Acoustics & Noise Control

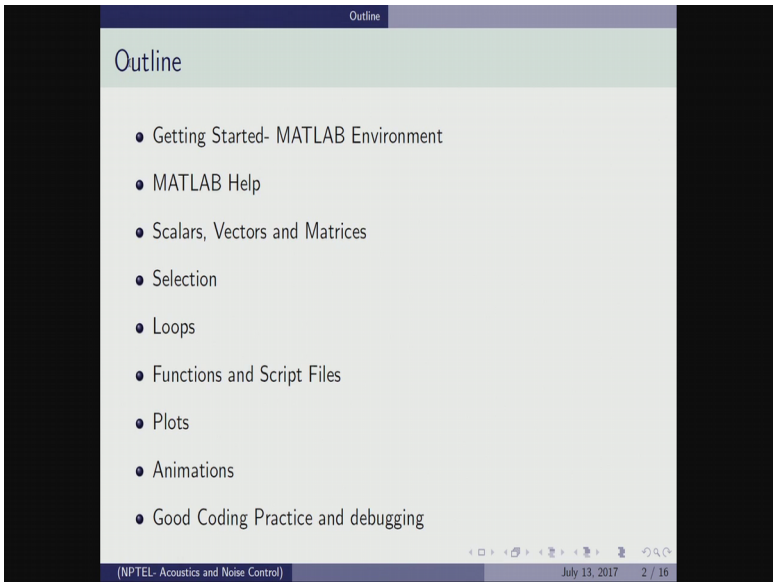
Dr. Abhijit Sarkar
Prof. Ajinkya A Baxy
Department of Mechanical Engineering
Indian Institute of Technology, Madras

Lecture – 39
MATLAB Tutorial – 1

Hello all I am Ajinkya A Baxy, TA for acoustics and noise control and I welcome you all to this MATLAB tutorial. This tutorial is meant agent introduction to the wonderful programming language MATLAB. In this tutorial we will go through some of the basic topics in MATLAB that will be sufficient for you to solve the assignments. If you have gone through the course material that has been uploaded, you will notice that we are going to deal with different kinds of waves like standing wave, propagating waves, something called as evanescent wave etcetera.

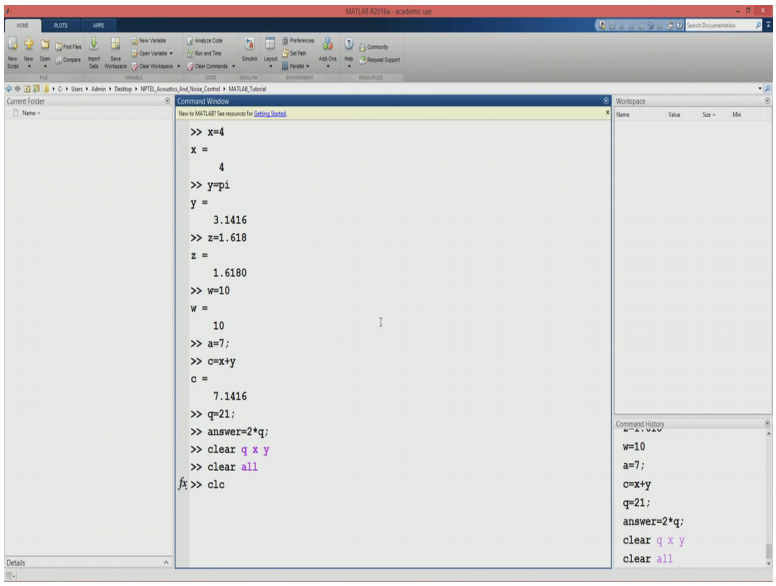
So, will not be better if we can actually see what a wave looks like how it propagates what happens if that propagating wave meets an obstacle. So, it will be better if we clawed the wave ourselves as it will not only help in visualization, but also it will cement our understanding regarding the physics. So, why MATLAB? Well MATLAB is a special purpose programming language with that can be used to solve problems involving numerical computations. MATLAB is easy to learn it is versatile and it as a excellent graphics package it makes programming fun and easy.

(Refer Slide Time: 01:53)



So, let us enter the fascinating world of MATLAB. So, before we actually diving to MATLAB we will get acquainted with some of its concepts. Firstly, we will see what all those windows are which appear when we open MATLAB then we will see how to get help from MATLAB when we are stuck in programming. Then we will see thus building blocks of computation a in MATLAB what are scalers, what are vectors and matrices how to define them and all. After that we will start the actual programming using if a statement that is selection then we will see how to use loops, then we will also write something called as function files and script files. Towards the end we will plot benefactors and we will see how we can use those plots to actually animate things so that we can actually see what is happening. We will conclude is tutorial with some discussion on good coding practices and debugging. So, once you open MATLAB you will see many windows for example, here we can see command window. So, this is the main window in MATLAB in the sense that we type all the commands here.

(Refer Slide Time: 03:04)

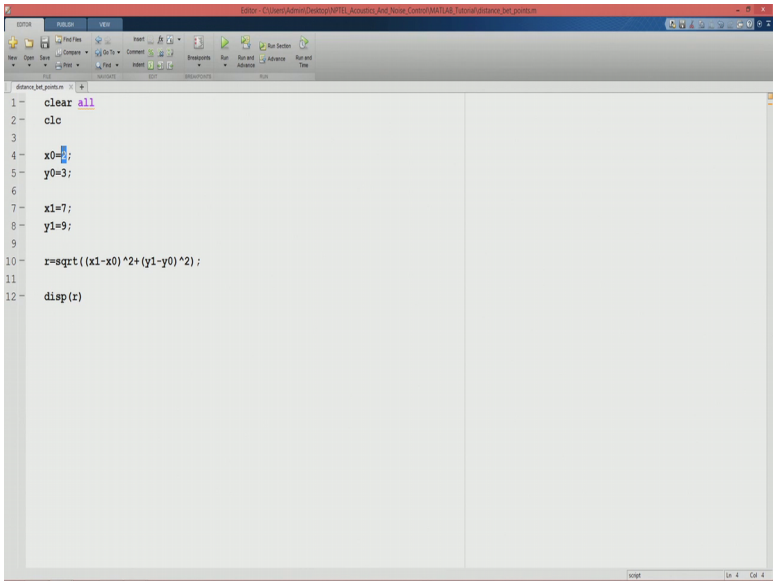


So, let us see x is equal to 4 and enter. So, what MATLAB does here is that, it creates a variable x and assigns it a value 4. We need not declare x to be an integer variable initially, MATLAB will automatically assume it to be an integer. So, similarly we can define many other variables for example, let us defined y to be pi and z to be let us say 1.618 and w to be 10. So, one thing you observe here is that when press enters, MATLAB will print the variable and its value. Suppose if we want to surprise this output, to achieve that we will put a semicolon at the end of the line. So, for example, a is equal to 7 and a semicolon. If we press enter now MATLAB will assign a 7, but it has not displayed the output. Command window can be also used as a calculator. So, let us say we want to add 2 numbers we will add x and y and store the result in c. So, c is equal to x plus y.

So, this is what we get, we will get a edition of x and y let us define some other variable q to be 21, suppress the output and will define a variable answer to be 2 into q. So, we are multiplied a scalar and store the value in variable answer. So, now, coming to the workspace, the workspace will list and display all the variables that having created in the command window. So, we can see here that we have a variable names and their values, suppose we want to see much more details that we can do by doing right click on the panel and selecting what we want to see. Here we can select the size and let us say what is the minimum value and maximum value. So, workspace can also be used to edit the

variables. Suppose I want to edit a currently a has value 7, if I want to edit a to let say ten. So, I will do it this way, I will click a press enter and it will show me a table.

(Refer Slide Time: 05:48)



So, there its value I will select 7 make it 10 and press enter. So, here we can see that a has been updated to value 10, we can even delete some variables from workspace. Suppose we want to delete z, we will do this by selecting z press delete MATLAB will from for confirmation. If you say MATLAB will delete that variable, we can achieve this result from command window also, the command for that is clear suppose we want to delete q x and y. So, how we do it clear space q space x space y. Now observe the workspace as a hit enter, you can see that the assign variables have been deleted. Suppose I want to clear the workspace completely.

So, instead of typing all the variables space x space y space a space answer, we will just try to clear all and work space will be gone or there is one more command to clear the screen that is c l c, if I press enter the command window is clear. Just take a note that c l c will not delete the variables, c l c will just clear the command window. Now we will go to command history; command history will keep record of all the commands that have been typed in command window. Suppose you want to execute this a equal to 7 once again, we can do this by double clicking on this command and it will get executed as you can see in workspace a has been assigned value 7. Now this current folder will store all the files that have been created in this working directory, essentially the current folder is

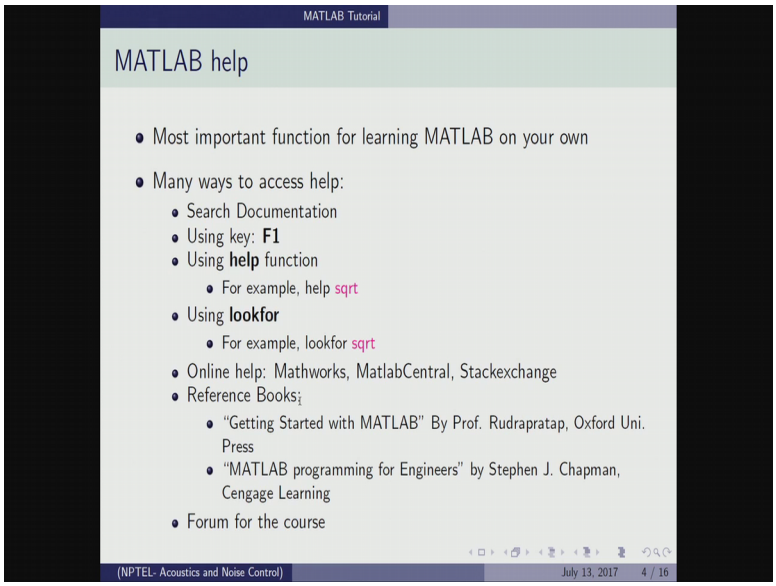
your working directory, you can create new folder share delete few files if you want. If you want to change the current directory, we can do this by using the drop menu here and selecting the path that we want.

Suppose for some reason the path is not listed there we can always use this browse icon browse for folder, we can click on this and then we can select whatever path we want and the working directory will be change to that. Now we will come to the editor window it looks like this, show the editor window is used to write script files and function files, for now we will take a peek at script files and we will save function files for later. So, it may happen that our program contents let us say 10 15 lines and we have to run the program daily for sometimes, and we are seen that we can do that by writing each command one by one in the command window, but it is not feasible right. So, what will do, we will write all those exact commands in a script file save it as a dot m file. Dot m being extension for MATLAB files, then whenever we want to run the program we will open that script file and we will just say run and it will run it.

So, let us do an example, suppose we have 2 points and we have to find the distance between them. So, we will start by clearing the workspace clear all, then we will clear the command window same we will define our variables let us say x naught is 2 why not is 3 x 1 is 7 and y 1 is let us say 9, and let us say r is the variable where the distance will be stored. So, formula for distance is square root. In MATLABs the function s q r t will compute square root. So, the formula is x 1 minus x naught square plus y 1 minus y naught square and a semicolon. Now if you want to display r we will use the function this disp d i s p r. So, this is our program this is a script file, now we will save it let us say we will call it distance between points and press enter.

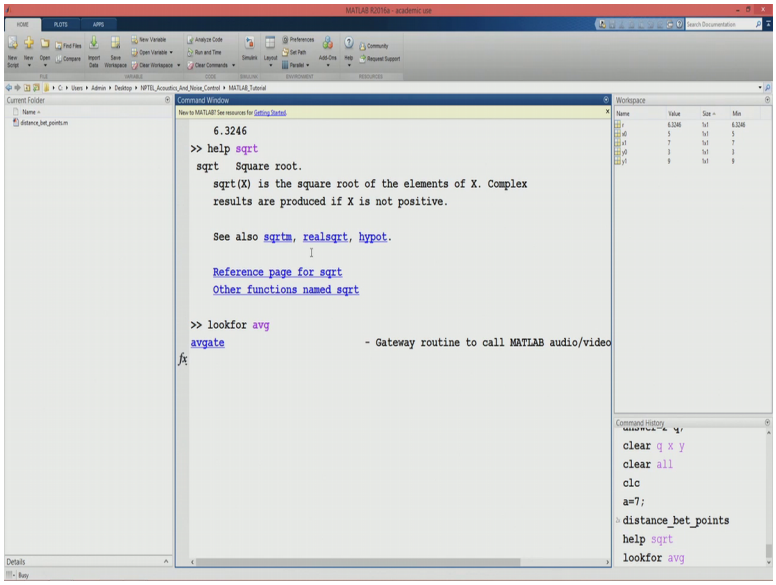
Now we can see observe in current folder we have a filename distance between points. So, this is our working directory and we store save the file there. So, it appears here, now if we run this program by this run button, run and let us see command window. So, MATLAB has calculated the distance to be this. Now suppose someone says change x not to five and. So, we will just change x naught to 5 and will run the program again and the distance has changed. So, if this has to be done in command window, we would have to type everything again and again and again. So, writing script file essentially simplifies programming before we dive into MATLAB.

(Refer Slide Time: 12:04)



Let us see how we can get help if we are stuck during programming MATLAB documentation itself provides a great help. So, since we have seen the square root function, let us see how we can search that in documentation. If you go to the main window on the right hand topside, you can see search documentation. There if we write s q r t MATLAB will show me the functions and search suggestions, we can select what we want. We can search the documentation from command window also for that we have to press f 1 and there we can follow the same procedure that we followed here. Another way to get help without searching documentation is to use the command help.

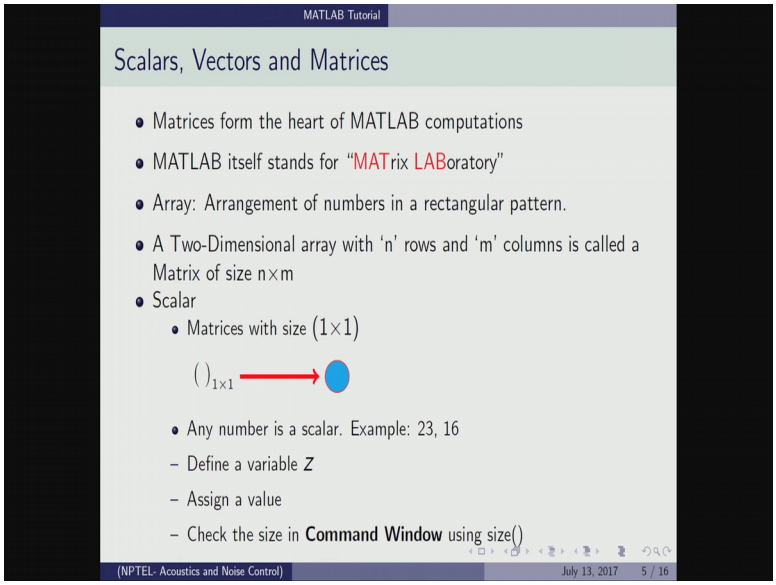
(Refer Slide Time: 13:09)



So, how to we do it is h e l p help space s q r t and we need press enter. So, here MATLAB will show the in information regarding square root function, it will also suggest few other functions. It may happen sometimes that we remember certain keyword from the function, but not the entire function. So, in that case the help command is not useful. So, for that we will ask MATLAB to look for something. So, let us we want we just remember the word a v g, but we do not remember the entire function. So, we will say MATLAB look for a v g and press enter. So, MATLAB will search for these keywords enter documentation and it will print all those functions which contain a v g.

So, we can see here that it has found one and it has printed. So, it may list maybe around 3 or 4more functions which contain a v g, and once it has listed we can select what we want let us clear the screen and move ahead. We have seen look for online help is also great for MATLAB you can browse through the blocks of math works, you can visit forums like MATLAB central stack exchange there you can find very good help, there are also number of books which are written on MATLAB. Particularly this getting started with MATLAB by Professor Rudra Pratap is a very nicely written and a fantastic book it is easy to follow and of course, you can use the course forum to get help.

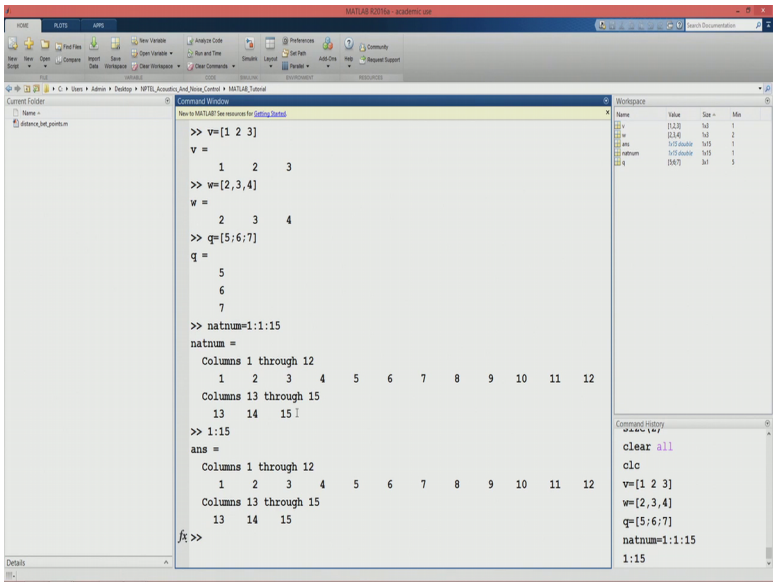
(Refer Slide Time: 15:08)



So, now let us enter the basic building blocks of MATLAB. Matrices are the at the heart of MATLAB computations, they occupy a very special place here. MATLAB indeed was

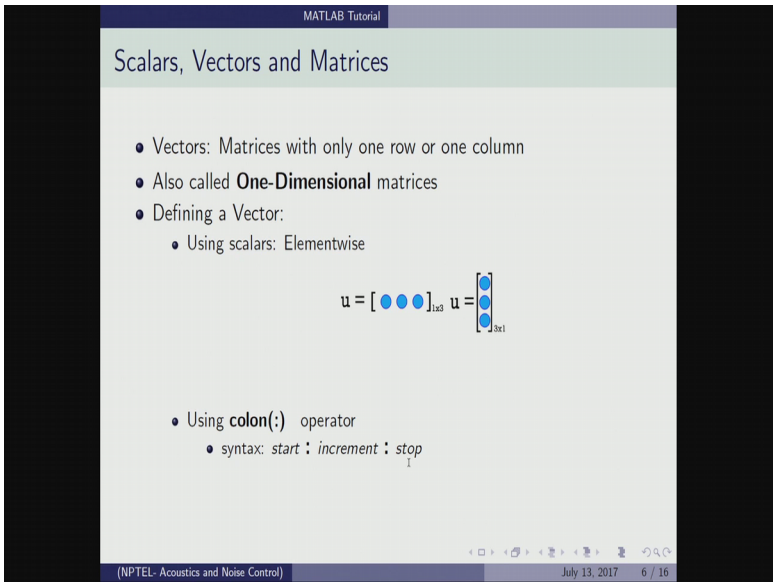
designed to operate on matrices and if you no MATLAB itself stand for matrix laboratory. So, in different programming language you may have come across the term. So, a 2 dimensional array with n rows and m columns is called a matrix of size n cross m. Scalar is basically a matrix with one row and one column, that is it has size one cross one. So, any number is a scalar for example, 23, 16 any number will define it assign it to value and will check it size.

(Refer Slide Time: 16:14)



So, let us say we will define z equal to 17. So, if we see workspace z has size one cross one it is a scaler, we can check size by using function size parentheses z close parentheses and it will show one cross one.

(Refer Slide Time: 16:43)



Now let us come to vectors, vectors are one dimensional arrays and these are matrices where one of the dimension is one that is vectors are matrices with only one row or one column. If it has only one row we call it a row vector if it has only a one column we call it a column vector. So, how do we define a vector? First let us see using scalars. So, let us say we will first clear the workspace clear all and command window 2. So, let us say we have to create a vector 1 2 3.

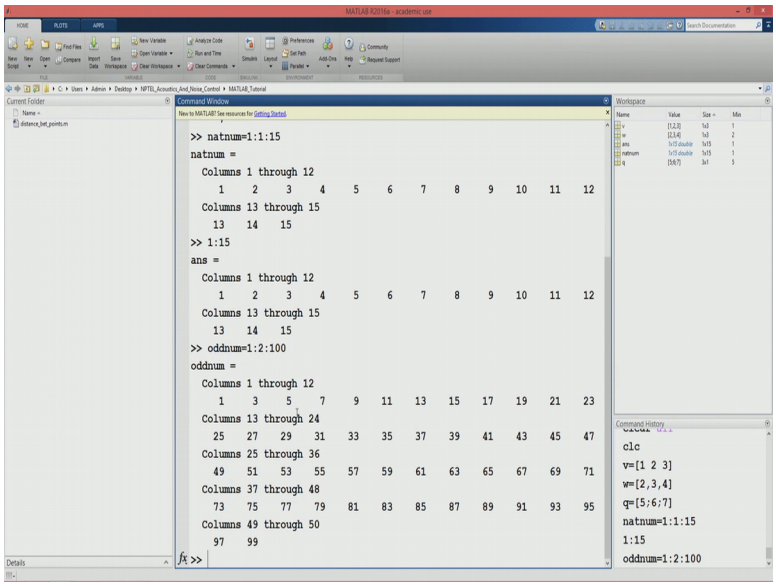
So, we will say v equal to square bracket. So, observed that we have to indicate in 2 MATLAB start of a vector by a square bracket, this will tell that now we (Refer Time: 17:41) defining a vector. So, now let us enter elements let us say 1 space 2 space 3 and we will indicate the end of a vector by a closing square bracket, and if you press enter MATLAB will create a row vector you can check size in workspace 1 by 3 which has elements 1 2 and 3. We can also a separate this elements with comma let us say w starting with a square bracket 2 comma 3 comma 4. This will again create a row vector with elements 2 3 4. So, if you separate 2 variables by a comma or a space, MATLAB will put a them in the same row. Now if we want to generate a column vector, we can do it this way let us say q is equal to 5 semi colon 6 semi colon 7. So, MATLAB has generated a column vector check size in workspace its 3 cross 1.

So, this semicolon will tell MATLAB to begin a new row. Now suppose that I have to create a vector of first 15 natural numbers. So, how do we do it? One way to do it is to

type all the 15 natural numbers with space or semi colon, smart way to do it is using a colon operator. So, what actually is a colon operator let us see. Colon operator has syntax you have to specify add element, you have to specify increment how much MATLAB will increase with each ton, and you have to indicate where MATLAB should stop. So, let us generate our first 15 natural numbers using colon operator. So, let us say natural numbers is one colon increment by one colon stop at 15. So, here we can see MATLAB has generated a vector with size 1 cross 15, and it begins at one and n z 15. Observe one thing here unlike the previous case we have not enclosed this in a square bracket.

This is. So, because colon operator is defined to generate a vector. So, we did not enclose it in square brackets, also if we miss the increment. Suppose if I miss increments if I just say one colon 15, MATLAB will by default assume it to be 1. See now let us say we have to we have to create a vector of odd numbers between 1 and 100. So, we will again use colon operator, but in this case the increment will not be 1 it will be 2. So, let us say odd num is equal to 1 increment by 2 and stop at 100.

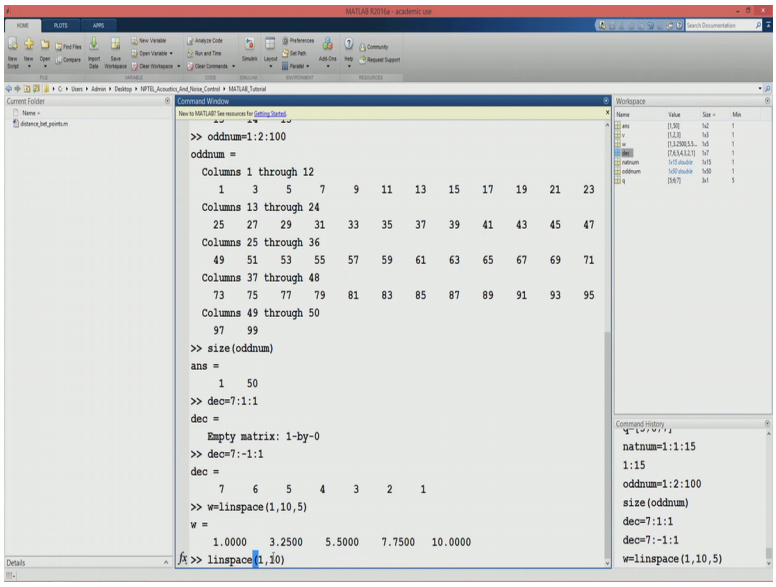
(Refer Slide Time: 21:07)



So, MATLAB as printed all the odd numbers between one and 99. So, what happens here is that MATLAB will select first number 1 at 2, it the edition will be 3. Now MATLAB will check if 3 is less than or greater than 100, if it is less than it will continue the operation this happens until MATLAB reach at 99. So, what happens at 99? MATLAB will add to 2 it we will get one naught one MATLAB will check whether one naught one

is greater than or less than 100. MATLAB see that it is greater than 100 and it will stop at 99, that is why we get a vector which ends at 99 and not at 100. We can check the size of this vector size parentheses oddnum it is a row vector of size 1 by 50.

(Refer Slide Time: 22:09)



Also till now, using the colon operator we have generated all the vector that are in increasing order, that is we begin with a small number and we end up at a big number, can we create vector in the reverse order? Well yes let us do it.

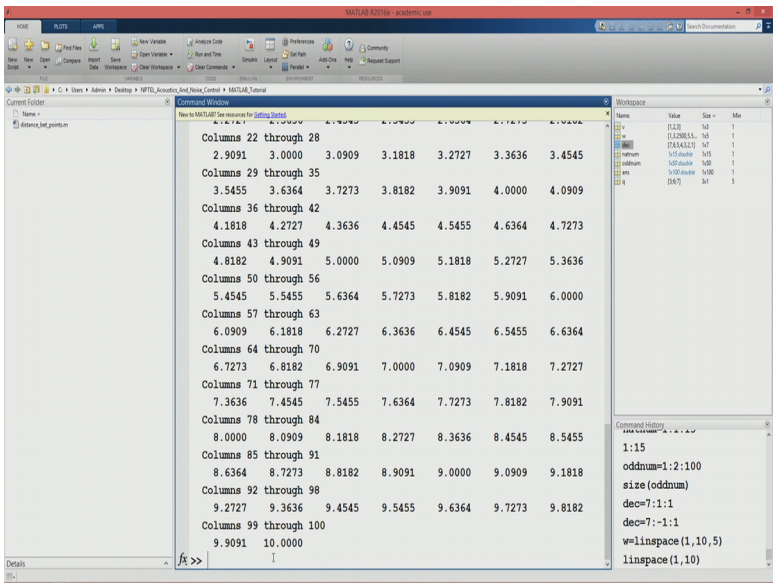
Let us say we define a vector which will begin at 7 decrease by one and it will end up at one. So, let us define our vector to be d e c for decrement, and let us say 7 begin at 7 increment one and end will be 1. Now what happens if we press enter? MATLAB as return and empty matrix why again MATLAB will take first element 7, it will add one to eight and it will check the end condition is 8 greater than or less than1. Since the condition has failed MATLAB will stop at initial stage itself, and will return us and empty matrix. You can see in workspace the vector which we planned with name d e c it has value 0, not even 0 it is empty. So, we will rectify our mistake like we have to tell MATLAB increment is not one, but increment is minus 1.

So, we will correct or mistake now, now if you want to access browse through the previous commands in the command window, we can use the up arrow. Like here I can go to any command which I have type previously, since it is just the first command I will just try to press once and then I will go to increment section, I will put minus 1 and let us

say. So, MATLAB has now displayed and defined an array d e c which starts at 7 ends at 1 and decreases by 1 at each time; sometimes it happens that we want we want a vector we know the start point of vector, we know the end point of vector and we also know how many points are there in between these 2 points, but we do not know by how much we need to increment to get to that points.

So, in that case we can use one more function called as linspace, which stands for a linear spacing. So, let us say we want a vector which starts at one end at 10 with 5 points in between them. So, we will do it in this way, let us say our vector has name w. W is linspace MATLAB will prompt us help, it ask for first number let us say 1 comma second number is 10 and we need 5 points. So, will write 5 close and enter. So, you can see that w is a vector with 5 points suppose we miss that n number suppose we just say linspace 1 comma 10.

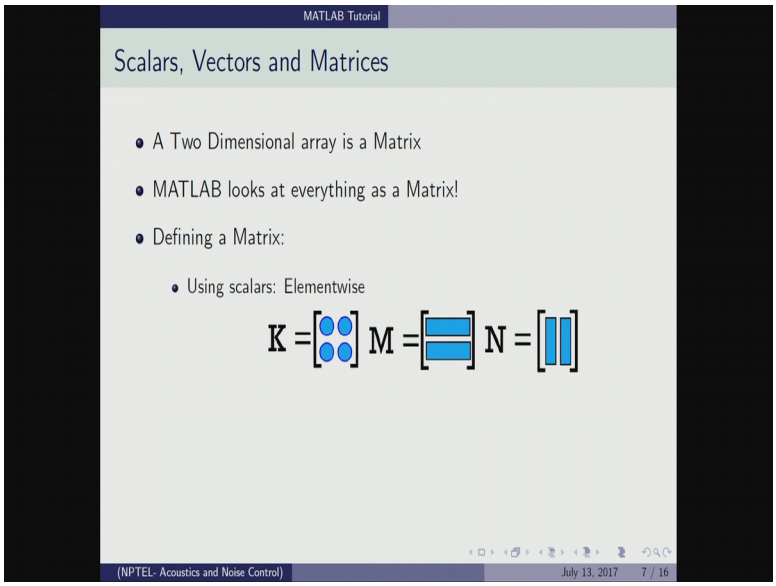
(Refer Slide Time: 25:40)



So, what happens in this case? MATLAB will by default assume and to be 100, it has given me a vector of size 1 cross 100.

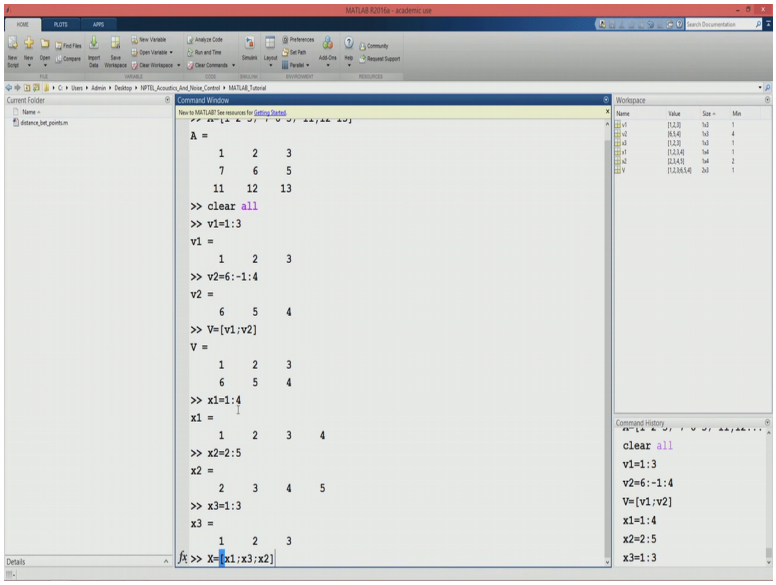
So, we have seen this, this complete the vectors now let us enter the matrix we have already seen that matrix is a 2 dimensional array.

(Refer Slide Time: 26:01)



And MATLAB looks everything as a matrix, even a scalar for MATLAB is a one cross one matrix and vector is maybe n cross one matrix or one cross m matrix. So, let us see how we will define matrix. Firstly, again we will start using scalars, our matrix is a and we have to create a matrix of size let us say 3 by three.

(Refer Slide Time: 26:43)

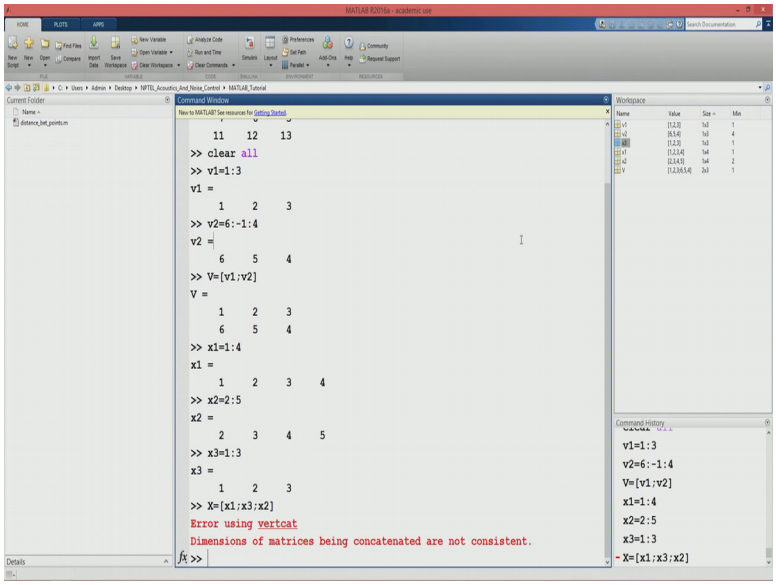


So, will say 1, 2, 3. So, space will ensure that MATLAB will put this digits in a same row, now if we now we want to tell MATLAB to begin a new row. So, we will press a semicolon and will start our new row. Let us say 7 6 5, again we will press semi colon to

tell MATLAB start a new row. Let us say 11 comma 12 comma what say space 13. Does not matter if you combine commas and space, MATLAB will still put them in the same row; we will close indicate the end of matrix by a closing of square bracket replace enter, and MATLAB has define a matrix with these elements and the size is 3 by three.

So, we can also define a matrix using 2 vectors, let us say we will first clear the work space again. So, let us define our vector v 1 to be 1 use colon operator since we have learnt it, and let us say v 2 to be 6 minus 1 and 4. Now let us say we want to create a matrix using v 1 and v 2. So, let us a call over matrix capital V and let us say the first row is v 1 and now we have to tell MATLAB that v 2 should occupy a new row. Same thing again you will put a semicolon type v 2 close the bracket and we have generated v matrix you can check size always in workspace, the size is 2 cross three. Now let us do one more example let us see what MATLAB does in this case, will define one to be 1 2 4 and let us say x 2 is 2 to 5 and x 3 is 1 2 3. Now if you want to create a matrix using these 3 vectors let us see, x is first row will be x 1 let us say next row maybe x 3 and next row and final row will be x 2.

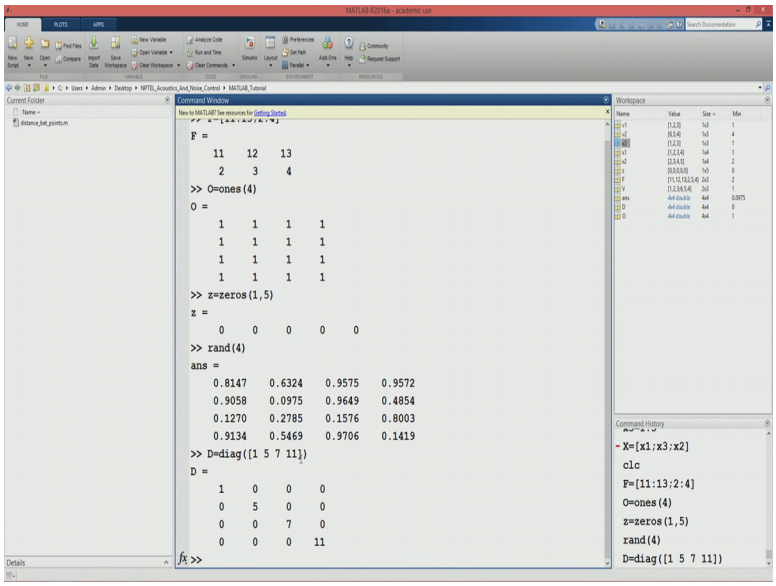
(Refer Slide Time: 29:22)



So, MATLAB has given us and error, what it says that the dimensions of the matrices that are being concatenated are not consistent, you can actually see that while x 1 and x 2 have same number of columns, but x 3 is only a 3 column have column matrix you can even see the sizes of x 1 x 2 and x 3, they are not same x 1 and x 2 or 1 by 4 and 1 by 4

and x 3 is 1 by 3 matrix. So, this thing should be kept in mind that if you want to create a matrix using row vectors the number of columns should be same, and if we are creating a matrix using a column vector then the number of rows in all those matrices should be same.

(Refer Slide Time: 30:22)

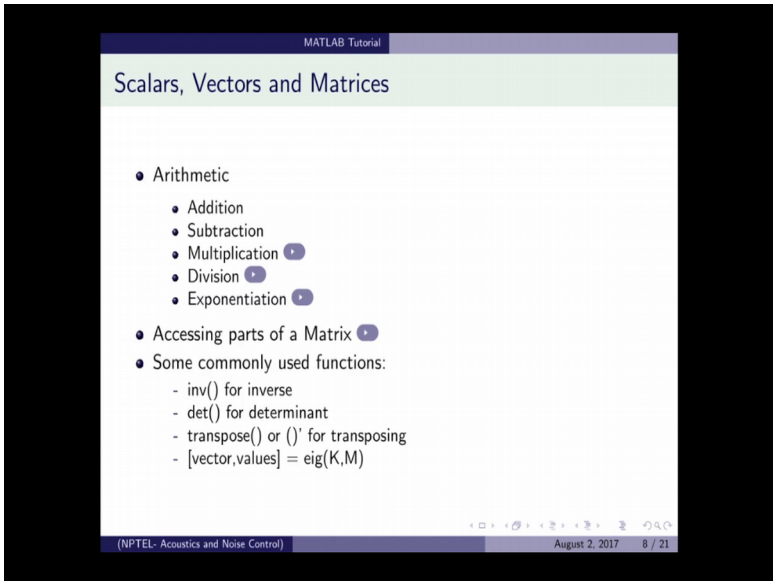


So, we can even a define matrix using colon operator directly. Let us say our matrix is f and we say 11 colon 13 and next row we say 2 colon 4 and let us see what we get. So, if you can even use colon operators to define matrices that we have seen yeah and there are some predefined functions in MATLAB to generate useful matrices.

So, we will see few of them like 1 0 rand and dag. So, what ones does is that it will create a matrix with all elements to be 1. So, let us say we call that matrix 4 and will say ones 4. So, MATLAB will generate a matrix 4 by 4, that will all be containing once. We can even create a vectors using these functions let us create a vector of zeros let us say we call it z zeros, we create a row vector it is a 1 cross 5. And will press enters and we get a row vector with all zeros unit. The rand command will generate a matrix of random numbers between 0 and 1. In this case it will generate a 4 by 4 matrix, and suppose we want to create a diagonal matrix whose half diagonal terms are 0 and we have to and we know only the diagonals terms. So, this command will do it let us D is diag, will have to specify this v as a vector. So, let us say it 1, 5, 7, 11 and will close it. So, we here we can see that daig command has created a 4 by 4 diagonal matrix, since we had 4 elements in

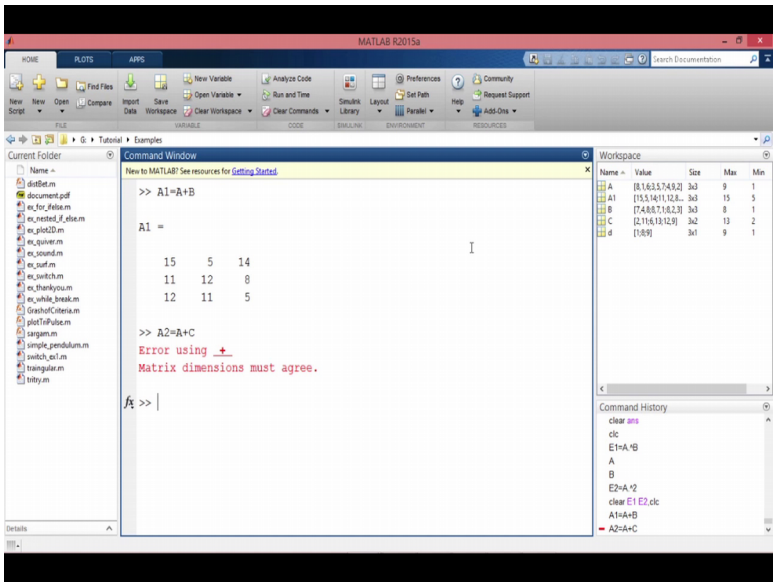
our vector all the of diagonal terms are 0 and the diagonal terms are the terms which have appeared in the vector.

(Refer Slide Time: 32:51)



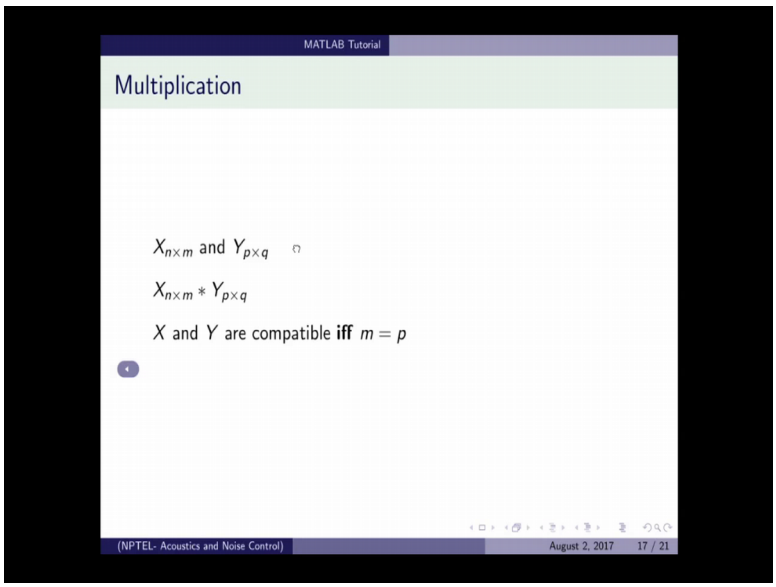
Now, that we know how to define matrices and vectors we will see there arithmetic. In that we will see these following operations addition, subtraction, multiplication, division and exponentiation let us see addition. So, here we have predefined few matrices as you can see in the workspace a is a 3 by 3 matrix, b is also a 3 by 3 matrix c is a 3 by 2 matrix, and d is a 3 by 1 vector. Let us add a and b and store the result in a 1, a 1 is a plus b, now let us add a and c and MATLAB has tots an error.

(Refer Slide Time: 33:35)



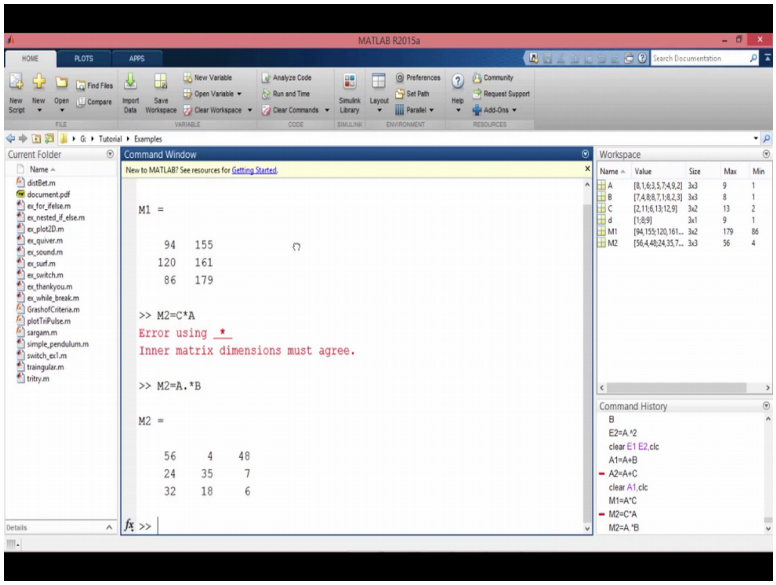
So, this is. So, because while performing addition operation MATLAB will add element of first matrix with the corresponding element in the second matrix, and due to lack of elements in c, a plus c is not possible similarly a plus d is not feasible. Does to add 2 or more matrices their sizes should be same, same goes with subtraction. There are 2 types of matrix multiplications defined in MATLAB, one is the regular matrix multiplication which we learn in linear algebra, other is called array multiplication we will come to that in a minute.

(Refer Slide Time: 34:48)



First let us deal with the regular matrix multiplication. So, for 2 matrices getting multiplied in this way, there either dimension should be same. What I mean by that is let us say we have 2 matrices X of size n cross m and Y of size p cross q, and we have to evaluate X star Y. Now X star Y will healed result if and only if m is equal to p that is number of columns of the first matrix should be equal to the number of rows of the second matrix. So, if that is the case then we say that x and y are compatible, but this does not mean that y and x will be compatible also.

(Refer Slide Time: 35:39)



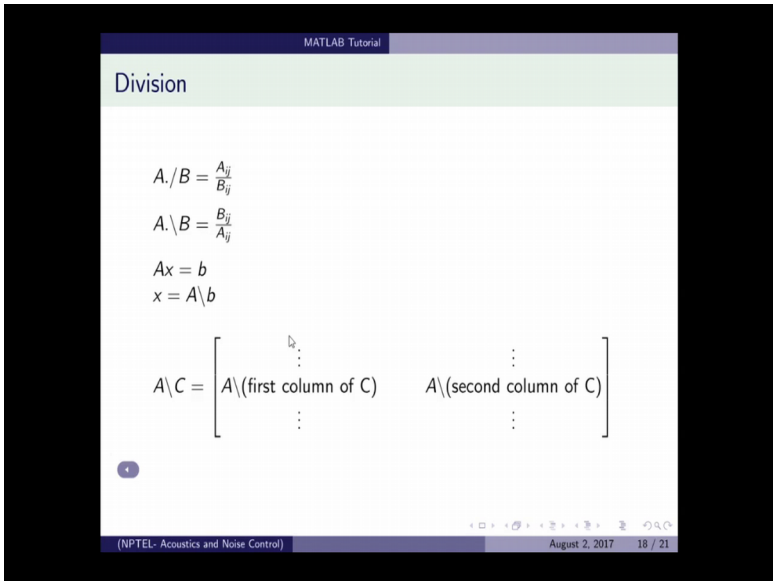
Let us see an example here we will multiply matrix A star C. We know that a is a 3 by 3 matrix and c is a 3 by 2 matrix therefore, there inner dimensions match and A star C should yield result, which it does. Now what happens if I tried C star A since the dimensions are not same we expect MATLAB to give us an error and MATLAB does not disappoint.

Now, let us replace star with a dot star and we will get array multiplication. Let us do an example and see what it gives, and 2 will be A dot star B, A dot star we will result in to this. So, A is this matrix and B is this matrix, observe that each element in the new matrix m 2 is product of corresponding elements of matrices a and b. For example, the first element 56 in m 2 is the product of 8 and 7, which are first elements in matrix a and b respectively, does with dots star MATLAB is doing an element wise multiplication therefore, like edition in this case to their sizes should be same, this dot will tell

MATLAB to perform array operations. To summarize for regular matrix multiplication 2 matrices should be compatible that is there inner dimensions should be same, and for array multiplications their sizes should be same.

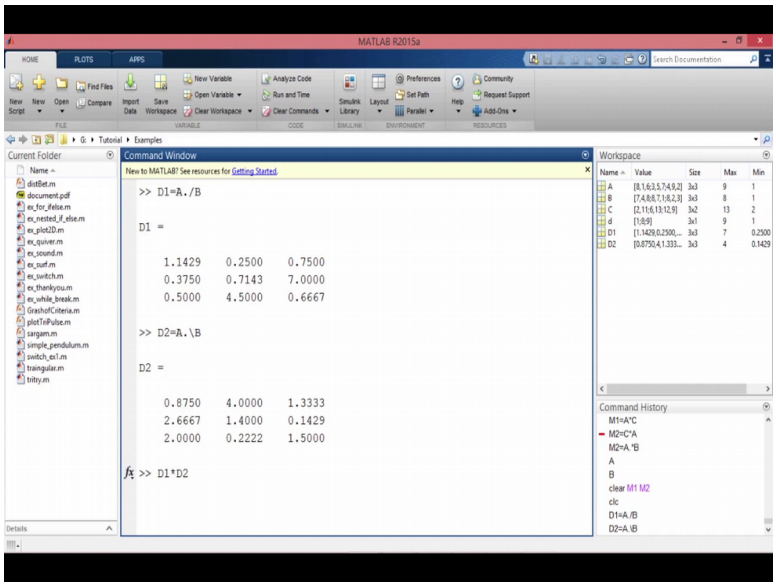
Now, let us come to division first let us see array division. So, along with the regular slash operator MATLAB has one more operator, for a division which is a backslash.

(Refer Slide Time: 37:57)



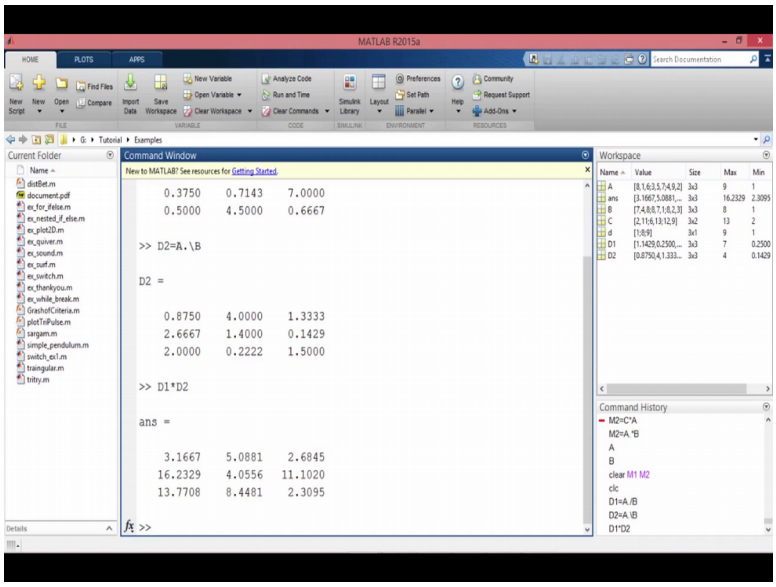
Let us see what is the difference among these two. So, A dot slash B which we also call A over B will be evaluated as A i j over B i j, that is element in matrix A over the corresponding elements in matrix B, and A dot backslash B will be evaluated as B i j over A i j, that is elements in matrix B over the corresponding elements in matrix A. So, are these 2 matrices inverses of each other let us see.

(Refer Slide Time: 38:42)



So, D 1 will store A dot slash B and D 2 will store A dot backslash B. Now if D 1 and D 2 are inverses of each other, the product D 1 star D 2 should yield an identity matrix, but it does not.

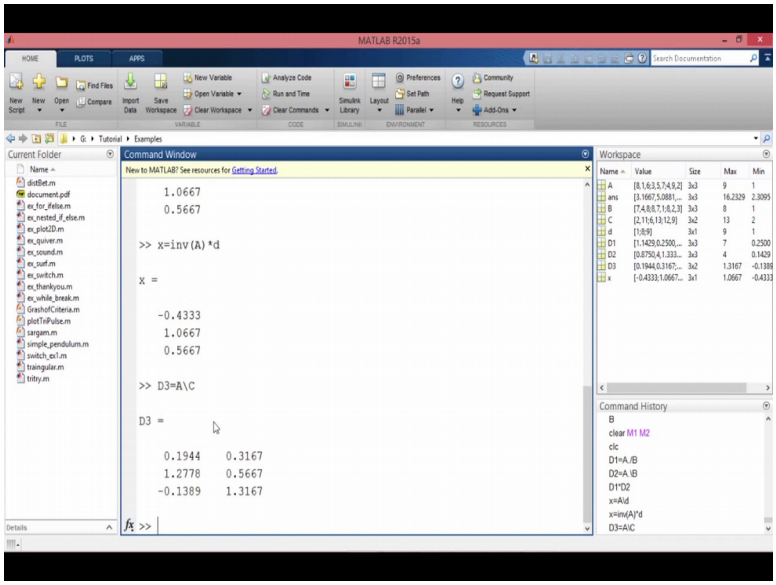
(Refer Slide Time: 39:08)



So, it should be kept in mind that these operations A dot slash B and A dot backslash B will result in matrices whose elements are reciprocals of each other, but the matrices themselves are not inverses of each other. Many a times we have to solve equations in matrix form as A x equal to b, where a is A known matrix b is a known vector and x is an

unknown vector which we have to evaluate. So, we can use the backslash operator to find x here.

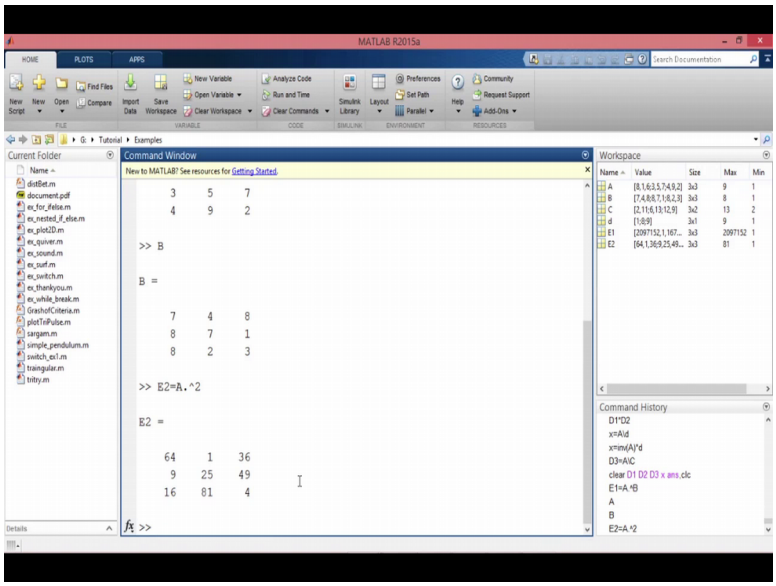
(Refer Slide Time: 40:15)



Let us say x is A backslash d, d is our vector here. We can also use the inverse function inverse of A star d. Now what happens if we do A backslash C does MATLAB evaluate this or will it give us an error. So, let us see, what actually MATLAB has done here is with A backslash C MATLAB is calculating the first column of the resultant matrix as A backslash the first column of C and the second column is A backslash second column of C. So, this backslash operator we can say it is similar to the multiplication in the sense that the sizes of the 2 matrices involved should be compatible.

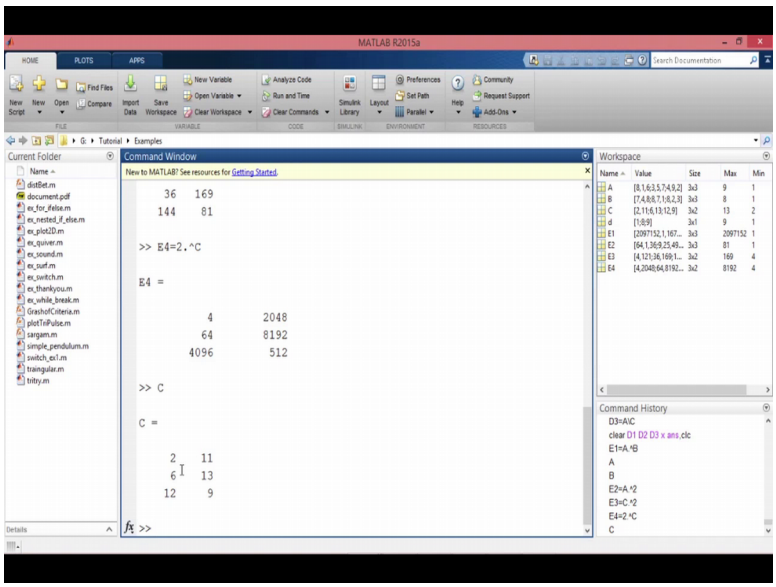
Now, let us come to exponentiation, first we will see array operation. So, when we right A dot raise to B, you know that the operation will yield result if sizes of a and b are same since it is an array operation. And that each element in the base matrix a will be raised to the corresponding element in the index matrix b, let us see a is this and b is this. So, 81 is 9 squared and 8 is 2 cubed, now what happens if the exponent is a scalar.

(Refer Slide Time: 42:28)



So, let us try A dot square and each element in A has been squared, since it is just squaring elements of the base matrix we can even work with a non square matrix or rectangular matrix like C. So, C dot square should also yield result and it does again what happens if we do 2 dot.

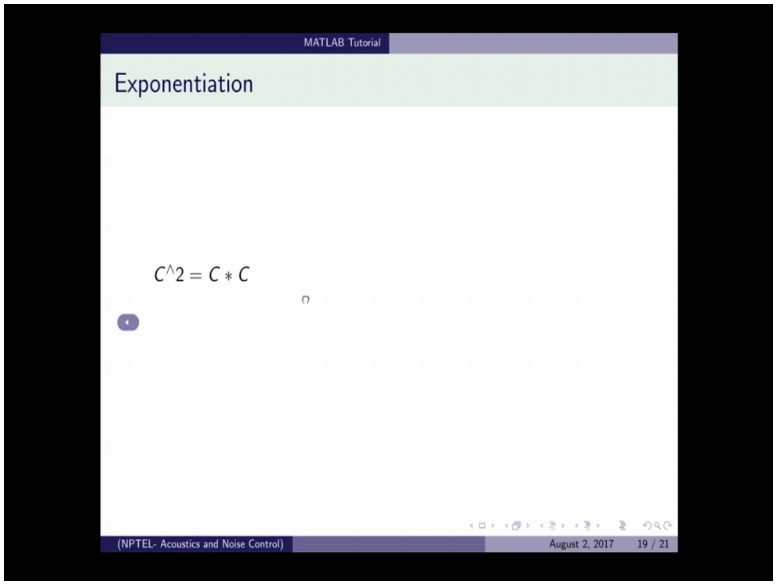
(Refer Slide Time: 42:58)



Just to see what we expect here is and this is matrix c. So, 2 has been raised to each element in the matrix c. So, 2 squared is 4, 2 raise to 6 is 64 and so on. Thus if we are

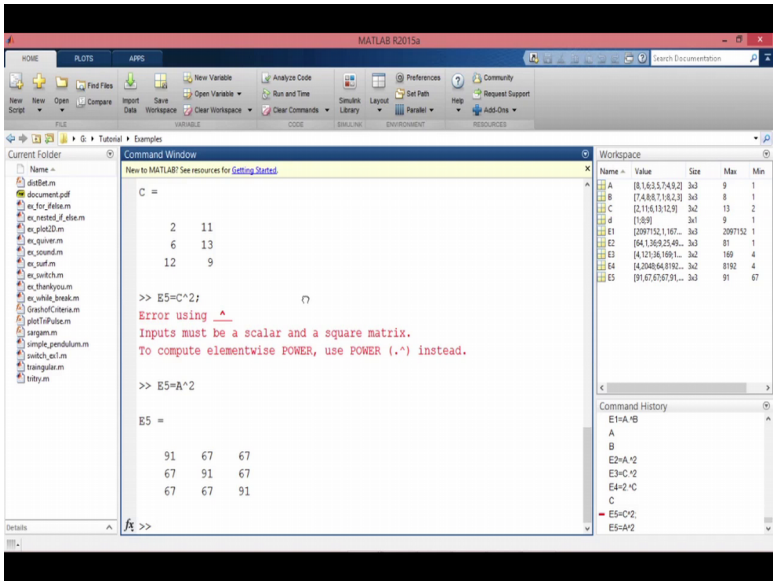
doing array exponentiation and if both the operands are matrices their sizes should be same, but if one of the operands is a scalar the size of the other matrix does not matter.

(Refer Slide Time: 43:47)



Now, what will this we evaluated as? So, C square is basically C star C and we know that for C star C to yield result the inner dimensions should be same, but C is a 3 by to matrix, it is not compatible and C star C will give me an error. In what cases this operations will give me a solution? Matrix x will be compatible to itself if and only it is a square matrix.

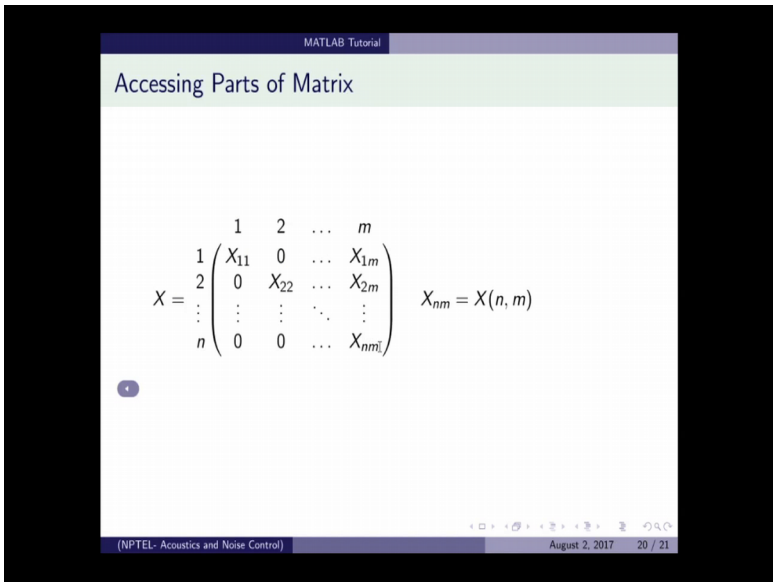
(Refer Slide Time: 44:23)



So, we expect that c square should give me an error which it does, but A square will give me solution as expected. So, to summarize the entire exponentiation operation we see that if we are using array operation and the 2 operands are matrices, there sizes should be same if one of the operand is a scaler, the size of the other matrix does not matter and if we are doing regular exponentiation with scaler, the base matrix should be a square matrix.

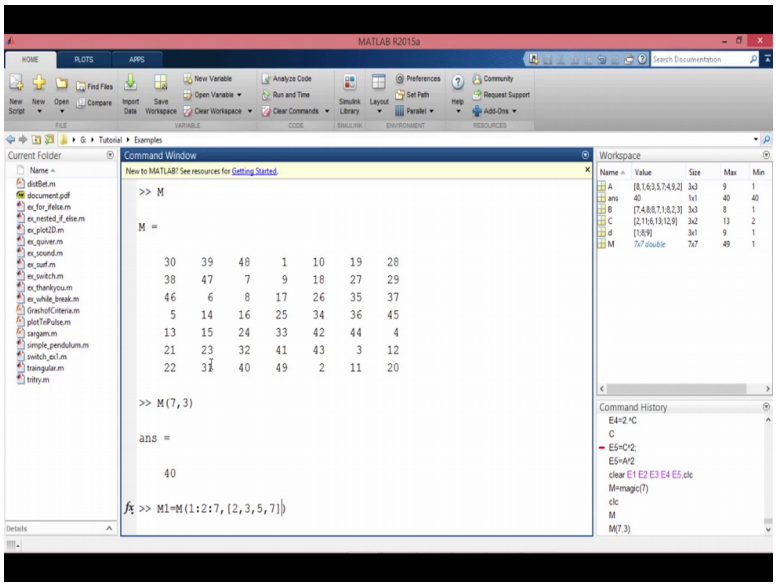
Now, that we are adapt in arithmetic let us see how we can manipulate matrices, how we can extract elements from them, how we can add new elements, how we can replace the existing ones et cetera. So, let us first see how the rows and columns are numbered.

(Refer Slide Time: 45:39)



So, let us say we have a matrix x which contains n rows and m columns. So, the row numbering will start from top to bottom with 1 2 up to n, and column numbering will start from left to right with 1 2 up to m. Let us say we have to extract an element which occupies nth row and mth column. So, to do that we tell MATLAB extract element in x which is in row n, and column m. So, the row numbers always go first and followed by the column number.

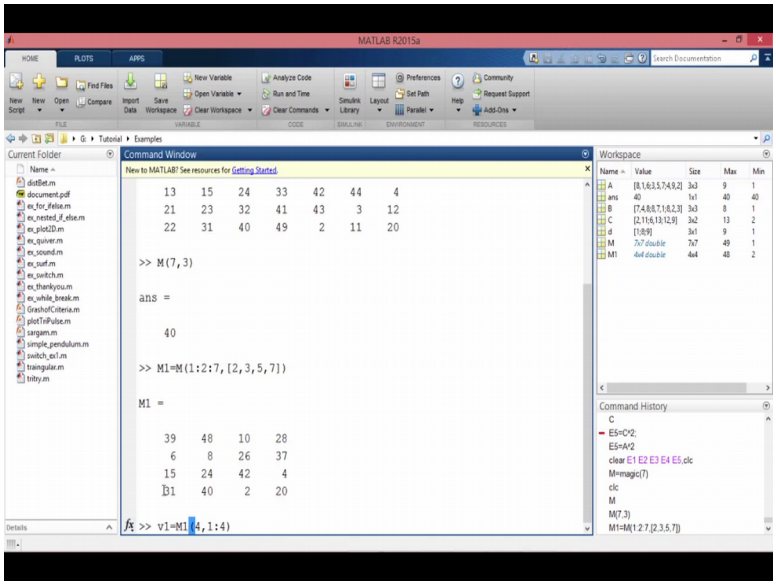
(Refer Slide Time: 46:27)



So, here we have matrix m which is a 7 cross 7 matrix, and suppose we have to extract element which lies in seventh row and third column that is we have to extract this element.

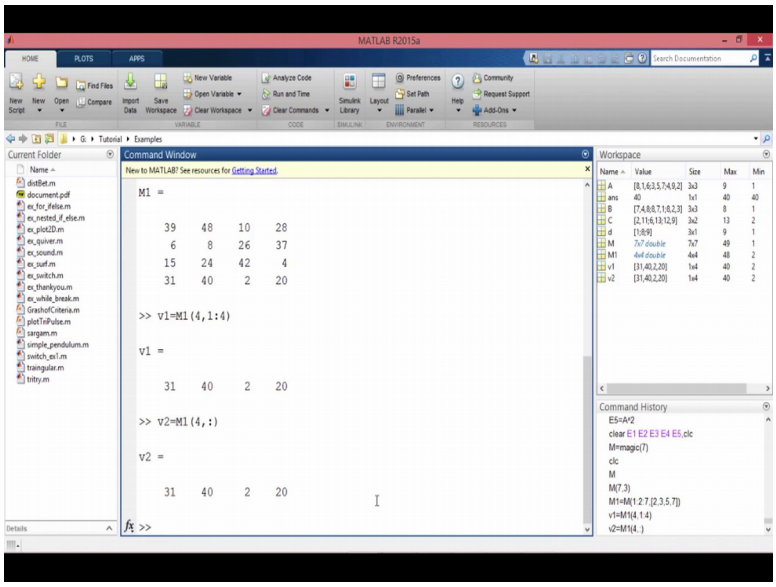
So, how we tell MATLAB is m parentheses, seventh row and third column close parentheses and enter and MATLAB has extracted that particular scaler or element. Now suppose we have to extract a submatrix which is an intersection of columns 2 3 5 7 and rows 1 3 5 and 7. So, how do we do it is? This way.

(Refer Slide Time: 47:48)



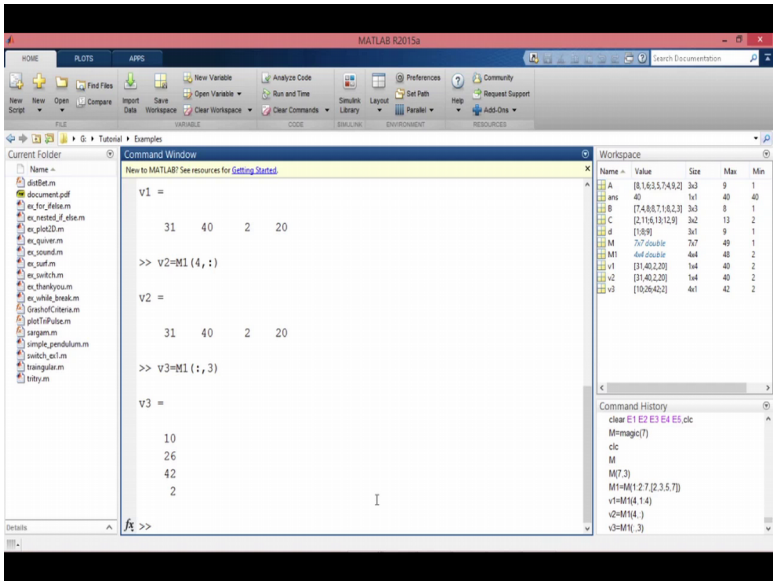
The rows are 1 3 5 7 and we can be smart this this time and we will use colon operator up to 7, and columns will be 2 3 5 and 7. So, m 1 will contain intersection of these columns and these rows. Now suppose I want to extract entire fourth row of m 1. One way to do it is row number 4 1 2 4, another way of doing it is using just a colon.

(Refer Slide Time: 48:29)



So, it will be m 1 of fourth row and all columns. If you place this colon symbol in the rows place MATLAB will assume we want all the rows, now suppose I want to extract column number 3 of m 1 matrix.

(Refer Slide Time: 49:04)

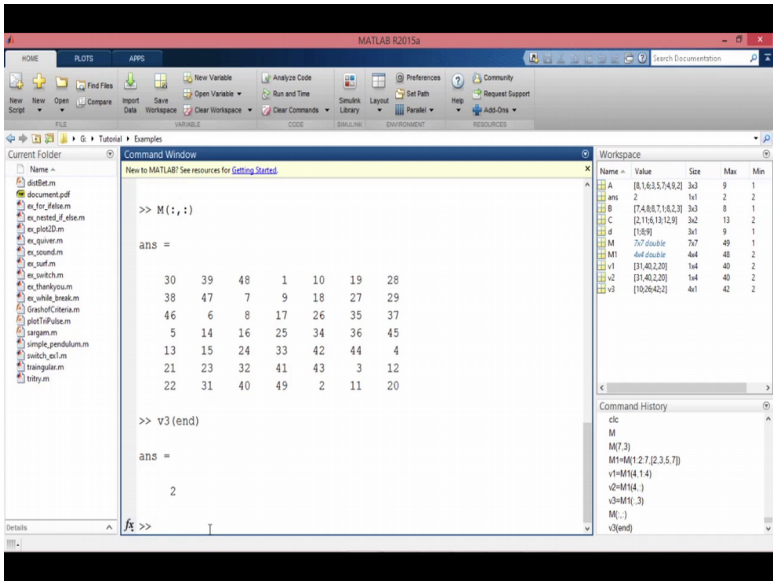


So, how to do this is instead of placing it in the column space, we will say all rows of third column and MATLAB will give me all rows of third column this.

So, it is easy to see what this operation will do i say all rows and all columns of matrix m which is the matrix m itself. There is a handy keyword called end and we can use it quite effectively it may.

Happen in programming that we end up with a huge matrix whose size we do not know and for some purpose we have to extract its last 2 columns. So, how do we access the last 2 columns if you do not know the size of the matrix? So, in such situations end comes into play suppose I write v 3 end what this will give me is, it will give me the last element in vector v 3 which is 2.

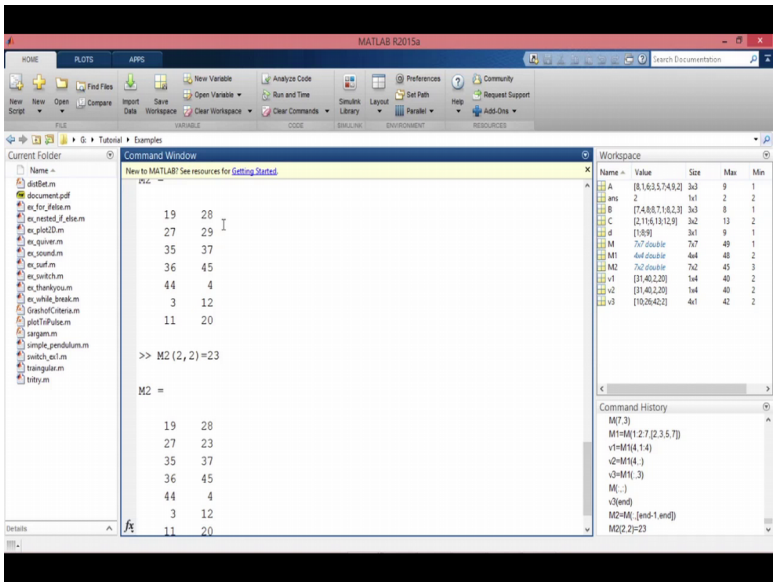
(Refer Slide Time: 50:33)



So, let us see what MATLAB gives and it gives 2. This end will tell MATLAB to give me the last element in case of vector. We can also use this end keyword with arithmetic operations such as addition subtraction etcetera.

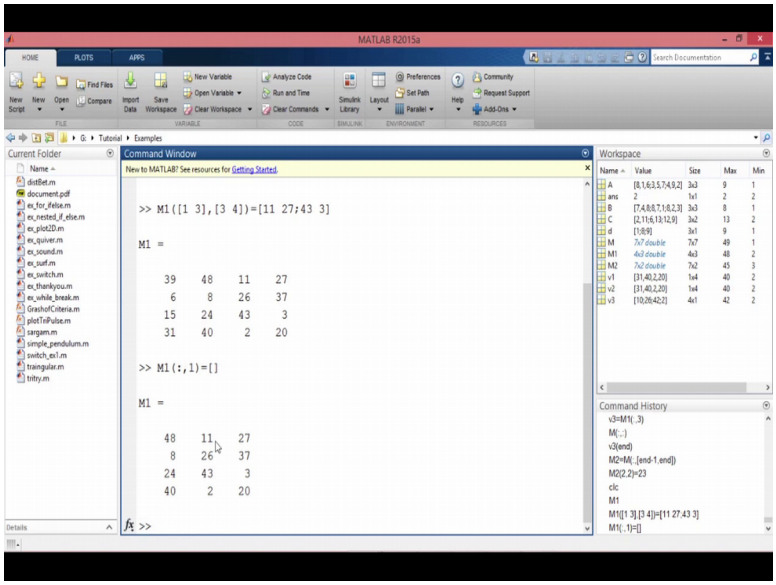
Now, we will try to extract last 2 columns of m using the keyword end.

(Refer Slide Time: 51:09)



So, M 2 will be M we want all rows end minus 1 comma end MATLAB will extract the last 2 columns of matrix M. We can even replace elements in the matrix suppose you want to replace this 29 with 23 in matrix M2. So, we tell MATLAB to replace entry in the second row and second column by 23 and as you can see MATLAB has updated entry 2 comma 2.

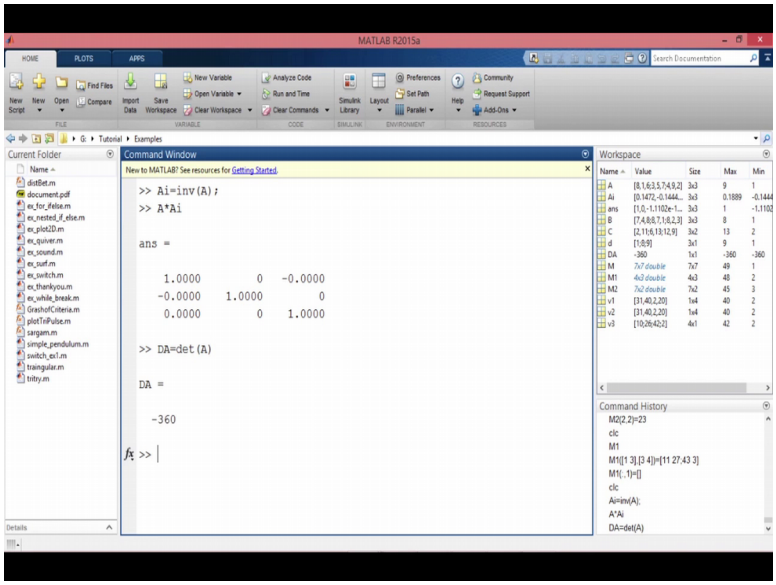
(Refer Slide Time: 52:11)



Now, suppose I have to replace a sub matrix which is 1 3 1 4 3 3 and 3 4. So, again it is actually a straight forward thing you will tell MATLAB to replace rows 1 3 and columns 3 4.

By let us say 11 27 43 and 3 and as you can see here intersection of row 1 and 3 and column 3 and 4 has been replaced by 11 27 43 and 3. Now if you want to delete let us say first column of m one matrix that we do by setting all rows of first column in matrix M1 to n empty square bracket. So, as you can see here the first column has been deleted. So, here are few commonly used functions inverse will give me inverse of the matrix determinant will give me the determinant, transpose will give me the transpose let us illustrate this with examples.

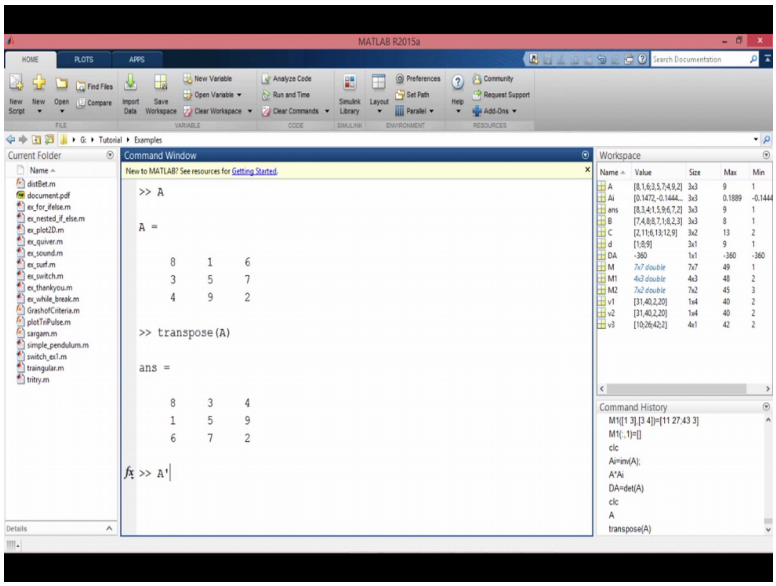
(Refer Slide Time: 54:12)



So, inverse of a will give me A inverse we can verify this by doing A star A inverse which is an identity matrix.

Now, determinant of A will calculate determinant, which is minus 360.

(Refer Slide Time: 54:51)



So, A matrix is this, and transpose A will transpose the matrix similarly A prime will also transpose the matrix this i command will calculate eigenvectors and the vectors will be stored as column matrix in the variable vector, and the values will store the eigenvalues as diagonal matrix.

So, for details regarding eig function, look in MATLAB help; this concludes our tutorial one we have gotten acquainted with MATLAB and its environment. We know where to look for help in case, we get stuck we know how to define and manipulate scalars vectors and matrices. In the next tutorial we will start actual programming we will the new and interesting topics, and we will see many examples see you in the next tutorial.