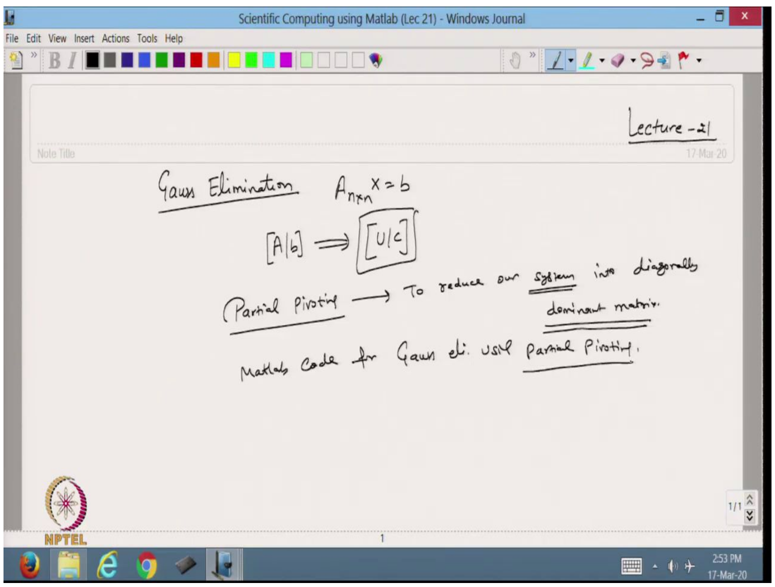
Scientific Computing using Matlab
Professor Vivek Aggarwal
Profesor Mani Mehra
Department of Mathematics
Indian Institute of Technology,
Lecture 21
MATLAB Code for Gauss Elimination Method

Hello viewers, welcome back to the course on Scientific Computing Using MATLAB. So, let us start with lecture 21.

(Refer Slide Time: 00:35)

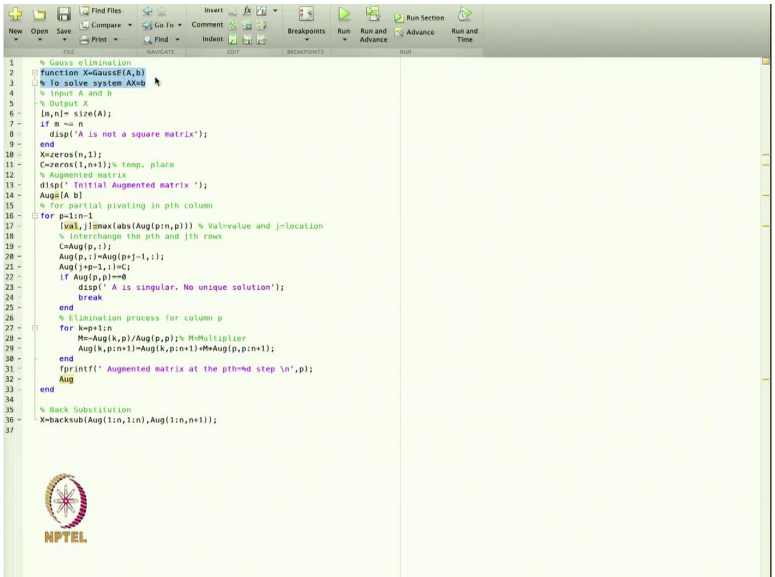


So in the previous lecture we have discussed about the method the direct method, so that was the Gauss elimination and in that one we have discussed that suppose I have a system of a matrix $n \times n$ matrix and this is my system of linear equations and then what we do is that we transform this as an augmented matrix and we transform this matrix into the upper triangular matrix using the Gauss elimination.

And then using the backward substitution will solve this system of equations to find the solution of this equation. So, in this case and then we have discussed that the system should be so after that we have discussed what is that partial pivoting, so partial pivoting we have used to reduce our system into diagonal dominant or system means that the system is deal with the matrix, so we want to reduce the corresponding matrix into the diagonal dominant matrix.

So, once we have a diagonal dominant matrix, then we know that we are always going to have the solution using the Gauss elimination method. So, today after doing this one, let us make the MATLAB code, MATLAB code for Gauss elimination using partial pivoting. So, let us go to the MATLAB code.

(Refer Slide Time: 02:35)

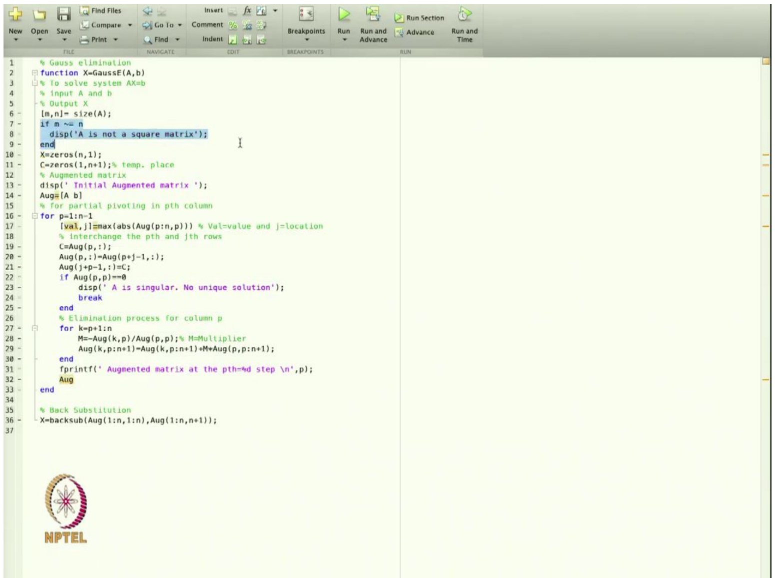


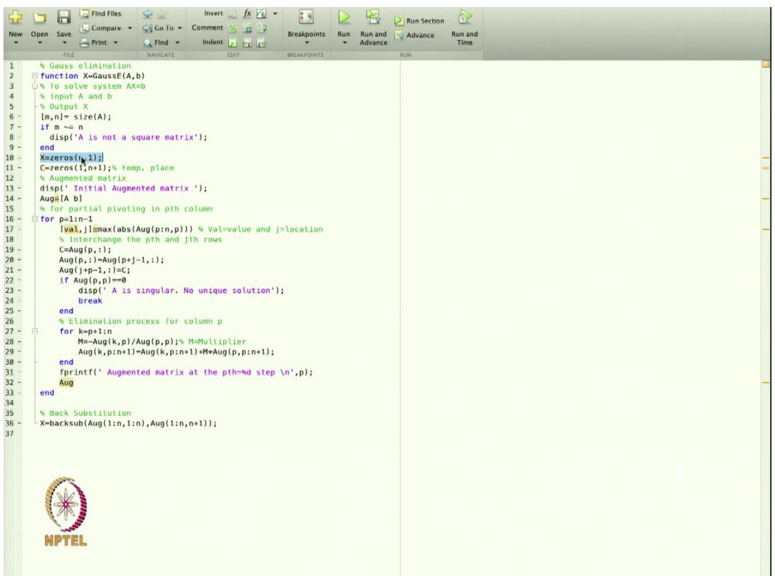
So, this is the program I have made for the Gauss elimination method, so now we already know how to make the functions, so let us start with the, this function that is called the Gauss elimination. So, we have written this I have given the name of this function as a GaussE, GaussE means Gauss elimination, so I know that in this case, I have a system AX= b, so if I have to input the value of A that the matrix and b, then only I will get the value of X.

So, in this case I will input the two arguments, one is A that is in my matrix and b is the right hand vector. So, I will input this one, so they have written that to solve system AX=b, input A and b and output will be X. Now, what do I do? I turn once I give the value of A then I will check the size of A.

So it will give you the number of rows and columns, if in this a say that $m \neq n$, means if the number of rows is not equal to the number of columns then a matrix is not a square matrix and we know that the Gauss elimination method is we are dealing with the n cross and means is a square matrix.

(Refer Slide Time: 03:57)



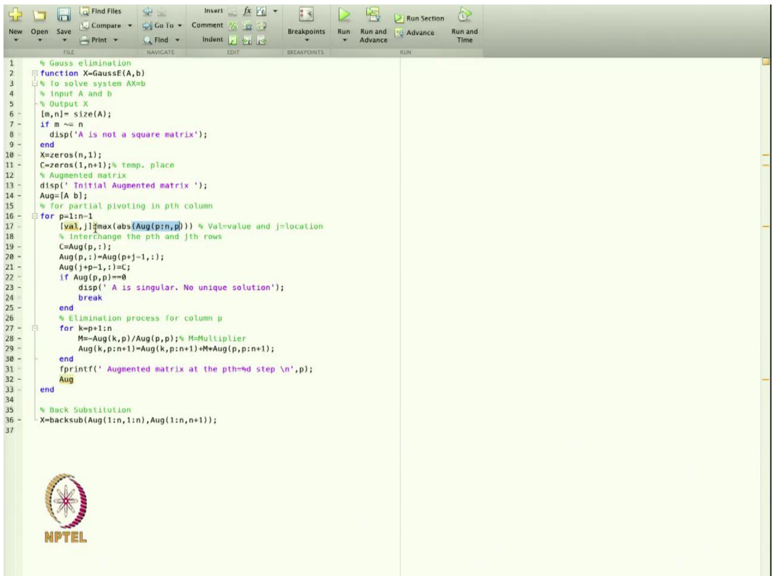


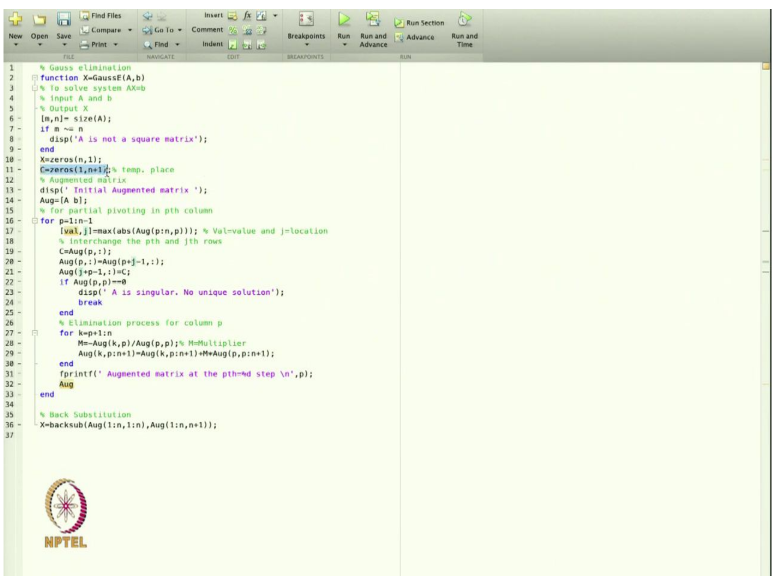
So, in this case we will say that this is not a square matrix. Now, after that I will define the value of X, so X is a vector, so in this case I will define a vector of zeros (n,1), n means n number of rows with 1 column, it means using this one I am initiating the vector X with all zero values and it is a column vector whose length is n. so, it is basically a matrix of n number of rows and 1 column, so it is a column vector.

Now, I define a temporary vector C, so that is also zeros (1, n+1) so it means in this case I am defining the row vector with all the values 0, 1 row and n+1 columns. So, this is why we are doing it because we have to deal with the augmented matrix and the augmented matrix. We know that the size of the number of columns increased by 1, so that is why we are taking n+1.

Now, I define the augmented matrix, so here I have written the display initial augmented matrix, so this is the augmented matrix, so I will put the semicolon, so I define a matrix Aug means is the augmented matrix, so this is the A matrix and the next column will be b. So, now I want to do the partial pivoting, so for the partial pivoting what I do? I will start with the p, p means partial pivoting I am doing and I will move from 1 to n-1.

(Refer Slide Time: 05:35)



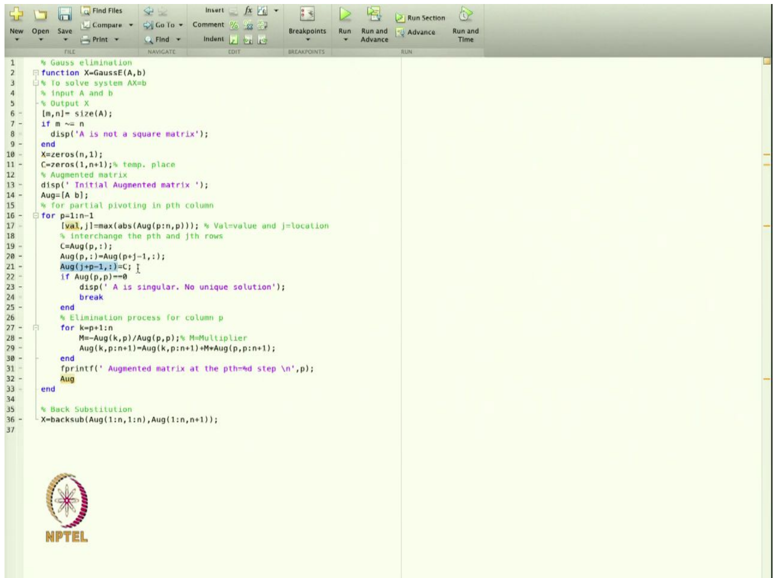


```matlab
% Gauss elimination
function X=GaussE(A,b)
% To solve system AX=b
% Input A and b
% Output X
[m,n]= size(A);
if m ~= n
  disp('A is not a square matrix');
end
X=zeros(n,1);
C=zeros(1,n+1);% temp. place
% Augmented matrix
disp(' Initial Augmented matrix ');
Aug=[A b];
% for partial pivoting in pth column
for p=1:n-1
    [val,j]=max(abs(Aug(p:n,p))); % Val=value and j=location
    % interchange the pth and jth rows
    C=Aug(p,:);
    Aug(p,:)=Aug(p+j-1,:);
    Aug(j+p-1,:)=C;
    if Aug(p,p)==0
        disp(' A is singular. No unique solution');
        break
    end
    % Elimination process for column p
    for k=p+1:n
        M=-Aug(k,p)/Aug(p,p);% M=Multiplier
        Aug(k,p:n+1)=Aug(k,p:n+1)+M*Aug(p,p:n+1);
    end
    fprintf(' Augmented matrix at the pth=%d step \n',p);
    Aug
end

% Back Substitution
X=backsub(Aug(1:n,1:n),Aug(1:n,n+1));
```


```matlab
% Gauss elimination
function X=GaussE(A,b)
% To solve system AX=b
% Input A and b
% Output X
[m,n]= size(A);
if m ~= n
  disp('A is not a square matrix');
end
X=zeros(n,1);
C=zeros(1,n+1);% temp. place
% Augmented matrix
disp(' Initial Augmented matrix ');
Aug=[A b];
% for partial pivoting in pth column
for p=1:n-1
    [val,j]=max(abs(Aug(p:n,p))); % Val=value and j=location
    % interchange the pth and jth rows
    C=Aug(p,:);
    Aug(p,:)=Aug(p+j-1,:);
    Aug(j+p-1,:)=C;
    if Aug(p,p)==0
        disp(' A is singular. No unique solution');
        break
    end
    % Elimination process for column p
    for k=p+1:n
        M=-Aug(k,p)/Aug(p,p);% M=Multiplier
        Aug(k,p:n+1)=Aug(k,p:n+1)+M*Aug(p,p:n+1);
    end
    fprintf(' Augmented matrix at the pth=%d step \n',p);
    Aug
end

% Back Substitution
X=backsub(Aug(1:n,1:n),Aug(1:n,n+1));
```



So, here I am finding the first I am finding the this is the augmented matrix I am taking, so in this case I am finding augmented matrix p to n means in this case first starting with p=1:n, so I am going through all the rows and with the first column only, because I need to find the maximum element only in the first column in the step 1, so here I am taking all the elements from 1 to n with the first column.

And then I am finding the absolute value because we want in magnitude and then I am finding the max, so it will give you the maximum value, so I am finding this maximum value, so it will give you the vector value, so suppose the maximum value is coming 4 and it is in the second row, so it will give you the value, value will be 4 and j will be 2, so it gives you the value and the location where it is find it is getting the maximum value.

So, from here I will get the val and j, because this value I am not going to use. So, I just finding that what is the value and j is the, so once I find the value of j, then what I do is that augmented matrix so 1 and all columns, so I take it to the C. So, I just define because here I am defining this one, so that is the temporary place I have defined.

So now I am using this temporary place here that p means the first because p is 1 here, so I am taking the p is equal to 1 first row and all the columns. Then I am putting the value C, now what I do is that I am interchanging rows with the jth, so I am interchanging the pth row with the jth. So, in this case what am I doing? I am changing my augmented value in the pth row.

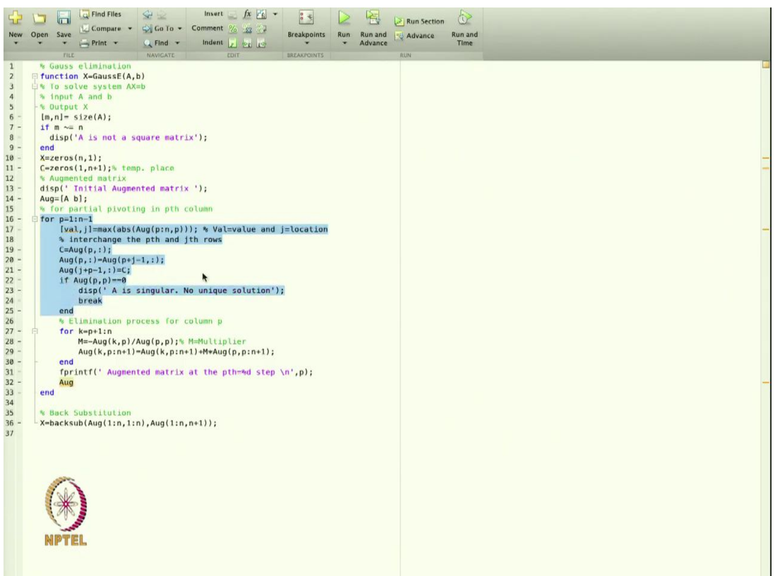
So p is the row and moving all the columns it means I am finding the row pth row a, so I am changing this value of with the p+j-1 i. So, this is the p and j suppose in my case it was supposed

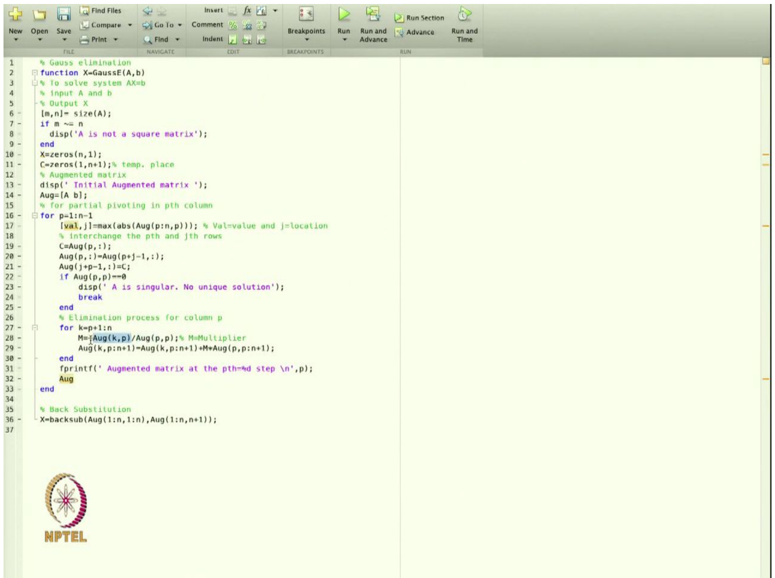
2, so I am finding the value 1+2-1 so 2. Because we are doing here p+j because as the step will increase then our, the size of the matrix will reduce.

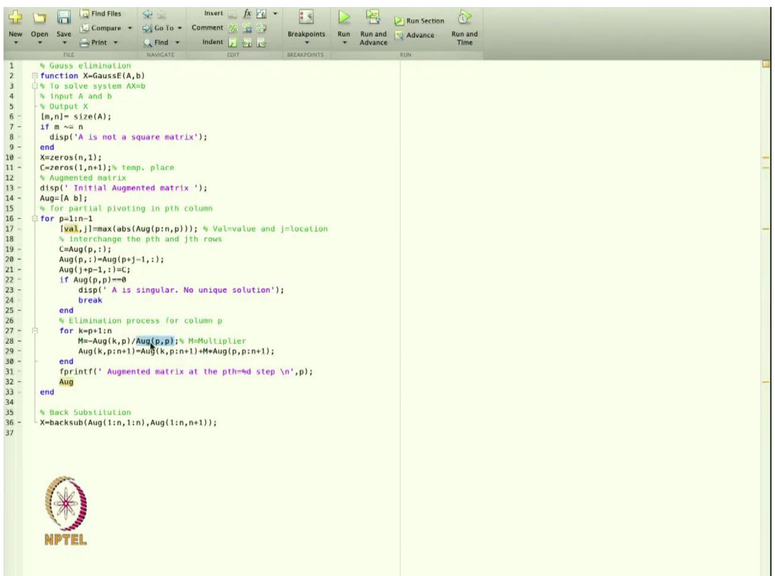
And in that case we have to find out if we have to keep it p+j, so I am putting the p+j-1 and all the column vectors. So, I am sending all this value in the pth row and then in the pth row I am finding because here I have saved the value of the pth row so I am putting back this value of the pth row in this row.

So, using this one I have interchanged the rows. And now I am finding that if the augmented value p p means 1 1 is 0 then I will say that okay this matrix is singular no unique solution in there, because you know that I am reducing it to the upper triangular matrix and in the upper triangular matrix somewhere if I am finding that the that the diagonal elements becoming zero, then I will say that this is a similar matrix and then it will be no solution and it will break that.

(Refer Slide Time: 09:06)







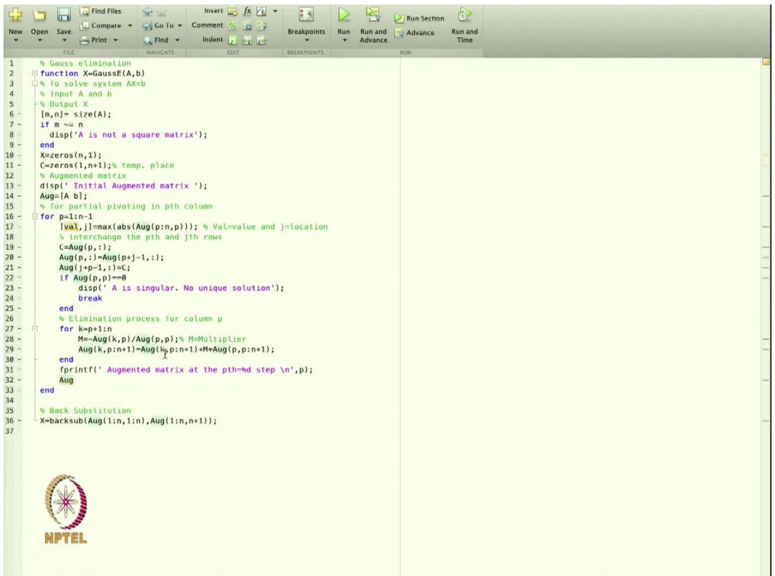
So, after that what do I do? So, once I am using this for loop I am finding the diagonal dominance matrix or I am finding that wherever in the first row or in the first column I am able to find that where is my maximum element I am and I have interchange that maximum element with the first row. So, after that I want to do the elimination process, so in the elimination process, I have to find that what is my multiply and then so I again start with the for loop and k is from p+1 to n because if p is 1 then I have to start with the elimination from the second row, second, third and onward.

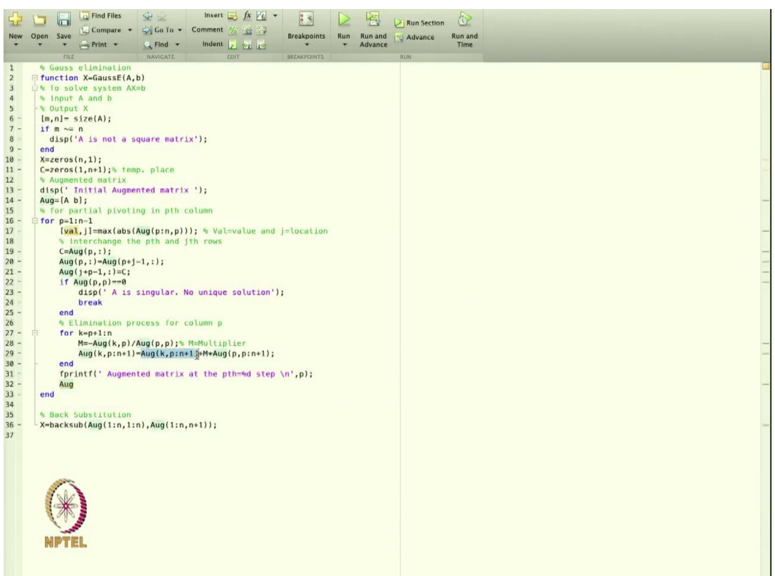
So, I will start some p+1 to n then this is my multiplier. So, minus the whatever the matrix was there augmented matrix the k and p, so I will divide by the element where I want to do the elimination divided by first element because if I take p=1 it will be the element p_{11}, so I will divide by this 1 and multiply by that number and take the minus sign.

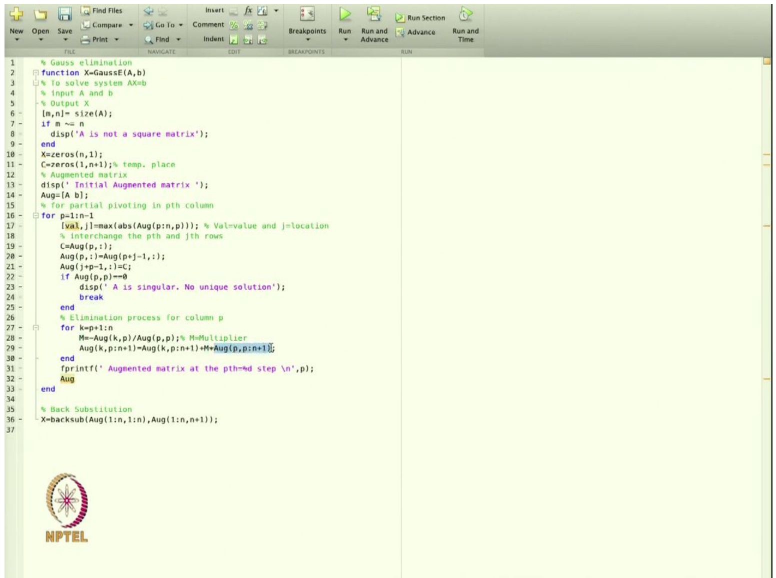
Then what I am doing is finding this value k, so whatever the value a k is there, so p is from 1 to n+1. So, I am taking that row the kth row and I am moving all these values from p to n +1 the same columns. Because I have to do all the things in the first column, suppose I take the first column.

So in the first column I have to make all the elements 0 except the first element. So, in this case my k is 2 and starting from 1 to n+1, so that will give me the value of the column.

(Refer Slide Time: 11:01)





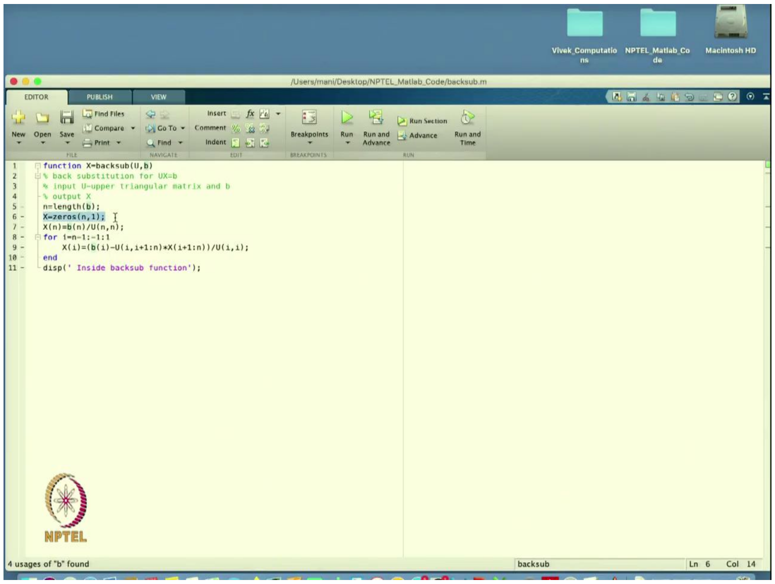


So, in this case, it is equal to the value the matrix k p to n+1+m that is a multiplier and multiply by the values on the previous row which I am using to give the elimination. So, this is the if p=1 it is the 1 and 1 to n + 1, so it is the row and all the columns. So, I will this is the first row and all the columns I am multiplying by the multiplier and adding to the next row where I want to make the element zero.

So, it will reduce to the it will give the elimination process and then I this will end, so using this one I will make all the elements I will do the elimination process in the pth in the kth row so whatever the value of the k here, then after that I have written that fprintf augmented matrix at the pth step, so I am putting the pth step and I am printing the value of whatever the matrix is there. And now this is the end of this for loop.

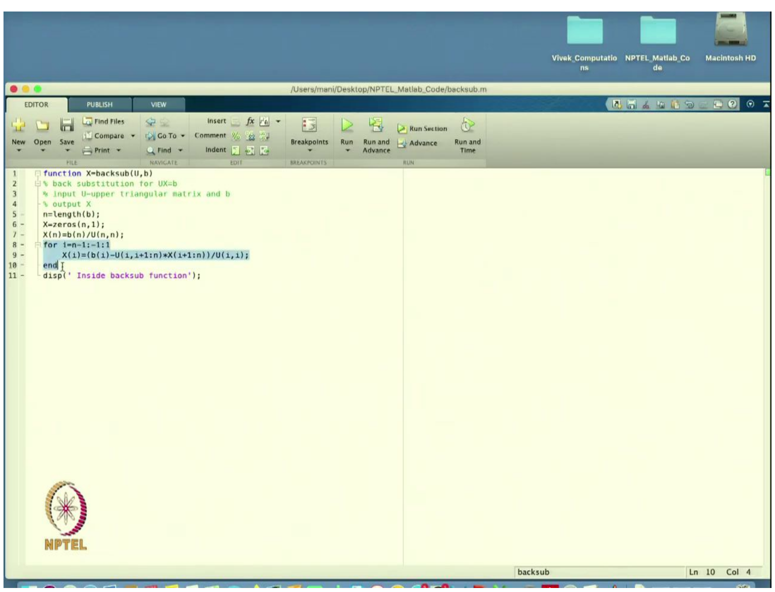
And after that I am finding the applying the back substitution method to get the solution. So, in this case, I am calling a function from itself that is called a back substitution. So, what is the back substitution? So, let us see another one.

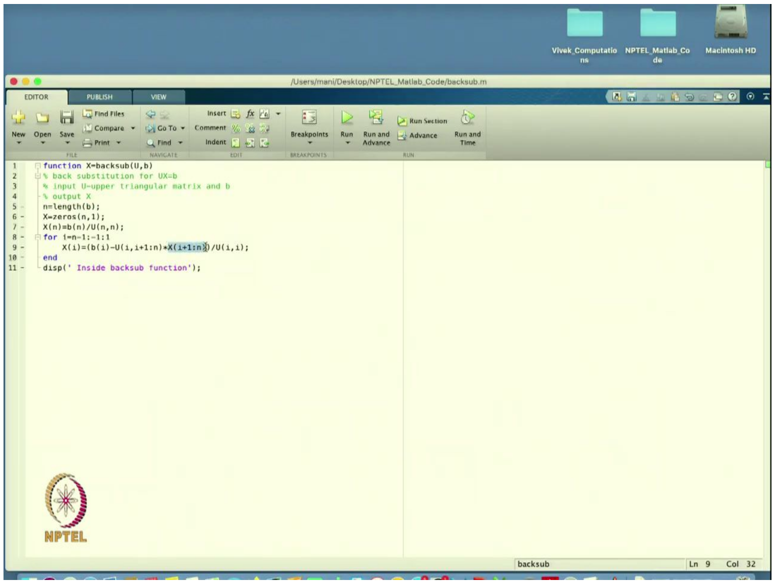
(Refer Slide Time: 12:32)



So, I have defined another function and that is back substitution. So, in the back situation, I know that the matrix should be upper triangular, so I will input the value of the upper triangular matrix and this is the right hand side vector. So, using this one I will find the length of the, what is the length of b, then I put X=zeros (n,1) so it is a column vector basically. And then I know that the last element $X(n) = b(n)/U(n,n)$, because it is going from bottom to top.

(Refer Slide Time: 13:10)



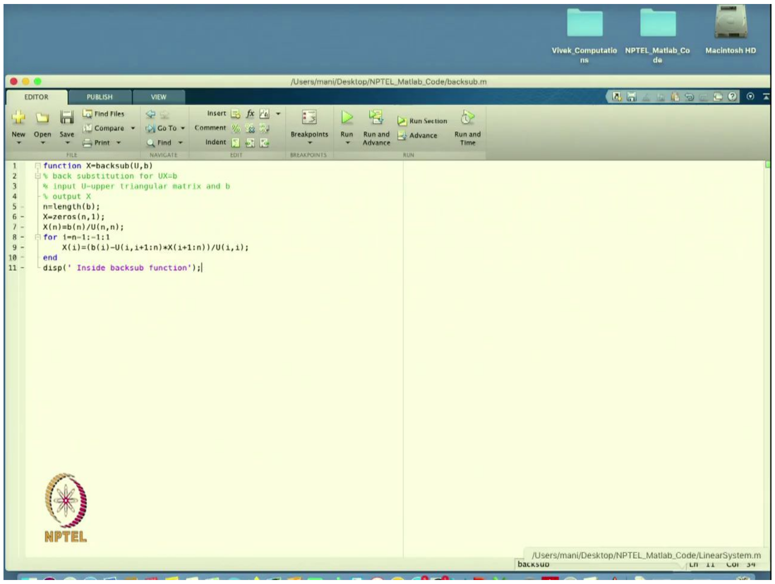


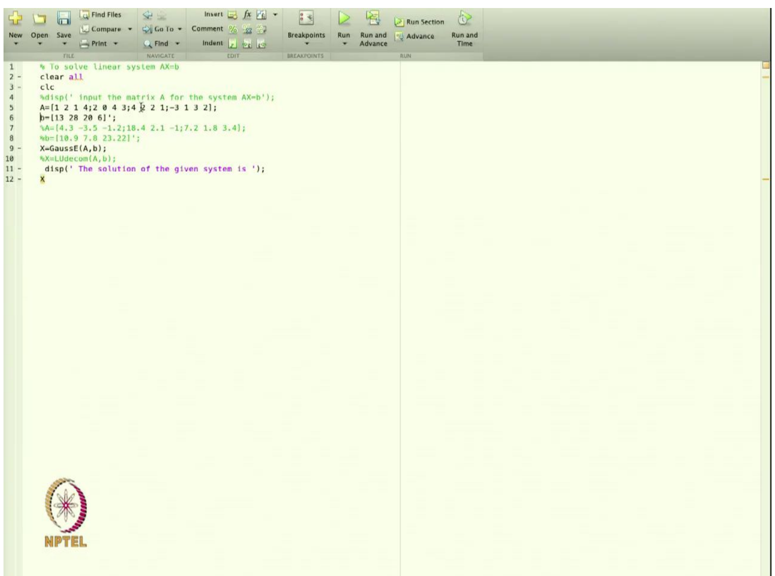
So, first value will be X(n), I am finding this one and then with the help of this iteration, I will find all the values because once I know the value of X(n), then I will find the value of i from n-1 up to 1 with the increment of -1, so it will give you the value of n -1, n- 2, n-3 up to 1.

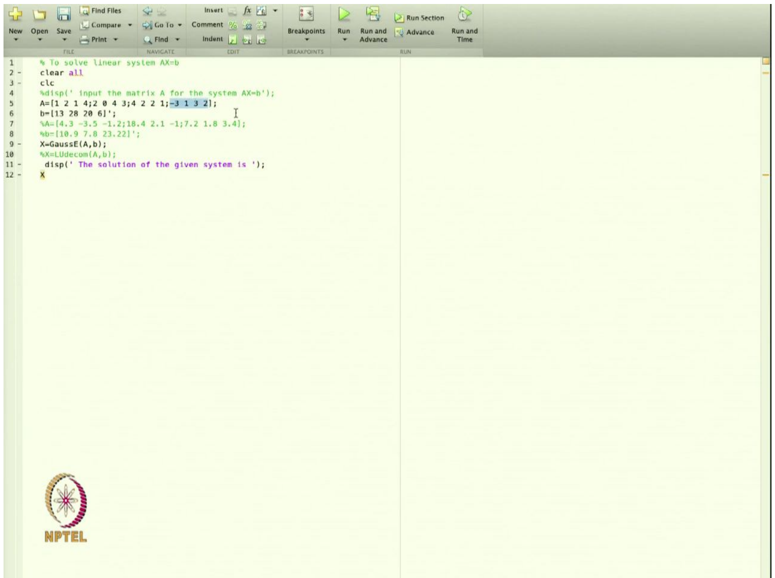
And then I will find the value of solution $X(i) = b(i) - U(i, i+1:n)$ multiplied by the previous value, whatever the previous value we have found from here and divided U(i, i). And this is the end of the loop and just to check that whether if I run the code my code is going to this function by generally writing like this one.

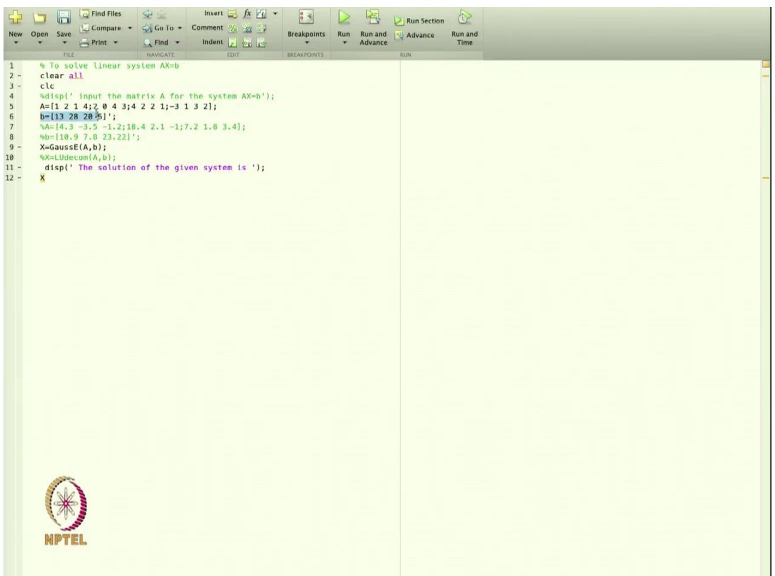
So, it will show you that inside backs substitution function. So, it will give you the clearance that okay you whenever you run the program your function is going to this one. So, after doing this one.

(Refer Slide Time: 14:10)





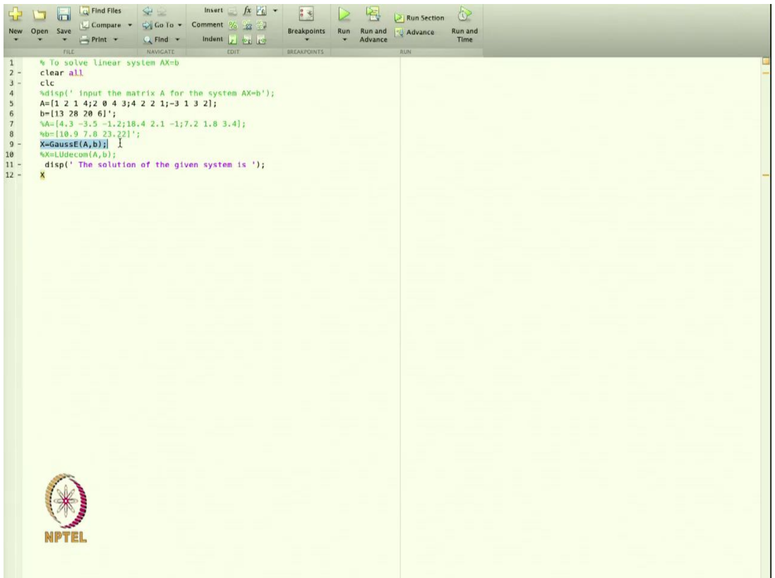




Now, I define the main script, so that means the script I write as a linear system. So, it means that using this script I am solving the linear system. So, I have written that to solve the linear system AX=b, clear all clc and then now I will start with the matrix. So, let us just take this matrix and this is the right vector, so in this case, I am starting with the matrix A, so this is 4*4 matrix, so 1 2 1 4 that is the first row, 2 0 4 3 second row and this is the third row and that is the fourth row.

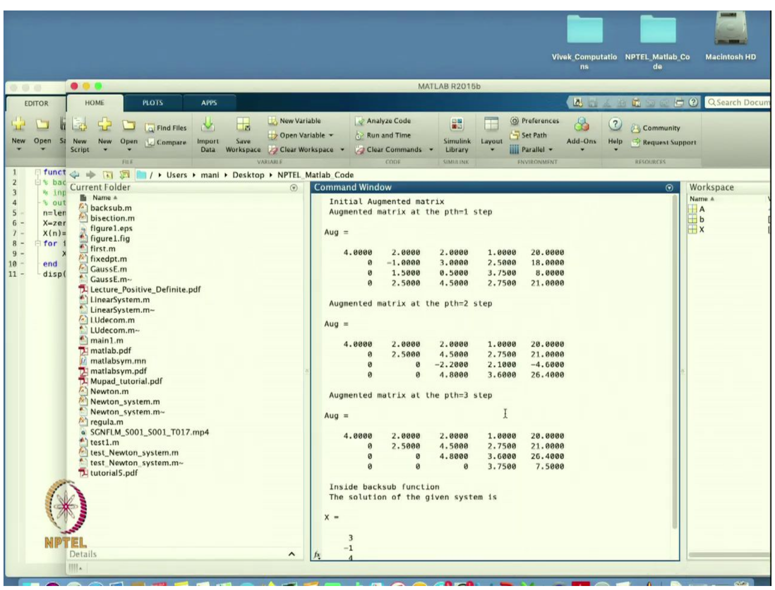
So, this is the matrix I am entering and the right hand side vector is [13, 28, 20 and 6], so this is the right hand side vector. And I am putting dash here it means that it is the column vector. So, I am passing the matrix A and the column vector at the b.

(Refer Slide Time: 15:14)



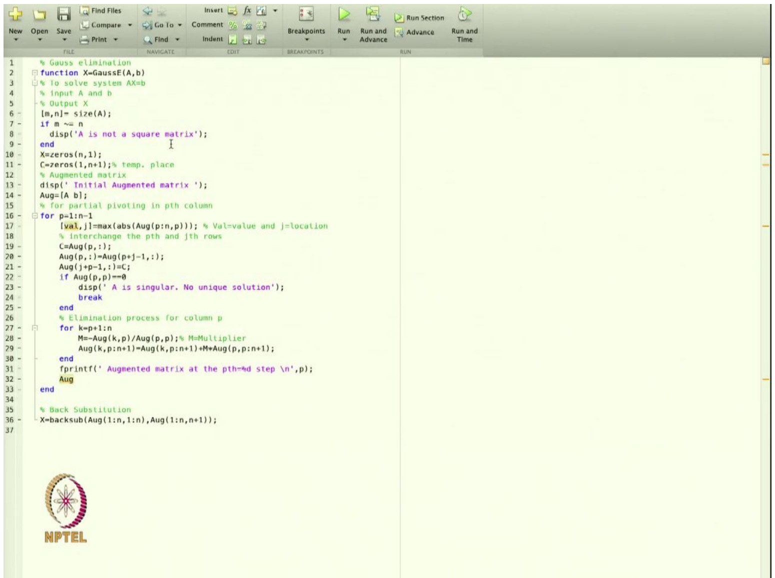
Now, after getting this value I am calling my function X=GaussE(A,b), so here I am passing this one and then once I will get the solution from here, then I am displaying that the solution of the given system is X. So, let us run this one. So, once I have done this one.

(Refer Slide Time: 15:36)



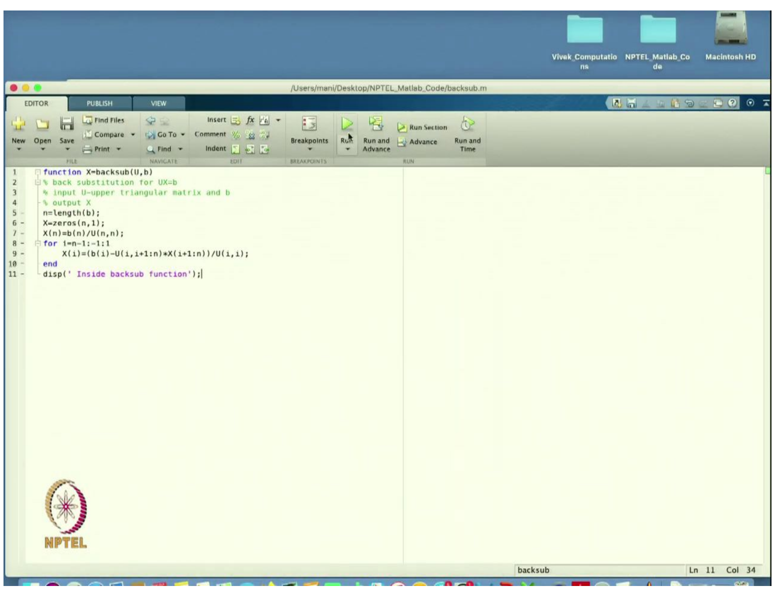
So, that is the I am getting the solution. So, initial augmented matrix,

(Refer Slide Time: 15:54)



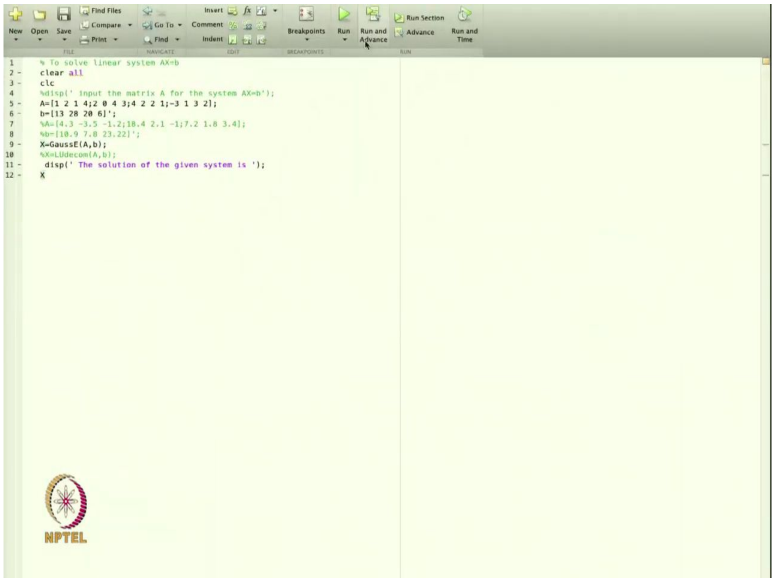
So, first I have to find the this one see, so I have made the initial augmented matrix, so after that I made this semicolon, so I will remove the semicolon, now it will based on this one, so it will show the values, so now I think it will give the value so I will save this one.

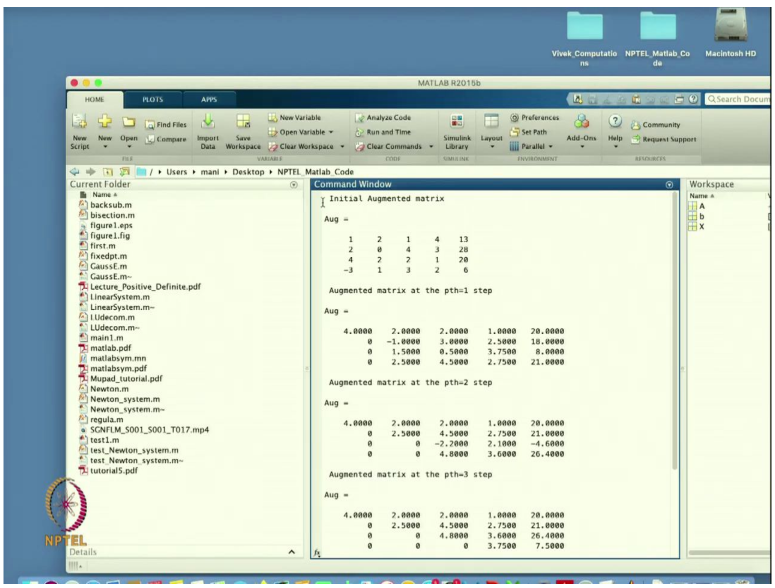
(Refer Slide Time: 16:14)



And then I will save this one also. So, it is no use.

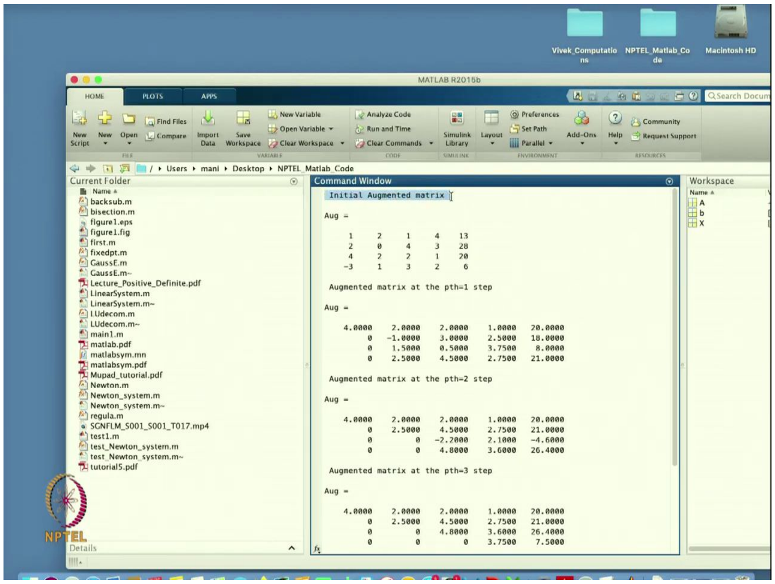
(Refer Slide Time: 16:21)

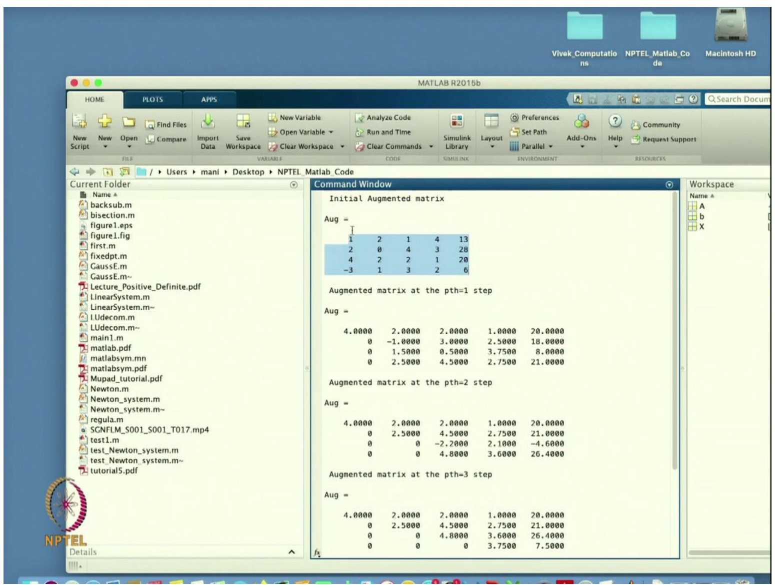




So, this is my value, so I will rerun the code and that is my value.

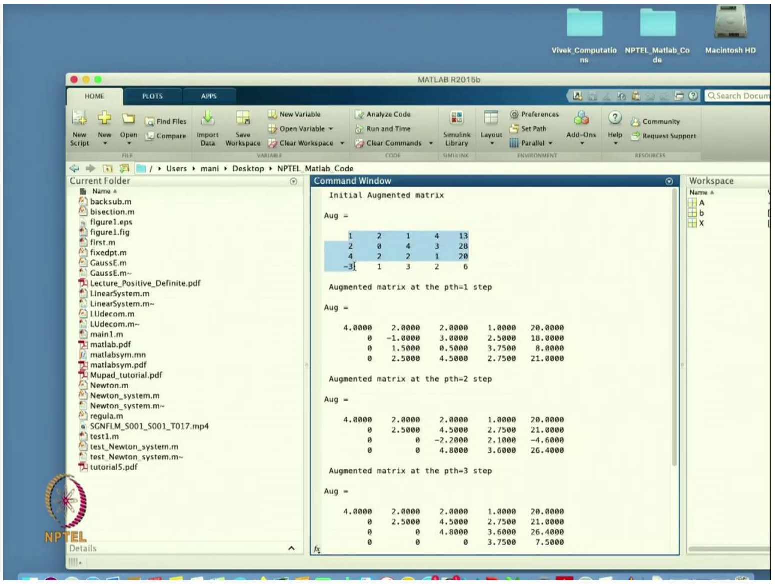
(Refer Slide Time: 16:30)

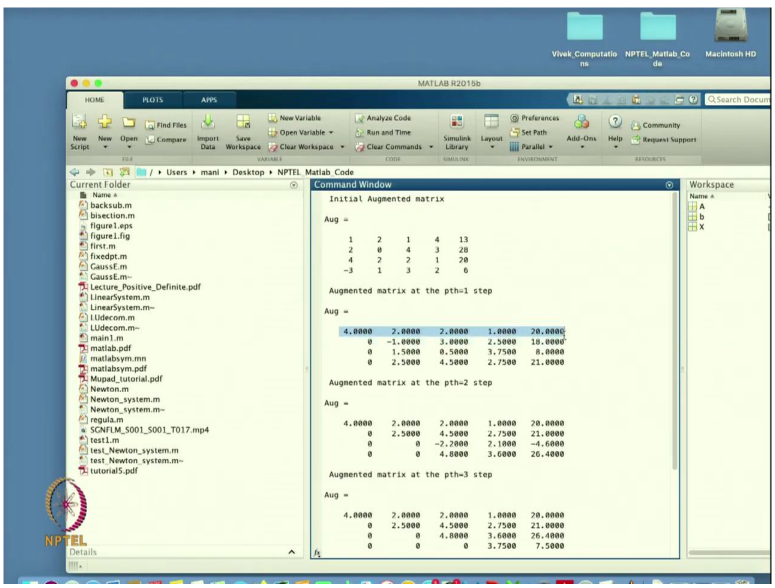




So, in this case I can see from here that the initial augmented matrix is this one it means this is my augmented matrix, so that is my 4*4 is the matrix and this is the right-hand side b.

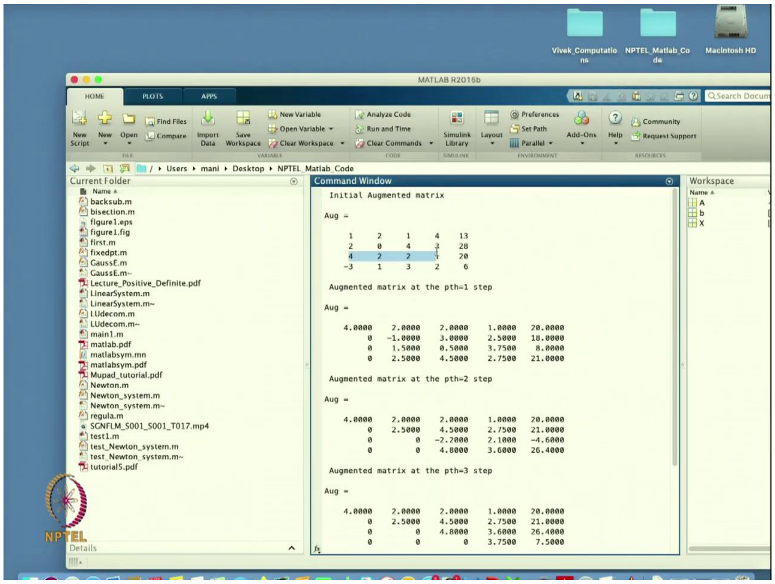
(Refer Slide Time: 16:48)

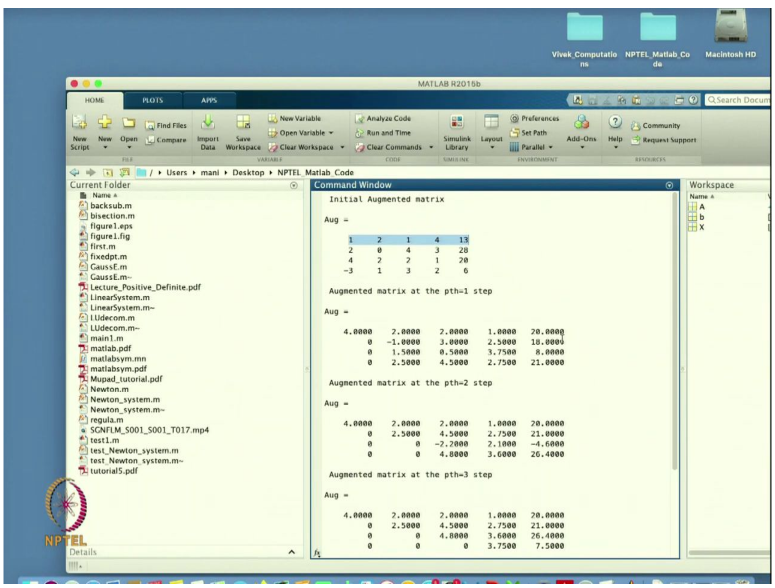


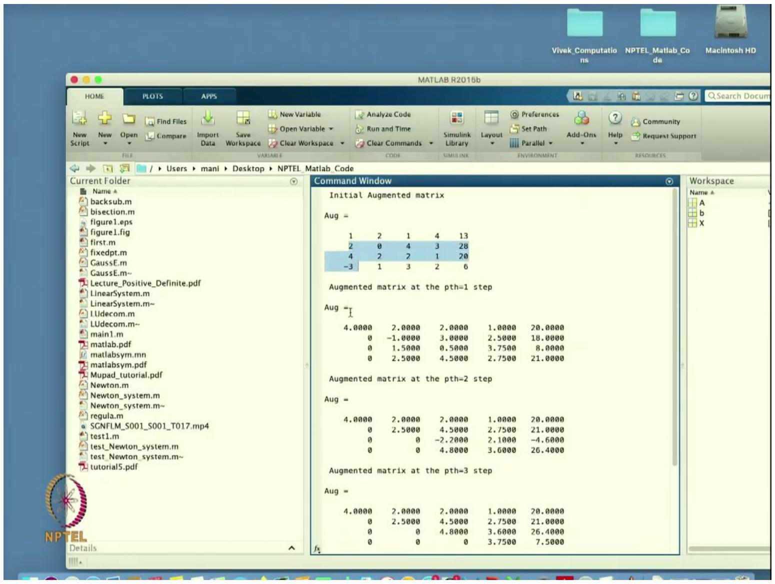


So, now after the first step, you will see that in this case, it will first check that way the maximum element and you can see that 4 is a maximal element. So, first it will change this element to the first row, so here you can see that it has changed the values.

(Refer Slide Time: 17:03)

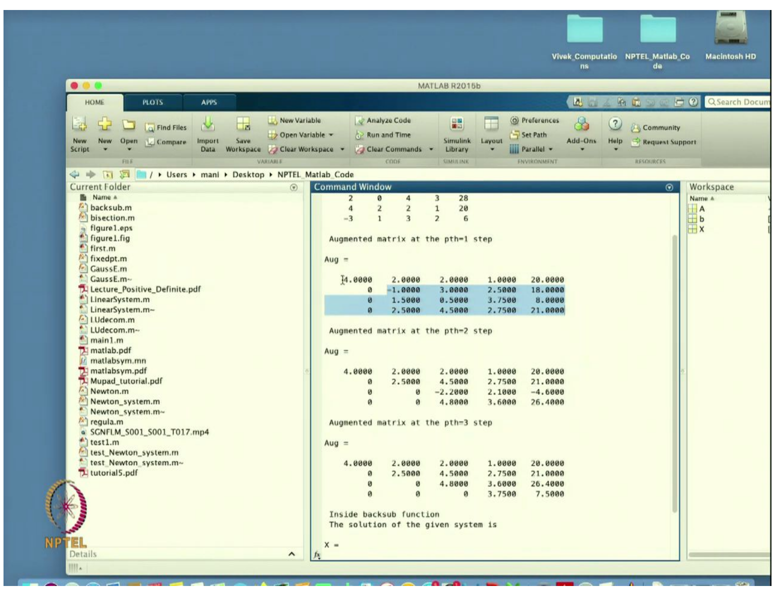


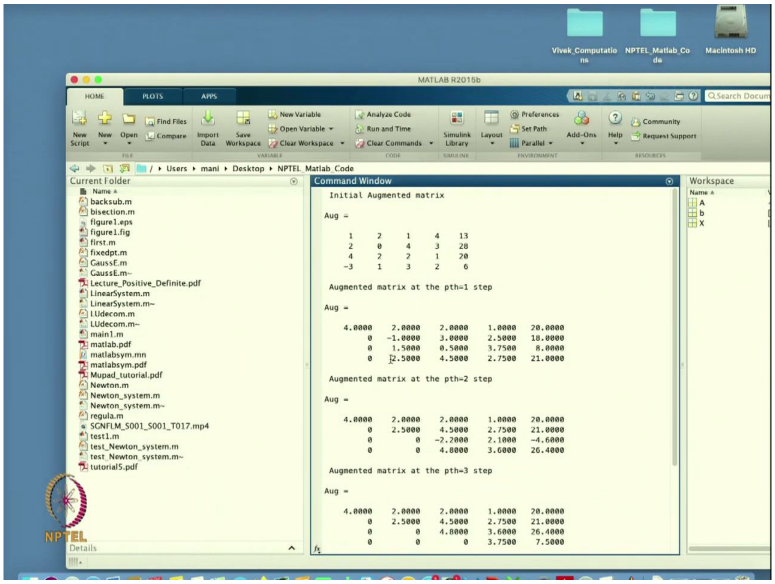


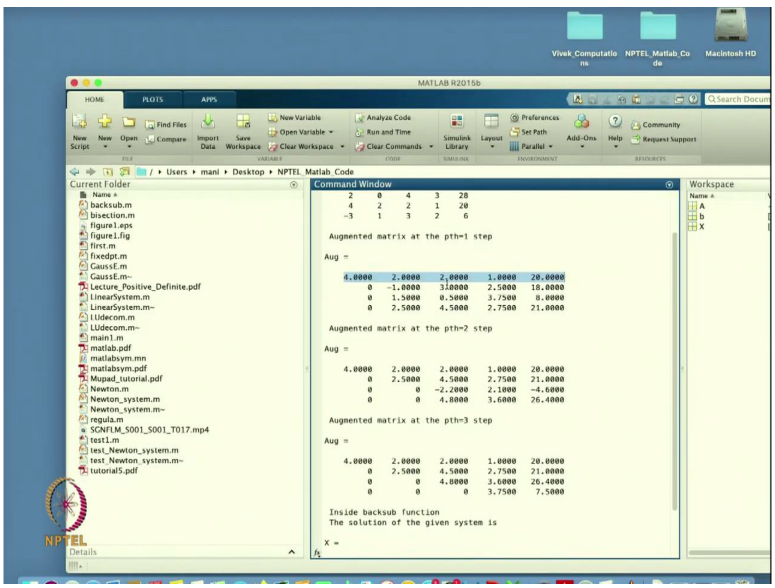


So, it is interchange this row with this, so 4, 2, 2, 1, 20 so that is the partial pivoting has happened at the step 1, after that it is (move) eliminating all the elements in the first column, below the first one, so here it is putting 0 getting 0, 0 and 0 so that is my first step.

(Refer Slide Time: 17:32)



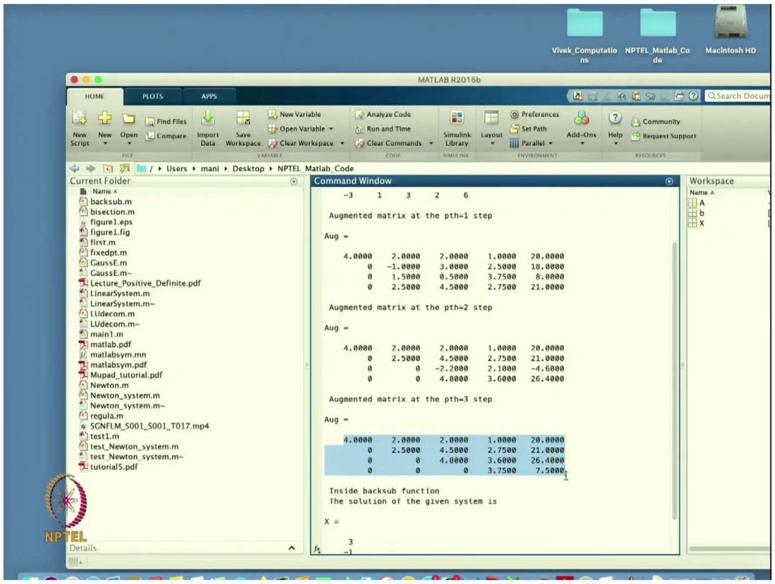


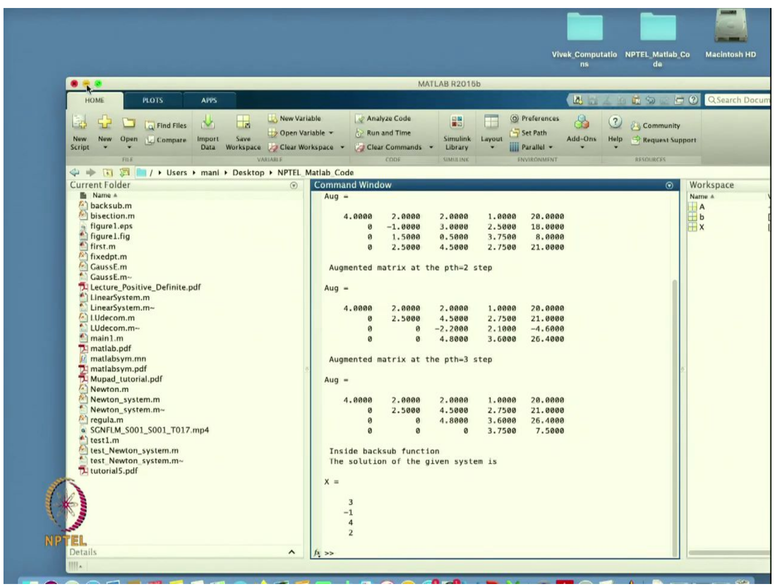


So, after the second step now again, you can see from here nowadays checking which one is the largest value in this, below this so I will leave the first row now because that is over, now I will check the maximum element in the remaining element, remaining element in the second row. So, the second row is 2.5 in the fourth row, so what will I do now? I will swap this one with this value. So, it will be 2.5, 4.5, 2.75 and 18 and 21.

So, this one has been changed and now I am making all the elements below this one is 0 and this is after the augmented matrix at the third step will be this one. So, this, this and this and then this is my upper triangular matrix.

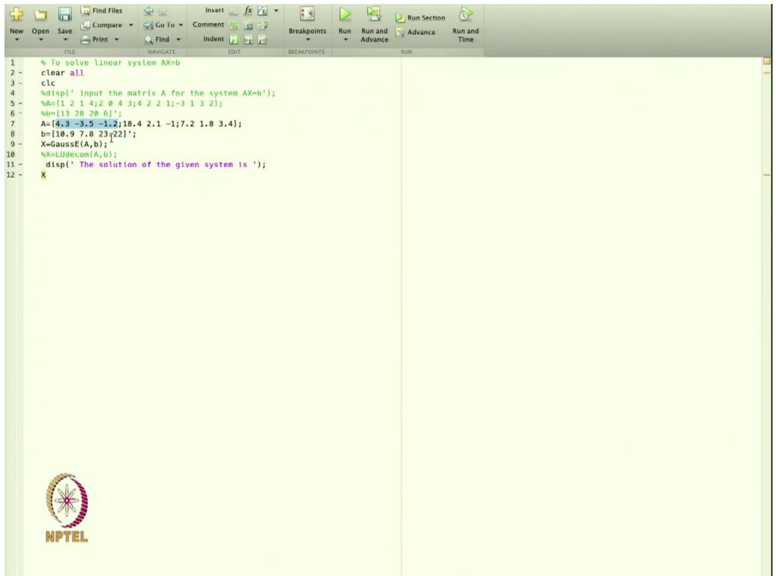
(Refer Slide Time: 18:29)

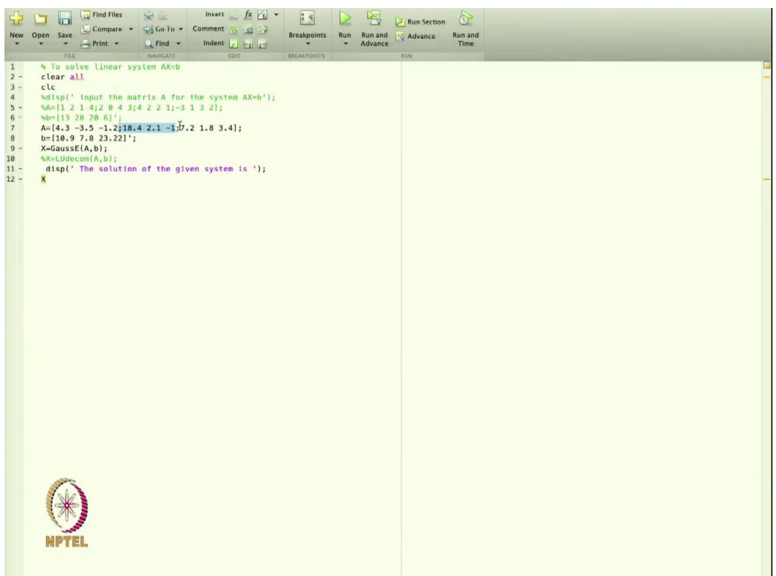


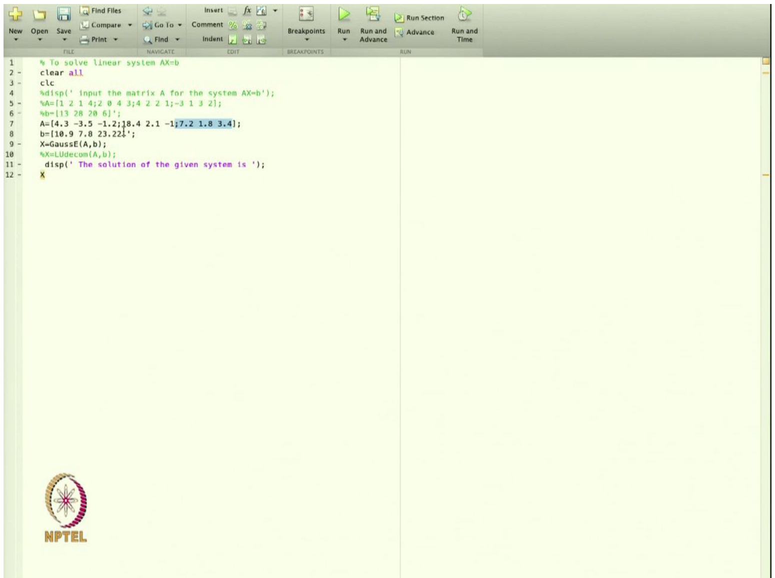


So, based on this one, I can say that this is my u, so the first four columns of the u and this is the right hand side new vector. Now, after that it will go inside the back substitution function and once based on this one, I will give the solution, so solutions are coming 3, -1, 4 and 2, so that is my solution. So, from here, I can say that my solution is 3, -1, 4 and 2. So, that is my solution. Now maybe I can change the system and I will go with the new system.

(Refer Slide Time: 19:13)

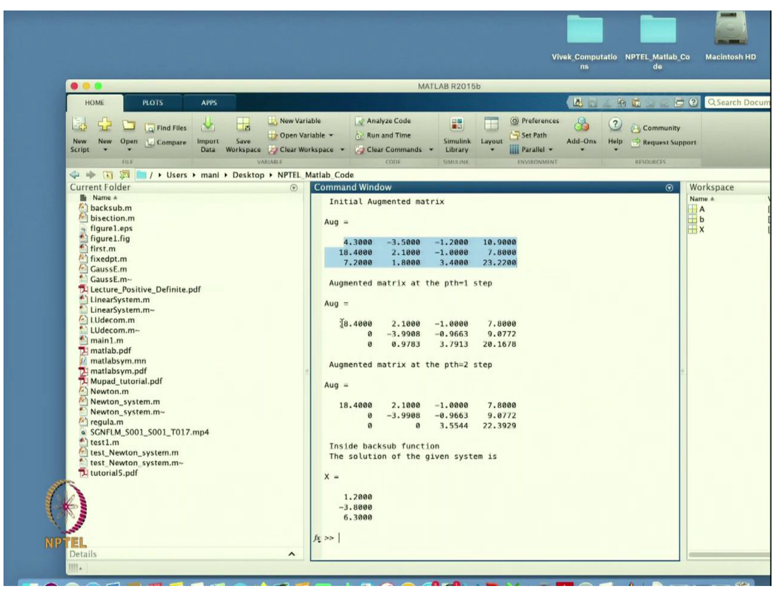


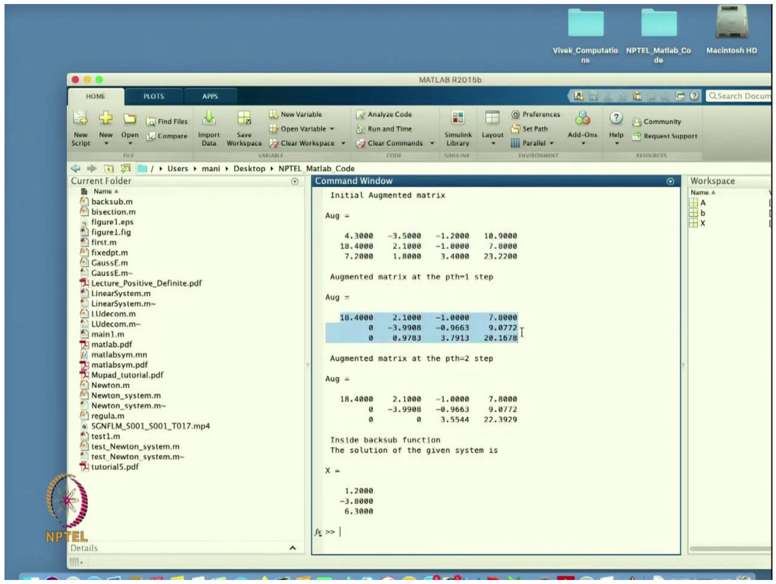


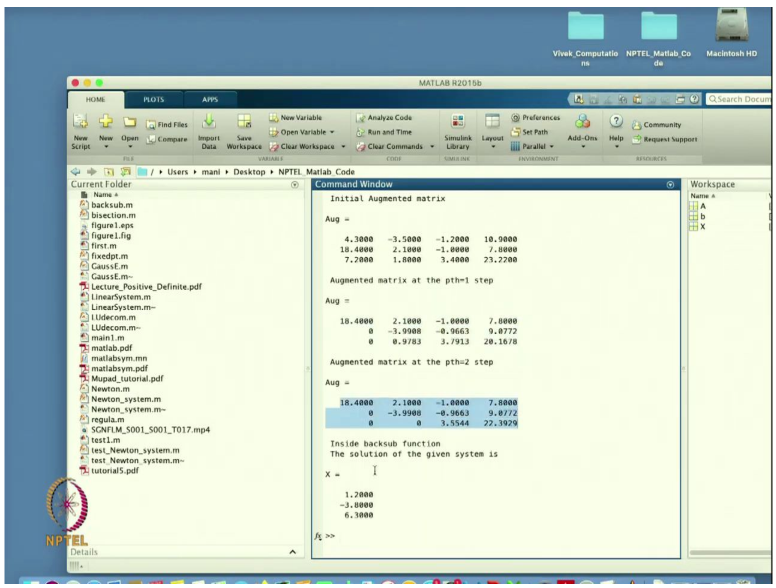


So, this is the matrix I can just comment this one and I can take down another matrix this one, so the previous matrix has all the values as an integer, but what will happen if I take the floating point numbers or real number? So, I am taking a 3*3 matrix now. So, this is my first row, it has the value 4.3, -3.5 and -1.2. So that is my second row. So, after the semicolon will whatever we write that is the second row and this value is the third row and right hand side vector this. So, I will now run this code.

(Refer Slide Time: 20:03)

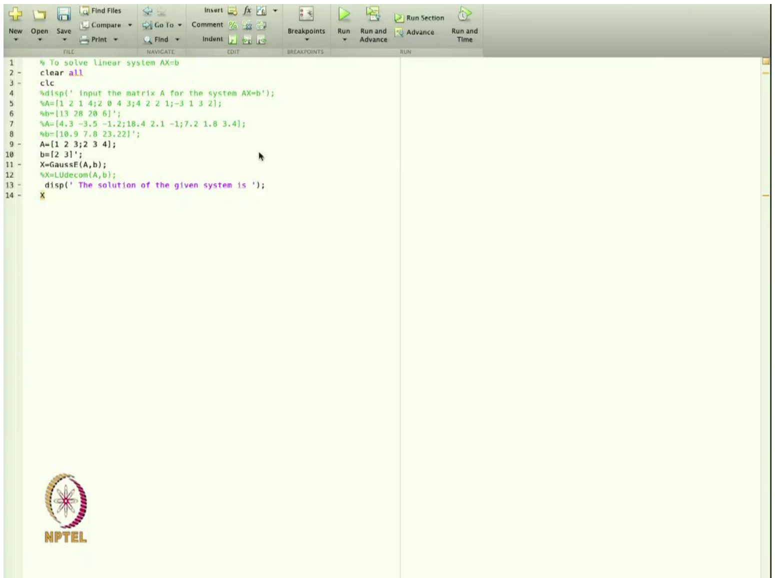


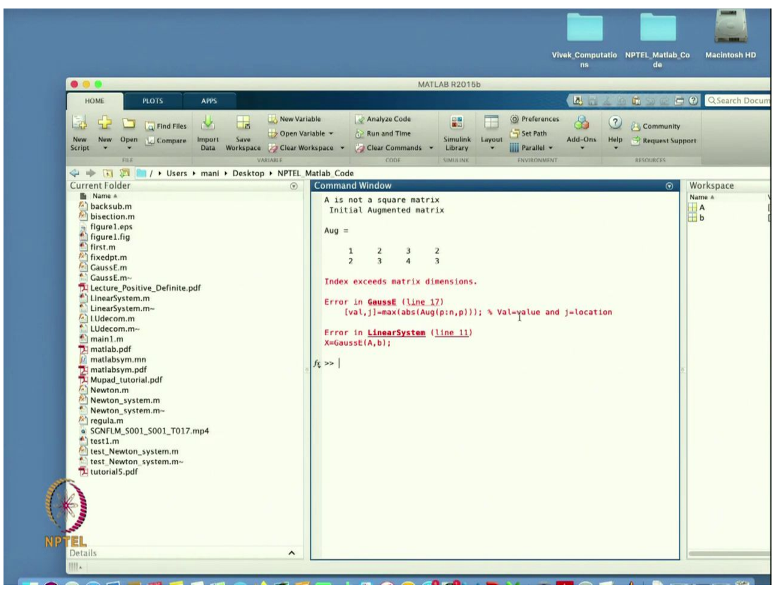




And this is my value. So, my initial augmented matrix is this one, basically, that is my system and at the ith step first step because in this case only two steps are needed. So, the first step will give me this value and the next step will give me this value. So, this is the upper triangular matrix and then I will use the back substitution and that is my value solution.

(Refer Slide Time: 20:36)

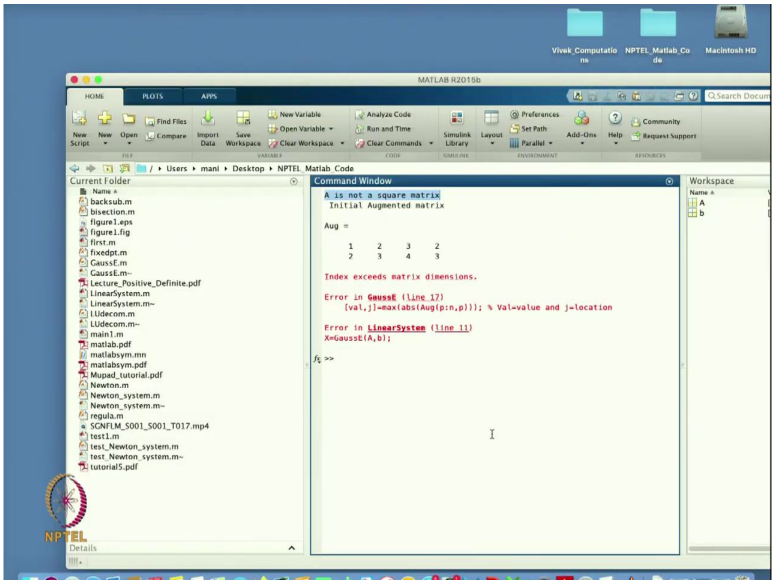




So, in this case, you can see that it is giving you first it is doing the partial pivoting to make the matrix diagonal dominant and then you are able to find that the solution is there using the Gauss elimination method. So, in this case, we are able to solve this one. Now, what should I take? Now, suppose I take the institute of this one, I will give you a matrix, which is not a square matrix. What will happen?

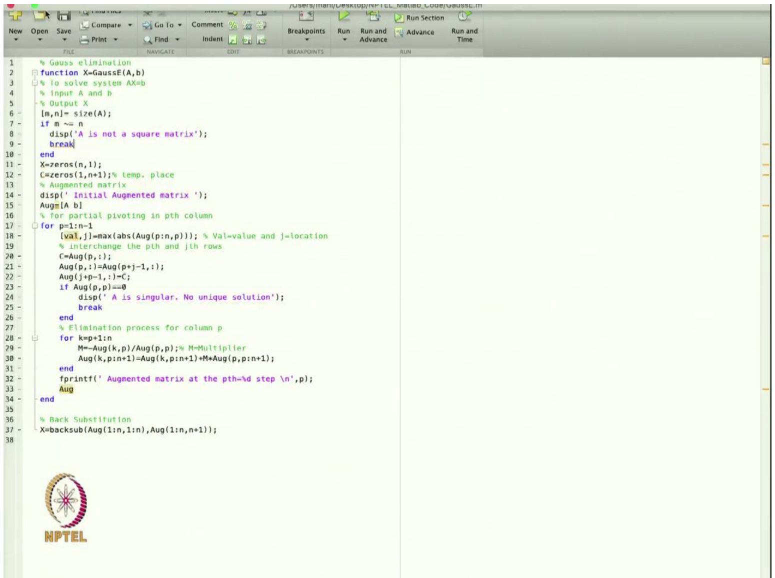
So, let us take a square matrix rectangular matrix, so I am taking 1 2 3 and then 2 3 4, so suppose I take this method. So, this matrix is a 2*3 matrix and then I am taking my b as 2 3, so it is a column vector with two elements and let us check what will happen. So, if I run this code it will go to this one. So, exceeds dimensions so in this case index exceeds matrix dimension and the error in this, so it will show you how I can find the maximum values.

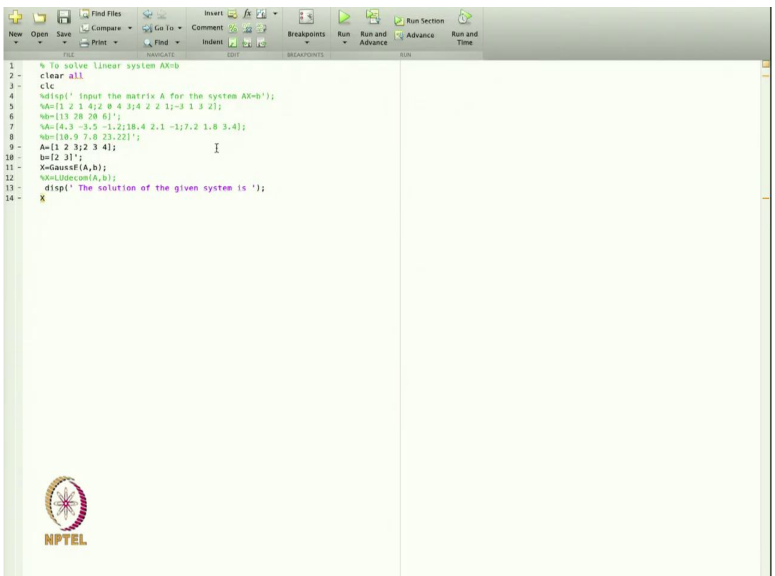
(Refer Slide Time: 22:12)



And so in this case the first command is coming A not a square matrix and then we have a. So, if it is not a square matrix because if you know that the matrix is not a square, then you have infinitely many solutions. So, in that case we are unable to use the Gauss elimination method.

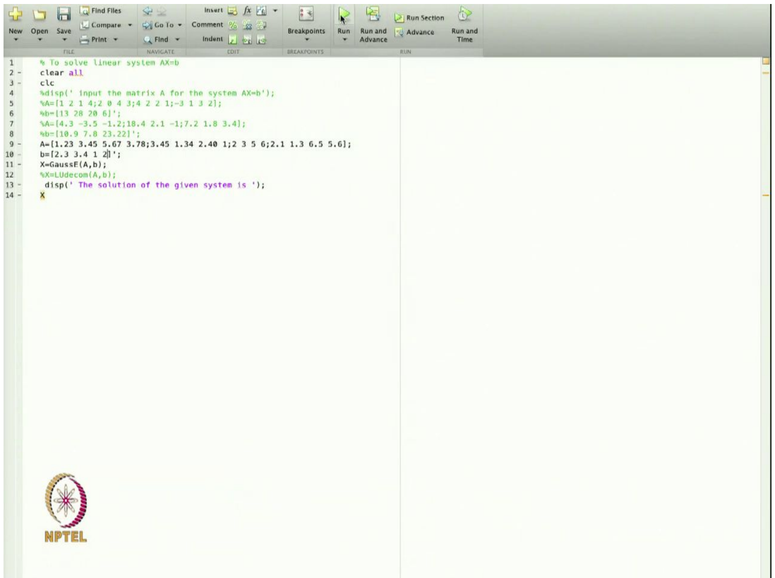
(Refer Slide Time: 22:33)





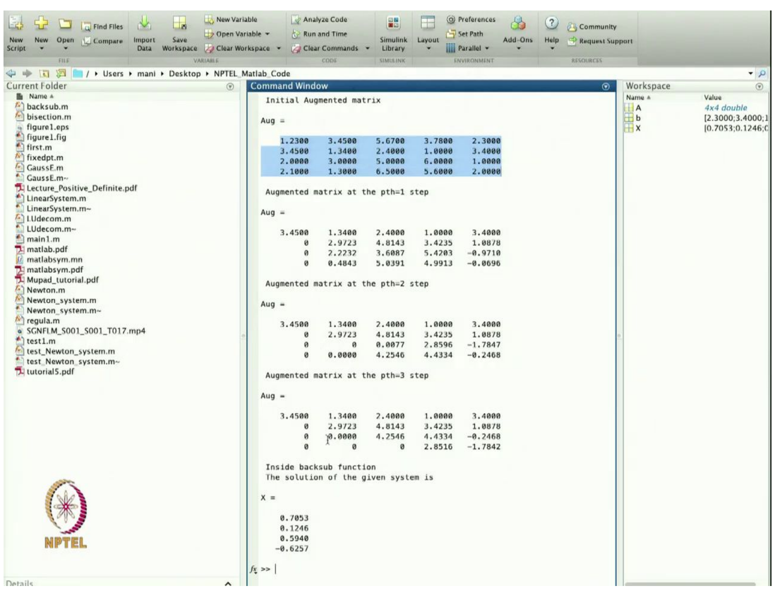
So, this is a how we can deal with this matrix. So, let us change this matrix to some other value and let us see that how we can find the value.

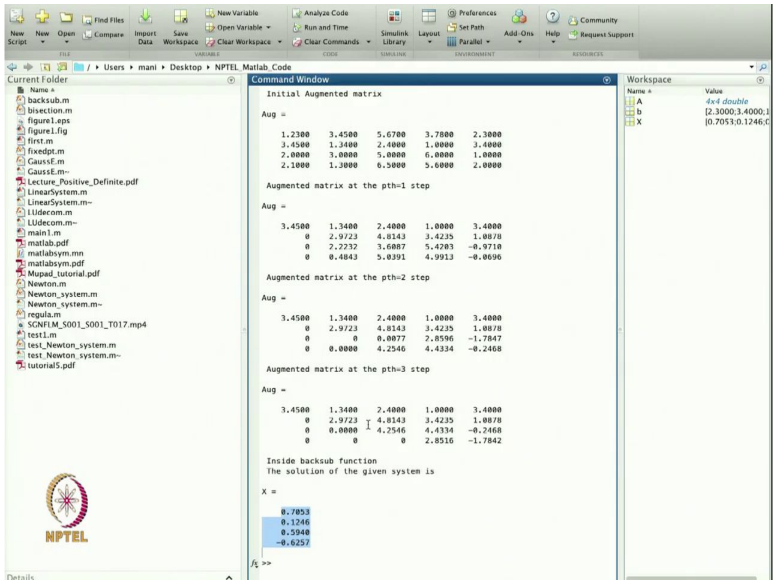
(Refer Slide Time: 22:43)

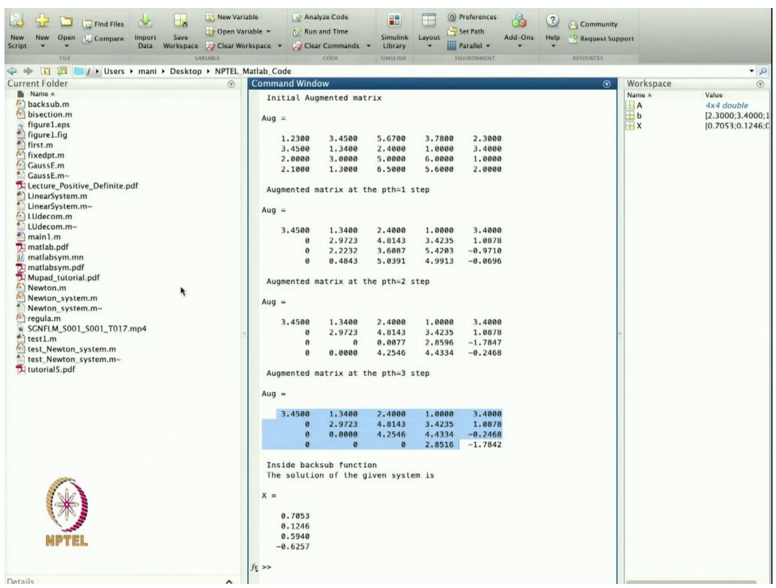


So, I just take another matrix 1.23, 3.45, 5.67 and 3.78. So, this is the first column I am getting, so I am taking 4*4 matrix then I am writing 3.45, 1.34, 2.40 and 1 so this is the second row now third row is 2 3 5 6 and the last row I am taking 2.1, 1.3, 6.5, 5.6, so I am taking 4*4 matrix and then I am the right hand side I am taking maybe suppose I am taking 23, 3.4, 1, 2, so in the case I do not know that whether this system is going to have a solution not because it is a just I am taking randomly, but I have just run this code, let us see,

(Refer Slide Time: 23:54)

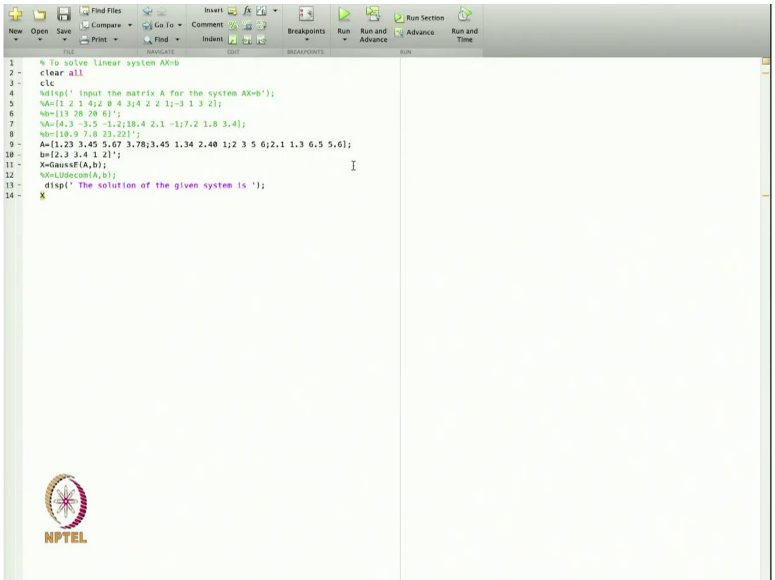






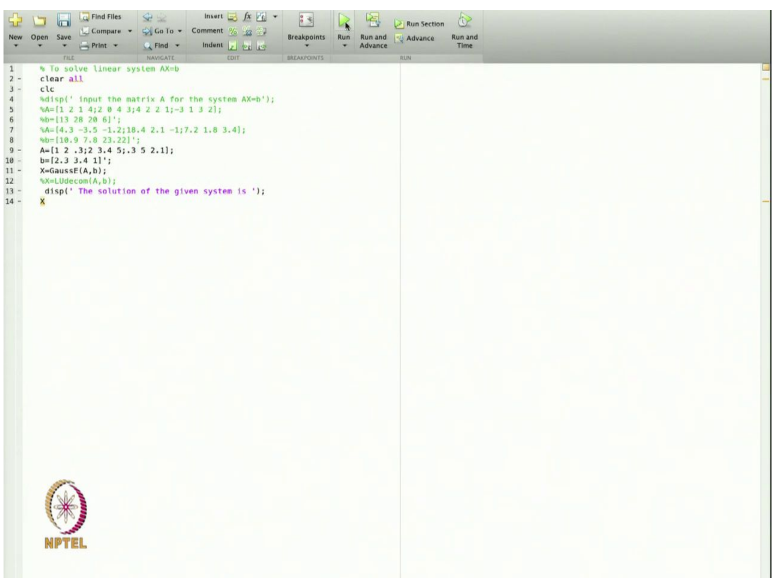
So, that is the answer, it means the matrix is a non-singular type and we are able to find this, so this is the matrix we are getting. So, this is a matrix we have started to be, so after the first step it will reduce to this, the second step will reduce to this form and the third step will reduce to this form. So, this is the upper triangular matrix and that in my solution. So, this is the solution. So, from here also, I can see that the rank of this matrix is 4, so it is non-singular matrices basically.

(Refer Slide Time: 24:39)



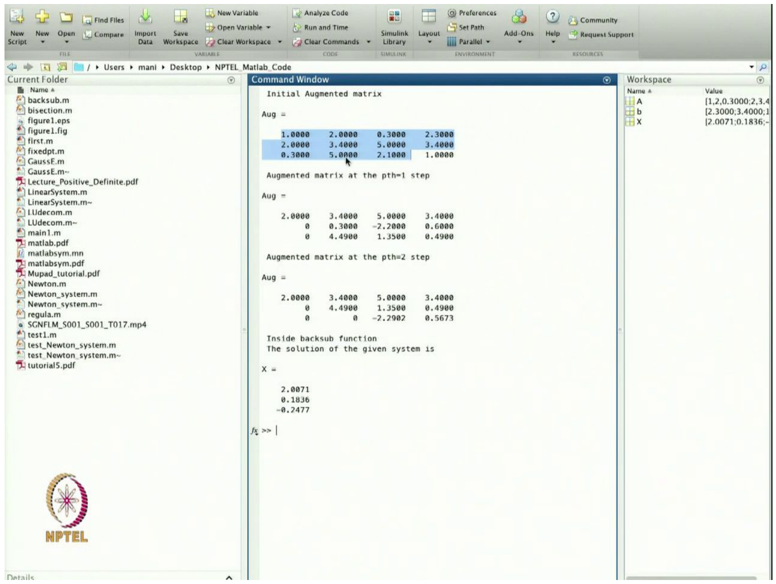
So, using this one I can find. So, let us see what will happen, if I take a matrix that is a symmetric matrix.

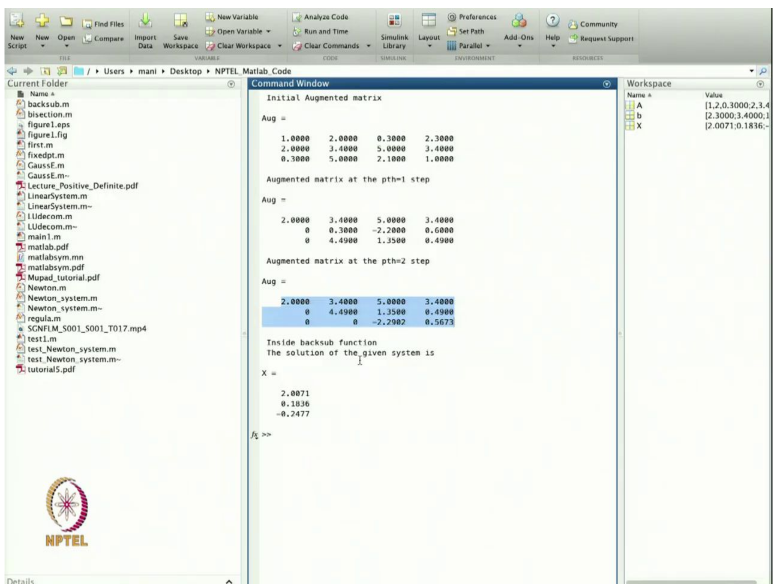
(Refer Slide Time: 24:52)

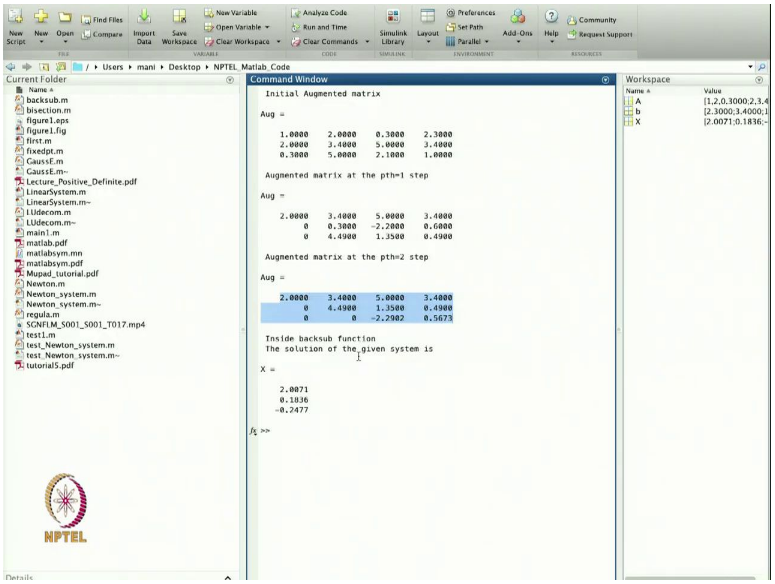


So, just give some examples, so I will start with the 1, 2, 0.3, so that is the just taking the first element and the second element I am taking 2, 3.4 and 5. So, this is the second row, the third row I am taking 0.3 and then it will be 5 and then 2.1. So, suppose I take this one, so that is the symmetric matrix I am taking and then I will take this 3 by 3 matrix, so let us run.

(Refer Slide Time: 25:41)

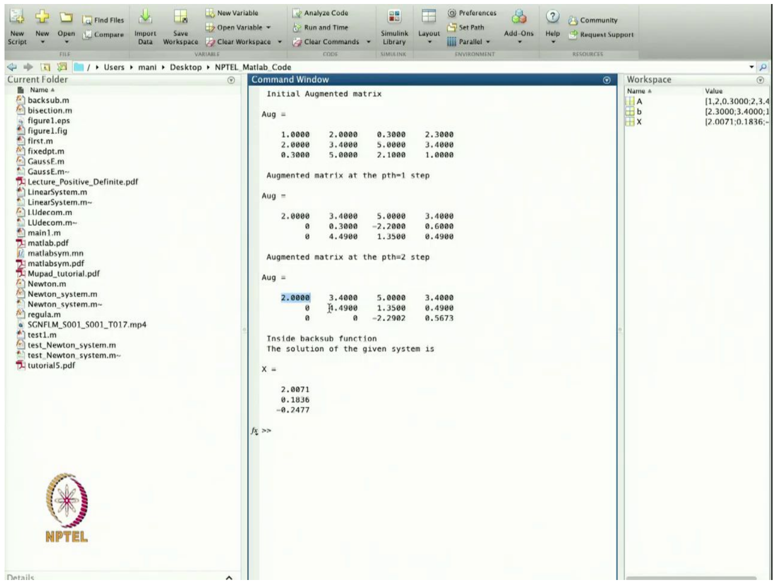


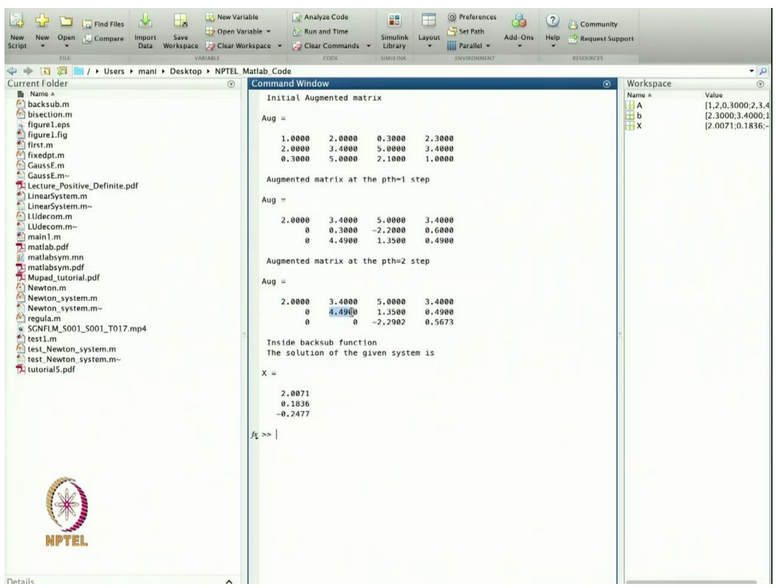


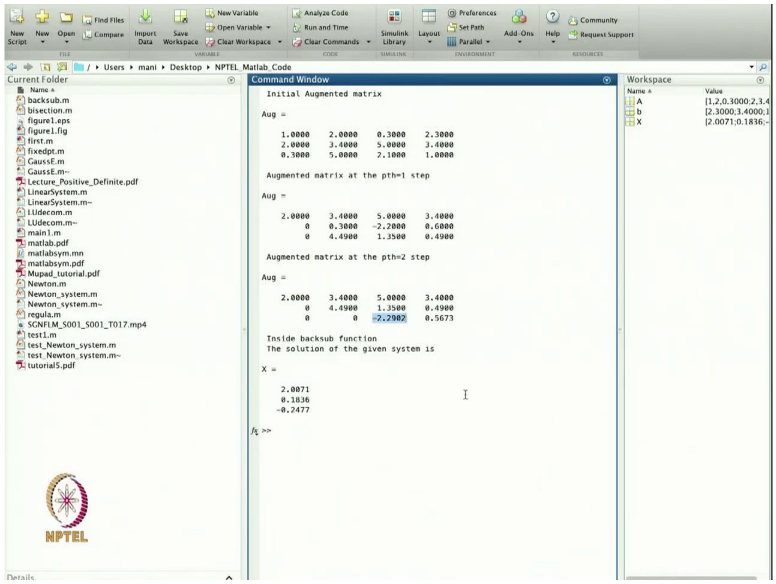


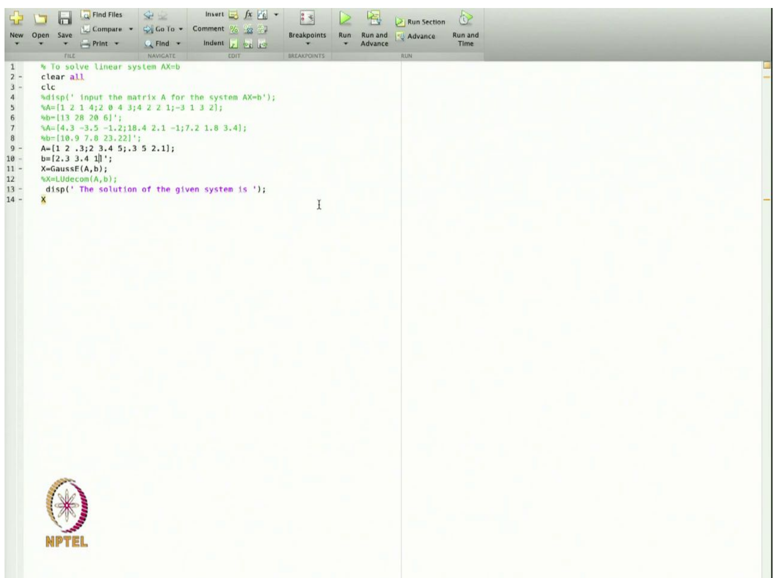
So, that is my matrix now. Now, you can see that this is a symmetric matrix, so 1 2 3, 2 3 5 and then 0.3 0.3 5 5 this one the right hand side, so in this case I start with the augmented matrix. And then it gives me this upper triangular matrix and based on this one that is my solution.

(Refer Slide Time: 26:12)


Command Window
Initial Augmented matrix
Aug =
1.0000 2.0000 0.3000 2.3000
2.0000 3.4000 5.0000 3.4000
0.3000 5.0000 2.1000 1.0000
Augmented matrix at the pth=1 step
Aug =
2.0000 3.4000 5.0000 3.4000
0 0.3000 -2.2000 0.6000
0 4.4900 1.3500 0.4900
Augmented matrix at the pth=2 step
Aug =
2.0000 3.4000 5.0000 3.4000
0 4.4900 1.3500 0.4900
0 0 -2.2902 0.5673
Inside backsub function
The solution of the given system is
X =
2.0071
0.1836
-0.2477
>>
Current Folder
Workspace
NPTEL


Command Window
Initial Augmented matrix
Aug =
1.0000 2.0000 0.3000 2.3000
2.0000 3.4000 5.0000 3.4000
0.3000 5.0000 2.1000 1.0000
Augmented matrix at the pth=1 step
Aug =
2.0000 3.4000 5.0000 3.4000
0 0.3000 -2.2000 0.6000
0 4.4900 1.3500 0.4900
Augmented matrix at the pth=2 step
Aug =
2.0000 3.4000 5.0000 3.4000
0 4.4900 1.3500 0.4900
0 0 -2.2902 0.5673
Inside backsub function
The solution of the given system is
X =
2.0071
0.1836
-0.2477
>>
Current Folder
Workspace
NPTEL





So, in this case, it is a symmetric matrix and in the symmetric matrix if you see these are the pivots 2 4.49 and minus 2.29, so based on this one I can say that this matrix is symmetric matrix and its pivots has two positive pivots and one is a negative provides, so it's eigenvalues are mixed signed. So, its eigenvalue is always also positive as well as negative. So, a mixed type of eigenvalue it has.

So, that way we are going to discuss today what is the positive definite matrix and this one. So, based on this one we are able to solve any matrix so using the Gauss elimination method, so in the next lecture will go further about this Gauss elimination method to the other method that is LU decomposition and other one. So today we will go. We should stop here and maybe in the next lecture we will go to define the other methods, so thanks for watching, thanks very much.