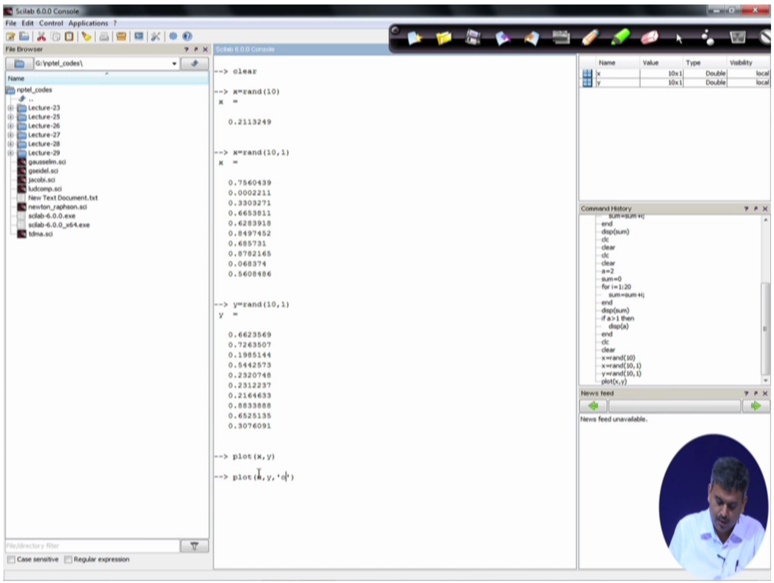
Now we can directly write this plot command, plot and x comma y, this will show you the output. That means this will show the points or this will show the connecting lines of the points. And definitely in this case we have x on this case and y on this axis.

(Refer Slide Time 39:10)



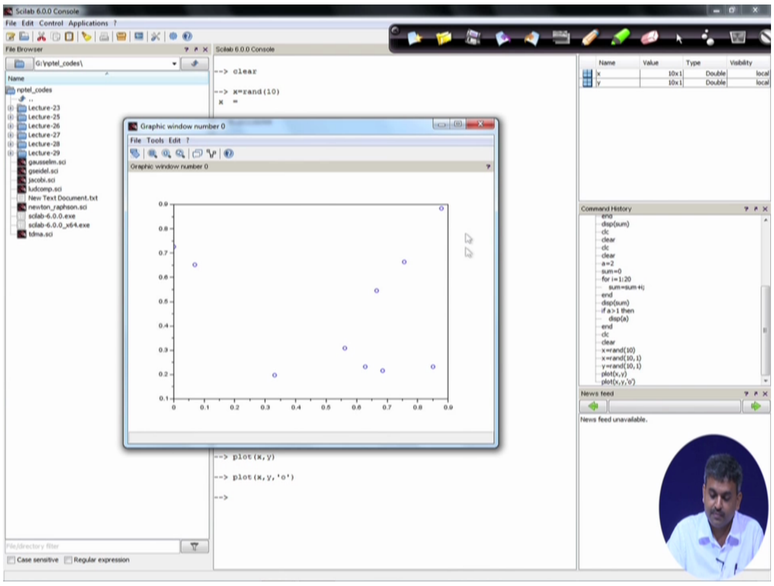
If we close this, if you want to display only the points, so we can use another comma and within code we can use o.

(Refer Slide Time 39:26)



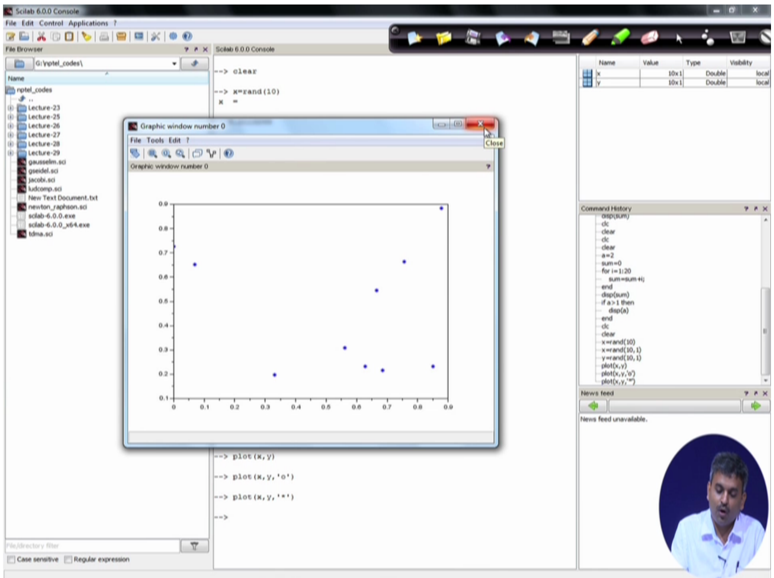
So this will show you only dotted points but lines are not shown in this case.

(Refer Slide Time 39:37)



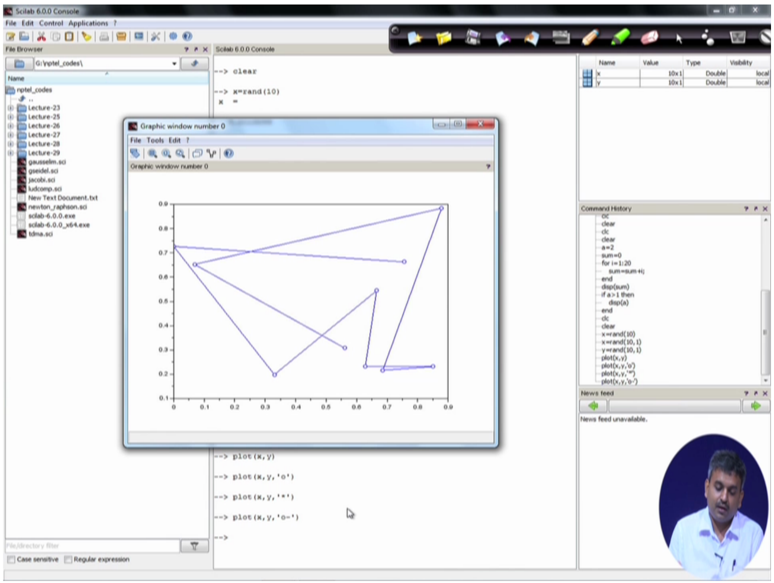
You can use star instead of o. The result will be same. Only points will be displayed in your plot window.

(Refer Slide Time 39:50)



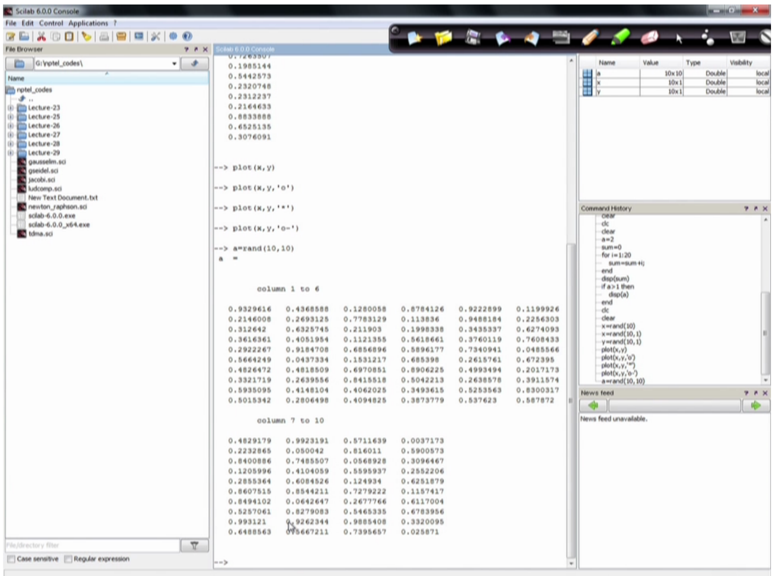
If you want to plot both lines and points then you can use o and dash. That is with o dash we can show both points and line.

(Refer Slide Time 40:20)



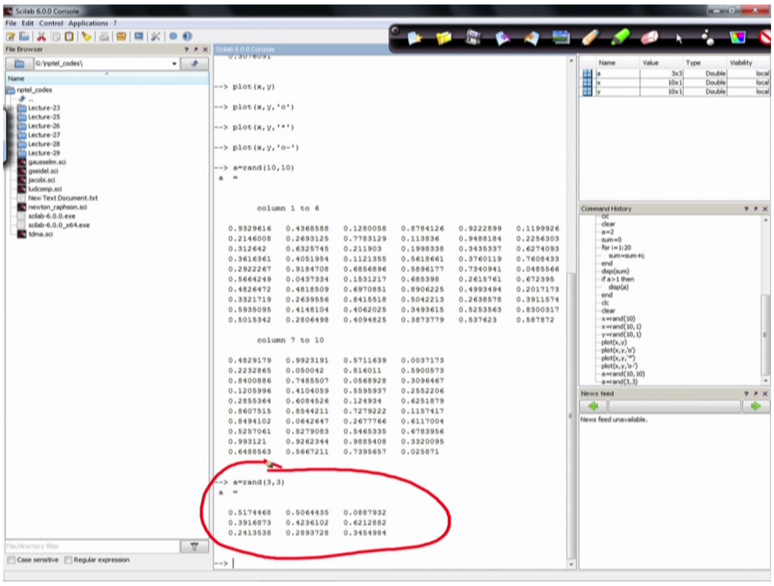
That means within a single software or programming framework we can write our matrix. If you want to write a matrix A equals to random 10 by 10, this will give you 10 by 10 matrix.

(Refer Slide Time 40:44)



Or if it is maybe in simple terms 3 by 3, we can easily see that values can be easily represented using matrix.

(Refer Slide Time 41:02)



And advantage of this scilab thing is that we can write small codes using our sci notes and we can execute those codes to solve our computational hydraulics related problems. So next lecture onwards we will be using this scilab for our coding purpose and we will try to write simple codes to get the solution of the matrix related problems or our discretization related problems. Thank you.