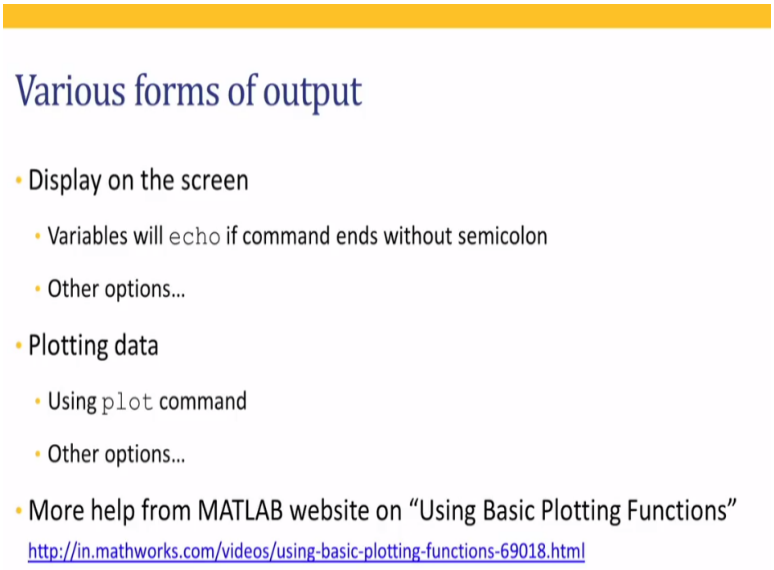
MATLAB Programming for Numerical Computation

Dr. Niket Kaisare
Department of Chemical Engineering
Indian Institute of Technology, Madras

Module No. #01
Lecture No. #1.5
Introduction to MATLAB programming – Plotting and Output

Hello and welcome to MATLAB programming for numerical computations. We are at the end of module 1 this module we are covering introduction to MATLAB programming in lecture 1.5. We are going to cover plotting and output using MATLAB.

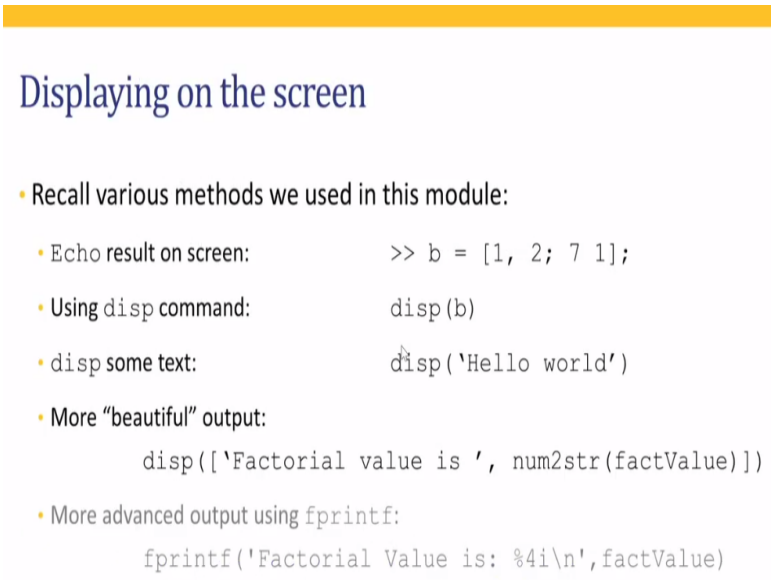
(Refer Slide Time: 00:31)



We have seen various forms of output in the first 4 lectures of this particular module. And what we have seen is that when we do not end of particular line with the semicolon, the variables will echo on this screen. That is one of the ways of displaying on the screen the other way that we had used in the previous lecture was, to use the command called disp or display. So that is the displaying on the screen that we are going to cover in this lecture.

I am also going to cover how to plot the data using what is known as the plot command. For more help on plotting we can go to MATLAB website called using basic plotting functions the link for which is given over here.

(Refer Slide Time: 01:21)



So let us now focus our attention to displaying on the screen and these are the various examples that we have taken on how to display on the screen. The first one is when you give a command b = 1, 2, 7, 1 with a semicolon , there will not be any echo but if you do not give this trailing semicolon, this particular result will be displayed on the screen. This is what we have seen in lecture 1.1 as well as multiple times.

The other way to display is using the command disp, display. It is going to display the contents of this variable b on the screen. The third way is to use display a more advanced version of it. And display can be used in order to display certain texts such as disp hello world will display this particular text on the screen. And that idea can be used to make a more beautiful output. So let us go at MATLAB and look at this.

(Video starts 02:25) We have this b = 1, 7 say 1, 3 close this and press enter. I am going to get this as an echo. So this is what we have seen multiple times. Same thing if I am going give this b, if this is going to display the contents of b but it is not going to give the initial part that is b equal to part that we see what here, when I will click on this b and press enter this is what I am going to see. Likewise if I give this hello world, it is going to display that is string as it is. So display the string hello world.

If we recall what we had done in the previous lecture using my factorial, this was what I had shown as a way to display the factorial value. What does that do? Let we just copy this and paste toward here and let us go over each on this. So this uses basically the same rules as we do of with respect 2 brackets.

The innermost bracket is what is going to be executed first. So this is the innermost bracket and let us looks at that. Let us removed rest of the stuff and let us look at this. This is the innermost bracket. So this particular guy is a string that says factorial value colon space. And this is a command with says num2str myFact.

So let us look at what numb2str does and for that we will do help num2str. Num2str converts a number into a string. So let us look at what say num2. Let us say myFact = 6!, so we can do that using prod 1 to 6 and that gives the value myFact as 720. Because I did not end this command with semicolon. I got the result myFact = 720.

What happens if I give, let us, we just clear the screen clc num2str myFact. If I give this command, let us see what happens. What happens is, again ans = 720. So how this ans differ from myFact, let us look at the work space over here, ans that you see of here is as a string that we know that it's a string. Because it has the apostrophe, the slash sign in the front and back which means this is not a number 720 but it is consists of 3 characters the first character is a 7, second character is 2 and third character is 0.

So that is the difference between ans and myFact. Let us look again at that particular display command control c and control v, the display command and let us look at the innermost bracket of the display command that we had done earlier. So now we have the square bracket which has one part of the string and then we are catenating the other part of the string. Let us look at the similar thing again. Let us say part 1 that is the first part and we say part 2 and let us see what happens.

What we should get is the words part space 1, part space 2. There is no space between 1 and p of the second part that is what result we expect. Press enter and that is exactly what we get. Now we

were replace part 2 with this ans value is 720 and we were to replace part 1 with factorial value equal to this.

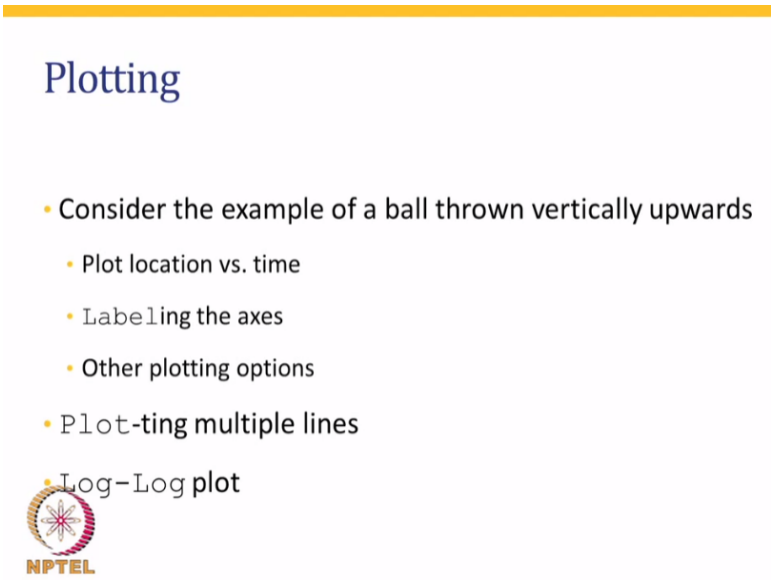
What we are going to get is, factorial value colon space and 720. That is also in slashes because this is a string. When I press enter, I am going to get factorial value 720. So, what I want to do is, replace that 720 with myFact. If I do this, I just replace it with myFact, what I am going to get is, I am trying to take a string and next to string I am trying to put a number.

When I try to do that what I get is, I do not get anything over here because that number is not converted into ASCII characters 720. I am just getting factorial value colon followed by some kind of a character that you probably cannot see over here. So instead what we need to do is use the num2str command to convert 720 into 720 as 3 characters and when I do that I am going to get factorial value colon 720.

Now I did not want this ans to be displayed like this. In order to do that I am going to give command with says disp sorry, disp and in circular brackets and when I give this I am just going to get factorial value 720 without this ans equal to factorial value 720. I will just clear the screen and type again. When I give this disp factorial value colon num2str myfact and I press enter, this is just displayed as is on the screen. (Video Ends: 08:52)

So this is what disp function is going to do for us. So, a more beautiful output can be created using disp and using numb2str. Sometimes we want to type out more complex things, we want to output more complex things on the screen and for that we want to use the function called fprintf. The syntax is fprintf is a similar to what we follow in C.

However, in this particular lecture series we are not going to cover fprintf for all our purposes. In this particular lecture series the disp function is going to be sufficient. However for the sake of completeness I have given more advanced output using fprintf as well.
(Refer Slide Time: 09:34)



So that covers the outputting on the screen part, let us now look at how to plot. So we will consider the example of the ball thrown vertically upwards, we will plot location versus time, we will label the axes and we will go the more other features of plotting.

(Video Starts: 09:56) So let us go back to MATLAB and the function that we are going to use for plot is a function called plot. So let us look at help for this. Help plot and this is what we are going to get. For now let us ignore the overloaded method part that is not of a concern for us right now. That is go right at the top and what does plot do? Plot, plots a linear plot x, y, plots vector y versus vector x. If x or y is a matrix then the vector is plotted against rows or columns of the matrix whichever line up.

So, plot is a very powerful way of plotting things. So plot x, y is going to plot y as a function of x. Let us take that example that we had done a couple of lectures earlier, which is of ball being throwing vertically upwards. This was the script that we had used for the ball that was being thrown vertically upwards. When run this particular script what we got was, we were getting this display at multiple times.

Recall what we have just done a couple of minutes back. When we talked about the disp command and when we used this disp command using multiple num2str, the result that we are going to get is, as we can show in the screen. At various times we are getting the values of the

location where the ball is. So now what we want to do is, we want to capture all of these in 2 vectors. Let us called these 2 vectors time and location.

So time equal to let us make the blank vector location equal to again a blank vector. For let us actually put as time = 0 and location = 0 because that is what we have starting off with. And at end of this, we will put t sorry, time = time s; t. What that is going to do is, at each time this while loops runs, the new value of time t is going to be catenated to this time vector time and likewise, we will do location = location: y.

And finally, we want to plot this and we will give the command plot time, comma location. Enter semicolon, save this and we execute this. We will clear the screen first and we will just execute ball vertical and this is going to plot. So, this is the trajectory of that particular ball as the ball goes from time 0 to time 4. Now what we want to do is we want to label these two axes the x axis is time and y axis is location.

We do that the x axis we will label it by using the command x label, time and time was in seconds enter and y label is location and location was in meters. And let us go to that figure again and now we this has been label time and it has been label location. Now let us say that what want to say increase the font size of this. Way to do that, just click on this arrow right click on font and we will be able to increase the font size to 14. Likewise we will increase the font size again to 14. And we can do this with the access as well.

We click on font and make the font size 14 and we will get this particular figure and now we have the font size increased to 14. Now, there are options to change the way of this particular line that was displayed and also. And the way to do that again is using the plot command plot. So this we can be done again by using plot command and the way to do is plot, our time, location, give MATLAB how to plot this and for example we wanted a dashed line which is red in color we will give --r that will plot a red dash line. And let us go and see this.

And what we now have is a red dash line as seen over here. Now instead if we wanted to have dash dot line which is green in color, we will change it to -. g that will plot a line that is green dash dotted in color. Instead now if we wanted in addition to that we also wanted, let us say to plot circles at the end of it a blue color line with circles far at the various locations, we will do - b followed by o and o will tell MATLAB that each data point we want to have a marker that is circled.

Press this, enter and let us see what happens at each data point. We now this circled plotted as we seen over here. We can also change the size or thickness of this line again by right clicking this, line styles we have different styles of lines as well as line width we can change this let us say the line width of 2.0. Again these markers, we can change the markers to two different types of markers as we can have plus sign, circle, star, square, diamond and so on. So, if we wanted a star, we can press this and we will get star and so on.

So, these are multiple ways of one plotting and manipulating the plot. Now let us say we wanted to plot multiple things on the same plot, how we can do that? So, for that lets us have a value of x let us say x = 0 to 5 in steps of 1. And let us say now we wanted to took plot sin x and cos x. So all we need to do plot x, sin x, - b that will be blue color line and x, cos x, - r sorry, let us say -- r that will be the red dash line.

When we will do that this is what we are going to get. I actually made a mistake. When I type x, I did x in the steps of 1 that is reason why we got plot in that particular manner if I had x = 0 to 0.1:5. And then I give the plot command and let us see what happens. And we now get a smoother plot. Now we have this particular blue line and this red line. Both of them are plotted on the same plot.

So the way to do this is when we have multiple x and y vectors, we can have them comma separated and we will be getting multiple plots. Another way to do this is also define a matrix let us say y = sin x: cos x. Let us type size of y. So y has 2 rows and 51 columns that basically means we have the first row is sin of x and second row is cos of x. And let say when we do plot

x, y we are going to get plot of the 2 lines and 2 lines will be plotted using a default way of plotting.

And default is the first line going to be blue dash and second line is going to be a green dash. So, this is the default way of plotting using matrices rather than using vectors. So you can plot multiple lines on the same plot. And let say now we wanted to plot third line over here but if let say what happens if we do plot x comma say sin x+ cos x what we will see this, we will get only one single line and that is because when we do that the previous plot gets overwritten.

And that is why we are getting a single line as seen over here. If we do not want the previous plot to get overwritten, we need to use the command hold on. So, let see how that works of close that plot. Let me plot x, y again as before. So these are the 2 lines blue and green lines. Now if I do not want to erase that plot, I will give command hold space on. And now when I plot x, sin x + cos x this is, what I am going to get. Let me go to this file again.

So these were the original 2 lines in blue and green color and this is third line that got plotted in blue color. Now let say we actually wanted to plot this in red color what we will do, just click on this and delete this particular line. And plot this say using -- or that will give me a red dash line on this figure. So, this is the way you can use the plotting commands. (Refer Slide Time: 20.50)

What we have done is, we have look plotting location versus time, we have looked at way to label the axis, we have to look certain other plotting options. And we have seen how to plot multiple lines using matrices as well as how to plot multiple lines using hold on. The final thing is how to plot log logplot.

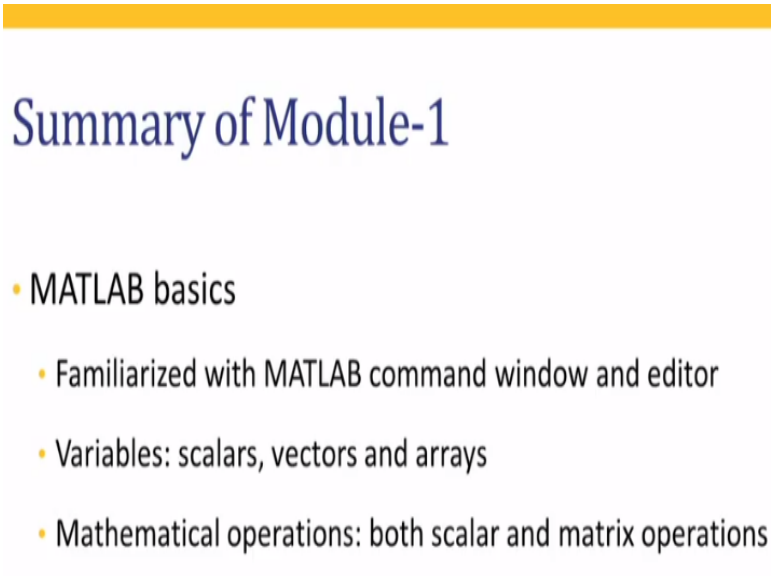
So for that again we have to go to MATLAB and we do not know how to plot a log logplot. So let us just do help log and see whether we can find out how to plot log logplot. At the end close to the end of the help we will see see also. And see also will give you commands that are related to plot. So, there are plot tools semi log x, semi log y, loglog and so on so forth. So, loglog is probably the one that we need to use.

So let us just click on loglog and look at that help loglog is nothing but a loglog scale plot log log you works in exact same way as plot except the scales are logarithmic. So let say I do loglog x, y and I am going to have x and sin x and, x and cos x on loglog plot and this is how I am going to get.

So what is the funny thing that is happened over here? Remember sin x and cos x have negative values what we happened when we plotted using a loglog is that the negative data that was ignored during the plotting. So only the positive data were plotted in the loglog plot. So what we have covered is the basic plotting functions? (Video Ends: 22:39)

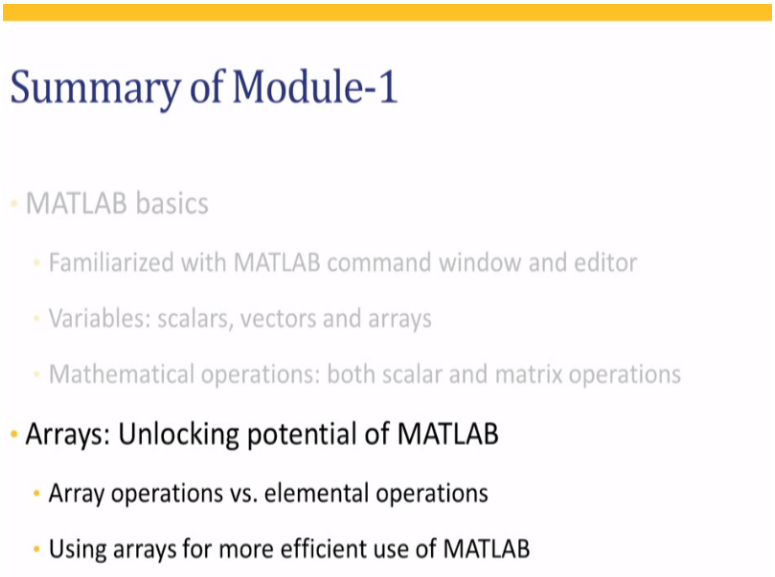
We have covered how to label the plot, we have covered how to plot multiple lines, how to use hold on in order to keep the previous lines, we have looked at how to change the fonts for axis labels as well as the axis titles, we have also look at plotting log log plot. In the same way we can plot semi log plot using either semi log x or semi log y command.

With that we come to the end of this particular lecture and also to the end of this module. Before I end, I will quickly recap in the next minute and show what we covered in this module. So, what we covered in this module was, introduction to MATLAB programming.
(Refer Slide Time: 23:22)



We looked at MATLAB basics familiarized ourselves with MATLAB command window and MATLAB editor. We looked at variables scalars, vectors and arrays. And became comfortable using array operations. We also looked at both scalar as well as matrix operation a multiplied by b using a * b as well as a.* b, where that was an element by element multiplication. So both of these types of mathematical operations we covered. We also covered trigonometric operations, covered operations such as sum, prod, cumsum, cumprod and so on so forth. So wide range of operations we have covered in lecture 1.2.

(Refer Slide Time: 24:07)

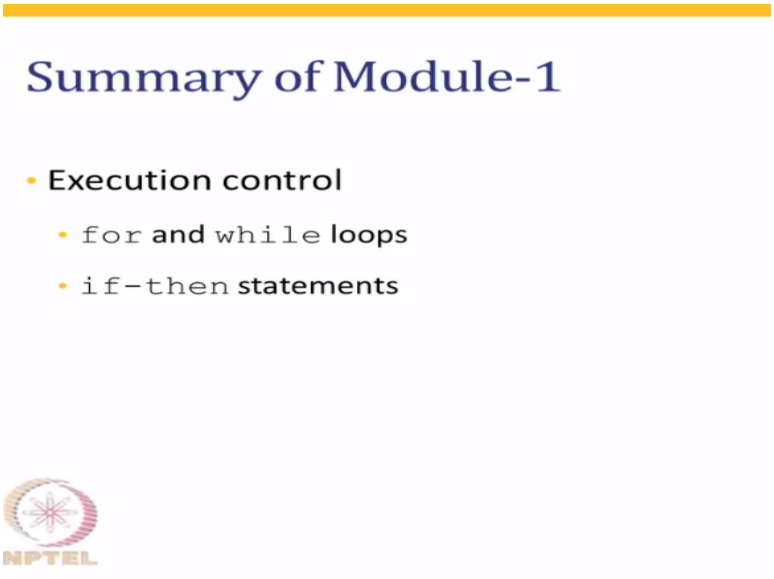


We have also look at arrays and use arrays to unlock the potential of MATLAB compared array operations versus elemental operations. And use arrays for more efficiently using MATLAB. Let me finally go to quick example for doing this.

(Video Starts: 24:34) Let us consider the example of factorial. A better way of calculating the factorial as we had seen rather than having all these number of lines, we can place all of this with just a prod command myFact = prod 1: n. That is exactly going to do what the entire function with the loop was going to do. So I have replaced that entire loop with this prod command and when I execute this that is myfactorial, I get the exact same result but in a more efficient way. (Video Ends: 25:13)

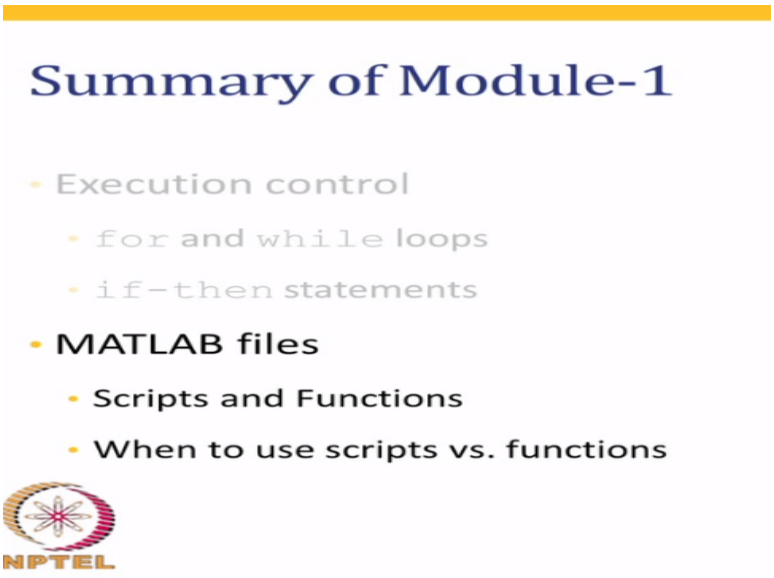
So, this was something that is one of the most important features of MATLAB. That it was able to work with arrays and we will need to be comfortably working with arrays to unlock the full potential of MATLAB. Of course this is going to come with lot of practices to get more and more using the MATLAB you will find we are using arrays in much more efficient way.

(Refer Slide Time: 25:36)



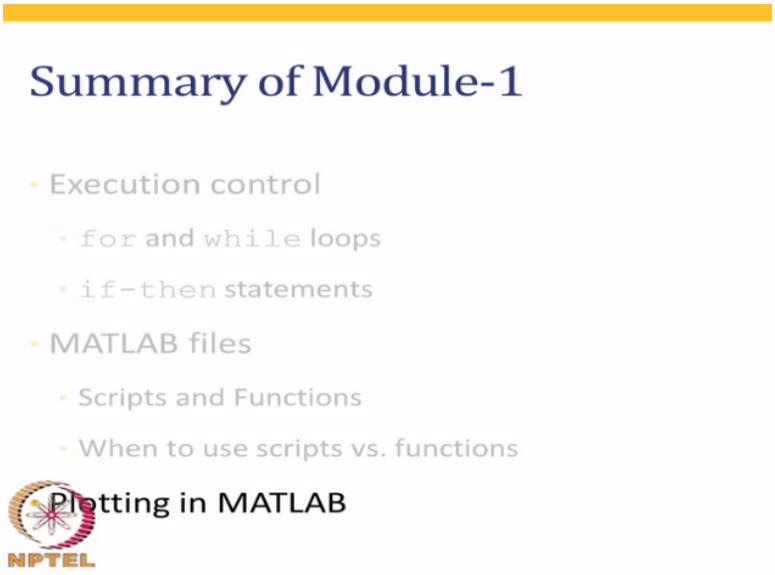
We then used look at execution control looked at for loop and while loop if then statements is something that we have not exactly covered but essentially if then statements are used when we want to compare certain condition, if that condition is met certain commands or executive, if it is not met then certain other commands and if there is statement or executed. Again, if you do help if we will get how to use them, if then commands

(Refer Slide Time: 26:09)



And then we looked at MATLAB scripts and functions.

(Refer Slide Time: 26:13)



We also in detail covered when to use scripts versus when to use functions and finally we will add plotting in outputting in MATLAB. So that we come to the end of the first week we come to the end of module 1. And the next module we are going to cover errors, error analysis and approximations and using numerical techniques so thank you and see you next week bye.